THE UNIVERSITY OF ASTON IN BIRMINGHAM

CONTINUOUS PHASE SYNCHRONISED DRIVES (FOR A ROD-MAKING MACHINE)

by DAVID ROBERT SEAWARD

Doctor of Philosophy, 1989

SUMMARY

Traditional high speed machinery actuators are powered and coordinated by mechanical linkages driven from a central drive, but these linkages may be replaced by independently synchronised electric drives. Problems associated with utilising such electric drives for this form of machinery were investigated. The research concentrated on a high speed rod-making machine, which required control of high inertias (0.01-0.5kgm^2), at continuous high speed (2500 r/min), with low relative phase errors between two drives (0.0025 radians).

Traditional minimum energy drive selection techniques for incremental motions were not applicable to continuous applications which require negligible energy dissipation. New selection techniques were developed. A brushless configuration constant enabled the comparison between seven different servo systems; the rare earth brushless drives had the best power rates which is a performance measure.

Simulation was used to review control strategies, such that a microprocessor controller with a proportional velocity loop within a proportional position loop with velocity feedforward was designed. Local control schemes were investigated as means of reducing relative errors between drives: the slave of a master/ slave scheme compensates for the master's errors: the matched scheme has drives with similar absolute errors so the relative error is minimised, and the feedforward scheme minimises error by adding compensation from previous knowledge.

Simulation gave an approximate velocity loop bandwidth and position loop gain required to meet the specification. Theoretical limits for these parameters were defined in terms of digital sampling delays, quantisation, and system phase shifts. Performance degradation due to mechanical backlash was evaluated. Thus any drive could be checked to ensure that the performance specification could be realised.

A two drive demonstrator was commissioned with 0.01kgm^2 loads. By use of simulation the performance of one drive was improved by increasing the velocity loop bandwidth fourfold. With the master/ slave scheme relative errors were within ±0.0024 radians at a constant 2500 r/min for two 0.01 kgm^2 loads.

Indexing terms:- BRUSHLESS SERVO, DRIVES, SYNCHRONISE, CONTROL,SIMULATION

DEDICATION

Dedicated to Phillippa for her support and friendship during the last three years

ACKNOWLEDGEMENTS

I would like to thank Dr R.C. Johnson for his excellent supervision of this project, and for always having an open door. My kind thanks to Mr B. James and Dr. A. M. Featherstone for their moral support.

Thanks to Mr. B. Harrison and Mr. L. Radford for the use of all their tools and expertise when down in the laboratory.

Special thanks must go to Mr. H. Arthur for his early industrial supervision of this project and for the development of the microprocessor controller which proved invaluable to the project.

Thanks must also go to the Science and Engineering Research Council and Molins plc for funding the research.

LIST OF CONTENTS PAGE

LIST OF CONTENTS (CONTINUED) PAGE

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (CONTINUED) Page

LIST OF SYMBOLS

a	Half the number of parallel paths in a machine
d	Diameter/m
d/dt	Differential operator/s^{-1}
dP_M	Motor power rate/Ws^{-1}
dP_L	Load power rate/Ws^{-1}
e_1, e_2	Absolute errors of axes to their demands/radians
i	Instantaneous current/A
n	Speed/rs^{-1}
p	Number of pulses per revolution from an Encoder
s	Laplace operator/s^{-1} [NB The Laplace operator is used to denote d/dt in block diagrams containing non-linearities for convenience]
t	Time/s
t_a	Acceleration/deceleration time/s
t_c	Time for increment/s
t_{co}	Optimum time for increment/s
t_s	Slew time/s
t_ϕ	Sampling time/$srad^{-1}$ ($\Delta T/2\pi$)
A	Gain of controller position loop/s^{-1}
B	Flux density/Wbm^{-2}
B_v	Viscous damping/Nm/$rads^{-1}$
D	Demand position/radians
E	Energy dissipation/J
E_o	Optimal energy dissipation/J
F	Force/N
F_c	Current loop gain/V/$rads^{-1}$
G	Gain of controller velocity loop/As^{-1}
G_f	Feedforward gain
G_m	Modulus of rigidity/Nm^{-2}
G_p	Position loop gain/$Arad^{-1}$ or s^{-1}
G_v	Velocity loop gain/A/$rads^{-1}$
I_a	Motor armature current/A
I_c	Current in conductor/A
I_p	Peak armature current/A
I_{rms}	Root mean square current/A
J_L	Load inertia/kgm^2
J_M	Motor inertia/kgm^2
J_T	System Inertia/kgm^2
K	Ratio of natural to sampling frequency
K_a	Current loop feedback attenuation/A/$rads^{-1}$
K_{bc}	Motor brushless configuration constant
K_d	Motor damping factor/Nm^2W^{-2}
$K_{e\phi}$	Motor back emf constant per phase/V/$rads^{-1}$
K_m	Universal motor constant/NmW^{-1}
K_p	Profile constant
K_s	Torsional Spring constant/$Nmrad^{-1}$
K_t	Total motor Torque constant/NmA^{-1}
$K_{t\phi}$	Motor Torque constant per phase/NmA^{-1}
L_a	Length of wire in field/m
L_s	Active torsional shaft length/m
L_ϕ	Motor armature inductance per phase/H
M	Mass/kg
N	Gear ratio
N_o	Optimum gear ratio
N_p	Number of pole pairs

LIST OF SYMBOLS (CONTINUED)

P	Power/W
R	Resistance/Ω
R_m	Motor Regulation/W^2Nm^{-2}
R_ϕ	Motor armature resistance per phase/Ω
R_{TH}	Thermal resistance/$^oCW^{-1}$
T	Torque/Nm
T_f	Frictive torque/Nm
T_g	Motor generated torque/Nm
T_L	Load torque/Nm
T_p	Peak torque/Nm
T_{rms}	Root mean square torque/Nm
T_Δ	Temperature rise/^oC
ΔT	Encoder pulse count sampling time/s
V_{bus}	Voltage of the dc bus/V
V_{emf}	Back emf/V
X, Y	Demand and output signals
Z_a	Total number of conductors
Z_s	Effective number of conductors in series
α	Acceleration/$rads^{-2}$
α_L	Load acceleration/$rads^{-2}$
α_M	Motor acceleration/$rads^{-2}$
β	Current loop gain
γ	Normalisation constant
ζ	Damping factor
η	Efficiency
θ_e	Position/electrical radians
θ_L	Load position/radians
θ_M	Motor position/radians
λ	Normalisation constant
π	Constant pi $\approx$ 3.1416
τ	Time constant/s
τ_c	Current sensing circuitry time constant/s
τ_e	Electrical time constant/s
τ_m	Mechanical time constant/s
ϕ	Flux per pole/Wb
ψ	Normalisation constant
ω	Speed or frequency/$rads^{-1}$
ω_b	Bandwidth/$rads^{-1}$
ω_{bm}	Bandwidth with motor inertia only/$rads^{-1}$
ω_{bt}	Bandwidth with motor and load inertias/$rads^{-1}$
ω_{in}	Demanded velocity/$rads^{-1}$
ω_L	Load speed/$rads^{-1}$
ω_M	Motor speed/$rads^{-1}$
ω_{max}	Unconstrained maximum velocity/$rads^{-1}$
ω_n	System natural frequency/$rads^{-1}$
ω_{out}	Output velocity/$rads^{-1}$
ω_p	Constrained peak velocity/$rads^{-1}$

Common Subscripts

1,2,3	Represents phases 1,2 and 3
approx	Approximate value of
cont	Continuous
L	Load
M	Motor
p	Peak
o	Optimum
rms	Root mean square

Abbreviations

ac	Alternating current
dc	Direct current
B/D	Bandwidth/ Damping
EMI	Electro-magnetic interference
P/I	Proportional/ Integral
PID	Proportional Integral and Derivative control
PLC	Programmable Logic Controller
PWM	Pulse Width Modulation
RMS	Root mean square
SERC	Science and Engineering Research Council
SPP	Specially Promoted Programme

CHAPTER 1

1.0 INTRODUCTION

1.1 GENERAL

The research project described within this volume concentrates upon the design, selection and application of electric servo systems for position control. By a servo system is meant the combination of a load, electric motor, power amplifier and closed loop controller designed to control output position (of the motor shaft) within defined error bounds with respect to a demand signal. In particular control of the relative error between a number of independent servos was required to within predetermined limits.

The research was funded as part of the Science and Engineering Research Council's (SERC) Specially Promoted Programme (SPP) into high speed machinery and by Molins plc. It was a collaborative initiative between industry and academe aimed at solving a particular industrial problem.

The engineering problem faced was to attempt a retro-fit replacement for a process control machine. Part of the conventional mechanical transmissions used to coordinate its actuators were to be replaced by independent electric servos. The machine was known as a rod-maker and it processed paper and tobacco to form cigarette rods. The problem was chosen as one of the most onerous available within Molins, as it would give scope for the investigation of present technological limits. The independent servos were required to run at a high continuous speed with relatively slow acceleration at start up and stop.

With respect to earlier position servo work the research described investigated some new problems: the inertia of the load was very high (0.01-0.5 kgm^2) when compared to the motor inertias, the required control speed range was also large, with a high top speed (0-3000r/min) and the positional accuracy required was near the limit of encoder resolution (3-4 counts of an 8000 line per revolution encoder).

The work therefore concentrated on developing design methodologies for phase (position) synchronism of drives with particular reference to Molins' rod making high speed machine. Thus the major problem to investigate and overcome was one of accurate position control rather than efficiently controlling energy flow to and from the load by the motor as in the majority of other position servo problems.

1.2 THE SERC SPECIALLY PROMOTED PROGRAMME

The research was funded as part of the SERC SPP into high speed machinery. The programme was initiated by government to aid industrial research and development. Initially the SPP into high speed machinery had two phases. Phase one would address particular industrial engineering problems and would discover more fundamental underlying topics for research which would be addressed in phase two. The research described was a phase one project.

The phase one projects were set up such that an academic institution (Aston University) and an industrial company (Molins plc) would collaborate to try to solve a fundamental problem the company had encountered on the development of high speed machinery.

1.3 SERVO DRIVE OPERATING MODES

A servo system monitors the condition of an output variable by comparing it to an input command. The system attempts to adjust the output such that there is zero error. The most common output quantities used as controlled variables are position, speed or torque. For the particular application position is of interest, but as will be explained later the other two variables must also be effectively controlled.

Servo systems generally operate in one of two operating modes: incremental or continuous.

The incremental mode of operation is where the demand changes its value in discrete steps with a static condition between steps (dwell). The continuous mode is where the command variable remains constant or changes linearly. Some systems are required to operate in a dual or hybrid mode which is a mixture of the two previous modes. Newton [52] further split the incremental modes into class I and class II; class I was a series of infrequent major position changes each followed by a long dwell, whereas class II was where an extended number of moves and dwells were required in rapid succession.

The above definitions are too general when considering position servos. An incremental mode is usually considered as a rapid repetitive demand often with a dwell, even if the demand is continuously changing. A slow or zero changing demand is then considered to be continuous mode. The distinction therefore becomes blurred as "rapid" is not well defined.

The industrial application that will be described in section (1.5) will require a continuous mode position servo.

1.4 SIMULATION

Simulation is a good design aid and was used extensively in the work described. A simulation language is a design tool which enables concepts to be tested before prototypes are built, or to predict how a system will react to parameter changes. Analogue simulators used operational amplifiers to simulate systems whilst simulation languages use powerful microprocessors and numerical integration routines. It is possible to simulate a system using a standard programming language and the correct use of integration algorithms but this is complex. A simulation language allows the programmer to concentrate upon mnemonics corresponding to blocks in a block diagram rather than the integration algorithms. Once the model is defined the way in which it is exercised (parameter variations) is separate from the model derivation which eases analysis.

The advanced continuous simulation language [1], Acsl (often pronounced "axle") was designed for modelling continuous systems described by time dependent, non-linear differential equations and/or transfer functions. Acsl simulation of a system can be split into either defining, or, exercising the model. When defining the model the system must be broken down into a number of functional groups defined in terms of Laplace Transforms, integrators, limits or tables of inputs to outputs which must be programmed in Acsl mnemonics. Once defined simulation tests can be run on the model by defining the parameters to be monitored or plotted, and parameters may be changed without redefining the model so their effect may be monitored.

This clear separation of simulation into two parts follows the structure established by the Technical Committee on Continuous System Simulation Languages in 1967, for efficient simulation.

The model is entered into Acsl in the following form:-

```
PROGRAM
        INITIAL
                Statements performed before the run begins
        END
        DYNAMIC
                DERIVATIVE
                Statements needed to calculate derivatives ie. The dynamic model
                END
                DISCRETE
                Statements defining sampled data systems
                END
                Statements executed every communications interval
        END
        TERMINAL
                Statements executed at the end of the simulation
        END
END
```

The program is based upon FORTRAN and supports most of its commands, with its own special "macro" routines. Acsl programs therefore resemble Fortran in style and layout, with mnemonics that make programs readable.

1.5 THE PARTICULAR ENGINEERING PROBLEM

A high speed machine designed to control a process will have numerous actuators to manipulate, form and combine components and materials to manufacture a finished product. Typically the actuators will be powered and co-ordinated from a central prime mover via mechanical transmissions. Complex actuator motions and co-ordinations are formed by utilisation of cams, gears, geneva mechanisms, four bar linkages etc.

This project was concerned with a requirement for Molins' new breed of rod making machines, which will use phase synchronised drives. A rod making machine forms cigarette rods from tobacco and paper. Previously a single variable speed main motor drove through precision mechanical transmissions to power the sub-assemblies within the machine. To increase flexibility, reduce acoustic noise, and reduce costs it is aimed to "de-couple" existing mechanical drives and use independently driven sub-assemblies. By modularising the design approach one is able to reduce design times and the whole design process is freed from the constraints of the connecting mechanisms which often dictated machine layout.

A rod maker schematic is shown in figure (1.1); the tobacco is fed down into the hopper where it is carded before being transferred by pneumatic systems onto a moving tape. The tobacco falls onto the garniture tape which holds the paper. A continuous rod is formed, glued and has the logo printed. The continuously formed rod is advanced by the garniture drive through the arc of a rotating sickle knife mounted on the cut-off drive. The knife passes through a narrow slit in a rotating ledger supporting the rod during cutting. If at the moment of cut the motion of the ledger is not in correct synchronism with the cut-off the knife may fail to pass through the gap and be broken, which must be avoided. Similarly, incorrect synchronism of the garniture to the cut-off would cause the rod to be badly cut, in the wrong place or of unequal successive lengths. The drives may be synchronised at 10% full speed before either the rod or the ledger is advanced into the arc of the rotating knife, and the machine is then run up to full speed over a five second ramp. As a concession the original five second ramp was extended to ten seconds.

There are many axes on a making machine, and not all would receive the same benefits from de-coupling. The major benefits would come from de-coupling the cut-off, ledger and garniture but the initial exercise was to synchronise the worst case which was to keep the ledger and cut-off to within 0.0025 radians during a ten second ramp from 250 r/min to 2500 r/min, which corresponds to 10,000 cigarette rods per minute. The cut-off carried a 0.05 kgm^2 load and the ledger 0.01 kgm^2. This specification is shown in more detail in table (1.1).

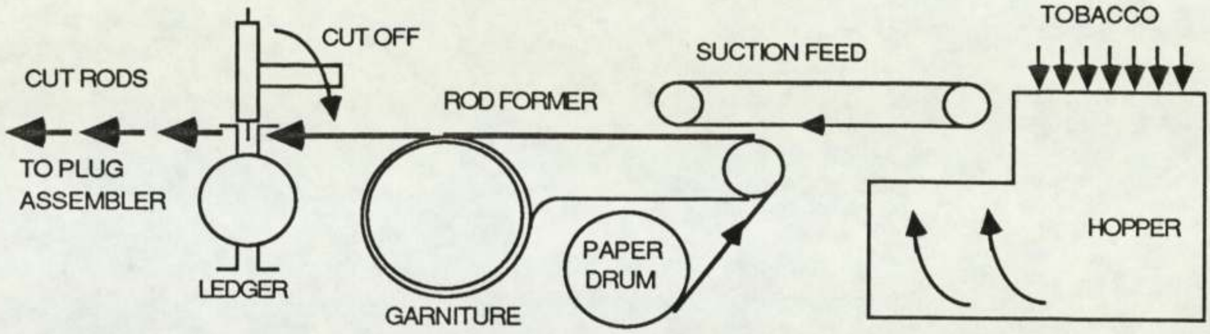


Figure 1.1 A rod making machine schematic

CUT-OFF (The drum holding the cut-off knife)

SPEED RANGE	- 250-2500 r/min
ACCELERATION	- 2250 r/min IN 5s (10s)
DISTURBING TORQUE	- UNKNOWN SINUSOIDAL TORQUE TWICE PER REVOLUTION
DRIVEN INERTIA	- APPROXIMATELY 0.244kgm^2 (possibly 0.5kgm^2)

LEDGER (The thin tube which supports the continuous cigarette rod during the cut)

SPEED RANGE	- 250-2500 r/min
ACCELERATION	- 2250 r/min IN 5s (10s)
POSITIONAL ACCURACY	- 0.0025 RAD (8.6 MINUTES) WITH RESPECT TO CUT-OFF
VELOCITY RATIO	- 1:1 WITH RESPECT TO CUT-OFF
DRIVEN INERTIA	- 0.01 kgm^2

GARNITURE (The main cigarette rod drive)

SPEED RANGE	- 300-3000 r/min
ACCELERATION	- 2700 r/min IN 5s (10s)
POSITIONAL ACCURACY	- NOT SPECIFIED BUT LESS STRINGENT THAN LEDGER
VELOCITY RATIO	- VARIABLE NOMINALLY SET AT 5:6 TO CUT-OFF
FRICTIVE LOAD	- APPROX. 2KW AT 2500r/min
DRIVEN INERTIA	- 0.05 kgm^2

Table 1.1 The Molins' Specification

1.6 AIMS OF THE RESEARCH

The research aims were to address the problems and obtain solutions to enable high accuracy synchronisation between high inertial loads running at a continuous high speed. The present limits of performance were to be investigated and increased if possible.

The problem posed in section (1.5) was chosen as a very severe case for retro-fitted electrical phase synchronised drives. It gave the opportunity to probe the present technological limits of drives and controllers available. The aim was not only to identify present performance limits but to obtain accurate methodologies for identifying performance from any future systems. Thus the factors limiting performance must be identified and methods of overcoming such limits proposed.

A demonstrator rig was to be commissioned on which to: develop theoretical approaches: prove any theoretical methodology and to act as a show-case for the present technological performance limits. Any such rig would require compatibility with Molins' existing equipment and must be able to be manufactured/assembled by Molins. For this reason and to save lengthy development time it was decided to buy-in as much equipment as was practicable. Thus the research concentrated upon application of technology rather than development work.

Brief analysis of the problem showed that the key components of any system would be the motor, the drive amplifier, the controller and any synchronisation controller. As the drive amplifier and the motor were the most complex elements to design and have manufactured, these would be purchased from vendors. Thus selection and comparison techniques for such systems were analysed and developed. Controllers are also available as proprietary equipment and these were also reviewed and purchased where necessary. Only equipment not readily available in the market place was designed and constructed.

1.7 OVERVIEW OF PREVIOUS WORK IN THE FIELD

During the literature review no author was located who had worked on a similar problem to that posed. The work does however overlap upon the servo drive field which has been thoroughly researched in the past in a number of particular areas. Selection of drives and comparison of drives for incremental applications has been investigated by a number of authors. The more prolific writers in the area are Tal [68-76], Persson [53-57] and Tomasek [77-84] who were all based in the United States of America.

Tal wrote two notable books in the fields of servo motor selection and control by microprocessors [71], and of multimotor incremental drive system applications [68]. His technical publications concentrate upon phase locked servo systems [69,73-76], upon servo system modelling [70,72,74] and upon drive selection for incremental applications [70,71]. Tomasek in the early 1980's concentrated upon optimisation of brushless dc drive motors and amplifiers [77,78,80,81,83], upon feedback devices [79] and upon load identification techniques [82]. Persson worked closely with Electrocraft Ltd. on brushless motor designs. His technical publications concentrate upon modelling, performance prediction and improvement in performance of brushless dc motors.

Two major annual conferences were also identified as having the most relevant papers. These were the annual international motor conference (MOTOR-CON) [15,36,48,53-55,74,77,81] and the annual symposium on incremental motion control, systems and devices [4,26,30,52,57,69,75,80,82-84]. The latter concentrates upon incremental applications which does not directly map onto the continuous problem posed.

A further problem found during the literature survey was one of company confidentiality. Much of the "front-edge" development of servo systems is being performed by company research and development establishments rather than in academe. As a result up-to-date results are not published unless in the form of advertisements for new products.

1.8 OVERVIEW OF THESIS FORMAT

The work has been split into nine chapters with relevant appendices. The chapter order and content was selected to give a clear progression through the work, and as such no chronological order has been maintained.

This chapter has introduced the problem faced and the research aims.

Chapter 2.0 details the simple mathematical representation of a brushed dc servo motor that will be used and developed throughout the thesis. This chapter also reviews the commonly used methods for selecting and comparing brushed servo systems.

Chapter 3.0 develops the mathematical drive representation of chapter 2.0 to include brushless dc machines which are now the standard choice for high performance servo systems. A rigorous procedure for comparing drives is presented and demonstrated upon seven drive systems.

The work of chapters 2.0 and 3.0 develop methods for selecting a "best" electrical machine and amplifier. Once selected the position of the motor when connected to the load must be controlled. Thus chapter 4.0 concentrates upon controller strategies available. A number of controller topologies are compared for the particular application, and the primary reasons for using the controller that was developed are given. When considering relative error between drives there are a number of "local control" schemes that may be employed to "enhance" performance and these are discussed.

Using the first four chapters it is possible to obtain the highest performance servo drive and controller to maintain low relative errors between axes. Unfortunately no knowledge of the errors is available, so uncertainty exists as to whether a selected system will maintain errors within specification. Chapter 5.0 introduces a simulation method to give an indication of required performance to meet a specification in terms of velocity loop bandwidth and position loop gain. This then forms the basis for the remaining work of chapter 5.0 and for chapter 6.0 where methods of obtaining the maximum servo performance are presented. The limits have been found to be due to sampling delays, encoder feedback, quantisation, backlash in gearing and natural system resonance exasperated by high load/ motor inertia ratios.

Chapter 7.0 and chapter 8.0 contain the practical work carried out. The methods which led to a substantial improvement in performance from a proprietary servo system by the use of simulation is described in chapter 7.0. Chapter 8.0 describes the demonstrator rig equipment that was developed and presents measured results, which are presented as position errors along a profile, expressed in terms of encoder counts. A discussion of these results is given, drawing comparisons with the earlier theoretical approach.

Chapter 9.0 contains the suggestions for future work and draws the main conclusions from the work.

The appendices contain all information not pertinent to the main flow of the discussion. These include details of the unsuccessful work carried out early in the project on an alternative drive system, of the simulation suite of programs developed for the successfully implemented drive system and of the circuitry developed to monitor the system performance.

CHAPTER 2

2.0 REVIEW OF TRADITIONAL MOTOR SELECTION TECHNIQUES

2.1 INTRODUCTION

Motor selection or sizing techniques have evolved over many years, but unfortunately there is no widely accepted technique available. Often drive systems are purchased and tested in the laboratory or first applied to actual systems using ad hoc methods. The majority of servo systems employed in the past have been brushed dc machines, so that, this chapter concentrates upon selection techniques for these machines, and a method for altering the procedures to take account of other drive types is shown in chapter 3.0.

Prior to discussing the relevant techniques the simple block diagram representation of a dc machine is introduced, with the symbols that will be used throughout. This simple representation has been used for techniques developed in the remaining chapters.

A selection procedure begins by ensuring the motor has sufficient torque at the required speeds and that it will not overheat. Once drives capable of driving the load have been established various factors of merit may be used to compare drives: damping factor, regulation, universal motor constant, electrical and mechanical time constants, torque to inertia ratio and power rate. All are introduced and discussed. Tal [71] observed that the limiting factor when selecting drives systems was overheating of the motor due to energy dissipation. Thus, applicable mainly to incremental point-to-point applications, optimisation techniques were developed to select motors, gear ratios and velocity profiles to reduce energy utilisation.

2.2 SIMPLE REPRESENTATION OF A DC MACHINE

Detailed analysis of dc machines is available in many electro-technology textbooks (Hindmarsh [35]). The following analysis will therefore be superficial. A typical servo system is assumed to have a machine which is permanently excited such that the air-gap flux is constant, and that the motor torque is directly proportional to both the air-gap flux and the armature current. Therefore the generated torque is:-

$$T_g \quad \alpha \quad \phi . I_a \qquad \ldots .(2.1)$$

$$= \quad K_{t\phi} I_a \qquad \ldots .(2.2)$$

This relationship will be briefly derived and related to the back emf constant, V_{emf} which is:-

$$V_{emf} = 2N_p.\phi.Z_s.n \qquad \ldots .(2.3)$$

where N_p = Number of pole pairs
ϕ = Flux/pole/Wb
Z_s = Effective number of conductors in series
n = Speed/ rs^{-1}

Also Force on a conductor, F,

$$F = B.L_a.I_c \qquad \ldots .(2.4)$$

where B = Flux Density/Wbm^{-2}
L_a = Length of wire in the field/m
I_c = Current in conductor/A

If the diameter of the rotor is d then the area per pole is $\pi dL_a/2N_p$ so the total flux per pole is

$$\phi = B.\pi.d.L_a/2N_p \qquad \ldots .(2.5)$$

$$\text{or} \quad B = \phi.2N_p/(\pi.d.L_a) \qquad \ldots .(2.6)$$

From (2.4)

$$F = \phi.2N_p.L_a.I_c/(\pi.d.L_a)$$

$$= \phi.2N_p.I_c/(\pi.d) \qquad \ldots .(2.7)$$

If there are 2a parallel paths, it follows that there are $I_a/2a$ amps per conductor, and Z_a conductors, so torque is:-

$$T = \phi.2N_p.I_a.Z_a/(2\pi.2a) \qquad \ldots .(2.8)$$

$$\text{or} \quad K_{t\phi} = \phi.2N_p.Z_a/(2\pi.2a) \qquad \ldots .(2.9)$$

If there are Z_a conductors, there must be $Z_a/2a$ conductors in series, that is, $Z_s = Z_a/2a$ thus the back emf becomes

$$\text{Total } V_{emf} = 2N_p.\phi.Z_a.n/2a \quad \text{Volts} \qquad \ldots .(2.10)$$

$$= 2N_p.\phi.Z_a/(2a.2\pi) \quad \text{Volts/rads}^{-1} \qquad \ldots .(2.11)$$

Thus

$$V_{emf} = K_{e\phi}.\omega \text{ where } K_{e\phi}=2N_p.\phi.Z_a/(2a.2\pi) \qquad ....(2.12)$$

and by comparison with equation (2.9)

$$K_{t\phi} = K_{e\phi} \qquad ....(2.13)$$

This is true for any phase of a machine or for a brushed machine which is single phased.

The voltage seen by the armature windings of a dc machine is the applied voltage less the back emf voltage which is proportional to velocity, equation (2.12). The current seen in the windings is dependent upon the armature resistance and inductance (capacitive effects neglected), thus:-

$$V_{bus} = I_a.R_\phi + L_\phi\frac{dI_a}{dt} + K_{e\phi}.\omega \qquad ....(2.14)$$

where V_{bus} = Voltage applied to the motor terminals/V
R_ϕ = Armature resistance per phase/Ω
L_ϕ = Armature inductance per phase/H

If a purely inertial load with a frictive element is assumed then the load equation becomes

$$T_g = T_f + J_T\frac{d\omega}{dt} + T_L + B_v.\omega \qquad ....(2.15)$$

where T_g = Torque generated by the motor/Nm
T_f = Coulomb friction/Nm
T_L = Load torque/Nm
B_v = Viscous friction/Nm/rads^{-1}
J_T = Total driven inertia/kgm^2

The total inertia referred to the motor shaft, J_T, is made up of the load and the rotor inertias ($J_M + J_L/N^2$), if 1:N gearing is used (the inertia of the gearbox is neglected).

By using the above relationships and Laplace transforms a block diagram of the system may be created as shown in figure (2.1) which has proportional current feedback control included (with feedback gain $1/K_a$ and feedforward gain of F_c); this is standard practise.

(K_{bc} is used in chapter 3.0 to deal with brushless systems and should be assumed to be 1 for brushed systems, that is neglect it for the moment. Thus K_t is used, where $K_t = K_{t\phi}K_{bc}$.)

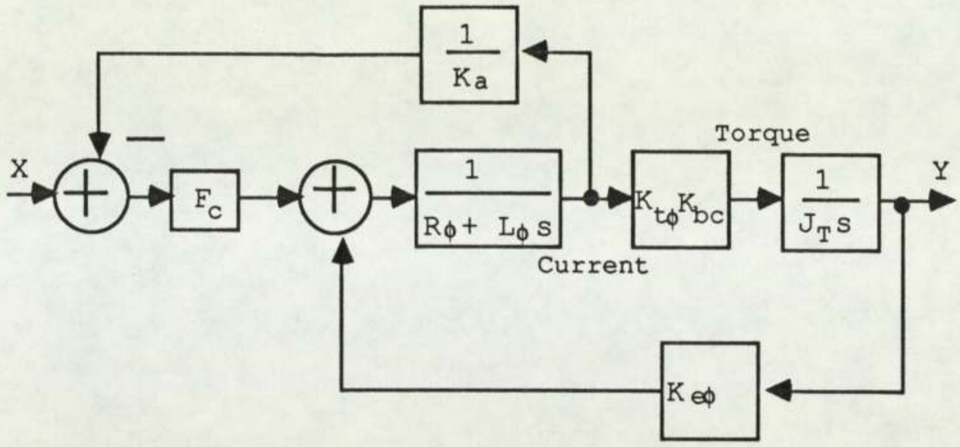


Figure 2.1 A dc machine block diagram

The transfer function of the system is:-

$$\frac{Y(s)}{X(s)} = \frac{F_c/K_{e\phi}}{\frac{J_T L_\phi s^2}{K_t K_{e\phi}} + \left(\frac{J_T F_c}{K_a K_t K_{e\phi}} + \frac{R_\phi J_T}{K_t K_{e\phi}}\right)s + 1} \qquad \ldots.(2.16)$$

The current loop dc gain is defined as $\beta = F_c/R_\phi K_a$ and thus the transfer function becomes:-

$$\frac{Y(s)}{X(s)} = \frac{F_c/K_{e\phi}}{\frac{J_T L_\phi s^2}{K_t K_{e\phi}} + \frac{R_\phi J_T (1+\beta)}{K_t K_{e\phi}} s + 1} \qquad \ldots.(2.17)$$

This second order system has undamped natural frequency, ω_n, and damping factor, ζ, given by:-

$$\omega_n = \sqrt{\frac{K_t K_{e\phi}}{J_T L_\phi}} \qquad \ldots.(2.18)$$

$$\zeta = \frac{R_\phi}{2}(1+\beta)\sqrt{\frac{J_T}{L_\phi K_t K_{e\phi}}} \qquad \ldots.(2.19)$$

The effect of β is to alter the resistance, R_ϕ, to an effective value $R_\phi(1+\beta)$. Thus current feedback alters the damping factor of the system, leaving the undamped natural frequency as before.

The mechanical time constant of such a system characterises the speed increase for an input voltage step, and is defined as:-

$$\tau_m = \frac{R_\phi J_T (1+\beta)}{K_t K_{e\phi}} \qquad \ldots.(2.20)$$

and the electrical time constant characterises the current increase in the motor coils for a voltage step, and is defined as:-

$$\tau_e = \frac{L_\phi}{R_\phi(1+\beta)} \qquad \ldots.(2.21)$$

If we now consider,

$$(s\tau_m+1)(s\tau_e+1) = \frac{J_T L_\phi s^2}{K_t K_{e\phi}} + \left(\frac{L_\phi}{R_\phi(1+\beta)} + \frac{R_\phi(1+\beta)J_T}{K_t K_{e\phi}}\right)s + 1 \qquad \ldots.(2.22)$$

It can be seen that this is approximately the denominator of equation (2.17) where:-

$$\omega_{napprox} = \sqrt{\frac{K_t K_{e\phi}}{J_T L_\phi}} \qquad \ldots.(2.23)$$

$$\zeta_{approx} = \frac{R_\phi(1+\beta)}{2}\left(1 + \frac{L_\phi K_t K_{e\phi}}{R_\phi^2(1+\beta)^2 J_T}\right)\sqrt{\frac{J_T}{L_\phi K_t K_{e\phi}}}$$

$$= \zeta + 1/4\zeta \qquad \ldots.(2.24)$$

or $\zeta >> 1/2$ for an accurate approximation. For 10% accuracy $\zeta>1.6$ and for a 1% accuracy $\zeta>5$.

It should be noted that the approximation does not require K_t to be equal to $K_{e\phi}$. For a current source $\beta\Rightarrow\infty$, $\tau_m\Rightarrow\infty$, $\tau_e\Rightarrow 0$, and $Y(s)/X(s) \Rightarrow K_t K_a/sJ_T$. Also for a voltage source $\beta\Rightarrow 0$.

2.3 SELECTING DC DRIVE SYSTEMS

2.3.1 INTRODUCTION

When purchasing any item, the selection of that item depends upon many various, and often non-quantifiable, factors such as, fitness for job, initial costs, running costs and life-span. The selection of a servo drive system is no exception, but it has the added complication that it is often not clear whether a given drive will be "fit for the job", that is, whether it will be capable of meeting the specification. Often drives are selected ad hoc and tested for suitability but this can be costly if either the drive cannot meet the specification and has to be replaced or if it has too much installed power such that the initial capital costs and running costs are higher than necessary.

This section reviews the standard methods of selecting drives and some optimisation techniques that have been applied to this problem. Often selection is carried out by comparison between drives by use of a figure of merit and these are defined with explanation of their use and relevance.

2.3.2 GENERAL MOTOR SELECTION

The first stage in any drive selection process is to identify as accurately as possible the worst case load requirements. Numerous papers and text books detail calculations that may be made to identify a load's inertia, friction, viscous friction and move requirements Tal [71], Electrocraft [25], Berger [9], Tomasek [82], Hopper [36] and Barber[5]. If the profile of the movement has not been specified then one of the optimised profiles of section (2.3.5.1) can be utilised (a trapezoidal motion may be used for a first approximation).

The next stage is to construct the torque speed curve for the load and to compare this with motor torque/speed curves (taking account of an approximate value for gear ratio where appropriate), noting any peak/intermittent torque requirements. The motor must always be capable of delivering the required torque.

Next the motor temperature and power supply requirements should be identified by finding the root mean square torque, Barber [7] (remembering any motion dwells).

$$T^2_{rms} = \left(1/_{\text{time period}} \int_0^{\text{time period}} T^2 dt \right) \qquad \ldots.(2.25)$$

and since $I_{rms} = T_{rms}/K_t$ and power $P = I^2_{rms}R_\phi$ the temperature rise can be calculated from $T_\Delta = R_{TH}P$ where R_{TH} is the thermal resistance.

Thus motors can be approximately selected that would be capable of supplying torque without overheating. The following sections (2.3.3)-(2.3.7) describe methods for selecting an optimum servo system from between a number of candidates capable of driving the load along its profile.

2.3.3 DAMPING FACTOR, REGULATION AND THE UNIVERSAL MOTOR CONSTANT.

These are essentially the same factor of merit: Damping factor, K_d, is defined as $K_t.K_{e\phi}/R_\phi$, that is the product of torque and back emf constants divided by the phase resistance: regulation, R_m, is the reciprocal of the damping factor: the universal motor constant, K_m, is the square root of the damping factor.

It may be shown that these factors are dependent upon motor size, Bartlett and Shankwitz [8], and that the gradient of a no-load servo motor speed curve is the damping factor, so a lower gradient curve relates to higher torque availability at high speed for a fixed terminal voltage, [25]. Since $T_g = K_t.I_a$ and $I_a = (V_{bus} - K_{e\phi}.\omega)/R_\phi$, then:-

$$T_g = \frac{K_t.V_{bus} - K_t.K_{e\phi}.\omega}{R_\phi} \qquad \text{....(2.26)}$$

The gradient of which is the damping factor.

When drawing comparisons between servo drives, the highest performance will come from the drive with the highest damping factor or universal motor constant, or the lowest regulation.

For a brushed motor ($K_{e\phi}=K_{t\phi}, K_{bc}=1$) the universal motor constant can be re-expressed as:-

$$K_m = \frac{K_{t\phi}}{\sqrt{R_\phi}} = \frac{T_p}{\sqrt{I_p{}^2 R_\phi}} \qquad \text{....(2.27)}$$

Thus it represents the peak torque (or acceleration) capability of a drive divided by the square root of its peak copper losses. It is a measure of efficiency in terms of torque squared per unit loss. This figure may be used as a first selection point by identifying the peak torque to peak power ratio requirements of the load and choosing the motor rating accordingly, Fleisher [29].

2.3.4 ELECTRICAL AND MECHANICAL TIME CONSTANTS

Both the mechanical, equation (2.20), and electrical, equation (2.21), time constants should be small for a high performance system. The mechanical constant of a brushed machine may be represented by $R\phi J_T/Kt\phi^2$ and this figure of merit can be used to optimise the power used by a system, see section (2.3.5.4). These figures of merit tend not to be very useful since the effective resistance value used in the calculations is effected by any current feedback. Most servo drives available have very stiff current loops, that is, the drive acts like a current rather than a voltage source to the motor. This has the effect that the electrical time constant becomes negligibly small with reference to the mechanical one. Furthermore if the reflected load inertia on the motor shaft is many times bigger than the motor inertia then the mechanical time constant is dominated by the motor torque constant, and load inertia alone.

The product of the two time constants $\tau m \tau e = J_T.L\phi/KtKe\phi$ is the square of the natural frequency of the servo system. It is therefore of interest, as its value is unaffected by β, although the second order effect is masked by a large β acting on the 's' term of the denominator, see equation (2.17).

2.3.5.0 ENERGY AND POWER OPTIMISATION

The limiting factor for many conventional dc drive systems working for incremental applications has been the overheating of the armature winding on the rotor. In order to reduce heating the energy dissipation within the motor must be minimised; this also reduces running costs of a system. Optimisation may be carried out to minimise motor temperature but this can be misleading as the thermal behaviour of the motor can be modified by forced air or other cooling methods so Tal [71] suggested an optimisation technique by minimising energy dissipation within a motor.

In order to optimise energy, the energy dissipation within a system must be identified. For a dc system the energy losses are as shown in figure (2.2).

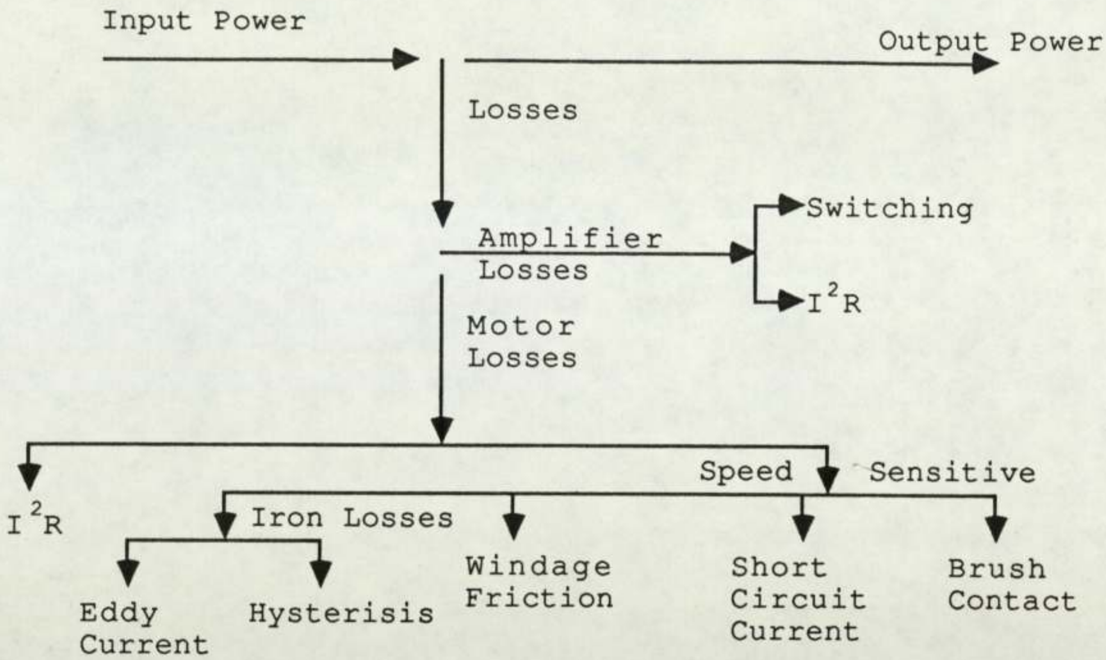


Figure 2.2 Energy Losses in a Conventional dc Machine

The amplifier losses will predominantly be from Joule $I_a{}^2R$ losses and from losses associated with switching power electronic devices.

Winding losses are the Joule $I_a{}^2R_\phi$ losses which are related to torque since $I_a \propto T_g$.

Brush contact losses are due to current flow between the brush and the dielectric film it creates on the commutator. This loss is minor and not evident in brushless drives.

Eddy or circulating current losses are produced in the armature as it rotates in the magnetic field whilst hysteresis loss is caused by resistance to magnetic domain boundary shifts. Both losses are proportional to speed and are lumped to form iron losses.

Short circuit losses are caused by the momentary shorting of armature windings as the brushes contact more than one commutator bar at a time. The loss appears as a viscous drag and is proportional to speed.

The motor losses create the temperature rise of the motor and they can be lumped together ($T_f + B_v.\omega$), thus:-

$$T_g = (T_f + B_v.\omega + T_L + J_T\frac{d\omega}{dt}) \qquad \ldots.(2.28)$$

and $I_a = T_g/K_t$, $V_{bus} = R_\phi I_a + K_{e\phi}\omega$. Therefore, power is:-

$$P = V_{bus}I_a = I_a(R_\phi I_a + K_{e\phi}.\omega) \qquad \ldots.(2.29)$$

$$= I_a{}^2R_\phi + \frac{K_{e\phi}}{K_t}.\omega(T_f + B.\omega + T_L + J_T\frac{d\omega}{dt}) \qquad \ldots.(2.30)$$

$$= I_a{}^2R_\phi + \frac{K_{e\phi}}{K_t}.\omega(T_f + B_v.\omega) + \frac{K_{e\phi}}{K_t}.\omega T_L + \frac{K_{e\phi}}{K_t}.J_T\omega\frac{d\omega}{dt} \qquad \ldots.(2.31)$$

where, $I_a{}^2R_\phi$ is the loss in the motor

$K_{e\phi}.\omega(T_f + B_v.\omega)/K_t$ is the frictive loss in the motor

$K_{e\phi}.\omega T_L/K_t$ is the power delivered to the load

$\frac{K_{e\phi}}{K_t}.J_T\omega\frac{d\omega}{dt}$ is the power required to accelerate the load

2.3.5.1 Optimum velocity profile

For incremental moves there will be an optimum velocity profile. Optimisation could be carried out on minimum peak speed, or minimum peak or average current, but for the majority of incremental applications the limit to performance is the motor armature heat dissipation. From the previous discussion the major contributor to heating is $I_a{}^2R_\phi$ losses. Consider a velocity profile $\omega(t)$ which moves the motor shaft through θ_M radians in t_c seconds, starting and finishing at rest.

Position is related to the velocity profile by:-

$$\theta_M = \int_0^{t_c} \omega(t)\,dt \qquad \ldots.(2.32)$$

$$T_g = I_aK_t = J_T\frac{d\omega}{dt} + T_L \text{ (neglecting motor and viscous losses)}$$

and $\text{Energy per step} = R_\phi \int_o^{tc} I_a(t)^2.dt \qquad \ldots.(2.33)$

Firstly T_L contributes a loss independent of velocity profile which is $R_\phi.T_L{}^2.t_c/K_t{}^2$

and the optimum velocity profile may be found (Tal [71]) to be:-

$$\omega_o(t) = 6\theta_M(t_c-t)t/t_c^3 \qquad \ldots.(2.34)$$

with an associated energy loss per step (neglecting the T_L contribution) of:-

$$E = 12.R_\phi.J_T^2.\theta_M^2/K_t^2.t_c^3 \qquad \ldots.(2.35)$$

The peak acceleration is $6\theta_M/t_c^2$ and the peak velocity is $3\theta_M/2t_c$. This may be compared to the worst case profile where the acceleration and deceleration rates are equal, that is, a triangular profile. In this instance the losses are $16.R_\phi.J_T^2.\theta_M^2/K_t^2.t_c^3$ and the peak acceleration is $4\theta_M/t_c^2$ and the peak velocity is $2\theta_M/t_c$. Of all the trapezoidal velocity profiles the best is a near optimum condition where the acceleration, deceleration and slew times are equal. In this instance the losses are $13.5.R_\phi.J_T^2.\theta_M^2/K_t^2.t_c^3$ and the peak acceleration is $4.5\theta_M/t_c^2$ and the peak velocity is $3\theta_M/2t_c$.

As all the energy equations are similar the profile constant is defined, K_p, such that for the optimum profile it is 12, for the triangular profile it is 16 and for the trapezoidal one it is 13.5. Also note that if J_T is fixed, that is, a fixed ratio between motor and load inertia exists, $E \alpha K_p.R_\phi.J_T^2./K_t^2$. Thus for a fixed step energy dissipation is proportional to the product of the profile constant, the load inertia and the mechanical time constant (for a brushed machine). Figure (2.3) shows the important figures that can be gained for the important velocity profiles.

The profile selection may ultimately be dependent upon an individual system's limiting factor, for example the triangular profile requires the lowest peak torque (acceleration) but the highest power requirements. Normally energy dissipation is the limiting factor so that the parabolic or sinusoidal profiles should be used unless the controller cannot create these in which case the trapezoidal profile should be used.

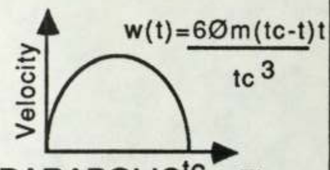
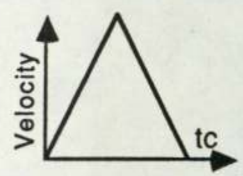
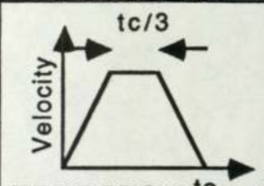
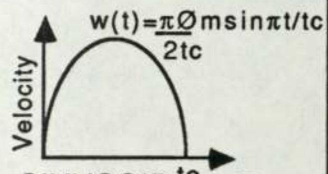
VELOCITY PROFILE	Motor Energy Dissipation per step/J	Ratio of Motor Energy Dissipation per step	Peak Acceleration required/ms $^{-2}$	Ratio of Peak Acceleration	Peak Velocity required/ms $^{-1}$	Ratio of Peak Velocity	Average current modulas/A	Ratio of Average current modulas	Ratio of Kinetic Energy for 1/2 cycle/J	Kinetic Energy for 1/2 cycle/J	Rms Current/A	Ratio of Rms Current/A
 PARABOLIC	$12\frac{R\phi J_T^2 \varnothing_m^2}{K_t^2\ tc^3}$	1	$\frac{6\varnothing m}{tc^2}$	1	$\frac{3\varnothing m}{2tc}$	1	$\frac{3\varnothing m J_T}{tc^2 Kt}$	1	$\frac{9J_T\varnothing_m^2}{8tc^2}$	1	$2\sqrt{3}.\frac{J_T\varnothing m}{Kt.tc^2}$	1
 TRIANGULAR	$16\frac{R\phi J_T^2 \varnothing_m^2}{K_t^2\ tc^3}$	1.33	$\frac{4\varnothing m}{tc^2}$	0.67	$\frac{2\varnothing m}{tc}$	1.33	$\frac{4\varnothing m J_T}{tc^2 Kt}$	1.33	$\frac{2J_T\varnothing_m^2}{tc^2}$	1.78	$4\frac{J_T\varnothing m}{Kt.tc^2}$	1.15
 TRAPEZOID	$13.5 \times \frac{R\phi J_T^2 \varnothing_m^2}{K_t^2\ tc^3}$	1.125	$\frac{9\varnothing m}{2tc2}$	0.75	$\frac{3\varnothing m}{2tc}$	1.00	$\frac{3\varnothing m J_T}{tc^2 Kt}$	1.00	$\frac{9J_T\varnothing_m^2}{8tc^2}$	1.00	$\sqrt{13.5} \times \frac{J_T\varnothing m}{Kt.tc^2}$	1.06
 SINUSOID	$\frac{\pi^4}{8}\frac{R\phi J_T^2 \varnothing_m^2}{K_t^2\ tc^3}$	1.01	$\frac{\pi^2\varnothing m}{2tc^2}$	0.82	$\frac{\pi\varnothing m}{2tc}$	1.05	$\frac{\pi\varnothing m J_T}{tc^2 Kt}$	1.05	$\frac{\pi^2 J_T\varnothing_m^2}{8tc^2}$	1.10	$\frac{\pi^2}{2\sqrt{2}}\frac{J_T\varnothing m}{Kt.tc^2}$	1.01

Figure 2.3 Comparison of velocity profiles

2.3.5.2 Gear ratio optimisation

Another element of a servo system that may be optimised is the gear ratio N.

$$J_T = J_L/N^2 + J_M \qquad ....(2.36)$$

and T_L is actually T_L/N, thus using the energy equation (2.35), and including the load loss component the energy per step is:-

$$E = R_\phi[\ K_p(J_M + J_L/N^2)^2 N^2 \theta_L^2/t_c^3 + T_L^2 t_c/N^2\ ]/K_t^2 \qquad ....(2.37)$$

This can be normalised by using $\gamma = [T_L t_c^2/\theta_L J_L]^2/K_p$

$$E = K_p R_\phi \theta_L^2 J_L^2 [N^2(J_M/J_L + 1/N^2)^2 + \gamma/N^2]/K_t^2 t_c^3 \qquad ....(2.38)$$

which when optimised for N (dE/dN=0), gives the optimum gear ratio:-

$$N_o^2 = (J_L/J_M)\sqrt{(1+\gamma)} \qquad ....(2.39)$$

which if $\gamma = 0$ (negligible load torque) $N_o^2 = J_L/J_M$.

Note that this is independent from the profile used. The use of a non optimum gear ratio can be observed by considering the ratio of energies when neglecting load torque:-

$$\frac{E\ (N)}{E_o(N_o)} = \frac{N^2(J_M + J_L/N^2)^2}{N_o^2(J_M + J_L/N_o^2)^2} \qquad ....(2.40)$$

Using $N_o^2 = J_L/J_M$ gives

$$\frac{E\ (N)}{E_o(N_o)} = \frac{1}{4}\left(\frac{N}{N_o} + \frac{N_o}{N}\right)^2 \qquad ....(2.41)$$

see figure (2.4).

Using equation (2.38) the effect of load torque can be examined if the gear ratio was chosen without taking it into account:-

$$\frac{E\ (N)}{E_o(N_o)} = \frac{N_o^2\{N_o^2(1/N_o^2 + 1/N_o^2)^2 + \gamma/N_o^2\}}{4}$$

$$= 1 + \gamma/4 \qquad ....(2.42)$$

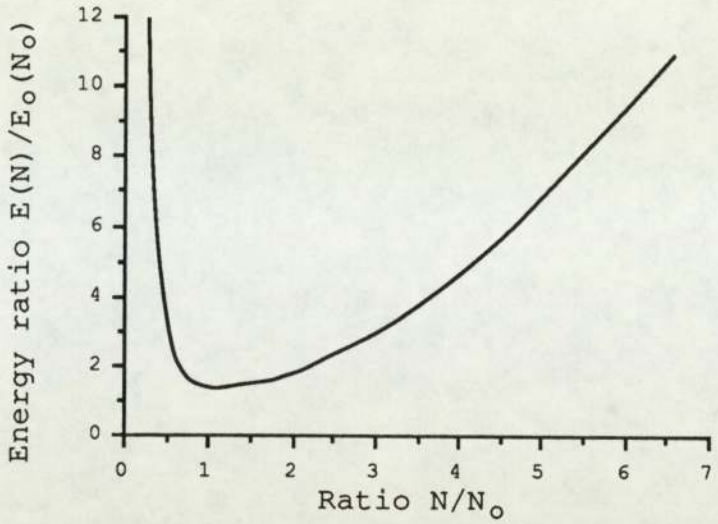


Figure 2.4 Energy dissipation using a non-optimum gear ratio

Similarly the effect of γ can be monitored with respect to the optimum no load case by use of $N^2 = N_o^2\sqrt{1+\gamma}$ in equation (2.38) such that:-

$$\frac{E_c(N)}{E_o(N_o)} = \frac{(\sqrt{1+\gamma})}{4}\{(1 + 1/\sqrt{1+\gamma})^2 + \gamma/\sqrt{(1+\gamma)}\} \qquad \ldots.(2.43)$$

see figure (2.5).

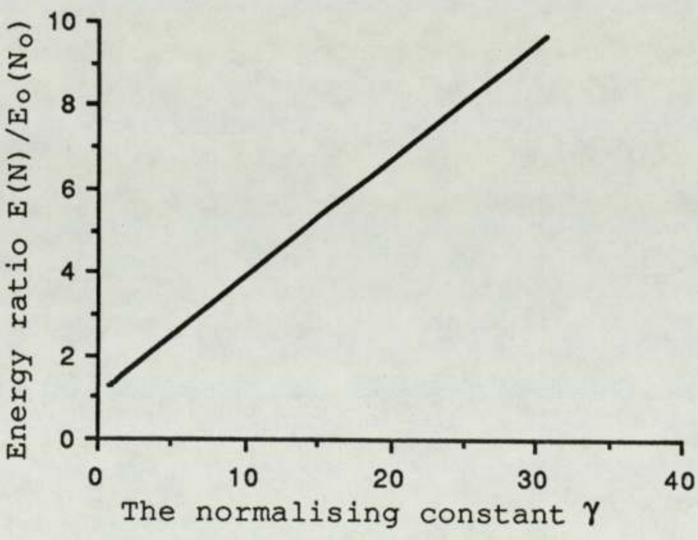


Figure 2.5 Energy dissipation when load torque is evident

2.3.5.3 Constrained design

The previous discussion optimised energy dissipation in a motor assuming that the motor could actually follow the specified velocity profile under an optimum coupling ratio condition. Often a motor has a maximum velocity such that a constrained optimisation is required.

Assume, for example, the profile to be trapezoidal with equal acceleration and deceleration periods, t_a, and a constant slew velocity period t_s at speed ω_p as shown in figure (2.6).

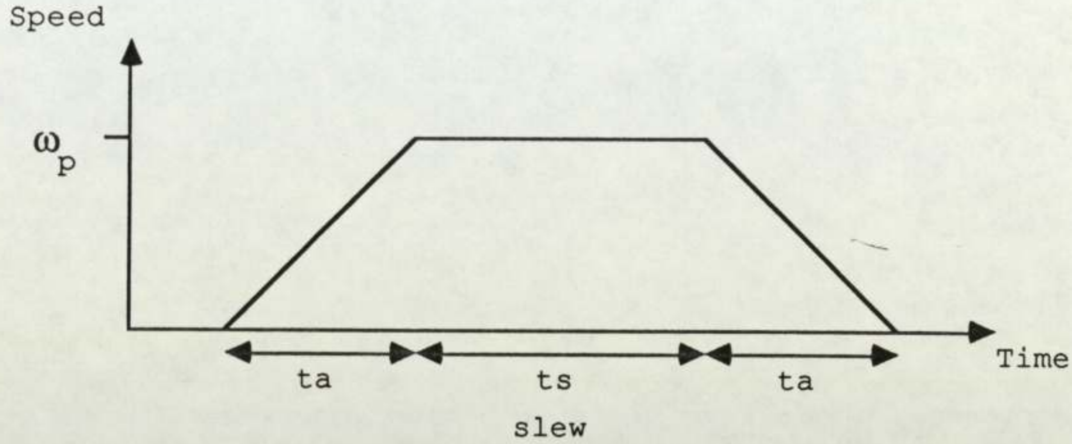


Figure 2.6 A trapezoidal profile

The acceleration and current required are given by $\alpha = \omega_p/ta$ and $I_a = J_T\omega_p/K_t.t_a$

Thus the energy during acceleration = $2R_\phi J_T^2\omega_p^2/K_t^2.t_a$ and $t_c=2t_a+t_s$

Therefore, $\theta_M = \omega_p(t_c-t_a)$ and $t_a = t_c-\theta_M/\omega_p = t_c(1-\theta_M/\omega_p.t_c)$

or $$E = \frac{2R_\phi J_T^2\omega_p^2}{K_t^2.t_c(1-\theta_M/\omega_p.t_c)} \qquad ....(2.44)$$

Converting for the load

$$E = \frac{2R_\phi(J_M+J_L/N^2)\omega_p^2}{K_t^2.t_c(1-N\theta_L/\omega_p t_c)} \qquad ....(2.45)$$

The unconstrained maximum velocity ω_{max} is defined as:-

$$\omega_{max} = 3\theta_M/2t_c \qquad ....(2.46)$$

and let $\lambda = \omega_{max}/\omega_p$ be the constrained factor, and $\psi = N\sqrt{(J_M/J_L)}$ which is the relative reduction in coupling ratio due to velocity constraints.

By minimising energy dissipation

$$E \quad \alpha \quad \frac{J_M+J_L/N^2}{1-N\theta_L/\omega_p.t_c} \qquad ....(2.47)$$

$$\text{or,}\quad E \quad \alpha\ \frac{J_M^2(1+1/\psi^2)^2}{1-2/3\lambda\psi} \qquad \ldots.(2.48)$$

λ is selected as a function of ψ and the solution is presented graphically by Tal [71] to show the increase in energy utilised due to constrained design.

2.3.5.4 To select for minimum temperature

The temperature rise in a motor is the product of power dissipation and thermal time constant R_{TH}, therefore ER_{TH} must be minimised:-

$$ER_{TH} = R_{TH}(KpR_\phi J_T{}^2\theta_M{}^2/K_t{}^2t_c{}^3 + R_\phi T_L t_c{}^2/K_t{}^2) \qquad \ldots.(2.49)$$

which may be reformed by substituting equation (2.39) in (2.38) to give:-

$$ER_{TH} = 2R_{TH}KpR_\phi J_M J_L\theta_L{}^2(1+\sqrt{1+\gamma})/K_t{}^2t_c{}^3 \qquad \ldots.(2.50)$$

so that for a given load and movement the variable terms are $R_{TH}R_\phi J_M/K_t{}^2$ such that $R_{TH}\tau_m$ must be minimised.

2.3.6 TORQUE TO INERTIA RATIO, OR THEORETICAL ACCELERATION

For a servo system with a predominantly inertial load the torque to total inertia ratio is directly proportional to acceleration. High acceleration rate is of key importance in achieving high dynamic response. If a fixed ratio is maintained between the motor and reflected load inertias (such as with the minimum energy criteria) then the torque to motor inertia ratio can be used to compare drives. This defines the theoretical acceleration of a dc servo drive without a load attached.

If a fixed ratio of inertias is not maintained in comparison and the load inertia is many times that of the motor then the torque to reflected load inertia is of interest.

Thus far no comment has been made as to whether the torque value used should be the continuous or peak intermittent value. When considering dynamic performance in terms of bandwidth the continuous value should be taken, since the torque required to maintain a bandwidth is continuously required. If the drive is being selected to follow a particular incremental position path then the peak value may be used if consideration is made for the duty cycle of the system. Floresta [30] suggested comparison using torques at rated speed and also at peak power in order to standardise "at a conceivably common operating point for many different types of actuators".

A further problem introduced by using acceleration to compare servos is that "like" is not compared with "like" if gearing is used, since gearing affects torque linearly and inertia by the second power.

2.3.7 TORQUE SQUARED TO INERTIA RATIO, OR POWER RATE

One particularly useful comparative figure of merit for incremental motion servo selection is power rate, which is defined as the continuous torque squared divided by the motor's rotor inertia, Arnold [3], Arnold and Floresta [4], Floresta [30], Newton [52] and Powell [58]. It is often considered as a servo drive's ability to "inject" power into a load, and it is independent of gearing. It is derived from the rate of change of power.

$$\frac{d}{dt}(\text{Power}) = \frac{d}{dt}(T_g\omega) = T_g\frac{d(\omega)}{dt} = T_g.\frac{T_g}{J_M} = \frac{T_g^2}{J_M}\ \text{Ws}^{-1} \qquad \ldots.(2.51)$$

For many applications a load is required to be moved "instantaneously" between two points. The bandwidth/ performance of a system will be limited by the system's ability to rapidly move the load to a desired position or correct for any errors.

It can be shown that the time to move through a fixed distance is minimised when the power rate is maximised by considering the simplest move between two points of figure (2.7).

$\theta_M = 0.5\alpha t_c^2$ where $T_g = J_T\alpha$ neglecting viscous and friction effects:-

$$t_c = \sqrt{(2\theta_M J_T/T_g)} = \sqrt{(2\theta_L N(J_L/N^2+J_M)/T_g)} \qquad \ldots.(2.52)$$

and $$\frac{dt_c}{dN} = \sqrt{\frac{2\theta_L N}{T_g}}\left(\frac{J_L}{N}+NJ_M\right)^{-1/2}\left(J_M - \frac{J_L}{N^2}\right) \qquad ....(2.53)$$

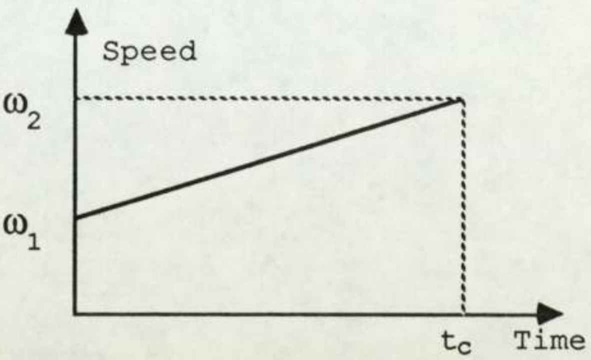


Figure 2.7 A simple velocity step

Thus for a minimum or optimum move time $N_o^2 = J_L/J_M$ which is the optimum gear ratio as would be expected. Using this ratio the minimum move time can be found:-

$$t_{co} = 2\sqrt{(\theta_L(\sqrt{J_M}.J_L)/T_g)} \qquad ....(2.54)$$

and since θ_L and J_L are constant for an application the minimum move time is proportional to $\sqrt{J_M}/T_g$, or inversely proportional to power rate. Therefore maximising a motor power rate will minimise the move time if the optimum gear ratio is used. Elkington [26] showed that the effect of load inertia on minimum motion time could be divided into three distinct regions. Firstly when the motor inertia dominates there is no significant change in motion time due to an increase in load inertia. Secondly when near the optimum coupling ratio (roughly equal motor and load inertias) there exists a non-linear relationship between load inertia increase and minimum motion time, such that power rate analysis is very relevant. Thirdly when the load inertia dominates the minimum motion time, torque to inertia ratio is more applicable than power rate analysis.

If the optimum gear ratio is not used then the move time can be normalised to give:-

$$\frac{t_c}{t_{co}} = \sqrt{\frac{\left(J_L/N^2 + J_M\right)N}{\left(J_L/N_o^2 + J_M\right)N_o}} \qquad \text{....(2.55)}$$

using $J_L=N_o^2 J_M$:-

$$\frac{t_c}{t_{co}} = \frac{1}{\sqrt{2}}\sqrt{\frac{N_o}{N} + \frac{N}{N_o}} \qquad \text{....(2.56)}$$

which can be compared to the earlier result (2.41) such that $E/E_o = (t_c/t_{co})^4$.

Since t_{co} is dependent upon power rate, the non-optimum move time is dependent upon power rate and the ratio of optimum to non-optimum gear ratios.

For a directly coupled system:-

$$t_c = \sqrt{2\theta_L(J_L+J_M)/T_g} = \sqrt{(2\theta_L J_M/T_g)} + \sqrt{(2\theta_L J_L/T_g)} \qquad \text{....(2.57)}$$

Thus, maximising acceleration rate or T_g/J_M becomes more important, and power rate less important as the gear ratio moves away from the optimum condition.

Also if $J_L>>J_M$ then $t_c \approx \sqrt{2\theta_L J_L/T_g}$ and in this instance one merely seeks the highest torque output from a motor at a given speed for high performance.

Rearranging equation (2.52) gives an insight into the power rate required by the load.

$$\frac{t_c^2}{2\theta_L} = \frac{1}{\alpha_L} = \frac{1}{T_g}\left(\frac{J_M N_o^2}{N} + N J_M\right) \qquad \text{....(2.58)}$$

$$\frac{1}{\alpha_L} = \frac{\sqrt{J_M}}{T_g}\cdot\frac{\sqrt{J_L}}{N_o}\cdot\left(\frac{N_o^2}{N} + N\right) \qquad \text{....(2.59)}$$

Therefore,

$$T_g^2/J_M = J_L\alpha_L^2(N_o/N + N/N_o)^2 \qquad \text{....(2.60)}$$

$J_L\alpha_L^2$ is the power rate of the load.

Thus,

$$dP_M = dP_L(N_o/N + N/N_o)^2 \qquad \ldots.(2.61)$$

Load power rate can be more exactly defined, Floresta [30] as,

$$dP_L = (J_L\alpha_L + T_L)\alpha_L/\eta \qquad \ldots.(2.62)$$

where η is the gearbox efficiency.

From these definitions it is therefore possible to locate the minimum power rate of a drive which will enable the profile to be followed. For the inertial match situation $N_o/N = 1$ and $dP_M=4dP_L$. That is the actuator must have a power rate of at least four times that of the load.

Furthermore power rate can be shown to be the ratio of motor copper losses to the mechanical time constant [4] and this forms the basis for a drive selection procedure, Newton [52], thus:-

$$\frac{T_g^2}{J_T} \times \frac{1}{I_a^2R_\phi} = \frac{T_g^2/I_a^2}{R_\phi J_T} = \frac{K_t^2}{R_\phi J_T} = \frac{1}{\tau m} \qquad \ldots.(2.63)$$

2.4 CONCLUSIONS

The simple model derived and the drive selection techniques of the foregoing discussion concentrated upon brushed dc machines. The equations have however been presented such that they can equally be used for brushless systems (unless otherwise stated), see chapter 3.0. The model was further simplified by using mechanical and electrical time constants and the accuracy of this approach was presented.

Damping factor, regulation and the universal motor constant have been discussed as factors of merit but they are thought to be not very useful. The mechanical and electrical time constants were discussed and again thought not to be useful for drive comparisons due to the standard practice of using high gain current loops.

Power dissipation analysis concludes that a parabolic or trapezoidal incremental move should be used and that the gear ratio should be chosen such that the inertia of the load is "matched" to that of the motor (adjustments being made for velocity limitations or the effects of viscous damping).

Power rate has been identified as a useful factor-of-merit for comparing drives and a selection procedure developed with it, since the load's power rate can be calculated. Again power rate analysis identified the benefits of using matched inertias.

Whilst the techniques described in this chapter have proved successful for selecting drives for incremental applications, such as the problem described in appendix 1, no mention has been made of selecting drives for a continuous application as with the specification of section (1.5). The power rate, acceleration, power and energy requirements are not exacting in the specific engineering application. The requirement is for high accuracy, high speed control of high inertia loads which prohibits inertial match or the effective use of gearing. A relatively small powered drive would be able to meet the torque and power requirement but would it be able to meet the exacting positional accuracy? The methods presented in this chapter were developed for incremental applications and cannot be directly related to this research programme. Chapters 5.0 and 6.0 develop drive selection methods for achieving a required positional accuracy.

CHAPTER 3

3.0 A FORMAL PROCEDURE FOR MOTOR COMPARISON

3.1 INTRODUCTION

Chapter 2.0 presented standard methods for selecting dc servo systems, but unfortunately none of these methods was directly applicable to the problems posed in section (1.5), or for selecting other forms of drive. No method exists to select a drive upon profile following accuracy, as required for the application described in section (1.5), of tight position control of a high inertial load rotating at a continuous high speed. Earlier attempts to meet the performance of the specification by Molins (appendix 2) had failed whilst using drives with greatly over-rated power capability. It was therefore uncertain whether the specification could be met by any available system or technology.

The decision was made to attempt to select the highest performance drives available in the market place and to test them against the specification; performance rather than cost was therefore important.

The types of drive system considered are discussed and compared subjectively in section (3.2) to present a background to the remainder of the chapter.

In order to compare drive systems some form of commonality is required, so section (3.3) extends the modelling of dc machines of chapter 2.0 to include brushless machine systems. A brushless configuration constant is derived which allows brushed and brushless systems to be compared.

Finally section (3.4) presents a formal unambiguous way to compare drives for the particular application using figures of merit introduced in chapter 2.0. The method avoids the problems of past authors who have used conflicting data when comparing different drive genre, and ambiguous base measures. Results for seven high performance drive systems are given and comparisons are made.

3.2 REVIEW OF VARIABLE SPEED ELECTRIC DRIVES

In order to meet the specification laid down in section (1.5) a very high performance servo system would be required. There are two main categories of prime mover that could be used, electrical and hydraulic, the latter whilst having high performance, Firoozian et al [28] was not considered under the remit of this research.

This section will briefly review subjectively the main electric drive systems available at the moment, highlighting their suitability for high performance servos. Both Hall [33] and Bose [12] review and compare electric drive systems. A modern electric drive can be considered as comprising four main elements: the controller, the power electronic amplifier, the electric machine and the load. Theoretically each element may be individually selected, but in practice a fixed topology is purchased to suit the load, from a vendor.

