FLEXIBLE HIGH SPEED MACHINERY

LEE FENNEY

Doctor of Philosophy

THE UNIVERSITY OF ASTON IN BIRMINGHAM

April 1989

Thesis Summary.

The University of Aston in Birmingham

FLEXIBLE HIGH SPEED MACHINERY.

Lee Fenney

Doctor of Philosophy

1989

The thesis describes an investigation into methods for the design of flexible high–speed product processing machinery, consisting of independent electromechanically actuated machine functions which operate under software coordination and control. An analysis is made of the elements of traditionally designed cam–actuated, mechanically coupled machinery, so that the operational functions and principal performance limitations of the separate machine elements may be identified. These are then used to define the requirements for independent actuators machinery, with a discussion of how this type of design approach is more suited to modern manufacturing trends.

A distributed machine controller topology is developed which is a hybrid of hierarchical and pipeline control. An analysis is made, with the aid of dynamic simulation modelling, which confirms the suitability of the controller for flexible machinery control. The simulations include complex models of multiple independent actuators systems, which enable product flow and failure analyses to be performed.

An analysis is made of high performance brushless d.c. servomotors and their suitability for actuating machine motions is assessed. Procedures are developed for the selection of brushless servomotors for intermittent machine motions.

An experimental rig is described which has enabled the actuation and control methods developed to be implemented. With reference to this, an evaluation is made of the suitability of the machine design method and a discussion is given of the developments which are necessary for operational independent actuators machinery to be attained.

Key Library Index Words: FLEXIBLE MACHINE, MODULAR MACHINE, INDEPENDENT DRIVES, BRUSHLESS D.C. MOTOR.

To Clive

Acknowledgements

I would like to take this opportunity to express my thanks to the many members of staff at Aston University and Molins P.L.C. who gave me guidance and encouragement during this research project.

Most particularly, I would like to thank Professor K. Foster for arranging and supervising my research project and for continuing with my supervision after moving to Birmingham University.

I would also like to express my thanks to Dr. D.J. Holding and to Dr. J.E.T. Penny for assisting with my supervision after Professor Foster left the University.

Finally I would like to thank all other people who were involved with this project. If I attempt to list names, I will almost certainly miss out a few, but I include everyone who's had anything to do with all this.

Lee Fenney
April 1989

Thesis Contents.

List of figures.

List of Tables

Notation

a_l	Load acceleration
BAND	Bru–500 controller bandwidth parameter
DAMP	Bru–500 controller damping parameter
e	Exponential constant
e_k	Error signal at sample k
DM	Bru–500 drive module model constant
G1,G2	Bru–500 model constants
G_h	Sampler model
G_{Moog}	Moog controller model function
G_p	Motor model
G_{p2}	Motor model with forward path integrator
G^*	Combined sampled system model
I_a	Armature current
I_a^{sat}	Saturated armature current
J	Inertia
J_l	Load inertia
J_m	Motor inertia
K_b	Motor back e.m.f. constant
K_d	Derivative gain
Kenc	Bru–500 model constant
K_i	Motor current feedback gain
K_i	Integral gain
Kin	Bru–500 model gain
K_{Moog}	Moog 306–023 motor model gain constant
K_p	Proportional gain
K_t	Motor torque constant
K_v	Velocity loop gain
K_{vff}	Velocity feedforward gain
K1, K2,K3	Bru–500 model constants
L_a	Motor inductance
M	Mass
M_l	Load mass
N	Drive ratio
P_{diss}	Dissipated power
P_f	Friction power
P_{load}	Power required to move a load

P_M	Motor power consumption
R	Drive radius
R_a	Motor resistance
s	Laplace operator
S_k	Sum of e_k
t	Time
T_a	Bru–500 model expansion time constant
T_d	Derivative constant
T_d	Drive load torque
T_f	Load friction torque
Tf	Bru–500 model time constant
T_i	Integral constant
T_l	Load torque
T_l^P	Peak load torque
T_l^{RMS}	RMS load torque
t_m	Motion profile period
T_m	Motor torque
T_m^P	Peak motor torque
T_m^{RMS}	RMS motor torque
T^P	Peak torque
T^{RMS}	RMS torque
T_{samp}	Sample period
t_1, t_2, t_3	Profile segment lengths
U_k	Output of controller
V	Velocity
V_{in}	Motor applied e.m.f.
V_l	Load velocity
V_a	Armature voltage
W_m	Motor energy consumption
α	Angular acceleration
α_l	Load acceleration
α_m	Motor acceleration
α^p	Peak angular acceleration
α_r	Required acceleration
θ	Angular position
θ_l	Load position
θ_m	Motor position
θ_r	Required position