Dc drives have been used in high performance servos for many years. Permanent magnet machines are the most suited to servo applications since their inherent time constant due to the field is smaller than that for electromagnetically excited systems, Klein [41]. The drive amplifier may use a rectifier bridge with ac input "point-on-wave" switching or pulse width modulation (PWM) with high frequency dc switching. The PWM systems have higher bandwidths and are therefore more applicable to high performance servo systems.

The performance of a dc drive is dependent upon the magnet material used for the field. There are three major factors influencing the choice of a magnet material, Christensen [18]: cost, the maximum energy product (MEP) which is a performance figure of merit and the temperature coefficient which measures the stability of magnet properties with increasing temperature (the higher the coefficient the more likely that the magnet would weaken if operated at high temperature). Ceramic or ferrite magnets lose 0.13%/°C of their remanence above 25°C, whilst rare earth and alnico lose only 0.03%/°C, Zimmermann and Bosch [86]. Typically the MEP for ferrite magnets is 1.8-4.5 MGOe, for Alnico 5-12 MGOe, for Samarium Cobalt 18-26 MGOe and for Neodymium-Iron-Cobalt 8-35 MGOe, Carlisle [16]. Rare earth are the most costly and have the highest MEP's; Neodymium are better than Samarium Cobalt magnets but they exhibit irreversible magnetic losses under wide temperature changes. Alnico and ceramic magnets are also prone to irreversible magnetic loss if subjected to peak load currents from the armature winding. The highest performance systems utilise rare earth magnet excitation in particular Neodymium magnets.

Thus far the discussion has concentrated upon conventional brushed dc machines as these predominated the high performance servo market up to about ten years ago (≈1978). With the advent of reliable power electronics ac machines have been successfully competing in markets traditionally dominated by the dc machine, since they overcome some of the problems associated with brushed systems. AC machines generate most heat in the armature winding which is located in the stator rather than the rotor, thus aiding cooling, the rotor has low inertia due to the permanent magnets and ac machines do not possess a commutator; a dc machine's commutator has a limited circumferential speed, limited segment voltage, it emits electro-magnetic interference and it requires maintenance, Tomasek [81].

Ac motors are more reliable, efficient and compact than a comparable dc machine, typically weighing 35-65% less, being 20-30% shorter in length and reduced in diameter by 10-20%, Carlisle [17]. They can also run at higher velocities and withstand higher peak currents.

Switched reluctance drives are being increasingly used for variable speed applications because of their inexpensive, efficient and reliable machine. Their operation relies upon the torque produced by the difference in reluctance between the direct and quadrature axis of a salient pole machine. This form of machine has been proved to have more torque at higher efficiencies than a comparable induction machine, Firoozian et al [28], but unfortunately for this application they have unacceptable torque ripples and are difficult to brake.

Stepping motors operate in a similar manner to reluctance machines but utilise active permanent magnet rotors. They are very attractive in many applications due to their excellent open-loop performance, but they have poor torque speed characteristics with torque ripple, [28] and are only commonly available at relatively low powers.

Induction machines can perform servo duties by using position feedback with sophisticated vector control, but they are less efficient with larger volume and inertia for a given torque than a synchronous machine, Barber [6]. It is also difficult to control position since torque is not proportional to current as in a dc machine.

Synchronous machines are now employed in brushless ac or brushless dc schemes using position feedback. The machines have high performance since rare earth permanent magnet rotors have extremely low inertia. The stator usually has a three phase star connected winding which may be concentrated or distributed; the concentrated winding is used in brushless dc systems requiring trapezoidal current waveforms, whilst the distributed winding is used for brushless ac systems which require sinusoidal excitation. In both schemes a reference current signal is produced by the controller which is proportional to torque as in a conventional dc machine. The sinusoidal system whilst more complex than the trapezoidal scheme is superior as it exhibits negligible torque ripple. These systems are often described as "inside-out" dc machines and the combination of position feedback device and power amplifier can be considered to be the commutator.

Comparisons between conventional and brushless dc machines are presented by Horner and Lacey [37], Brown and Moore [14] and Zimmermann and Bosch [86], between brushless dc and induction machines by [14] and Brosnan and Barker [13] and between brushless dc and pancake machines by Mazurkiewicz [46]. The conclusion from these references and this discussion was that the highest performance servo system for the application of section (1.5) will be a PWM driven brushless ac machine utilising rare earth (Neodymium) magnet materials. Section (3.4) presents a formal approach to motor comparison which substantiates the previous subjective discussion.

3.3 MATHEMATICAL REPRESENTATION OF BRUSHLESS SYSTEMS

Brushless machines are similar to permanently excited synchronous machines but are similar to brushed dc drives in that they are usually controlled by a velocity loop supplying a current loop, and the torque produced by a brushless dc motor is directly proportional to armature current, thus simplifying control. The power electronics perform the commutation sequence in a brushless drive with the aid of an appropriate position transducer.

These systems behave like conventional dc machines, with all the benefits of easy control, but they also have the advantages associated with an ac machine.

Complex modelling of brushless machine systems may be undertaken. Detailed finite element analysis of the motor's magnetic circuit can be carried out, Jones [39], but such a detailed approach is not required. The general machine equation (3.1) can be used to model a three phase machine, Bolton and Ashen [11], Gipper [32] and Persson [56,57]. The effects of power electronic switching are modelled by Nehl et al [51]. Phasor and d-q axis analysis was utilised by Vaidja [85] and Persson and Meshkat [53] to optimise motor power factor and torque under various load conditions. A simpler model and simulation of a brushless dc drive was required, similar to the approach taken by Blank and Wrobel [10] and Lim and Kangsanant [44] which produced block diagrams similar to those presented in chapter 2.0, such that, brushed and brushless systems may be compared.

$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} R_1 & & O \\ & R_2 & \\ O & & R_3 \end{bmatrix} \begin{bmatrix} i_1 \\ i_2 \\ i_3 \end{bmatrix} + \begin{bmatrix} L_{11} & L_{12} & L_{13} \\ L_{21} & L_{22} & L_{23} \\ L_{31} & L_{32} & L_{33} \end{bmatrix} \begin{bmatrix} di_1/dt \\ di_2/dt \\ di_3/dt \end{bmatrix} + \frac{d\theta}{dt} \begin{bmatrix} d\phi_1/d\theta \\ d\phi_2/d\theta \\ d\phi_3/d\theta \end{bmatrix} \quad \ldots.(3.1)$$

We must define the torque and back emf constants for a brushless machine with care as:-

Torque constant $K_{t\phi} = \dfrac{\text{Stall Torque}}{\text{Winding current at stall with only one phase energised and the rotor positioned to maximise torque}}$

Back emf constant $K_{e\phi}$ = $\dfrac{\text{Peak output voltage/phase}}{\text{Constant Angular speed with the motor driven as a generator \& the winding terminals o/c}}$

For a lossless system $K_{t\phi} = K_{e\phi}$ as with a brushed case.

A sinusoidal brushless dc drive is driven by sinusoidal currents and has a sinusoidal position/torque characteristic, as shown in figure (3.1).

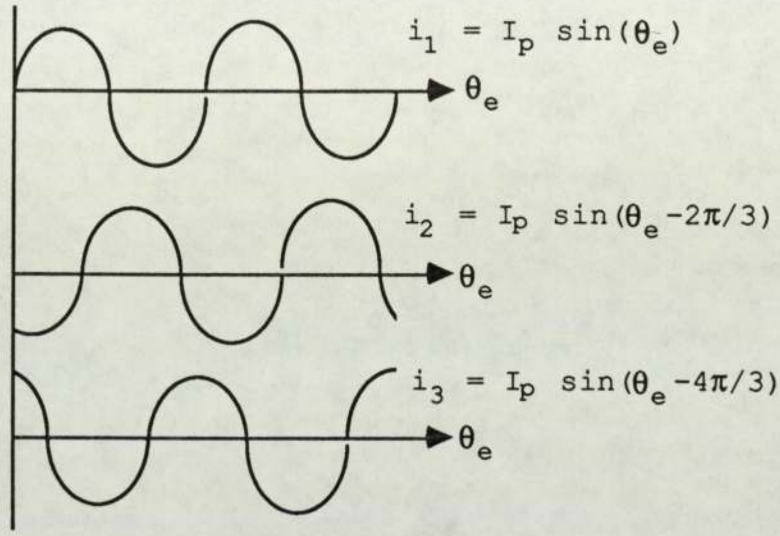


Figure 3.1 The position/torque curves of a sinusoidal brushless dc machine

The three individual phase torques are therefore:-

$$T_1 = K_{t\phi} i_1 \sin(\theta_e) \quad \ldots.(3.2a)$$
$$T_2 = K_{t\phi} i_2 \sin(\theta_e - 2\pi/3) \quad \ldots.(3.2b)$$
$$T_3 = K_{t\phi} i_3 \sin(\theta_e - 4\pi/3) \quad \ldots.(3.2c)$$

but the phase currents are,

$$i_1 = I_p \sin(\theta_e) \quad \ldots.(3.3a)$$
$$i_2 = I_p \sin(\theta_e - 2\pi/3) \quad \ldots.(3.3b)$$
$$i_3 = I_p \sin(\theta_e - 4\pi/3) \quad \ldots.(3.3c)$$

The total generated torque is therefore:-

$$T_g = T_1 + T_2 + T_3 \quad \ldots.(3.4)$$
$$= K_{t\phi} I_p (\sin^2(\theta_e) + \sin^2(\theta_e - 2\pi/3) + \sin^2(\theta_e - 4\pi/3))$$
$$= \frac{3}{2} K_{t\phi} I_p \quad \ldots.(3.5)$$

That is, a constant torque 1.5 times the peak torque of one phase.

A trapezoidal dc drive is driven somewhat differently in that the windings are switched either on positive, on negative or off dependent upon the torque position curves, as shown in figure (3.2). For a standard three phase trapezoidal drive there are essentially six different commutation states. For a sinusoidal torque position curve as before there would be considerable torque ripple, but manufacturers of trapezoidal systems alleviate this problem by slewing the stator slots and by concentrating stator windings so as to produce trapezoidal torque/position characteristics.

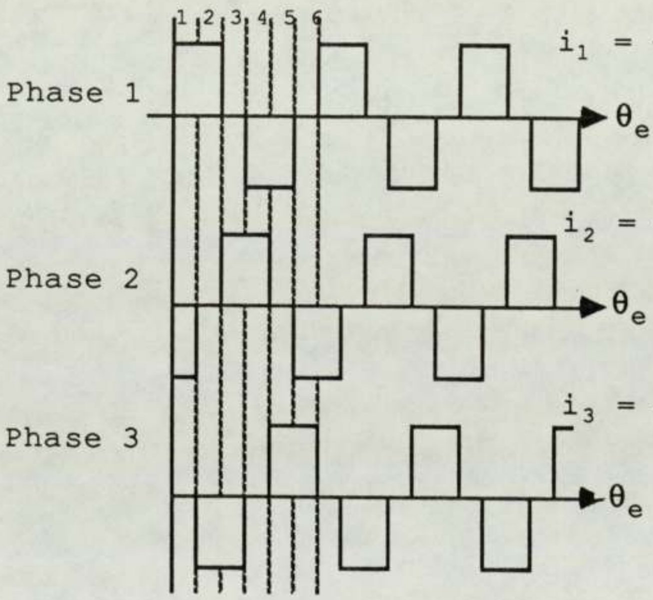


Figure 3.2 The switching sequence for a trapezoidal brushless dc machine

If zone 1 of figure (3.2) is considered then,

$$T_g = T_1 + T_2 + T_3 \qquad \ldots.(3.6)$$

$$= K_{t\phi}I_p((+1)^2 + (-1)^2 + (0)^2)$$

$$= 2K_{t\phi}I_p \qquad \ldots.(3.7)$$

The other zones produce the same result so a trapezoidal system has theoretically no torque ripple and the output torque is $2K_{t\phi}I_p$. Unfortunately the commutation can never be instantaneous and is rarely at precisely the correct rotor position so that some torque ripple will be evident. For a given peak current more torque is available from a trapezoidal drive when compared to a sinusoidal drive and there is therefore better utilization of the power electronic devices. Sinusoidal drives also require more complex control but produce negligible torque ripple.

It can be seen from the preceding discussion that a brushless system can be represented in a similar manner to a brushed dc drive as in figure (2.1).

The brushless configuration constant, K_{bc} was introduced to alleviate the problem that $K_{bc}K_{t\phi} \neq K_{e\phi}$. K_{bc} is 3/2 for a sinusoidal and 2 for a trapezoidal drive. When comparing drives, though, great care must be taken to ensure the correct parameters are being used. Unfortunately manufacturers often quote different and misleading data; peak and rms values are often used, line or phase quantities are not always defined clearly and the torque constant quoted is usually $K_{bc}K_{t\phi}$, the motor's total torque constant. Tomasek [77] attempted to rationalise the data used when specifying brushless dc drive systems. He utilised a "composite" current and line-to-line voltages for trapezoidal systems such that the torque and back emf constants appear equal; peak line values for voltage and current were used. For a sinusoidal system he derived equation (3.5) but suggested the use of rms rather than peak values and line-to-line rather than phase values for current and voltage such that K_T [Nm/A_{rms}] = $\sqrt{3}$ K_E [$V_{rms(L-L)}$/rads^{-1}]. Thus the values are associated with resistive power loss and rms torque. However this approach does not easily allow the representation of a brushless system in the same way as a conventional dc servo system.

3.4 COMPARING SERVO DRIVES OF SIMILAR POWER

Chapter 2.0 presented methods for selecting conventional dc drives and a number of factors of merit were introduced. By use of the brushless configuration constant it is now possible to extend this work to brushless drive systems. K_t is replaced by $K_{t\phi}K_{bc}$ in any equation whilst $K_{e\phi}$ remains unaltered. Care must however be taken if considering motor resistive energy losses, since equation (2.35) gives the loss for one phase only and brushless dc drives have three phases.

In order to assess the relative dynamic characteristics of servo systems (of similar torques at a given speed), a number of different factors of merit have been evolved. These factors of merit give an indication of the relative dynamic performance that may be obtained, between drives; no indication of absolute performance can be gained by these comparisons. Dynamic performance is very important when using the drive within a servo system, as it effects the velocity or position accuracy that may be obtained.

For any given factor of merit it is important to compare "like with like" motors. What index should be used against which to measure a factor of merit? In the past authors have used continuous rated power, peak rated power, continuous torque, and peak torque available from the motor, Arnold [3] and Fenney [27]. The problem with using any of these measures is that they do not precisely relate to similar drives; it is not useful to compare two drives of the same continuous power if one drive is a low speed torque motor whilst the other a high speed servo motor. The index of comparison has to be related to the end use of the servo drive, that is, the load it will have to carry and the velocity/position profile it will have to follow. The combination of load and profile will enable an envelope of torque versus speed to be formed. At any particular speed the drive will have to deliver a particular maximum torque and a continuous torque derived from the frictive and viscous forces present and the torque required to accelerate the load. If the servo is to run at constant velocity and attainable positional or velocity accuracy are required then only continuous torque is of interest; in this instance drives of similar continuous torques at a given speed (top speed of the load) may be compared. Thus all factors of merit will be compared against continuous torque at a given speed which defines also a particular continuous power capability from the drive (although it will commonly be different to any powers quoted by manufacturers).

If the drive has to follow a varied profile involving acceleration and deceleration of the load then the index to measure against depends upon whether the drive has to meet a peak torque demand (for short infrequent intermittent motions of the load requiring non-continuous torque) or whether the drive is continuously accelerating and decelerating requiring the use of the continuous torque again.

It is therefore proposed that the drive factors of merit are compared either against peak or continuous torque available at a specific velocity (thus defining a specific power).

As they do not reveal any absolute performance of a system, different authors have used different parameters when defining these factors; for example, torque constant and back emf constant are interchanged as they are the same in a brushed dc machine but this is not the case for a brushless system. Thus, when using any of these figures of merit care must be taken to compare "like with like", that is, always use per phase motor parameters for R, L, K_e, I, V, and the full-phase total for K_t, Seaward and Johnson [64], and as discussed in section (3.3).

3.5.0 A COMPARATIVE EXAMPLE

3.5.1 GENERAL

Data sheets for 36 different electric servo systems were obtained from manufacturers. Many of the data sheets did not contain the relevant data required for the comparison. Seven drive systems were chosen for the comparison. The drive systems used in the comparison were chosen because the information required was available, and that they were typical of their genre. The high performance brushless drives were chosen as they represented some of the most technologically advanced systems available.

The seven drive systems ranged from poor performance ceramic brushed dc machines to high performance rare earth magnet brushless machines. All were classed as high performance servo drives by their manufacturers. Importantly the data derived was not that of the electrical machine in isolation but of the motor/drive package combination recommended by the manufacturer, so that high performance machines may have degraded performance due to a poor amplifier and vice versa. Most modern servo drives are bought as a motor/drive package and incompatibilities prevent matching a motor with any drive (this is certainly true of the brushless systems which rely upon specific drive modules for commutation). Often the continuous torque and current rating of a system are defined by the motor winding capability, whilst the peak rating is determined by the power electronics in the drive package.

The particular application of section (1.5) required continuous control of drives up to 2500r/min with the typical operating velocity of 2000r/min. There were various different torque requirements, so the selection procedure was for drive series rather than a selection for a specific load/profile combination.

Thus the data shown in the comparisons is quoted against the continuous torque available at 2000r/min. For completeness the plots for peak torque are also shown which would be of relevance for any intermittent application. Also shown are a number of other useful plots to show the differences between drives and which factors of merit should most heavily be relied upon when comparing drives.

The drives were classed drive system 1 to drive system 7, the drive type and manufacturer are given in appendix (3).

Drive system 1 used conventional brushed dc machines with ceramic permanent magnets for the field excitation.
Drive system 2 was as in drive 1, with the addition of a cooling blower.
Drive system 3 was as in drive 2, with Samarium Cobalt field magnets replacing the ceramic ones.
Drive system 4 used sinusoidal excitation of a brushless machine. The machine had Samarium Cobalt magnets and a low inertia disc rotor.
Drive system 5 used sinusoidal excitation of a brushless machine which had Neodymium Boron Iron magnets.
Drive system 6 used trapezoidal excitation of a brushless machine which had Samarium Cobalt magnets.
Drive system 7 used sinusoidal excitation of a brushless machine which had Samarium Cobalt magnets.

3.5.2.0 COMPARISON OF DRIVES

3.5.2.1 Introduction

Figures (3.3)-(3.7) show the variety of plots that may be obtained when comparing drives. All the plots have been shown on normal and logarithmic scales as the logarithmic scales give an indication of trends, whilst the normal plots are given for clarity. When comparing drives a common base is required for a true comparison. In the plots the continuous torque and the peak torque, both at 2000 r/min have been used; each will be concentrated upon in turn.

3.5.2.2 Continuous torque comparisons

Figures (3.3a-b) show the comparison with damping constant. It is seen that there is very little difference between motors of the same continuous power. On the logarithmic scale there is an approximate linear relationship between continuous torque and damping constant: careful examination suggests a square law between the two variables. As there is little difference between drives this measure cannot give conclusive information as to the performance of one drive system compared to another.

Figures (3.3c-d) show the comparison with electrical time constant. There is wide variation between drives at all torque capabilities. As continuous torque increases so does the time constant at very approximately 1ms for 1Nm. The time constants vary from 2-28ms. Drives 1,2,3 and 4 have the worst time constants whilst 5,6 and 7 have the best, when considering possible performance. Thus the standard brushless motors (synchronous machines) have lower electrical time constants than their conventional brushed counterparts. The disc rotor brushless drive on this figure of merit is particularly poor for a brushless drive. The brushless drives appear to have the best performance, but the electrical time constant should never be considered in isolation from the mechanical time constant.

Figures (3.3e-f) show the comparison with mechanical time constant. There is wide variation between drives at all torque capabilities. There appears to be no relationship between mechanical time constant and continuous torque, in fact it appears to be near constant within any drive type. The time constants vary from 0.3-35ms, a wide variation. When considering possible performance, Drives 1,2,3 and 4 have the worst time constants whilst 5,6 and 7 have the best. Thus the brushless motors have lower mechanical time constants than their conventional brushed counterparts. The disc rotor brushless drive on this figure of merit is particularly poor for a brushless drive. The brushless drives therefore appear to have the best performance. Drive 7 is particularly good.

Figures (3.7a-b) show the comparison with the product of the time constants. This figure of merit shows the "residual" second order element of the current loop, see section (2.3.4). There appears to be no relationship between continuous torque and this figure of merit, and it is approximately constant within a drive type. Drives 5 and 7 are particularly good, whilst drives 1 and 2 are particularly poor. This again highlights the increased performance one can expect from a brushless system. It should be noted that the rare earth brushed systems have much higher performance than ceramic ones, which in some cases is better than brushless drives. Whilst the time constants give a measure of performance the current loop feedback "masks" these effects and so figures of merit based on time constants can only give an indication of the ability to close a high bandwidth current loop. The drive amplifier is critical here and a good machine coupled to a poor drive, will have poor performance. Much better figures of merit to use when comparing drives are those that combine the properties of the amplifier/machine package.

Figure (3.4) shows important figures of merit when selecting a drive on continuous performance. Figure (3.4a-b) show the torque to inertia ratio plot. Since some selection procedures use the machine inertia, lines of constant inertia have been shown. As the torque increases the ability to accelerate decreases, that is, torque increases slower than inertia. As a machine increases in physical size there exists an approximate square law on the increase in torque and quartic law on the increase in inertia, Hindmarsh [35]. For a given torque drives 5 and 7 appear to have the best performance and 1,2 and 3 the poorest. When comparing against constant inertia the brushless drives are still superior to conventional systems.

Figures (3.4c-d) show the power rates of the drives against continuous torque, and as might be expected the drives, follow the same relative pattern as for torque to inertia ratio. Thus the brushless drives are also able to "inject power" better than their brushed counterparts.

Figure (3.4e-f) show the power rate of a drive plotted against the acceleration. This is a very useful plot as constant torque and constant inertia lines may be drawn. This plot is considered to be the easiest to use when comparing drives as it contains most of the pertinent information required. The drives in the top right corner of figure (3.4e-f) will have the best performance, that is, a high acceleration rate combined with a high power rate. From this plot it is obvious that drive performance increases from drive 1, to drive 2, drive 3, 4, 5 and 6, to drive 7. Thus the brushless systems are far superior to the brushed systems.

If the drives are to be used at some continuous torque value, but an intermittent high torque is required then figure (3.3) is useful. These figures show the peak torque, peak power rate and peak accelerations against continuous torque.

Figures (3.5a-b) show the relation between peak and continuous torque. All the drives considered had peak to continuous ratings of between 1:1 and 10:1. Drive 2 is particularly poor at about 1.5:1 and drive 4 particularly good at 8:1 - 9:1. The remaining drives have ratios of 2:1 - 5:1. The plots show no conclusive evidence of a particular drive type having superior intermittent performance except perhaps the disc rotor system of drive 4, which on all other measures has poor performance for a brushless system.

3.5.2.3 Peak torque comparison

Peak torque is used as a base for comparison when an intermittent duty cycle is required from a drive, figure (3.6). Since the ratio of peak to continuous torque is approximately constant for many of the drives, as discussed above, the results obtained vary little from those using the continuous base as a measure. Drive 4 has better, and drive 2 has poorer performance.

A detailed discussion of these results is not given, as the conclusions are again that the brushless systems have higher performance than conventional brushed types.

3.6 CONCLUSIONS

This chapter has extended the dc machine modelling technique introduced in chapter 2.0 to include brushless systems, by the inclusion of the brushless configuration constant. This constant has been calculated for the two common types of three phase brushless drives, that is, those with trapezoidal and those with sinusoidal excitation schemes.

A formal analytical method for drive comparison has been derived. The method was used on seven drive types and it showed that at 2000 r/min brushless drive systems were far superior to brushed drives. The flat rotor brushless drive had the poorest performance of all the brushless systems considered which was still higher than any of the brushed systems. The rare earth brushed system had the highest performance of the brushed systems. All the drives were quoted as high performance servo drives by their manufacturers and there is great disparity in the apparent performance.

The curve of power rate against torque to inertia ratio was the most useful plot to use for comparison as the most information can be gleaned from it.

Two important factors which have been neglected are cost and reliability. the brushless systems are generally more expensive than conventional systems but the price differential is reducing. Reliability is difficult to assess but experience has shown that some brushless systems suffer from reliability problems associated with "new" technology but this is becoming less of a problem.

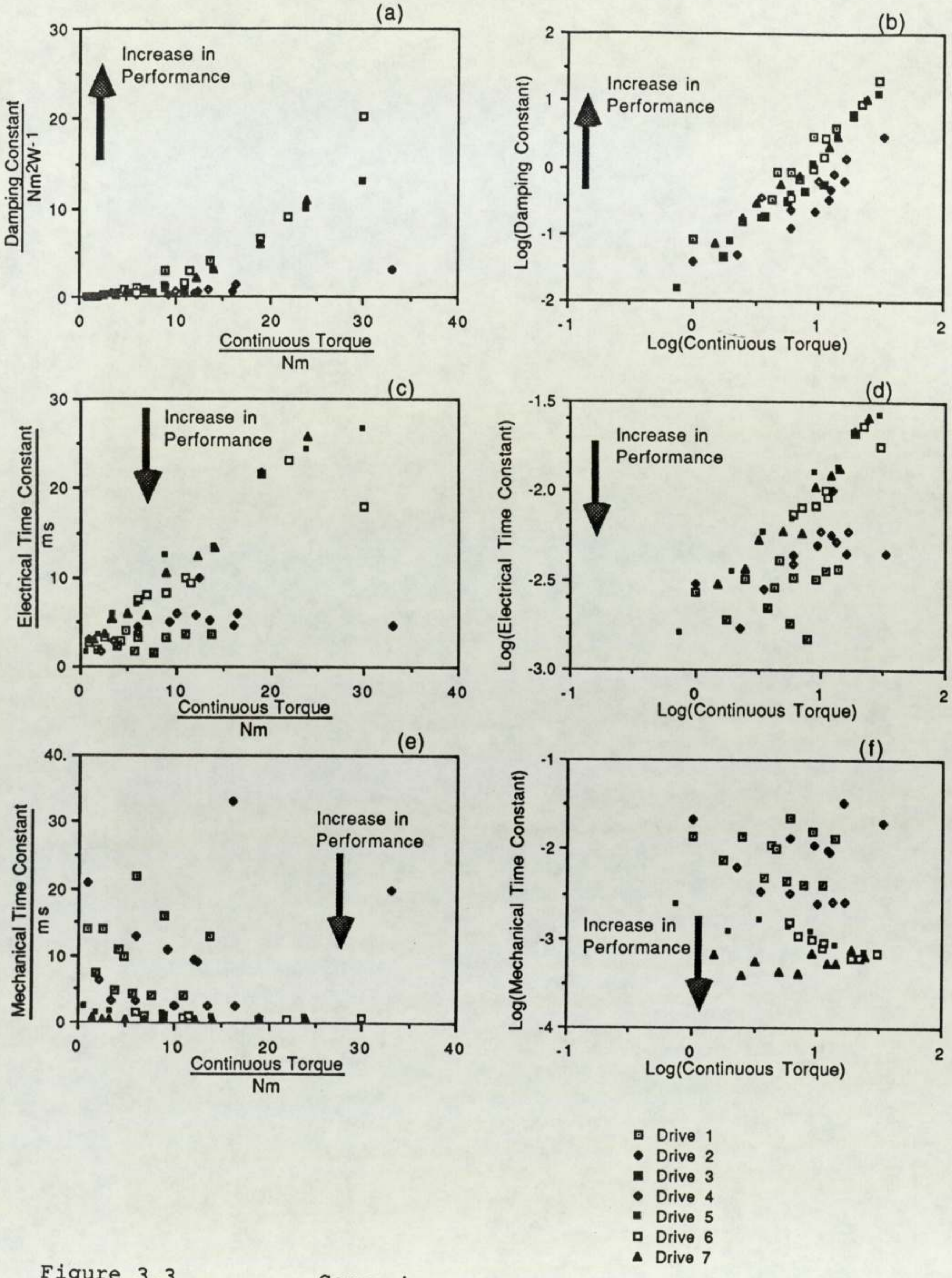


Figure 3.3 Comparison of drives

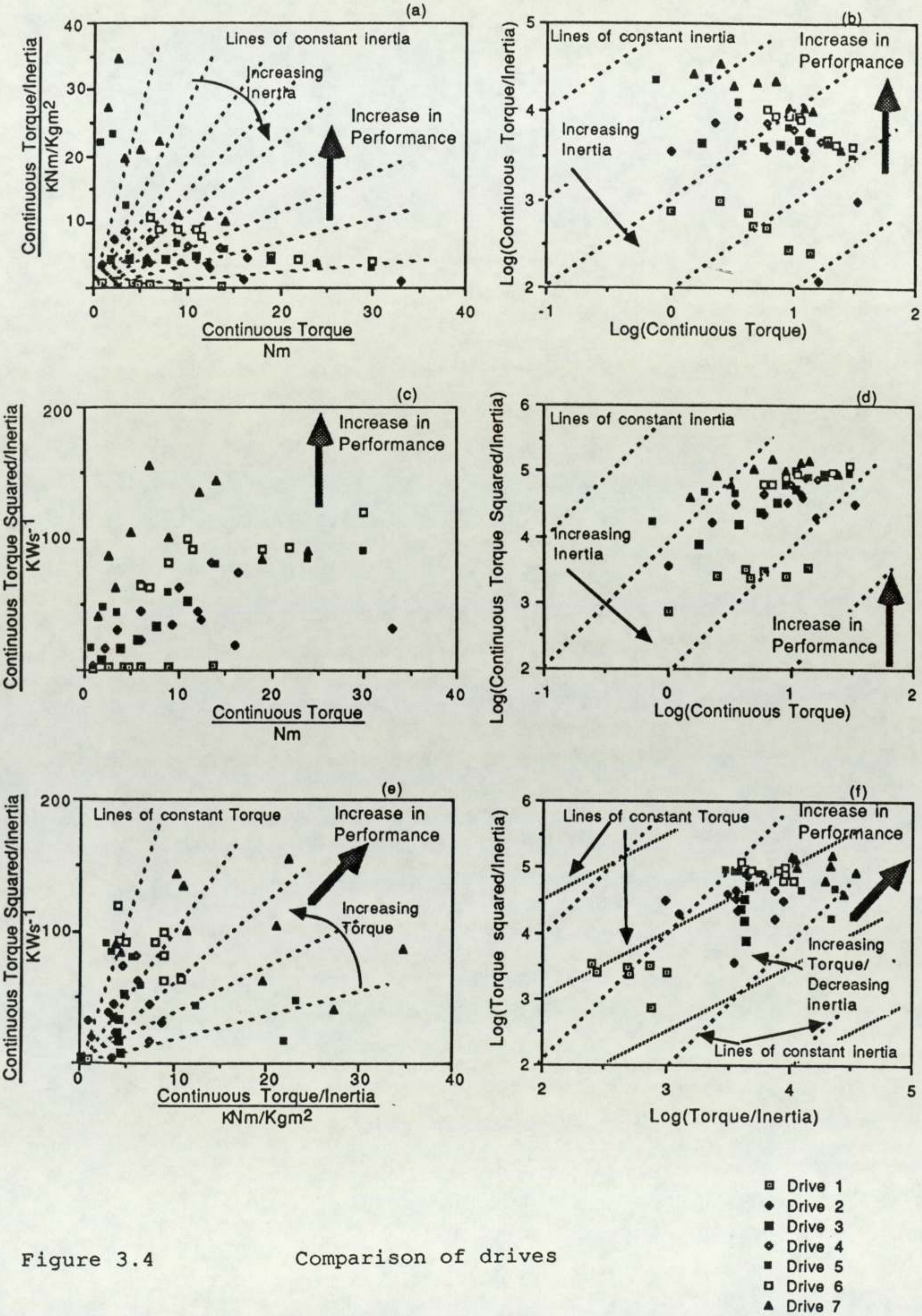


Figure 3.4 Comparison of drives

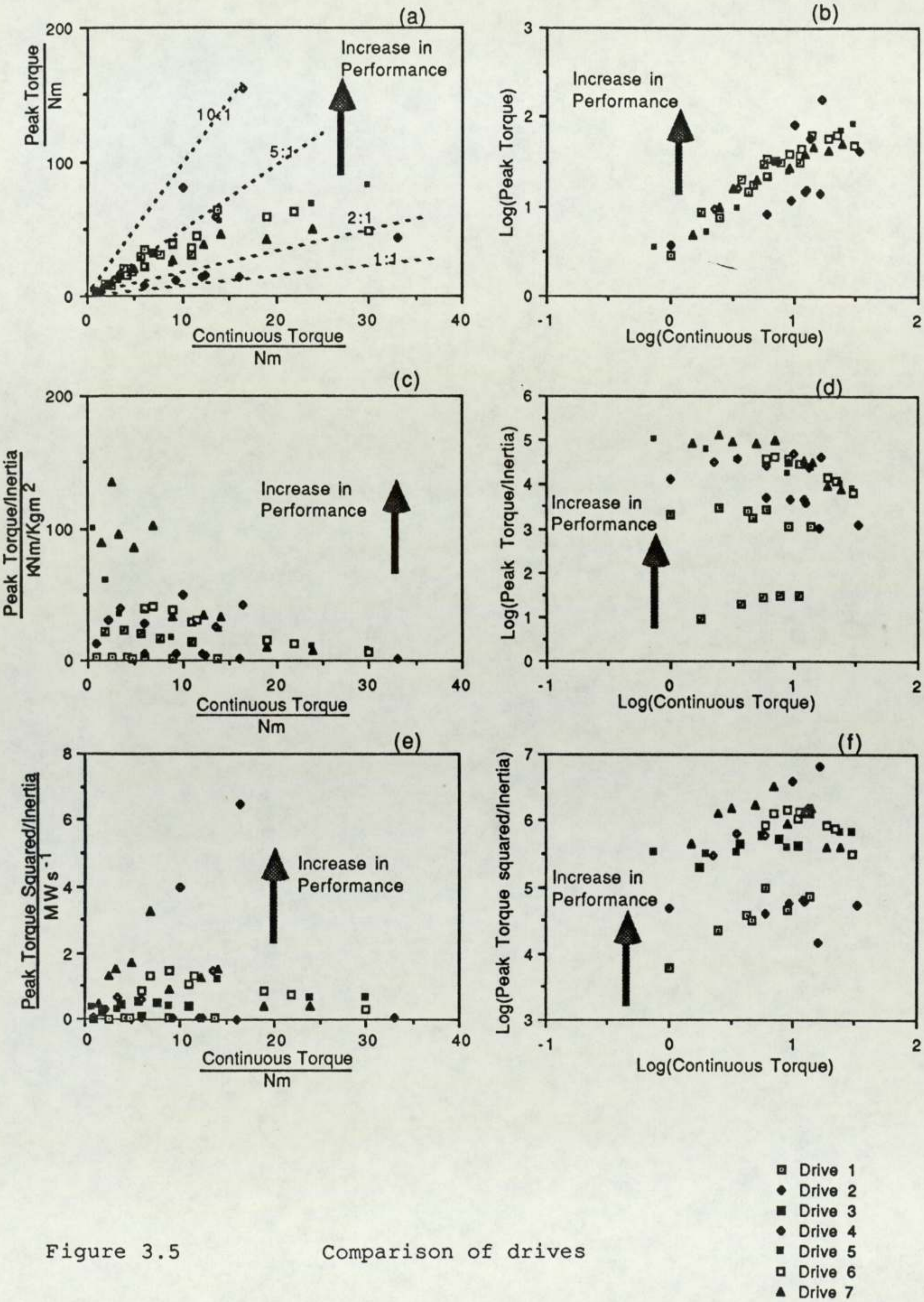


Figure 3.5 Comparison of drives

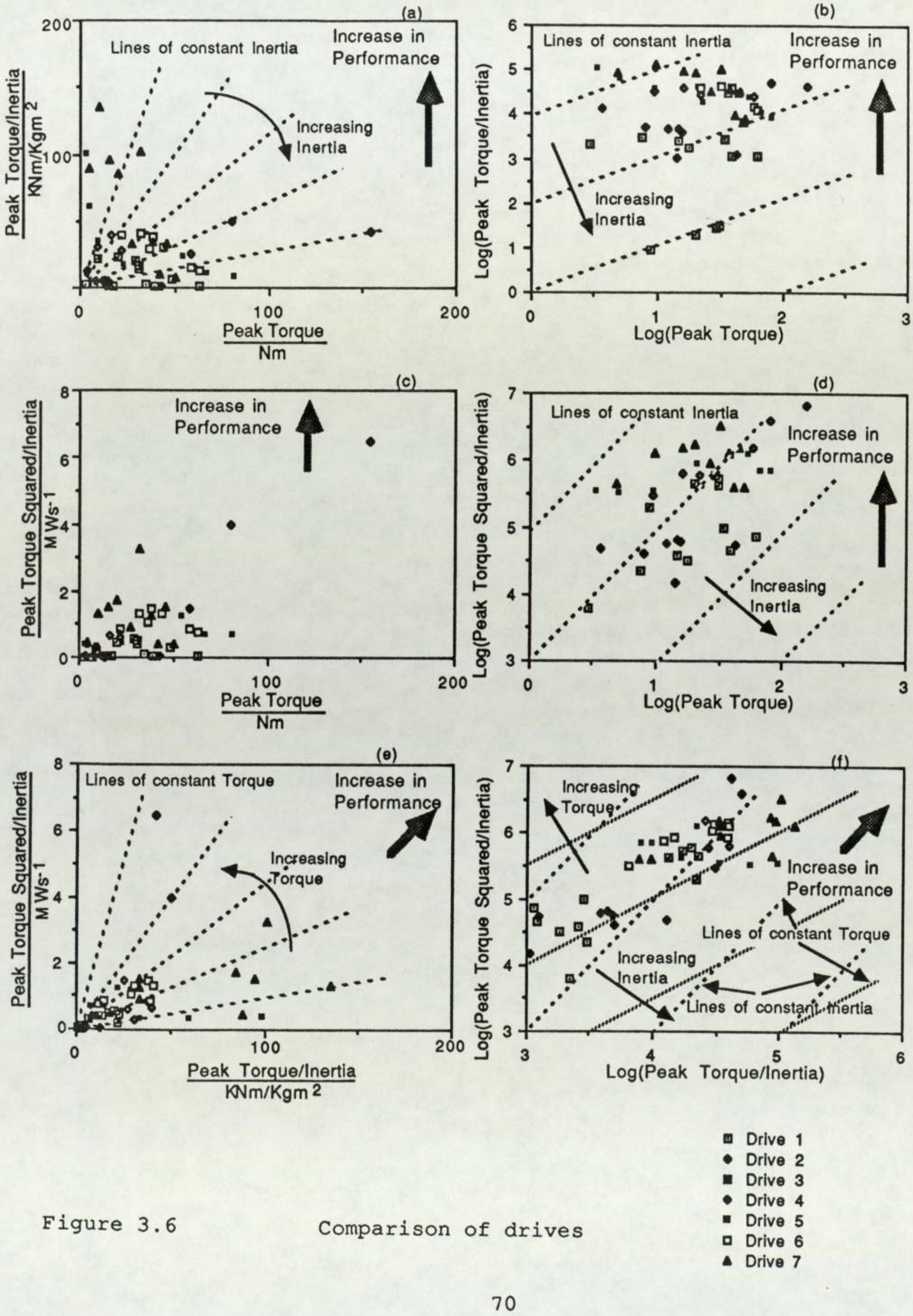


Figure 3.6 Comparison of drives

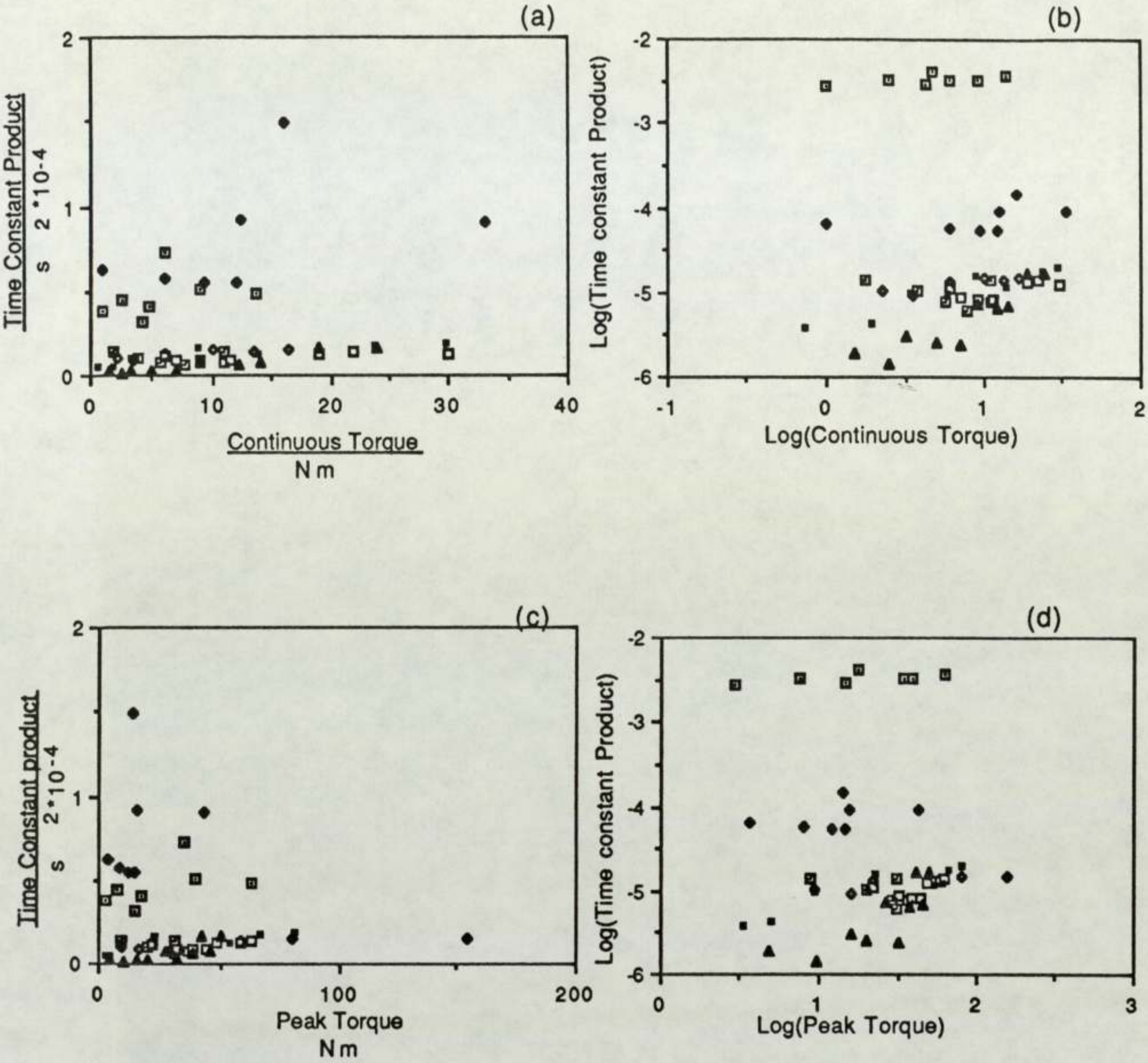


Figure 3.7 Comparison of drives

- Drive 1
- Drive 2
- Drive 3
- Drive 4
- Drive 5
- Drive 6
- Drive 7

CHAPTER 4

4.0 POSITION SERVO LOOP CONTROL AND RELATIVE ERROR CONTROL SCHEMES

4.1 INTRODUCTION

Chapters 2.0 and 3.0 introduced methods for selecting the highest performance electric motors and amplifiers from those available. Once selected some control scheme must be used to reduce the following errors between the output shaft and demanded position. There are numerous control strategies available and the most applicable are described within this chapter. The controller implemented upon the demonstrator rig was developed by Molins on an eight bit microcontroller card. This chapter shows the evolution of the implemented controller algorithm. The main requirement of the controller algorithm was simplicity so that mathematical analysis could be carried out (chapters 5.0 and 6.0) and that the microcontroller sample time was as small as possible. The controller algorithm also had to be of a generic nature such that it could be enhanced later, whilst retaining all the key elements required to reduce position following error.

Essentially any controller compares the demanded with the fed-back position and derives a current demand from the motor/amplifier combination aimed at reducing following position errors. As the feedback information from modern servo systems is digital, Tomasek [79] it is sensible to use a digital or microprocessor based controller.

The specification of section (1.5) required relative errors between axes to be reduced, but the controller strategies aimed to reduce the absolute following error of a single axis to its demand. A number of schemes were therefore examined which could theoretically reduce relative error. Three such schemes are discussed in detail; they have been classed the master/slave, the matched servo and the feedforward schemes.

4.2.0 CONTROLLER DESIGN

4.2.1 INTRODUCTION

A position controller can take many forms, Leighton et al [43] and Brown [14], but there were constraints placed upon the design of any controller used by the research programme by Molins plc. They developed the controller and it had to have a simple control algorithm, since the eight bit microcontroller to be used could execute a limited amount of code in the update period which was hoped to be less than 1 millisecond at the outset.

The following discussion is supplemented by results from simulation exercises carried out using the BRU-500 simulation suite described in appendix 4. In particular the S-6100 motor with a DM-50 driving a 0.01kgm^2 load over a 10 second ramp from rest to 2500 r/min was considered. Approximate figures gained from early experience, described in appendix 2 were used within the model, such as a 1ms sample period and a velocity loop bandwidth of 20 Hz.

Simulation results must be treated with care as actual systems may be unable to match the same operating conditions. For example, high order effects are neglected within the model used for simulation, producing a stable system for a given set of gains whilst the real system may be unstable.

The discussion on controllers assumed that the drive amplifier controlled current accurately with a high gain current loop such that the motor transfer function may be assumed to be K_t/J_Ts, in the absence of friction effects.

4.2.2 PHASE LOCKED SYSTEMS

The simplest form of position control is where the phase error of a system is fed directly to the current amplifier of a drive as shown in figure (4.1). These systems are often described as phase locked loops as they were originally derived from telecommunications systems used to boost weak signals.

Phase locked servo systems can be traced back to early work by Millar, 1968, [49], Moore, 1973, [50] and Smithgall, 1975, [67]. Tal [69,73-76] also caried out extensive work in this field during the mid 1970's. Phase locked loops inherently have 180° of open loop phase shift at low frequency and are therefore difficult to stabilise. The systems tend to be very oscillatory in nature especially during load changes. Tal's work attempted to accurately model phase locked servo systems, thus predicting and extending the stable region. The phase error detection may be carried out in many different ways; in the past product detectors, phase frequency detectors and counter detectors have been used, Geiger [31]. The phase detector multiplies the demand and feedback sinusoidal signals (which are often derived from square waves) together and feeds the resultant through a low pass filter, and it requires an auxiliary "accelerator" as no accelerating or decelerating demand is produced, McLaren [47] and Eapen et al [22-24]. The phase frequency detector provides an accelerating demand but it exhibits a severe non-linearity in the frequency domain, Le-Huy [42] and Margaris and Petridis [45]. The most popular system for motor control uses a counter and analogue to digital converter, Elkington [26]. This system is particularly useful when digital feedback is employed. The output to the motor is derived by counting the difference between feedback pulses and demand pulses (from a accurate frequency input). The input to the motor is therefore proportional to absolute position error. In order to overcome the stability problems associated with phase locked servos many complex control strategies have been used: Prasad et al [59] suggested a system which switched between velocity and phase-frequency detector systems dependent upon the frequency or phase error in the system, and Shaderma [66] discussed a system which switched between five different control modes dependent upon the relationship between input and output frequency and/or phase.

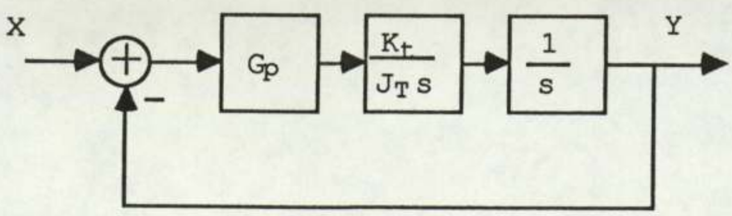


Figure 4.1 A phase control system

The system of figure (4.1) was simulated using the S-6100 motor with a 0.01kgm^2 load. The position loop gain, G_p, was a pure proportional gain for simplicity although commonly it will include a lead-lag, lag-lead or proportional plus integral plus derivative (PID) filter to enhance control. With the position gain greater than 2Arad^{-1} the system was unstable and figure (4.2) shows the time response with a gain of 0.5Arad^{-1}. With this gain following the proposed profile the position error is only held to within 4.2 radians which is more than 1500 times the required accuracy and as previously mentioned the system is unacceptably oscillatory. Reduction in phase error would require increased gains which is not practicable due to instability problems; thus this form of control would not be of benefit to the project.

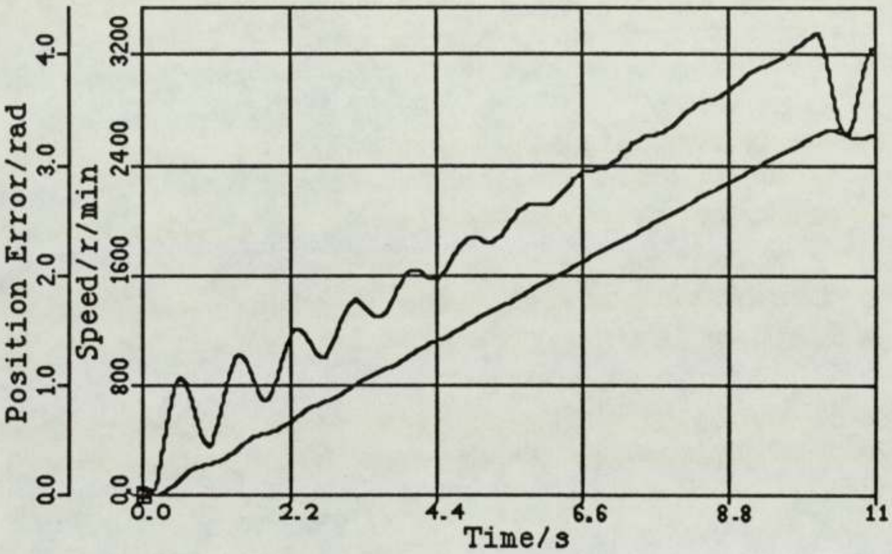


Figure 4.2 The time response of a position control system with a gain of 0.5Arad^{-1}

4.2.3 VELOCITY CONTROL

A velocity feedback system can produce damping in a position loop. On its own velocity feedback can control velocity but not position accurately. The integrator due to the load pole is converted into a high frequency pole by the addition of feedback. Figure (4.3) shows such a system and figure (4.4) the inability of such a system to control position following error to the required accuracy. The gain G_v of the velocity loop has been set to 10.8A/rads^{-1} which relates to a -3dB bandwidth of

20Hz which is a reasonable value from early experimental work, appendix 2. This value will be used in the remaining discussion.

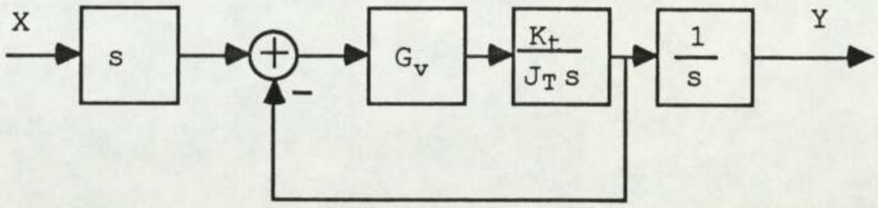


Figure 4.3 A velocity control system

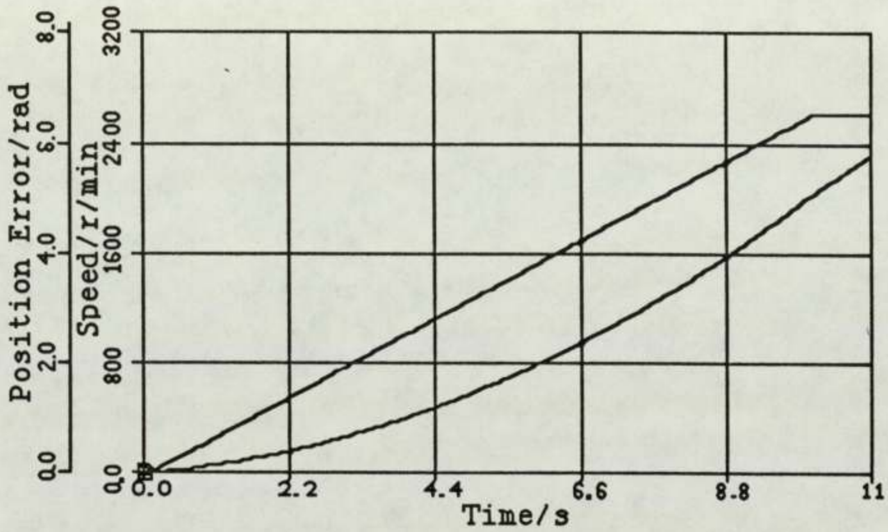


Figure 4.4 The time response of a 20Hz bandwidth velocity control system

4.2.4 COMBINED POSITION AND VELOCITY CONTROL

It is possible to use the position control system and velocity control system together. The velocity feedback acts as damping to the position control system allowing higher position loop gains than before. In this form of control as shown in figure (4.5) the position error increases to force the velocity of the system to the correct level and there must therefore be an inherent phase error in the system. In simulation it was possible to leave the velocity loop bandwidth as in section (4.2.3) whilst increasing the position loop gain, G_p, to $1500s^{-1}$ before instability occurred. Figure (4.6) shows the time response with G_p set to $1200s^{-1}$; the following error is held to within 0.22 radians which whilst an improvement is still nearly 90 times too large. Thus this form of control is also unsuitable for the particular application.

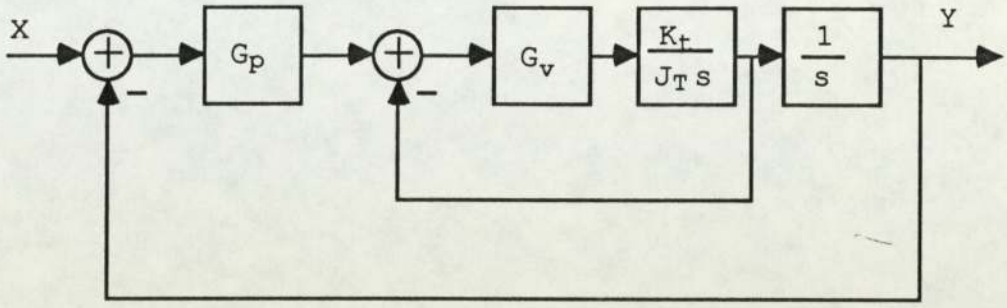


Figure 4.5 The combination of velocity and position control

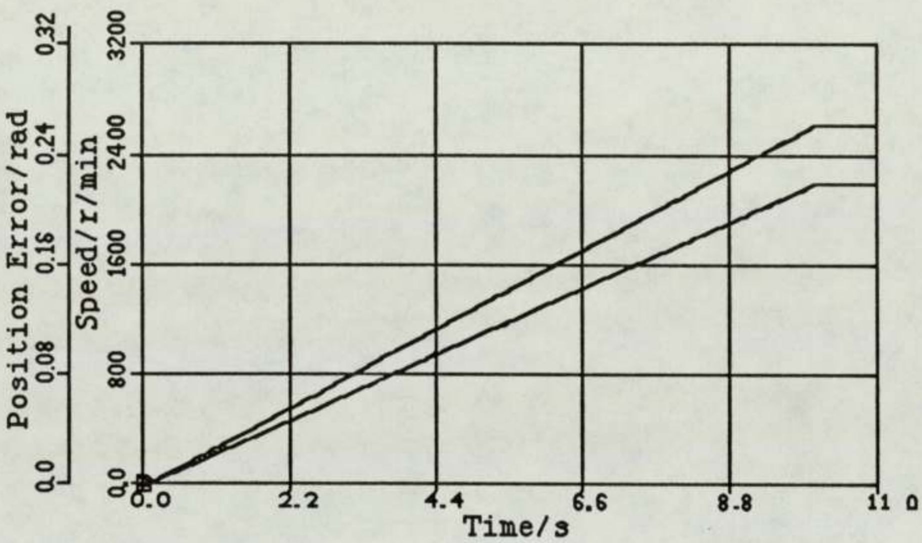


Figure 4.6 The time response of a 20Hz bandwidth velocity loop and position control with a gain of $1200s^{-1}$

4.2.5 VELOCITY FEEDFORWARD CONTROL

The major drawback of the scheme described in section (4.2.4) is that the position error has to "drive" the velocity loop. This problem can be overcome by feeding forward a signal representing the velocity demand, that is G_f multiplied by the derivative of position demand (velocity demand), figure (4.7). Feedforward does not effect loop stability and behaves like an external disturbance which enhances performance. For rapid incremental moves G_f is typically 0.75 to avoid overshoot, but for this application it is sensible to let G_f be unity to reduce steady-state position following error.

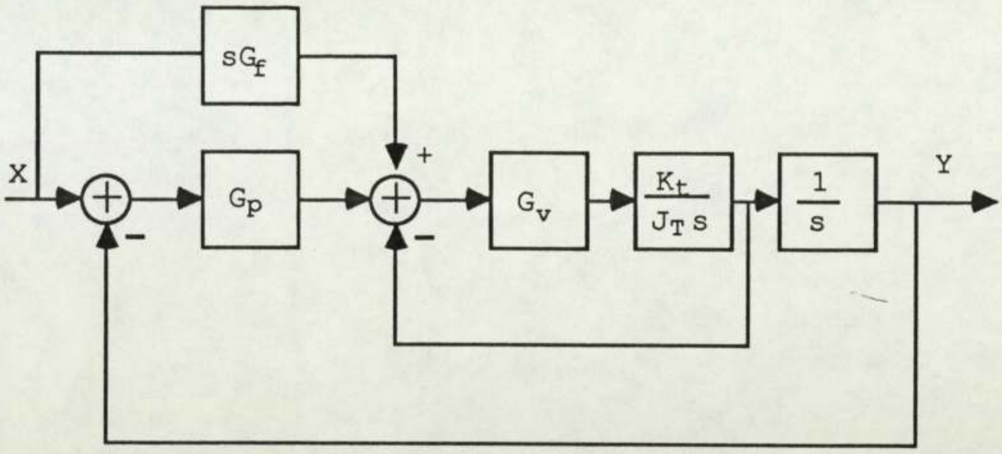


Figure 4.7 Combined velocity, position and feedforward control

With the velocity loop as before and unity feedforward a position loop gain of $350s^{-1}$ is required to achieve the specification on following error as shown in figure (4.8). Low magnitude high frequency position error oscillations are seen on figure (4.8) which suggests that an actual system with these gains may exhibit unacceptable resonances. This control stategy is the least complex found using simple simulation that could meet the specification. Thus this general form of control was implemented upon the microcontroller.

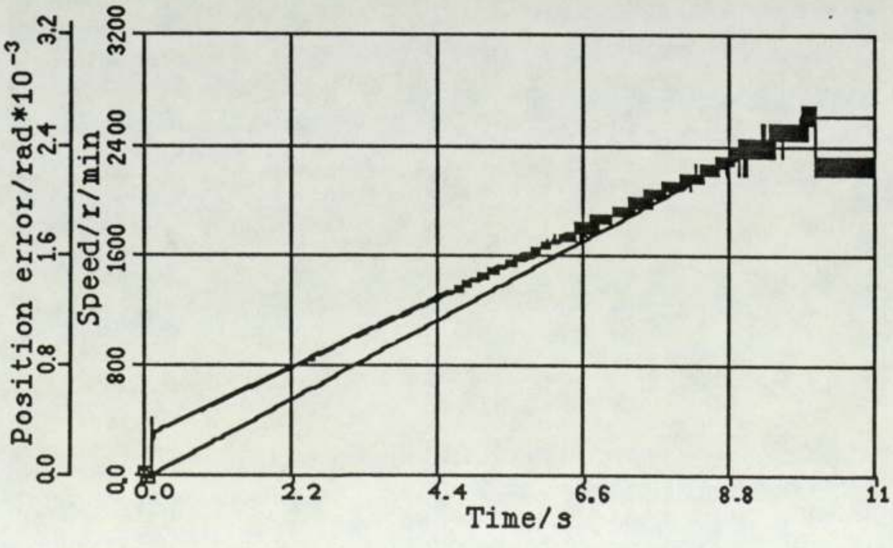


Figure 4.8 Time response of the combined 20Hz bandwidth velocity loop, position loop of $350s^{-1}$ gain and unity feedforward control

4.2.6 THE USE OF CONTROLLER FILTERING

The control algorithms discussed in sections (4.2.2)-(4.2.5) used pure proportional gains within the loops. Most modern servo controllers contain some form of filtering to extend the bandwidth of the system. PID and lead lag networks are the most common.

PID control is the most common form of control and vendors offer it in both velocity and position loops. The integral term is used to reduce steady-state errors within the system, whilst the derivative term is used to increase speed of response. The drawback of integral gain is that it can make a system sluggish. In the velocity loop it acts identically as the position loop and it increases open loop phase shift which will limit maximum gain before instability. In the position loop it is highly undesirable as the phase shift associated can cause serious stability problems. The derivative term can be used to extend the bandwidth of the system in theory but in practise it can cause problems as it is a "noise" amplifier; this is particularly problematic with sampled systems. Thus it was decided that PID control would not greatly improve the system response.

Lead-lag networks have been used in the traditional control design techniques using Bode plots. A lead network acts like a derivative term and a lag network like a integral term in the PID controller. Simulation suggested that the use of a lead-lag network would enhance the control of a system but only marginally, and substantially more calculations would be required by the microprocessor to implement this form of control. It was therefore decided not to use a lead-lag network.

Other control methods include the use of filters to smooth quantisation noise from the output of the digital controller. "Notch" or simple pole filters may be used to cancel any high frequency resonances in the system but they can add additional phase shift to the system which will degrade performance. Digital Kalman filtering techniques may be employed but again additional computing time is required.

Thus the controller implemented had pure proportional velocity and position loops with unity velocity feedforward implemented in software, with gain settings being controlled by analogue operational amplifier circuitry. There was also included the possibility of including analogue filtering if required to smooth any quantisation noise. This system was designed and built by Molins plc and was classed the Dservo controller.

4.3 SPECIFICATION OF A SERVO SYSTEM

For later work it is necessary to specify the performance of a servo loop in some way. Performance for the application of section (1.5) was measured as the relative error between drives. Unfortunately it is difficult to assess the following or relative error of a system and control engineers have difficulty relating to such a term as it refers to the type of input. A better performance indicator is bandwidth. For the purpose of this thesis bandwidth is assumed to be that frequency where the closed loop gain of a system has fallen by 3dB of its steady state value. It is also useful to quote the closed loop phase lag at the bandwidth as this gives an indication of the stability of the system; generally the smaller the phase lag the greater the stability.

With reference to the controller of section (4.2.5) it has proved useful to define controller performance in terms of the velocity loop bandwidth (quoting the phase shift) and the position loop gain in rads^{-1}/rad. These parameters can define both required performance by relation to following error for a particular application and actual values from a system. Thus a system's performance can be discerned from these figures as discussed in chapters 5.0 and 6.0.

4.4.0 LOCAL CONTROL SCHEMES

4.4.1 INTRODUCTION

When a system requires the relative error between axes to be reduced as in the specification of section (1.5) there are a number of "local" control schemes that may be used to reduce relative errors. Thus far the controllers discussed were designed to reduce absolute errors between an axis' output and demand. In the absence of an unpredictable disturbing influence the absolute errors at any point along a profile can be found with a good degree of certainty. If the relative error at any point, which will be the difference between the absolute errors, is too large then a local scheme may be employed to further reduce relative errors.

Three schemes have been investigated which can theoretically reduce errors. The matched scheme and feedforward scheme rely upon being able to predict the absolute errors, that is that they are consistent along a profile. The master/slave scheme does not rely on this principle and is therefore very useful when unexpected disturbances exist in a system. The matched scheme endeavours to make the axes behave in a similar way such that the absolute errors are the same and the relative error is therefore zero. The feedforward scheme requires a complex controller which "memorises" previous moves and feeds forward recorded (or calculated) errors such that an individual axis will run at practically zero absolute error. The master/slave scheme requires a cross-couple between axes such that the slave axes compensates for the errors in the master in an attempt to force the relative error to zero.

All the schemes are discussed for the simplest situation where there are two axes driven from the same demand signal, that is a one to one relationship. The methods are equally applicable to systems with more coordinated axes and where gear ratios exist although some adaptation is required.

4.4.2 THE IDEAL CASE

Once the axes under consideration have been optimised, in terms of motor, drive and controller selection, the ideal situation would be were the absolute and therefore relative errors between input and output are negligible. Figure (4.9) attempts to represents this situation for one point on the profile. The arrows point to the output position of the motor shafts which are rotating (anti-clockwise in the case shown). The two axis vectors are shown parallel to the demand vector and hence the absolute errors are zero. Obviously if the absolute errors are zero then the relative errors are also zero.

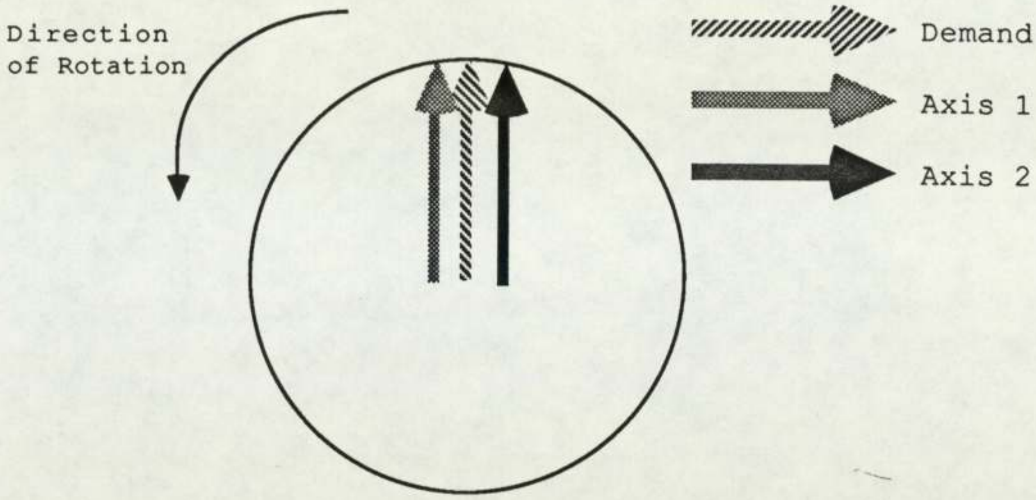


Figure 4.9 The ideal case

4.4.3 A PRACTICAL SITUATION

The absolute errors in a system will rarely be zero; figure (4.10) shows a practical situation when the demand is increasing and the two axes lag behind the demand. It would be unusual for the relative error to be zero as the dynamics of two differing axes would suggest that one axis (axis2) would lag further behind than the other (axis1). If e1 is the absolute error of axis one and e2 that of axis two then since e1 < e2, the relative error is e2-e1, that is less than the largest absolute error. This "snap-shot" is for one point on the profile only but generally the absolute errors will be in the same proportion but of different magnitude as the demand varies over a set profile.

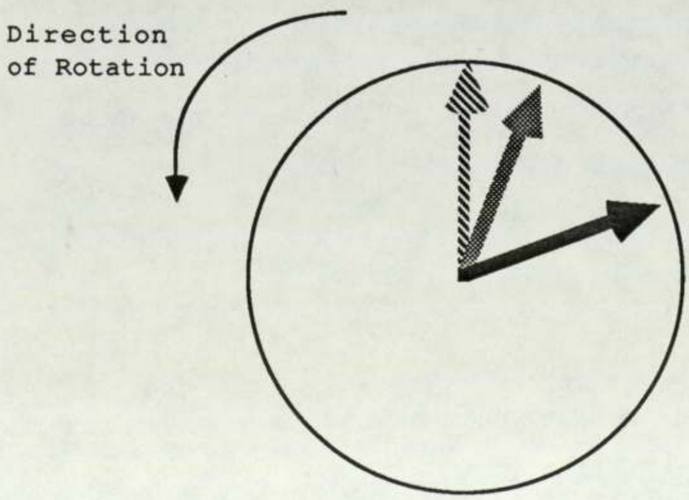


Figure 4.10 The practical case

4.4.4 THE MASTER/ SLAVE SCHEME

One method of decreasing the relative errors is to make one axis the master to the other. In this situation the drive with the largest absolute error follows the demand whilst the drive with the smaller absolute error has the output of the first axis as its demand. As long as the input does not vary rapidly then the relative error is reduced to the smallest absolute error, e1, as shown in figure (4.11). Thus the relative error is reduced from e2-e1 to e1. Therefore for this scheme to be of practical benefit e1<e2-e1 or e2>2e1, that is there must be at least a difference in absolute errors of a factor of two between axes. This system is useful as it can overcome both predicted and non-predicted errors in the system, that is, disturbance torque effects can also be negated.

The implementation of such a system can be achieved in a number of different ways. The simplest implementation is to use the encoder pulse train of the master as the slave's input demand, but this implementation requires hardwiring which is inflexible. The method used on the Molins Dservo equipment was to pass errors between two axes on a parallel bus. Switch settings enabled the microcontroller to decide whether it should feed information from the bus or interpret information from it. The master axis monitored its position error and at clocked intervals passed the value to the slave which added the additional signal to its demand. In this way the systems could be fed from the same demand.

Gear ratios may be implemented with either scheme by use of division/multiplication of the transferred information. If the master encoder pulse train fed to the slave is multiplied by 245/13 for example then a 245:13 gear ratio exists. The common range for gear ratios is 0-255:0-255 which can be handled by a microprocessor. For the other scheme if the axes demand were in a 245:13 ratio as before then the passed error information must also be multiplied by 245/13.

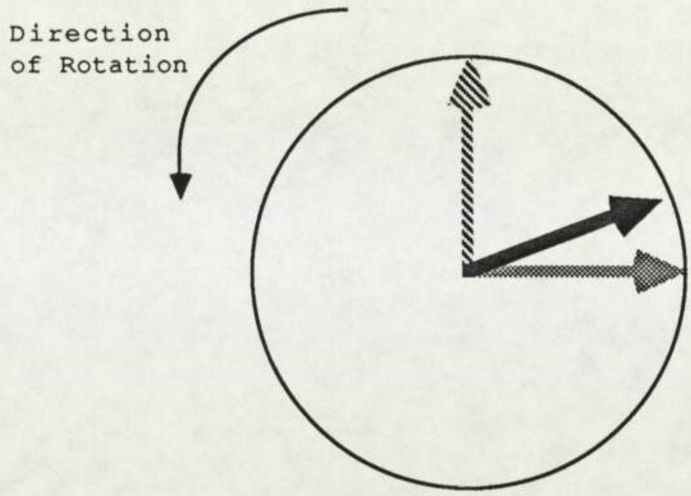


Figure 4.11 The master/ slave scheme

4.4.5 THE MATCHED SERVO SCHEME

If the drives have well defined errors over the total profile and the errors are in an approximately fixed ratio then the matched servo scheme may be used. This is a simple and cost effective way of reducing errors. The scheme simply requires the degradation in performance of the best axes to that of the worst, that is make e1=e2, figure (4.12). Thus whilst axis one's absolute error increases the relative error is practically zero. The scheme requires that the axes all resemble accurately one another and if disturbance torque exist on only one axis then the scheme will reduce performance.

Matching may be achieved in many ways. The simplest way using the control scheme of section (4.2.5) is to make the velocity and position open loop gains the same. This assumes that the friction forces are negligible in all the drives, which if high accuracy is required they will not be. Thus it is more suitable to ensure that the velocity loops of the axes have the same bandwidth by adjusting the gain appropriately and monitoring the -3dB point on the frequency response and to set the position open loop gains the same. A further method of matching applicable to fixed speed applications is to run the axes at fixed speed and simply reduce axis 1's gain until the monitored relative error is negligible.

This scheme is difficult to accurately implement in the presence of disturbing torques or in situations where a gear ratio is required although theoretically the method can still reduce relative errors.

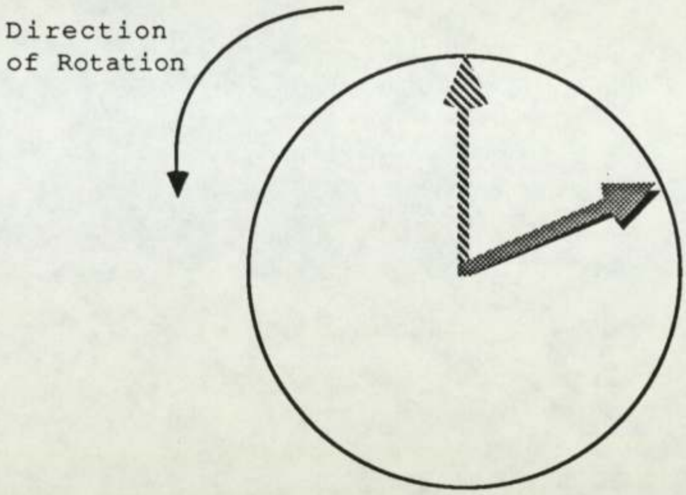


Figure 4.12 The matched scheme

4.4.6 THE FEEDFORWARD SCHEME

If the errors in a system can be predicted in any way then they may be added as an additional input to the system. Figure (4.13) shows how such a scheme may work. Axis 1 has an error e1 which can be predicted such that it is added to the demand D, and the new demand becomes D + e1. The absolute position of the axis with no feedforward is D - e1, such that when the demand is modified the position of axis 1 becomes (D+e1) - e1 = D. Thus the error has been reduced to zero, and if all axes in a system are controlled in this way then all relative errors will be reduced.

The method is very powerful but it relies upon accurate information on the expected error. Any inaccuracies in this information will directly translate into inaccuracies in the output and if the information is grossly incorrect then the system may be degraded further than without feedforward.

Many methods have used feedforward techniques and some of these will now be discussed. The control scheme of section (4.2.5) used the simplest form of feedforward, that is a velocity feedforward term was derived by differentiating the demand. A current feedforward term may also be derived if there is accurate knowledge of the load inertia. This will reduce errors when accelerating the load and it is derived by differentiating the velocity demand and multiplying it by a constant to take account of the inertia and motor torque constant.

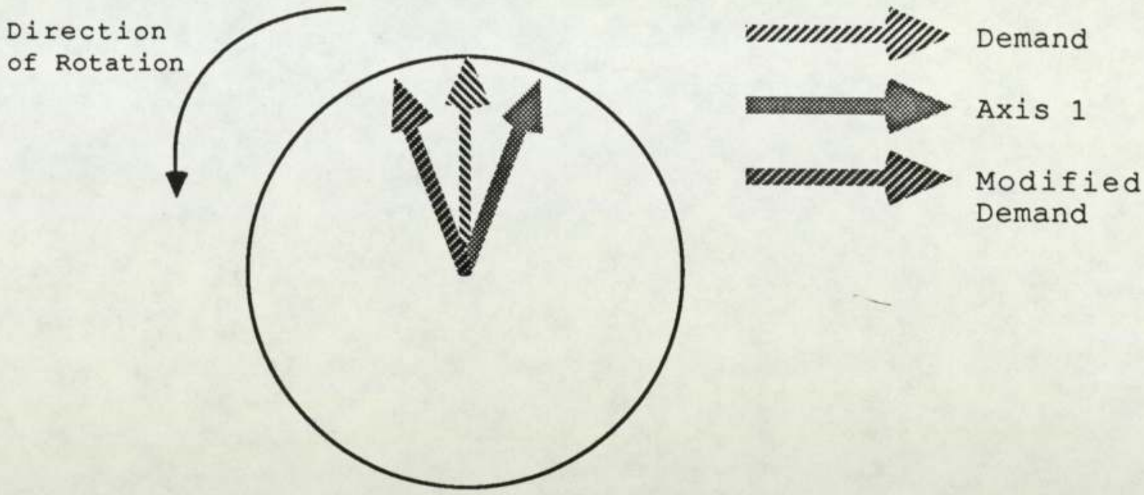


Figure 4.13 The feedforward scheme

Full feedforward control uses a model of the system that can predict the error for any given input and then add this feedforward term to the demand. This form of control requires a very accurate model which may require long calculations that cannot be achieved by a microprocessor within the sample period. The model will not be completely accurate so that it is impossible to eliminate the following error completely. Rather than predict error it is also possible to feedforward recorded errors.

Rees-Jones [60] proposed a scheme for repetitive profiles whereby the profile is split into a number of increments and the error is recorded at these increments. As the profile repeats a combination of the demand and previous run errors are outputted as the axis' demand. The system uses an averaging process such that the error progressively reduces as the system "learns" the error at each point in the profile. This control scheme is very useful as it will compensate for changes in the system such as wear or temperature rise.

4.4.7 OTHER CONTROL SCHEMES

Research in recent years has concentrated upon non-classical control techniques. Such schemes are the subject of research projects in themselves and will therefore only be mentioned here.

Robust control attempts to achieve uniform system response when parameters change such as the effects of mechanical torsional resonance which is described later in section (6.4.3). Robust control provides a general form compensator which may be a notch filter consisting of a complex pair of zeros and two real poles in an attempt to cancel the complex poles of the motor and load mechanics. Robust control is sometimes implemented by several control loops, where the inner loop is aimed at modifying the behaviour of the system and the outer loops control the position.

Non-linear designs techniques may also be used such as adaptive control whereby the compensator characteristics are altered by the controller itself, that is, system gains are altered on-line. Such systems require large amounts of computing to be implemented and as such would not be effective for the type of problem addressed in this work. They can control effectively systems with slow variations but the sample time is often very large due to the lengthy calculations required.

4.5 CONCLUSIONS

This chapter has reviewed the common methods that may be implemented to accurately control position of servo motors with current feedback amplifiers. The controller topology was chosen to close a velocity loop within a position loop and to add unity velocity feedforward. The loops were implemented using pure proportional gains. More complex algorithms were not utilised as it was thought that the additional computation time required did not warrant the minimal increase in performance. Also the simple loops are easy to analyse mathematically as in chapters 5.0 and 6.0.

Local control schemes have also been discussed. The matched scheme is the simplest and easiest to implement but it cannot overcome the effects of disturbances on individual axes. The master/ slave system is equally flexible but requires more complex controllers. The feedforward system is very flexible and copes with system variations over time but is best suited to repetitive moves. Thus only the matched and master/ slave schemes were implemented on the demonstrator rig described in chapter 8.0.

CHAPTER 5

5.0 LIMITS ON SERVO SYSTEM PERFORMANCE DUE TO SAMPLING DELAYS AND PHASE SHIFTS

5.1 INTRODUCTION

Selecting a servo drive that is capable of satisfying general load torque requirements is a relatively simple task as discussed in chapter 2.0. These methods however take no account of the required accuracy of profile following during any motion. Accuracy is dependent upon the servo loop performance which is essentially dependent upon the closed loop gain within the loop; the higher the gain, theoretically, the lower the following errors. It is difficult to calculate the profile following ability of a drive system when coupled to a given load. Indeed it is often not clear what performance is required to meet a particular specification. Conventionally, extensive tests on the selected drive/ actuator are needed to identify achievable accuracy/ performance, however a significant saving in time or cost is possible by using simulation modelling to test design concepts prior to implementation. This chapter introduces a formal method of specifying the required performance from a drive system by the use of simulation. Once performance criteria are identified a drive system must be selected; one constraint on performance is the maximum gain that can be applied to the servo loop, before some limiting factor is reached.

The chapter also develops methods for determining limits on servo loop gain. Conventional frequency domain analysis is reviewed with reference to a simple servo drive system, and it is shown how the performance can be related directly to the digital sampling period within the servo loop. The method is extended to show how an actual system can be quickly tested for performance with the aid of simulation.

5.2.0 ASSESSMENT OF REQUIRED PERFORMANCE FROM A SERVO DRIVE SYSTEM

5.2.1 INTRODUCTION

For a drive system required to move between two points (incremental mode) it is normal to require rapid movement and accurate holding at end points with minimal overshoot. It is unusual to require accurate profile following during the move. For the particular continuous application profile following error must be minimal at all times.

Chapters 3.0 and 4.0 suggested methods of selecting high performance drives and controllers, but there was no mention of ability to reduce following error. It is difficult to assess the ability of a particular drive to reduce errors but it is possible, as shown later in this chapter and chapter 6.0 to obtain values for probable velocity loop bandwidths and position loop gains. This section shows an approximate method, using simulation, to discover the required velocity loop bandwidth and position loop gain to achieve a required following error.

Velocity loop bandwidth and position loop gain have been used as they have proved to be the easiest parameters to define for a servo system. The velocity loop bandwidth for actual systems was taken as the -3dB point on the closed-loop gain frequency characteristic and it was found useful to define the phase shift at this frequency which also gives an indication of both stability and ability to close a position loop around it.

5.2.2 METHOD

The method used to define velocity loop bandwidth and position loop gain is very straight forward. A simple control system is shown in figure (5.1).

The velocity loop is defined as a simple pole (the current loop has infinite gain and proportional gain on velocity error), and the position loop has proportional error gain with unity velocity feedforward. The model input can then be exercised in simulation with the required profile and position following errors monitored. For any given value of velocity loop bandwidth ($1/\tau$) there will be a corresponding minimum position loop gain that will be able to maintain following position error within specification. In the simulation environment variables can be rapidly changed and the model re-exercised, so results can be achieved by trial and error.

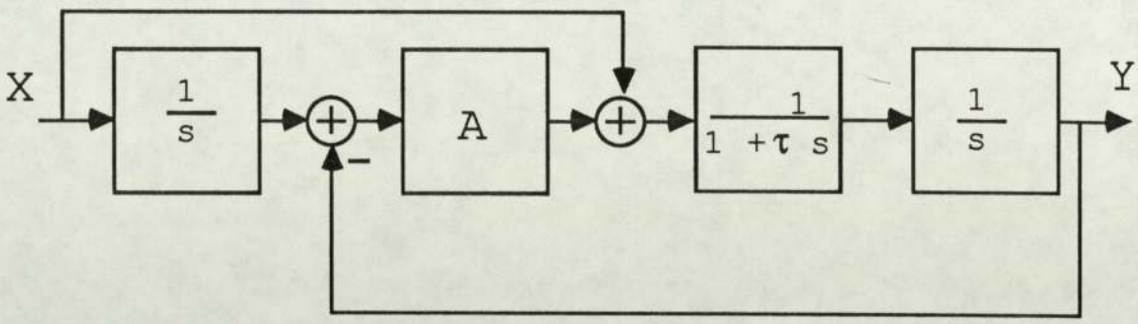


Figure 5.1 A simple control scheme

This model however, is over simplistic since no frictional effects are considered, the current loop is assumed "perfect" and sampling delays are neglected. All these effects cause phase shift which degrade performance and can cause instability at high gain settings; the model of figure (5.1) cannot be unstable (provided $A>0$).

An artificial way to include additional phase shifts is to replace the original simple pole representation of the velocity loop by a second order system of the form $1/(s^2/\omega_n^2 + 2\zeta/\omega_n + 1)$. Experience has shown that actual servo system velocity loops have similar frequency responses to second order systems, and Al-Anbuky and Aubaidy [2] and McLaren [47] used this representation for analysis. For a flat top response ζ is set to $1/\sqrt{2}$. The earlier method can be repeated to locate effective position loop gain and velocity loop bandwidth, and stability limits can be determined. Whilst being an artificial representation of the system many of the additional high order effects are approximately lumped by this representation.

5.2.3 APPLICATION

The technique was applied to the engineering problem described in section (1.5), that is to control the following error to below 0.0025 radians during a 0-2500 r/min ramp over 10 seconds (the modified specification). Program listing (5.1) shows the ACSL source code for this application.

Results are shown in table (5.1) for the simple pole velocity loop representation and table (5.2) for the second order representation.

```
PROGRAM ORDER2
      "COPYRIGHT D.R. SEAWARD 1988"
      "POSITION LOOP AROUND A SECOND ORDER SYSTEM "
      "TO INVESTIGATE BANDWIDTH REQUIREMENTS"
INITIAL
      "DEFINE CONSTANTS"
      CINTERVAL   CINT  = 0.1
      MAXTERVAL   MAXT  = 5E-5
      CONSTANT    TSTOP = 15.0,      TOR   = 0.005
      CONSTANT    EGAIN = 1,         RSTOP = 10
      CONSTANT    RAMP  = 26.18,     PI    = 3.1415927
END $"OF INITIAL"

DERIVATIVE

  PROCEDURAL(X=T)
      IF(T.LT.RSTOP)X=RAMP*T
  END $"OF PROCEDURAL"
      X1    = X + EGAIN*EPOS
      VEL   = REALPL(TOR,X1,0.0)
      POS   = INTEG(VEL,0.0)
      EPOS  = INTEG(X,0.0) - POS
      TERMT(T.GE.TSTOP)
END $"OF DERIVATIVE"
END
```

Program Listing 5.1

Natural Frequency /Hz	Position Loop Gain Required to meet specification/ $rads^{-1}$/rad
1	1750
5	325
10	165
20	75
30	57
50	33
70	24
100	17

Table 5.1 Results from the simple pole velocity loop representation

Natural Frequency /Hz	Position Loop Gain Required to meet specification/ rads^{-1}/rad	Maximum Gain before instability/ rads^{-1}/rad
1	Unable to achieve	15
5	Unable to achieve	40
10	Unable to achieve	80
20	125	175
30	75	280
50	48	420
70	34	700
100	24	900

Table 5.2 Results from the second order velocity loop representation

If it is assumed that the sampling period will be approximately 1ms (a typical value for proprietary equipment) then from the later analysis of sections (5.3.2, 5.3.3), the maximum velocity loop bandwidth will be approximately 100Hz and the maximum position loop gain will be $215s^{-1}$.

In order to meet the following error specification with a 100Hz bandwidth a gain of $17s^{-1}$ (table 5.1) or $24s^{-1}$ (table 5.2) is required whilst for a $215s^{-1}$ gain a velocity loop bandwidth of 8Hz (table 5.1) or 25Hz (table 5.2) is necessary. Allowing a tolerance for uncertainty, values less than the predicted maximums should give adequate position following errors in the absence of any disturbing force, for example 90Hz and $200s^{-1}$.

5.3 ANALYSIS OF DIGITAL FEEDBACK IN A SERVO LOOP BY CONVENTIONAL CONTROL THEORY

5.3.1 INTRODUCTION

With the increasing use of microprocessors in servo control applications it is natural that the feedback transducer is also digital. The tachogenerator is being replaced by optical and magnetic encoders, or resolvers; these have the advantage that both speed and position information is available from the same transducer.

A microprocessor digital control system has many commonly listed advantages, such as repeatability, accuracy, flexibility, and self-diagnostics, but to the control loop it introduces problems. Firstly a microprocessor requires time to execute its algorithms, thus introducing a delay into the servo loop which is destabilising; most modern controllers have update times in the order of 1-4ms, with some systems as low as 150μs.

There are many phenomena which effect the maximum achievable bandwidth of a servo system, and often many are present at once. It is therefore extremely difficult to accurately predict or quantify these effects. The following is a discussion of the effect of sampling delays on the properties of a modern servo system.

The simple control system developed in section (4.2) will be used for all analysis. A block diagram of the velocity loop of such a system is shown in figure (5.2). It is a first order system, $K_t = K_{bc}K_{tp}$, and the encoder has "p" pulses per revolution, sampled every ΔT seconds.

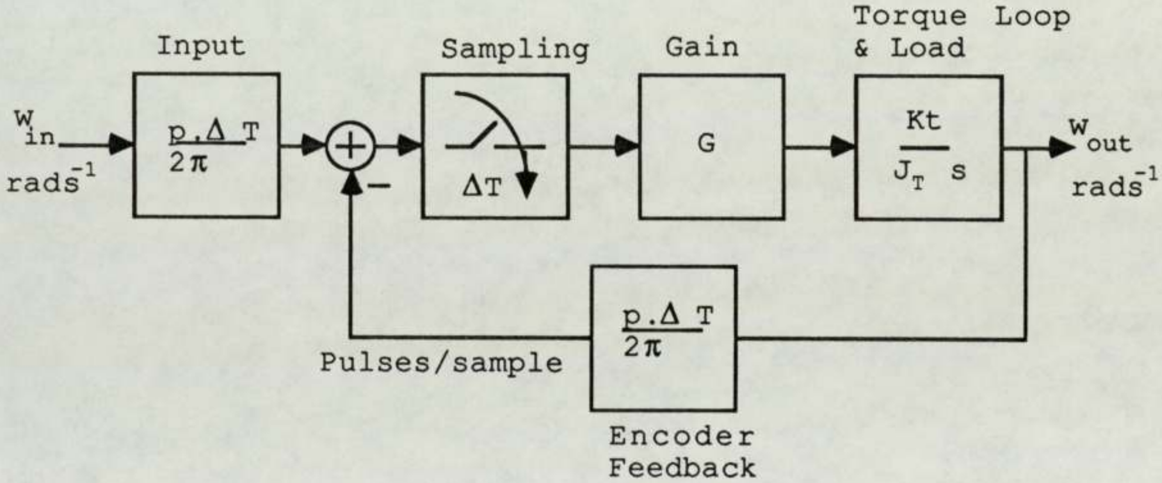


Figure 5.2 A digitally controlled dc servo system block diagram

If the sampling effect is neglected the transfer function of the system is given by:-

$$\frac{\omega_{in}}{\omega_{out}} = \frac{1}{1 + \tau s} \qquad \ldots(5.1)$$

Where the time constant of the simple model is given by:-

$$\tau = \frac{2\pi J_T}{\Delta T . p . G . K_t} \text{ seconds} \qquad \ldots(5.2)$$

5.3.2 VELOCITY LOOP BANDWIDTH LIMITATION FROM CONVENTIONAL CONTROL THEORY

Using conventional control techniques, Dorf [20], one can specify control loop stability in terms of phase and gain margins, on a Bode (open loop frequency) plot. Typical stable phase and gain margins are 35° and 8dB. The system bandwidth is approximately the gain cross-over frequency and it is therefore possible to locate a maximum stable bandwidth for a system.

The Laplace notation is useful for analogue control systems but equations takes no account of the sampling phase lags. The phase lag due to sampling is linear, such that the phase lag at the sampling frequency is 2π, and π at half the sampling frequency etc. It is therefore more convenient to introduce the normalised sample period, t_ϕ, and to use the modulus/argument notation:-

$$\frac{\omega in}{\omega out} = \frac{1}{1 + \tau.\omega\angle(\pi/2 + 2\pi.t_\phi.\omega)} \qquad (t_\phi=\Delta T/2\pi) \quad ....(5.3)$$

K is used to define the ratio of system natural frequency to sampling frequency such that $K=\tau/t_\phi=2\pi\tau/\Delta T$. Using conventional control techniques the value of K must be chosen to give a stable control loop. The phase and gain margins for the open-loop system may be calculated from:-

$$\text{Velocity open-loop gain} = 1/\tau.\omega \quad ....(5.4)$$

$$\text{Velocity open-loop phase shift} = -\pi/2 - 2\pi.\tau.\omega/K \quad ....(5.5)$$

Using this analysis one can prove that a sampling frequency of four times the system natural frequency would cause instability (gain margin = 0dB, phase margin = 0°), that is, the phase and gain cross-over frequencies are the same such that, $1/\tau\omega = 1$ and $\pi/2 = 2\pi\tau\omega/K$ giving $K = 4$.

The Bode plot defined by these equations is shown in figure (5.3) for K=10, 20 and ∞. The "reasonable" values for phase and gain margin used are 35° and 8dB respectively. At the gain cross-over frequency (Gain=1), $\tau.\omega = 1$, and therefore at the phase cross-over the frequency must be defined by $\tau.\omega$ = 8dB (2.51). The phase at this frequency is $-\pi$ so that, $-\pi = -\pi/2$ -

$2\pi*2.51/K$ or $K=10.0$. To check that the phase margin is adequate at the gain cross-over frequency, we can use the value of K, thus:-

Phase margin = $\pi - \pi/2 - 2\pi/10 = 3\pi/10$ (54°) which is adequately stable.

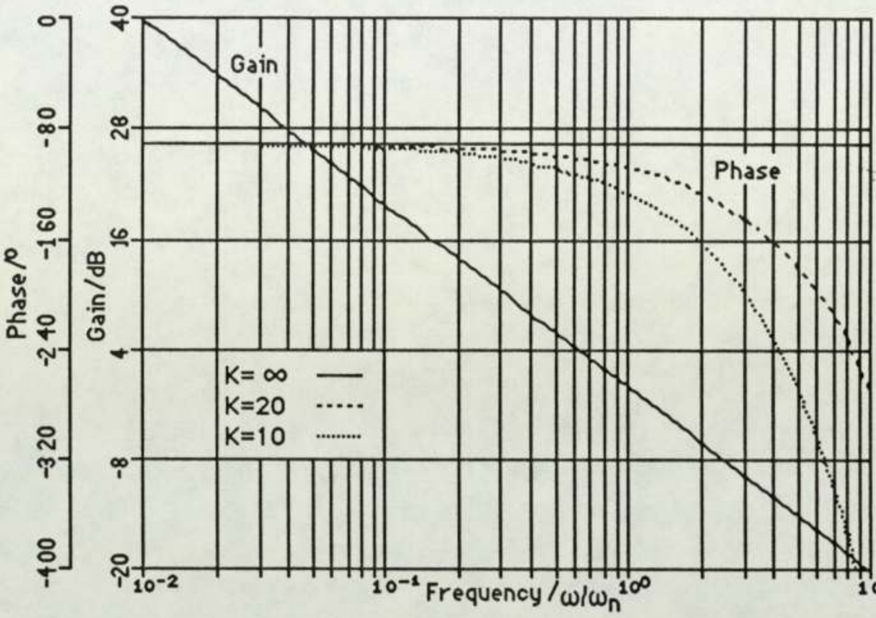


Figure 5.3 Open velocity loop frequency responses

From this analysis for an adequately stable system no attempt should be made to close a servo loop as described with gains such that the gain cross-over frequency is greater than one tenth the sampling frequency. The bandwidth of the system can be located with reference to the closed-loop equation:-

$$\frac{\omega in}{\omega out} = \frac{1}{1 + \tau.\omega\angle(\pi/2 + 2\pi.\tau.\omega/K)} \quad ....(5.6)$$

This can be split into real and imaginary parts:-

$$\frac{\omega in}{\omega out} = \frac{1}{(1 + \tau.\omega\cos(\pi/2 + 2\pi.\tau.\omega/K)) + j.\tau.\omega\sin(\pi/2 + 2\pi.\tau.\omega/K)} \quad ....(5.7)$$

and therefore in argument/modulus form this becomes:-

$$\left|\frac{\omega in}{\omega out}\right| = \frac{1}{\sqrt{(1 + \tau.\omega\cos(\pi/2 + 2\pi.\tau.\omega/K))^2 + (\tau.\omega\sin(\pi/2 + 2\pi.\tau.\omega/K))^2}}$$

$$\angle -\tan^{-1} \frac{(\tau.\omega\sin(\pi/2 + 2\pi.\tau.\omega/K))}{(1 + \tau.\omega\cos(\pi/2 + 2\pi.\tau.\omega/K))} \quad ....(5.8)$$

The above equation defines the frequency responses shown in figure (5.4) Notice the resonant peak in gain for K =10.

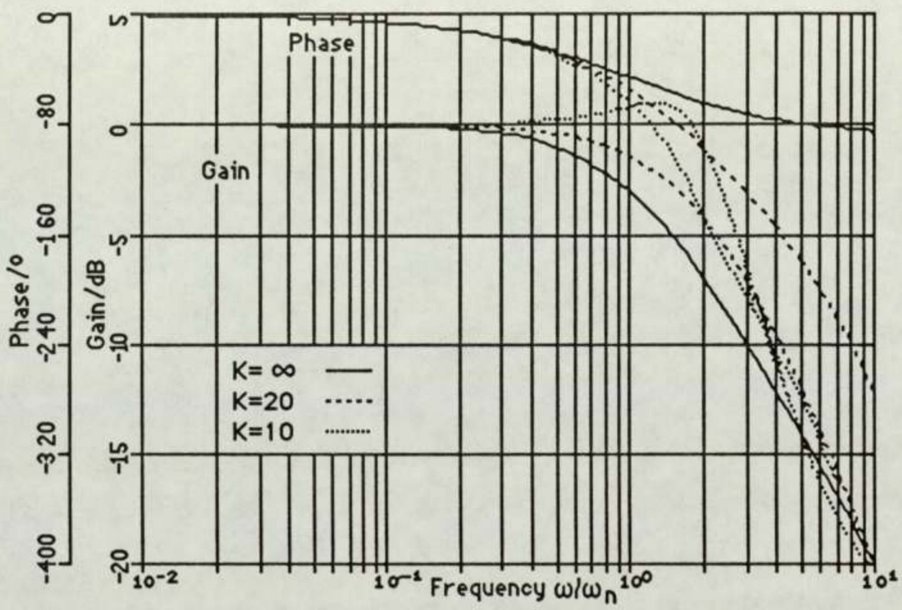


Figure 5.4 Velocity closed loop frequency responses

Using this equation it can be found that without the sampling the bandwidth and open-loop gain cross-over frequencies are the same; for the case with K=10, at the gain cross-over the closed-loop system is described by +0.84dB $\angle$ -63° and the bandwidth is -3dB $\angle$ -175° at 2.4 times the open-loop gain cross-over frequency. Thus the maximum theoretical bandwidth achievable from a velocity loop is approximately one quarter of the sampling frequency. This very high phase shift at the closed-loop bandwidth suggests that it is would be difficult to close a stable position loop around the velocity loop. A value of K=20 as shown in figure (5.4) gives a much more satisfactory response.

5.3.3 POSITION LOOP PERFORMANCE FROM CONVENTIONAL CONTROL THEORY

The standard servo loop configuration for a position loop is as shown in figure (5.5), that is, proportional gain on position error and velocity feedforward to reduce steady-state errors. The sampling delays are also evident in the position loop. The velocity feedforward term is not within the loop and so can be neglected as it does not effect stability. If the gain in the position loop is "A" then the open position loop is defined by:-

$$\text{Gain} = \frac{A/\omega}{\sqrt{(1 + \tau.\omega\cos(\pi/2 + 2\pi.\tau.\omega/K))^2 + (\tau.\omega\sin(\pi/2 + 2\pi.\tau.\omega/K))^2}}$$

$$\text{Angle} = -\pi/2 - 2\pi\tau.\omega/K - \tan^{-1}\frac{(\tau.\omega\sin(\pi/2 + 2\pi.\tau.\omega/K))}{(1 + \tau.\omega\cos(\pi/2 + 2\pi.\tau.\omega/K))} \quad \ldots.(5.9)$$

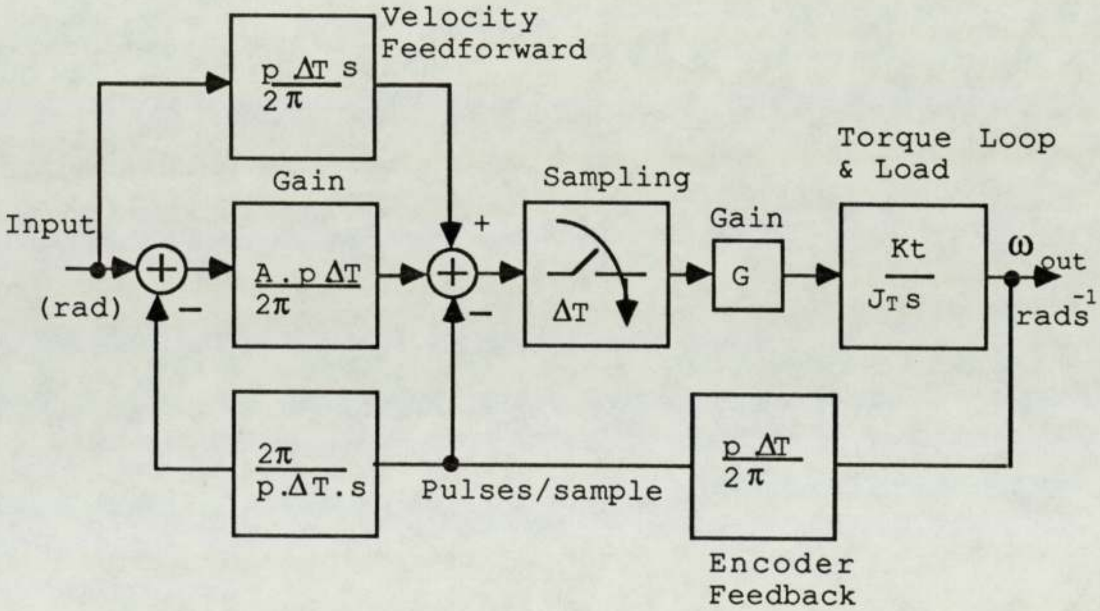


Figure 5.5 A position loop block diagram

These equations define the position loop responses shown in figure (5.6) with A/τ equal to unity. The analysis of the velocity loop can be repeated for the position loop. For K=10 the phase cross-over point is at $\tau.\omega$ =0.92, which has a corresponding position open-loop gain of $A/0.917\omega$. Therefore for an 8dB gain margin A/ω=0.365, and $A = 0.365 \times 0.92/\tau = 0.365 \times 0.92/t_\phi/K = 0.034/t_\phi$. The phase margin in this case is 59° which is well within stable bounds. Thus for a simple servo system with digital sampling the stability can be defined simply from the sampling period itself. This method does not take into consideration any secondary effects that may further reduce gain or phase lag at frequencies below the sampling frequency, which would further reduce system performance.

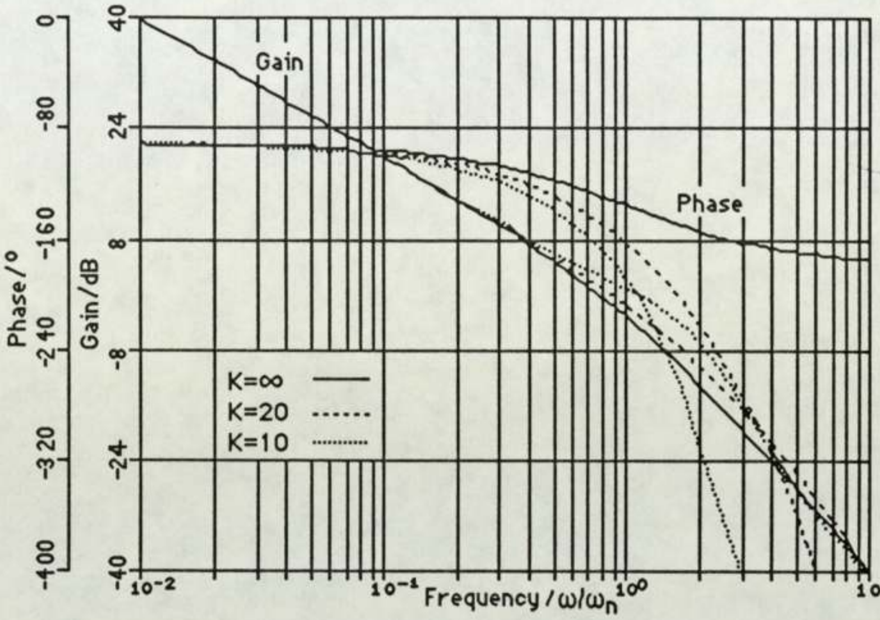


Figure 5.6 Open position loop responses

The method was repeated with K set equal to 20 and 40. With K=20 the velocity loop was found to have a phase margin of 72° and a gain margin of 14dB, that is a very stable system. The velocity loop bandwidth was found to be at $\tau.\omega$=1.6 with 81° of phase shift. For the position loop with an 8dB gain margin, and a 45° phase margin it was found that A=0.64ω =0.032/t_ϕ, that is the position loop gain is almost unchanged, from the value with K=10. Therefore whilst the inner velocity loop has a reduced bandwidth the achievable position loop gain is unchanged.

With K=40, the velocity loop gain margin was 20dB and the phase margin 81° which is very stable but the system is becoming sluggish in comparison with lower values of K. The phase shift at the bandwidth is -56°; this velocity loop therefore resembles the simple first order pole, that is unaffected by sampling. The position loop had a gain margin of 8db and a phase margin of 31.5° with A=1.296ω =0.0324/t_ϕ. This very small phase margin suggests that the system would tend towards instability and therefore a larger gain margin would be required by reducing the position loop gain.

From this brief discussion it seems reasonable to use a value of K which is between 20 and 30, and a position loop gain of $0.03/t_{\phi}$.

In practise it must be remembered that there may be many other factors which could reduce the attainable velocity loop bandwidth or position loop gain.

5.4 A SELECTION PROCEDURE FOR ACTUAL DRIVE SYSTEMS

5.4.1 INTRODUCTION

Simulation may be used to enhance the method of section (5.3) by determining an approximate figure for the maximum performance achievable from an actual drive, which has different characteristics to that of the ideal case of figure (5.5). The method produced results similar to those found on a test-rig, so the technique can be assumed to be valid.

A high performance rare earth brushless dc machine system was selected for a case study. The drive was fully characterised and a complex simulation model derived, shown in appendix 4. The academic exercise of producing the accurate model of the system took a number of man-months, due to the great care needed to validate all the steps taken, and to fully understand the system in question. This is the nature of academic research, but the time scale for producing simulation models must be reduced if it is to be of practical, cost effective benefit to Industry.

The method outlined in this section allows rapid performance prediction by using the standard control method of fitting an approximation to a measured frequency response. A simulated approximation to the measured open velocity loop is made. Simulation experiments on the approximate model are used to optimise both the velocity and position loops. The loops are optimised by finding the maximum forward loop gains that may be used prior to the loop becoming too oscillatory, when examining both time transient and frequency responses.

5.4.2 THE DRIVE SYSTEM ANALYSED

The Electrocraft Bru-500 drives were chosen for the exercise, in particular the S-6100 motor, with a DM-50 and an inertial load of 0.0026 kgm^2 were used during tests (see chapter 7.0 for a description of the system).

During tests with the system driving a 0.05 kgm^2 load it had been shown experimentally that the limit on performance was a velocity loop bandwidth of approximately 38Hz at -140°, see section (7.4). With bandwidths higher than this the system entered the non-continuous current region and automatic shut-down due to over-current occurred.

5.4.3 THE RAPID EFFECTIVE SIMULATION DESIGN PROCEDURE

The essence of the design procedure was to characterise the open velocity loop of a drive in the frequency domain and to use this as the data to derive approximate positional accuracy available from the drive. The simulation model was derived by placing a frequency response fit to the open velocity loop within a simple velocity loop.

The open velocity loop is similar to the closed current loop response with the addition of the load pole. To obtain the open velocity loop frequency response one monitors the velocity response to a sinusoidal signal into the drive's torque or current mode input. For the procedure to work the drive load must be substantially inertial in nature, that is, the viscous damping must be minimal. The simplest way to achieve this would be to directly couple the drive to a large flywheel of known inertia, thus the inertia and viscous damping of the motor rotor will be "swamped" by the load inertia.

The method can be split into a number of steps:-

1) Obtain the open loop velocity frequency response of the drive with a known large inertial load (all responses must be in the linear region of the system's characteristic).

2) Identify any sampling, current limits, digital feedback or any other non-linearity in the loop.

3) Fit an approximation to the open-loop response by use of integrators, poles and zeros and form a block diagram with these elements. Check the simulation against the actual open-loop frequency response up to approximately five times the required bandwidth.

4) Simulate a simple proportional velocity loop in close-loop with the required load and obtain the optimum forward loop gain for fast response with minimal overshoot. Also check that the frequency response has a flat gain response and slow phase roll-off (c.f. $\zeta=0.7$ for a second order system).

5) Close a proportional plus velocity feedforward position loop around the optimum velocity loop using simulation, and optimise the forward loop gain, checking that the position errors for a required profile and load are within specification.

5.4.4 THE S-6100 DESIGN EXERCISE

Frequency responses of the S-6100 motor and DM-50 drive were taken with the drive in "torque" mode, that is, a voltage at the input to the drive is seen as a demand for current. The frequency response was taken by use of a standard transfer function analyser. Velocity feedback was achieved by use of dedicated digital circuitry which converted the 8000 pulse per rev encoder signal into a voltage with a gain of 0.05 V/rads^{-1}, sampled every 0.5ms.

Figure (5.7) shows a typical result, and table (5.3) summarises other tests. All the responses were very similar with 20dB/decade gain fall-off and approximately 90° of phase shift until rapid roll-off starting at about 20 Hz. The encoder feedback signal was compensated for, as it introduced phase shift, for example 18° at 100Hz.

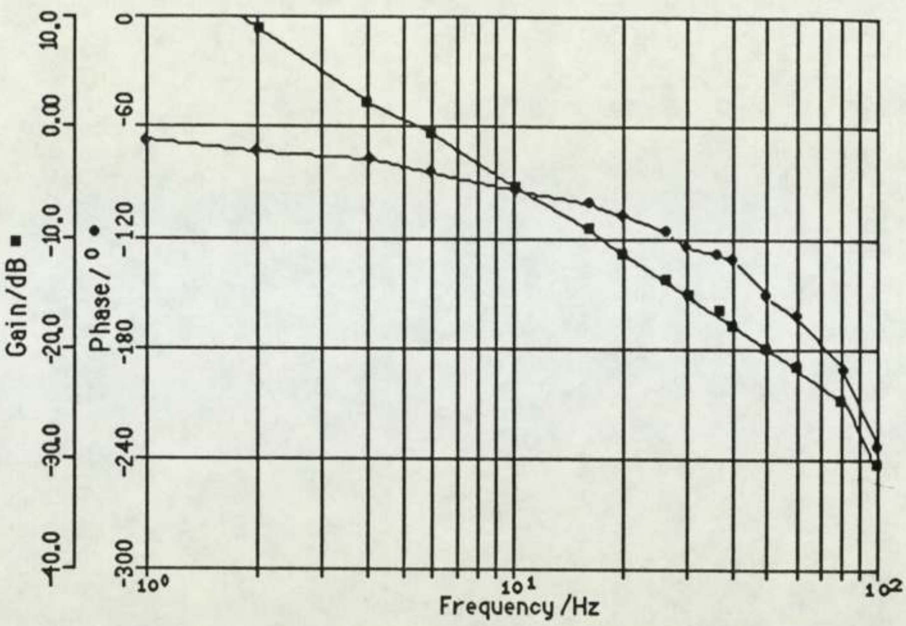


Figure 5.7 Plot number 3 of the 6100 drive's open velocity loop frequency response

PLOT NUMBER	INPUT MAGNITUDE/V	CURRENT SCALING/AV^{-1}	INERTIA /KGM^2	GAIN @ 10 Hz/Db	APPROX. ROLL OFF FREQ/Hz
1	4.0	5	0.004	-4.8	110
2	2.0	5	0.004	-4.9	90
3	1.0	5	0.004	-5.2	80
4	0.3	5	0.004	-7.0	75
5	4.0	5	0.0014	+3.7	120
6	0.5	5	0.0014	+2.5	120
7	4.0	1	0.004	-19.5	90
8	4.0	1	0.0014	-10.8	100

Table 5.3 Summary of the 6100 drive's open velocity loop frequency responses

5.4.5 FITTING AN APPROXIMATION

For a simple pole an accurate frequency approximation on log paper is to have a flat gain curve until the break frequency and then a 20 dB/decade roll-off. For the phase it is usual to have a linear curve of 90 degrees of phase shift drawn between 1/5 and 5 times the break frequency. This approximation is shown in figure (5.8). This method gives insight into fitting poles and zeros to an existing curve.

Any approximation to the real open loop velocity frequency response should include any poles and zeros up to 5 times the bandwidth of the system required. It was difficult to obtain accurate results beyond 100Hz, so a frequency curve was fitted to this value.

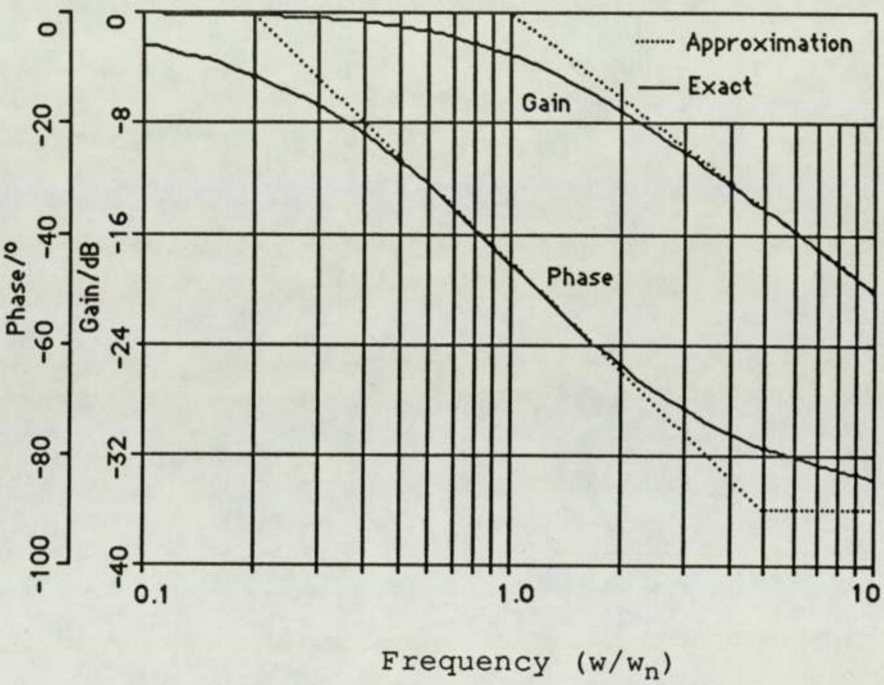


Figure 5.8 Exact and approximate frequency response of a simple pole

From figure (5.7) it is evident that the approximation for the open velocity loop has a integrator with a double pole at approximately 100Hz, the phase plot shows higher order poles but these will be neglected. There was known to be 1ms sampling and a 50A current limit in the system, so these were included in the model, which was checked by simulation to achieve the following solid line of figure (5.9).

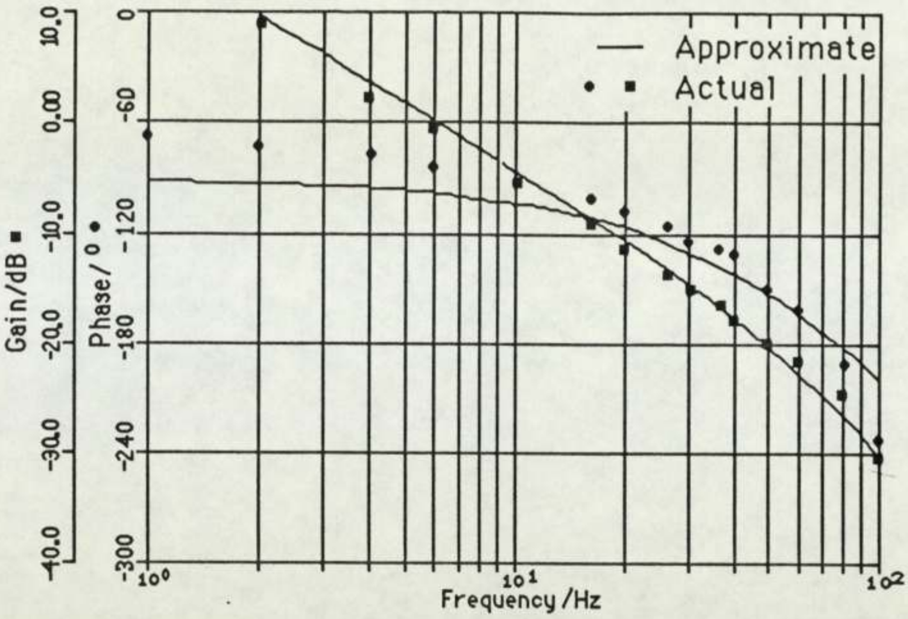


Figure 5.9 Actual and approximate frequency responses of the S-6100 open velocity loop

5.4.6 AN OPTIMUM VELOCITY LOOP

A simple velocity loop can be constructed around the open loop approximation with proportional gain on the velocity error, as shown in figure (5.10).

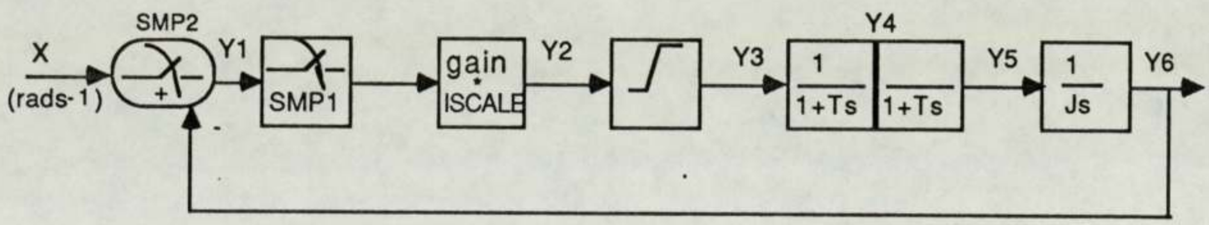


Figure 5.10 A closed velocity loop block diagram

It should be noted that ISCALE has been chosen as 5.0 A/V as the current limit of this machine is 50 A and the voltage input limit is 10V, thus the total available range is used. Since the system considered used a digital encoder any external velocity loop will use sampling techniques. A sampling rate of 1ms was chosen as this was the same as that used by the manufacturer of the equipment. The block diagram was coded in ACSL as shown in program listing (5.2).

```
PROGRAM VELOCITY_LOOP-SIMULATION
        "COPYRIGHT D.R. SEAWARD 1988"
        "SIMULATION FOR THE BRU500 SERIES OF BRUSHLESS MACHINES"
        CINTERVAL       CINT   = 0.0001
        MAXTERVAL       MAXT   = 5E-5
        NSTEPS          NSTP   = 1
        ALGORITHM       IALG   = 5
INITIAL
        JEQ     = JLOAD + JMOT
        AZ0     = 0
        AZ1     = 0
        AZ2     = 0
        AZ3     = 0
        BZ0     = 0
        BZ1     = 0
        BZ2     = 0
        BZ3     = 0
        CONVER = 4*ENC/2/PI
END $"OF INITIAL"

DERIVATIVE
        CONSTANT JMOT       = 0.0014,     JLOAD  = 0.05,      ISCALE  = 5.0
        CONSTANT TSTOP      = 0.1,        XMAG   = 0.8,       RADRPM= 9.55
        CONSTANT ILIM       = 50.0,       POLE1  = 0.0016,    POLE2  = 0.0016
        CONSTANT VBACK      = 0.05,       PULS   = 1.0,       WIDT   = 0.5
        CONSTANT KT         = 0.62,       GAIN   = 3,         PI     = 3.141593
        CONSTANT QUANT      = 1.0,        TSRT   = 1.0,       ENC    = 2000
        CONSTANT SMP1       = 0.00125,    STRT   = 0.0
        X       = XMAG*(1-PULSE(TSRT,PULS,WIDT))
        Y1      = (VELDEM-VELFB)/CONVER
        Y2      = ZOH((GAIN*Y1*ISCALE),0.0,STRT,SMP1)
        Y3      = BOUND(-ILIM,ILIM,Y2)
        Y4      = REALPL(POLE1,Y3,0.0)
        Y5      = KT*REALPL(POLE2,Y4,0.0)
        Y6      = 1/JEQ*INTEG(Y5,0.0)
        YRPM    = Y6*RADRPM
        YTACH   = Y6*VBACK
        THETA   = INTEG(Y6,0.0)
        ENCODE = THETA*CONVER           $"CONVERT TO PULSES FROM RADIANS"
        DEMAND = CONVER*INTEG(X,0.0)
        TERMT(T.GE.TSTOP)
END $"OF DERIVATIVE"
DISCRETE FEED
INTERVAL SMP2=0.001
PROCEDURAL
        AZ1     = ENCODE - AZ0 + AZ3   $"POSITION DIFFERENCE + REMAINDER"
        AZ2     = QNTZR(QUANT,AZ1)     $"QUANTISATION TO ONE PULSE"
        AZ3     = AZ1 - AZ2            $"THE REMAINDER"
        AZ0     = ENCODE               $"RESET LAST VALUE"
        VELFB   = AZ2/SMP2             $"SCALE TO PULSES/SEC"
        BZ1     = DEMAND - BZ0 + BZ3   $"POSITION DIFFERENCE + REMAINDER"
        BZ2     = QNTZR(QUANT,BZ1)     $"QUANTISATION TO ONE PULSE"
        BZ3     = BZ1 - BZ2            $"THE REMAINDER"
        BZ0     = DEMAND               $"RESET LAST VALUE"
        VELDEM = BZ2/SMP2              $"SCALE TO PULSE/SEC"
END $"OF PROCEDURAL"
END $"OF DISCRETE"

END $"OF PROGRAM"
```

Listing of Program 5.2

The simulated velocity loop was optimised with the required load (in this case 0.05kgm^2) to give a fast response, with minimal overshoot in the time domain, and a flat top gain with slow phase roll-off in the frequency domain.

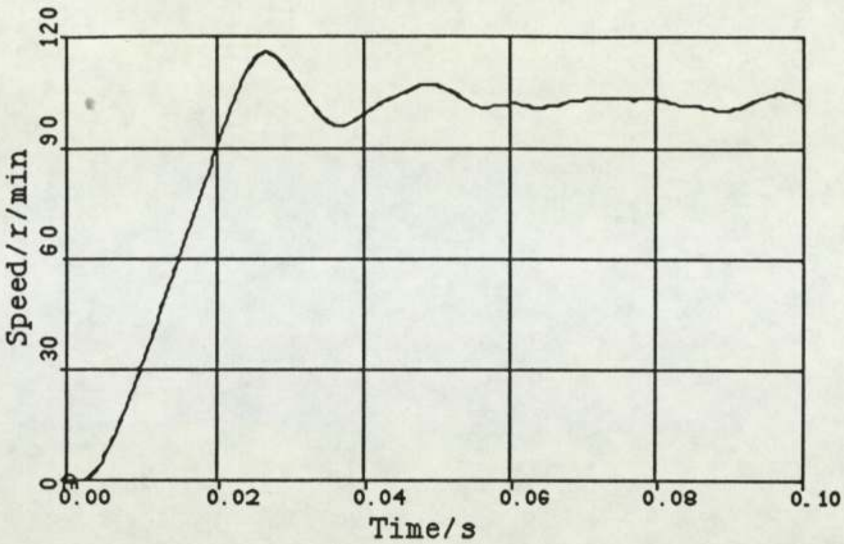


Figure 5.11 The velocity transient response with GAIN = $5V/rads^{-1}$

It was found that the system was unstable with GAIN set above 7, so the GAIN used should be much less than this (Gain margin of 12dB relates to GAIN=1.8). Figures (5.11)-(5.14) show the effect of forward loop gain increase on transient responses; the step magnitude was chosen to ensure that the drive was always in its linear region of the current loop, so that conventional linear stability could be analysed. Figure (5.11) shows the unacceptable settling time with a gain of 5 $V/rads^{-1}$, whilst figure (5.14) shows a slightly sluggish response with a gain of 1 $V/rads^{-1}$, and figure (5.12) shows a close to optimum response with a forward loop additional gain of 3 $V/rads^{-1}$. This is an approximate method so exact values are not necessary. Figures (5.15) and (5.16) show the velocity closed loop frequency response with the gain set at 2 $V/rads^{-1}$ and 3 $V/rads^{-1}$ respectively. With a gain of 3 $V/rads^{-1}$ there is a resonant peak in the gain characteristic which is undesirable; the peak is not seen with a gain of $2V/rads^{-1}$.

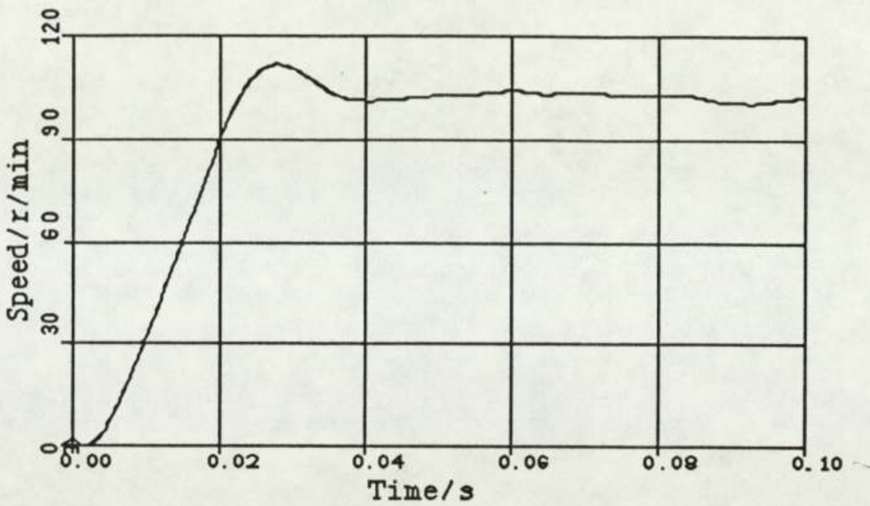


Figure 5.12 The velocity transient response with GAIN = $3V/rads^{-1}$

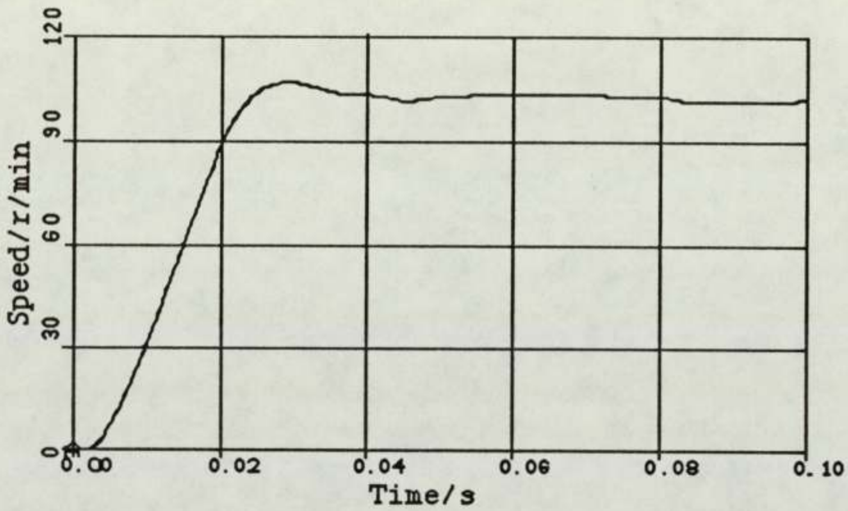


Figure 5.13 The velocity transient response with GAIN = $2V/rads^{-1}$

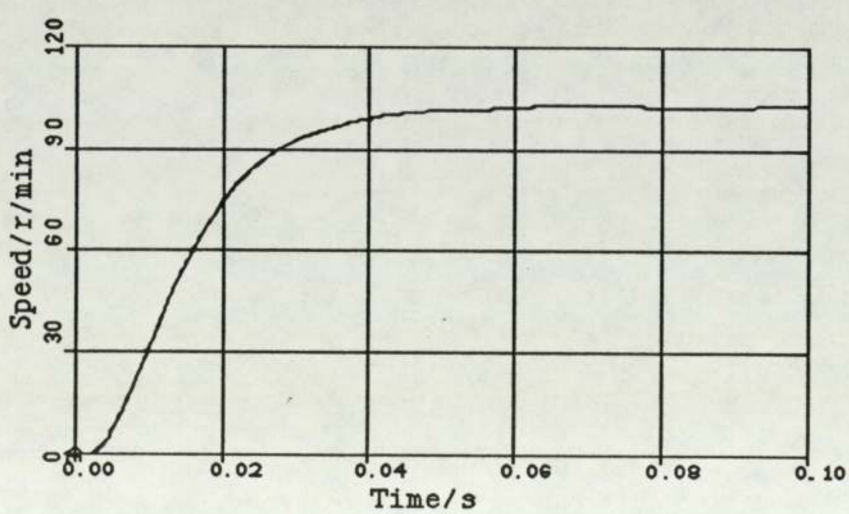


Figure 5.14 The velocity transient response with GAIN = $1V/rads^{-1}$

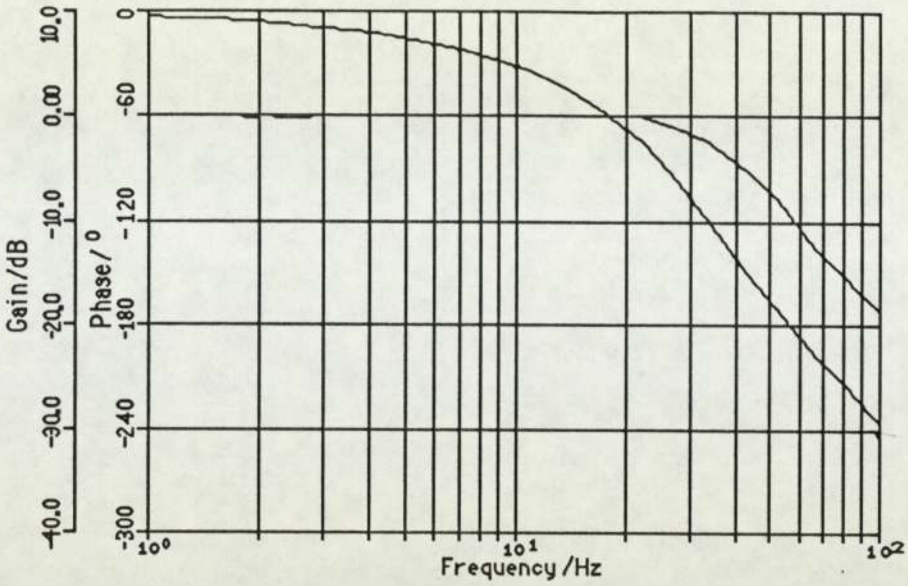


Figure 5.15 The velocity loop frequency response with GAIN = 2V/rads^{-1}

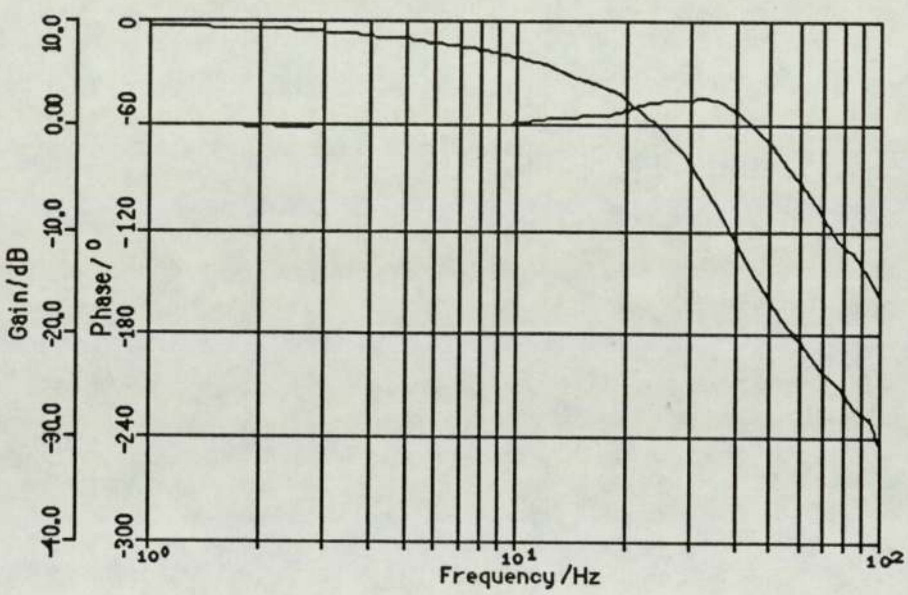


Figure 5.16 The velocity loop frequency response with GAIN = 3V/rads^{-1}

With reference to all the above diagrams the conclusion was that the maximum figure used for gain should be 2 V/rads^{-1} which gives a flat top frequency response, with a bandwidth of 35Hz at -125°. For this particular case an early experimental result using this system gave the maximum achievable bandwidth as 38Hz at -140°, which shows the relevance of this procedure, as the figures correlate well.(This result was taken using the Molins' Dservo system prior to by-passing the open loop phase shifts as explained in chapter 7.0).

5.4.7 AN OPTIMUM POSITION LOOP

Once the velocity loop has been optimised the position loop may be closed as shown in figure (5.17), with proportional error gain and unity velocity feedforward (other than unity velocity feedforward will increase errors for this form of application).

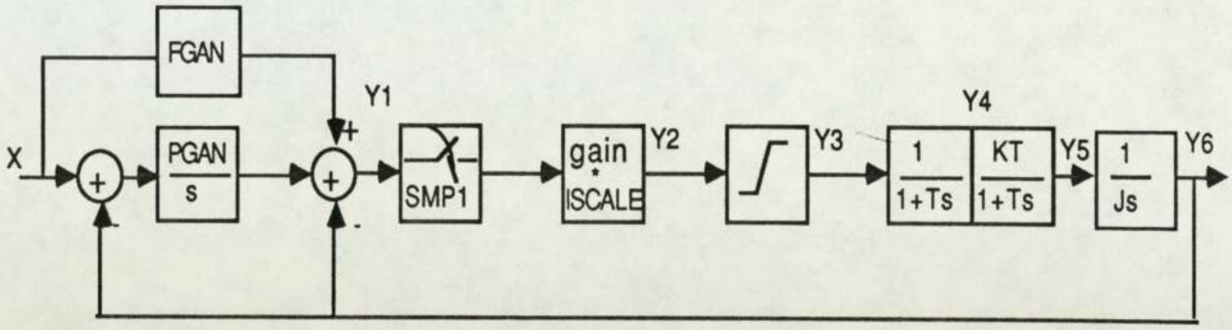


Figure 5.17 A position loop block diagram

Program listing (5.3) simulates the model of figure (5.17), being driven by the actual demanded profile of a 2500 r/min ramp over 10 seconds.

The position loop gain can be gradually increased to monitor the error between demanded and actual position over the profile. With PGAN=0 as in figure (5.18) the error is much too large as would be expected. Figure (5.19), and (5.20) have PGAN set to $100s^{-1}$ and $150s^{-1}$ with maximum errors of 0.0038 rads and 0.0030 rads respectively, therefore a position loop gain of $150s^{-1}$ will be sufficient for this application.

If a series of drives are similar then the current loop characteristics may be used for the other drives in the series by simply changing the current limit and torque constants. This is the case for the Electrocraft Bru-500 range of brushless drives.

The position loop was optimised by increasing the gain until the following errors were within specification, whilst ensuring an adequately stable response.

```
PROGRAM POSITION_LOOP-SIMULATION
        "COPYRIGHT D.R. SEAWARD 1988"
        "SIMULATION FOR THE BRU500 SERIES OF BRUSHLESS MACHINES"
        CINTERVAL      CINT = 0.001
        MAXTERVAL      MAXT = 5E-5
        NSTEPS              NSTP = 1
        ALGORITHM      IALG = 5
INITIAL
JEQ   = JLOAD + JMOT
AZ0   = 0
AZ1   = 0
AZ2   = 0
AZ3   = 0
BZ0   = 0
BZ1   = 0
BZ2   = 0
BZ3   = 0
CONVER  = 4*ENC/2/PI
END $"OF INITIAL"

DERIVATIVE
        CONSTANT  JMOT    = 0.0014,    JLOAD = 0.05,      ISCALE  = 5.0
        CONSTANT  TSTOP   = 6.0,       XMAG  = 47.1,      RADRPM = 9.55
        CONSTANT  ILIM    = 50.0,      POLE1 = 0.0016,    POLE2   = 0.0016
        CONSTANT  VBACK   = 0.05,      ENC   = 2000
        CONSTANT  SMP1    = 0.00125,   STRT  = 0.0
        CONSTANT  TRMP    = 0.1,       STRMP = 10.1,      FGAN  = 1
        CONSTANT  PGAN    = 0.0,       QUANT = 1
        CONSTANT  KT      = 0.62,      GAIN  = 2,         PI    = 3.14159
PROCEDURAL(X=XMAG)
 IF(T.GE.STRMP)GOTO JUMP1
  X      = XMAG*RAMP(TRMP)
 JUMP1..CONTINUE
    END$"OF PROCEDURAL"
    Y1     = (VELDEM*FGAN-VELFB+PGAN*PERROR)/CONVER
    Y2     = ZOH((GAIN*Y1*ISCALE),0.0,STRT,SMP1)
    Y3     = BOUND(-ILIM,ILIM,Y2)
    Y4     = REALPL(POLE1,Y3,0.0)
    Y5     = KT*REALPL(POLE2,Y4,0.0)
    Y6     = 1/JEQ*INTEG(Y5,0.0)
    YRPM   = Y6*RADRPM
    YTACH  = Y6*VBACK
    THETA  = INTEG(Y6,0.0)
    ENCODE = THETA*CONVER $"CONVERT TO PULSES FROM RADIANS"
    DEMAND= CONVER*INTEG(X,0.0)
    ERROR  = DEMAND/CONVER - THETA
    TERMT(T.GE.TSTOP)
    END $"OF DERIVATIVE"
    DISCRETE FEED
  INTERVAL SMP2=0.001
 PROCEDURAL
    AZ1    = ENCODE - AZ0 + AZ3  $"POSITION DIFFERENCE + REMAINDER"
    AZ2    = QNTZR(QUANT,AZ1)    $"QUANTISATION TO ONE PULSE"
    AZ3    = AZ1 - AZ2           $"THE REMAINDER"
    AZ0    = ENCODE              $"RESET LAST VALUE"
    VELFB  = AZ2/SMP2            $"SCALE TO PULSES/SEC"
    BZ1    = DEMAND - BZ0 + BZ3  $"POSITION DIFFERENCE + REMAINDER"
    BZ2    = QNTZR(QUANT,BZ1)    $"QUANTISATION TO ONE PULSE"
    BZ3    = BZ1 - BZ2           $"THE REMAINDER"
    BZ0    = DEMAND              $"RESET LAST VALUE"
    VELDEM= BZ2/SMP2             $"SCALE TO PULSE/SEC"
    PERROR= QNTZR(QUANT,DEMAND-ENCODE)
 END $"OF PROCEDURAL"
    END $"OF DISCRETE"

    END $"OF PROGRAM"
```

Program listing 5.3

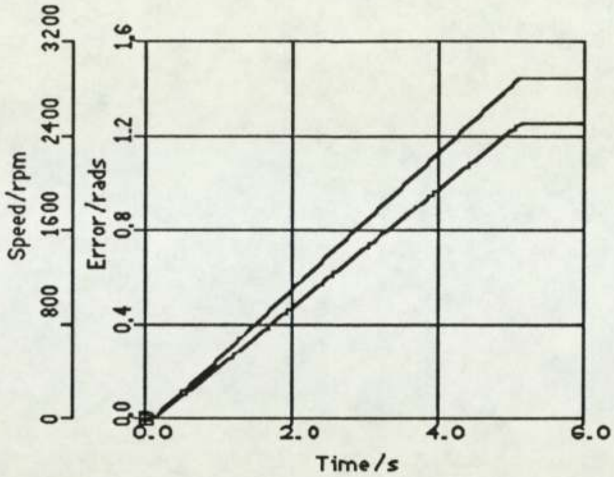


Figure 5.18 The position loop time response with PGAN = 0

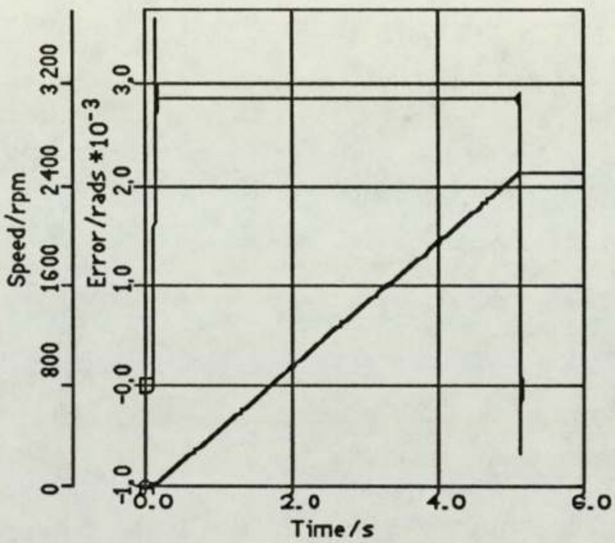


Figure 5.19 The position loop time response with PGAN = 100

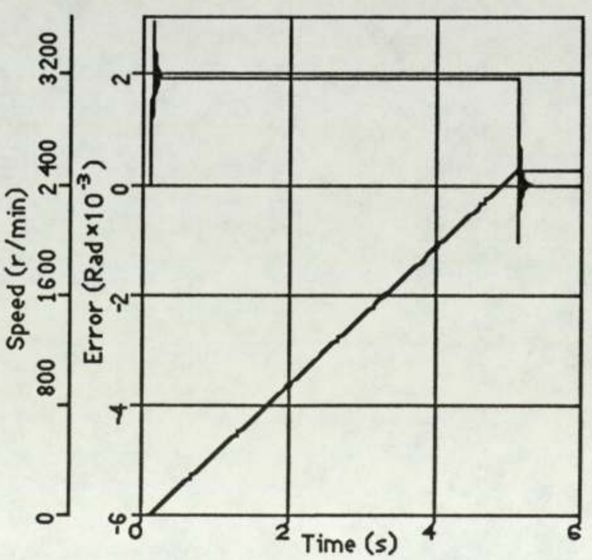


Figure 5.20 The position loop time response with PGAN = 150

5.5 CONCLUSIONS

It has been shown how to achieve a figure of attainable velocity loop bandwidth and position loop gain in terms of the digital sampling frequency for a particular class of servo drives, which have a current loop gain, inertial loads and digital sampling. The maximum attainable velocity loop bandwidth frequency is 1/4 of the sampling frequency, a stable limit is 1/10, and the limit to enable position control is approximately 1/20; the maximum position loop gain that can be used is approximately 0.03 times the sample frequency (rads^{-1}).