ϕ	Phase angle
τ	Time constant
τ_e	Electrical time constant
τ_m	Mechanical time constant
τ_{moog}	Moog 306–023 motor model time constant
ω	Angular velocity
ω_d	Damped natural frequency
ω_l	Load velocity
ω_m	Motor velocity
ω_n	Natural frequency
ω^p	Peak velocity
ω_r	Required velocity
ζ	Damping ratio

Chapter 1

Introduction

1.1. The background to the research programme.

This thesis describes work which has been performed by the author at Aston University, in conjunction with Molins High Speed Machinery Company Limited, as part of the Science and Engineering Research Council (S.E.R.C.) Specially Promoted Programme of Research (S.P.P.) into the design of high speed machinery. The S.P.P. was set up in March 1985 to improve the performance and operation of high speed machinery in Britain and includes investigations into simulation and modelling, drives and mechanisms, control and measurement, materials for machine construction and materials processing.Many of Molins' machines have inherent limitations which make them unsuitable for a modern flexible manufaturing environment, because the underlying philosophy of the manufacturing process has changed since the machines were first designed. The author's work has therefore included an examination of traditional machine designs, with particular reference to the design of intermittent packaging machinery, to identify the limiting factors in their performances. This has then been used to develop new design methods which are more suited to modern manufacturing practices.

1.2. The traditional approach to machine design.

Complex high speed processing machinery has traditionally been designed according to a philosophy of using cam–actuated mechanisms to process and combine materials as they move through the machine to form a finished product. The mechanisms can have complex motion requirements and are coupled via a series of shafts, gears, pulleys, etc. to a central prime mover. The cams provide tight control of the actuator functions, while the mechanical transmission system serves to coordinate and synchronise the interactive motions, to distribute energy about the machine and to redistribute energy during actuator braking. This method of machine design has remained basically unchanged for many years. Although improvements in materials and mechanisms have been incorporated, many of Molins' machines currently in production are in fact evolutionary developments of designs which were first produced in the 1920's.

In recent years, the trends in machine operation have been towards shorter duration, high–speed runs, with the machines being reconfigured between runs to

accomodate changes in product specification. The traditional machine configuration is very limited in its suitability for this type of operation. The speed of the machines can not be increased easily because high production speeds introduce fatigue and fretage failures in the mechanical parts. In order to compensate, these parts have to be redesigned or reinforced and this often leads to the use of increasingly massive machine elements. Such machines are expensive to produce and generate high intensity acoustic noise. In addition, high speeds are often not possible because the cams are designed for an optimum peak operating speed, so when higher speeds are attempted, the inflexible cam profiles induce intolerable acceleration loads on the product, which causes product damage. Further problems occur when attempting to reconfigure the machines between runs. The inflexible nature of the fixed function machine elements means that such reconfiguration can require significant periods of time, with some Molins' machines requiring up to eight man–weeks to reconfigure for product changes.

1.3. The independent actuators approach to machine design.

The author's research programme has investigated a new approach to machine design which is being developed to suit the needs of modern manufacturing methods. In this, the cam–actuated mechanisms are replaced by a system of electromechanical servomechanisms, acting under software coordination and control. The use of independent actuators reduces the number of wear parts in the machine, while software control introduces flexibility. Thus the two principal limitations to machine performance of poor high speed operation and down time during product change are reduced in an independent actuators machine.

The development of viable independent actuators machinery has been made possible by recent improvements in several areas of technology. The most important of these are the availability of high performance electric servodrives and the development of powerful high speed distributed real time control systems. The very stiff servodrives enable complex motion profiles to be independently driven, while distributed control methods enable large networks of independent servodrives to be concurrently controlled and their interactions to be properly coordinated. This research programme has included an examination of the new servodrives and the development of methods which enable them to be software coordinated and controlled.