The method was expanded to take account of practical systems by use of the velocity open-loop frequency response and simulation. The model approximates to the actual system such that predictions can be made about suitability to achieve a specification. For the system chosen the prediction was for a 35Hz (@-125°) velocity loop bandwidth, whilst the system could achieve 38Hz (@-140°) hence demonstrating good correlation (cf. K=20 gives a 50Hz bandwidth from section (5.3.2)). The prediction for a position loop gain of 150s^{-1} to meet the specification was not checked in practise, although it was close to the figure of 200s^{-1} which was the predicted maximum from section (5.3.3). It is therefore assumed that this system would probably not meet the specification.

CHAPTER 6

6.0 FURTHER LIMITS ON SERVO PERFORMANCE

6.1 INTRODUCTION

This chapter is designed to complement chapter 5.0 by presenting further limits to servo performance. Limits that have been identified are due to a combination of quantisation, current limit, mechanical backlash, mechanical torsional compliance and resonance.

6.2.0 THE EFFECT OF QUANTISATION ON PERFORMANCE

6.2.1 INTRODUCTION

Digital feedback in a servo drive can reduce performance due to sampling delays and quantisation. Section (5.3) analysed sampling delays, but the effects of quantisation are difficult to assess. As the gain of a system increases its output (velocity) will begin to follow the quantisation steps on the input due to the digital feedback. Eventually the open loop gain will be at a level such that the system would be expected to be stable by conventional control theory analysis, but the controller will shut down due to overcurrent. At this point it has been found that the the motor output shaft jitters and emits unacceptable levels of acoustic noise. This phenomenon has been found to be due to two separate reasons: firstly the current loop shut-down circuitry of the drive responds to the rapid fluctuations of current demand, and detects a high rms current: secondly the fluctuations in current and therefore torque demand are capable of exciting mechanical resonances in the system, causing a high rms current demand. The effect is analogous to problems experienced with early servo designs where system output followed tachogenerator ripple of an analogue feedback system as loop gain was increased.

Quantisation error used to be mainly due to the analogue to digital, or digital to analogue conversion, but 12 bit devices are now common place. The largest error now comes from the digital feedback in the majority of cases. High count encoders are available but they are speed limited; at present devices are limited to approximately 600 kHz, that is, a 3000r/min device is limited to approximately 12,000 pulses per revolution. The quantisation level can cause sustained oscillations in high gain systems, where the set point is between two quantisation levels and the system "jitters" between the two. This situation is most noticeable when trying to hold the system at a constant position (zero velocity), because the system is effectively open loop between two encoder pulses.

6.2.2 VELOCITY LOOP BANDWIDTH DERIVATION FROM DIGITAL QUANTISATION

A block diagram of a typical sampled servo system was shown in figure (5.2). It is a first order system, described by equations (5.1) and (5.2).

With a large gain the system behaves similarly to a pulse width modulating system, since the signal proportional to velocity error fed to the current loop can only be one of a small number of states, the worst case being where there are only three states; the current in the drive is fully on forward, off, or fully reversed. If the bandwidth of the system is less than one tenth approximately, of the digital controllers sampling frequency then the system can control velocity with sufficient stability (section (5.3)).

The maximum bandwidth that can be achieved is when one encoder pulse of error in the sample period creates a demand for the peak current capability of the drive. Increasing the gain further would force the system into its input circuit's cut-off region, and the output would still be the peak current. With reference to figure (5.2) when one pulse equates to the peak current rating, G is seen to be the peak current rating. Also the peak current rating multiplied by the torque constant is the peak torque rating of the machine. Thus if neglecting sampling delays the maximum bandwidth of the system is:-

$$\omega_b = \frac{T_p.\Delta T.p}{2\pi J_T} \quad \text{rads}^{-1} \qquad \ldots.(6.1)$$

The maximum bandwidth obtainable from a system may be less than this for other reasons, but fundamentally the bandwidth may not be greater than this. If a drive is in its intermittent current zone for any length of time then it must shut down to prevent damage to either the motor windings or the controller power electronics. From the previous discussion it can be seen that if one encoder pulse error creates a current demand which is higher than the continuous rating, then whilst the drive may be stable, it will shut down due to over-current if the current sensing circuitry is capable of responding to current oscillations at the sampling frequency. This philosophy leads to another theoretical maximum bandwidth achievable for a servo drive system:-

$$\omega_b = \frac{T_{cont}.\Delta T.p}{2\pi J_T} \text{ rads}^{-1} \qquad \text{....(6.2)}$$

The drives that have been investigated (Electrocraft Bru-500 range) have band limited current sensing circuitry, which resembles a first order pole (time constant τ_c). The maximum bandwidth due to continuous current rating can therefore be increased to take account of this fact. The maximum bandwidth must be multiplied by the first order pole attenuation at the sampling frequency, that is:-

$$\omega_b = \frac{T_{cont}.\Delta T.p.\sqrt{(1+(\tau_c/\Delta T)^2)}}{2\pi J_T} \quad \text{rads}^{-1} \qquad \text{....(6.3)}$$

6.2.3 POSITION LOOP GAIN LIMITATION DUE TO QUANTISATION

A similar expression for position loop gain can be derived, when one pulse of position error causes limited current. When one pulse of error equates to limited current (I_L is either I_p or I_{cont}), from figure (5.5):-

$$1.\frac{2\pi}{p}.A.\frac{p.\Delta T.G}{2\pi} = I_L \qquad \text{....(6.4)}$$

but from section (6.2.2) $G = I_L$, therefore,

$$A.\Delta T = 1 \qquad \text{....(6.5)}$$

Thus the position loop gain must be less than the sampling frequency if the velocity loop gain is also a maximum.

6.3 A DESIGN EXERCISE

Section (5.2) suggested that in order to meet the specification of section (1.5) a position loop gain of $200s^{-1}$ and a velocity loop bandwidth approaching 100Hz were required. The inertial loads to be considered from section (1.5) were the ledger with $0.01kgm^2$ and the cut-off with an approximate inertia of $0.25kgm^2$. The maximum acceleration required by the load was 26 $rads^{-2}$ (10s ramp), which relates to maximum powers of 70 W and 1.7kW for the two respective loads, at the top speed of 2500r/min.

The Electrocraft Bru-500 drives were selected as having the highest performance in the market place, chapter 3.0. All the servo drives in this range incorporated a factory fitted encoder of 2000 quadrature lines per revolution, which translates to 8000 pulses per revolution. The microprocessor controller sampled the encoder pulse count every millisecond. Thus the finest grain of velocity information is 7.5 r/min and the maximum position loop gain is 1000, equation (6.5), which is larger than that required.

The performance of each of the drives in the range capable of the 2500r/min is listed in table (6.1) with predicted maximum bandwidths calculated for the $0.01kgm^2$ load using equations (6.1) and (6.2).

Motor	Tcont /Nm	Tpeak /Nm	Power/W	Continuous Bandwidth/Hz	Peak Bandwidth/Hz
S-3005	0.79	3.1	414	16	63
S-3016	2.2	7.0	1152	44	140
S-4030	3.5	12.4	1466	71	251
S-4050	6.1	24.7	2555	123	498
S-6100	11.3	31.0	3550	229	628
S-6200	22.6	62.0	7100	458	1256

Table 6.1 Performance limits for a series of drives with a $0.01kgm^2$ inertial load

The theoretical bandwidths in excess of 100Hz are not feasible, section (5.3). No account of current sensing circuit filtering has been made so the actual bandwidth limit will lie between the two figures quoted. The S-3005 is unsuitable even though it has six times the required power for the application, as is the S-3016 with sixteen times the power. The S-4030 appears to be the marginally suitable for the 0.01kgm^2 load. It should be noted that prior to this analysis that the S-3016 proved experimentally to be unsuitable to this task, as did the S-4030 due to resonance effects (see chapter 8.0).

When considering the 0.25kgm^2 load the predicted bandwidth will be 0.25/0.01 or 25 times smaller than those listed. Thus even the S-6200 drive would be incapable of meeting the requirement in terms of absolute following errors since predicted maximum bandwidth is 18Hz.

Also note that whilst section (5.3) suggested a reduction in sampling time to increase performance this analysis suggests the opposite. Thus when selecting a drive system a compromise must be made whereby the limit due to sampling and due to quantisation occur at the same gain.

6.4.0 THE EFFECT OF THE LOAD ON SERVO PERFORMANCE

6.4.1 MECHANICAL TRANSMISSIONS IN GENERAL

The load actuator and the motor are very rarely designed as a single unit. Thus some form of coupling is required between the motor rotor and the actuator. Often the coupling incorporates some form of gearing. Geared couplings may be in the form of simple gear boxes, worm gears or belt systems.

Alternatively the load may be directly coupled to the motor by a key-way, splined shaft or taper coupling. There are many mechanical coupling systems to select from but these can be compared simply as geared or direct drive systems.

A direct drive servo can provide the motive power for each axis without using gears, belts or any other form of mechanical linkage. Apart from eliminating the mechanical transmission system, there are other gains to be made, but consideration of the trade-offs involved, must be made. These are discussed by Powell [58].

Traditional gear or belt driven servo-mechanisms are constrained by the limitations of mechanical speed ratioing devices, which introduce backlash, cogging, compliance, friction, inertia multiplication and wear; machines with mechanical transmissions can lose accuracy, repeatability and efficiency, as a result backlash and compliance in mechanical systems can also introduce undesirable resonances which effect performance if the resonant frequencies are within the servo bandwidth.

With the load connected directly to the motor shaft it is easy to contrast a zero backlash system with high mechanical stiffness, reduced inertia and extremely low friction. Since the direct drive system is simpler mechanically, it is more reliable and costs less to implement than a comparable high precision geared servo mechanism.

Backlash in gearing translates into a time delay which enters the control loop as a phase shift analogous to a sampling period, reducing stability and possibly leading to underdamped oscillations, which can often be seen as a limit cycle. Anti-backlash gearing can be implemented to relieve this problem, but it is not a cure and can be costly.

Designers of traditionally geared servo system were faced with a difficult choice regarding placement of the feedback sensors. With backlash separating the motor and load, placing the feedback transducer on the load side can cause oscillation by adding non-linearities and phase delay to the feedback signal. However, placing the sensor on the motor side of the gearbox backlash and compliance, will cause uncompensated open-loop position error. Usually the designer chooses to place the sensor on the motor side, since an inaccurate servo loop is less noticeable than an unstable one. Servo systems that cannot tolerate the position error or the stability problem require sensors on both sides of the gearset and some form of feedback hierarchy to mediate between the two conflicting feedback signals.

The use of a direct drive solves the above problem since the sensor is effectively coupled directly to both the load and the motor. Care must be taken to ensure a good mechanical connection between the load and motor shaft, as poorly fitting keyways can cause a backlash effect, whilst a motor shaft that is too thin will introduce too much compliance, which is often self evident by a "bell" resonance.

Thus, by virtue of direct connection to the load, direct drive systems excel in terms of acceleration, efficiency and power rate, while eliminating the problems of deadband, backlash, and compliance inherent in motor systems using mechanical gear systems. Therefore direct drive servo loops are much more stable than their mechanically linked counterparts.

Of the systems discussed some are better suited to high performance servos. If gearing is required then belt systems have proved better than conventional gears as they have zero backlash although they are compliant causing resonance problems. Direct drive couplings may also exhibit backlash, in particular simple keyways are poor, and also to a lesser extent splined couplings. The best form of coupling found is a taper lock fitted with a keyway, although these are not practical if exact alignment between motor shaft and load is required as the centres "float" upon tightening the taper.

6.4.2 THE EFFECT OF BACKLASH

It is difficult to analyse the effect of backlash in a servo system.

If the motor inertia J_M, and the load inertia J_L are lumped together to produce the total system inertia, J_T, then the techniques discussed in sections (6.3) can be used to determine the maximum system gains and performance. The effect of backlash is to momentarily decouple the load inertia from the motor, and during this period the system may be unstable as the effective open loop gain increases, once coupled the system will again be stable. The effect is more pronounced when the ratio of motor to load inertia is large and in this instance the system will vigorously "jitter" within the backlash zone. This can damage gearing, cause overcurrents and the system will emit high acoustic noise levels.

To enable satisfactory operation in the presence of backlash the system must be stable with only the motor inertia and input gearing, J_M.

Using the single pole model of equation (5.1) the system bandwidth is:-

$\omega_{bm} = 1/2\pi\tau_{bm}$ where $\tau_{bm} = 2\pi J_M/\Delta T.p.G.K_t$ taking the highest stable value for G from previous analysis.

Adding the load inertia with the same gain gives a new value for bandwidth:-

$$\omega_{bt} = 1/2\pi\tau_{bt} \qquad ....(6.6)$$

where $\tau_{bt} = 2\pi(J_M+J_L)/\Delta T.p.G.K_t$ thus,

$$\omega_{bt}/\omega_{bm} = J_M/(J_M+J_L) \qquad ....(6.7)$$

which if $J_L >> J_M$ can be approximated to

$$\omega_{bt}/\omega_{bm} \approx J_M/J_L \qquad ....(6.8)$$

Thus the achievable bandwidth is degraded by the ratio of the inertias.

This can be a serious problem for modern systems since the trend is for reduced motor inertia.

When considering the initial Molins' specification it was hoped to control the cut-off to a very high accuracy. To a first approximation inertia was $0.25kgm^2$, but it could be as high as $0.5kgm^2$ and it would probably have to be driven through some form of gearing because of its complex configuration and the machine geometry involved. The Bru-500 range of drives represent some of the highest performance drives available, and the S-6200 drive is the largest that could drive the cut-off at the required speed: its inertia is $0.0024kgm^2$. Thus the ratio of load to motor inertia is 104:1 (208:1 at worst) and section (5.3) placed the limit on bandwidth (without the load) at 100Hz due to the 1kHz sampling. Thus the maximum velocity loop bandwidth achievable if backlash is included is 0.95Hz (0.48 Hz at worst). Also for the $0.01kgm^2$ ledger load the maximum velocity loop bandwidth with backlash and the S-6200 motor would be 19Hz. This is obviously unsatisfactory leading to the conclusion that either direct drives must be employed or the original specification was not feasible.

One method to combat this problem is to add inertia to the motor side of the backlash but this will ultimately reduce acceleration and system performance.

6.4.3 MECHANICAL TORSIONAL RESONANCE

A further reason for drive shut down on overcurrent was due to the fluctuation in current and therefore torque demand due to mechanical oscillations often classed as "bell-resonance". The mechanism by which this occurs is that the quantisation ripple seen by the digital controller excites mechanical resonances.

Figure (6.1) shows a typical load block diagram. The viscous damping of the motor has been neglected to ease analysis, and in most practical situations the viscous damping of the load will be much higher than that seen at the motor. Backlash has been shown in the block diagram but this is difficult to include in any analysis, so it is assumed that the load and motor inertias are coupled directly via a torsional spring (the motor shaft).

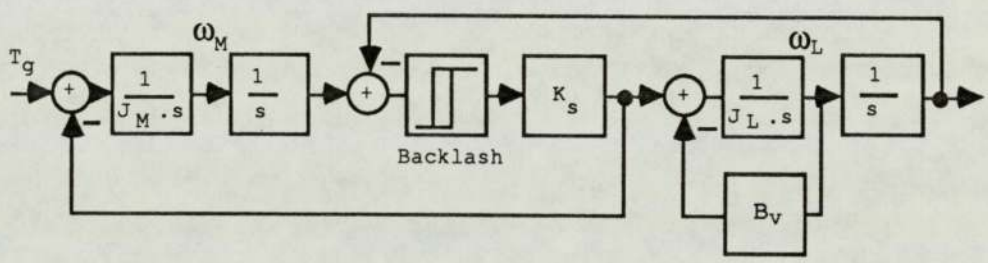


Figure 6.1 A typical load Block Diagram

The transfer function relating input torque to load velocity is:-

$$\frac{\omega_L}{T_g} = \frac{1}{(J_M.J_L/K_s)s^3 + (B_v.J_M/K_s)s^2 + (J_M+J_L)s + B_v} \quad ....(6.9)$$

which assuming B_v is small and K_s is large, can be approximated to:-

$$\frac{\omega_L}{T_g} = \frac{1}{[B_v+(J_M+J_L)s][(J_M.J_L/\{K_s(J_M+J_M)\})s^2+(B_v.J_M^2/\{K_s(J_M+J_L)^2\})s+1]} \quad ....(6.10)$$

This is the transfer function expected for a system with an infinitely stiff connection between the two inertias divided by a second order system. For many cases B_v is very small indeed which infers that the damping constant of the second order system will also be very small, which is undesirable as the system will exhibit a pronounced resonance at $\sqrt{K_s(J_M+J_L)/J_M.J_L}$ rads^{-1}. This situation is not an accurate representation of the majority of control systems, as the feedback transducer is normally placed upon the motor shaft directly and not upon the load as the above assumes. The relationship between load and motor velocities is:-

$$\omega_L = \frac{\omega_M.K_s}{J_L.s^2 + B_v s + K_s} \qquad \text{....(6.11)}$$

And therefore the approximate transfer function becomes:-

$$\frac{\omega_M}{T_g} = \frac{\{J_L/K_s\}s^2 + \{B_v/K_s\}s + 1}{[B_v+(J_M+J_L)s][(J_M.J_L/\{K_s(J_M+J_L)\})s^2+(B_v.J_M^2/\{K_s(J_M+J_L)^2\})s+1]} \qquad \text{....(6.12)}$$

Thus the system has been multiplied by another second order system in the numerator with another resonant frequency which is $\sqrt{\{J_M/(J_M+J_L)\}}$ times lower than the previous one, at $\sqrt{(K_s/J_L)}$ rads^{-1}. At the resonant frequency of the numerator the system has zero gain, whilst that due to the denominator is infinite. If $J_M \gg J_L$ then $(J_L + J_M)/J_M J_L$ tends towards $1/J_L$ and the effects cancel. Therefore if the load/ motor inertia ratio is small (less than unity) no resonance problems due to this phenomenon will occur. However, if $J_L \gg J_M$ then the system has a numerator resonances at $\sqrt{(K_s/J_L)}$ whilst the denominator resonances at approximately $\sqrt{(K_s/J_M)}$, which do not cancel. Infinite gain causes instability so it must be ensured that $\sqrt{(K_s/J_M)}$ is at a sufficiently high frequency that other system poles have equally high attenuation.

The value of K_s for a direct connection will depend upon the motor used. A value for K_s can be derived for a simple system as $K_s = \pi d^4 G_m/32L_s$ where d is the shaft diameter, L_s the active shaft length and G_m is $8 \times 10^{10} Nm^{-2}$ for steel (the common shaft material). The main problem in determining K_s, is that a value for the active length, L_s, is difficult to obtain. For a given series of motors of the same diameter, though, L_s will be approximately proportional to motor inertia, and d to the square of motor inertia ($J=Md^2/8$, where M is the rotor mass), so that K_s is approximately proportional to inertia. The achievable bandwidth will be dependent upon the value of K_s, the larger the value of K_s the higher the achievable bandwidth, and therefore the limit to velocity loop bandwidth due to torsional resonance will be approximately proportional to motor rotor inertia, for a given load.

In the situations described above it may be possible to expand the bandwidth by use of low-pass filtering, which would smooth the current loop demand signal. The filtering must be carefully selected such that disturbances at the sampling frequency are suitably attenuated to stop any excitation of mechanical resonances or so that the current loop safety circuitry does not react to the demand signal unfavourably. The filtering must also be such that the associated phase lag is not great enough to effect stability. One common method used in the past to overcome this problem was to use "notch" filters to negate the resonance effects, Tal [68] and Meshkat [48].

6.5 CONCLUSION

Further performance limits have been identified in high performance servo systems. The limits have been linked to quantisation in the digital controller which is dominated by encoder feedback, and the torsional mechanics of the system.

One factor that may be analysed simply is limits on velocity loop bandwidth and position loop gain due to the combination of drive torque limit and encoder quantisation. It may be shown that the velocity loop bandwidth is proportional to the torque limit, the sampling period and the number of encoder pulses per revolution and inversely proportional to system inertia, equation (6.1). The position loop gain is limited to the sampling frequency.

The types of load coupling available have been discussed, concluding that a direct coupling (using taper locks) is the most applicable. The effect of backlash has been shown to degrade achievable bandwidth approximately in the ratios of the motor to load inertias. Quantisation is also a factor in the mechanical resonance phenomenon which has proved to be complex to analyse as it is dependent upon many differing factors.

It has been shown how the original retro-fit specification (section(1.5)) was impossible to achieve with present drive technology, but calculations have been presented to identify the approximate performance bounds for a given system. For a cut-off drive with an approximate inertia of 0.25kgm^2 and the highest powered of the selected drives (Bru-500 range), with backlash, the maximum velocity loop bandwidth obtainable would be approximately 1Hz, and without it, would only be 18Hz; both figures are unacceptable.

CHAPTER 7

7.0 A SIMULATION AND DESIGN EXERCISE ON A BRUSHLESS AC DRIVE SYSTEM

7.1 INTRODUCTION

Except for chapter 3.0 the discussion has centred upon theoretical analysis of high performance servo systems; analysis being built up from practical experience gained from actual servo systems. In parallel to the theoretical work practical work was undertaken to try to achieve the specification of section (1.5).

Early in the project two brushless dc systems were loaned by Molins to investigate servo system responses. The drive systems had been unsuccessfully tested for fitting to a rod-making machine by Molins. Experiments upon the equipment included frequency and step responses which were compared to simulation results from a model that was derived. The drive systems were very unreliable and were returned to their manufacturer. The initial work carried out upon this system is described in appendix 2 and the system was drive 6 of section (3.5).

Using servo drive comparisons supplied by Fenney [27] a brushless ac drive system was identified as having the best performance available in the market place. These were the Electrocraft Bru-500 range of drives which were then only available in prototype form (January 1987). This system was drive 5 of section (3.5); the results presented in chapter 3.0 (taken in November 1988) suggested that there are now a number of higher performance drive systems available. There has therefore been rapid advances in drive system performance in the last few years, due mainly to advances in power semi-conductor technology and magnetic materials.

Two drives from the Bru-500 range were purchased for investigation by Molins at their plant and subsequently two further drives were supplied to Aston University. The selection was based primarily on previous experience from a brushless dc drive system (appendix 2); higher power rated drives were selected than had previously been investigated.

Drive system models were made available by the manufacturer and the models were enhanced as more laboratory information became available. The initial performance was disappointing compared with theoretical predictions from simulation and the reasons for discrepancies were investigated. The drives and controllers were modified to substantially improve performance.

The work can be considered in two concurrent areas: firstly since the drives originally had poorer than expected performance, modifications were made based on simulation results until some fundamental limit was reached (chapters 4.0-6.0): secondly, discrepancies between the model and system were investigated, so that the model could be improved. These two areas were complementary as often the model discrepancy was due to the same reason responsible for the poorer than expected performance.

This chapter outlines both the practical and simulation work carried out upon the drive systems in an attempt to meet the original specification. Laboratory research is described in a chronological order. Chapter 8.0 continues the practical work by presenting the actual following errors achieved from the systems once all the modifications had been carried out.

7.2 THE BRU-500 SYSTEM

The Electrocraft Bru-500 range of brushless drives appeared on the market early in 1987, and were considered to have the highest performance commercially available at that time, due to the large torques available with respect to the low rotor inertia. The BRU-500 system was essentially a brushless dc drive using sinusoidal excitation of permanent magnet Neodymium Boron Iron machines. The system supervision, diagnostics and control were carried out by a 16 bit microprocessor, which enabled system reporting and parameter changes to be accomplished via an RS232 link to a terminal. Many of the system characteristics were controlled by software which could be easily changed.

Within the series there were four drives modules (DM) rated at 25, 50, 100 and 150 amperes (DM-25, DM-50, DM-100 & DM-150 respectively), and 8 different machine sizes available (S-3008, S-3016, S-4030, S-4050, S-6100, S-6200, S-8100 & S-8200). The systems were fed from a nominal 300 Volt dc bus supplied from an Electrocraft power supply module capable of supplying 50 or 150 Amperes to the dc bus (PSM-50 or PSM-150), which was in turn supplied from an industrial 240 V three phase supply.

Pre-production versions of the brushless synchronous Bru-500 drives (S-6100 motor with a DM-50 drive for the cut-off axis and a S-3016 with a DM-25 for the ledger) were purchased from Electrocraft in order to carry out a feasibility study at Molins' Saunderton plant. Aston purchased a S-4030 with a DM-25 and a S-6100 with a DM-50 for investigation (a complementary research programme at Aston purchased a S-4050 with a DM-50 and a S-6200 with a DM-100). Thus a large number of drives were available to the research group.

Numerous loads were used, but all had a taper lock coupling with an inertia of approximately 0.0026kgm^2. The flywheels available were such that the S-3016 could be tested with a total load inertia of 0.01 kgm^2, the S-6100 with 0.0108 kgm^2 and the S-4030 with 0.0113 kgm^2.

The power supply module (PSM) was of a standard rectifier configuration, which had a "crow-bar" system to enable up to 4kW of regenerative motor braking energy to be "dumped" into a resistor bank. The module had limited LED diagnostics. In service they proved very reliable.

The drive module converted the dc bus and demand signals into appropriate three phase waveforms to drive the motor. The microprocessor controller and a three phase pulse width modulation system resided in this module. The microprocessor could control the system in VELOCITY or TORQUE modes, changed from the terminal interface. The 0-10V input to the system was treated as a velocity or current demand in each of these modes respectively. In velocity mode two control algorithms were available; initially only one called Bandwidth/Damping (B/D) control was available which was updated upon a suggestion from the author to Proportional/Integral (P/I) control. All the servo control constants were held in software and could be changed from the terminal interface. Change between algorithms was by interchange of three socket mounted integrated circuits. The terminal received fault diagnostic messages and could monitor "on-the-fly" system parameters, such as, speed, current and demand voltage. The drive amplifier interfaced directly with the encoder mounted on the motor shaft and used this position information to synthesise three phase current waveforms dependent upon rotor position and current demand, either from the internal digital velocity loop (in velocity mode) or externally (in torque mode). For full system integration various interfaces were available: forward and reverse directions could be independently enabled or disabled to prevent overtravel: the system could be disabled by a remote device: the encoder signals could be accessed by a peripheral control system, and the system had a status signal that could be monitored by an external device. The drive module was a very flexible and reliable piece of equipment, with only a small number of failures linked mainly to protection circuitry which was altered.

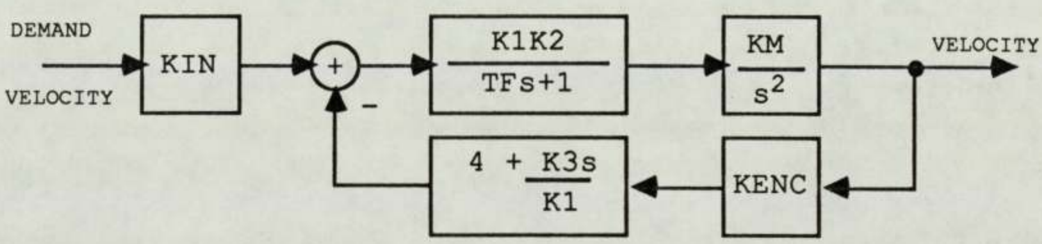
The S-Series servo motors were reliable, with low-maintenance, electronic commutation and thermal protection. The use of high energy Neodymium Iron Boron magnets in a motor design with high heat transfer characteristics resulted in high performance for peak and continuous duty. The prototype versions were unreliable, but no problems were encountered with production models. The integrally fitted encoder had 2000 quadrature lines which related to 8000 pulses per revolution and was a "weak link" in the reliability chain, with a number of failures, due mainly to the laboratory test environment.

The majority of information from the drives was derived experimentally. The evidence was collaborated later by Electrocraft with reference to circuit diagrams of the system, and by detailed technical exchanges with their sales and technical staff. The control algorithms used by the drives were derived from linear block diagrams of the systems provided by the American designers of the Bru-500 controller.

7.3 THE BANDWIDTH DAMPING CONTROL SOFTWARE

The drives were initially delivered with B/D software which controlled velocity. The S-6100 and S-3016 systems were tested at the Molins' Saunderton plant and compared against a simple lumped parameter linear model, figure (7.1), supplied by Electrocraft, which was simulated.

Figure (7.2) shows the simulated (solid lines) and the actual (points) closed-loop response of the Bru-500 with S-6100 motor and DM-50 amplifier as it was delivered, that is, with the variables, bandwidth set to 50, damping to 50, and filter bandwidth to 300. The load was 0.05 kgm^2, which was the initial estimate of the cut-off inertia subsequently altered to the values given in section (1.5). The correlation was very good. A number of these responses were taken with the S-6100 and the S-3016 motors to further substantiate the model.



where,

KIN = 0.2667x(NUMBER OF ENCODER LINES)x(RPM/VOLT)
KENC = $2/\pi$ x (NUMBER OF ENCODER LINES)
K1 = (BANDWIDTH)xG1/((DAMPING)x65536)
K2 = (BANDWIDTH)x(DAMPING)xG2/16777216
K3 = 0.001
KM = DMxKT/(128xJEQ)
DM = DRIVE MODULE (25,50 100 or 150)
KT = MOTOR TORQUE CONSTANT
JEQ = TOTAL INERTIA
TF = 1/(2πxFILTER BANDWIDTH)
Number of encoder lines=2000

Motor	G1	G2
S-3016	2375	2621
S-4030	1200	20972
S-4050	2376	3775
S-6100	1200	8192
S-6200	2145	4876

Figure 7.1 The initial Bru-500 model

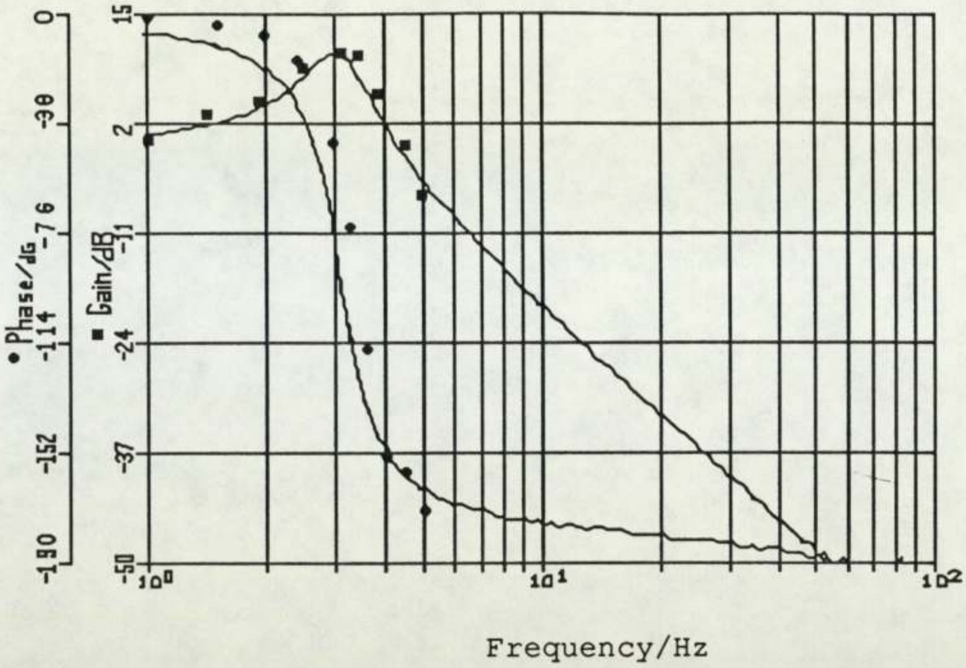


Figure 7.2 The simulated and actual frequency response of the S-6100 drive with bandwidth=50, damping=50, filter bandwidth=300 and a 0.05 kgm^2 load

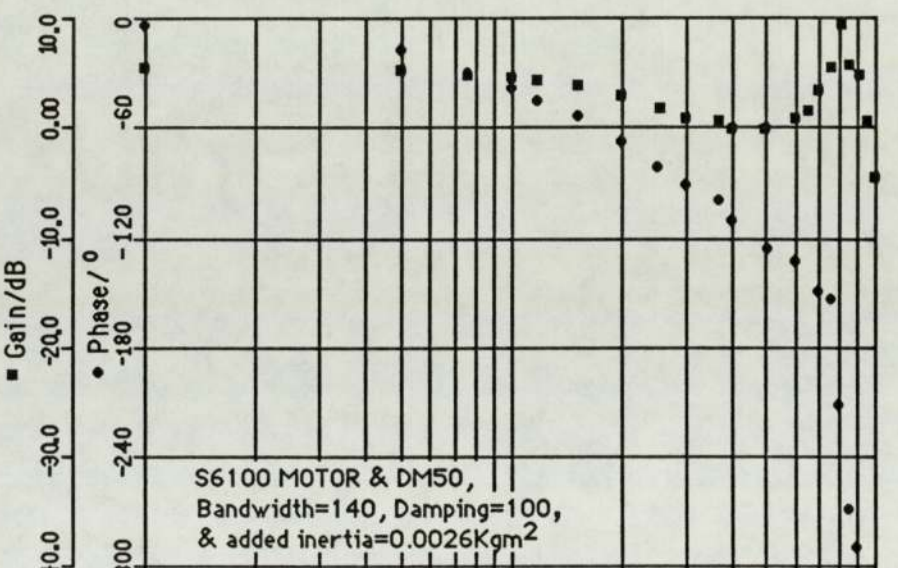


Figure 7.3 The actual frequency response of the S-6100 drive with bandwidth=140, damping=100, filter bandwidth=300 and a 0.0026 kgm^2 load

The system performance was disappointing, however, and the limit to performance at this stage of the research was thought to be much higher. As the gain was increased discrepancies between the simple model and the real system became evident. Also the model did not suggest any resonance or overcurrent problems that the actual system was exhibiting at high gains. The resonance could be seen on certain frequency responses, such as figure (7.3), which was taken with a lower inertia.

The highest practical bandwidth obtainable prior to over-current, is shown in figure (7.4), showing the S-6100 with 0.05 kgm^2 load and the bandwidth set to 300, damping to 200, and the filter bandwidth to 200. The bandwidth had been improved from the original 5Hz to 17Hz.

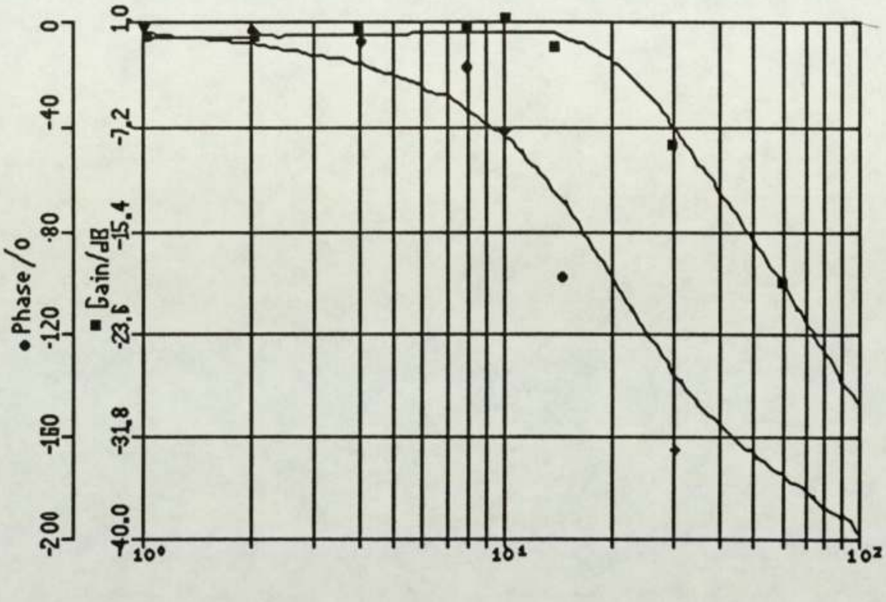


Figure 7.4 The simulated and actual frequency response of the S-6100 drive with bandwidth=300, damping=200, filter bandwidth=200 and a 0.05 kgm^2 load

The model used to generate the Bode plots was perfectly linear and as such could not represent the time domain response of the system. A current limit was included in the model to overcome this problem and it then correlated well for the step time response of the system, but discrepancies still existed at high gains.

To increase the performance it was hoped to remove the forward loop integration and feedback zero in the Electrocraft velocity loop as this would permit a higher bandwidth system with lower phase shifts. By simulating such a proportional loop the velocity servo with the "as-delivered" system gains had a 70Hz velocity loop bandwidth, figure (7.5). The point to note was the phase roll-off with frequency which was much less steep than for the B/D software case. For example, at the bandwidth the highest actual gain B/D software algorithm had -100° of phase shift at 17Hz, the "as-delivered" B/D algorithm had -165° at 5Hz, whilst the simulated proportional system had -58° at 70Hz.

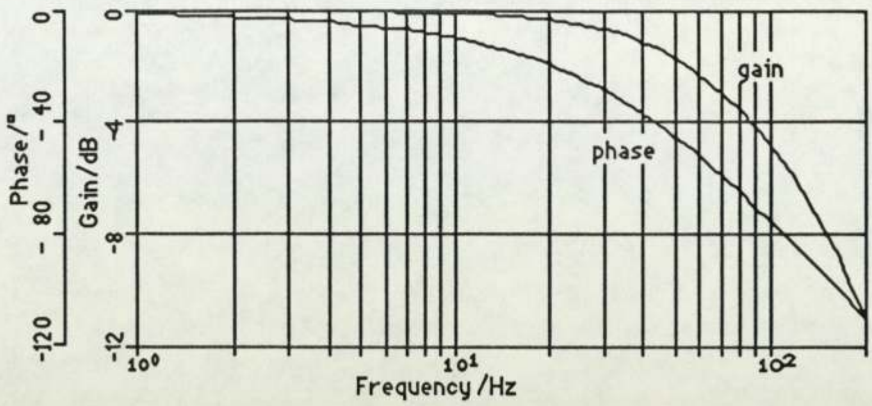


Figure 7.5 The simulated frequency response of the S-6100 drive with forward loop integration removed bandwidth=50, damping=50, filter bandwidth=300, and a 0.05 kgm^2 load

The Bru-500 system had proved inadequate in velocity mode. It was suggested that Electrocraft's software be changed to use pure proportional gain on velocity error and no acceleration feedback. Electrocraft could not guarantee delivery of such software so Molins developed an independent controller to be used with the drive in torque mode. In torque mode a voltage input to the system was considered a current demand.

7.4 THE DSERVO CONTROLLER

Considerable effort was taken to construct an external velocity loop. Initially one was constructed using a frequency to voltage converter as the pulse tachogenerator converting pulses from the encoder into a voltage proportional to velocity. This approach was abandoned as the tachogenerator exhibited poor low velocity characteristics, and it had a limited velocity range above which its output saturated. These problems could have been overcome but this tachogenerator required an analogue summing of input to feedback velocity, and it was hoped to use a digital system to increase accuracy.

A digital tachogenerator was therefore designed using a "differential tachogenerator". If the demand signal is a pulse train where one pulse is a demand for the servo to move by one encoder quadrature pulse, then the velocity demand is the pulse train frequency. If each demand pulse counts a digital counter up by one and every encoder pulse down by one, then if reset at the beginning of every sample period the end count is proportional to the velocity error which may be inputted directly into the current loop via amplification and a digital to analogue converter. In a conventional velocity servo there are errors due to inaccuracy of the tachogenerator but this system reduces these errors as both the forward and feedback paths are perfectly matched using the same electronic components. The actual electronic hardware is more complex than described as synchronised logic was required to ensure that the counter did not try to count up and down at the same time if an encoder and demand pulse were concurrent. The synchroniser ensured that the up count, down count, load into counter and write from counter could never occur simultaneously (similar circuitry is described in appendix 6). In the first prototype the additional loop gain was provided by operational amplifier circuits as were filter networks designed to reduce output ripple due to sampling and quantisation error; an offset potentiometer was used to run the servo at low velocity and to investigate the effects of a possible current feedforward network by offsetting current required to drive the viscous frictive load. The facility was also made to sum an external position error signal to the velocity error signal.

The circuit was built and commissioned on the Bru-500. Initial results were favourable so the system was implemented within a microcontroller system, classed the Molins' Dservo (for digital servo).

The Dservo system was designed, programmed and tested by Molins' personnel. It used an 8751 eight bit microcontroller (which was similar to the Intel 8085) to control both position and velocity with velocity feedforward, as described in section (4.2.5). The system required a series of pulses as its demand and forced the motor to revolve by an appropriate number of encoder pulses, by providing a current demand signal to the Bru-500. The demand also had a direction signal that was decoded. Essentially the demand and encoder feedback were buffered by a synchronising circuit which produced three outputs which were available to the microprocessor: the demand count, the number of encoder clockwise pulses and the number of encoder anticlockwise pulses. The system was also interrupted by an external clock which initiated the control loop, and took approximately 200μs to complete. The external clock could therefore approach this figure, thus defining the sampling period; in this way many systems could be run from a common clock with synchronised sampling times. The system performed simple mathematical operations on the three outputs together with the demand direction signal. The two encoder signals were first subtracted to obtain the absolute position of the motor, which was then subtracted from the last value of this figure, "saved" in the previous sample period, to give the change in motor position. The change in demand position could be found in a similar manner. The difference between the demand and the encoder differences in position gave a digital signal proportional to velocity error, whilst the integral (sum) of these signals gave a figure proportional to position error. These two outputs were then passed through two separate digital to analogue converters. Operational amplifiers were used to sum the analogue signals, one proportional to velocity error and the other to position error. The relative gains between the two signals could be altered by alteration of resistor values within the analogue circuitry. The signal could also be filtered before applying it to the Bru-500 current loop, as an analogue two complex pole filter was included, whose fundamental frequency could be altered by change of resistor values.

The Molins' Dservo system was commissioned at Aston and although it had better performance than the original Bru-500 software it still had lower performance than expected. The S-6100 with a $0.01kgm^2$ load could achieve a 38Hz (@-140°) bandwidth. Detailed investigation was undertaken to understand the reasons for this problem. By use of simulation comparisons with laboratory results it was found to be due to open loop phase shift in the input stage of the Bru-500 DM unit, as explained later in section (7.7.3), which would not have been a problem to any software servo loop Electrocraft closed within their own equipment.

7.5 THE PROPORTIONAL INTEGRAL CONTROL SOFTWARE

During April 1988 Electrocraft delivered the P/I software which used a simple velocity loop whereby there was software control over two gains. The system controlled proportional and integral gains on velocity error, as shown in figure (7.6).

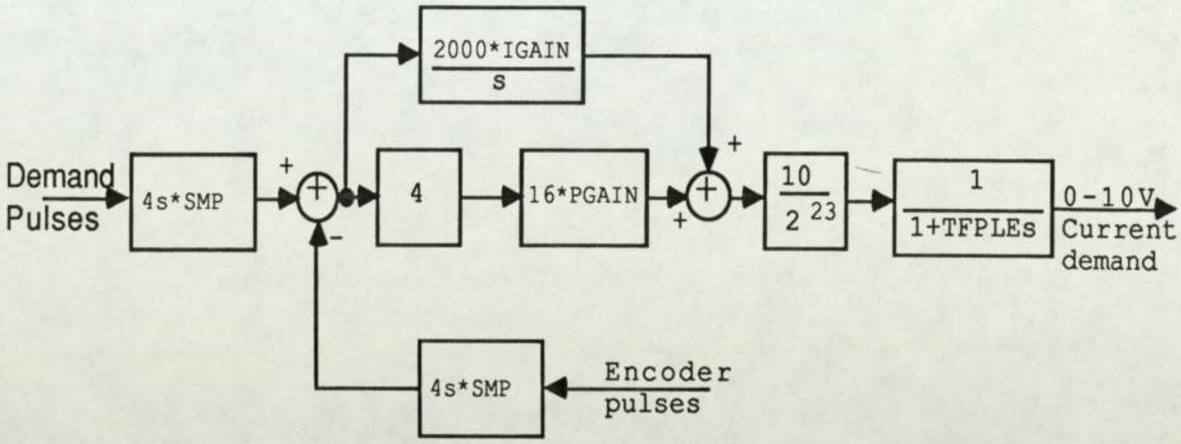


Figure 7.6 The proportional/integral algorithm

This software had superior performance to the B/D software with less phase shift at high frequency. Unfortunately the algorithm included a software limit on the gains (set to 2000) which meant that with a relatively high load (load/ motor inertia ratio approximately greater than 5) the maximum gains available were insufficient.

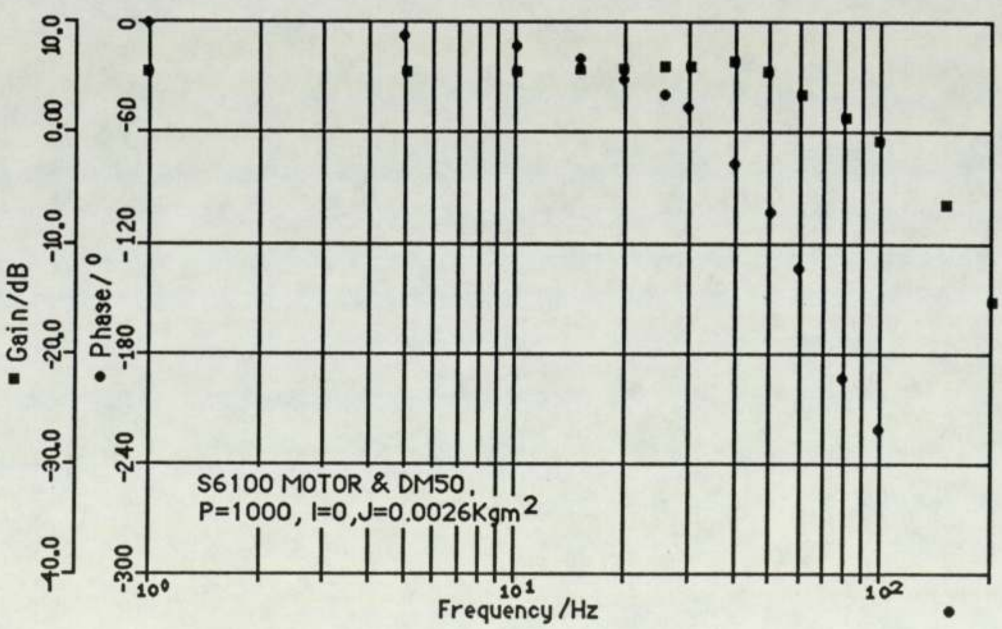


Figure 7.7 The actual frequency response of the S-6100 drive with proportional gain=1000, integral gain=0, filter bandwidth=300 and a 0.0026 kgm^2 load

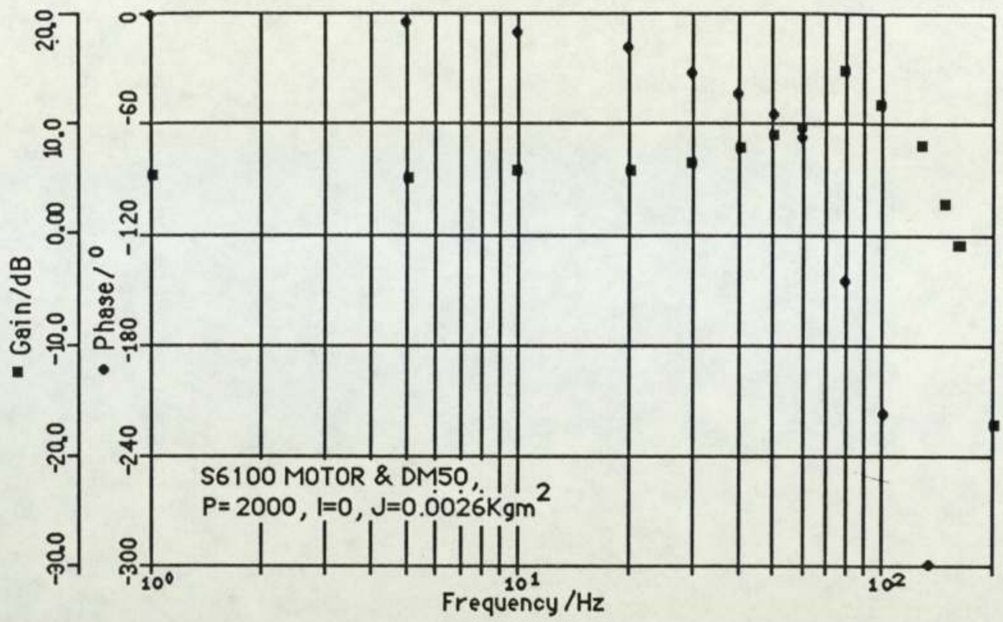


Figure 7.8 The actual frequency response of the S-6100 drive with proportional gain=2000, integral gain=0, filter bandwidth=300 and a 0.0026 kgm^2 load

Figures (7.7) and (7.8) show that with the relatively small inertia load of 0.0026kgm2, the S-6100 drive would be stable. The frequency response was "flat-topped" with a proportional gain of 1000. With a proportional gain of 2000 a resonant frequency of 80Hz was observed. For the S-6100 with a 0.0108kgm^2 load the maximum bandwidth achieved using this system was 18Hz (@-65°).

7.6 THE MODIFIED BRU-500 CURRENT LOOP

Since the P/I software performance was limited by an internal software limit, and the Dservo system by phase shifts associated with current loop input filtering and sampling delays, it seemed that the performance could not be improved further. Problems with the Bru-500 design rather than a fundamental physical limit were hampering progress. However, detailed discussions with Electrocraft personnel and access to the Bru-500 circuit diagrams led to improvement in performance.

When in torque mode, the current demand, which was a ±10V signal was filtered and fed to a voltage to frequency converter which was sampled every millisecond by the Bru-500 microprocessor. The resultant signal was monitored for safety functions, such as peak current and continuous current limits, and then outputted to the current loop as a 0-10V signal (5V is zero amperes), via a digital to analogue converter. The sampling and filter delays required by-passing in some way whilst still retaining the safety shut-down functions.

This was achieved by breaking the connection between the digital to analogue converter and current loop (by removal of a resistor) and by altering resistor values such that the current loop accepted a ±10V signal. The output from the Dservo was then fed both to the original input, and hence through the microprocessor safety functions and directly to the current loop. Thus the undesirable phase shifts had been by-passed.

The results from this system were more in line with the predictions of chapters 5.0 and 6.0, and results are presented in section (8.2). The bandwidth of the S-6100 drive with a $0.0108kgm^2$ load was extended to 92Hz (@-110°). It was then possible to close a high gain position loop around the velocity loop.

7.7.0 ENHANCEMENT TO THE SIMULATION MODEL

7.7.1 INTRODUCTION

It is evident from the block diagrams of figures (7.1) and (7.6) that the Bru-500 system could be represented by a very simple model, but was the original model a valid representation? The real system used a three phase machine and the original model had a single phase representation of this; the real system was digital with quantisation and samplers at the digital to analogue interfaces, whilst the original model was purely analogue in nature; the real system had a non-linear current characteristic as a current limit was used, and the original model did not show this. Consideration of these differences posed questions relating to the validity of these simple models for simulation studies.

A comparison was carried out which showed that at low gains without current constraints the model used for simulation was very accurate in both the time and frequency domains. However, at high gain the simulation model was more stable than the actual system and thus predictions from simulation were invalid. Thus, many different phenomena were examined to try to explain this.

The most common cause of instability was additional phase shift. Experimentation with the simulation package further substantiated this concept as increasing the forward loop sampler to 4ms from 1ms gave similar effects to those observed on the real system. Likewise the inclusion of two poles in the forward loop of the simulation model at approximately 100Hz also gave results in-line with reality. Thus a "search" for this additional phase shift was undertaken. The final model that was derived is described in more detail in appendix 4, but the areas of investigation are highlighted below.

7.7.2 THREE PHASE REPRESENTATION OF TORQUE PRODUCTION

It was noticed while conducting frequency response tests on the S-6100 motor, which is an 8 pole machine, at high gains the system exhibited a resonant frequency at about 80Hz, which altered audibly four times per revolution as the motor shaft crept round due to a small offset in the system. Similarly the six pole S-4030 resonated three times per revolution. This suggested that instability could be due to different properties between the three phase windings, the most obvious of these being the torque or back emf constant which is an open velocity loop gain. The three motor back emf's were monitored and slight differences did exist. To investigate whether these differences where the cause of the instability, the simulation model was altered to include three phase torque generation; this effect was found to be minimal and not the reason for the unknown phase shift. The resonance observed was thought to be due partially to reluctance torque variations in the motor.

7.7.3 DETAILED CURRENT LOOP SIMULATION

The simulated current loop up to this point had infinite bandwidth as current demand was translated immediately into a current. By examination of the Electrocraft current loop in terms of its internal gains and the electrical time constant, no effects should be seen until within the kHz region. To further substantiate this the block diagram of the current loop was obtained from Electrocraft and simulated.

The current loop was investigated further using the S-6100 drive. Fortunately Electrocraft provided a number of test points on the drive which could be used to monitor variables. Two of these points were of interest; the command (CMD) point which showed the voltage seen by the drive microprocessor; the microprocessor's output to the current loop (IMAG), which represented the magnitude of peak phase current required, scaled such that 10 V represented the drive's maximum current rating. A number of frequency response tests on the open loop section of the drive between the voltage input and IMAG were undertaken (the drive in torque mode). The results are given in detail in section (5.4.4), and a sample plot is shown in figure (7.9).

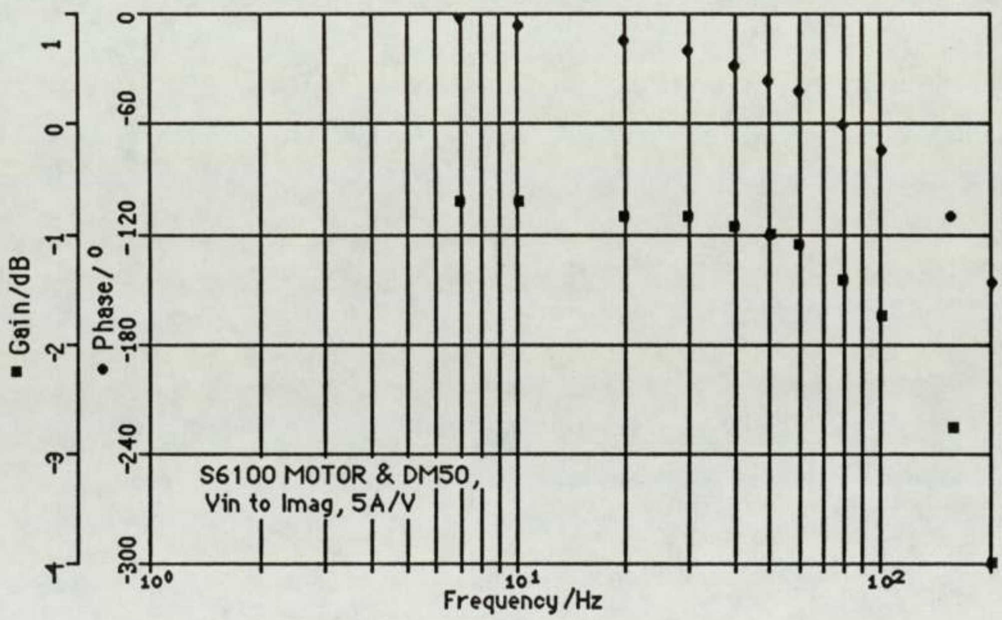


Figure 7.9 The frequency response of the Bru-500 input circuitry

From conventional theory it was expected that the open loop response would have a 20dB/ decade gain fall off with 90° of phase lag up to the kHz regions. The plots seemed to show two additional poles placed at approximately 100Hz. This response therefore showed some unexplained phase shifts seen in the open velocity loop frequency response. Thus the phase shifts and gain fall-off were not due to the current loop or formulation of the three phase current but in simply passing the voltage at the input to an ideally identical voltage waveform at the output of the microprocessor. It was known that the drive sampled at 1ms and updated current at 0.25ms intervals. In order to locate the poles causing the gain fall off step responses were taken. Two poles could be identified, one with a time constant of 0.8ms and one with a time constant of 0.2ms; these were due to a small amount of analogue filtering on the drive input, which was confirmed by Electrocraft.

Thus the Bru-500 input circuitry was by-passed as described in section (7.6).

7.7.4 DISCRETE SIMULATION OF ENCODER FEEDBACK

The simulation model used in all the above discussion was correct except for its treatment of one element, the encoder feedback. The encoder fed pulses back to the drive which were input to a counter which was reset every sample. Thus the count in any one sample was proportional to the velocity to one count in one sample accuracy (this is 1/8000th rev per 1ms, or 7.5r/min). Such systems can never lose a pulse so quantisation errors or the remainders from one sample are added to the next. Thus whilst the instantaneous error may be only within 0.8rads^{-1} the average velocity signal over many seconds is exact.

What was not appreciated at the beginning of the simulation exercise was the effect of these remainders. The simulation originally used the ZOH operator which successfully mimicked sampling delays but the QNTZR operator (see the Acsl manual [1]), whilst quantising a signal did not add up the cumulative errors. Thus in the simulation model the instantaneous and long term feedback signals were in error by up to one bit per sample. The nature of the feedback waveform was thus incorrect as the true waveform changed between two states at a constant velocity, whilst the simulation feedback was smooth.

In order to investigate this phenomenon it was necessary to write Acsl code in a DISCRETE section.

7.7.5 BACKLASH

Backlash introduces a phase shift into a system, but it was very difficult to include in simulation, as the low torsional time constants combined with negligible damping caused numerical instability of the software. Since taper lock couplings with keyways were used this problem was considered insignificant.

7.7.6 RESONANCES

Throughout the experimental phase of the research mechanical resonances were observed, especially upon the S-4030 system which had a high load/ motor inertia ratio. At low speed a high frequency audible resonance was observed. A microphone pick-up monitored the 1.25 kHz resonance from the S-4030 system with its 0.0113 kgm^2 load. The resonance was still evident at a slightly higher frequency without the load. By moving the load along the motor shaft the level of resonance but not the frequency altered. Also the amplitude increased substantially at three points per revolution suggesting a link with the number of pole pairs and therefore the machine design. This phenomenon was thought to be due to the dead-band introduced by the encoder: the system "rattled" between two encoder lines at standstill. The effect was exasperated by any change of gain or increase in phase shift due to other reasons such as movement of the load along the shaft or changes in reluctance torque (thought to be the reason for the three times per revolution problem).

A further audible resonance was observed when the drives were running at high speed. The large motor test-bed vibrated with a peak amplitude at 2500r/min. By use of a microphone it was found that this vibration altered with the motor speed. it was due to slight "out-of-balances" in the load flywheels. The resonant frequency of the test-bed was 42Hz (2520 r/min), which was altered by adding large weights to the bed, such that test-bed resonance was negligible at the running speed of 2500 r/min.

A further resonance that was observed was due to the resonant frequency of the position loop. The amplitude again increased as the load was moved further along the shaft, which suggested that it could be excited by interaction with system mechanics.

Resonance due to the test-bed was irrelevant to the work and so no attempt was made to simulate it. The resonance at low speed was simulated in some degree by using a discrete section as described in section (7.7.4), but the simulated effect was not as severe as seen on the actual system. The system's natural resonant frequency was also observed in simulation but again not as severe as seen on the actual system.

The disparity between simulated and actual system phenomenon was due primarily to the lack of an accurate load model as described in section (6.4.3), as the "bell-resonance" effects exasperate the other phenomena described. Attempts to accurately simulate load resonances proved unsuccessful for the reasons described in section (7.7.5).

7.8 CONCLUSIONS

The drives were initially supplied with the B/D software, but laboratory tests showed that this was a poor control algorithm in respect of the large phase shifts found at high gain. The phase roll off beyond the system bandwidth was very steep which introduced problems when trying to close position loops. It was suggested to Electrocraft, using simulation evidence, that they implement a simple proportional loop, as this would give superior performance. The P/I software was far superior to the original algorithm, but it included a software limit on the available proportional and integral gains. The Molins Dservo system was a pure proportional velocity loop and a position loop with velocity feedforward. It originally had disappointing performance due to the input filtering and sampling of the Bru-500 system; once this was by-passed the Dservo gave satisfactory performance. The system bandwidths were substantially increased on the S-6100 system: with a 0.05kgm^2 load the maximum B/D control velocity loop bandwidth was 17Hz (@-100°), and with a 0.0108kgm^2 load the P/I control achieved 18Hz (@-65°), the Dservo control had 38Hz(@-140°), which was improved by modifications to the Bru-500 to 92Hz (@-110°), see Seaward and Johnson [62]. Thus nearly a fourfold increase in bandwidth was achieved.

Extensive simulation of the Bru-500 drive was undertaken as detailed in appendix 4. It has been shown that simulation can be a powerful tool to aid the design process, but a valid model must be available and a methodical approach is required should the system and simulation give different results. From the original lumped parameter model that was invalid for high gain settings a much more complex model was derived, although attempts to simulate the load in greater detail were unsuccessful. The torque derivation was represented in a three rather than single phase form and the current loop control was fully implemented. The controller was also represented in a DISCRETE block of code for greater accuracy. Phase shifts due to input circuitry and sampling were also discovered in the Bru-500 that effected performance of any external control loop.

The Electrocraft Bru-500 drives were investigated to a sufficient level to enable the simulation models derived to predict accurate results that could be used with confidence.

CHAPTER 8

8.0 RESULTS FROM THE DEMONSTRATOR TEST BED

8.1 INTRODUCTION

This chapter presents the laboratory results that are of particular relevance to the research. The laboratory equipment used is described with justifications for the system configuration.

The S-4030 and S-6100 systems were both optimised to give maximum velocity loop bandwidths and position loop gains. The S-6100 system was also "matched" to the S-4030 as described in section (4.4.5). The frequency and step responses of these systems are presented. As the limit to the S-4030 system was due to resonance, an investigation into the causes of this phenomenon are also discussed.

As the original specification of section (1.5) called for positional following capability, the following and relative position errors are presented for ten second ramps between 0-2500r/min. These position errors were obtained by use of dedicated circuitry which is described in appendix 6.

8.2 THE EQUIPMENT USED

The demonstrator rig that was commissioned as part of the research project consisted of two servo drives that were to be synchronised. The drives were both from the Electrocraft Bru-500 range of drives, one being a S-6100 motor with a DM-50 amplifier and the other a S-4030 motor with a DM-25 amplifier. The inertia loads fitted to the motors were 0.0108 kgm^2 to the S-6100 and 0.0113 kgm^2 to the S-4030; these loads were chosen as they approximated to the ledger system inertia and because the flywheels were available for use on the project. The load inertias were of similar magnitude but the ratio of load to motor inertia was 7:1 for the S-6100 system and 43:1 for the S-4030; thus the effects of large inertia mismatch could be investigated on the S-4030 system, where torsional resonance was thought to be one of the limiting factors.

The motors were controlled using the Molins Dservo controllers as described in chapter 7.0, with the modifications made to the Electrocraft systems to by-pass the Bru-500 microprocessors and their associated phase shifts. The controller gains were each set at maximum without causing instability or drive shut-down, and in a second experiment gains were adjusted such that the drives were matched as described in section (4.4.5). The Dservo systems were set up such that a master/ slave relationship could be utilised if required. The system was such that both drives received an individual position demand, and the position error of one drive was passed between the controllers to be added to the demand of the other.

The position demand signals had to be derived as pulse trains. Indexer systems were primarily designed to derive set points for stepper motors and they output a series of pulses and direction lines as demand signals. Thus an indexer could be used as the master demand controller. A review of indexers suggested that the most cost-effective and flexible solution was to purchase a Digiplan PC23 indexer, which consisted of a single card fitted within an IBM PC clone and being capable of controlling up to three individual axes. It also had the ability to synchronise the output pulse trains to set up master/ slave relationships with "zero error between the demands". The system could then be controlled by software residing within the PC. Thus a PC23 was successfully commissioned for use within an available IBM PC clone.

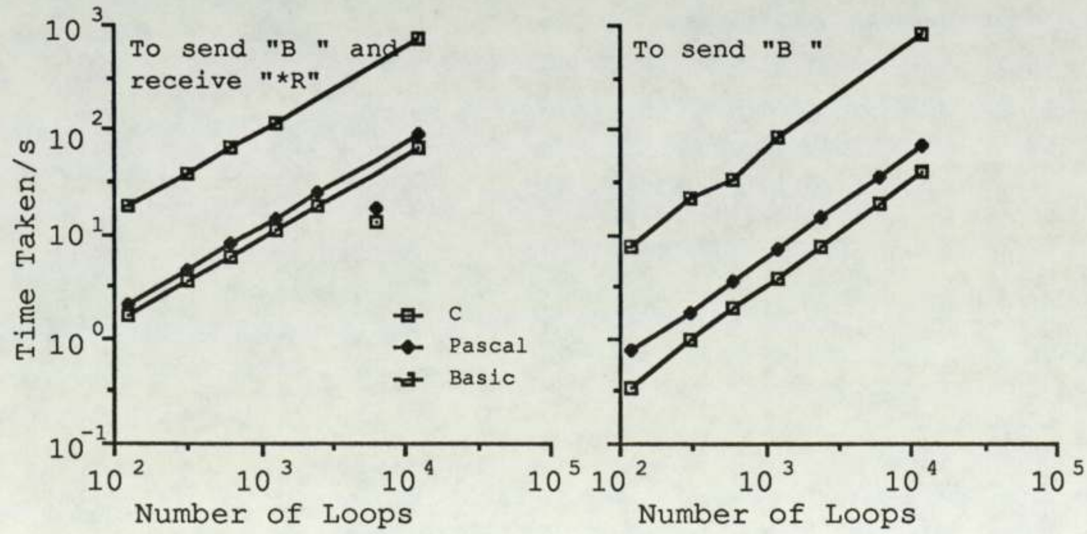


Figure 8.1 The comparison of language ability to send and receive characters to and from the PC23

A brief review of software languages suggested that Lattice "C" was the most suitable for the project; simple communications were timed for large numbers of loops. Figure (8.1) shows that "C" could communicate with the PC23 card faster than the other languages tested. The program written for the demonstrator is shown in appendix 5 and it was written such that the two axis system could be retro-fitted to a rod making machine. It was able to show the various control possibilities available. The system was menu driven with the master menu allowing:-

1) Homing of the axes to the index pulses of the encoders
2) Offsetting the axes to allow for mechanical misalignment
3) Running the axes up a user defined ramp in synchronism

For any practical system the axes must be homed to some mechanical datum and this was achieved by moving the axes to the index pulses on the encoders and then moving one axis the correct distance such that the mechanical system was aligned. The PC23 could home axes to their indexer pulses automatically as the encoder signals were fed back to it, and it fed pulses to the drive at a pre-determined slew rate until the indexer pulse was located. The offset routine was written such that the system could be manually offset with the system recording the offset from the datum position. The final offset could be saved to disc and the drives could be automatically moved to some previously saved value.

The menu to move the drives in synchronism was such that any ramp could be programmed from the keyboard, so long as predefined acceleration rates or maximum speed were not exceeded. The profile was defined by the maximum steady-state velocity (in r/min with a direction) and the time to achieve this speed (in seconds). A default profile of 0-2500 r/min in 10 seconds was included. The drives could be stopped in one of two modes: firstly by pressing "S" on the keyboard the motors would decelerate at the same rate as the acceleration, or by pressing "E" an emergency stop would be demanded whereby the drives use the maximum deceleration rate, but synchronism could be lost under this option. The system had a further enhancement such that by pressing "<" or ">" the drives could be advanced or retarded by one pulse relative to one another, and the system stored a running total of these moves. This was introduced to show the flexibility of the system, and to allow the closure of further slow acting control loop which could integrate out steady-state errors in the system. All control was from the keyboard but the "interrupts" could equally be initiated from an external signal.

Thus a system was developed that could be used upon a rod-making machine and also introduced great flexibility into the demonstrator equipment.

The default ramp of 0-2500r/min in 10 seconds was chosen as it ran the S-6100 motor at the velocity required by section (1.5). Thus the limits of performance could be identified using the demonstrator rig.

8.3 SERVO SYSTEM OPTIMISATION

The individual servo systems required optimisation with respect to velocity loop bandwidth and position loop gain. The sampling period was set to 0.5ms and the in-line filter to 150Hz to reduce any quantisation ripple; these figures were selected from previous experience with the systems. The velocity loop was first optimised by increasing the forward loop gain until near the point of instability. The step responses and frequency responses were used as indicators of stability. When the time response started to become oscillatory or the frequency response showed a resonant peak the gain was reduced slightly.

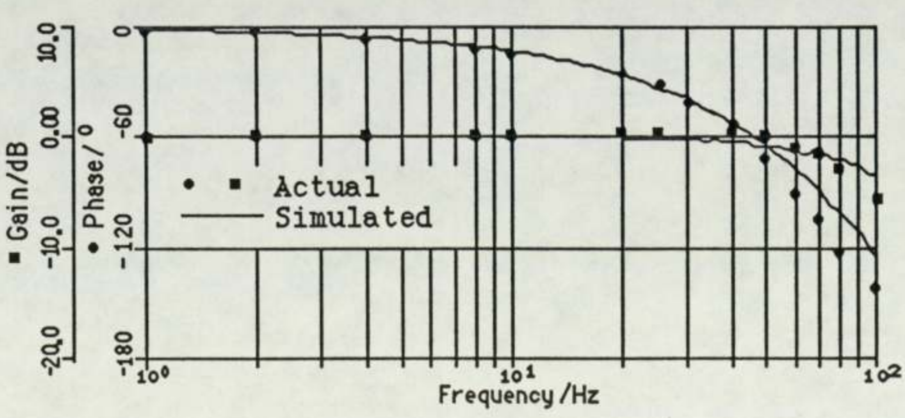


Figure 8.2 The frequency response of the S-6100 system with a 0.0108 kgm^2 load and with the maximum velocity loop bandwidth of 92 Hz (@-110°)

The position loop was optimised by increasing the gain until the system became too oscillatory in nature; this was exhibited by excessive acoustic noise being generated by the system.

The S-6100 had a maximum velocity loop bandwidth of 92Hz (@-110°), as shown in figure (8.2), and position loop gain of $275s^{-1}$. The simulated step response with and without the position loop activated are shown in figures (8.3) and (8.4).

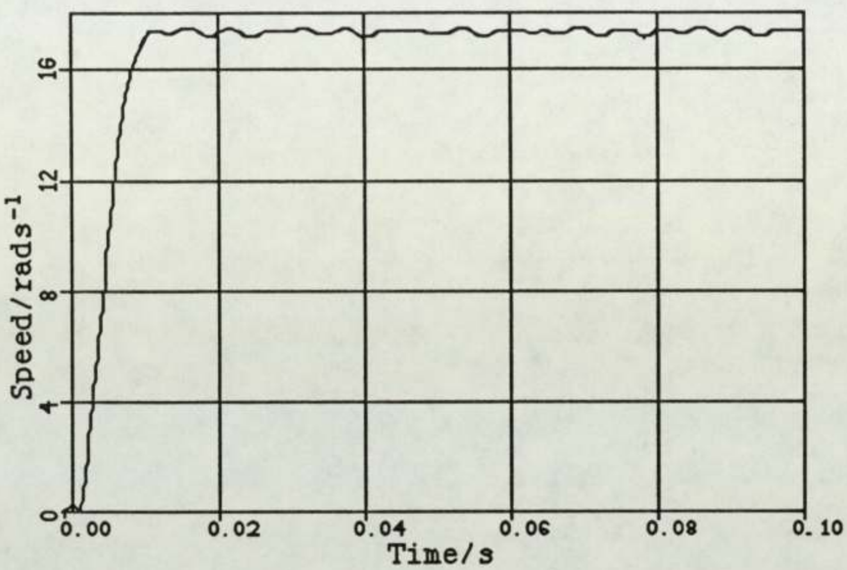


Figure 8.3 The simulated step response of the S-6100 system with a 0.0108 kgm^2 load and with the maximum velocity loop bandwidth of 92 Hz (@-110°) and no position loop

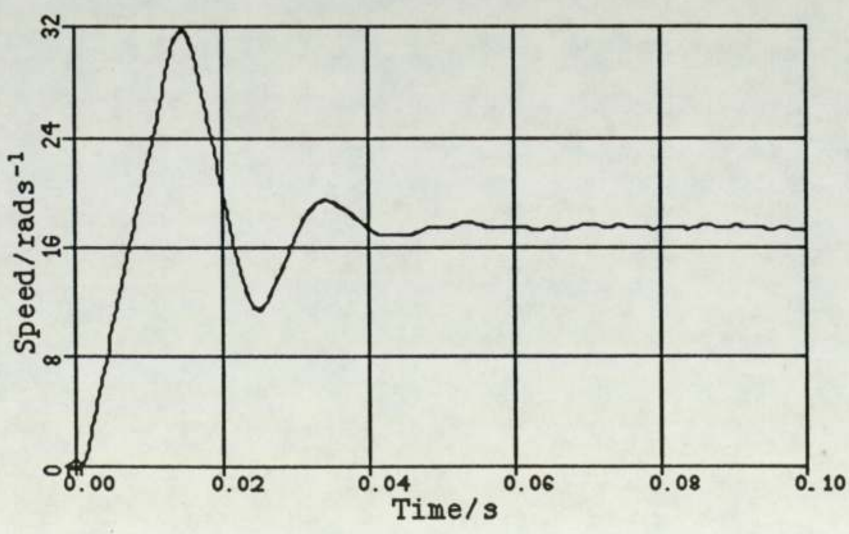


Figure 8.4 The simulated step response of the S-6100 system with a 0.0108 kgm^2 load and with the maximum velocity loop bandwidth of 92 Hz (@-110°) and a position loop of gain $275s^{-1}$

The S-4030 system was optimised such that the maximum velocity loop bandwidth was 18Hz (@-65°), as shown in figure (8.5) and position loop gain was $200s^{-1}$. The limitation was due to resonance problems. The step responses are shown in figure (8.6) and (8.7).

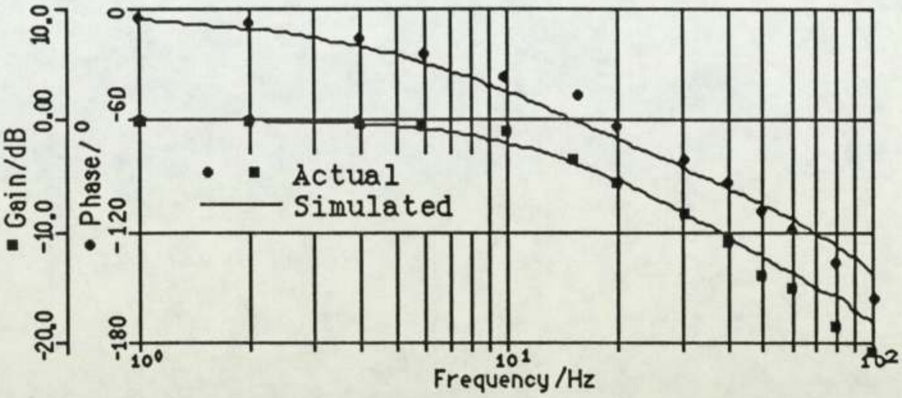


Figure 8.5 The frequency response of the S-4030 system with a 0.0113 kgm^2 load and with the maximum velocity loop bandwidth of 18 Hz (@-65°)

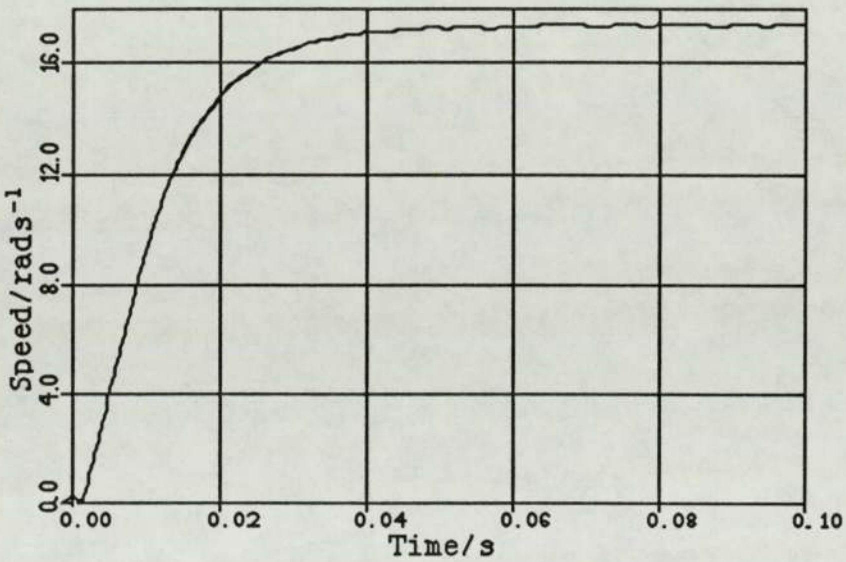


Figure 8.6 The simulated step response of the S-4030 system with a 0.0113 kgm^2 load and with the maximum velocity loop bandwidth of 18 Hz (@-65°) and no position loop

To analyse the effects of matching the drives the S-6100 had its servo loop gains altered such that the velocity loop had a bandwidth of 18Hz (@-55°), figures (8.8) and (8.9), and position loop gain of $200s^{-1}$, which were very similar to the S-4030 settings.

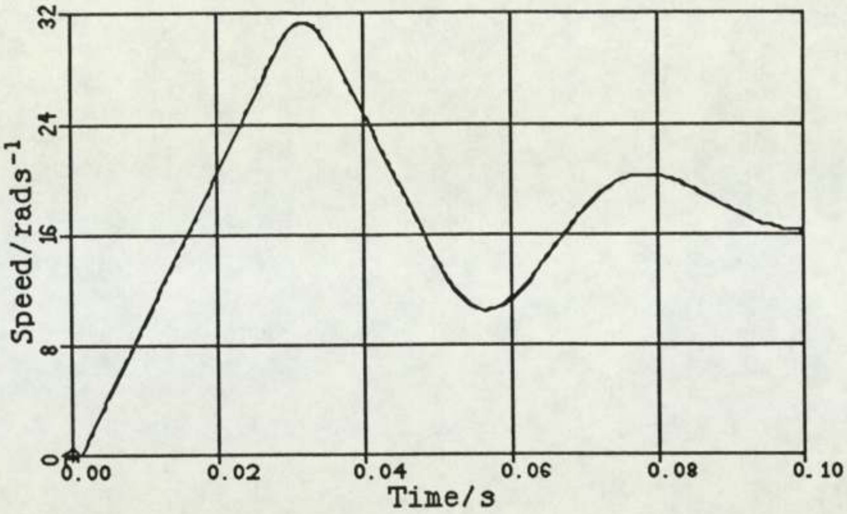


Figure 8.7 The simulated step response of the S-4030 system with a 0.0113 kgm^2 load and with the maximum velocity loop bandwidth of 18 Hz (@-65°) and a position loop of gain $200s^{-1}$

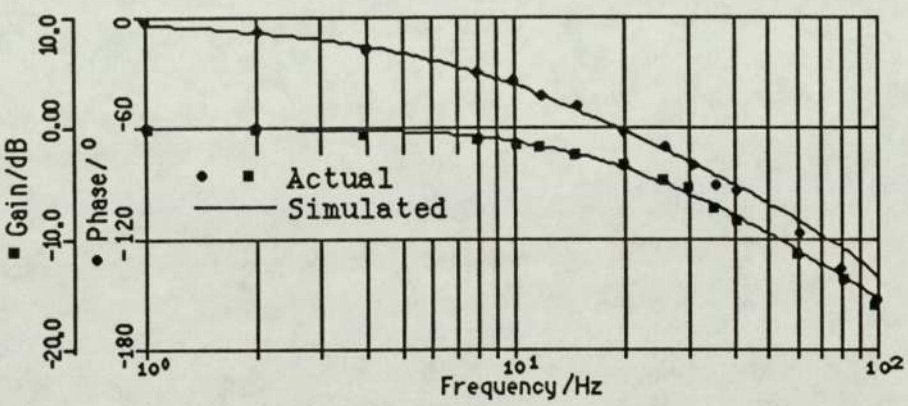


Figure 8.8 The frequency response of the S-6100 system with a 0.0108 kgm^2 load and with the matched velocity loop bandwidth of 18 Hz (@-55°)

Analysis from previous chapters suggested that with a 0.5ms sampler the maximum velocity loop bandwidth due to phase shifts with K=20 was approximately 100Hz, section (5.3.2), and the maximum position loop gain was 400 s^{-1}, section (5.3.3). The effect of quantisation was analysed in section (6.2), which suggested maximum values for velocity loop bandwidth of 630Hz for the S-6100 and 140Hz for the S-4030 system; the maximum position loop gain due to quantisation was $2000s^{-1}$. Thus, the limit to the S-6100 performance was very close to that predicted when considering phase shifts due to sampling whilst the S-4030 system was substantially less than this. The reduction in the S-

4030 system performance was due to the mismatch in load to motor inertia (43:1) which caused resonance as the system gains were increased. Quantisation effects were not limiting factors for these systems.

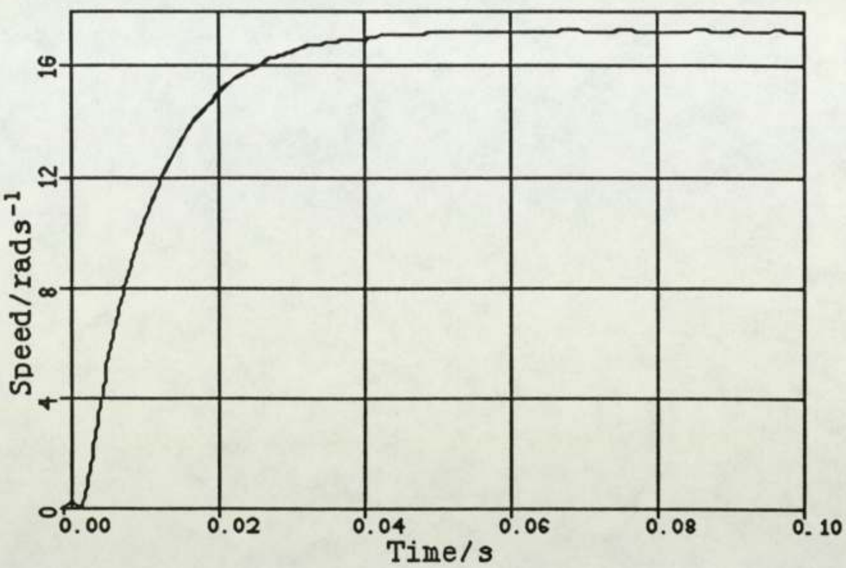


Figure 8.9 The simulated step response of the S-6100 system with a 0.0108 kgm^2 load and with the matched velocity loop bandwidth of 18 Hz (@-55°) and no position loop (cf. figures (8.3) and (8.6))

8.4 RESONANCE EFFECTS

The performance of the demonstrator system was greatly reduced due to resonant problems which were observed on the S-4030 system. The load inertia fitted to the motor was 0.0113kgm^2 as compared to the motor inertia of 0.00026kgm^2 a mismatch of 43:1. Reference to section (6.4.3) suggested that the torsional resonant frequencies that should be exhibited would be in the order of 1 kHz for the motor denominator resonance and 150 Hz for the load numerator resonance (L_s=0.1m, d=0.019m therefore K_s=10235). The actual frequency observed was 19.5 Hz as shown in figure (8.10).

A simulation investigation found no evidence for resonance when the system followed the ramp profile. However, step responses such as those shown in figure (8.4) and (8.7) showed under-damped oscillations at the frequency expected (20Hz). Thus far no reference to the position loop frequency response has been made as it proved impossible to take on the actual system, however, the position loop frequency responses were obtained from the simulation environment. Figure (8.11) shows the frequency responses for the optimised S-4030 system, figure (8.12) for the optimised S-6100 system and figure (8.13) for the matched S-6100 system.

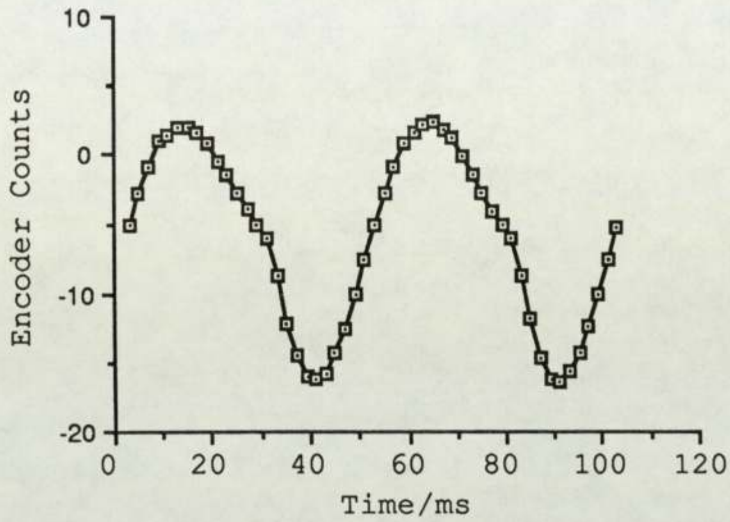


Figure 8.10 Absolute error of the S-4030 system 5s into a 10s ramp from 0-2500 r/min showing the resonance

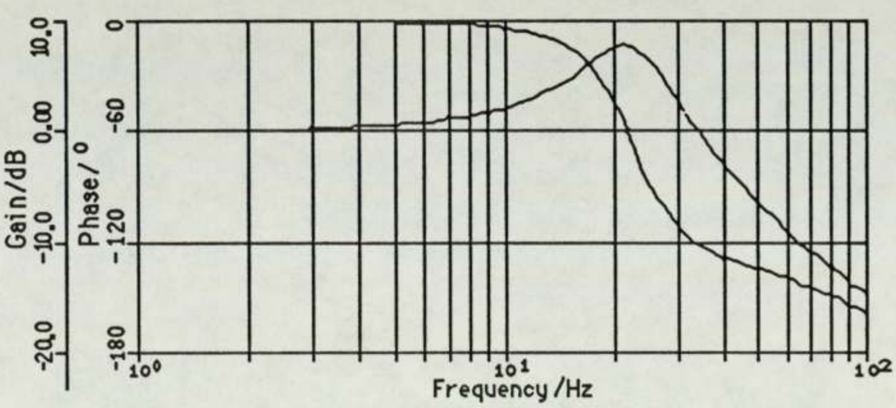


Figure 8.11 The simulated frequency response of the S-4030 system with a 0.0113 kgm^2 load and with the optimised velocity loop bandwidth of 18 Hz (@-65°) and a position loop gain of $200s^{-1}$

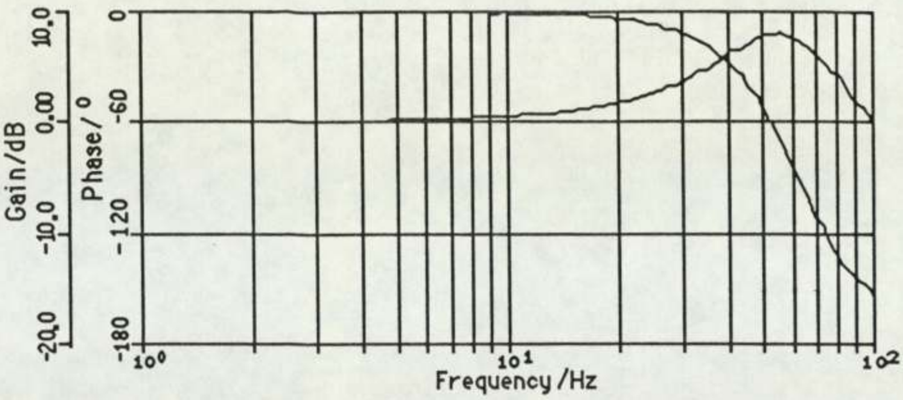


Figure 8.12 The simulated frequency response of the S-6100 system with a 0.0108 kgm^2 load and with the optimised velocity loop bandwidth of 92 Hz (@-110°) and a position loop gain of $275s^{-1}$

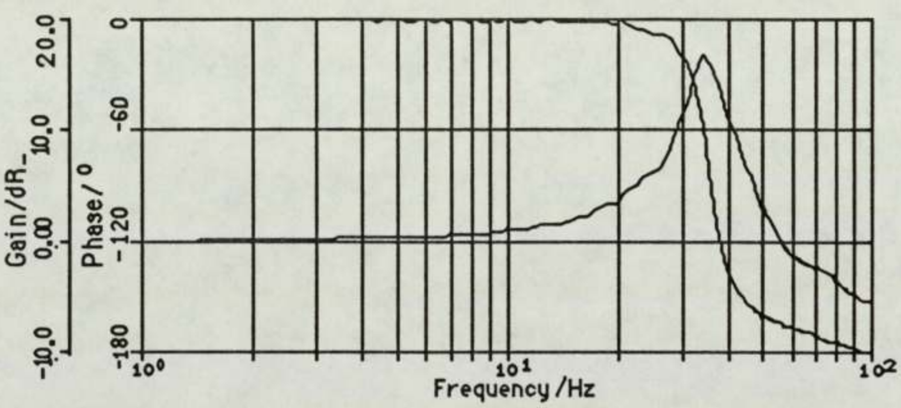


Figure 8.13 The simulated frequency response of the S-6100 system with a 0.0108 kgm^2 load and with the matched velocity loop bandwidth of 18 Hz (@-55°) and a position loop gain of $200s^{-1}$

The S-4030 had a resonant frequency of 21 Hz with a +8dB rise in gain, the optimised S-6100 a +8dB rise at 55Hz and the matched S-6100 system a +16dB rise at 34Hz. From this result alone the matched S-6100 system would therefore be expected to exhibit the most pronounced resonance at 34Hz with both the optimised S-4030 and S-6100 systems to have similar amplitude resonances at 20Hz and 55Hz respectively. The actual systems behaved differently though with the S-4030 systems showing resonance with an approximate amplitude of ±10 counts from the average at approximately 20Hz, the matched S-6100 system a resonance of ±4 counts at 40 Hz and the optimised S-6100 system no noticeable resonance.

The standard method for overcoming such a problem, Dorf [20], is to use proportional plus derivative action in the control loop usually in the form of derivative feedback. Unfortunately the maximum derivative feedback in the form of the velocity loop was being utilised. The only other method to reduce resonance would be to reduce the position open-loop gain in some way. In simulation a fourfold reduction in position loop gain was required to reduce the resonant peak to within +3dB, and such a system would have approximately four times the following position errors. Thus the error was minimised for the S-4030 with the resonances and the system had maximum performance.

The resonant peak in the position loop frequency response could not be the only reason for the resonance observed, as not all the drives exhibited the phenomenon. The major difference between the drives was the mechanical torsional resonance due to mismatch in load and motor inertias. The S-4030 had a motor denominator resonance (see section 6.4.3) at 1kHz with a 43:1 mismatch whilst the S-6100 system had a 1.8kHz resonance with a 7:1 mismatch.

The increase in gain with increased phase lag at the motor inertia resonant frequency in the denominator is not offset by the load inertia resonant frequency in the numerator if a large inertia mismatch exists. This therefore exasperates the natural resonant problems of the system.

The design methods outlined in chapters 5.0 and 6.0 did not predict this problem and will require enhancement to take account of this phenomenon.

8.5.0 RESULTS

8.5.1 PRESENTATION OF RESULTS

Circuitry described in appendix 6, was designed to monitor errors between two pulse trains, and to produce accurate results for the relative errors between the drives or the absolute errors between output and demand. These results have been presented in the simplest form possible without losing pertinent information. During a simple 10s ramp from 0-2500 r/min the motor moved by 208 revolutions which relates to 1.7 million transition states of the encoder and 1.7 million of the demand. Obviously each transition could not be recorded. The system errors were therefore sampled every 10ms by a logic analyser which still produced 1000 data points for one run. Tests with higher frequency sampling were taken to ensure that no significant high frequency oscillations were being lost and the highest frequency disturbance found was approximately 20Hz so that 100Hz sampling was sufficiently fast to capture all information. The 20Hz disturbance was due to the resonance effect on the S-4030 motor, discussed in section (8.4) which meant that results from one run to another were not consistent at any instant in time. To represent this the average over 0.2s was taken to show the underlying error whilst the maximum and minimum values of position error were also shown to show the effect of the resonance. In this way each run was split into 50 time zones each represented by 3 blocks of data, the maximum, the minimum and the average position error.

Whilst no formal statistical analysis was undertaken upon the data, the average values were consistent with what would be expected. Occasional spurious maximum or minimum results were observed. The systems oscillated between ±2 encoder counts at standstill, so that when the counter system was initialised it could have up to four counts of error. Also due to the counting system within the error monitor circuitry its output could have an error of one count. Thus a conceivable error of five counts could be placed on any result taken, although it was probable that the results were more accurate than this. Some of the minor inconsistencies could be explained by this phenomenon.

8.5.2 TESTS ON THE PC23 DEMAND SIGNAL

The first tests using the error monitor circuitry were tried on the PC23 demand signals to ensure that they were perfectly synchronised as quoted in the user manual [19]. The difference between the two demand signals was monitored during a 10s ramp up to 2500 r/min and errors between the signals were observed as shown in figure (8.14).

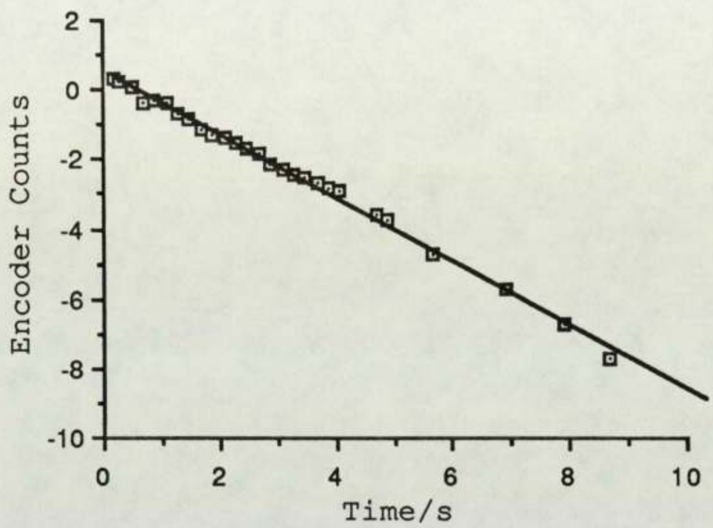


Figure 8.14 Relative error in the PC23 demand signals during a 10s ramp from 0-2500 r/min

The error was linear with time and further tests showed that there was a relationship between the relative error and pulse frequency, such that there was one pulse of error for every 300r/min (40kHz). Thus it could be deduced that the demand signals had a constant time delay between them of approximately 25μs. An error of 10 pulses in the demand at 3000r/min was unacceptable when the original specification called for accuracies in the order of 3 counts. The error could be compensated by software alteration.

The short-term solution however was to use one of the demand signals fed to both the motors for the remaining tests described within this chapter.

8.5.3 RESULTS FROM THE OPTIMUM SYSTEM

The two servo systems were optimised such that the velocity loop bandwidth and position loop gains were the largest obtainable values without the system either becoming unstable or shutting down due to overcurrent.

The S-4030 system was capable of maintaining average absolute position error between +2 and -9 counts throughout the acceleration, steady-state and deceleration zones, but due to the resonance problem described in section (8.4) the instantaneous absolute position error varied between +10 to -20 counts. The system therefore varied by as much as ±11 counts from the average, shown in figure (8.15). The S-4030 could only hold position within 20 counts which is seven times larger than the required value.

As with all the other plots the steady-state error was not effected significantly by resonance. The standstill error of the other drive was within ±2 counts and at 2500r/min the absolute position error had an average value of -6.75 counts varying between -6 and -8 counts. There was therefore more certainty of the error measurement at constant speed.

The average error increased linearly with speed due to the effect of viscous damping which increased with velocity, and the position error required to accelerate the load was negligible in comparison to the viscous drag since a very low acceleration rate was used. The resonance effect though was more noticeable during acceleration/ deceleration zones.

The average absolute position errors for the optimised S-6100 drive varied by the same amount as the S-4030 system, that is, between +2 and -9 counts but the instantaneous variation was only between +7 and -10 counts which was much lower. The variation from the mean was no more than ±6 counts. At 2500 r/min the average error was -9 counts varying between -8 and -10 counts as shown in figure (8.16).

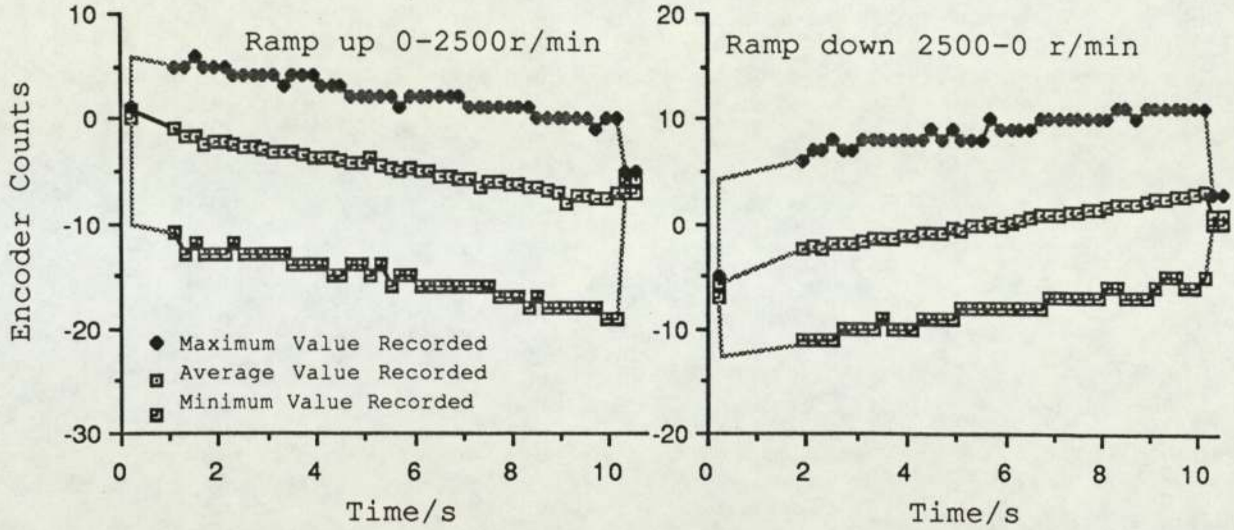


Figure 8.15 Absolute error of the S-4030 system during 10s ramps between 2500 r/min and 0 r/min

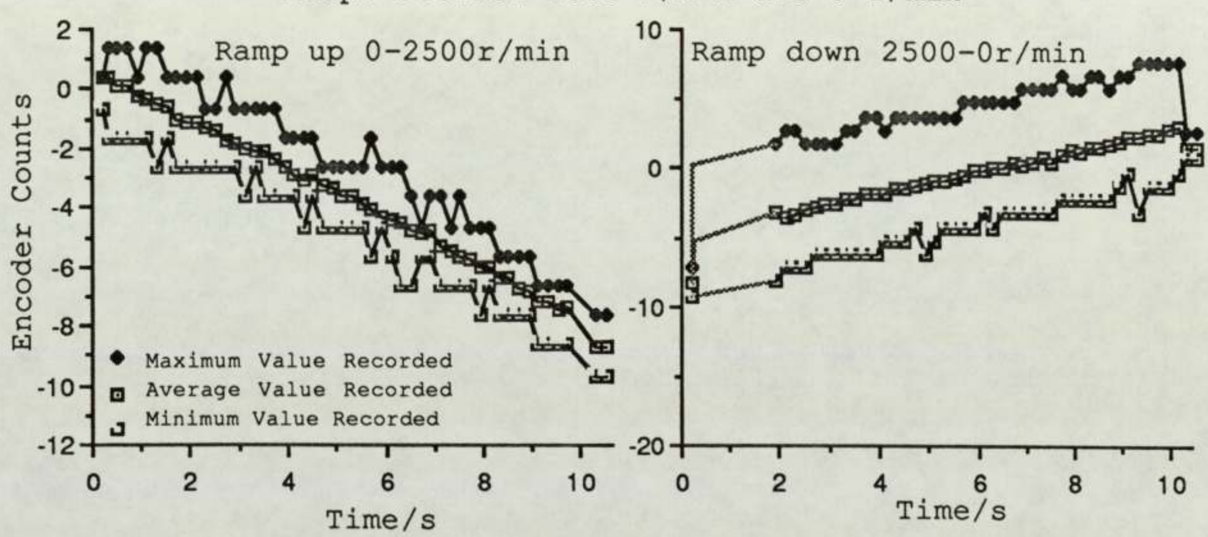


Figure 8.16 Absolute error of the S-6100 system during 10s ramps between 2500 r/min and 0 r/min

When the system was placed in master/ slave mode the absolute errors of the S-6100 system were recorded, as shown in figure (8.17). This error represented the combined individual absolute error of the two systems. The S-6100 attempted to follow the S-4030 as its demand. In this instance the average position error varied between +3 and -11 counts which instantaneously varied between +17 and -26. The maximum and minimum errors observed were in-line with the theory that the absolute error of the S-6100 when slaved to the S-4030 was the summation of the two individual absolute errors. However, the average errors were substantially less than would be expected, which was thought to be due to phase differences between the resonance of the S-4030 and the "sympathetic" resonance of the S-6100 in master/ slave mode. These were very large absolute errors but consideration should be made of relative rather than instantaneous errors.

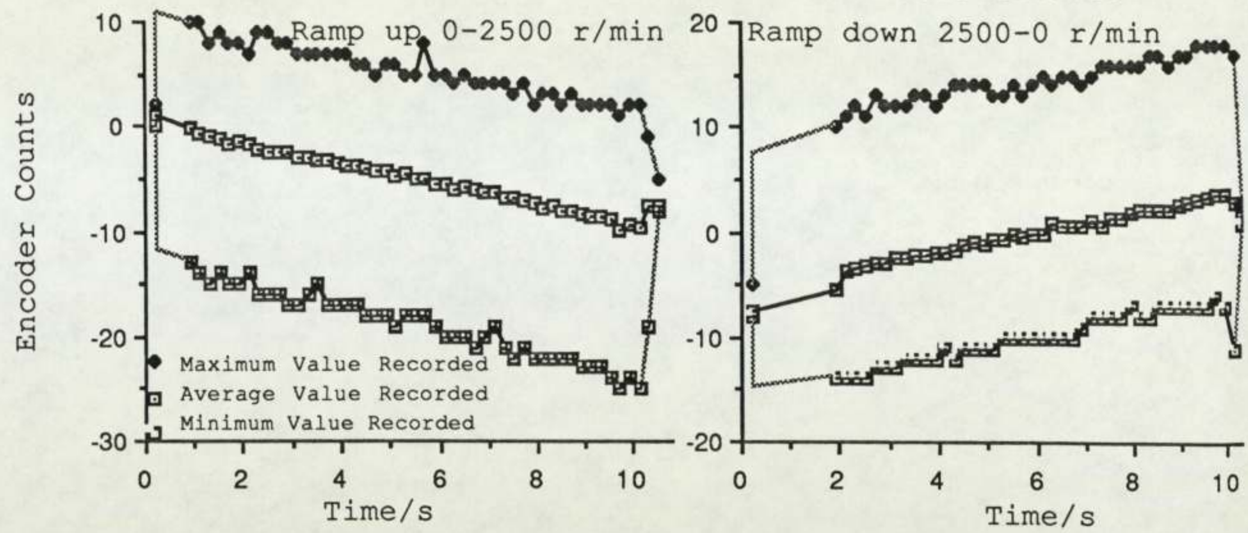


Figure 8.17 Absolute error of the S-6100 system with 10s ramps between 2500 r/min and 0 r/min with the master/slave topology

The relative error without using the master/slave relationship is shown in figure (8.18). The average relative position error between the drives varied between +3 to -2 counts with an instantaneous variation between +15 and -13 counts. The values were steady throughout the acceleration and deceleration zones suggesting that the drives were equally effected by viscous damping. The average error was within the original specification but the system could not be implemented because of the higher instantaneous errors which were caused by the resonance in the S-4030 motor. At a continuous velocity of 2500r/min the average error was only +1 count varying between +3 and -1 counts which was acceptable. This error was the difference between the two absolute errors which were -9 and -6.75 counts, a difference of 2.25 counts. One possible way of using this system would be to further reduce the acceleration time such that synchronisation was kept within limits.

The optimised drive relative position error was also tested using the master/ slave control topology. The average relative position error was similar to the system without the inclusion of the master/ slave topology at between +4 and -2 counts but the instantaneous error varied between +8 and -4 counts. Thus the instantaneous error was improved by a factor of two upon the simple system and it was only 2.5 times larger than required, as shown in figure (8.19). This improvement was due to the fact that the S-6100, with an 92Hz velocity loop bandwidth, could effectively follow the 20Hz oscillations in the S-4030 system. At steady state the error was larger than for the simple system at an average of +3.65 counts varying between +5 and +3 counts. This was as would be expected because the relative error was essentially the absolute error of the S-6100 system (rather than the difference between the two absolute errors), there was however a discrepancy in result here as the expected absolute error of the S-6100 was -9 counts at 2500 r/min such that the expected relative error at this speed would be +9 counts not +3.65 counts as recorded.

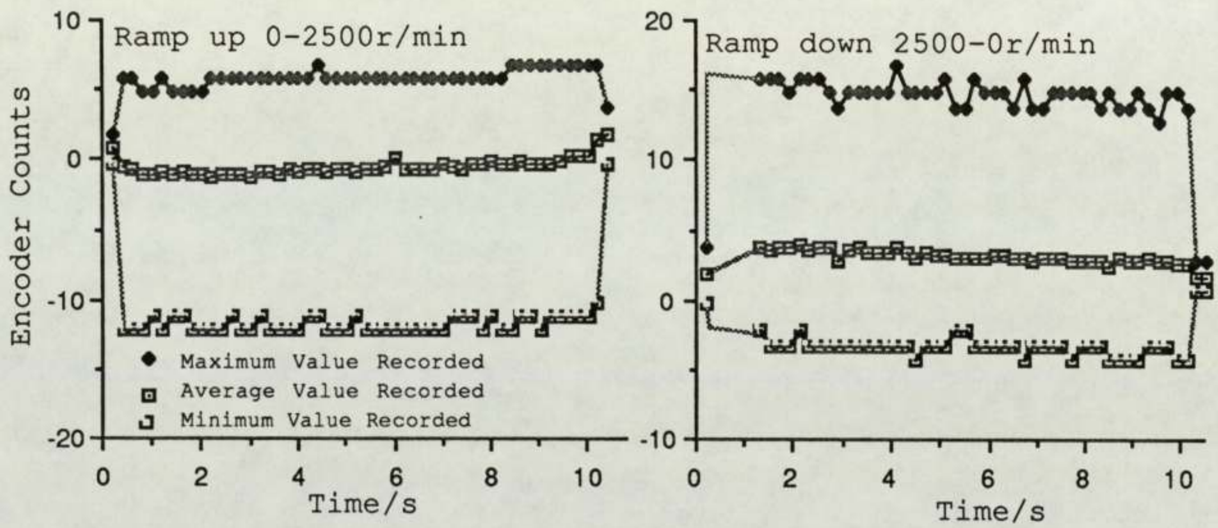


Figure 8.18 Relative error between the drives during 10s ramps between 2500 r/min and 0 r/min

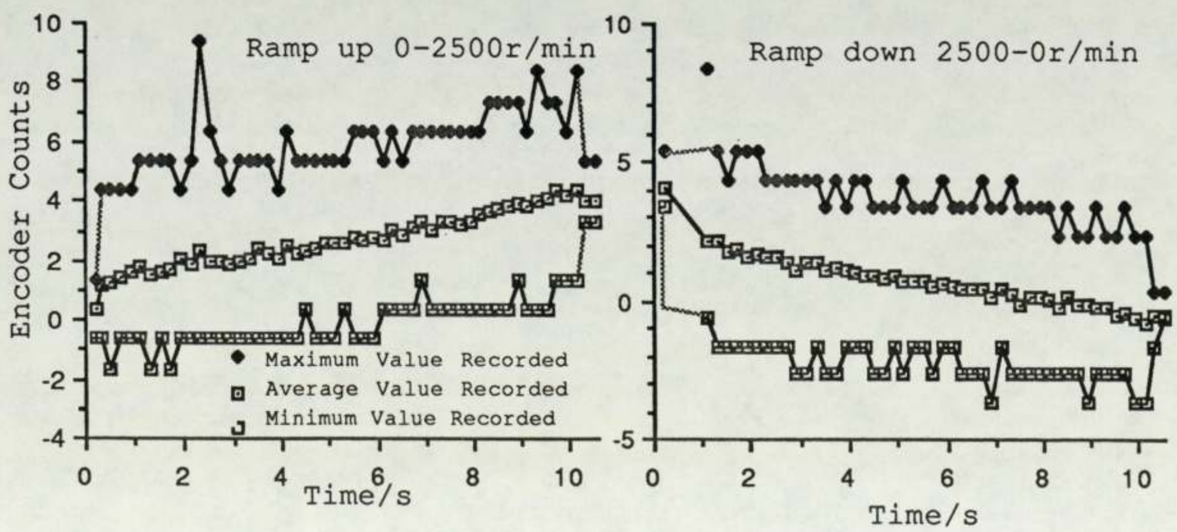


Figure 8.19 Relative error between the drives during 10s ramps between 2500 r/min and 0 r/min with the master/slave topology

The results from the optimised drive systems showed that at a steady-state velocity the relative position error between the drives was reduced by using the simple system, whilst the master/ slave topology gave improvements during the ramp. The use of matching was therefore considered to enable further error reduction.

8.5.4 RESULTS FROM THE MATCHED SYSTEM

The S-6100 was set to have a similar velocity loop bandwidth and position loop gain as the S-4030 system.

The absolute average errors of the matched S-6100 now varied between +1 and -11 counts (cf. +2 to -9 previously) varying between +7 and -15 as shown in figure (8.20). At a continuous high speed of 2500r/min the average absolute error was -10.4 counts varying between -8 and -12 counts (cf. -6.75 counts between -6 to -8 for the S-4030 system).

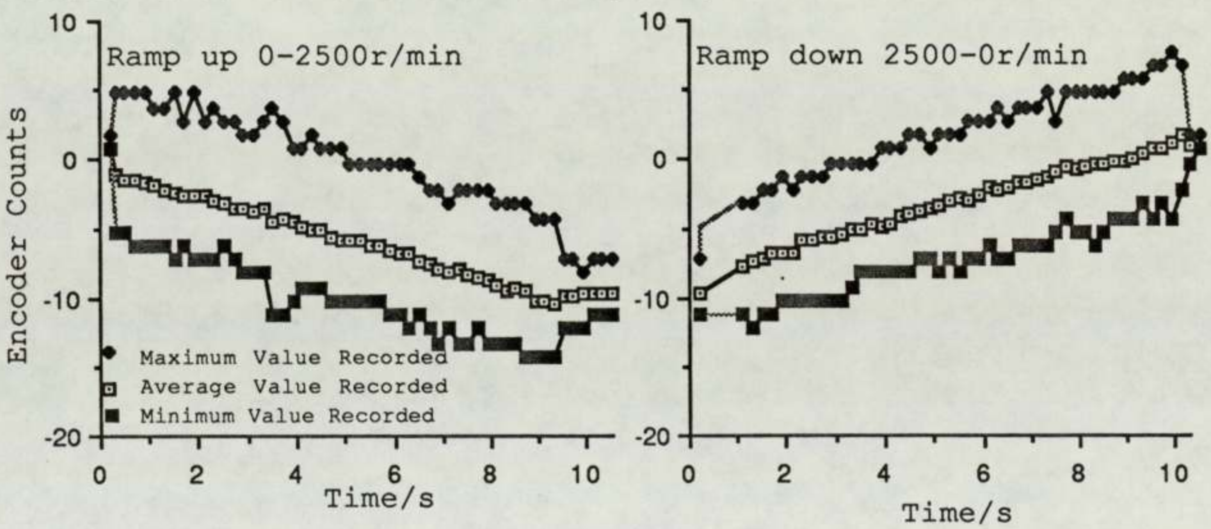


Figure 8.20 Absolute error of the matched S-6100 system during 10s ramps between 2500 r/min and 0 r/min

The absolute position error of the optimised S-6100 drive was actually higher than the S-4030 drive due to viscous effects which had been neglected in the earlier analysis. Thus matching the drives not only degraded the performance of the S-6100 drive but of the overall system. The change in average absolute error of the S-6100 system at 2500r/min was from -9 to -10.4 counts, thus the average relative error at this speed would be expected to degrade similarly. The actual recorded average relative error varied between ±3 counts with an instantaneous variation between

+15 and -9 counts which was very similar to the optimised case. The relative position error at 2500 r/min was +3.3 counts varying between +7 and +1 counts which as expected was slightly worse than the optimised case. Figure (8.21) shows the relative errors of the matched drives.

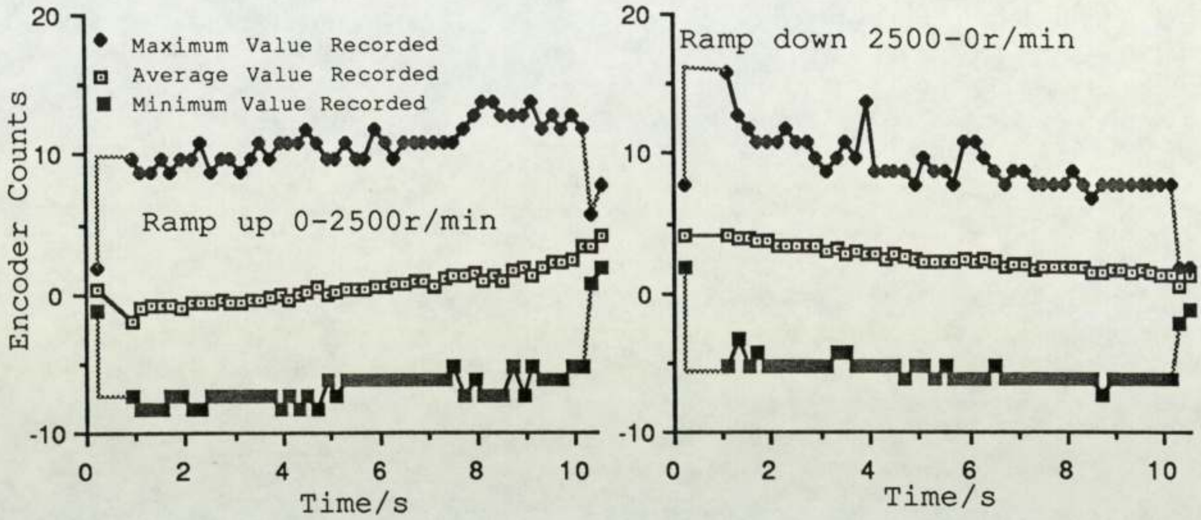


Figure 8.21 Relative error between the matched drives during 10s ramps between 2500 r/min and 0 r/min

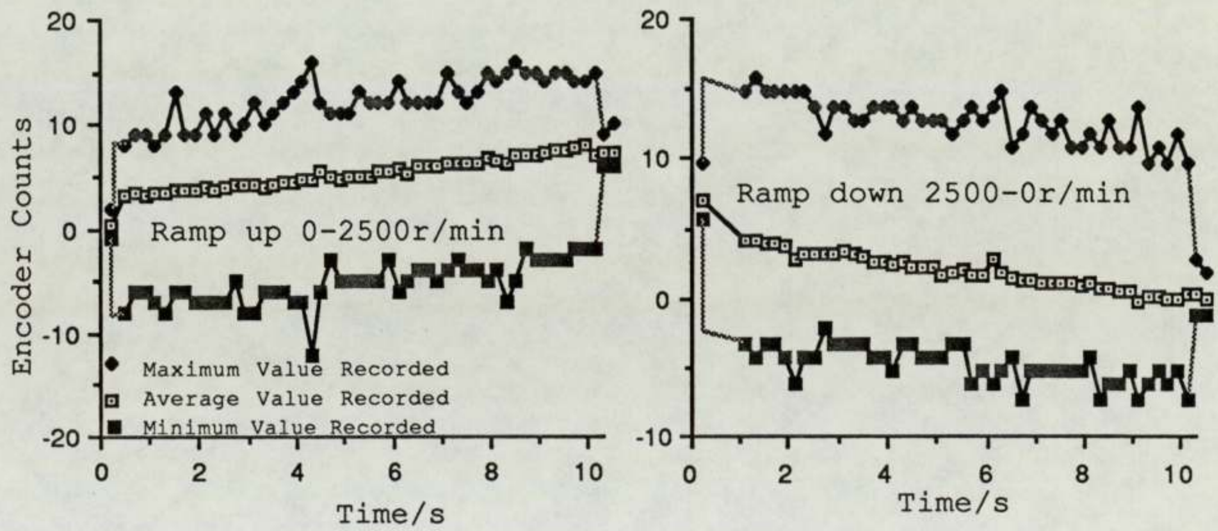


Figure 8.22 Absolute error of the matched S-6100 system during 10s ramps between 2500 r/min and 0 r/min with the master/slave topology

The matched drives were also monitored with the master/slave topology invoked. The absolute errors of the S-6100 were very large as would be expected with the average varying between +1 and -15 counts with an instantaneous variation of +15 to -31 counts, as shown in figure (8.22). This was only a few counts worse than the optimised case as would be expected as the absolute error of the S-6100 system had only been degraded by a few counts. The relative position error however was much worse with the average error varying between +7 and -1 counts, and with the instantaneous variation between +15 to -13, as shown in figure (8.23). The degradation in relative errors was because the S-6100 system could no longer follow the 20Hz perturbations of the S-4030 system.

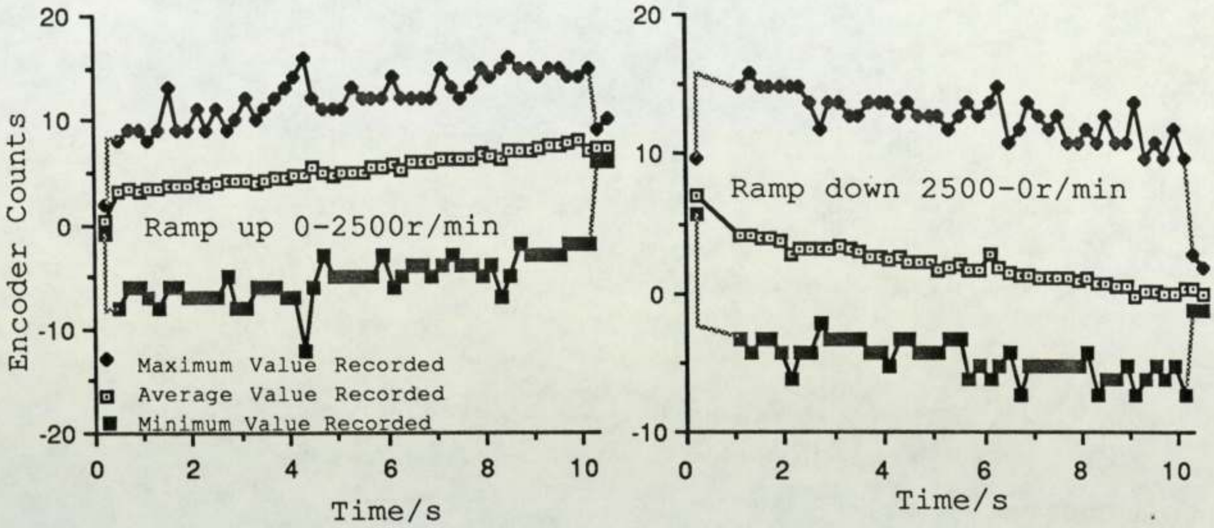


Figure 8.23 Relative error between the matched drives during 10s ramps between 2500 r/min and 0 r/min with the master/slave topology

8.6 CONCLUSIONS

A system was successfully designed and commissioned that could accurately monitor the following errors in the system. The data obtained has been presented in the simplest form possible without losing any pertinent information.

The systems were optimised such that the S-6100 had a 92 Hz (@-110°) velocity loop bandwidth and $275s^{-1}$ position loop gain, and the S-4030 had a 18 Hz (@-65°) velocity loop bandwidth and $200s^{-1}$ position loop gain. Neither drive was limited in performance by quantisation effects but the S-6100 limit was in-line with limitations due to sampling delays. The S-4030 had poorer than predicted performance due to a position loop natural resonance, exasperated by mechanical torsional effects due to the high load/ motor inertia ratio. The S-6100 was also "matched" to the S-4030 such that it had a 18 Hz (@-55°) velocity loop bandwidth and $200s^{-1}$ position loop gain.

Using the error monitor circuitry it was possible to show that the PC23 had a one pulse per 300 r/min error between "synchronised" demand pulse trains. Both drives therefore had to be driven from the same demand when investigating absolute and relative position errors. Both the optimised S-4030 and S-6100 had similar average errors varying between +2 to -9 counts, which is three times larger than the original specification. The instantaneous errors varied by different amounts due to resonance effects, such that the S-4030 varied by as much as ±11 counts from the average, whilst the S-6100 varied by ±6 counts from its average. Thus the absolute instantaneous errors were many times larger than required. The relative error between the optimised drives had an average error between +3 and -2 counts which was within specification but unfortunately due to the resonance of the S-4030, the instantaneous relative error varied between +15 and -13 counts which was unacceptable.

The relative error was improved by use of the master/ slave topology since the S-6100 had a high enough bandwidth to respond to the 20 Hz resonance of the S-4030 system. During the ramps the average error was between +4 and -2 counts, which was similar to the situation without the master/ slave topology, but the instantaneous errors where much better, varying between +8 and -4 counts which gave more than a doubling in performance. The instantaneous errors were still however larger than the 3.2 counts suggested in the specification.

The S-6100 system had its velocity loop bandwidth and position loop gain matched to that of the S-4030 system. Unfortunately the performance of the system was degraded by this topology; the average relative error varied between ±3 counts whilst the instantaneous error varied between +15 and -9. Matching was therefore not beneficial primarily because of the resonance in the S-4030 system which gave rise to high instantaneous errors. In the absence of these oscillations matching may have been beneficial but a better method is required to match the drives.

These results are worse that predicted by simulation. Simulation can therefore only be used as a guide to performance using the model developed since it was unable to take account of the resonance effect which greatly affected results.

CHAPTER 9

9.0 **CONCLUSIONS AND SUGGESTIONS FOR FURTHER WORK**

9.1 FUTURE WORK

9.1.1 CONTROLLER DESIGN

The controller used for the research project was intentionally simple in design. The Dservo system could not sample faster than 200µs as its algorithm required this amount of time to complete each cycle. The servo gains were set up using resistors upon a header chip, and it lost pulses during a direction reversal, such that the motor had to pause between reversals. The digital counters were also of too few bits such that they "toppled" if the drive accelerated at too high a rate. The controller required an external pulse train demand whereas most modern position control systems have the facility for internal profile generation.

A much more sophisticated controller would therefore be required to further the research. Ideally it would be capable of high speed computation such that a large controller algorithm could be completed in under 100µs for example. The controller gains should ideally be adjustable within the software with user access. The algorithm should also be capable of controlling velocity and position with velocity feedforward, and with the facility to add further compensation, either to enable a master/ slave relationship or to allow an external system to monitor errors from previous runs and to compensate for them by "feeding forward".

The controller algorithm should be made easy to change (written in a high level language) such that it may be adapted for differing applications, such as the modern control methods reviewed by Meshkat [48]. The inclusion of model reference control would be very useful. It would also be beneficial to change gain as a function of speed to overcome the encoder "rattle" observed during low speed operation. One modern control technique which is being increasingly applied is to switch off integral gain when operating in the saturated region to reduce overshoot, and this could be incorporated.

In order for the controller to be of industrial benefit it must be able to interface with other devices for full system integration. Thus a number of digital and analogue input/ output lines would be required, with the controller being able to interpret signals it received to alter its action. It would also be beneficial if the controller could be produced such that versions were available to interface with standard buses such a VME or PC bus.

Very high performance servo controllers are now available but they have fixed control algorithms and either accept pulsed, voltage or internal profile demands but not all three. In-line with common practise it should be possible for the controller to generate its own profile (from a user defined program) or accept an external pulse train to further increase system flexibility.

Thus it is proposed that in the future a very sophisticated controller be designed such that it is of industrial benefit, but that the controller algorithm is such that it could be easily altered to enable further academic research.

Such a sophisticated controller would require great computing power to accomplish the control within an acceptable sample period, especially for multi-axis operation. Thus a multiprocessor (possibly transputer) system or a digital signal processor (DSP) would be required.

9.1.2 FURTHER CONTROL LOOPS

The systems described thus far have used single axis controllers, with a simple master/ slave cross-couple between drives. The demonstrator project had the facility to manually alter the relation between the demand signals to the drives such that at constant velocity the relative error between the drives could be held at practically zero. Thus a slow acting (integral) control loop could be incorporated to monitor the relative error between drives and add compensation to the demand to produce a reduced relative error.

9.1.3 COMPLEX SYNCHRONISATION

The synchronisation strategies that were identified were only applied to the simple two axes case. Independently driven machines will have many more axes than this, and therefore these strategies require alteration to accommodate more complex situations. The master/ slave topologies and matching techniques will still be applicable, but a mix of the methods will probably be required.

It is probable that a multi-machine system may have an overall electronic master related to the machine cycle speed, and groups of closely coupled drives will be slaved from this. Any group of drives may have further master/ slave or matched groups of drives within it. The system could therefore be split into a hierarchy of control with electronic masters, motor masters, motor sub-masters, and motor slaves. Dudzinski [21] stated that the demand to a drive can be a function of three inputs: the master machine reference, a reference for a group of drives and signals from adjacent drives. The control becomes even more complex when considering gear ratios that will require implementation. Also for incremental motions trip-points where drives must be at specific positions or take a specific action related to another drive, must be accommodated.

Thus if independent drives are to be used to create a complete manufacturing system, research is required to investigate tight coordination and control of drive interactions together with product information and programming for machine flexibility. A specification methodology is required such that all the complex interactions between drives can be specified accurately and ambiguities can be checked. Thus a drive "compiler" is required that can take a formal specification for a drive system and produce a control stategy, and possibly the software code to produce the system.

Future systems will require great flexibility such that the machine becomes modular and the machine modules linked in different ways to produce different machine functions.

9.1.4 COMMON BUS

This particular application for independent drives required only a relatively small amount of power to accelerate the load and overcome negligible friction forces. For independent drives in other applications this is often not so, as in the application described in appendix 1. Chapter 2.0 described motor selection techniques formulated for incremental applications. Incremental applications of independent drives are very inefficient when compared to conventional mechanically linked systems as flywheels do not exist to store energy. In a conventional incremental motion, an accelerating actuator removes kinetic energy from the flywheel and when decelerating it returns it. Thus the system is energy efficient over a full cycle. There are however energy losses due to gearing, which could be compared to the heat losses in electric motors. In order for independent drives to gain acceptance with machine manufacturers they must be shown to be energy efficient. A comparison of energy losses between mechanically linked and independent systems would be beneficial. Unfortunately present drive systems draw energy from the power supply when accelerating and "dump" energy into resistors when braking. Thus present incremental drives are very energy inefficient.

Drive systems require an "electrical flywheel" of some sort. One possible method would be to pass regenerative energy back to the supply rather than dispose of it in resistor banks, Persson [55]. This would not be desirable due to corruption of the supply and injection of harmonics due to power electronic switching.

A much better solution would be to use a "common dc bus", Harmoinen [34], that is to run all drives from a common bus such that as one drive accelerates it can use the energy of a decelerating one.

If there was a net flow of energy back to the supply it could be stored in capacitor banks rather than dissipating it in resistors. Power supply units of large enough power rating or with sufficient energy storage capability are not presently available to the servo system designer. Research could be directed towards the problem, of designing supply systems to make overall independent drive schemes more energy efficient.

9.1.5 RESONANCE PROBLEMS

The limit to the S-4030 system was due to resonance problems. The resonance was due to the combination of a position loop natural resonant peak and load/ motor inertia mismatch. Reluctance torque and/or encoder digital feedback may also play a part in this phenomenon. At present most servo system vendors suggest that the maximum load/ motor inertia ratio should be in the region of ten (the S-4030 system had 43:1), and this limit will prove restrictive to the design of large independently driven machines.

The mechanism of the resonance encountered was not fully understood and it was not observed during the simulation experiments. Therefore a detailed analysis of the phenomenon is required such that it may be understood, simulated accurately, and finally such that proposals may be made as to how it may be overcome.

9.1.6 ENCODER RESOLUTION

A digital encoder introduces a backlash effect into any digital servo loop, so it is therefore desirable to have as high a resolution as possible for high performance. Whilst chapter 8.0 showed that the performance limitation of the particular systems was not influenced by quantisation as described in chapter 6.0, the trend towards reduced sample time will mean that eventually this limit will be met. For example, the S-6100 was capable of 92 Hz bandwidth with a 8000 pulse encoder and 0.5ms sampling, and the predicted limit due to quantisation was 630Hz. If the sampling approached 100μs the limit could however be due to quantisation. Thus encoder resolution must increase; it is thought that encoders with 15,000-20,000 pulses per revolution capable of 3000 r/min will be required if the trend for faster sample times continues. This will further require high speed electronics to accommodate the increased data rates.

9.1.7 ELECTRO-MAGNETIC INTERFERENCE

All the drive systems investigated emitted high levels of electro-magnetic interference (EMI), due to the PWM switching of the power electronic devices that were utilised, Kendall [40]. This led to the corruption of many analogue signals around the drives which hindered instrumentation. With ever stringent legal limits to EMI being introduced, it will become necessary to reduce the EMI emissions from drive systems. Research should be directed at this problem.

9.1.8 SAFETY

When a conventional mechanically linked system shuts down under an emergency condition all the actuators remain in synchronisation due to the "hard" links between them. However, independently driven systems may lose synchronism during fault conditions. It may be beneficial to lose synchronism as the actuators may be stopped quicker than with a mechanically linked system, but if the actuators interact a collision may occur which will be dangerous and costly. Research is required to prevent interacting actuators from clashing under abnormal situations, such as a power failure. Uninterruptable power supplies may be required to overcome power failure problems

9.1.9 SIMULATION ENHANCEMENT

The simulation experiments proved very useful in understanding the drives systems utilised during the project and as a design aid to improve performance. Simulation was still inaccurate due to inability to accurately model the resonance effects observed on the S-4030 system. The simulation model therefore requires enhancement.

Furthermore, more complex control algorithms may then be designed and tested in the simulation environment, prior to implementation on an actual system.

9.1.10 LIMIT PREDICTION ENHANCEMENT

The research presented has identified gain limitations to system performance for a simple control loop. These limitations were found to be due to quantisation and sampling delays. The methods may be enhanced to include similar limits for systems utilising derivative or integral gain terms.

Resonance effects also require further investigation such that simple calculations can be used to predict the limits imposed by this phenomenon.

9.2 CONCLUSIONS

The research has concentrated upon design techniques for accurate control of servo drives required to run at continuous high speed, and to reduce relative positional errors between two drives; many of the techniques developed were published in Seaward and Johnson [61-65]. In particular the research was aimed at retro-fitting such drives to a rod-making machine, and thereby replacing the existing mechanically linked system. This area of control has not been investigated in the past by the leading authors in the servo field who have concentrated upon incremental or point-to-point applications. The majority of drive selection methods have been aimed at incremental applications where the limit to performance is usually energy dissipation in, and therefore overheating of, motor windings. These methods were reviewed in chapter 2.0, and were successfully applied to an application described in appendix 1. Figure (2.3) shows the common incremental velocity profiles which may be utilised with lists of the common limiting factors, such as energy dissipation, maximum velocity, peak acceleration or rms current. For example the parabolic profile is associated with the least energy dissipation and the lowest peak velocity required, whereas the triangular profile required the least peak acceleration. It was concluded in-line with previous authors that the parabolic profile was the most beneficial to use in the majority of situations, but due to modern controller practise of only allowing constant acceleration zones the trapezoid is the most practically beneficial. When gearing can be used it was also shown by consideration of both energy and power rate analysis that there is an optimum gear ratio which is the square root of the load/ motor inertia ratio (in the absence of frictive effects). This leads to the principle of inertia matching.

A review of electric servo systems was carried out. Since conventional dc, trapezoidal brushless dc and sinusoidal brushless drives were all compared a rigorous method was required to ensure that similar parameters were being examined. The brushless configuration constant was developed such that dissimilar machines may be compared. The constant overcame the problem that in the simple block diagram representation of a brushed machine the torque and back emf constant were identical which was not the case for brushless systems. The torque constant of a brushless drive was shown to be the brushless configuration constant multiplied by the back emf constant of one phase. For a sinusoidal drive the brushless configuration constant was shown to be 1.5 and for a trapezoidal drive, 2. Thus by using the brushless configuration constant different drive genre could be compared, as long as the constants used were the per phase motor parameters, and the currents considered were peak values. The common base on which to measure performance was chosen as continuous torque at a quoted speed, which defined a given power. A drive comparison was carried out upon seven different drive systems. Standard measures of drive comparison were used such as damping factor, electrical and mechanical time constants, torque to inertia ratio and power rate. The simplest graph to use to ascertain performance was that of power rate against torque to inertia ratio. The drives comparison showed the superior performance of brushless systems over brushed ones. Also the drive chosen for the application had slightly poorer performance than some of the newer systems, showing the rapid increase in drive performance over the past few years.

A review of common controller topologies was carried out in chapter 4.0. By use of simulation results it was shown that the simplest controller that could achieve the specification would have purely proportional velocity and position loops with velocity feedforward to reduce the position following errors. The master/ slave, matched and error feedforward schemes were also proposed as possible ways to increase performance, in terms of relative error between drives. It was shown that for the master/ slave topology to be of benefit there must exist at least a factor of two between the absolute following errors of the individual drives.

By use of simulation in chapter 5.0 it was shown that in order to reach the specification laid down in section (1.5) a velocity loop bandwidth in the order of 90Hz and position loop gain of $200s^{-1}$ would be required. It was also shown by use of conventional control theory that systems were limited in performance due to sampling delays; the velocity loop gain cross-over frequency had to be less than one quarter of the sample frequency to achieve stability, less than one tenth for stable control and in the order of one twentieth to close an adequately stiff position loop. Also due to sampling delays it was shown that the position loop gain must be less than the 0.03 times the sample frequency (in $rads^{-1}$).

The analysis of a simple representation of a servo loop used to reach the above conclusions was expanded to include actual systems, by use of simulation. Chapter 5.0 had a case study, where simulation was used to predict the performance of a servo system from its open-loop frequency response. The results matched experimental findings well.

A further limit to performance was due to quantisation, since there comes a point were gain cannot be increased further as one pulse of error (from the encoder feedback) produces a current demand outside the continuous region. An expression was derived which showed that the maximum velocity loop bandwidth due to this phenomenon was $T.\Delta T.p/2\pi J_T$, and the position loop gain was limited to the reciprocal of sample period.

Chapter 6.0 investigated the effects of the mechanical load on a servo system. It was concluded that the mechanical system had to be very stiff with negligible backlash, and taper lock coupling with keyways were suggested as the most applicable types of coupling for these drives. It was shown that in the presence of backlash the velocity loop bandwidth is reduced in the ratio of motor to total (motor and load combined) inertia. It was therefore shown that with present technology the original Molins specification could not be met due to the high inertia of the cut-off. A block diagram to show the effects of torsional resonance was presented, and in-line with common practise it was suggested that the load/ motor inertia ratio should be less than ten to prevent this from degrading the control loop.