As part of the research work, the author has investigated the problem of retrofitting two independently driven and software coupled interactive intermittent mechanisms from a Molins packaging machine, which were originally cam driven and

mechanically coupled. Each element in the solution of this retrofit problem has been developed and tested with the aid of simulation modelling before implementation in actuator hardware. In this way, the author has been able to expand the simulation modelling work to develop techniques for the driving, software coupling and controlling of sets of independently driven mechanisms.

1.4. Related S.P.P.work.

The author's research programme is an S.E.R.C. S.P.P. investigation, so it has formed part of a larger programme of work aimed at improving high speed machinery. The work has therefore been modelled to complement related work being performed elsewhere, most particularly in a Molins research programme in the Department of Electrical Engineering at Aston University. This has investigated the problem of retrofitting two phase synchronised drives to continuous motions from a Molins' rod making machine. Although the problem of synchronising continuous motions is dissimilar to the author's work on asynchronous motions, many aspects of the drive simulation and implementation work are similar. So close liasons have been maintained between the two projects.

Apart from the second Aston / Molins investigation, related S.P.P. work is being performed at Birmingham University and Liverpool Polytechnic. Discussion meetings have been held with investigators from these research programmes to exchange results and avoid duplication of effort. It has been of interest that the four related research programmes each independently made similar selections of actuator hardware. This has enabled confidence that correct selection decisions have been made and has introduced mutual areas for discussion at progress meetings.

1.5. Outline of the thesis.

The work described in this thesis has been separated into seven chapters, with each of the following chapters covering a separate area of the overall research programme. Since the subject areas covered by each chapter overlap, references to other chapters are included throughout the text to enable the reader to understand the relationships between the separate aspects of the work.

A survey of related research publications is discussed in chapter 2. This includes descriptions of work being performed in the related S.P.P. investigations. Areas of interest not covered in available literature are identified and the goals for the research programme are stated. Chapter 3 examines product processing machinery and identifies the operational roles of separate machine parts. These are then used to

define the requirements for independent actuators machinery and a discussion is included of the optimum flexible machine designs.

Chapters 4 and 5 describe the development of methods for implementing independent actuators machinery. Chapter 4 describes the software coupling methods, including the development of synchronisation algorithms which are of flexible form and the design of a controller structure which supports machine modularity. Techniques are described which enable the synchronisation algorithms for sets of independent actuators to be tested using simulation modelling and examples are given of the use of these techniques to simulate multiple independent actuator systems. Chapter 5 describes the design of servomechanisms for independent machine motions, including developments in new high performance electric servodrives and the requirements for feedback controllers which are suitable for synchronised actuators applications.

Chapter 6 describes the implementation of the two intermittent actuator system in the laboratory. This includes descriptions of the failures which occured in the new drives and a discussion of the problems associated with retrofitting independent actuators to traditionally designed machine configurations. Chapter 7 is a discussion covering all the work which has been performed during the research programme and includes the conclusions which have been reached by the author during the work. The discussion includes proposals for further work which must be performed so that independent actuators machinery become an operational concept.

The Appendix of the thesis includes details of the servomechanism design work which has been performed to assist the laboratory application. Appendix 1 describes the development of the drive selection procedure, which was specifically modelled to enable the optimum brushless d.c. servomotor to be selected for an intermittent actuator application. Appendix 2 describes the development of a microprocessor based controller for a brushless d.c. servodrive, including the design of the control algorithms using a w–plane analysis and the configuration of controller hardware which best suited the requirements for the independent actuators application. Appendix 3 describes an evaluation of control algorithms for brushless d.c. motors which was performed to assist the selection of a commercially produced position controller. This includes extensive use of simulation for the comparison of algorithm performances.

Chapter 2

The problem defined.

2.1. Introduction

In chapter 1 the research aim of developing flexible processing machinery incorporating independently driven mechanisms was introduced, along with the S.E.R.C. S.P.P. which this work forms part of. The tasks of selecting drives for the mechanisms, developing controllers for the dynamics of the drives, simulating the controlled servomechanisms, developing algorithms for the coordination and synchronisation of the separate motions and implementing the complete software–coupled system involve work in several engineering disciplines, including aspects of mechanical, control, electrical, electronic and software engineering. It was clear that the author's research programme could not include detailed study of all aspects of the problem, even though each aspect had to be included in order to achieve the final operational independent actuators system. So the work has aimed at complementing work performed elsewhere as part of the S.P.P., with the intention that duplication be avoided.