The laboratory work carried out during the research project concentrated upon two Electrocraft Bru-500 drive systems with approximately $0.01 kgm^2$ loads. The velocity loop performance of the S-6100 system was increased from the original 17Hz (@-100°) to 18Hz (@-65°) by a software update within the Bru-500 system suggested by this research using simulation evidence. By use of an external controller the performance was further increased to 38Hz (@-140°) which was a doubling on the previous figures. This performance was further increased by identifying phase shifts in the input to the system, with the aid of simulation, and by-passing it; this modified system increased performance to 92Hz (@-110°) which is more than fourfold the original performance. Thus the power of simulation as a design tool could be observed, as it enabled greater understanding of a system, and design concepts could be tested at a terminal rather than by building expensive demonstration rigs. The performance of the S-4030 system was disappointing, due to a position loop natural resonance exasperated by mechanical resonance problems. Thus, whilst the limit to performance of the S-6100 system fitted well with predictions from conventional control theory analysis of sampling delays, the limit to performance of the S-4030 system was not predicted by any of the analysis carried out.

Using the two drive systems, the Dservo controllers and a PC23 indexer card within an IBM PC clone a demonstrator rig was successfully commissioned. Specialised error monitor circuitry was developed to measure the errors within the system, such that the results presented in chapter 8.0 were obtained. It was therefore possible to show that during acceleration and deceleration ramps the optimised system could hold average relative error between the drives at +3 to -2 encoder counts which was within specification. Unfortunately due to mechanical resonance of the S-4030 system the instantaneous error was only within +15 and -13 counts, which was outside the specification. At constant velocity however the error could be maintained between +3 and -1 counts which was within the specification; a slower acceleration period could therefore be used to reduce errors during the ramps. By use of the master/ slave topology the instantaneous errors during the ramp were reduced to between +8 and -4 counts, since the S-6100 system was able to follow the 20Hz resonance disturbances in the S-4030 system. Due to the resonance problems the matched system had poorer performance than the optimised system. Therefore matching should only be attempted upon systems without unpredictable disturbances. It was also discovered that the indexer used exhibited an error in demand signals of one pulse per 300r/min. Thus, the electronics had poorer performance than the master/ slave topology!

This research has therefore presented a methodology for selecting drive systems for use with independently driven machine systems. Since the particular application required minimal following error between drives at a high speed, the selection techniques were derived due to dynamic performance of the system rather than upon energy consumption. Simulation has been used to obtain approximate figures for velocity loop bandwidth and position loop gain, and limits in these parameters due to sampling delays, phase shifts and quantisation have been defined. Methods for comparing and simulating drive systems have been presented. Single axis controllers and local control schemes used to reduce relative errors were also introduced, and practical results presented for a two axis case.

Thus this research has investigated some of the problems associated with the introduction of flexible independently driven actuators, such that, machinery can be quickly reconfigured to accommodate short production runs and product changes. The methods are being adopted within Molins and early results show significant improvements over traditional design methods for high speed machinery.

REFERENCES

[1] Acsl, 1981, "Advanced continuous simulation language user guide/reference manual", Mitchell & Gauthier Associates.
[2] Al-Anbuky.A.H., Al-Aubaidy.K.M., 1984, "Microprocessor based controller for the control of position and speed of a dc servo motor", Proc. IECON, 22-26 Oct., v1, p595-9.
[3] Arnold.F., 1984, "Power rate is the best indicator of transducer efficiency", Proc. International Conf. on Electrical Machines, Lausanne, Switzerland, 18-21st Sept., p177-180.
[4] Arnold F., Floresta J.G., 1984, "Power rate a most important figure of merit for the incremental motion designer", Proc. 13th Ann. Symp. on Incremental Motion Control and Devices, p11-18.
[5] Barber.N.T., 1987, "High performance and precision drives, servos torquers and positioners - A review of the state of the art", Drives/Motors/Controls Tutorial update, sess. 4, 12th Nov.
[6] Barber.N.T., 1985, "Coming to terms with brushless servo drives", Electric Drives and Controls, April/May, p25-29.
[7] Barber.N.T., 1984, "Benefits of using brushless drives on high performance robots", 7th British Robot Association Ann. Conf., May, p95-105.
[8] Bartlett.P.M., Shankwitz.H., 1983 "The specification of brushless dc motors as servomotors", Powerconversion International, v9, n8, Sept., p10-17.
[9] Berger.M.A., 1978, "Six steps to an optimum servomotor", Machine design, v50, n8, April 6, p248-252.
[10] Blank.G.L., Wrobel.J.J., 1985, "Simulation of a velocity controller for a brushless dc motor", Computer simulation summer conference, Chicago, Il, USA, 22-24th July, p202-205.
[11] Bolton.H.R., Ashen.R.A., 1984, "Influence of motor design and feed current waveform on torque ripple in brushless dc drives", IEE Proceedings, v131, PtB, n3, May, p82-90.
[12] Bose.B.K., 1982, "Adjustable speed ac drives - A technology status review", Proc. IEEE, v70, n2, Feb., p.116-135.
[13] Brosnan.M., Barber.N.T., 1985, "Rare-earth high performance brushless drives & their application", Power conversion and intelligent motion, v11, n10, Nov., p.20,22,24-26.
[14] Brown.E.M., 1988, "The best linear servo design - the sort that works", Drives and controls one day seminar, Coventry, 10th March, pD1-D7.
[15] Brown.M., Moore.D., 1982, "Brushless dc or inverter motor drives: a comparison of attributes", Proc. of 3rd Motorcon, Sept., p111-123.
[16] Carlisle.B.H., 1986, "Rare earth motors shed their exotic image", Machine Design, v58, n2, Jan., p18-26.
[17] Carlisle.B.H., 1985, "Ac drives move into dc territory", Machine Design, v57, n10, May, p61-64.
[18] Christensen.L., 1988, "Permanent magnet brushless servo motor development", MMPA Permanent magnet Conf., Dayton Ohio, 26-27th June.
[19] Compumotor, "PC23 Manual Number 88-007015-03".
[20] Dorf.R.C., 1980, "Modern control systems", Addison-Wesley, 3rd edition.
[21] Dudzinski.M., 1985, "Digital motor drives keep complex paper machines humming", I&CS Ind & process control mag. [USA], v58, n9, Aug., p31-34.
[22] Eapen.K., Venkatesan.K., 1981, "Phase-locked loop dc motor drive with improved transient performance", IEEE Trans., vIECI28, n4, Nov., p347-352.

[23] Eapen.K., Venkatesan.K., 1980, "Stabilisation of pll controlled dc motor drive", Journ. Inst. Eng. [India], v61, ptEL2, Oct., p59-65.
[24] Eapen.K., Venkatesan.K., Gupta.S.C, 1979, "A pll dc motor control system", Journ. Inst. Eng. [India], v59, ptEL6, June, p315.
[25] Electrocraft, 1980, "Dc motors, speed controls, servo systems. An Engineering Handbook", 5th Edition Published by Electrocraft.
[26] Elkington.B., 1975, "Application of dc servo motors in incremental motion control", Proc. Symp. on Incremental Motion Control and Devices, pU1-U20.
[27] Fenney.L., 1986, "Selection of a brushless dc servomotor for the 3rd transfer mechanism", Aston university internal report 10th April.
[28] Firoozian.R.,Brimson.J.R., Foster.K., 1983, "The comparison of performance of servo feed-drive systems", IMechE C301/83.
[29] Fleisher.W.A., 1989, "Criteria for selecting dc motors", Drives and Controls, March, p52-56.
[30] Floresta.J.G., 1985, "Designing incremental motion systems with power rate", Proc. 14th Ann. Symp. on Incremental Motion Control and devices.
[31] Geiger.D.F., 1981, "Phaselock loops for dc motor speed control", Wiley-Interscience.
[32] Gipper.P.D., 1988, "Accurate torque-speed performance prediction for brushless DC motors", Proc. IEEE National Aerospace and Electronics Conf., 23-27 May, p1092-7.
[33] Hall.J., 1982, "Motors keep pace with changing technology", Instruments & Control Systems (USA], v55, n9, Sept, p28-33.
[34] Harmoinen.M., 1988, "Common dc bus in variable speed ac drives", Power Electronics and Variable Speed Drives Conference, 13-15th July, London, IEE Conf. Publication 291.
[35] Hindmarsh.J., 1977, "Electrical machines and their applications", 3rd edition, Pergamon.
[36] Hopper.E., 1985, "Choosing the right brushless drive for your application", Motorcon, April, p361-370.
[37] Horner.G.R., Lacey.R.J., 1982, "High-performance brushless pm motors for robotics and actuator applications", Proc. Drives/ Motors/ Controls Conf., June, p91-99.
[38] Inland, "Technical manual BHT", Issue 1 from Inland, Bedford, England.
[39] Jones.D., 1985, "Motor magnetics design changes in higher energy rare-earth pm dc motors", Power conversion and intelligent motion, v11, n12, Dec., p28-37.
[40] Kendall.P.G., 1984, "The electrical environment of variable speed drives", Conf. on power electronics & variable speed drives, May, p154-157.
[41] Klein.F.N., Kenyon.M.E., 1984, "Permanent magnet dc motors design criteria and operation advantages", IEEE Trans., vIA20, n6, Dec., p1525-1531.
[42] Le-Huy.H., 1979, "A synchronous thyristorised dc motor drive", IEEE Trans., vIA15, n2, March, p183-8.
[43] Leighton.N.J., Jones.B., Fletcher.H., 1988, "Precision press feed system using digitally controlled drives", IMechE, One day conference on High Speed Machinery, 24th Nov., p3-9.
[44] Lim.P.Y., Kangsanant.T., 1988, "Computer - integrated brushless motion control system with digital position controller", Power Electronics and Variable Speed Drives Conference, 13-15th July, London, IEE Conf. Publication 291, p324-327.

[45] Margaris.N., Petridis.V., 1980, "Pll control schemes of linear and non-linear dc motors", Int. Conf. on Electrical Machines, Sept., p586-93.
[46] Mazurkiewicz.J., 1983, "Analysis of new compact brushless versus pancake motors", Motorcon, p521-531.
[47] McLaren.S.G., 1985, "Pll control of ac and dc motors", Elektron, S.Africa, v2, n7, July, p18-21.
[48] Meshkat.S., 1987, "Advanced motion control algorithms: feasibility and performance improvement using digital signal processors", Motorcon, Sept., p1-21.
[49] Millar.R.A., 1968, "Digital control of shaft & speed & position", IEEE Spectrum, Jan., p90-95.
[50] Moore.A.W., 1973, "Phase locked loops for motor speed control", IEEE Spectrum, April, p61-67.
[51] Nehl.T.W., Hijazi.T.M., Demerdash.N.A., McHale.T.L., 1985, "Automatic formulation of models for simulation of the dynamic performance of electronically commutated dc machines & impact of winding inductances and other parameters on the design & performance of brushless motors", IEEE Trans., vPAS104, n8, Aug., p2206-2222.
[52] Newton.G.C., 1981, "Selecting the optimum servo-motor for incremental positioning applications", Proc. 10th Symp. on Incremental Motion Control and Devices, May, p353-362.
[53] Persson.E.K., Meshkat.S., 1985, "Brushless servo system with expanded torque speed operating range", Motorcon, April, p96-107.
[54] Persson.E.K., 1983, "Brushless motor technology", Motorcon, April, p129-140.
[55] Persson.E.K., 1981, "Brushless dc motors: a review of the state of the art", Motorcon, June, p1A1.
[56] Persson.E.K., 1976, "Brushless low-inertia motors", Conf. on Small Electrical Machines, March, p19-22.
[57] Persson.E.K., Buric.M., 1976, "Mathematical modelling and simulation of high performance brushless dc motors", 5th Ann. Symp. on Incremental Motion and Control, pw-1, w-8.
[58] Powell.B., "Direct drive dc servo motors for gearless motion control", Motornetics Corporation, Santa Rosa, California, USA.
[59] Prasad.E.S.N., Dubey.G.K., Prabdu.S.S., 1985, "High-performance dc motor drive with pll regulation", IEEE Trans., vIA21, n1, Feb., p192-201.
[60] Rees Jones.J., 1988, "Some dynamic issues in programmable continuous and logically controlled motions in machines", IMechE Seminar on "The Engineering contribution to Food Packaging in the 90's".
[61] Seaward.D.R., Johnson.R.C., 1989, "Replacement of mechanical transmissions by synchronised variable electrical speed drives", Power electronics and variable-speed drives conference, Sept., UK.
[62] Seaward.D.R., Johnson.R.C., 1989, "Digital simulation and model validation of brushless dc machines", IEE colloquium on Model validation for control system design and sim., p6/1-5.
[63] Seaward.D.R., Johnson.R.C., 1988, "Technology transfer from academia to industry of a phase synchronised drives project", IMechE, One day conference on High Speed Machinery, 24th Nov., p11-18.
[64] Seaward.D.R., Johnson.R.C., 1988, "On the simulation of brushless drives", University Power Engineering Conference, Sept., UK.

[65] Seaward.D.R., Johnson.R.C., 1988, "Considerations for directly coupled phase synchronised drives", University Power Engineering Conference, Sept., UK.
[66] Shaderma.M., 1983, "Wide range digital pls system for dc motors", University of Wales, Swansea, Thesis, Dec..
[67] Smithgall.D.H., 1975, "A pll motor control system", IEEE Trans., vIECI22, Nov., p487-490.
[68] Tal.J., 1989, "Motion Control applications", available from Galil Motion Control Inc. CA. 94303.
[69] Tal.J., 1987, "Phase-locked servo systems produce wide speed range and high resolution", Proc. Symp. on Incremental Motion Control Systems and Devices, May, p229-35.
[70] Tal.J., 1985, "Step by step procedure for designing servo systems", Proc. of the first IMS Conference, Chicago, Il, USA, 21-25th Oct., p164-72.
[71] Tal.J., 1984, "Motion control by microprocessors", available from Galil Motion Control Inc. CA. 94303.
[72] Tal.J., Baron.W., 1984, "Motion control for automation and robotics", Robotics age, Dec., p8-11.
[73] Tal.J., 1977, "Speed control by phase locked servo systems-New possibilities and limitations", IEEE Trans., vIECI24, n1, Feb., p118-125.
[74] Tal.J., 1977, "Modeling and stability analysis of phase-locked servo systems", Proc. of the 10th International Motorcon, Hannover, 1-3 April, p54-67.
[75] Tal.J., 1975, "Pll servo-systems - A new approach to speed control", Proc. symp. on Incremental Motion Control Systems and Devices, April, pAA1-5.
[76] Tal.J., 1975, "Synchronisation characteristics of phase locked servo systems", Conf. of electrical and electronic engineers in Israel, April, pA-3-3/1-9.
[77] Tomasek.J., 1986, "Electro mechanical performance data for sine-wave brushless servo systems", Motorcon, Oct., p16-29.
[78] Tomasek.J., 1986, "Basic performance specifications and ratings for sine-wave brushless servo systems", Proc. 15th Symp. on Incremental Motion Control and Devices, 3-5th June, p27-37.
[79] Tomasek.J., 1985, "Velocity & position feedback in brushless dc servo systems", Power conversion & intelligent motion, v11, n8, Aug, p55-60.
[80] Tomasek.J., 1983, "Analysis of torque-speed performance limits in brushless dc servo systems", Symp. on Incremental Motion Control and Devices, p1-9.
[81] Tomasek.J., 1983, "An optimum performance match of dc servo motors and amplifiers", Motorcon, April, p1-15.
[82] Tomasek.J., 1981, "Dictionary of motion control calculations", Symp. on Incremental Motion Control and Devices, p1-12.
[83] Tomasek.J., 1980, "Design criteria & results in high performance brushless dc servo technology", Symp. on Incremental Motion Control and Devices, June, p289-300.
[84] Tomasek.J., 1979, "Brushless servo motor amplifier optimisation", Symp. on Incremental Motion Control and Devices,, May, p279-291.
[85] Vaidya.J.G., 1982, "Optimisation of brushless dc motor design", Drives and Controls International, June, p20-22,31,51.
[86] Zimmermann.P., Bosch.R, 1982, "Electronically commutated dc feed drives for machine tools", Drives and Controls International, Oct., p13-19.

APPENDIX 1

A1.0 FEASIBILITY STUDY FOR DIRECT DRIVE OF A RAPIER MECHANISM IN CARPET MAKING

A1.1 INTRODUCTION

This appendix reviews the feasibility study carried out at Aston University for Brintons Ltd. The study investigated the possibility of applying brushless dc drives, to drive the rapier and needle head mechanism within a 5m carpet loom.

The original Brintons' specification is outlined and the inertia identification techniques are given for the equipment loaned to Aston University.

A large number of calculations are presented to support the arguments for and against servo drive selection.

Finally laboratory results are presented to show typical performance and its relation to the earlier theoretical calculations.

A1.2.0 THE BRINTONS' SPECIFICATION

The traditional Brintons' carpet making looms utilise a single continuously running electrical machine which controls the various machine functions/ actuators via complex mechanisms. The rapier mechanism carries yarn across the length of the loom (up to 5m) and was been selected as a possible application of independent servo drives.

The rapier was a carbon fibre reinforced strip over 5m in length with holes punched in it to accept the teeth of the driving wheel which was made from a light alloy. To the tip of the rapier was attached a needle head to carry the yarn, as shown in figure (A1.1).

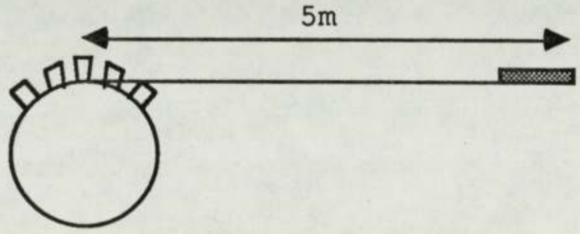


Figure A1.1 The full loom rapier system

It was possible to use a two rapier system with a 2.5m throw driving from each end as in figure (A1.2), with the yarn being transferred in the middle of the carpet.

The diameter of the driving wheel altered the effective gear ratio and the one loaned by Brintons had a 0.39m diameter.

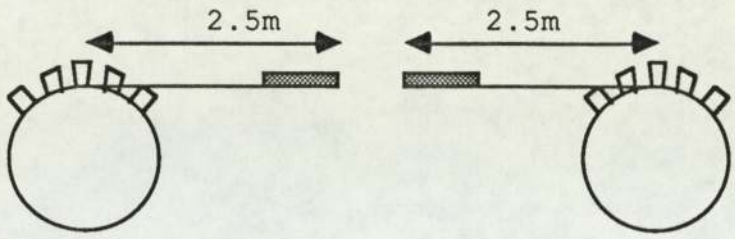


Figure A1.2 The half loom rapier system

A1.2.1 INERTIA/WEIGHT SPECIFICATION

One toothed wheel as supplied
Needle head weight:- 1/20 lb/ft or $0.07441 kgm^{-1}$
Rapier rod weight:- 1/4 lb or 0.113kg
Rapier head moves 5+0.25 = 5.25m for the full throw system
Rapier head moves 2.5+0.25 = 2.75m for the half throw system

A1.2.2 REQUIRED MOVEMENT

The motion required was to propel the rapier head through the carpet, dwell for 0.166s to enable another operation and then return back to the starting position. The dwell could be after a move of 5m for a full carpet traverse or 2.5m with two rapiers changing over in the middle; 0.25 overshoot was required.

Positional accuracy was not defined and predictable overshoot would not be a problem. Also the move profile was not important. The time to make the move determined essentially the loom speed in "picks per minute". A figure suggested by Brintons to aim for was 80 picks per minute such that the total move time was 0.75s and the time during which movement took place was 0.584s. Also since a full stroke system was preferential to half stroke a compromise speed of 60 picks/minute may be allowed.

A1.3.0 IDENTIFICATION OF THE LOAD

An accurate knowledge of the load is a prerequisite for any design and selection techniques used with servo systems. Thus the inertia of the toothed wheel supplied by Brintons and the taper-lock coupling used to couple the load to the motor were found. The inertia of the rapier and needle were also calculated from data supplied. The torque to overcome viscous damping and friction were assumed negligible compared to that required to accelerate the inertial load.

Two methods for identifying the inertia were used. One was to calculate the inertia directly from measurement whilst the second used the experimental tri-filar suspension method.

The mass of the components was measured as:-
Machined coupling = 0.838kg
Brintons' toothed wheel = 0.884kg
5x(studs, washers, nuts) = 0.050kg

A1.3.1 METHOD 1: CALCULATION

Considering the tooth wheel as:-
One large disc diameter 0.3900m,
less 6 holes diameter 0.1150m pitch circle radius 0.1225m
less 6 holes diameter 0.4000m pitch circle radius 0.1660m
less 12 holes diameter 0.0160m pitch circle radius 0.1780m
less 6 holes diameter 0.0160m pitch circle radius 0.0720m
less 6 holes diameter 0.0080m pitch circle radius 0.0540m
less 5 holes diameter 0.0070m pitch circle radius 0.0420m
less 1 holes diameter 0.0600m in the centre

Thus the density of the disc, ρ = 20.732kgm^{-2}, and the total inertia was calculated as 0.0195kgm^2

Similarly the coupling had a calculated inertia of 0.00103kgm^2 and the fittings contributed 0.000097kgm^2

A1.3.2 METHOD2: TRI-FILAR SUSPENSION

Using this standard method for calculating inertia:-

$J=Mt^2a^2g/4\pi^2l$, which for the test a=0.19m, l=0.86m, g=9.81ms^{-2}

Sets of 20 oscillations were used for each reading and an average "t" obtained.

For the toothed wheel alone, t=1.435s, therefore J=0.01899kgm^2 (cf.0.0195).

For the total load, t=1.065s, therefore J=0.02096kgm^2 (cf.0.02063).

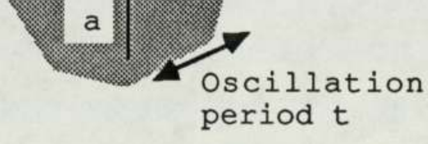


Figure A1.3 The tri-filar suspension method

A1.3.3 CALCULATION OF RAPIER AND NEEDLE INERTIA

From the data supplied by Brintons the rapier was assumed to be 1/20 lb/ft or $0.113 kgm^{-1}$. Therefore a 5.25m rapier and needle weighed 0.485kg and a 2.75m rapier and needle weighs 0.299kg.

Taking the radius of rotation of the rapier from the outside diameter of the toothed wheel (worst case = 0.398m) gives the:-

Inertia due to a 5.25m rapier and needle=mr^2=$0.019 kgm^2$
Inertia due to a 2.75m rapier and needle=mr^2=$0.012 kgm^2$

A1.3.4 CONCLUSION

Thus the load used in calculations was $0.021 kgm^2$, for the wheel, $0.019 kgm^2$ for the full stroke rapier and $0.012 kgm^2$ for the half stroke one.

A1.4.0 IDENTIFICATION OF PERFORMANCE LIMITS

Figure (2.3) lists the common velocity profiles with associated standard formulae that were used. The performance figures used for the Electrocraft Bru-500 system are listed in table (A1.1) and (A1.2).

Motor Module	Drive	Top Speed r/min	Rotor Inertia kgm^2	Cont Torq Nm	Peak Torq Nm	Torque Constant Nm/A	Resistance Ω
S-6100	DM-50	3000	0.0014	11.3	24	0.65	0.44
S-6200	DM-100	3000	0.0024	22.6	54	0.62	0.18
S-8350	DM-100	2000	0.0068	40	80	0.95	0.12
S-8500	DM-150	2000	0.0100	55	125	0.99	0.09

Table A1.1 Bru-500 Performance

The torques reduce with speed, such that the above figures are at stall whilst those below are at top speed. Other figures for specific speeds have been used in the later analysis and these were taken directly from the motors' torque speed curves.

Motor	Drive Module	Cont Torq Nm	Peak Torq Nm
S-6100	DM-50	7.5	15.5
S-6200	DM-100	9.8	33
S-8350	DM-100	19	60
S-8500	DM-150	23	82

Table A1.2 Bru-500 High speed torque

A1.4.1 SPEED LIMITATIONS

Speed limitations of the motor would inevitably effect the picks/ minute ability of the loom. Whilst the motors may be able to run at speeds in excess of those quoted by Electrocraft the torque available from the systems decays rapidly above the quoted speed. This would limit the accelerating ability and so little would be gained from over-speeding the motors.

For the triangular motion the maximum velocity=$2Ø_m/t_c$, which was set at 2000 r/min. For the supplied wheel the step was 26.9 radians which resulted in a time, t_c, of 0.26s. Therefore the maximum loom speed was 60/(2x0.26 +0.166) or 88 picks per minute.

Similarly for other profiles and speed, the maximum loom speed can be calculated as summarised in table (A1.3) for the triangular and trapezoidal velocity profiles, for the top speed of the S-6000 and S-8000 Bru-500 ranges, that is 3000 and 2000r/min respectively. It is also given for the full and half loom movements.

	Full Loom	Half Loom (quoted in picks/min)
Trapezoid		
V=2000r/min	165	227
V=3000r/min	202	259
Triangular		
V=2000r/min	87	140
V=3000r/min	117	176

Table A1.3 Motor speed as a limit to machine speed

NB. Reduction in the toothed wheel diameter would also reduce these machine speeds.

For this particular application the form of velocity profile was not specified, nor would it directly alter the loom performance.

A1.4.2 TORQUE LIMITATIONS

A number of different configurations will now be considered.

A1.4.2.1 80 picks/minute with the existing wheel rapier and head, for the full stroke motion

$Ø_m$=26.9 radians for a 5.25m stroke

$$T_{rms}=\sqrt{\frac{1}{\text{Total cycle time}}\sum\frac{K_p J^2 Ø^2}{t_c^3}}$$

There are three moves, in, dwell and out. Cycle time = 0.75s, with t_c=0.292s, dwell=0.166s, therefore for this case:-

$$T_{rms}=J\emptyset\sqrt{\frac{2K_p}{t_c^3 t_{total}}}$$

A1.4.2.1.1 <u>For the S-6200 motor</u>

J_{total}=0.0424kgm^2

T_{rms} =0.0424x26.9√(2x16)/(0.292^{3}x0.75) (for the triangular profile)

= 47Nm (with a required top speed of 1760r/min)

Thus the S-6200 (continuous torque=22.6Nm) could not meet this specification.

A1.4.2.1.2 <u>For largest S-8500 motor</u>

$\emptyset_m$=26.9radians

J_{total}=0.050kgm^2

For 80 picks/minute for a trapezoidal profile

T_{rms} =0.05x26.9√(2x13.5)/(0.292^{3}x0.75) (for the triangular profile)

= 51Nm (with a required top speed of 1760r/min)

The S-8500 (continuous torque=55Nm) would just be capable of this specification, but no account for torque roll-off at high speed has been made since at 1760r/min the continuous rating is 33Nm.

A1.4.2.2.1 <u>Half stroke with the S-6200 motor and a trapezoidal profile</u>

$\emptyset_m$=14.1radians (2.75m)

J_{total}=0.0354kgm^2

For 80 picks/minute for a trapezoidal profile

T_{rms} =0.0354x14.1√(2x13.5)/(0.292^{3}x0.75)

= 18.9Nm (The motor could supply 22.6Nm at the top speed of 691r/min and the peak torque was 26.3Nm which was less than the 54.2Nm peak rating of the machine)

Also by reverse working the peak machine speed was 87 picks/minute with the peak continuous torque of an S-6200 of 22.6Nm.

Thus the S-6200 could meet the half stroke specification, but with little room for increasing machine speeds.

A1.4.2.2.2 Half stroke with the S-8350 motor and a trapezoidal profile

$\O_m$=14.1radians (2.75m)

J_{total}=0.0398kgm^2

For 80 picks/minute for a trapezoidal profile

T_{rms} =0.0398x14.1$\sqrt{}$(2x13.5)/(0.292^{3}x0.75)

= 21.3Nm (with a required top speed of 691r/minute and the peak torque is 29.6 Nm)

This may be met but by reverse working (using peak velocity of 2000r/min) with t_c=0.101s which gave t_{total}=0.368s and an T_{rms}=149Nm, which showed that torque availability was the limiting factor.

T_{rms}=39.5Nm gives t_c=0.209 or 103 picks/minute with a peak torque of 57.8Nm which was within limits and a peak speed of 966r/min. This case cannot be met though because at 966r/min the motor can only supply a T_{rms} of 29Nm. This value gave rise to a revised value of t_c of 0.248s or 91 picks/minute, which may be met at a speed of 814 r/min and peak torque of 41Nm.

A1.4.2.2.3 Half stroke with the S-8500 motor and a trapezoidal profile

$\O_m$=14.1radians (2.75m)

J_{total}=0.043kgm^2

This motor would be able to meet the specification so setting

T_{rms} =55Nm=0.043x14.1$\sqrt{}$(2x13.5)/(t_c^3(2t_c+0.166)

which gave t_c=0.183s or 113 picks/minute with a peak torque of 81Nm and peak velocity of 1104r/min. Which again cannot be met due to reduction in continuous torque rating with speed. A re-working gave a more appropriate figure for t_c of 0.205s or 104 picks/minute, with T_{rms} =45Nm, peak torque of 64Nm and peak speed of 985r/min which can be met.

A1.4.2.3 The reduced 60 picks/minute specification

It was suggested by Brintons that a full stroke move at 60 picks/minute may be acceptable.

This may be met by both the S-8350 and S-8500 motors. For the S-8350:-

$\varnothing_m$=26.9radians (2.75m)

J_{total}=0.0468kgm^2

t_c= 0.417s

$$T_{rms} = 0.0468 \times 26.9\sqrt{((2 \times 13.5)/(0.417^3 \times 1))}$$

$$= 24.3\text{Nm.}$$

The peak torque was 32.6Nm at a peak speed of 924r/min which was well within the motors limitations.

A1.4.3 CONCLUSIONS

Throughout the analysis the rms torque required by the load has been the limiting factor for this application rather than peak velocity or peak torque. Thus the velocity profile should be parabolic but the trapezoidal has been the focus of analysis since it was easier to apply to practical situations.

The full stroke 80 picks/minute specification cannot be met even by the S-8500 machine. The half stroke 80 picks/minute specification can be met by the S-6200 machine or larger. The limiting machine speed were as follows:-

S-6200 - 87 picks/minute
S-8350 - 91 picks/minute
S-8500 - 104 picks/minute

Note: Load inertia reduction would allow increased machine speeds

A1.5 THE EFFECT OF REDUCING THE TOOTHED WHEEL DIAMETER

The major contribution from inertia was the toothed wheel, which if reduced in diameter would substantially reduce overall inertia, but unfortunately this would necessitate higher motor running speeds. Thus an iterative procedure was developed to identify when the machine speed limits due to rms torque and peak motor running speed were at the same point.

This analysis showed that for the S-8500 motor following a trapezoidal half stroke profile the limit was approximately 130 picks/minute. No account for loss of torque with speed was made.

Inertia of the toothed disc was assumed to have a quartic relationship with diameter as with a solid disc.

The diameter was found to be 0.275m by trial and error.

$$\text{Modified inertia} = \underset{\text{rapier+needle}}{0.299\times0.275^2/4} + \underset{\text{toothed wheel}}{0.021\times(0.275/0.39)^4} + \underset{\text{S-8500}}{0.01}$$
$$= 0.0208\,\text{kgm}^2$$

This assumed that a 0.275m diameter disc could be made with an inertia of $0.0052\,\text{kgm}^2$

$\text{Ø}_m = 2.75/0.1375 = 20$ rads

Thus for a maximum speed of 2000r/min, $t_c=0.143$ or 132 picks/minute, and if the peak rms torque was 55Nm, $t_c=0.149$s or 129 picks/minute (peak torque=84.3 Nm and peak velocity required was 1922r/min).

A1.6.0 LABORATORY WORK

A1.6.1 VELOCITY LOOP STABILISATION

The laboratory tests were conducted to show that the motors would perform as predicted earlier in section (A1.4).

The highest power servo system available at Aston University was the Electrocraft S-6200 motor with a DM-100 drive amplifier. The velocity control by a microprocessor was internal to the amplifier and the software version used had proportional and integral gain settings. A pure proportional loop was to be implemented, such that only one main variable required selection.

The position loop was closed using the Electrocraft PROPOS-E position controller because of its ease of use when prototyping. The PROPOS-E was a stand-alone microprocessor system which generated its own position demand from within user defined software: it could interface to other microprocessors, switches or PLCs to integrate within "machine wide" control systems.

The scale factor was set at 300 r/min per volt such that the maximum output of the PROPOS-E (±10V) caused a demand for the maximum velocity (±3000 r/min).

The input filter was set at 300 Hz from experience to negate the effects of digital sampling and noise in the system.

The integral gain was set at zero to enhance the position loop performance.

The velocity loop gain was set at 2000, which was the maximum available from the system, due to an internal software limit. The loop as described above was frequency response analysed to show a 45-50Hz (@-100°) bandwidth as shown in figure (A1.4).

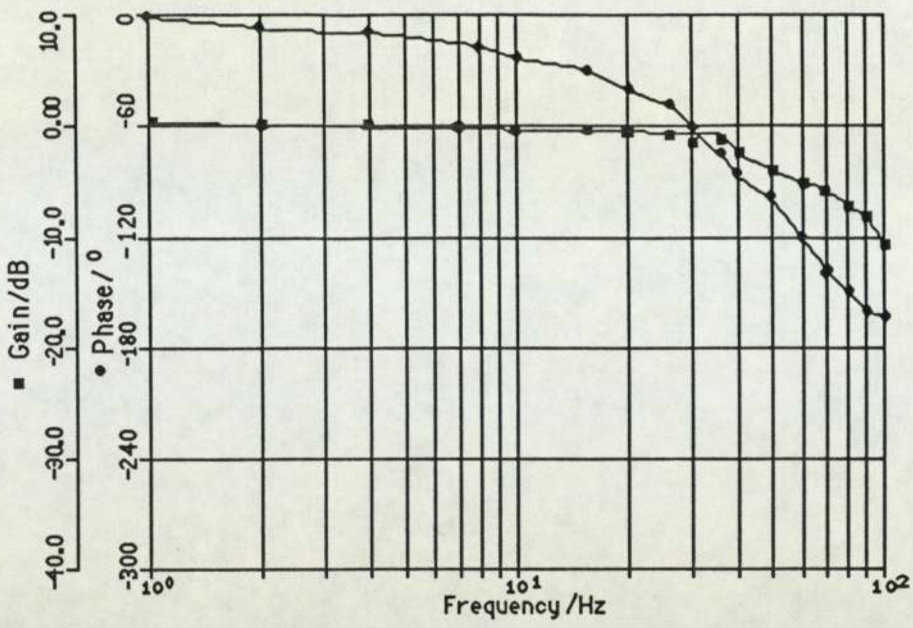


Figure A1.4 The velocity loop frequency response of the S-6200 system

A1.6.2 THE POSITION LOOP

The selection of the position loop parameters was by experience to try to reduce oscillations in output velocity.

The figures used in the PROPOS-E were as follows (figures had encoder counts and seconds as the base units)

FFSCAL	266667	(full speed feedforward of 2000r/min)
KP	700	(1000 was conditionally stable)
KD	0	
KI	0	

There were 8000 pulses per rev for the half stroke 80 pick/minute, and the toothed wheel has 89 teeth pitched at 1.335cm. Thus 2.75m relates to 2.215revs ≈ 17400 pulses. Also t_c was 0.29s and the dwell was 0.166s.

Even though the S-6200 could not maintain the full stroke application the PROPOS-E was programmed with four profiles: full and half stroke with trapezoidal and triangular profiles.

The motions were comprised of linear sections "glued" together by P/V commands where the first variable is the incremental position to be moved and the second the final velocity to be achieved.

<u>Full stroke triangular one shot</u>

001	P/V	17400	238337	(accelerate in half the distance)
002	P/V	17400	0	(decelerate in half the distance)
003	P/V	-17400	-238337	(accelerate out half the distance)
004	P/V	-17400	0	(decelerate out half the distance)
005	DWL	166		(dwell for 0.166s)
006	END			(end of program)

Half stroke trapezoidal continuous

001	AA	RPT	0
002	P/V	4350	89383
003	P/V	8700	89383
004	P/V	4350	0
005	P/V	-4350	-89383
006	P/V	-8700	-89383
007	P/V	-4350	0
008	DWL	166	
009	AA	RPE	
010	END		

A1.7 CONCLUSIONS

It was unlikely that inertias could be reduced enough to make the full stroke rapier viable for independent drives, unless a maximum machine speed of less than 80 picks/minute was acceptable.

For the half stroke rapier system, the use of independent drives was more promising. The use of the largest power Bru-500 system would allow speeds around 130 picks/minute with acceptable alterations to the toothed gear wheel. With the equipment loaned by Brintons the maximum speed that could be achieved would be around 110 picks/minute.

A half stroke independent drive system could be satisfactorily employed within a 5m loom. The S-6200, S-8350 or S-8500 drive systems could realise the original specification. Ultimate selection of a drive would depend upon machine speed required by Brintons and the inertia reductions that could be carried out on the toothed wheel, coupling, rapier and needle (the toothed wheel and needle being areas for particular attention).

For a repeated stroke 5.25m traverse in 0.292s, dwell of 0.166s, and return in 0.292s (80cycles/minute) as specified by Brintons even the largest power S-8500 could not achieve the specification. The full stroke system was not thought applicable to direct independent drives without drastic reduction in load inertias, or cycle times which was probably unfeasible.

For a repeated 1/2 stoke 2.75m traverse with two drives the results were promising. The S-6200, S-8350 and S-8500 were all capable of achieving the 80 picks/minute specification. Furthermore the S-6200 would be capable of 87 picks/minute, the S-8350 of 91 picks/minute and the S-8500 of 104 picks/minute.

It was thought that by reduction of the toothed wheel's diameter to 0.275m the S-8500 would be capable of achieving nearly 130 picks/minute which would be the maximum achievable by the Bru-500 range of drives.

APPENDIX 2

A2.0 EARLY TESTS ON A BRUSHLESS DC SYSTEM

A2.1 INTRODUCTION

The initial choice of drive was that of a brushless dc system from Inland; two drives were purchased one being kept by Molins at their Saunderton site whilst the other was loaned to Aston University. The two drives purchased from Inland comprised a 2206L and a 2202M motor with appropriate servo-amplifiers and power supply modules. The system was part of the company's BHT brushless drive range, refer to the user manual [19].

Numerous tests were carried out on the 2206L drive at Aston in order to characterise it. The velocity control was characterised by a bandwidth of 22Hz (@-115°) and accelerated to 3500 r/min in 0.6 seconds with 70 r/min overshoot.

The Acsl model developed represented the actual Inland system very well. The time and frequency responses obtained from both the real and the software system had good correlation inferring accurate modelling.

Unfortunately the Inland equipment was considered unsuitable for the particular application due to reliability problems, and it was returned in March 1987.

A2.2 THE INLAND SYSTEM

The system was designed on a modular axis approach, with the amplifiers and power supplies able to drive conventional and brushless dc servomotors. The system was automatically configured by a "plug-in" board to operate either type of motor. This board also adjusted the control parameters according to the expected load duty of the system. The drives operated in either velocity or torque mode responding to an analogue 10 V input; only velocity mode was used. The motor was driven using PWM techniques to reduce motor noise and heating, and to improve system efficiency. The current was controlled so that both motor and controller were protected, and fault diagnosis was available in the form of an LED display.

The amplifier functioned as a three-phase brushless dc motor controller performing electronic commutation. The controller had a servo loop whilst forming the currents for the three motor windings. The windings were excited by current pulses according to a rotor position feedback system utilizing digital Hall sensors.

The motor itself was a three phase, six pole, Samarium Cobalt rare-earth magnet brushless dc machine, with trapezoidal back emf waveforms.

A2.3 THE 2206L SYSTEM

The 2206L brushless dc system was loaned to Aston and was the higher powered of the two systems purchased. It was nominally a 3.7kW machine with a maximum operating speed of 3500 r/min. The system comprised three main elements: the power supply/ rectifier, the amplifier/ controller and the motor/ tachometer, as shown in figure (A2.1).

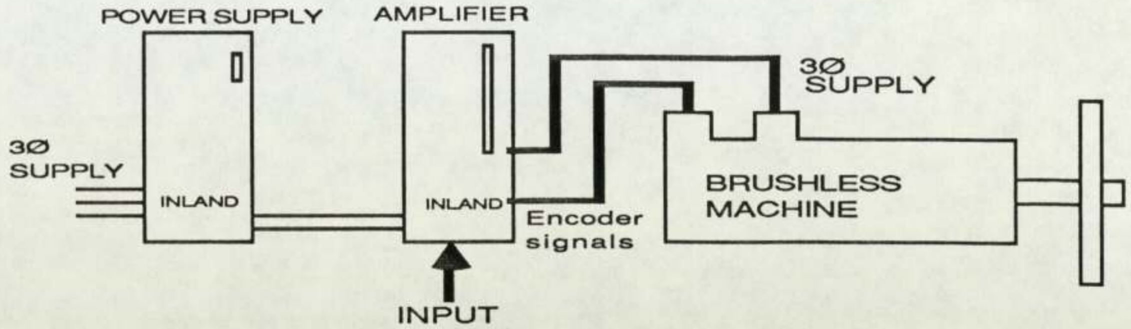


Figure A2.1 The general Inland arrangement

The power supply converted a nominal 220V 3 phase ac supply to 300V smooth dc bus, and it was capable of delivering a constant 60 Amperes dc. This unit also contained the regenerative dump resistor and the crow-bar circuit which dumped energy if the dc bus exceeded 375V dc.

The amplifier could control up to 40A continuous current, and used a 9.5kHz PWM system. It contained a motor control card with all the electronics for processing the position signals to select the motor phases in order to achieve electronic commutation. The servo loop circuit, the protection circuits and the power supplies were also in this module. The power driver module contained the six base-drive circuits and transistors which supplied the motor phases.

The analogue demand signal was filtered to remove any common mode noise, scaled through a potentiometer and passed through a low pass filter in the differential input and filter section of the electronics. This signal was then passed to the servo control loops, the first of which being the velocity loop. This loop compared the desired velocity to the actual velocity and provided an error signal to the current loop. Both lead and lag compensation networks were provided in the plug-in compensation board, and an offset potentiometer was used to correct for any drift during a zero-speed demand (compare this analogue system with the Electrocraft digital one).

The current loop compared the current demand with the actual current derived from current sample circuits and multiplexer. Again lead and lag networks were provided on the plug-in board. There were three current sampling circuits, one per phase, which were physically isolated from the power stage via pulse transformers. The current waveforms were multiplexed, using the rotor position as a decoder, to give a single value of current analogous to that of a conventional dc machine.

One of the features of the system was the protection circuits, in particular the three parallel current limit circuits; there was a peak limit, an I^2t limit which allowed peak current for four seconds before reducing to steady state current over ten seconds and an external programmable current limit was available, which was not utilised.

	2202M	2206L
POWER/kW	2.1	3.7
MAX. SPEED/R/MIN	5000	3500
CONT. TORQUE/Nm	6.6	15.3
PEAK TORQUE/Nm	21.8	60.4
CONT. CURRENT/A	13.6	26.9
PEAK CURRENT/A	50	80
MAX. TERMINAL VOLTAGE/V	300	300
TORQUE SENSITIVITY/Nm/A	0.486	0.812
DC RESISTANCE/Ω	0.65	0.40
INDUCTANCE/mH	4.6	3.8
ROTOR INERTIA/kgm^2	0.00054	0.00141

Table A2.1 General performance of the Inland system

A2.4 TESTS ON THE INLAND EQUIPMENT

Once the Inland equipment had been successfully commissioned a series of tests were devised to fully characterise the system.

The step responses from stand-still to a constant velocity took less than one second, and were monitored using the tachogenerator fitted to the system. Step responses were taken at speeds varying from 3500 r/min in a clockwise direction (measured from the load end of the shaft) to 3500 r/min in an anticlockwise direction. The traces were overlaid (as in figure (A2.3)) and followed the same profile. This was as expected as the velocity loop was saturated until close to the required value, that is, the machine ran "flat out" until near the demand.

A difficulty in taking the closed-loop frequency response of the system was that current limit greatly effected results; if the input sinusoid was less than 50 r/min peak to peak then no current limit effects were seen and so this response was the most meaningful. Responses were taken up to 400 r/min peak to peak, but above 50 r/min. two traces were obtained, one for steady-state values, that is, after current limit had taken full effect, and immediately after switching on, that is, before the current limit had any major effect. The sinusoid was superimposed upon a bias velocity ranging from standstill to maximum rated velocity, but there was very little variation in results at the same amplitude sinusoid; this suggests that the effects of friction were negligible. A sample result is given in figure (A2.4).

For small inputs the response resembled that of a second order system with a bandwidth of 22Hz (@-115°). Current limit increased the system phase lag and reduced gain as would be expected for a saturating element.

The system viscous damping was obtained by first calculating the system inertia by weighing components; the result was 0.05141 kgm^2. Secondly the motor was run up to overspeed condition, tripping the safety circuits, and recording the resulting run down to zero velocity. In this manner the run down was not effected by any motor regeneration and the trace was characterised by the load pole $1/J_Ts+B_v$. The time constant of the run down was found to be 34.8s and therefore $B_v=0.05141/34.8=1.48\times10^{-3}$ Nm/rads^{-1}.

A2.5 THE INLAND MODEL

Inland gave a block diagram for their system, but unfortunately the details of the contents of each block were not given; these had to be derived, from circuit diagrams. Detailed circuit analysis using conventional operational amplifier theory gave a number of relations which described the input filters, the velocity loop, the current loop, and the tachometer feedback elements. The derived relationships are given below:-

INPUT FILTER

$$Vout = \frac{Vin\times6700\times10330785}{(s+6700)(s+2273)(s+4545)}$$

TACHOMETER FILTER

$$Vout = \frac{Vin\times1485410160}{(s+30160)(s+49251)}$$

VELOCITY LOOP

$$Vout = 2752.5(Vin+Vtacho/10)\frac{(1+0.038s)}{s(1+0.0008s)}$$

CURRENT LOOP

$$Vout = \frac{70.1(1+0.05s)}{s(1+0.0001s)}\left(Vin+Vcurrent\frac{(1+0.0105s)}{(1+0.0041s)}\right)$$

Both the velocity and current loop operational amplifiers had 10V zener diodes limiting the loop outputs. From a calibration test on the tachogenerator its sensitivity was found to be 0.1 V/rads^{-1}. Inland suggested that an accurate model for the PWM system was to divide the dc bus voltage by the analogue signal voltage, that is 327V/10V = 32.7. The motor was modelled as a simple pole determined by its inductance and resistance. The load was modelled as a simple pole determined by viscous damping and inertia; a small static friction component was also included.

A block diagram of the system was formed as shown in figure (A2.2). The input was X and the state variables were denoted by Y's, with Y13 as the motor current and Y15 the output velocity in radians per second. All the poles and zeros were found from the equations given earlier, from the Inland data and from experimental tests. The program listing (A2.1) shows the ACSL implementation of the model.

A2.6 SIMULATION RESULTS

The model was tested with a number of different magnitude step inputs by varying X, from figure (A2.2), at run time. All the responses were plotted together and overlaid upon replotted actual step responses, as shown in figure (A2.3). The correlation between the two is excellent with negligible difference in overall settling value and with less than 10% difference on the time scale for a given velocity. From these results the model did represent very well the Inland system in the time domain.

```
PROGRAM STEP RESPONSE OF INLAND 2206L
       "INLAND 2206L STEP RESPONSE"
      CINTERVAL       CINT = 0.05
      NSTEPS          NSTP = 1
      MAXTERVAL       MAXT = 1.5E-5
      CONSTANT        TSTP = 6,              X = 7.0,               MAXTXZ = 1.5E-5
      CONSTANT        POLE1 = 1.5E-4,        POLE2 = 4.4E-4,        POLE3 = 2.2E-4
      CONSTANT        POLE4 = 8E-4,          POLE5 = 1E-4,          POLE6 = 0.0041
      CONSTANT        POLE7 = 3E-4,          PWM = 32.7,            ILIM = 80.0
      CONSTANT        ZERO4 = 0.03854,       ZERO6 = 0.01,          K7 = 0.10218
      CONSTANT        POT1 = 0.535,          POT2 = 0.27,           ZDIODE = 10.0
      CONSTANT        K9 = 0.812,            K10 = 0.35,            K6 = 0.2
      CONSTANT        R = 0.36,              L = 3.99E-3,           Kbc= 2.0
      CONSTANT        J = 0.05141,           B = 0.00144,           NSTPMN = 10.0
      CONSTANT        R366 = 43200,          R9 = 150000,           C5 = 330E-9
      "DIFFERENTIAL INPUT & SCALE FACTOR"
      Y1 = POT1*REALPL(POLE1,X,0.0)
      "INPUT FILTER"
      Y2 = REALPL(POLE2,Y1)
      Y3 = REALPL(POLE3,Y2)
      "VELOCITY LOOP"
      Y5 = Y3-Y4
      VGAIN = 2312.3/(1.11-POT2)
      VUL = ZDIODE/VGAIN
      VLL = -VUL
      Y6 = LIMINT(Y5,0.0,VLL,VUL)
      Y7 = VGAIN*BOUND(VLL,VUL,LEDLAG(ZERO4,POLE4,Y6,0.0))
      "CURRENT LOOP"
      Y9 = Y7-Y8
      IUL = R9*C5*ABS(ZDIODE*R366/R9-ABS(Y9))
      ILL = -IUL
      Y10 = Y9*R9/R366 + (1/(R366*C5))*LIMINT(Y9,0.0,ILL,IUL)
      Y11 = BOUND(-ZDIODE,ZDIODE,Y10)
      "Outer current loop and p.w.m."
      Y12 = PWM*REALPL(POLE5,Y11)
      "MOTOR (Y13=I,Y=W,K9=BACK EMF,K10=LOAD TORQUE"
PROCEDURAL(Y13=Y12,Y15)
      Y13 = (1/R)*REALPL((L/R),(Y12-K9*Y15))
      IF(ABS(Y13).GT.ILIM)Y13=SIGN(ILIM,Y13)
END $"OF PROCEDURAL"
PROCEDURAL(Y14=Y13)
      Y14=Y13*K9*Kbc-SIGN(K10,Y13)
      IF(ABS(K9*Kbc*Y13).LT.K10)Y14=0.0
END $"OF PROCEDURAL"
      Y15 = 1/B*REALPL((J/B),Y14,0.0)
      "Convert to r.p.m."
      RPM = Y15*9.55
      "Convert to input volts"
      Y = RPM/500
      "CURRENT FEEDBACK"
      Y8 = K6*LEDLAG(ZERO6,POLE6,Y13,0.0)
      "TACHOMETER VELOCITY FEEDBACK"
      Y4 = (K7/10)*REALPL(POLE7,Y15,0.0)
       TERMT(T.GE.TSTP)
END $"OF PROGRAM"
```

Program A2.1 Simulation of the Inland 2206L System

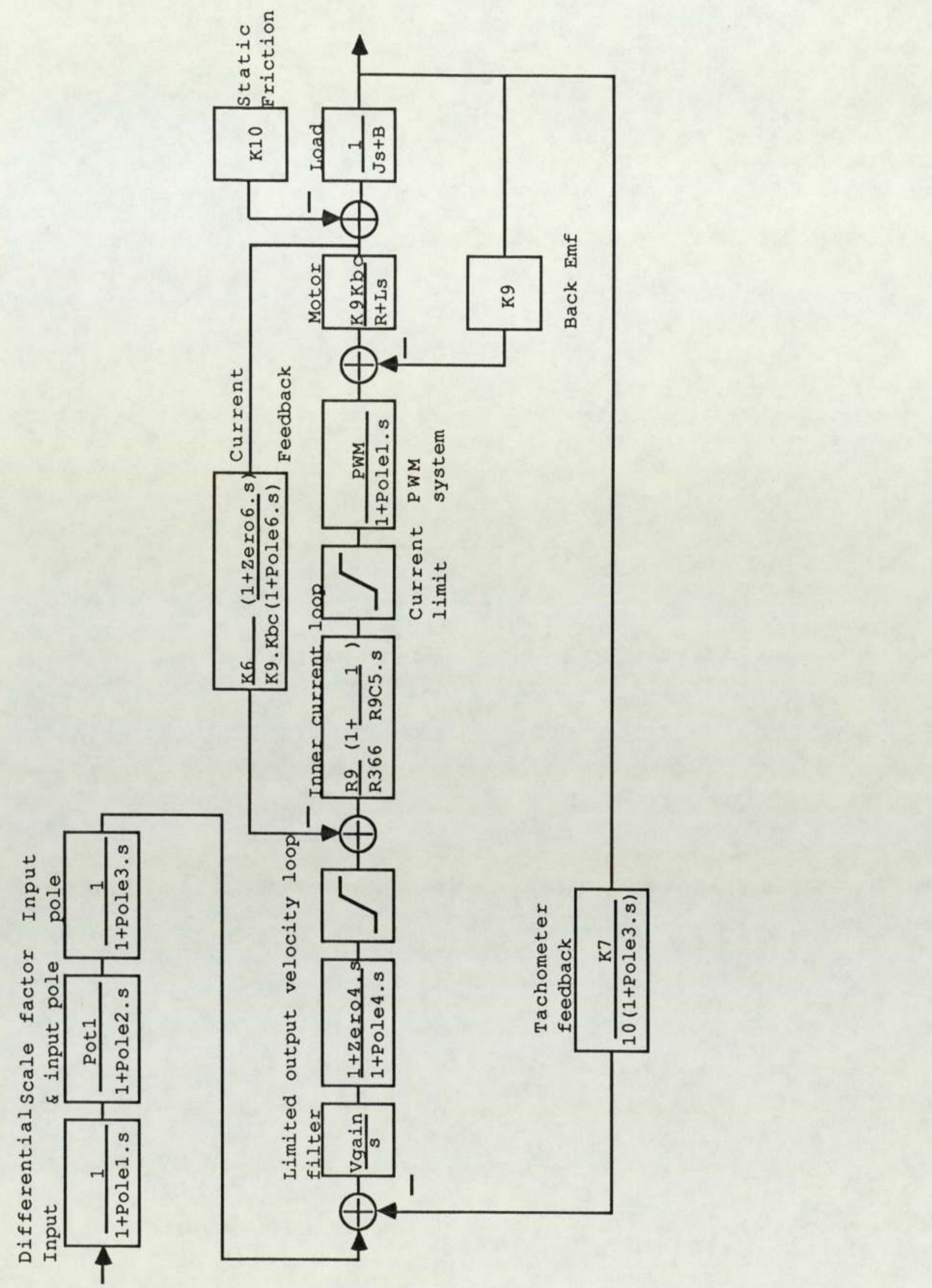


Figure A2.2 Block diagram of the Inland model

Often a better indicator of model accuracy is the comparison in the frequency domain. Thus the program was altered to take frequency responses of the system.

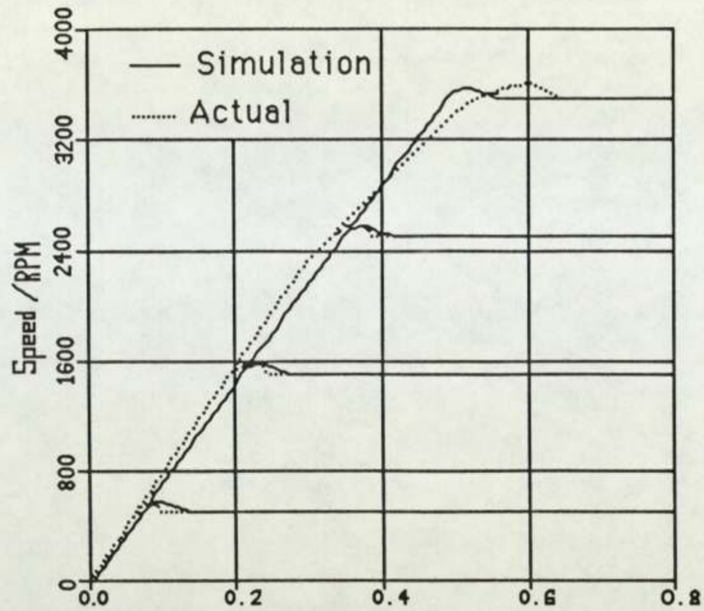


Figure A2.3 The step responses of the Inland system

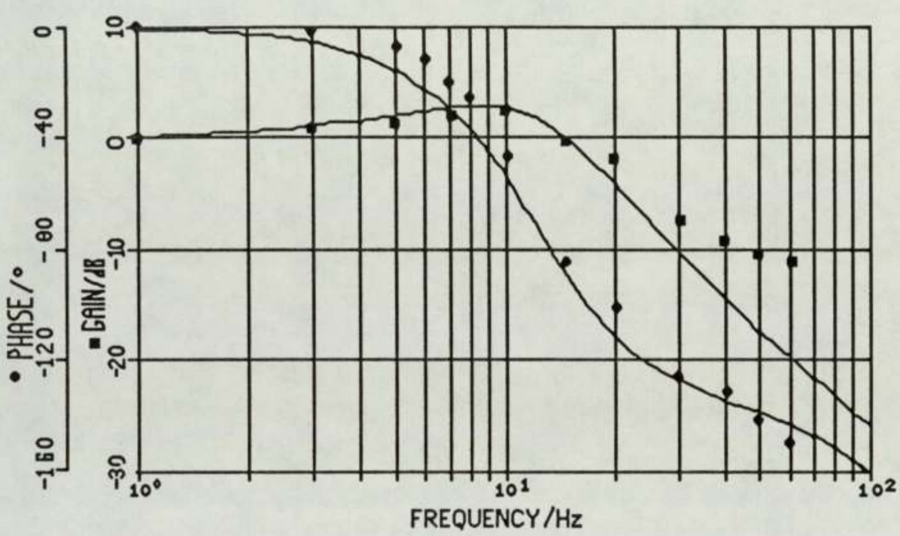


Figure A2.4 A typical frequency response from the Inland system

The frequency response was taken for the whole model with various input amplitudes corresponding to the waveforms applied to the real system. The most relevant one was with a demand amplitude of 50 r/min as this was before the effects of current limit were noticed, as shown in figure (A2.4) which has points from the actual system superimposed. Again good correlation was gained, although slight differences are due to two main reasons. Firstly, experimental error could account for some of the discrepancies especially at the higher frequencies where it was noticeably difficult to distinguish between noise and signal (when taking manual gain readings there was a lot of noise induced blur on the signal and it was thought that some of the readings could have been read high by as much as 3 dB). Secondly, it was discovered that the system response depended greatly, as might be expected, upon the feedback, and the simulation used figures which were quoted only to within ±10% accuracy. If the feedback elements were more accurately given then a better correlation could be expected.

APPENDIX 3

A3.0 A LIST OF SEVEN MANUFACTURER'S DRIVE SYSTEMS USED IN THE COMPARATIVE EXAMPLE

The comparison of chapter 3.0 used seven different drives systems which are listed below:-

Drive 1 was the Gould M100 and M200 series DC servo motors
Drive 2 was the ASR Servotron SD4000 & SA7000 DC servo motors
Drive 3 was the ASR Servotron SX4100 DC servo motors
Drive 4 was the Infranor Series SE disc rotor ac servomotors
Drive 5 was the ElectroCraft BRU500 brushless servo drives
Drive 6 was the Inland series BHT brushless servo drives
Drive 7 was the Contraves MACS ACR ac servo drives

APPENDIX 4

A4.0 THE BRU-500 SIMULATION SUITE

A4.1 INTRODUCTION

The suite of simulation programs developed had a modular structure to facilitate further development; the major blocks were the input control, the velocity loop digital control algorithm, the analogue current loop, and the load equations. The programs simulated the control algorithms supplied by Electrocraft for use with their drives, and the specially developed Molins Dservo microprocessor controller. The models developed were complex but for rapid simulation many of the blocks could be switched out to yield a simple block diagram structure.

The simulation language used throughout was the Advanced Continuous Simulation Language (ACSL). The suite of programs was developed upon the Vax Cluster at Aston University, but the programs could equally be run on an IBM PC or Clone using the PC version of Acsl. Unfortunately there are some differences between the two systems.

Dependent upon the requirements for any given simulation run, the correct program from the suite required selection. The programs were written such that the complexity of the simulation could easily be changed from a simple simulation which could yield results rapidly, to slower, complex simulations.

The simulations were built up over a period of approximately 10 months as a result of intensive laboratory tests upon the equipment. As such the simulation models gave an in-sight into the servo system, and could be a very accurate guide to the systems actual performance.

A4.2 BLOCK DIAGRAM DERIVATION

The majority of information from the drives was derived experimentally. The evidence was collaborated later by Electrocraft with reference to circuit diagrams of the system, and by detailed technical exchanges with their staff. The control algorithms used by the drive were derived from linear block diagrams of the systems provided by the American designers of the Bru-500 controller.

Six block diagrams are included for reference which show all the variants used by the simulations, figures (A4.1) to (A4.5).

The general configuration, figure (A4.1), shows how the overall simulation block diagrams for the Electrocraft system were derived, from essentially four major elements. There was an input section, a digital control algorithm comparing the actual to demanded position in pulses, which fed a current loop, which in turn produced torque to rotate the mechanical load. The input to the system was derived by use of switches SX1-4, whereby the input could be a step, a pulse train, a ramp, a sinusoid, or a combination of these. Each type of input had one start time and one stop time so there were many different waveforms that could exercise the model. It was found experimentally that the input to the BRU was filtered before being passed into the microprocessor. Thus the input section shows this filter, the input limit (of 10V) and the scaling which first scaled the input from Volts to r/min, and then to pulses.

There were two different internal velocity control algorithms used within the Bru-500 system, which were classed Bandwidth/ Damping (B/D) control and Proportional/ Integral (P/I) control.

Both the Electrocraft control algorithms used the difference between the demanded encoder pulses and the actual fed-back encoder pulses as an error signal to drive the current loop. The system produced an equivalent 0-±10V signal into the drive's current loop proportional to zero to full drive current. The algorithm updated the current demand every millisecond.

The B/D control, figure (A4.2), used two parameters to change the nature of the drive; these were termed "bandwidth" and "damping". The control algorithm as originally described by Electrocraft was a velocity loop, whereby the velocity and acceleration (scaled in pulses per sample and pulses per sample per sample) were fed back to a summing junction, and compared with the demanded velocity (again scaled in pulses per sample). This error was multiplied by a gain value, integrated to reduce steady state error and filtered (the single pole software filter value was controlled from the keyboard interface between 0-300Hz). The bandwidth term altered the forward loop gain, and hence the actual bandwidth of the system, whilst the damping controlled the forward loop gain and the derivative velocity (acceleration) feedback term; it had a similar effect on the control of the system as the damping factor in a second order system. This derivative feedback had the effect of canceling some of the forward loop integration to enhance stability. It was noticed that the finest level of feedback information was one pulse per sample (0.8 rads^{-1}) for velocity, and one pulse per sample per sample (800 rads^{-2}) for acceleration. This seemed very coarse information to use so Electrocraft were consulted. They admitted that the algorithm was not a velocity loop, but a position loop.

Thus a second B/D algorithm block diagram, figure (A4.2), was derived for the position loop case. Here there was no forward loop integration and only position and velocity feedback. The algorithm was much the same as described above.

The P/I algorithm, figure (A4.3) compared the velocity error between demand and actual as before, but formed the current signal by adding a proportional and an integral part of this error together. These parts were changed by two keyboard controlled constants PGAIN and IGAIN which were limited to a value of 2000. A software filter was also included, as in the B/D algorithm, which was useful in suppressing resonances.

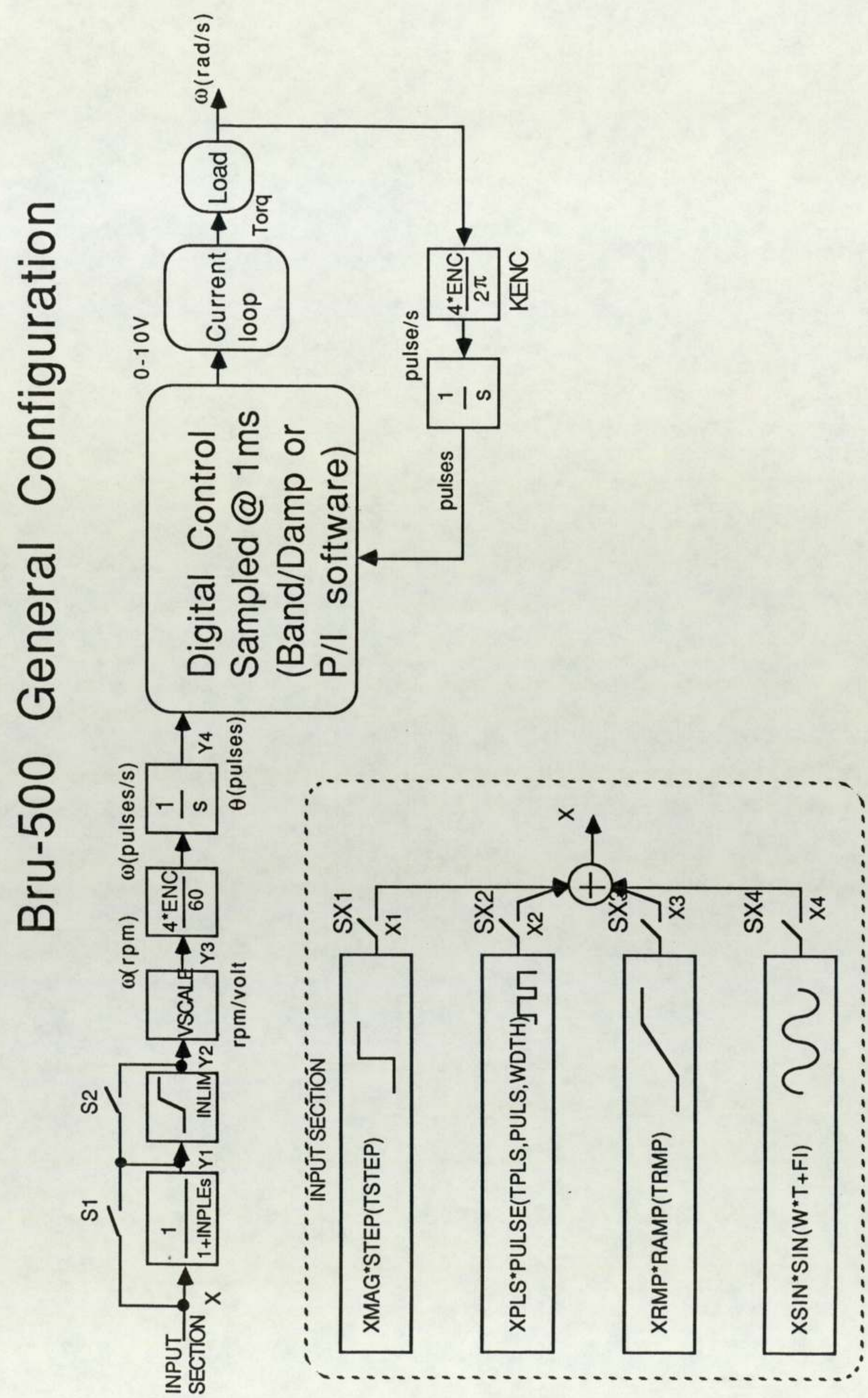


Figure A4.1 The general configuration block diagram

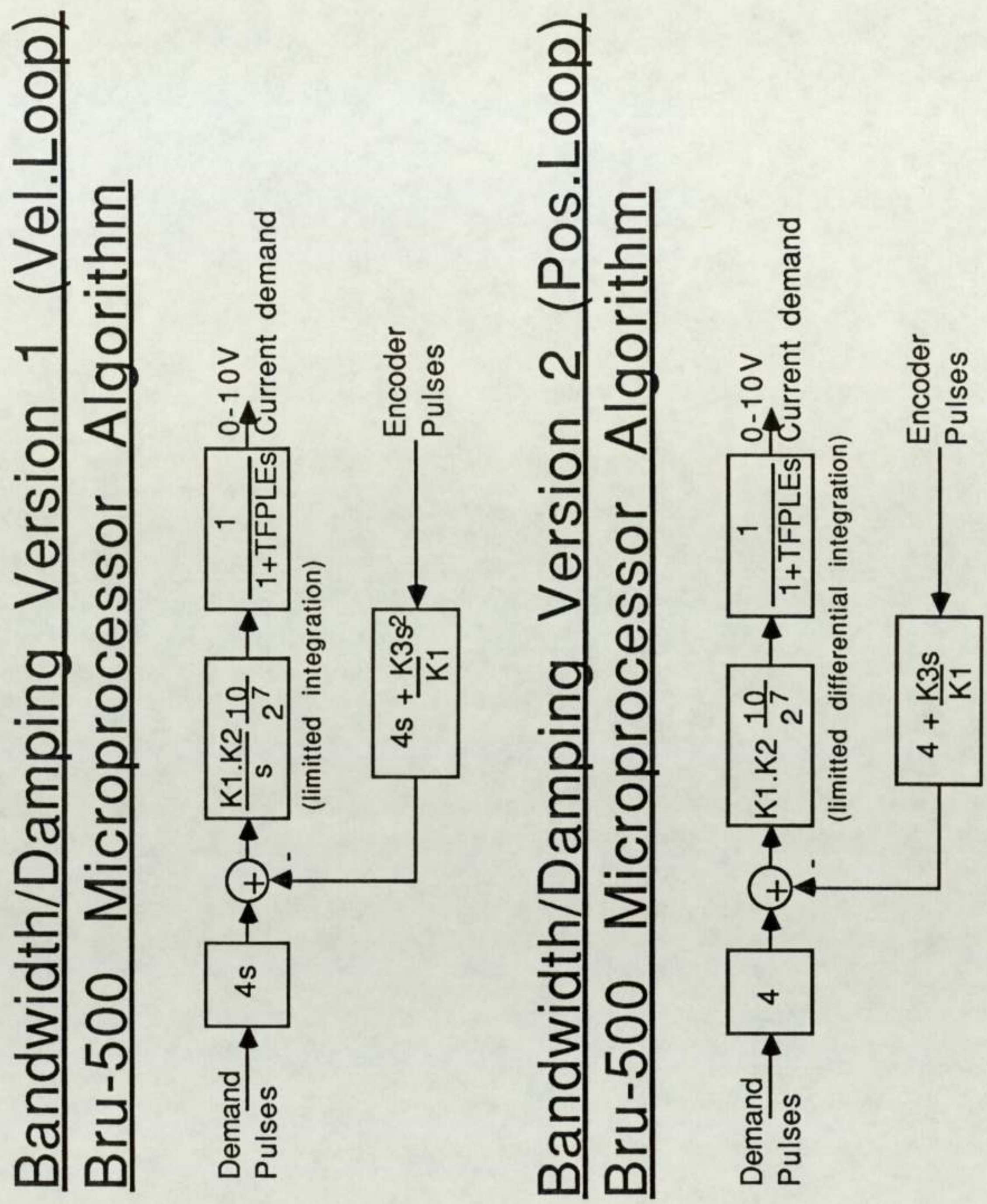


Figure A4.2 The Bandwidth/Damping algorithm

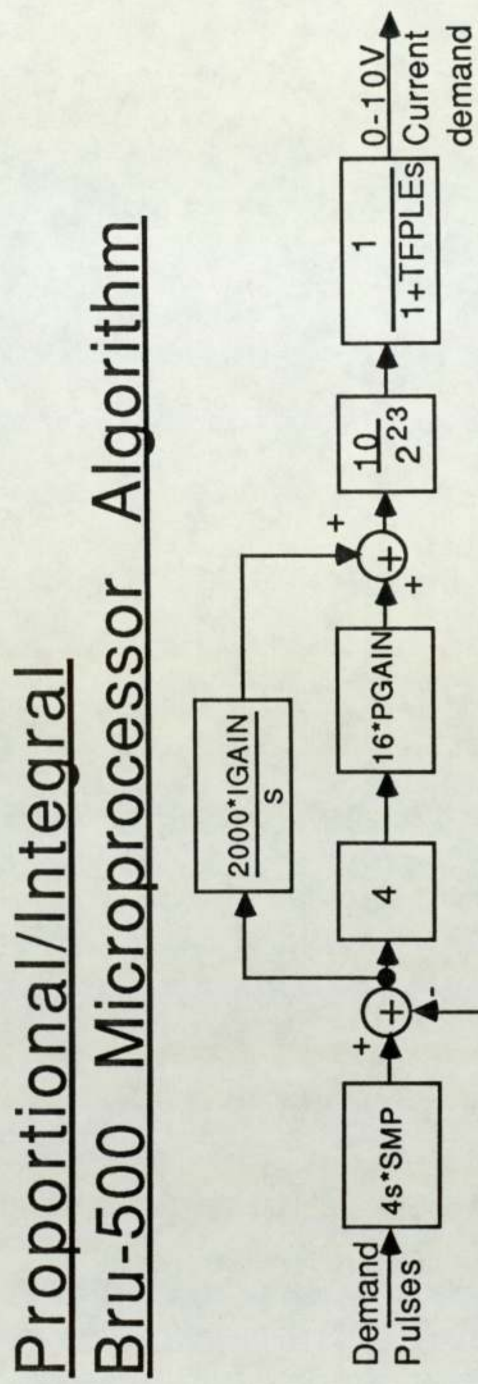


Figure A4.3 The Proportional/Integral algorithm

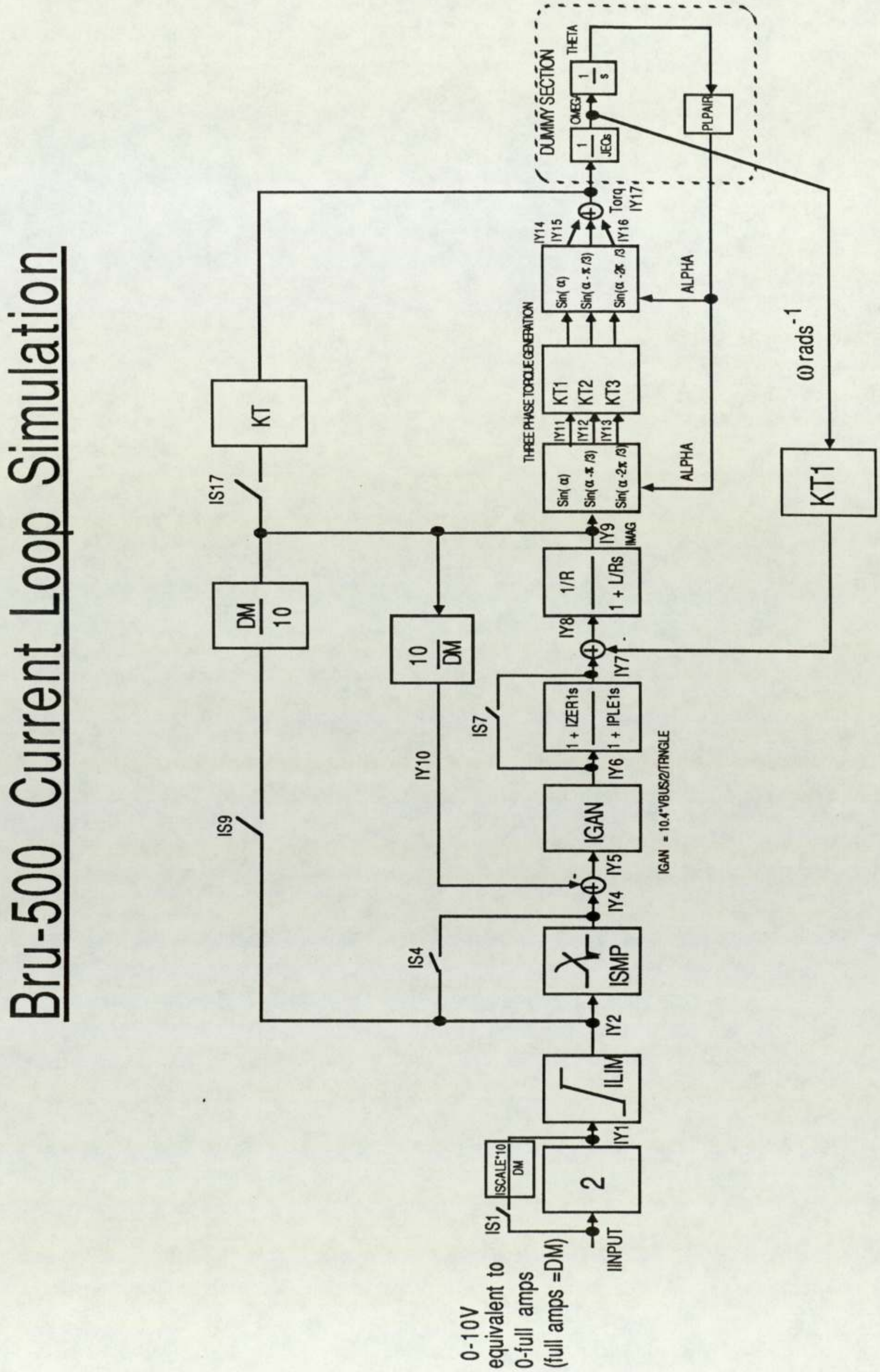


Figure A4.4 The current loop block diagram

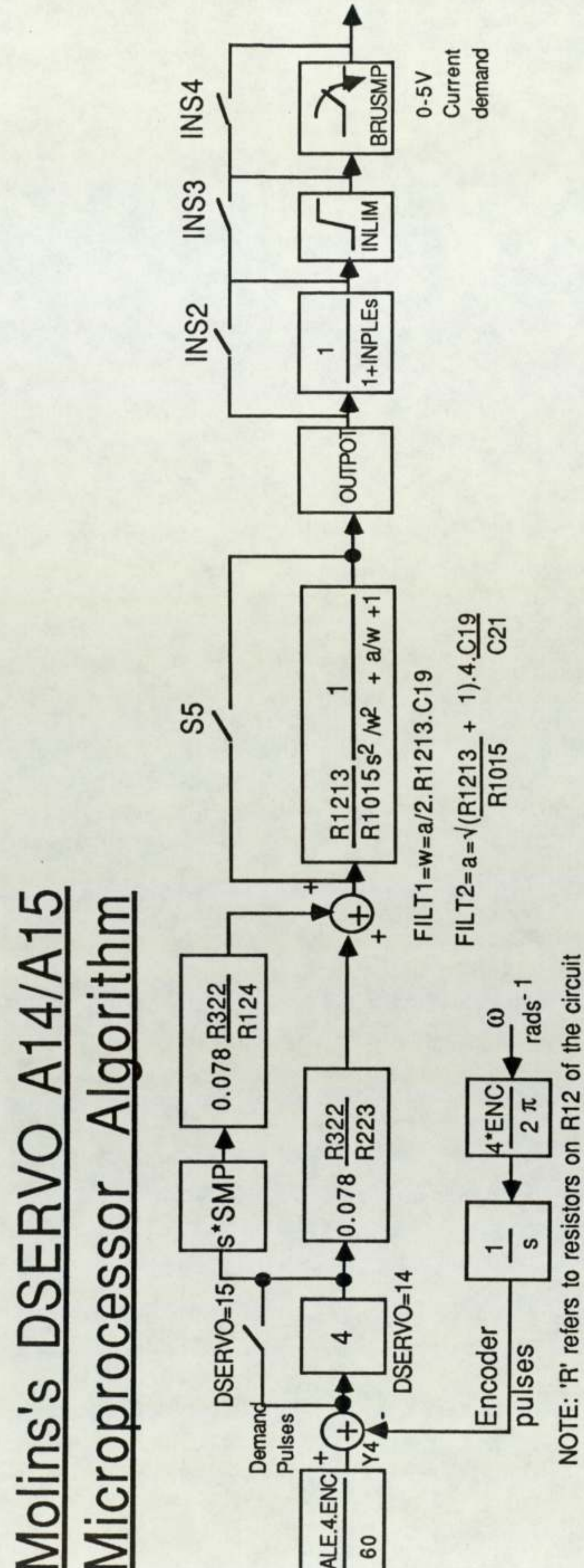


Figure A4.5 The Dservo algorithm

In all early simulations it was assumed that the drive was fed from a pure current loop, that is, the torque produced was a constant multiplied by the demanded current. A detailed study of Electrocraft circuit diagrams, and of torque generation in three phase machines, yielded the block diagram shown in figure (A4.4). The diagram shows the software switches that were included in the simulation. The simplest simulation model was when IS1,IS9,IS17 were closed. In this case the torque produced was ISCALE*KT multiplied by the demand voltage. ISCALE was a constant held within the BRU microprocessor which determined the scaling of amperes per volt ; normally it is set such that 0-10 Volts was equivalent to 0-full Ampere rating of the system, for example, a DM-50 had an ISCALE of 50/10 = 5 A/V.

IS1 was used to switch between the normal state (1) and the Molins Dservo state (0). The normal state was where the current was controlled by the BRU's internal control algorithm or when the system was in torque mode, and the voltage representing current demand was fed into the DM via the normal "VCS" input. For use with the Molins Dservo system it was found prudent to by-pass the BRU microprocessor and to input directly into the current loop by minor alteration to the BRU circuitry. In this case 5 Volts represented full load current, or the system gain was K_T.DM/5 Nm/V (K_T was the motor total torque constant, and DM the Drive Module number which represented the full load current).

If IS9 and IS17 were open (0) then the majority of the current loop simulation was invoked. IS4 controlled whether the current loop sampling was utilised, as an EEPROM was used every 250μs to decode position and current demand to form the three phase excitation. IS7 switched the high-order analogue lead-lag current loop shaping network in or out of the simulation model. The current loop consisted of three separate loops each one controlling one phase; these were very similar and so were modelled as one. The current loop compared the demanded current magnitude with the actual current to produce an error signal which was multiplied by the current loop gain and passed through a lead-lag network. The current loop gain, IGAN was determined by a constant multiplied by the dc bus voltage divided by the amplitude of the PWM base frequency triangle wave. This amplitude could be changed by the microprocessor to give higher gain at higher motor speeds to combat the effects of back emf, but tests showed that for the drives available this facility was not utilised in the software supplied by Electrocraft. The PWM produced the effective dc voltage applied to the machine, and when the back emf voltage was subtracted from this, the voltage remaining was available to force current through the motor winding which consisted of the per phase resistance and inductance; mutual inductances were neglected as they would greatly complicate the model and data was difficult to obtain for these parameters.

In order to investigate the differences between the drive's phases, and to relate this to torque ripple, the model included the possibility of operating with either single or multi-phase torque generation. KT1-3 were the per phase torque constants which could vary by as much as 10% in a machine (although generally within 5%). The simulation synthesised three sinusoidal current waveforms from the dc current value and the rotor electrical position (alpha), and produced three sinusoidal torque waveforms which were summed to form the total torque production of the machine. When balanced there was no torque ripple but if unbalanced then a ripple would be observed. The torque was passed to the load for simulation purposes which produced a velocity and position value from the torque.

Whilst the load simulation was understood it presented some of the worst problems when actually running simulations. This was due to two reasons, firstly numerical instability when running simulation, and secondly it was extremely difficult to obtain accurate figures for such parameters as torsional stiffness, backlash and viscous damping. Since the torsional stiffness of the motor shaft was so high, the load poles were at a very high frequency when compared to the other system poles, and this proved difficult to simulate because of numerical instability (rounding errors). Thus it proved impossible to simulate backlash, or a direct coupling between motor and load with high torsional stiffness, that is, the motor shaft. The loads used in the simulation model assumed a perfect (infinitely stiff) direct coupling between the load and motor, with some viscous damping, calculated from "run-down" tests.

Two variants of the Molins Dservo controller, A14 and A15 were simulated, as shown in figure (A4.5). They differed in that A14 had an internal software gain of four. Variant A16 also existed, which had A14 software allowing for cross coupling between axes, so that A14 simulations may be used. The input to the actual microprocessor board was by pulse trains, but to interface with previous simulation experiments it was decided to use an analogue equivalent demand. Thus a voltage input was scaled to r/min and pulses as with the Electrocraft system, and this was compared to the feedback pulses from the encoder. Switch DSERVO set to 14 or 15 determined which revision of the servo card was being used. The control scheme was very simple in that the microprocessor output via a two stage digital to analogue converter a signal proportional to velocity error and one proportional to position error. These signals were scaled in analogue circuitry and summed. The "R" numbers in the block diagram refer to the resistors in the header chip upon the microprocessor card. The signal was filtered and scaled via an output potentiometer (OUTPOT); the filter could be switched in or out by switch S5. Originally the output from the Dservo card was fed directly into the BRU input VCS, but this caused control problems due to the input pole, INPLE, and the sampling within the BRU microprocessor. This section was by-passed (switches INS2-4) by alteration of the BRU circuitry.

Thus the simulation block diagrams evolved using detailed analysis from simple systems to the complex form. The complexity increased as the understanding of the system became better. The block diagrams were drawn in such a way that they mimicked the action of the simulation programs that are discussed in section (A4.3).

A4.3 BLOCK DIAGRAM ENCODING

{Note: Program names in parenthesis [] refer to the IBM PC system}

Programs were written which simulated the control algorithms, the current loop and the load in isolation. These programs could be run on there own and they contained "dummy" sections which contained any additional programming required to run them in isolation from other blocks. For example, the algorithm programs contained a simple dummy current loop and load, and the current loop program, a dummy load section.

One program was developed in isolation from the others. This was EMPTYBODE.CSL [BODE.CSL] and it was used to take the closed and open loop frequency responses of the other simulation programs. Incorporation of a program into EMPTYBODE.CSL [BODE.CSL] was a simple operation. The program used a sinusoidal input excitation to the model (X) and integrated the in-phase and quadrature components of the output (Y), to calculate gain and phase at a given frequency.

The digital controllers were implemented in a DISCRETE block of code. The first part of each of these DISCRETE blocks for each of the different algorithms used was the same; it consisted of a section of code which obtained the quantised, to one pulse, position and velocity demand and feedback in pulses and pulses per second, respectively. These parameters were passed to the algorithm section for use in computing the current loop demand voltage.

The first block of code in each program contained default information for simulation. A brief description of the particular program, it's version number and the date of the last revision were given. The default integration algorithm used in each case was the Runge Kutta fourth order (IALG =5). The CINTERVAL and MAXTERVAL were also set in this section.

The initial section of each program contained all the operations carried out prior to any simulation run. Commonly used constants were calculated to save recalculation every Acsl sample period, and certain constants were set to their initial value (usually zero) to prevent a value from a previous run being entered and causing unpredictable results. The initial section in some of the programs was split into two parts, those operations required by the program's main section and those required by any dummy section. The areas were distinguished to aid with any later editing of the program.

The main area of the program was the DYNAMIC block of code which contains the DERIVATIVE and DISCRETE sections. All the required constants were at the beginning of the DERIVATIVE section and the constants were held in distinguishable blocks. Any dummy section was also at the beginning of the DERIVATIVE section with its own constants. There was a block of constants not required by the program which were set to zero. These constants were included in order to prevent error messages such as "CONSTANT NOT FOUND" when running the common program suite run-time file. The other groupings of constants were those used only by the program in question, those used by this and other programs in the suite and those common constants such as pi.

After the constants came the main body of program which was written following the block diagram very closely. The variables were described approximately in the order they appeared, reading left to right from the block diagram, and they were named in order. Each set of variables had an alpha numeric name, the alpha prefix referred to a particular part of the program, and they were numbered left to right on a block diagram. Y was used to denote a continuous variable, whilst Z denoted a digital variable. The prefixes used are as follows:-

X	was the system input
Y	defines the early input stages of a model
IY	defines a variable in the current loop
LY	defines a load constant
AZ	defines a digital constant concerned with feedback
BZ	defines a digital constant concerned with demand
BDZ	defines a digital constant concerned with the B/D algorithm
PIZ	defines a digital constant concerned with the P/I algorithm
MZ	defines a digital constant concerned with the Molins Dservo algorithm

Also some particular variables were given mnemomic names if they were of particular relevance:-

ALPHA	the electrical rotor angle (°)
DEMAND	the demanded rotor position (pulses)
ENCODE	the feedback rotor position (pulses)
IINPUT	the input into the current loop (V)
IMAG	the current magnitude in the motor (A)
OMEGA	the mechanical rotor speed (rads^{-1})
THETA	the mechanical rotor angle (°)
TORQ	the torque produced by motor (Nm)
YRPM	the mechanical rotor speed (r/min)
YTACH	the output voltage from a tachometer (V)

The constants used by the simulation all had mnemomics which gave some indication of their relevance. The most important ones could be altered by use of the run-time file BRU.DAT [BRU.CMD]:-

BAND	the bandwidth gain in the bandwidth damping algorithm
BLSH	the backlash between load and motor (rad)
C19	Capacitor value on Molins Dservo card (F)
C21	Capacitor value on Molins Dservo card (F)
DAMP	the damping gain in the bandwidth damping algorithm
DM	the drive module used with the motor, eg 25,50
DSERVO	the revision number of the Molins Dservo (14 or 15)
ENC	the number of pulses per revolution on the encoder
FBAND	the bru 500 software filter bandwidth (Hz)
G1	gain constant for bandwidth damping algorithm
G2	gain constant for bandwidth damping algorithm
IGAIN	the integral gain constant in the P/I algorithm
ILIM	the motor/drive current limit (A)
INLIM	the limit on input to the BRU (V)
INPLE	the BRU input pole time constant (s)
IPLE1	the current loop shaping pole time constant (s)
ISCALE	the current loop input scale factor (A/V)
ISMP	the sample period of the current loop
IZER1	the current loop shaping zero time constant (s)
JLOAD	the load inertia (kgm^2)
JMOT	the motor rotor inertia (kgm^2)
K3	feedback gain constant for bandwidth damping algorithm
KT1	the torque constant of phase one of the motor (Nm/A)
KT2	the torque constant of phase two of the motor (Nm/A)
KT3	the torque constant of phase three of the motor (Nm/A)
KT	the motor total torque constant
L1	the motor inductance of phase one (H)
MOT	defined the motor default constants, eg 3016,4030 etc
OUTPOT	Output potentiometer setting on Molins Dservo card
PGAIN	the proportional gain constant in the P/I algorithm
PLPAIR	the number of motor pole pairs
QUANT	the number of pulses the algorithm quantises to (usually one)
R1	the motor resistance of phase one (Ω)
R1015	Resistor value on Molins Dservo header (Ω)

R124	Resistor value on Molins Dservo header (Ω)
R1213	Resistor value on Molins Dservo header (Ω)
R223	Resistor value on Molins Dservo header (Ω)
R322	Resistor value on Molins Dservo header (Ω)
SMP	the control algorithm sample period (s)
SPRNGK	the motor shaft torsional spring constant (Nm/rad)
TRNGLE	the amplitude of the PWM reference triangle waveform (V)
VBUS	the dc bus supplied to the drive module (V)
VSCALE	the algorithm input scale factor (r/min/V)
VSCDMP	the load viscous damping (Nm/rads^{-1})
VTACH	the digital tachometer gain (V/rads^{-1})

The software switches were named according to the last variable they switched out and contained an "S" for switch; for example, switch ABCS3 switched out a block between variables ABC2 and ABC3, although in a few cases more than one block was switched out as in the current loop.

There were seven basic programs from which the others were derived. BRULOAD.CSL [BL.CSL] simulated a simple load situation where the load inertia was coupled to the motor by an infinitely stiff motor shaft, and there was some load viscous damping. BRUCURRENTLOOP.CSL [BI.CSL] simulated a full BRU-500 current loop with three phase torque generation. EMPTYBODE.CSL [BODE.CSL] took a frequency response of a given model imbedded in it. BRU_BAND_DAMP_CONTROL1.CSL [BBD1.CSL], BRU_BAND_DAMP_CONTROL2.CSL [BBD2.CSL], BRU_P_I_CONTROL.CSL [BPI.CSL] and MOLINS_DSERVO_CONTROL.CSL [BM.CSL] simulated the B/D algorithm as a velocity loop, as a position loop, the P/I algorithm and the Molins Dservo controller respectively. These were all basic programs holding blocks that could be used to build up more sophisticated simulation models.

The algorithm programs were integrated with the load and current loop blocks to create full simulation models taking the form *_FULL.CSL [*F.CSL}; for example BRU_BAND_DAMP_CONTROL2_FULL.CSL [BBD2F.CSL] simulated the B/D position loop algorithm in fine detail. The "FULL" programs were also integrated into the EMPTYBODE.CSL program to create programs capable of taking frequency responses; these had the name _BODE.CSL [*B.CSL], such as, BRU_BAND_DAMP_CONTROL2_FULL_BODE.CSL [BBD2FB.CSL].

There were fifteen programs in the suite, eight of which were derived from the seven more basic programs.

A4.4 SPECIAL RUN-TIME SIMULATION COMMANDS

The Acsl manual [1] gives detailed reference to run time commands, but the system was set up to save time. The following were commands recognised by the system:-

S3016, S4030, S4050, S6100, S6200 introduced the constants into the simulation for each particular Bru-500 motor.

NORM set the system for a time domain response and OUTPUTed and PREPARed the time (T) and motor velocity (OMEGA).

LOG set the system for a frequency domain response and OUTPUTed and PREPARed the frequency (WHZ), phase (PDG) and gain (GDB) of the system.

PLOTNORM plotted the velocity (OMEGA in rads^{-1}) against time (T in seconds), and titled the graph with relevant information.

PLOTLOG plotted the phase (PDG in degrees) and gain (GDB in decibels) against the logarithmic frequency (WHZ in log(Hz)).

A4.5 PROGRAM LISTINGS

From the fifteen programs that were derived, relevant sections are re-produced here. The full programs given are the current loop simulation, the empty Bode program, the full B/D velocity loop control algorithm simulation software and the simple Molins Dservo control software. Also included are excerpts from other programs to show the B/D position control loop algorithm and the P/I control software. The run-time file is also included.

BRUCURRENTLOOP.CSL;1 [BI.CSL]

```
PROGRAM BRU500_SIMULATION_OF_CURRENT_LOOP
      "VERSION 1 JULY 1988---------------------------------------------------"
      "COPYRIGHT DAVID R. SEAWARD JULY 1988-----------------------------------"
      "SIMULATION OF THE ELECTROCRAFT BRU500 SERIES OF BRUSHLESS MACHINES"

      "THIS PROGRAM SIMULATES THE CURRENT LOOP--------------------------------"
      "IT MAY BE USED ALONE OR PARTS EXTRACTED TO FORM PART OF A LARGER--"
      "SIMULATION PROGRAM-----------------------------------------------------"

      "THE SWITCHES SHORT OUT A BLOCK WHEN SET TO ONE-----------------------"
      "THE SWITCH NUMBER REFERS TO THE VARIABLE IT SWITCHES-----------------"
      "IE. IS2 SWITCHES OUT THE BLOCK BETWEEN IY1 AND IY2--------------------"

      "DEFAULT COMMUNICATION INTERVAL-----------------------------------------"
      CINTERVAL   CINT = 0.001
      "DEFAULT MAXIMUM STEP LENGTH--------------------------------------------"
      MAXTERVAL   MAXT = 5E-5
      "DEFAULT NUMBER OF INTEGRATION STEPS PER COMMUNICATION INTERVAL----"
      NSTEPS         NSTP = 1
      "DEFAULT ALGORITHM IS RUNGE-KUTTA FOURTH ORDER----------------------"
      "IALG = 1 IS ADAMS-MOULTON, VARIABLE STEP, VARIABLE ORDER----------"
      "IALG = 2 IS GEARS STIFF, VARIABLE STEP, VARIABLE ORDER------------"
      "IALG = 3 IS RUNGE KUTTA FIRST ORDER OR EULER----------------------"
      "IALG = 4 IS RUNGE KUTTA SECOND ORDER------------------------------"
      "IALG = 5 IS RUNGE KUTTA FOURTH ORDER------------------------------"
      "IALG = 7 IS USER SUPPLIED ROUTINE---------------------------------"
      "IALG = 8 IS RUNGE-KUTTA-FEHLBERG SECOND ORDER---------------------"
      "IALG = 8 IS RUNGE-KUTTA-FEHLBERG FIFTH ORDER----------------------"
      ALGORITHM   IALG = 5

  INITIAL
      "INITIAL SECTION REQUIRED TO RUN THIS PROGRAM IN ISOLATION---------"
      JEQ       = JMOT + JLOAD

      "INITIAL SECTION REQUIRED BY THIS PROGRAM--------------------------"
      "CURRENT LOOP GAIN-------------------------------------------------"
      IGAN      = 10.4*VBUS/2/TRNGLE

  END $"OF INITIAL"

  DERIVATIVE
      "DUMMY SECTION-REMOVE WHEN INCLUDING IN LARGER SIMULATION----------"
      "VARIABLES REQUIRED TO RUN PROGRAM IN ISOLATION--------------------"
      CONSTANT    IINPUT = 1.0
      OMEGA     = INTEG(TORQ/JEQ,0.0)
      THETA     = INTEG(OMEGA,0.0)
      "END OF DUMMY SECTION----------------------------------------------"
```

```
    "CONSTANT SECTION-------------------------------------------------"
    "ALL THE VARIOUS CONSTANTS REQUIRED BY OTHER PROGRAMS--------------"
    "THESE ARE INCLUDED SO THAT THE RUN-TIME PROCEDURES DO NOT GIVE----"
    "VARIABLE NOT FOUND ERRORS-----------------------------------------"
    CONSTANT  K3     = 0,          G1     = 0,          G2     = 0
    CONSTANT  ENC    = 0,          FBAND  = 0,          BAND   = 0
    CONSTANT  DAMP   = 0,          SMP    = 0,          VTACH  = 0
    CONSTANT  VSCALE = 0,          INPLE  = 0,          VSCDMP = 0
    CONSTANT  SPRNGK = 0,          INLIM  = 0,          QUANT  = 0
    CONSTANT  PGAIN  = 0,          IGAIN  = 0,          BLSH   = 0
    CONSTANT  DSERVO = 0,          R322   = 0,          R124   = 0
    CONSTANT  R223   = 0,          R1213  = 0,          R1015  = 0
    CONSTANT  C19    = 0,          C21    = 0,          OUTPOT = 0
    "ALL THE VARIOUS CONSTANTS REQUIRED BY THIS AND OTHER PROGRAMS-----"
    CONSTANT  MOT    = 6100,       JMOT   = 0.0014,     JLOAD  = 0.00258
    CONSTANT  DM     = 50,         KT     = 0.62,       ILIM   = 50
    CONSTANT  PI     = 3.141593,   RADRPM = 9.55,       TSTOP  = 0.5
    "ALL THE VARIOUS CONSTANTS REQUIRED ONLY BY THIS PROGRAM-----------"
    "DEFAULT MOTOR CONSTANTS FOR BRU S-6100 MOTOR AND DM-50------------"
    CONSTANT  L1     = 5.5E-03,    R1     = 0.22,       VBUS   = 300
    CONSTANT  IPLE1  = 0.0009,     IZER1  = 0.00026,    TRNGLE = 11
    CONSTANT  KT1    = 0.41,       KT2    = 0.42,       KT3    = 0.40
    CONSTANT  ISMP   = 250E-06,    ISCALE = 5.0,        PLPAIR = 4
    "DEFAULT SWITCH POSITIONS...BLOCKS SWITCHED IN---------------------"
    CONSTANT  IS1    = 1,          IS4    = 0,          IS7    = 0
    CONSTANT  IS9    = 0,          IS17   = 0
    "END OF THE CONSTANT SECTION---------------------------------------"

    "CURRENT LOOP MAIN PROGRAM SECTION---------------------------------"
    "INPUT VOLTAGE SCALED FOR INPUT VIA OR BY-PASSING MICROPROCESSOR---"
    IY1      = RSW(IS1.EQ.1,IINPUT*ISCALE*10/DM,IINPUT*2)
    "BOUNDED CURRENT INPUT---------------------------------------------"
             IY2  = 1/ISCALE*BOUND(-ILIM,ILIM,IY1*ISCALE)
    "CURRENT SAMPLER AT 250 MICRO SECS---------------------------------"
    IY3      = ZOH(IY2,0.0,0.0,ISMP)
    IY4      = RSW(IS4.EQ.1,IY2,IY3)
    "TORQUE GENERATION SECTION-----------------------------------------"
     IY5     = IY4 - IY10
     IY6     = IGAN*IY5
     IY7     = RSW(IS7.EQ.1,IY6,LEDLAG(IZER1,IPLE1,IY6,0.0))
     IY8     = IY7 - KT1*OMEGA
    "CURRENT FROM VOLTS------------------------------------------------"
     IY9     = RSW(IS9.EQ.1,IY2*DM/10,1/R1*REALPL(L1/R1,IY8,0.0))
    IY10     = 10*IY9/DM
    "THREE PHASE CURRENT GENERATION------------------------------------"
    IY11     = IY9*SIN(ALPHA)
     IY12    = IY9*SIN(ALPHA-PI/3)
     IY13    = IY9*SIN(ALPHA-2*PI/3)
    "THREE PHASE TORQUE GENERATION-------------------------------------"
     IY14    = KT1*IY11*SIN(ALPHA)
     IY15    = KT2*IY12*SIN(ALPHA-PI/3)
     IY16    = KT3*IY13*SIN(ALPHA-2*PI/3)
     IY17    = RSW(IS17.EQ.1,KT*IY9,IY14+IY15+IY16)
    "ELECTRICAL ANGLE = POLE PAIRS* MECHANICAL ANGLE-------------------"
    ALPHA    = PLPAIR*THETA
    "END OF CURRENT LOOP MAIN PROGRAM SECTION--------------------------"
    "USEFUL PARAMETERS-------------------------------------------------"
    IMAG     = IY9
    TORQ     = IY17
    TERMT(T.GE.TSTOP)
  END $"OF DERIVATIVE"
END $"OF PROGRAM"
```

EMPTYBODE.CSL;1 [BODE.CSL]

```
PROGRAM PHASE AND GAIN
       "VERSION 1  AUGUST 1988-----------------------------------------------"
       "COPYRIGHT DAVID R. SEAWARD AUGUST 1988-------------------------------"
       "COMPUTE PHASE AND GAIN OF A GIVEN MODEL OF THE-----------------------"
       "ELECTROCRAFT BRU500 BY INTEGRATING OVER A COMPLETE CYCLE-------------"
       "CONTINUE INTEGRATE TILL PHASE CHANGE FROM CYCLE TO CYCLE-------------"
       "IS LESS THAN SOME PRESET MINIMUM-------------------------------------"
       "THE OPEN LOOP BODE PLOT IS ALSO CALCULATED FROM THE CLOSED-----------"
       "LOOP RESPONSE--------------------------------------------------------"
       "THE VARIABLES OF INTEREST ARE FOR CLOSE LOOP:------------------------"
       "PDG=PHASE IN DEGREES-------------------------------------------------"
       "PRD=PHASE IN RADIANS-------------------------------------------------"
       "GDB=GAIN IN Db-------------------------------------------------------"
       "GAN=NORMAL GAIN------------------------------------------------------"
       "WHZ=FREQUENCY IN Hz--------------------------------------------------"
       "WFR=FREQUENCY IN RADS-1----------------------------------------------"
       "AND FOR CLOSED LOOP--------------------------------------------------"
       "OPGAN=GAIN-----------------------------------------------------------"
       "OPGDB=GAIN IN Db-----------------------------------------------------"
       "OPPRD=PHASE IN RADIANS-----------------------------------------------"
       "OPPDG=PHASE IN DEGREES-----------------------------------------------"

       CINTERVAL      CINT   = 200 $"ENSURE NO SPURIOUS DATA OUTPUT----"
       NSTEPS    NSTP        = 1
       ALGORITHM      IALG   = 5
       CONSTANT       RMN    = 1.0E-30 ,    RMX    = 1.0E30
INITIAL
       "SET FIRST FREQUENCY AND PHASE----------------------------------------"
       W          = WMX
       FI      = 0.0
       "SET PREVIOUS---------------------------------------------------------"
       PP      = 1.0
       QP      = 1.0
       PDGP    = RMX
       "INITIALISE PLOT VARIABLES--------------------------------------------"
       PDG     = 0.0
       GDB     = 0.0
       OPGAN   = 0.01         $"PREVENTS LOG OF ZERO---------------------------"
       WFR     = 1000         $"STOP SPURIOUS PLOT-----------------------------"
       WHZ     = 1000         $"STOP SPURIOUS PLOT-----------------------------"

       "MODEL INITIAL SECTION------------------------------------------------"

       "END OF MODEL INITIAL SECTION-----------------------------------------"
END $" OF INITIAL "

DISCRETE
       INTERVAL       PERIOD = 0.0 $"PERIOD WILL BE CALCULATED-----------"
       CONSTANT       RADDEG = 57.2958,     PI = 3.141592654
       CONSTANT       EPDG   = 0.5,         EPM = 1.0E-7
       CONSTANT       KW     = 0.9,         TSTP = 10000.0
       CONSTANT       RADRPM = 9.55
     PROCEDURAL
       "CHANGE IN IN-PHASE AND QUADRATURE INTEGRALS OVER LAST CYCLE------"
       DLP     = P - PP
       DLQ     = Q - QP
       "IF RELATIVE CHANGE TOO SMALL FOR MACHINE  ACCURACY---------------"
       TERMT((DLP**2 + DLQ**2)/(P**2 + Q**2 + RMN) .LT. EPM**2)
       "SAVE NEW INTEGRALS AS PREVIOUS-----------------------------------"
       PP      = P
       QP      = Q
       "CALCULATE NEW PHASE AND GAIN-------------------------------------"
       PDGN    = ATAN2(DLQ, DLP + RMN)*RADDEG
       IF(DLQ.GT.0.AND.DLP+RMN.LT.0)PDGN =PDGN-360
       GDBN    = 10.0*ALOG10((DLP**2 + DLQ**2)*(W/(PI*XMAG))**2)
       "IF CHANGE IN PHASE NOT SMALL ENOUGH YET--------------------------"
       IF(ABS(PDGN - PDGP) .GT. EPDG) GO TO SKIP1
       "IGNORE RESULTS UNTIL AFTER SETTLING TIME-------------------------"
       IF(T .LT. TSETTL) GO TO SKIP1
       "TERMINATE ON FREQUENCY SWEEP-------------------------------------"
       TERMT(W .LE. WMN)
       "SAVE VALUES  NAME FOR PLOTTING-----------------------------------"
```

```
      PDG      = PDGN
      PRD      = PDGN/RADDEG
      GDB      = GDBN
      GAN      = 10**(GDB/20)
      WFR      = W
      WHZ      = W/2/PI
      "CALCULATE CLOSE LOOP VALUES--------------------------------------------"
      PTR(REALPT,IMAGPT = GAN,PRD)
      DENOM      = (1 - REALPT)**2 + IMAGPT**2
      NUMER      = REALPT - REALPT**2 -IMAGPT**2
      RTP(OPGAN,OPPRD = NUMER/DENOM, IMAGPT/DENOM)
      IF((NUMER/DENOM.LT.0).AND.(IMAGPT/DENOM.GT.0))OPPRD=OPPRD-2*PI
      OPGDB      = 20*ALOG10(OPGAN)
      OPPDG      = RADDEG*OPPRD
      "ADVANCE FREQUENCY GEOMETRICALLY----------------------------------------"
      W          = AMAX1(WMN, KW*W)
      "CALCULATE NEW PHASE FOR CONTINUITY-------------------------------------"
      "OF FORCING FUNCTION AT NEW FREQUENCY-----------------------------------"
      FI         = FI + T*(WFR - W)
      "ENSURE PREVIOUS PHASE SET TO FORCE AT LEAST TWO CYCLES-----------"
      PDGN       = RMX
      "FORCE A DATA LOGGING ACTION--------------------------------------------"
      CALL LOGD(.FALSE.)
      SKIP1..CONTINUE
      "RESET PREVIOUS PHASE FOR NEXT TIME-------------------------------------"
      PDGP       = PDGN
      "RECALCULATE NEW PERIOD AND STEP SIZE-----------------------------------"
      PERIOD     = 2.0*PI/W
      TSETTL     = 10*PERIOD
      MAXTC      = AMIN1(PERIOD/NSTPMN, MAXTXZ)
      END $" OF PROCEDURAL "
    TERMT(T .GT. TSTP)
END $" OF DISCRETE "
DERIVATIVE CONTIN
      MAXTERVAL      MAXTC  = 5.0E-05
      CONSTANT       WMN    = 6.28,        WMX    = 628
      CONSTANT       XMAG   = 0.01
      X       = XMAG*SIN(W*T + FI)
      "DEFINE MODEL-----------------------------------------------------------"

      "END OF MODEL-----------------------------------------------------------"
      "INTEGRATE FOR IN-PHASE AND QUADRATURE COMPONENTS-----------------"
      CONSTANT       MAXTXZ = 5.0E-05    , NSTPMN = 10.0
      P       = INTEG(Y*SIN(W*T + FI), 0.0)
      Q       = INTEG(Y*COS(W*T + FI), 0.0)
END $" OF CONTINUOUS SECTION "
END $" OF PROGRAM "
```

BRU_BAND_DAMP_CONTROL1_FULL.CSL;1   [BBD1F.CSL]

```
PROGRAM BRU500_SIMULATION_OF_BANDWIDTH_DAMPING_CONTROL
       "VERSION 1 AUGUST 1988-------------------------------------------------"
       "A VELOCITY LOOP WITH A FULL LOAD AND CURRENT LOOP SPECIFICATION---"
       "COPYRIGHT DAVID R. SEAWARD AUGUST 1988----------------------------"
       "SIMULATION OF THE ELECTROCRAFT BRU500 SERIES OF BRUSHLESS MACHINES"
       "TO BE USED WITH ITS DATA FILE-----------------------------------------"

       "THIS PROGRAM SIMULATES THE ELECTROCRAFT BANDWIDTH/DAMPING CONTROL-"
       "SOFTWARE SCHEME--------------------------------------------------------"
       "IT MAY BE USED ALONE OR PARTS EXTRACTED TO FORM PART OF A LARGER--"
       "SIMULATION PROGRAM-----------------------------------------------------"
       "THE SWITCHES SHORT OUT A BLOCK WHEN SET TO ONE-------------------"
       "THE SWITCH NUMBER REFERS TO THE VARIABLE IT SWITCHES-------------"
       "IE. S2 SWITCHES OUT THE BLOCK BETWEEN Y1 AND Y2------------------"
       "DEFAULT COMMUNICATION INTERVAL----------------------------------------"
       CINTERVAL      CINT   = 0.001
       "DEFAULT MAXIMUM STEP LENGTH-------------------------------------------"
       MAXTERVAL      MAXT   = 5E-5
       "DEFAULT NUMBER OF INTEGRATION STEPS PER COMMUNICATION INTERVAL----"
       NSTEPS         NSTP   = 1
       "DEFAULT ALGORITHM IS RUNGE-KUTTA FOURTH ORDER--------------------"
       ALGORITHM      IALG   = 5
  INITIAL
```

```
     "TOTAL EQUIVALENT INERTIA-------------------------------------------------"
     JEQ       = JMOT + JLOAD
     "FEEDBACK SCALED IN PULSES------------------------------------------------"
     KENC      = 2*ENC/PI
     "VELOCITY LOOP GAIN FACTORS-----------------------------------------------"
     K1        = BAND*G1/DAMP/65536
     K2        = BAND*DAMP*G2/16777216
     "SOFTWARE INPUT FILTER----------------------------------------------------"
     TFPLE     = 1/(2*PI*FBAND)
     "CURRENT LOOP GAIN--------------------------------------------------------"
     IGAN      = 10.4*VBUS/2/TRNGLE
     AZ0       = 0
     AZ1       = 0
     AZ2       = 0
     AZ3       = 0
     AZ4       = 0
     BZ0       = 0
     BZ1       = 0
     BZ2       = 0
     BZ3       = 0
     BZ4       = 0
     VELFB     = 0
     VELDEM    = 0
     PERROR    = 0
     BDZ1      = 0
     BDZ2      = 0
     BDZ3      = 0
     BDZ4      = 0
     Y1        = 0
     IINPUT    = 0
END $"OF INITIAL"

DERIVATIVE
     "CONSTANT SECTION---------------------------------------------------------"
     "ALL THE VARIOUS CONSTANTS REQUIRED BY OTHER PROGRAMS--------------"
     "THESE ARE INCLUDED SO THAT THE RUN-TIME PROCEDURES DO NOT GIVE----"
     "VARIABLE NOT FOUND ERRORS------------------------------------------------"
     CONSTANT   OUTPOT = 0,       R124   = 0,        IGAIN  = 0
     CONSTANT   PGAIN  = 0,       DSERVO = 0,        R322   = 0
     CONSTANT   R223   = 0,       R1213  = 0,        R1015  = 0
     CONSTANT   C19    = 0,       C21    = 0
     "ALL THE VARIOUS CONSTANTS REQUIRED BY THIS AND OTHER PROGRAMS-----"
     CONSTANT   PI     = 3.141593, RADRPM = 9.55,    TSTOP  = 0.5
     CONSTANT   ILIM   = 50,      DM     = 50
     "ALL THE VARIOUS CONSTANT REQUIRED ONLY BY THIS PROGRAM------------"
     "DEFAULT INPUT MAGNITUDES-------------------------------------------------"
     CONSTANT   XSTP   = 1.0,     XPLS   = 1.0,     XRMP   = 1.0
     CONSTANT   XSIN   = 1.0,     PULS   = 1.0,     WDTH   = 0.5
     CONSTANT   FI     = 0.0,     W      = 62.8 $"10 Hz"
     "DEFAULT INPUT START TIMES------------------------------------------------"
     CONSTANT   TSTP   = 0.0,     TPLS   = 0.0,     TRMP   = 0.0
     CONSTANT   TSIN   = 0.0
     "DEFAULT INPUT STOP TIMES-------------------------------------------------"
     CONSTANT   STSTP  = 100,     STPLS  = 100,     STRMP  = 100
     CONSTANT   STSIN  = 100
     "DEFAULT SWITCH SETTINGS--------------------------------------------------"
     CONSTANT   XS1    = 1.0,     XS2    = 0.0,     XS3    = 0.0
     CONSTANT   XS4    = 0.0
     CONSTANT   S1     = 0.0,     S2     = 0.0
     "DEFAULT MOTOR CONSTANTS FOR BRU S-6100 MOTOR AND DM-50------------"
     CONSTANT   ENC    = 2000,    INPLE  = 0.0008,  MOT    = 6100
     CONSTANT   INLIM  = 10.0,    QUANT  = 1
     CONSTANT   K3     = 0.001,   G1     = 1024,    G2     = 6711
     "SERVO SET UP CHARACTERISTICS DEFAULT-------------------------------------"
     CONSTANT   BAND   = 50,      DAMP   = 50,      FBAND  = 300
     CONSTANT   VSCALE = 400
     "END OF THE CONSTANT SECTION----------------------------------------------"

     "INPUT SECTION------------------------------------------------------------"
    PROCEDURAL(X=X1,X2,X3,X4)
     IF(T.GE.STSTP)GOTO JMP1
     X1        = XSTP*XS1*STEP(TSTP)
    JMP1..CONTINUE
     IF(T.GE.STPLS)GOTO JMP2
     X2        = XPLS*XS2*(2*PULSE(TPLS,PULS,WDTH)-1)
```

```
  JMP2..CONTINUE
   IF(T.GE.STRMP)GOTO JMP3
   X3        = XRMP*XS3*RAMP(TRMP)
  JMP3..CONTINUE
   IF(T.GE.STSIN)GOTO JMP4
   X4        = XSIN*XS4*SIN(W*T+FI)
  JMP4..CONTINUE
   X         = X1+X2+X3+X4
END$"OF PROCEDURAL"
    "END OF INPUT SECTION-----------------------------------------------"

    "BANDWIDTH DAMPING ALGORITHM MAIN PROGRAM--------------------------"
    "INPUT POLE DUE TO ANALOGUE FILTERING------------------------------"
    Y1        = RSW(S1.EQ.1,X,REALPL(INPLE,X,0.0))
    "INPUT VOLTAGE LIMIT..USUALLY 10VOLTS------------------------------"
    Y2      = RSW(S2.EQ.1,Y1,BOUND(-INLIM,INLIM,Y1))
    "VOLTAGE SCALING INTO R/MIN----------------------------------------"
    Y3        = VSCALE*Y2
    "INPUT IN PULSES---------------------------------------------------"
    Y4        = 4*ENC/60*INTEG(Y3,0.0)
    "CURRENT LOOP INPUT DELIVERED FROM DIGITAL ALGORITHM---------------"
    IINPUT    = REALPL(TFPLE,BDZ3,0.0)

    "CURRENT LOOP MAIN PROGRAM SECTION---------------------------------"
    "CONSTANT SECTION--------------------------------------------------"
    "DEFAULT MOTOR CONSTANTS FOR BRU S-6100 MOTOR AND DM-50------------"
    CONSTANT   L1          = 5.5E-03,  R1     = 0.22,     VBUS   = 300
    CONSTANT   IPLE1       = 0.0009,   IZER1  = 0.00026,  TRNGLE = 11
    CONSTANT   KT1         = 0.41,     KT2    = 0.42,     KT3    = 0.40
    CONSTANT   ISMP        = 250E-06,  ISCALE = 5.0,      PLPAIR = 4
    CONSTANT   KT          = 0.62
    "DEFAULT SWITCH POSITIONS...BLOCKS SWITCHED IN---------------------"
    CONSTANT   IS1         = 1,        IS4    = 0,        IS7    = 0
    CONSTANT   IS9         = 0,        IS17   = 0
    "END OF THE CONSTANT SECTION---------------------------------------"
    "INPUT VOLTAGE SCALED FOR INPUT VIA OR BY-PASSING MICROPROCESSOR---"
    IY1       = RSW(IS1.EQ.1,IINPUT*ISCALE*10/DM,IINPUT*2)
    "BOUNDED CURRENT INPUT---------------------------------------------"
    IY2       = 1/ISCALE*BOUND(-ILIM,ILIM,IY1*ISCALE)
    "CURRENT SAMPLER AT 250 MICRO SECS---------------------------------"
    IY3       = ZOH(IY2,0.0,0.0,ISMP)
    IY4       = RSW(IS4.EQ.1,IY2,IY3)
    "TORQUE GENERATION SECTION-----------------------------------------"
    IY5       = IY4 - IY10
    IY6       = IGAN*IY5
    IY7       = RSW(IS7.EQ.1,IY6,LEDLAG(IZER1,IPLE1,IY6,0.0))
    IY8       = IY7 - KT1*OMEGA
    "CURRENT FROM VOLTS------------------------------------------------"
    IY9       = RSW(IS9.EQ.1,IY2*DM/10,1/R1*REALPL(L1/R1,IY8,0.0))
    IY10      = 10*IY9/DM
    "THREE PHASE CURRENT GENERATION------------------------------------"
    IY11      = IY9*SIN(ALPHA)
     IY12     = IY9*SIN(ALPHA-PI/3)
     IY13     = IY9*SIN(ALPHA-2*PI/3)
    "THREE PHASE TORQUE GENERATION-------------------------------------"
    IY14      = KT1*IY11*SIN(ALPHA)
    IY15      = KT2*IY12*SIN(ALPHA-PI/3)
    IY16      = KT3*IY13*SIN(ALPHA-2*PI/3)
    IY17      = RSW(IS17.EQ.1,KT*IY9,IY14+IY15+IY16)
    "ELECTRICAL ANGLE = POLE PAIRS* MECHANICAL ANGLE-------------------"
    ALPHA     = PLPAIR*THETA
    "END OF CURRENT LOOP MAIN PROGRAM SECTION--------------------------"

    "LOAD MAIN PROGRAM SECTION-----------------------------------------"
    "CONSTANT SECTION--------------------------------------------------"
    CONSTANT   JMOT   = 0.0014,        JLOAD  = 0.00258
    "DEFAULT LOAD CONSTANTS--------------------------------------------"
    CONSTANT   VSCDMP = 1.0E-04,       BLSH   = 0.0,     SPRNGK = 39000
    CONSTANT   VTACH  = 0.05
    "END OF THE CONSTANT SECTION---------------------------------------"
    "LOAD INERTIA/TORSIONAL SPRING 2ND ORDER EFFECT NEGLECTED----------"
    "SIMPLE DIRECTLY CONNECTED LOAD CONSIDERED-------------------------"
    LY1       = TORQ - LY3
    LY2       = 1/JEQ*INTEG(LY1,0.0)
    LY3       = VSCDMP*LY2
```

```
      "END OF LOAD MAIN PROGRAM SECTION----------------------------------"
      "VARIABLES OF INTEREST--------------------------------------------"
      "SPEED OUTPUT IN RADS-1-------------------------------------------"
      OMEGA    = LY2
      "MECHANICAL ANGLE-------------------------------------------------"
      THETA    = INTEG(OMEGA,0.0)
      "POSITION IN PULSES-----------------------------------------------"
      ENCODE   = KENC*THETA
      "OUTPUT SCALED IN RPM FROM DIGITAL TACHO--------------------------"
      YRPM     = OMEGA*RADRPM
      "OUTPUT SCALED BY VTACH FROM DIGITAL TACHO------------------------"
       YTACH   = OMEGA*VTACH
      "INPUT TO DIGITAL SECTION IN PULSES-------------------------------"
      DEMAND   = Y4
      "CURRENT MAGNITUDE------------------------------------------------"
      IMAG     = IY9
      "TORQUE PRODUCED--------------------------------------------------"
      TORQ     = IY17
      TERMT(T.GE.TSTOP)
 END $"OF DERIVATIVE"

 DISCRETE CNTRL
       INTERVAL SMP=0.001
       PROCEDURAL
      AZ1       = ENCODE - AZ0 + AZ3 $"POSITION DIFFERENCE + REMAINDER-----"
      AZ2       = QNTZR(QUANT,AZ1)   $"QUANTISATION TO ONE PULSE-----------"
      AZ3       = AZ1 - AZ2          $"THE REMAINDER-----------------------"
      AZ0       = ENCODE             $"RESET LAST VALUE--------------------"
      AZ4       = QNTZR(QUANT,ENCODE)$"QUANTISED POSITION SIGNAL-----------"
      VELFB     = AZ2/SMP            $"SCALE TO PULSES/SEC-----------------"
      AZ5       = VELFB-AZ6
      AZ6       = VELFB
      ACCLFB    = AZ5/SMP
      BZ1       = DEMAND - BZ0 + BZ3 $"POSITION DIFFERENCE + REMAINDER-----"
      BZ2       = QNTZR(QUANT,BZ1)   $"QUANTISATION TO ONE PULSE-----------"
      BZ3       = BZ1 - BZ2          $"THE REMAINDER-----------------------"
      BZ0       = DEMAND             $"RESET LAST VALUE--------------------"
      BZ4       = QNTZR(QUANT,DEMAND)$"QUANTISED POSITION SIGNAL-----------"
      VELDEM    = BZ2/SMP            $"SCALE TO PULSE/SEC------------------"
      PERROR    = QNTZR(QUANT,DEMAND-ENCODE)
      "THIS IS THE BANDWIDTH DAMPING CONTROL ALGORITHM------------------"
      BDZ1      = 4*VELDEM - 4*VELFB - K3*ACCLFB/K1
      BDZ2      = BDZ1*K1*K2*0.078125 $"ERROR TIMES GAINS------------------"
      BDZ3      = BDZ4 + BDZ2*SMP   $"SOFTWARE INTEGRATION-----------------"
      IF(BDZ3.GT.ILIM*10/DM)BDZ3=BDZ4  $"LIMITTED INTEGRATION------------"
      BDZ4      = BDZ3
      "IINPUT   = BDZ3*SOFTWARE FILTER-----------------------------------"
      "END OF BANDWIDTH DAMPING MAIN PROGRAM----------------------------"
       END $"OF PROCEDURAL"
  END $"OF DISCRETE"
END $"OF PROGRAM"
```

BRU_BAND_DAMP_CONTROL2.CSL;1   [BBD2.CSL]

```
 DISCRETE CNTRL
       INTERVAL SMP=0.001
       PROCEDURAL
      AZ1       = ENCODE - AZ0 + AZ3 $"POSITION DIFFERENCE + REMAINDER-----"
      AZ2       = QNTZR(QUANT,AZ1)   $"QUANTISATION TO ONE PULSE-----------"
      AZ3       = AZ1 - AZ2          $"THE REMAINDER-----------------------"
      AZ0       = ENCODE             $"RESET LAST VALUE--------------------"
      AZ4       = QNTZR(QUANT,ENCODE)$"QUANTISED POSITION SIGNAL-----------"
      VELFB     = AZ2/SMP            $"SCALE TO PULSES/SEC-----------------"
      BZ1       = DEMAND - BZ0 + BZ3 $"POSITION DIFFERENCE + REMAINDER-----"
      BZ2       = QNTZR(QUANT,BZ1)   $"QUANTISATION TO ONE PULSE-----------"
      BZ3       = BZ1 - BZ2          $"THE REMAINDER-----------------------"
      BZ0       = DEMAND             $"RESET LAST VALUE--------------------"
      BZ4       = QNTZR(QUANT,DEMAND)$"QUANTISED POSITION SIGNAL-----------"
      VELDEM    = BZ2/SMP            $"SCALE TO PULSE/SEC------------------"
      PERROR    = QNTZR(QUANT,DEMAND-ENCODE)
      "THIS IS THE BANDWIDTH DAMPING CONTROL ALGORITHM------------------"
      BDZ1      = 4*BZ4 - 4*AZ4 - K3*VELFB/K1
      BDZ2      = BDZ1*K1*K2*0.078125 $"ERROR TIMES GAINS------------------"
```

```
      BDZ3      = (BDZ2-BDZ4)        $"SOFTWARE DERIVATIVE----------------"
      BDZ4      = BDZ2
      BDZ5      = BDZ6 + BDZ3        $"SOFTWARE INTEGRATION---------------"
      IF(BDZ5.GT.ILIM*10/DM)BDZ5=BDZ6    $"LIMITTED INTEGRATION----------"
      BDZ6      = BDZ5
      "IINPUT = BDZ2*SOFTWARE FILTER-------------------------------------"
      "END OF BANDWIDTH DAMPING PROGRAM----------------------------------"
       END $"OF PROCEDURAL"
  END $"OF DISCRETE"
```

BRU_P_I_CONTROL.CSL;1 [BPI.CSL]

```
 DISCRETE CNTRL
       INTERVAL SMP=0.001
       PROCEDURAL
      AZ1       = ENCODE - AZ0 + AZ3 $"POSITION DIFFERENCE + REMAINDER-----"
      AZ2       = QNTZR(QUANT,AZ1)   $"QUANTISATION TO ONE PULSE-----------"
      AZ3       = AZ1 - AZ2          $"THE REMAINDER-----------------------"
      AZ0       = ENCODE             $"RESET LAST VALUE--------------------"
      AZ4       = QNTZR(QUANT,ENCODE)$"QUANTISED POSITION SIGNAL-----------"
      VELFB     = AZ2/SMP            $"SCALE TO PULSES/SEC-----------------"
      BZ1       = DEMAND - BZ0 + BZ3 $"POSITION DIFFERENCE + REMAINDER-----"
      BZ2       = QNTZR(QUANT,BZ1)   $"QUANTISATION TO ONE PULSE-----------"
      BZ3       = BZ1 - BZ2          $"THE REMAINDER-----------------------"
      BZ0       = DEMAND             $"RESET LAST VALUE--------------------"
      BZ4       = QNTZR(QUANT,DEMAND)$"QUANTISED POSITION SIGNAL-----------"
      VELDEM    = BZ2/SMP            $"SCALE TO PULSE/SEC------------------"
      PERROR    = QNTZR(QUANT,DEMAND-ENCODE)
      "THIS IS THE PROPORTIONAL/INTEGRAL ALGORITHM-----------------------"
      PIZ1      = 4*SMP*(VELDEM-VELFB)  $"VELOCITY ERROR-------------------"
      PIZ3      = PIZ2 + 2000*IGAIN*PIZ1*SMP
      IF(PIZ6.GT.ILIM*10/DM)PIZ3=PIZ2 $"LIMITTED INTEGRATION-------------"
      PIZ2      = PIZ3            $"THE INTEGRAL CALCULATION---------"
      PIZ4      = 64*PGAIN*PIZ1          $"THE PROPORTIONAL CALCULATION-----"
      PIZ5      = PIZ4 +PIZ3
      PIZ6      = PIZ5*1.1921E-06        $"SCALED FOR CURRENT LOOP----------"
       END $"OF PROCEDURAL"
      "END OF PROPORTIONAL INTEGRAL ALGORITHM MAIN PROGRAM---------------"
  END $"OF DISCRETE"
```

MOLINS_DSERVO_CONTROL.CSL;1 [BM.CSL]

```
PROGRAM BRU500_SIMULATION_WITH_THE_MOLINS_DSERVO
      "VERSION 1 JULY 1988-----------------------------------------------"
      "COPYRIGHT DAVID R. SEAWARD JULY 1988------------------------------"
      "SIMULATION OF THE ELECTROCRAFT BRU500 SERIES OF BRUSHLESS MACHINES"
      "TO BE USED WITH ITS DATA FILE-------------------------------------"
      "THIS PROGRAM SIMULATES DSERVO VERSIONS A14/A15--------------------"
      "SOFTWARE SCHEME WITH A FULL BRU CURRENT LOOP----------------------"
      "IT MAY BE USED ALONE OR PARTS EXTRACTED TO FORM PART OF A LARGER--"
      "SIMULATION PROGRAM------------------------------------------------"
      CINTERVAL     CINT = 0.001
      MAXTERVAL     MAXT = 5E-5
      NSTEPS        NSTP = 1
      ALGORITHM     IALG = 5
  INITIAL
      JEQ       = JMOT +JLOAD
      KENC      = 2*ENC/PI
      FILT2     = SQRT((R1213/R1015+1)*4*C19/C21)
      FILT1     = FILT2/2/R1213/C19
      AZ0       = 0
      AZ1       = 0
      AZ2       = 0
      AZ3       = 0
      AZ4       = 0
      BZ0       = 0
      BZ1       = 0
      BZ2       = 0
      BZ3       = 0
      BZ4       = 0
      VELFB     = 0
      VELDEM    = 0
```

```
    PERROR    = 0
    MZ1       = 0
    MZ2       = 0
    MZ3       = 0
    MZ4       = 0
    MZ5       = 0
    Y1        = 0
    IINPUT    = 0
END $"OF INITIAL"

DERIVATIVE
    "DUMMY CURRENT LOOP-----------------------------------------------"
    "USED ONLY TO TEST THIS ALGORITHM---------------------------------"
    "CONVERTS 0-10V INTO A VELOCITY-----------------------------------"
    CONSTANT   JLOAD = 0.00258,  KT     = 0.62,    DM     = 50
    CONSTANT   JMOT  = 0.0014
    DUM1     = BOUND(-ILIM,ILIM,IINPUT*DM/10)
    OMEGA    = KT/JEQ*INTEG(DUM1,0.0)
    "POSITION IN RADIANS---------------------------------------------"
    THETA    = INTEG(OMEGA,0.0)

    "CONSTANT SECTION-------------------------------------------------"
    "ALL THE VARIOUS CONSTANTS REQUIRED BY OTHER PROGRAMS-------------"
    CONSTANT   ISMP   = 0,        KT1    = 0,      INPLE  = 0
    CONSTANT   KT2    = 0,        KT3    = 0,      PLPAIR = 0
    CONSTANT   VSCDMP = 0,        SPRNGK = 0,      K3     = 0
    CONSTANT   G1     = 0,        G2     = 0,      DAMP   = 0
    CONSTANT   BAND   = 0,        BLSH   = 0,      VBUS   = 0
    CONSTANT   ISCALE = 0,        L1     = 0,      R1     = 0
    CONSTANT   IPLE1  = 0,        IZER1  = 0,      TRNGLE = 0
    CONSTANT   PGAIN  = 0,        IGAIN  = 0,      FBAND  = 0
    CONSTANT   PI     = 3.141593, RADRPM = 9.55,   TSTOP  = 0.5
    CONSTANT   ILIM   = 50,       VTACH  = 0.05
    "ALL THE VARIOUS CONSTANT REQUIRED ONLY BY THIS PROGRAM-----------"
    CONSTANT   XSTP   = 1.0,      XPLS   = 1.0,    XRMP   = 1.0
    CONSTANT   XSIN   = 1.0,      PULS   = 1.0,    WDTH   = 0.5
    CONSTANT   FI     = 0.0,      W      = 62.8 $"10 Hz"
    CONSTANT   TSTP   = 0.0,      TPLS   = 0.0,    TRMP   = 0.0
    CONSTANT   TSIN   = 0.0
    CONSTANT   STSTP  = 100,      STPLS  = 100,    STRMP  = 100
    CONSTANT   STSIN  = 100
    CONSTANT   XS1    = 1.0,      XS2    = 0.0,    XS3    = 0.0
    CONSTANT   XS4    = 0.0,      S5     = 0.0
    CONSTANT   ENC    = 2000,     MOT    = 6100
    CONSTANT   INLIM  = 10.0,     QUANT  = 1
    CONSTANT   R322   = 10000,    R124   = 10000,  R223   = 10000
    CONSTANT   R1213  = 18000,    R1015  = 13000,  DSERVO = 15
    CONSTANT   C19    = 22E-09,   C21    = 100E-09, OUTPOT = 0.36
    CONSTANT   VSCALE = 400
    "END OF THE CONSTANT SECTION--------------------------------------"

    "INPUT SECTION----------------------------------------------------"
    PROCEDURAL(X=X1,X2,X3,X4)
      IF(T.GE.STSTP)GOTO JMP1
      X1        = XSTP*XS1*STEP(TSTP)
     JMP1..CONTINUE
      IF(T.GE.STPLS)GOTO JMP2
      X2        = XPLS*XS2*(2*PULSE(TPLS,PULS,WDTH)-1)
     JMP2..CONTINUE
      IF(T.GE.STRMP)GOTO JMP3
      X3        = XRMP*XS3*RAMP(TRMP)
     JMP3..CONTINUE
      IF(T.GE.STSIN)GOTO JMP4
      X4        = XSIN*XS4*SIN(W*T+FI)
     JMP4..CONTINUE
      X         = X1+X2+X3+X4
    END$"OF PROCEDURAL"
    "END OF INPUT SECTION---------------------------------------------"

    "MOLINS DSERVO MAIN PROGRAM---------------------------------------"
    "VOLTAGE SCALING INTO R/MIN---------------------------------------"
    Y3          = VSCALE*X
    "INPUT IN PULSES--------------------------------------------------"
    Y4          = 4*ENC/60*INTEG(Y3,0.0)
    "CURRENT LOOP INPUT DELIVERED FROM DIGITAL ALGORITHM--------------"
```

```
      Y5        = CMPXPL(1/FILT1**2,FILT2/FILT1,MZ5,0.0,0.0)
      IINPUT    = RSW(S5.EQ.1,MZ5,Y5)
      "POSITION IN PULSES-------------------------------------------------"
      ENCODE    = KENC*THETA
      "VARIABLES OF INTEREST----------------------------------------------"
      "OUTPUT SCALED IN RPM FROM DIGITAL TACHO----------------------------"
       YRPM     = OMEGA*RADRPM
      "OUTPUT SCALED BY VTACH FROM DIGITAL TACHO--------------------------"
       YTACH    = OMEGA*VTACH
      "INPUT TO DIGITAL SECTION IN PULSES---------------------------------"
      DEMAND    = Y4
      TERMT(T.GE.TSTOP)
  END $"OF DERIVATIVE"

 DISCRETE CNTRL
       INTERVAL SMP=0.001
       PROCEDURAL
      AZ1       = ENCODE - AZ0 + AZ3 $"POSITION DIFFERENCE + REMAINDER-----"
      AZ2       = QNTZR(QUANT,AZ1)   $"QUANTISATION TO ONE PULSE-----------"
      AZ3       = AZ1 - AZ2          $"THE REMAINDER-----------------------"
      AZ0       = ENCODE             $"RESET LAST VALUE--------------------"
      AZ4       = QNTZR(QUANT,ENCODE)$"QUANTISED POSITION SIGNAL-----------"
      VELFB     = AZ2/SMP            $"SCALE TO PULSES/SEC-----------------"
      BZ1       = DEMAND - BZ0 + BZ3 $"POSITION DIFFERENCE + REMAINDER-----"
      BZ2       = QNTZR(QUANT,BZ1)   $"QUANTISATION TO ONE PULSE-----------"
      BZ3       = BZ1 - BZ2          $"THE REMAINDER-----------------------"
      BZ0       = DEMAND             $"RESET LAST VALUE--------------------"
      BZ4       = QNTZR(QUANT,DEMAND)$"QUANTISED POSITION SIGNAL-----------"
      VELDEM    = BZ2/SMP            $"SCALE TO PULSE/SEC------------------"
      PERROR    = QNTZR(QUANT,DEMAND-ENCODE)
      "THIS IS THE MOLINS DSERVO\ A14/A15 ALGORITHM-----------------------"
      MZ1       = 4*SMP*(VELDEM-VELFB)  $"VELOCITY ERROR-------------------"
      IF(DSERVO.EQ.15)MZ1=SMP*(VELDEM-VELFB)
      MZ3       = MZ2 + MZ1
      MZ2       = MZ3                         $"THE INTEGRAL CALCULATION---------"
      MZ4       = 0.078125*R322*(MZ1/R124+MZ3/R223)
      MZ5       = R1213*OUTPOT*MZ4/R1015
       END $"OF PROCEDURAL"
      "END OF MOLINS DSERVO MAIN PROGRAM----------------------------------"
  END $"OF DISCRETE"
END $"OF PROGRAM"
```

BRU.DAT;1 [BRU.DAT]

```
"RUN TIME FILE FOR USE WITH THE BRU500 SUITE OF SIMULATION PROGRAMS"
"VERSION 1.. JULY 1988--------------------------------------------"
"COPYRIGHT DAVID.R.SEAWARD JULY 1988------------------------------"
"TO ENVOKE USE.. ASSIGN BRU.DAT FOR0XX  ..WHERE XXIS A NUMBER------"
"AT RUN TIME USE ACSL>SET CMD=XX----------------------------------"
PROCED S3016
SET MOT=3016,    K3=0.001,         G1=2375,           G2=2621,         ENC=2000
SET FBAND=300,   KT=0.28,          BAND=50,           DAMP=50,         ISMP=0.00025
SET DM=25,       JMOT=9E-05,       JLOAD=0.01,        ILIM=20,         SMP=0.001
SET VTACH=.04625,KT1=0.186,        KT2=0.187,         KT3=0.185,       VSCALE=400
SET PLPAIR=2,    INPLE=0.0008,     VSCDMP=1.0E-04,    SPRNGK=1160,     INLIM=10
SET QUANT=1,     PGAIN=500,        IGAIN=0,           BLSH=0,          VBUS=300
SET ISCALE=2.5,  L1=2.5E-03,       R1=0.7,            IPLE1=0.0009,    IZER1=0.00026
SET TRNGLE=11,   DSERVO=15,        R322=50000,        R124=2000,       R223=20000
SET R1213=18000, R1015=13000,      C19=22.0E-09,      C21=100.0E-09,   OUTPOT=0.36
END

PROCED S4030
SET MOT=4030,    K3=0.001,         G1=1200,           G2=20972,        ENC=2000
SET FBAND=300,   KT=0.5,           BAND=50,           DAMP=50,         ISMP=0.00025
SET DM=25,       JMOT=0.00028,     JLOAD=0.00262,     ILIM=25,         SMP=0.001
SET VTACH=.04625,KT1=0.33,         KT2=0.32,          KT3=0.34,        VSCALE=400
SET PLPAIR=3,    INPLE=0.0008,     VSCDMP=1.0E-04,    SPRNGK=3800,     INLIM=10
SET QUANT=1,     PGAIN=500,        IGAIN=0,           BLSH=0,          VBUS=300
SET ISCALE=2.5,  L1=5.5E-03,       R1=0.95,           IPLE1=0.0009,    IZER1=0.00026
SET TRNGLE=11,   DSERVO=15,        R322=50000,        R124=2000,       R223=20000
SET R1213=18000, R1015=13000,      C19=22.0E-09,      C21=100.0E-09,   OUTPOT=0.36
END
```

```
PROCED S4050
SET MOT=4050,    K3=0.001,          G1=2376,          G2=3775,         ENC=2000
SET FBAND=300,   KT=0.5,            BAND=50,          DAMP=50,         ISMP=0.00025
SET DM=50,       MOT=0.00056,       JLOAD=0.00262,    ILIM=50,         SMP=0.001
SET VTACH=.04625,KT1=0.32,          KT2=0.33,         KT3=0.34,        VSCALE=400
SET PLPAIR=3,    INPLE=0.0008,      VSCDMP=1.0E-04,   SPRNGK=3200,     INLIM=10
SET QUANT=1,     PGAIN=500,         IGAIN=0,          BLSH=0,          VBUS=300
SET ISCALE=5.0,  L1=2.75E-03,       R1=0.4,           IPLE1=0.0009,    IZER1=0.00026
SET TRNGLE=11,   DSERVO=15,         R322=50000,       R124=2000,       R223=20000
SET R1213=18000, R1015=13000,       C19=22.0E-09,     C21=100.0E-09,   OUTPOT=0.36
END

PROCED S6100
SET MOT=6100,    K3=0.001,          G1=1200,          G2=8192,         ENC=2000
SET FBAND=300,   KT=0.62,           BAND=50,          DAMP=50,         ISMP=0.00025
SET DM=50,       JMOT=0.0014,       JLOAD=0.00258,    ILIM=50,         SMP=0.001
SET VTACH=.04625,KT1=0.40,          KT2=0.41,         KT3=0.42,        VSCALE=400
SET PLPAIR=4,    NPLE=0.0008,       VSCDMP=1.0E-04,   SPRNGK=41000,    INLIM=10
SET QUANT=1,     PGAIN=500,         IGAIN=0,          BLSH=0,          VBUS=300
SET ISCALE=5.0,  L1=2.75E-03,       R1=0.22,          IPLE1=0.0009,    IZER1=0.00026
SET TRNGLE=11,   DSERVO=15,         R322=50000,       R124=2000,       R223=20000
SET R1213=18000, R1015=13000,       C19=22.0E-09,     C21=100.0E-09,   OUTPOT=0.36
END

PROCED S6200
SET MOT=6200,    K3=0.001,          G1=2145,          G2=4876,         ENC=2000
SET FBAND=300,   KT=0.62,           BAND=50,          DAMP=50,         ISMP=0.00025
SET DM=100,      JMOT=0.0024,       JLOAD=0.00258,    ILIM=100,        SMP=0.001
SET VTACH=.04625,KT1=0.40,          KT2=0.41,         KT3=0.42,        VSCALE=400
SET PLPAIR=4,    INPLE=0.0008,      VSCDMP=1.0E-04,   SPRNGK=28000,    INLIM=10
SET QUANT=1,     PGAIN=500,         IGAIN=0,          BLSH=0,          VBUS=300
SET ISCALE=10,   L1=1.2E-03,        R1=0.09,          IPLE1=0.0009,    IZER1=0.00026
SET TRNGLE=11,   DSERVO=15,         R322=50000,       R124=2000,       R223=20000
SET R1213=18000, R1015=13000,       C19=22.0E-09,     C21=100.0E-09,   OUTPOT=0.36
END

"SET UP PROCEDURE FOR BODE PLOTS OF SYSTEMS------------------------"
PROCED LOG
OUTPUT "CLEAR"
SET NCIOUT=2
PREPAR "CLEAR"
OUTPUT WHZ,PDG,GDB
PREPAR WHZ,PDG,GDB
SET TSTP=10000
END
"PLOTTING PROCEDURE FOR LOG PLOTS----------------------------------"
PROCED PLOTLOG
PLOT "XLOG","XHI"=100,"XLO"=1
PLOT GDB,"HI"=10,"LO"=-40,PDG,"HI"=0,"LO"=-300
D MOT,JLOAD,ILIM,XMAG
END
"SET UP PROCEDURE FOR NORMAL PLOTS OF SYSTEMS----------------------"
PROCED NORM
SET CINT=0.0001
SET NCIOUT=1000
OUTPUT T,OMEGA
PREPAR T,OMEGA
END
"PLOTTING PROCEDURE FOR SPEED PLOTS--------------------------------"
PROCED PLOTNORM
PLOT OMEGA
D MOT,JLOAD,ILIM
END

SET CMD=5
```

APPENDIX 5

A5.0 THE "C" PROGRAMS WRITTEN FOR THE PC23

A5.1 INTRODUCTION

The PC23 controller was purchased from Digiplan. It was intended to be the master controller for the system, by providing the demand pulses to the Dservo controller cards. The PC23 resided in an IBM PC and derived pulse trains for three drives dependent upon commands received from the host PC. It had a number of command strings that performed specific functions like "home axis" or "move to position". The following code is the resultant program written to control the two Electrocraft drives via the Dservo controllers.