In this chapter, published literature relating to the author's research programme is reviewed, along with work which is currently being performed elsewhere on the S.P.P.. Areas of knowledge are identified in which no published material is available and which are not being investigated elsewhere. These are used to define the area in which the author's research programme will be concentrated.

2.2. The S.E.R.C. specially promoted programme of research into the design of high speed machinery.

The investigations which are being carried out as part of the S.P.P. are described by Sweeney (1). There are three projects whose work and results are directly applicable to the author's work. Seaward and Johnson's (2) work at Aston University, which is described in chapter 1, has included detailed examination and multi–phase simulation modelling of similar brushless d.c. drives to those employed in this research work, although the methods employed for the continuous synchronisation and control of their drives can not be transferred to the intermittent case. In a research project at Birmingham University, Leighton, Jones and Fletcher (3), (4) are investigating the use of independent drives for incremental sheet feed systems. The primary aim is to replace the mechanical feed systems and thus reduce set–up time and increase flexibility. The work has not aimed at an in depth

investigation of independent actuators machinery, but instead has concentrated on incorporating independent controlled drives within an existing traditional machine system. This has included extensive development of servocontrollers for controlling the dynamics of the sheet feed motion.

At Liverpool Polytechnic, Stamp and Rees Jones (5) are investigating the use of brushless d.c. drives to actuate a twin drive pantagraph mechanism. This is used as a transfer mechanism having a complex motion profile, with independent drives enabling the follower's motion to be programmable, which reults in an extremely versatile machine element. The research work has concentrated on the development of the pantagraph mechanism and the control algorithms for the independent drives. In a related paper, Rees Jones (6) describes further cases in which independently driven motions have been incorporated into traditional types of machine designs. This includes examples of how mechanisms may be designed to compensate for errors resulting from poorly synchronised motions and of the design of optimum motion profiles for independent drive systems, based on harmonic series functions. However, in each case the work involves adapting an independent drive system to suit an existing machine design, rather than an investigation into the problems associated with constructing machines entirely from independently actuated mechanisms

2.3. Independent actuators machinery literature not related to the specially promoted programme of research.

Apart from publications linked to the S.P.P., there are other relevant discussion papers which describe independent actuators processing machinery. Bruno (7) gives an interesting summary of the history of the movement towards increasing the number of drives incorporated into processing machinery. Beginning with early manufacturing systems in which all machines are coupled to a single water–wheel driven main shaft, Bruno describes the trend towards "drive splitting," in which each machine was driven by its own motor. The drive splitting trend then continued towards the current ideas of incorporating many drives in a single machine. However, Bruno describes the use of independent drives as being "more gimmicks than worthy" because of their increased energy consumption. Little mention is made of the advantages of flexibility, which is the prime reason for the movement away from centrally driven manufacturing systems. In the discussion accompanying the presentation of the paper, Bruno stated that all applications of independent drives which he had encountered involved "non–interfering mechanisms," so tight synchronisation of the machine motions was not critical.

In a similar form of discussion paper presented at the same conference, Marks (8) describes the movement towards independent actuators machines, beginning with 1950's designs of cam–actuated, centrally driven machines and following the development to the present, in which synchronised drives are used for material feeds and product removal. This introduces flexibility and reduces down time for size changes, which are programmable, but the machine itself is not a true independent actuators design. Instead it is an evolutionary design in which independent drives have been used to improve certain functions within the machine, with the cam–actuated operating characteristics remaining largely unchanged. However, it is stated that the introduction of independent drives for some functions serves to reduce wastage during run up to speed and improves the reliability of the machine.

Work which is being performed at Loughborough University towards a universal controller for manufacturing machinery is described by Weston, Harrison, Booth and Moore (9). This work has aimed at developing a framework for the design of flexible machine control systems which can serve as a successor to programmable logic controllers. The natural hierarchy of machine control functions is introduced and the controller is described as being capable of supporting this hierarchy in both simple and complex machines. The hierarchy forms the basis for the control system topology, with the communications between controller elements providing the connections between hierarchy elements. The paper describes the configuring and control of programmable machine elements, such as robots, tool selectors, work handlers, such that a machine can handle a wide range of related tasks. This contrasts with the work described in this thesis, in which the task of manufacturing a machine product and the overall configuration of the machine remain constant, with only the specification of the product changing between runs.