The programs were written in Lattice "C" for speed of execution. The program was complex but the header before each routine explains its function. The programs were not properly documented with comments. Each small procedure did however explain its function. Many of the character sequences may appear obscur but they were derived from the PC23 manual, reference [19]. The program was split into two sections: the first contains the screen handling and man-machine interface routines, whilst the second contained the routines to send the appropriate characters to the PC23 to control the motors. The motors may be homed, incremented relative to one another and run up a user-defined ramp. Once running at continuous speed the drives may be incremented relative to one another or the drives may be stopped in synchronism (which could take up to 30s) or in an emergency mode (which took 5s). When running in synchronism the PC23 had a specific mode that ensured that the demand pulses it issued were synchronised, although errors were discovered between the signals.

A5.2 THE LISTING

```
/*      ****************************************************************
        *  THE SYNCHRONISATION CONTROLLER FOR THE LEDGER/CUT-OFF RIG   *
        ****************************************************************
        *  BY DAVID R. SEAWARD    OCTOBER 1988        VERSION 1.0      *
        ****************************************************************
        *       COPYRIGHT ASTON UNIVERSITY/MOLINS PLC 1988             *
        ****************************************************************
        ****************************************************************
        * This program provides the synchronised signals, via the PC23 *
        * to drive the Molins DServo controllers, which in turn        *
        * control the Electrocraft Bru-500 drives representing the     *
        * ledger and cut-off actuators on a rod-making machine         *
        ****************************************************************
        ****************************************************************
        * THE FUNCTIONS ARE AS FOLLOWS:-                               *
        * char GETNUM(low,hi) mug trap to get character between low&hi *
        * void CLEAR_SCREEN() invokes the IBM CLS command              *
        * void SCREEN1() The introductory screen                       *
        * void SCREEN2() The main/master menu                          *
        * void SCREEN3() The menu for incrementing axes                *
        * void SCREEN4() The menu for running axis 1 & 3 in synch      *
        * int  SCREEN5() A sub-menu for incrementing axes,return offset*
        * void SCREEN6() A sub-menu for user defined ramp              *
        * void MAIN() The main control for the program                 *
        ****************************************************************
        * Many other required functions are held in *util*.c where this*
        * program is main.c                                            *
        **************************************************************** */
        include <stdio.h>
        include <stdlib.h>
        include <string.h>
        include <dos.h>
        include <ctype.h>
        include <limits.h>
/*      ******************************************************************
        GETNUM returns a character number between predefined limits
        ****************************************************************** */
        char getnum(low,hi)
        int low,hi;
        {
        char i=0;
        i = getch();
        if(((i-'0')<low)||((i-'0')>hi)){
        printf("\n Please repeat input between %d and %d\n",low,hi);
        i = getnum(low,hi);}
        return i;
        }
/*      ******************************************************************
        CLEAR_SCREEN:invokes the DOS clear screen command
        ****************************************************************** */
        void clear_screen()
        {
        char *cls;
        cls="CLS";
        system(cls);
        return;
        }
/*      *******************************************************************
        KEY_IN:gets a character string from the keyboard terminated by CR
        ******************************************************************* */
        void key_in(s)
        char *s;
        {
        while((*s++=getche()) != 13);
        *s='\0';
        return;    }
```

```
/*      *******************************************************************
        UPPER_CASE:changes a character string to upper case
        ******************************************************************* */
        void upper_case(s)
        char *s;
        {
        int j=0;
        while(s[j] != 13) {
        s[j] = toupper(s[j]);
        j++;
        }
        return;
        }
/*      *******************************************************************
        INC_IN: interprets string to increment axes 1&3 to gain offset
        ******************************************************************* */
        int inc_in(s)
        char *s;
        {
        int error_flag=0,j=3;
        key_in(s);
        upper_case(s);
        if(s[0]=='X')
        return 99;
        if((s[0]!='A')||((s[1]!='1')&&(s[1]!='3'))||((s[2]!='A')&&(s[2]!='C')))
                error_flag++;
        do {
        if(((s[j]-'0') <0) || ((s[j]-'0')>9)){
                error_flag++;
                }
                }
        while(j<7 && s[++j]!=13);
        if(error_flag>0){
                printf("\n Please try again, with the proper format");
        return error_flag;
                }
        s[0]=s[1];
        s[1]='D';
        s[j]=' ';
        s[j+1]='\0';
        if(s[2]=='A')
                s[2]='-';
        else
                s[2]='+';
        return 0;
        }
/*      **************************************************************
        *       screen1   The introductory screen to the system        *
        ************************************************************** */
        void screen1()
        {
        int dummy;
        clear_screen();
        printf("*****************************************************\n");
        printf("*   Welcome to the Demonstration Software for the     *\n");
        printf("*   SERC/Molins plc Specially Promoted Programe into  *\n");
        printf("*   Continuously Synchronised High Speed Drives       *\n");
        printf("*****************************************************\n\n");
        printf("*****************************************************\n");
        printf("*Research Assistant on the Project :- David R. Seaward *\n");
        printf("*****************************************************\n\n");
        printf("*****************************************************\n");
        printf("* This software is written in C and controls the PC23  *\n");
        printf("* which in turn controls the Molins Dservos and the    *\n");
        printf("* brushless dc machines                                *\n");
        printf("*****************************************************\n\n");
        printf("TYPE ANY KEY TO CONTINUE");
        dummy=getch();
        return;    }
```

```
/*      ****************************************************************
        Screen 2 with all the choices on
        **************************************************************** */
        void screen2()
        {
        clear_screen();
        printf("*******************************************************\n");
        printf("* Please make a choice from the MASTER MENU           *\n");
        printf("*******************************************************\n");
        printf("\n\n\n\n\n\n");


        printf("    1) Home the axis\n");
        printf("    2) Increment the axes to locate desired offset\n");
        printf("    3) Run the axes in synchronism\n");
        printf("    4) Return to DOS operating system\n\n\n\n\n");
        printf("please type 1-4?");    return;    }
/*      ****************************************************************
        Screen 3 with all the choices on for incrementing the axes
        **************************************************************** */
        void screen3()
        {
        int flag = 1,offset=0,ret=0;
        FILE *fp;
        char dummy,num3 = '0',*mode1,*mode2,*name,message[30],*pmessage;
        char offmess[10];
        int screen5(void);
        name="pc23.dat";
        mode1="w";
        mode2="r";
        message[0]=' ';
        message[1]='1';
        message[2]='A';
        message[3]='1';
        message[4]=' ';
        message[5]='1';
        message[6]='V';
        message[7]='1';
        message[8]='0';
        message[9]=' ';
        message[10]='\0';
        pmessage=&message[0];
        while(flag)
        {
        clear_screen();
        printf("*******************************************************\n");
        printf("* Please make a choice from one of the following      *\n");
        printf("*******************************************************\n");
        printf("\n\n\n\n\n\n");    printf("    1) Increment axes\n");
        printf("    2) Store the new offset value in file PC23.DAT\n");
        printf("    3) Move the axes to the last pre-stored offset\n");
        printf("    4) Return to Master menu\n\n\n\n\n");
        printf("please type 1-4?");
        num3 = getnum(1,4);
        switch (num3) {
               case('1'):
               offset=screen5();
               break;
               case('2'):
               fp=fopen(name,mode1);
               fprintf(fp,"%d",offset);
               ret=fclose(fp);
               break;
               case('3'):
               fp=fopen(name,mode2);
               fscanf(fp,"%d",&offset);
               ret=fclose(fp);
               printf("\nThe pre-stored offset is %d pulses",offset);
               printf("\n\n Press any key to move axis 1 by this offset");
               dummy = getch();
               /* movement is here*/
```

```
        go_home(1,0,1);
        writecmd(pmessage);
        itoa(offset,&offmess[0]);
        message[0]='1';
        message[1]='D';
        message[2]='\0';
        strcat(pmessage,&offmess[0]);
        offmess[0]=' ';
        offmess[1]='1';
        offmess[2]='G';
        offmess[3]=' ';
        offmess[4]='\0';
        strcat(pmessage,&offmess[0]);
        writecmd(pmessage);
        message[0]=' ';
        message[1]='1';
        message[2]='A';
        message[3]='1';
        message[4]=' ';
        message[5]='1';
        message[6]='V';
        message[7]='1';
        message[8]='0';
        message[9]=' ';
        message[10]='\0';
        break;
        case('4'):
        flag = 0;
        break;
        default:
        printf("\nThe software must be screwed....Byeeeee!!!!!!\n");
        exit(0);
        break;
        }
        }
return;
}
/*      ******************************************************************
        Screen 4 with all the choices for running the axis up to speed
        ****************************************************************** */
void screen4()
{
int  flag = 1;
char num4 = '0';
void screen6();
while(flag)
{
clear_screen();
printf("*********************************************************\n");
printf("* Please make a choice from one of the following        *\n");
printf("*********************************************************\n");
printf("\n\n\n\n\n\n");


printf("    1) Run up the default ramp (2500r/min in 10seconds)\n");
printf("    2) Run on user defined ramp\n");
printf("    3) Return to Master Menu\n\n");
printf("   'S' for Stop the axes (it may take 20s to occur)\n");
printf("   'E' for Emergency stop\n");
printf("   '>' for phase advance (axis 1 clockwise one pulse)\n");
printf("   '<' for phase advance (axis 1 anticlockwise one pulse)\n\n");
printf("please type 1-3?");
num4 = getnum(1,3);
switch (num4) {
        case('1'):
        control_ramp(494351,101);
        break;
        case('2'):
        screen6();
        break;
```

```
        case('3'):
        flag = 0;
        break;
        default:
        printf("\nThe software must be screwed....Byeeeee!!!!!!\n");
        exit(0);
        break;
        }
        }
return;
}
/*      ****************************************************************
Screen 5 with all the choices for incrementing the axis
**************************************************************** */
int screen5()
{
int  flag = 1, offset = 0,num=0, error=1;
/*Axis 1 forward is positive offset, Axis 2 forward is negative*/
char dummy,*message,*move;
message=" 3V10 3A1 1V10 1A1 ";
move="this doesn't have to be a long string";
while(flag)
{
clear_screen();
printf("***********************************************************\n");
printf("* Please give the offset command                         *\n");
printf("***********************************************************\n");
printf("\n\n\n\n\n\n");

printf("    The form of the offset command is anannnn\n");
printf("    Where a is a letter, n a number\n");
printf("    The first letter must be a 'A' for Axis\n");
printf("    The system is configured for axis 1 and axis 3 of PC23\n");
printf("    The first number designates axis '1' or axis '3'\n");

printf("    The second letter may be 'A' for anticlockwise movement\n");
printf("    or 'C' for clockwise movement\n");
printf("    Remaining numbers are the offset from 0-9999 pulses\n");
printf("\n    Type a 'X' to eXit to the Menu\n\n\n\n\n");
error=1;
while(!(error==0 || error == 99)){
        error=inc_in(move);
        num=atoi(move+3);
        if(error==99)
        flag=0;}
        if(error==0){
        /* calculation of offset */
        if(move[0]=='1'){
                if(move[2]=='+')
                        offset+=num;
                else
                        offset-=num;}
        else { /* it must be a three */
                if(move[2]=='+')
                        offset-=num;
                else
                        offset+=num;}
        writecmd(move);
        writecmd(message);
        move[1]='G';
        move[2]=' ';
        move[3]='\0';
        writecmd(move);
        printf("\nThe total offset applied is %d encoder
        pulses\n",offset);
        printf("\n\n Press any key to continue");
        dummy = getch(); }
}
return offset;}
```

```
/*      *******************************************************************
        Screen 6 the menu for user defined ramps
        ******************************************************************* */
        void screen6()
        {
        int  flag = 1,error=0;
        char dir,dummy,dummy1[20];
        long int speed=0l;
        long int dt;
        while(flag)
        {
        clear_screen();
        printf("*****************************************************\n");
        printf("* Please input the required ramp                    *\n");
        printf("*****************************************************\n");
        printf("\n\n\n\n\n\n");


        printf("The ramp requires a full speed, a time to reach full speed\n");

        printf("    and the direction of motion\n\n\n");     error=0;
        while(error==0){
        printf("\n    Please input the full speed in rpm? ");
        error=scanf("%ld",&speed);
        if(error==0)
        scanf("%s",&dummy1[0]);
        if(error!=0 && speed>3000l){
               printf("\n This speed is too large (greater than 3000rpm)\n");
        error=0;
               }
        else if(error!=0)
               speed=(speed*6667l)/1000l;
               /* for 50ms update time convert to pulse*/
        }
        error=0;
        while(error==0){
        printf("\nPlease input the direction (A - anticlockwise,C- clockwise)");
        dir=getche();
        dir=toupper(dir);
        switch (dir) {
        case('A'):
        error=1;
        break;
        case('C'):
        speed+=32768l;
        error=1;
        break;
        default:
        error=0;
        break;
        }
        }
        error=0;
        while(error==0){
        printf("\n    Input the time to reach full speed in integer seconds? ");
        error=scanf("%ld",&dt);
        if(error==0)
        scanf("%s",&dummy1[0]);    }
        printf("\n     Press any Key to Run\n");
        printf("   'S' Stops the axes (it may take 20s before deceleration)\n");
        printf("   'E' for Emergency stop\n");
        printf("   '>' for phase advance (axis 1 clockwise one pulse)\n");
        printf("   '<' for phase advance (axis 1 anticlockwise 1 pulse)\n\n\n");
        dummy=getch();
        control_ramp(speed,dt);
        flag=0;
        }
        return;
        }
```

```
/*      ****************************************************************
        THE MAIN PART OF THE PROGRAMME
        **************************************************************** */
        int main()
        {
        char num2='0';
        initialise(); /* RESET THE PC23*/
        screen1();
        for(;;){
        screen2();
        num2 = getnum(1,4);
        switch (num2) {
               case('1'):
               go_home(1,0,1);
               break;
               case('2'):
               screen3();
               break;
               case('3'):
               screen4();
               break;
               case('4'):
               clear_screen();
               exit(0);
               break;
               default:
               printf("\nThe software must be screwed....Byeeeee!!!!!!\n");
               exit(0);
               break;
               }
               }
        return 0;           }
/*      ****************************************************************
        *  THE SYNCHRONISATION UTILITIES FOR THE LEDGER/CUT-OFF RIG     *
        *  BY DAVID R. SEAWARD    OCTOBER 1988         VERSION1.0       *
        *       COPYRIGHT ASTON UNIVERSITY/MOLINS PLC 1988              *
        ****************************************************************
        * This program provides the synchronised signals, via the PC23  *
        * to drive the Molins DServo controllers, which in turn         *
        * control the Electrocraft Bru-500 drives representing the      *
        * ledger and cut-off actuators on a rod-making machine          *
        ****************************************************************
        * THE FUNCTIONS ARE AS FOLLOWS:-                                *
        * char READCH() reads a character from the PC23                 *
        * void WRITECMD(s) writes the string pointed to by s to PC23    *
        * void NO_BUFF_WRITECMD(s) as writecmd() without buffercheck    *
        * void READANSWER(s) reads a PC23 string into pointer s         *
        * void INITIALISE() initialises and resets the PC23             *
        * void WRITECH(alpha) writes character alpha to the PC23        *
        * void GO_HOME(axis1,axis2,axis3) homes to the encoder index    *
        *                                 pulse the "1" flagged axes    *
        * void ZERO_FILL(zfs,zfn) fills string pointer zfs with zeros   *
        *                         up to zfn characters                  *
        * double POWER(pb,pn) raises from base pb to pn power           *
        * void REVERSE(rs) reverse pointer string rs                    *
        * void ITOB(ibn,ibs,ibb) converts ibn in base 10 to the string  *
        *                       ibs in base ibb                         *
        * void ITOA(n,s) converts interger n to string s                *
        * void SYNCH_START() sets the PC23 to time/distance streaming   *
        * void SYNCH_END() ends the time/distance streaming mode        *
        * int  IS_BUFF_FULL() checks if the PC23 axis 1 command buffer  *
        *                     has more than 32 bytes free               *
        * int ARE_MOT_MOV() returns the number of axes still moving     *
        * int WANT_STOP() returns 1 if E has been hit, 2 for Selse 0    *
        * long int SYNCH_RAMP(rhi,rlo,rdt) runs axes 1&3 up/down a      *
        *     velocity ramp from rhi to rlo pulses/sec in rdt secs in   *
        *     direction determined by '+32768' is clockwise             *
        * long int SYNCH_CONT(cvel,cdt) constant velocity profile at    *
        *     cvel pulse/sec for cdt secs as determined in synch_ramp() *
        * long int STOP_RAMP(shi,sdt) stops the drives from shi pulses  *
        *                         per sec in sdt seconds                *
        **************************************************************** */
```

```
        include <stdio.h>
        include <string.h>
        include <limits.h>
        include <ctype.h>
        include <stdlib.h>
        include <dos.h>
        int stop_cond=0,fly_offset=0;
/*      ******************************************************
        VARIOUS STATUS AND CONTROL BYTE MASKS USED IN HAND-SHAKING
        WITH THE PC23. PAGES 22-30 OF THE PC23-03 MANUAL DESCRIBE
        THE FUNCTIONS OF EACH REGISTERS BITS (WITH THE EXCEPTION
        OF THE STATUS REGISTERS BIT 7 WHICH INDICATES IF THE PC23
        BUFFER IS OVER HALF FULL). PAGE 30 GIVES A STEP BY STEP
        PROCEDURE OF THE HANDSHAKING PROCESS. PAGE 62 EXPLAINS
        THE DIFFERENCES BETWEEN THE BINARY WRITE (TIME DISTANCE)
        FUNCTION AND THE NORMAL WRITE FUNCTION.
        ***************************************************** */
        define FAIL         0X20
        define MAXINT       32000
        define START_M      0X7F
        define INTCLR       0X20
        define RESTART      0X17
        define POSITIVE     0X8000
        define MOT_MOV      0X7
        define STOP         0X64
        define START        0X40
        define CB           0X60
        define CB1          0X61
        define IDB_M        0X10
        define IDB          0X70
        define IDB1         0X71
        define ODB          0X8
        define ACK          0XE0
        define BFULL        0X80  /* checks the buffer status -reserved */
        define RETURN       0XD
        define WTCHDG       1000  /* Communication Watchdog */
        int address = 0X300;      /* normally 300hex */
/*      ***************************************************************
        READCH: READS ONE CHARACTER OF A PC23 RESPONSE TO A STATUS REQUEST.
        RETURNS THE CHARACTER RESPONSE.
        ************************************************************** */
        char readch()
        {
        char alpha = 0;
        while (!(inp(address+1) & ODB));
               alpha = inp(address);
               outp (address+1,ACK);
        while ((inp(address+1) & ODB));
               outp (address+1,CB);
        return(alpha); }
/*      **************************************************************
        WRITECMD: WRITES A COMMAND STRING TO THE PC23.
        ************************************************************** */
        writecmd(s)
        char *s;
        {
        while(is_buff_full());
        while (*s)
        writech (*s++);
        return; }
/*      **************************************************************
        NO_BUFF_WRITECMD: writes a command string without checking the
        buffer status
        ************************************************************** */
        no_buff_writecmd(s)
        char *s; {
        while (*s)
        writech (*s++);
        return; }
```

```
/*      ************************************************************
        READANSWER: READS A COMPLETE PC23 STATUS REQUEST RESPONSE STRING.
        ************************************************************ */
        readanswer (s)
        char *s;
        {
        while ((*s++ = readch()) != 13);
        *s = '\0';
        return; }
/*      ************************************************************
        INITIALISE: RESET THE PC23. THE PC23 MUST HAVE ITS ADDRESS SET TO
        THE FIRST VALUE IN THE USER DATA FILE OR THE PROGRAM WILL EXIT
        ************************************************************ */
        initialise () {
        int i = 0;
        outp (address+1, STOP);
        while (!((inp(address+1)) & FAIL));
                outp (address+1,START);
        outp (address+1,CB);
        while(((inp(address+1) & START_M) != RESTART) && (i++ !=MAXINT));
        if (i >= MAXINT)
                { printf ("invalid address\n");
        exit(0); }
        outp(address+1,INTCLR);
        outp(address+1,CB);
        return; }
/*      ************************************************************
        WRITECH: WRITES A SINGLE CHARACTER TO THE PC23
        ************************************************************ */
        writech ( alpha )
        char alpha; {
        while (!(inp(address+1) & IDB_M));
                outp (address,alpha);
        outp(address+1,IDB);
        while (inp(address+1) & IDB_M);
                outp(address+1,CB);
        while (!(inp(address+1) & IDB_M));
        return; }
/*      ************************************************************
        GO_HOME: Homes axis 1,2 or 3 to the encoder index pulse
        arguments set to one home the appropriate axis
        ************************************************************ */
        void go_home(axis1,axis2,axis3)
        int axis1,axis2,axis3; {
        char *message;
        if(axis1==1){
                message="1K 1FSB1 1GH+0.1 1FSB0 ";
                writecmd(message);}
        if(axis2==1){
                message="2K 2FSB1 2GH+0.1 2FSB0 ";
                writecmd(message);}
        if(axis3==1){
                message="3K 3FSB1 3GH+0.1 3FSB0 ";
                writecmd(message);}
        /* could insert RC check for home position here */
        return; }
/*      ************************************************************
        ZEROFILL: Fills a character string with leading zeros up to a
        total character length of zfn
        ************************************************************ */
        void zerofill(zfs,zfn)
        int zfn;
        char *zfs; {
        int i,j;
        if((i=zfn-strlen(zfs))>0){
        zfs[zfn+1]='\0';
        for(j=zfn;j>=i;j--)
        zfs[j] = zfs[j-i];
        for(j=i-1;j>=0;j--)
        zfs[j] = '0';     }
        return; }
```

```
/*      ******************************************************************
        POWER raises from base pb to pn power
        ****************************************************************** */
        double power(pb,pn)
        long int pb,pn;
        {
        int i=0;
        double j=1;
        for(i=1;i<=pn;++i)
        j*=pb;
        return j;
        }
/*      ******************************************************************
        REVERSE reverses the character string rs[]
        ****************************************************************** */
        void reverse(rs)
        char rs[];
        {
        int c=0,i=0,j=0;
        for(i-0,j = strlen(rs)-1;i<j;i++,j--){
                c     = rs[i];
                rs[i] = rs[j];
                rs[j] = c;
                }
        return;
        }
/*      ******************************************************************
        ITOB converts base 10 number ibn to base ibb, in string ibs
        ****************************************************************** */
        void itob(ibn,ibs,ibb)
        char ibs[];
        long int ibn,ibb; {
        long int int1=0,int2=0,int3=0,i=0,sign=0;
        if(ibb==0)
        exit(0);
        if((sign=ibn)<0)
                ibn=-ibn;
        while(ibn>0){
        int2 = power(ibb,i+1);
        int3 = int2/ibb;
        int1 = (ibn%int2)/int3;
        ibn -= int1*int2/ibb;
        if(int1>=10)
        ibs[i] = int1 -10 +'A';
        else
        ibs[i] = int1 +'0';
        i++;}
        if (sign<0)
                ibs[i++] = '-';
        ibs[i++] ='\0';
        reverse(ibs);
        return; }
/*      ******************************************************************
        ITOA converts base number to a string
        ****************************************************************** */
        void itoa(n,s)
        int n;
        char *s; {
        int i,sign;
        if((sign=n)<0)          /*record sign */
                n= -n;          /* make n positive */
        i=0;
        do {
           s[i++] = n % 10 + '0';
        } while ((n /= 10) >0);
        if (sign <0)
                s[i++] = '-';
        s[i] = '\0';
        reverse(s);
        return; }
```

```
/*      ****************************************************************
        DELAY: introduces a delay of wait hundredths of a second
        **************************************************************** */
        void delay(wait)
        int wait;
        {
        unsigned char c[8],*pc,last;
        pc=&c[0];
        getclk(pc);
        last=c[7];
        do{
        getclk(pc);
        if(c[7]<last)
        c[7]+=100;
        }while((c[7]-last)<wait);
        return;
        }
/*      ****************************************************************
        SYNCH_START: sets PC23 up for synching output pulses
        **************************************************************** */
        void  synch_start()
        {
        char *ssmess;
        ssmess = "1Q2 3Q2 MSL1X1 1TD50 3TD50 MSS ";
        writecmd(ssmess);
        return;
        }
/*      ****************************************************************
        SYNCH_END: stops PC23 being in synch mode
        **************************************************************** */
        void  synch_end()
        {
        char *semess;
        semess = "1Q0 3Q0 ";
        while(are_mot_mov());
        writecmd(semess);
        return;
        }
/*      ****************************************************************
        SYNCH_KILL: stops PC23 being in synch mode
        **************************************************************** */
        void  synch_kill()
        {
        char *semess;
        semess = "1Q0 3Q0 ";
        writecmd(semess);
        return;
        }
/*      ****************************************************************
        IS_BUFF_FULL: checks if PC23 buffer is full (1)
        **************************************************************** */
        int is_buff_full()
        {
        char *bc;
        bc="rubbish";
        bc[0]='B';
        bc[1]=' ';
        bc[2]='\0';
        no_buff_writecmd(bc);
        readanswer(bc);
        if(bc[1]=='B'){
                return 1; }
        else
                return 0; }
```

```
/*      ******************************************************************
        ARE_MOT_MOV: are any of the motors moving (flag=numbermot moving)
        ****************************************************************** */
        int are_mot_mov()
        {
        int mot_flag=0;
        if((inp(address+1) & MOT_MOV) ^ MOT_MOV){
               mot_flag=1;    }
        else{
               return mot_flag;
            }             .    }

/*      ******************************************************************
        WANT_STOP: is the keyboard request for a stop
        ****************************************************************** */
        int want_stop()
        {
        char alpha=0;
        int flag=0,kbflag=0;
        kbflag=kbhit();
        if(!kbflag)
        return 0;
        alpha=getch();
        alpha=toupper(alpha);
        switch(alpha){
               case('E'):
               flag= 1;
               break;
               case('S'):
               flag = 2;
               break;
               case(60):  /* asci for < */
               flag=3;
               break;
               case(62):  /* asci for > */
               flag=4;
               break;
               default:
               printf("\nPlease do not hit keys whilst motor is running");
               flag=0;
               break;
               }
        return flag;
        }
/*      *****************************************************************
        SYNCH_RAMP runs up a ramp
        ****************************************************************** */
        long int synch_ramp(rhi,rlo,rdt)
        long int rhi,rlo,rdt;    /* in pulses per sample, pulsesper samp,secs */
        {
        long int rdelta=50l;   /*default for delta is 50ms*/
        long int i,j;
        long int rpulse,rinc;
        int stop_flag=0;
        extern int stop_cond;
        char *rmess, *rs;
        rmess="garbage    ";
        rs="nothing";
        j      = (1000l * rdt)/rdelta;
        rinc   = (rhi - rlo)/j;
        rpulse = rlo - rinc;
        for(i=0l;i<=j;i++){
               rpulse += rinc;
               itob(rpulse,rs,16l);
               zerofill(rs,4);
               rmess[0] = 'S';
               rmess[1] = 'D';
               rmess[2] ='\0';    /*have to reset end for conc*/
               strcat(rmess,rs);
               strcat(rmess,rs);
```

```
        rmess[10] = ' ';
        rmess[11] = '\0';
        if(stop_flag=want_stop()){
                stop_cond=stop_flag;
                if(stop_flag==1){ /* Emergency stop*/
                writecmd(rmess);
                return rpulse;
                }
                else if(stop_flag==2){ /*Normal Stop*/
                writecmd(rmess);
                return rpulse;
                }
                }
        writecmd(rmess);
        }
    return 0l; }
/*  ******************************************************************
    SYNCH_CONT: runs on a continuous section
    ****************************************************************** */
    long int synch_cont(cvel,cdt)
    long int cvel,cdt;
    {
    char *cmess,*cs,cmess9;
    long int cdelta=50l;    /*default for delta is 50ms*/
    long int i,j;
    long int cpulse;
    int stop_flag=0;
    extern int fly_offset,stop_cond;
    cs ="blank";
    cmess="rubish      ";
    cpulse = cvel;
    j      = (1000l * cdt)/cdelta;
    itob(cpulse,cs,16l);
    zerofill(cs,4);
    cmess[0] = 'S';
    cmess[1] = 'D';
    cmess[2] ='\0';   /*have to reset end for conc*/
    strcat(cmess,cs);
    strcat(cmess,cs);
    cmess[10] = ' ';
    cmess[11] = '\0';
    cmess9=cmess[9];
    for(i=0l;i<=j;i++){
        if(stop_flag=want_stop()){
                stop_cond=stop_flag;
                if(stop_flag==1){ /* Emergency stop*/
                writecmd(cmess);
                return cpulse;
                }
                else if(stop_flag==2){ /*Normal Stop*/
                writecmd(cmess);
                return cpulse;
                }
                else if(stop_flag==3){ /*Anticlockwise*/
                        ++cmess[9];
                        --fly_offset;
                        }
                else if(stop_flag==4){ /*Clockwise*/
                        --cmess[9];
                        ++fly_offset;
                        }
                }
    writecmd(cmess);
    if(stop_flag)
    cmess[9]=cmess9;
    }
    return 0l;
    }
```

```
/*      ******************************************************************
        STOP_RAMP runs down a ramp under stop condition (no keyboard check)
        ****************************************************************** */
        long int stop_ramp(slo,sdt)
        long int slo,sdt;    /* in pulses per sample, secs */
        {
        long int sdelta=50l,shi;    /*default for delta is 50ms*/
        long int i,j;
        long int spulse,sinc;
        char *smess, *ss;
        smess="garbage     ";
        ss="nothing";
        j       = (1000l * sdt)/sdelta;
        if(slo==0l | slo==32768l)
        return 0;
        if(slo>=32768l)
                shi=32768l;
        else
                shi    = 0l;
        sinc   = (shi - slo)/j;
        spulse = slo - sinc;
        for(i=0l;i<=j;i++){
                spulse += sinc;
                itob(spulse,ss,16l);
                zerofill(ss,4);
                smess[0] = 'S';
                smess[1] = 'D';
                smess[2] ='\0';   /*have to reset end for conc*/
                strcat(smess,ss);
                strcat(smess,ss);
                smess[10] = ' ';
                smess[11] = '\0';
                writecmd(smess);
                }
        return (0l);
        }
/*      ******************************************************************
        CONTROL_RAMP runs up a ramp with all stop control functions
        ****************************************************************** */
        void control_ramp(hi,dt)
        long int hi,dt;
        {
        int stop_flag=0;
        long intstop,stop1,time,accel=1660l,max_accel=3500l,lo=32768l;
        long int stop_ramp(),synch_ramp(),synch_cont();
        extern int fly_offset,stop_cond;     char dummy;
        if(hi<=32768l)
        lo=0l;
        accel=(hi-lo)/dt;
        if(accel>max_accel){
                printf("\n acceleration out of bounds\n");
                return;
                }
        synch_start();
        fly_offset=0;
        stop=synch_ramp(hi,lo,dt);
        if(stop){
                stop_flag++;
                if(stop>32768)
                stop1=stop-32768;
                else
                stop1=stop;
                if(stop_cond==1)
                accel=max_accel;
                time=stop1/accel;
                if(time==0)   time=1;
                stop=stop_ramp(stop,time);
                synch_end();
                printf("\nThe total offset applied was%d \n",fly_offset);
```

```
        printf("Press any key to continue");
        dummy=getch();
        return;
        }
while(!stop_flag){
stop=synch_cont(hi,51);    /* arbituary time to resetcontinuous section*/

if(stop){
        stop_flag++;
        if(stop>32768)
        stop1=stop-32768;
        else
        stop1=stop;
        if(stop_cond==1){
        accel=max_accel;
        synch_kill();
        synch_start();
        }
        time=stop1/accel;
        if(time==0)
        time=1;
        stop=stop_ramp(stop,time);
        }
        }
synch_end();
printf("\nThe total offset applied was%d \n",fly_offset);
printf("\nPress any key to continue");
dummy=getch();
return;
}
```

APPENDIX 6

A6.0 THE ERROR MONITOR DESIGN

A6.1 INTRODUCTION

It was necessary to design circuitry to monitor the relative difference in position between two drive motors. The encoders mounted upon the motor shafts were to be used to define position. Circuitry was also designed to monitor the error between set point and demand for each drive and also between the two set-points fed to the synchronised drives.

The design encorporated both TTL and CMOS devices and the encoder direction decoding was based upon the DSERVO microprocessor controller designed by Molins. The circuit layouts were designed using the SMARTWORKS package upon an IBM PC.

A6.2 DESIGN PHILOSOPHY

The positional feedback from the motor encoder was in the form of a quadrature pulse train, that is, two trains of square waves displaced by 90° to one another. Direction could be decoded from the relationship between the two signals and a times four could be performed on the number of pulses, by investigating each transition of the two signals. The demand signal was also in the form of a pulse train and a high-low signal which represented direction.

In order to use one circuit to decode errors between signals it was decided to use a signal conditioning circuit to change all the signals to the same form. One simple way to monitor error between two pulse trains is to utilise a counter circuit and have one pulse train counting up and the other counting down. An added complication with the motor case was that direction must also be decoded, so that each pulse train may count up or down. The conditioning used on the original signal was designed to produce two pulse trains from each source, one carrying up (or forward) pulses and the other down (or reverse) pulses. Thus if the demand or motor feedback was unidirectional only one pulse train would be active.

Thus the counter circuit had to decode four signals: input 1 up, input 1 down, input 2 up and input 2 down. Input 1 up and input 2 down counted the counter in a positive or up direction whilst input 1 down or input 2 up counted the counter negative or down.

Two circuit boards were designed, one conditioning board for two motors and two demand signals, called the up/down decoder, and one circuit board to derive an error between any two of the conditioned signals, called the error monitor. The schematic of both these circuits are shown in figures (A6.1) and (A6.2).

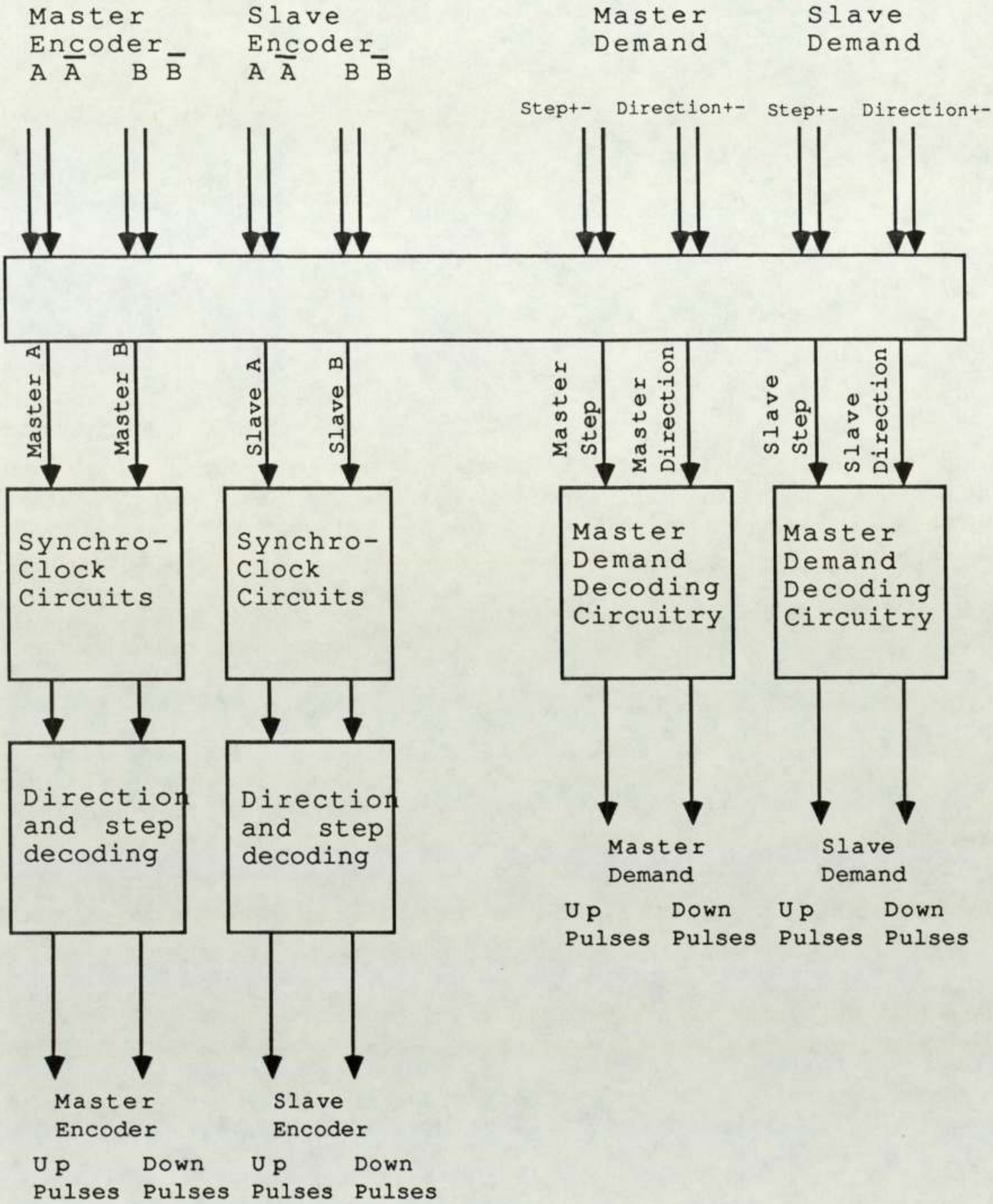


Figure A6.1 The up/down decoder schematic

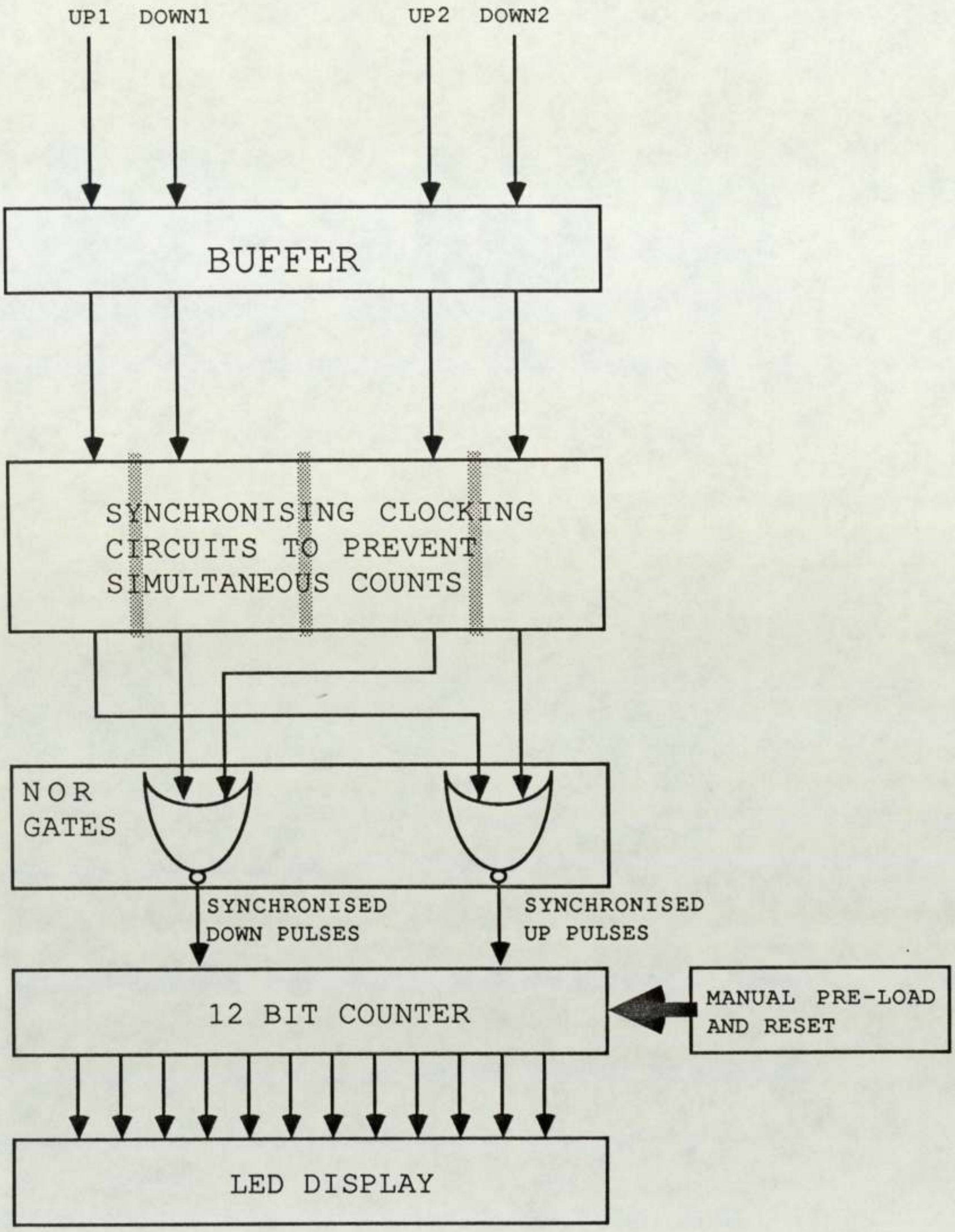


Figure A6.2 The error monitor schematic

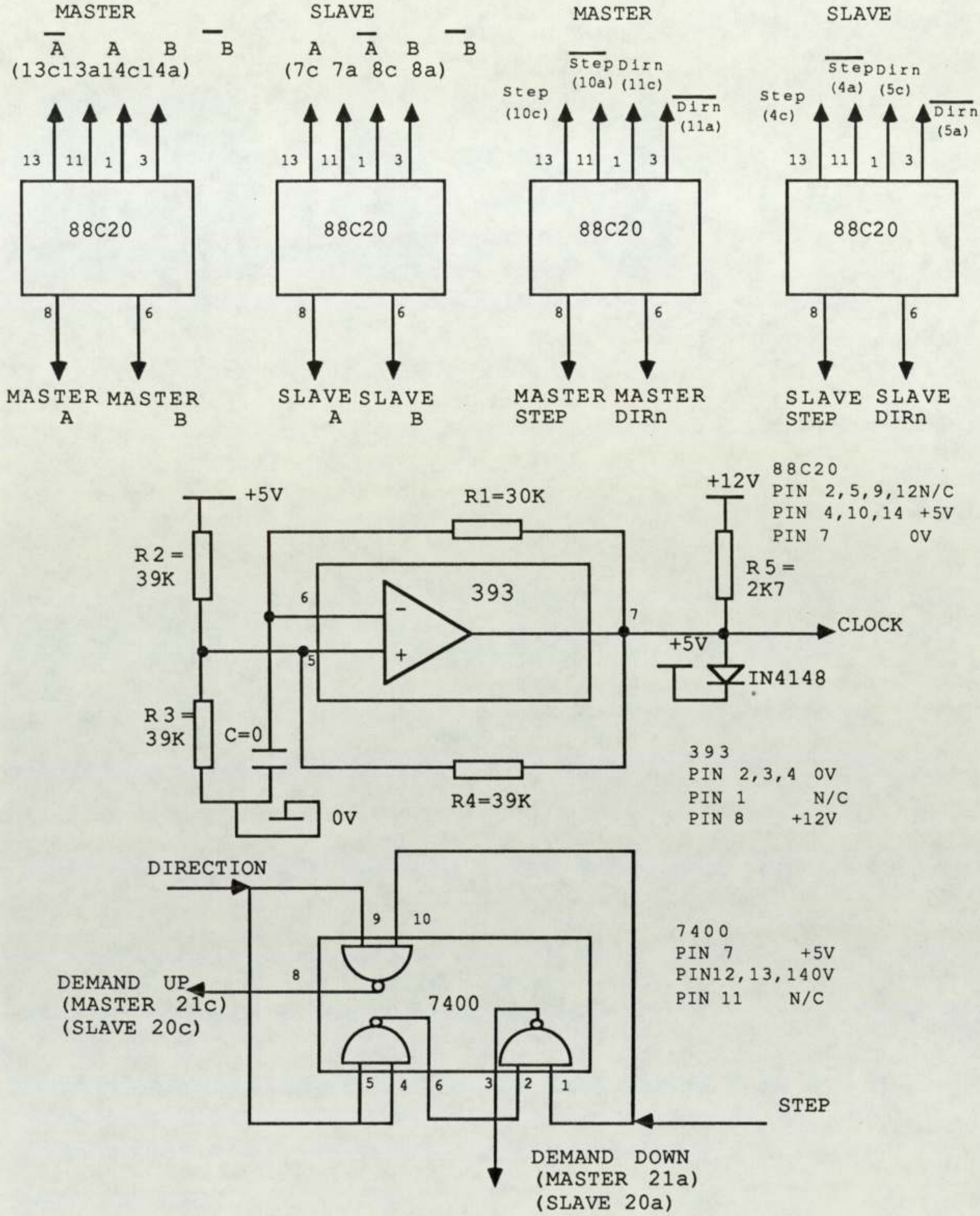


Figure A6.3a Up/down Decoder Circuitry

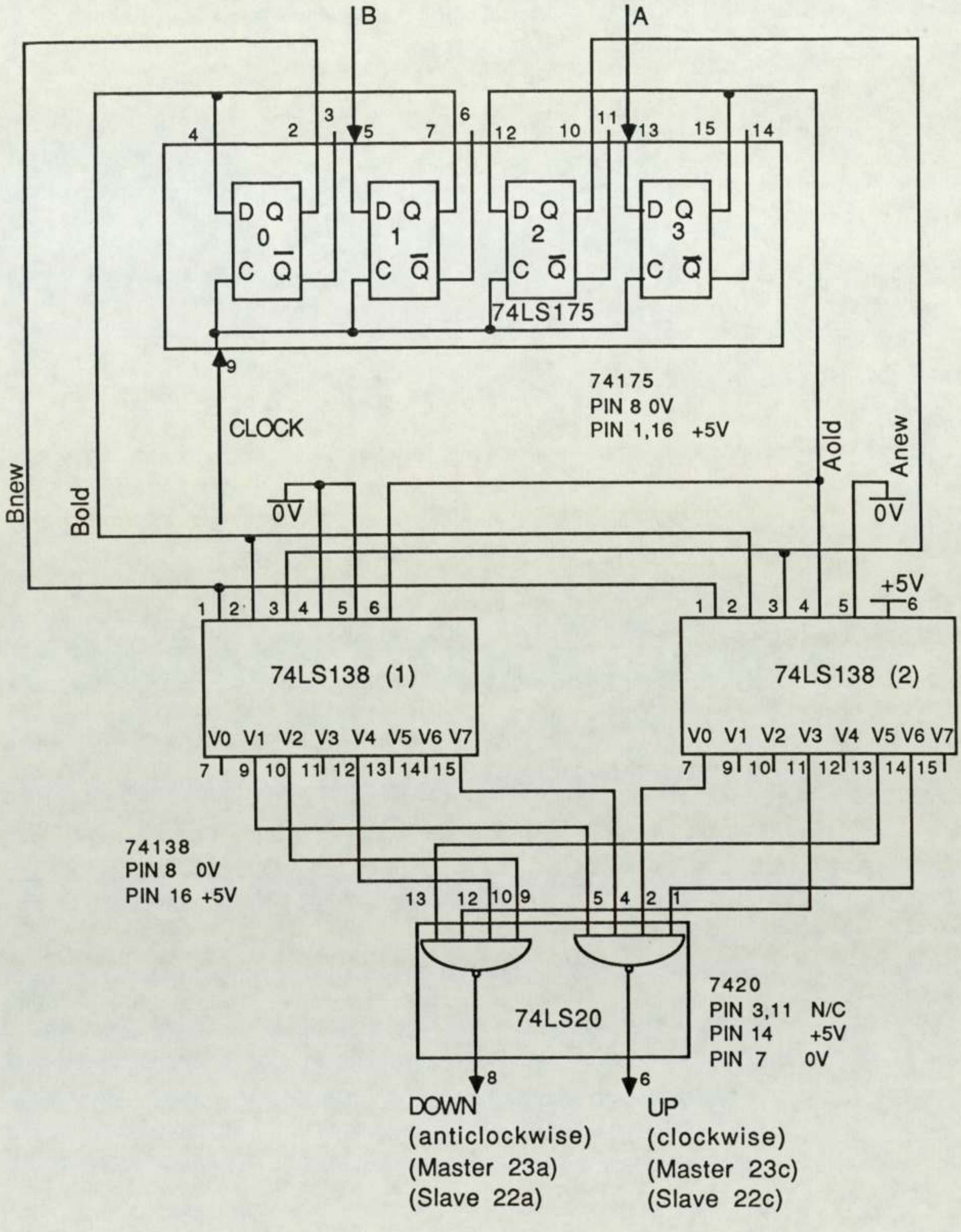


Figure A6.3b Up/down Decoder Circuitry

A6.3.0 THE DESIGN

A6.3.1 DECODING DEMAND SIGNALS

The demand signal for the servo system was derived from the PC23 indexer, which resided in an IBM PC. In order to use the same counting circuitry to monitor demand signals as that used by encoder feedback signals it was necessary to convert the pulse train and direction signals into two pulse trains, one representing forward motion the other reverse. The signals were first buffered using a 88C20 line receiver.

Signal conditioning could be achieved by use of appropriate NAND gates (74LS00). The direction and an inverted direction signal were both NANDed with the step signal, such that one NAND gate had the inverted pulse stream as its output when the direction signal was high whilst the other had no output. The pulse stream transfered across to the other output when the direction signal went low, see figure (A6.3)).

A6.3.2 DECODING MOTOR SIGNALS

The output from the motor encoder was classed as quadrature, that is, there were two pulse train outputs displaced by 90°. The phase shift between the signals allows the output pulse frequency that can be derived from the encoder to be four times that of the individual pulse train. Each change of state of the encoder pulse trains is converted into one pulse of the final pulse train. One can also derive direction from these signals by comparing the state (high or low) of one signal when the other has just gone through a change of state.

The circuitry used in the Molins Dservo controller was to be used to derive two pulse trains from the motor encoder, one being active for clockwise motion and the other for anti-clockwise motion.

The circuitry essentially had three stages:- the signals were first buffered, and applied to a synchronising clock arrangement such that the present state and recently past state of the pulse trains could be found, and the two present and two past states of the encoder were then decoded to produce the final pulse trains.

The buffering was implemented by use of 88C20 line receivers specially suited to receive differential encoder signals. This buffer eliminated noise from the encoder signals and acted as a safety device to prevent the encoder's line drivers from being shorted out and damaged.

The synchronising clock was constructed from clocked flip-flops, such that each clock-cycle allows the input state of the flip-flop to be transferred to its output until the next clock. Using two flip-flops in series such that the output of one fed the input of another, allowed the present input waveform state (displaced by one clock cycle), and the old input waveform state (displaced by two clock cycles) to be known and held for one clock cycle. Thus transfer of states was accomplished in a timely manner, see figure (A6.4).

At any given time four signals could be monitored from the synchronised clock circuit, the new and the old state of the input pulse waveforms. A pulse had been found when a new value was different from its corresponding old one.

The pulse could have a direction associated with it by looking at the state of the non-changing waveform. For example if the A channel had a positive to zero change then the new value will be zero and the old will be one, and the direction could be determined from the B channel which would be positive for clockwise and negative for anti-clockwise motion.

The decoding of waveforms is traditionally accomplished by using EEPROM's with the address lines connected to the flip-flop's outputs, but the Molins design used two 74LS138 decoding chips. The truth table of the two devices as connected showed that for any given input state from the four inputs only one of the output lines would go low (all others remaining high), see figure (A6.4). By NORing all the clockwise states and all the anti-clockwise states separately (four of each) one derives one pulse train for clockwise motion and one for anti-clockwise motion which could then be used by the counter circuitry.

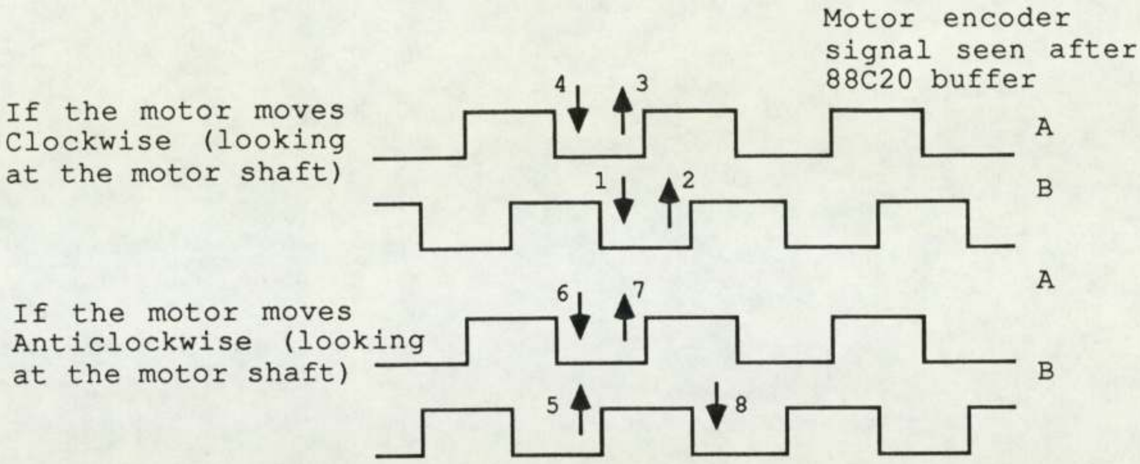
The clock frequency must be sufficiently high that encoder pulses are not lost. From figure (A6.4), the clock must run at least twice the speed of the main pulse frequency, that is, an 8000 pulse encoder at 3000 r/min or 400kHz. If an edge had occurred but was just missed by a clock and the transition on the other signal occurs before the next clock cycle then a pulse would be lost; for a perfect (90° degree displacement between every transition) encoder the clock frequency must be 800kHz. Unfortunately the encoders are never perfect so a safer clock frequency will be in the range of 1MHz.

A6.3.3 COUNTER CIRCUITRY

The error monitor counter circuitry used 4000 series CMOS technology. It was designed to receive two sets of two pulse trains, two up and two down, and to count the relative error between them. The error was output as a series a LEDs which illuminated for a 'high' count, and which could be interfaced to a logic analyser. The circuit had manual switches to preset the initial count and to reset the counter to this value. If large errors existed the counters would "topple", so three four bit counters were used to give a 12 bit binary range of +-2048 counts. The LEDs therefore had a binary loading.

The circuit had a synchronising clock from a 4584 chip, see figure (A6.5), which was divided into 8 phases by the 4022. These phases were used to control flip-flops in a timely manner. The inputs from the up/down decoder were pulled up to CMOS levels by 2K2Ω resistors before being clocked by the flip-flop circuitry. The appropriate up and down counts were then NORed by the 4001 chip such that two timely (ie non-concurrent) pulse trains could be applied to the counters up and down count inputs. The counters were 40193 four bit counters with preset, and three were used in series to produce a twelve bit count.

The output of the counters was fed to LEDs to give a visible indication of the count.



On the 74LS175,
pins 15,12 have A delayed by one clock (Anew)
pins 7,4 have B delayed by one clock (Bnew)
pin 10 has A delayed by two clocks (Aold)
pin 2 has B delayed by two clocks (Bold)

At each transition point (1-8) one output, only, of the 74LS138's goes "low", so that by NORing the corresponding low lines a series of pulses can be derived for clockwise motion (transitions 1-4) and one for anticlockwise motion (transitions 5-8).

TRANSITION	Anew	Aold	Bnew	Bold	"Low" Output (138 Number)	
1	L	L	L	H	1(2)	clockwise "NOR"
2	L	H	H	H	7(2)	
3	H	L	L	L	0(1)	
4	H	H	H	L	6(1)	
5	L	L	H	L	2(2)	anticlockwise "NOR"
6	L	H	L	L	4(2)	
7	H	L	H	H	3(1)	
8	H	H	L	H	5(1)	
	H	H	H	H	7(1)	No change
	L	L	L	L	0(2)	
	L	L	H	H	3(2)	
	H	H	L	L	4(1)	

Figure A6.4 Direction decoding logic diagram

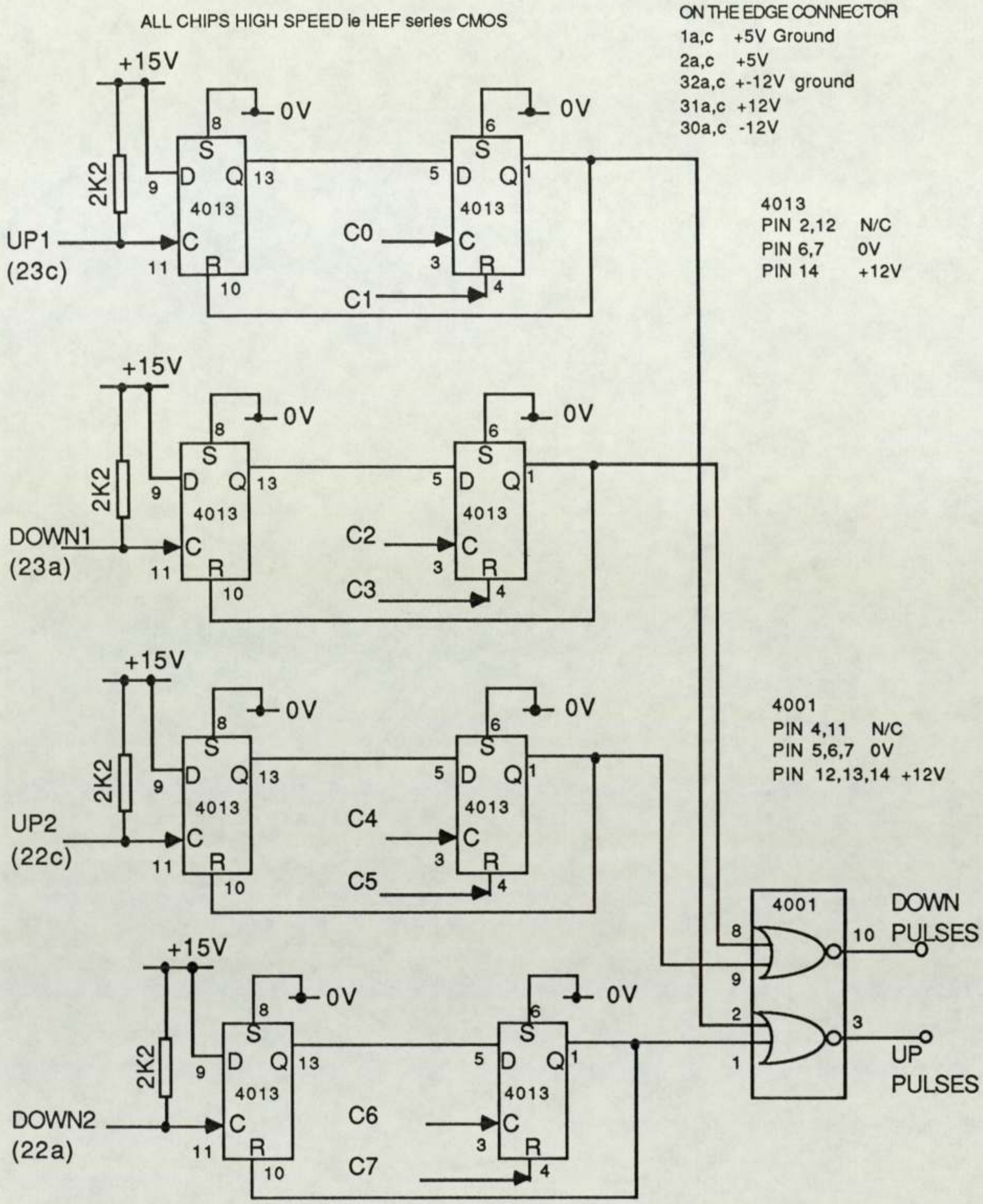


Figure A6.5a Error Monitor Circuitry

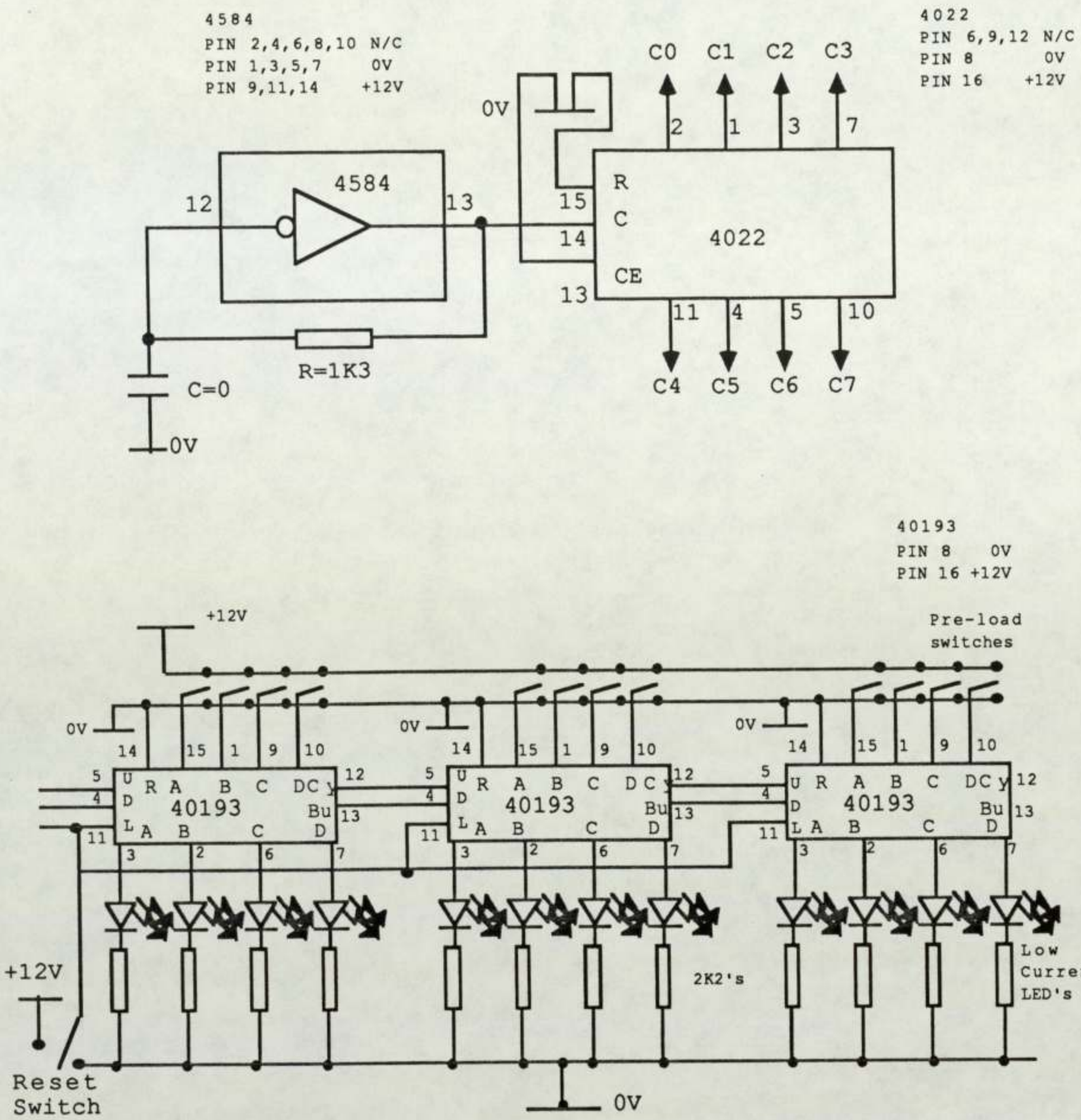


Figure A6.5b Error Monitor Circuitry

A6.4 <u>CONCLUSIONS</u>

The system initially was found to "lose" pulses and this fault was tracked down to the series of chips being used on the error monitor. High speed "HEF" 4000 CMOS chips were used and the system worked successfully giving confidence in results obtained.

The system was successfully commissioned and was used to monitor errors within the system giving greater in-sight into the operation of the drives performance. The results from this equipment are presented in chapter 8.0.