2.4. Literature relating to the selection of drives for the independent motions.

The loads to be driven and the characteristics of their motions are pre–defined in the machine's specification. In the intermittent mechanisms considered in this research programme, the specifications are for rotary and angular loads which must be moved through defined distances in fixed time periods, with pauses between motions. The requirement is for independent drives which can move the loads with the required dynamic performance, in terms of settling time and accuracy. In a previous study at Aston University, Firoozian and Foster (10) performed a comparison of sevo drives to assess their speeds of response and their abilities to

respond to external torques. The study included ceramic magnet d.c. motors, rare earth magnet d.c. motors, brushless d.c. motors, a.c. motors, stepping motors and hydraulic motors. It was found that for the high power rating, high load inertia applications being considered in this project, electric motors produce a faster speed of reponse than hydraulic motors. The emphasis in this research programme has been towards using electric motors.

Since the above mentioned comparison was performed, significant improvements have been made in large electric motor technology, particularly in the high torque, low inertia brushless motors which are used in the aerospace, robotics and numerically controlled machine tool industries. Many authors have described these improvements and the motors which have been produced. For the mechanical engineer, Carlisle (11) provides an introduction to the new motor technology which does not delve deeply into the electical design of the motors, but instead describes the operational advantages which the new motors provide. Barber (12) introduces the operation of the new motors, including diagramatic descriptions of their construction and of their drive amplifiers, but again the paper can be readily understood by the engineer who is new to electrical engineering theory and practice.

In the early part of the research programme, the characteristics of switched reluctance motors indicated that they would be suitable for independent actuators applications. Lawrenson, Stephenson, Blenkinsop, Corda and Fulton (13) describe the construction and operation of these motors, which were developed jointly at Leeds and Nottingham Universities. The motors are described as having a very high efficiency and a torque speed curve which is very similar to that of a d.c.motor, with voltage or current control modes being readily implementable. Thus the motors should be suitable for position servo applications, with the suitability being enhanced by the inclusion in the motor of a position encoder. However, in practice the high electrical efficiency of the design has led towards their being developed for constant speed applications, with no commercial manufacturer currently producing a servocontroller for the motors.

The implementation of the independent actuators has concentrated on using brushless d.c. motors under feedback control. There are many works describing the construction and operation of these motors, one of the most comprehensive being the book by Kenjo and Nagamori (14). However, many authors do not agree on what constitutes a brushless d.c. motor, with the terms a.c. synchronous motors, d.c. synchronous motors, brushless a.c. motors and asynchronous motors all commonly being used to describe the same basic design. The basic construction of a

brushless d.c. motor is shown in figure 2.1. The motor has a permanent magnet rotor and the electric windings are mounted around the stator. A rotor position transducer is used to provide information for controlling the current waveforms in the windings.

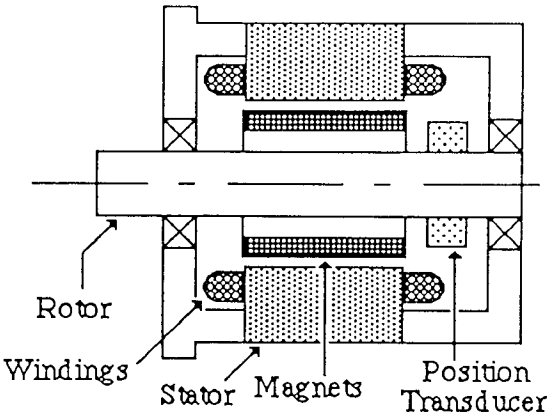


Figure 2.1. Brushless d.c. motor construction.

The operating principles of a brushless d.c. motor are essentially the same as those of a closed loop permanent magnet stepper motor. Much of the early development work concentrated on the closed loop control of stepping motors in an attempt to improve position response times and accuracy. Liska and Ulrich (15) implemented closed loop stepper motors in office machinery, such as printers and tape machines. They in fact define stepper motors as having no rotor position sensors and brushless d.c. motors as having sensors. The use of feedback produced a motor torque/speed characteristic which was the same as that of a d.c. motor and gave greater efficiency, higher starting torque and higher continuous torque. However, the increased inertia of the brushless d.c. motor and the high inductance of the windings gave it less favourable transient performance characteristics.

In more recent times the dividing line between brushless d.c. motors and closed loop stepper motors has become more clear due to improvements in two areas of motor design. The most important of these is the use of rare–earth magnets in the rotors, as described by Horner and Lacey (16). The magnets have high coercivity, making them extremely resistant to demagnetisation and they are shorter in the direction of magnetisation, so they can be oriented radially instead of circumferentially. This reduces the winding inductance in addition to decreasing the rotor inertias. The commutation system therefore runs under less load, which improves reliability. As will be demonstrated in Appendix 1, these improvements reduce the mechanical and electrical time constants of the motors, which results in improved transient performance and lower energy consumption. Horner and lacey also describe the second improvement in the design of the drive systems, which is the use of high current, high frequency switching transistors for the commutation.

Since the magnets have very high demagnetisation current ratings, the peak rating of the drive is greatly increased, while the very efficient cooling characteristics which result from mounting the windings around the outside of the motor enable high continuous currents to be tolerated.

Further developments in rare–earth magnet motor performance are discussed by Christensen (17). The torque generated by a motor is proportional to the magnetic flux density and a high flux density will produce a higher motor torque. Early rare–earth magnet brushless d.c. motors used samarium cobalt magnet rotors, which have a flux density approximately two and a half times greater than that of ferrite magnets and a demagnetisation current rating approximately four times that of ferrite magnets. So samarium cobalt magnet motors are capable of generating substantially larger torques. However, samarium cobalt mgnets are difficult to machine due to their brittle structure and are expensive to produce. A newer material, neodymium–iron–boron, has even higher flux density and demagnetisation ratings, coupled with lower cost and more acceptable machining properties than samarium cobalt. Christensen predicts that neodymium–iron–boron will soon be the preferred material in brushless d.c. motor construction, but due to its still much higher cost than ferrite magnets, these will continue to be widely incorporated in low cost d.c. motors.

For accurate modelling and optimum selection of the motors, it is important to understand their operating characteristics. The book by Tal (18) explains that although the brushless d.c. motor is a multiphase a.c. device, its characteristics are similar to those of a conventional d.c. motor, which is the reason for its title. The torque generated by the motor, T_m, is proportional to the current in the windings, I_a,

$$T_m = K_t . I_a \qquad (2.1)$$

where K_t is the torque constant of the motor. The current is produced in the windings by applying a voltage and is reduced by back e.m.f.. In Laplace transform notation, the current equation may be written as,

$$V_a = (L_a . s + R_a).I + K_b . \omega_m \qquad (2.2)$$

Where V_a is the applied voltage, L_a is the inductance and R_a is the resistance of the windings, ω_m is the motor's angular velocity and K_b is the back e.m.f. constant of the motor. For brushless motors having sinusoidal current exitation waveforms, K_b

is equal in magnitude to K_t. Assuming that the load consists of a rotational inertia J_m and a steady load torque T_l, the motion of the motor is given by

$$T_m = J_m\, s.\omega_m + T_l \qquad (2.3)$$

Thus the basic block diagram of the motor is shown in figure 2.2.

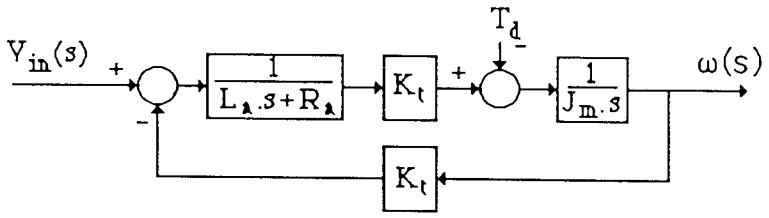


Figure 2.2. Brushless d.c. motor block diagram.

Two important measures of the motor's performance are its continuous and peak torque ratings. The continuous torque rating is defined by Barber (19) as being the R.M.S. torque requirement for a "worst case" duty cycle. This may be evaluated by the equation,

$$T^{RMS} = \sqrt{\frac{T_1^2 + T_2^2 + \ldots + T_n^2}{t_1 + t_2 + \ldots + t_n}} \qquad (2.4)$$

where T_n is the motor torque required for time t_n. Barber includes several examples of the use of this equation for the calculation of motor torque requirements for robotics applications. Many authors describe the peak torque rating of the drive as being limited only by the demagnetisation current of the magnets, which can be misleading since for rare earth magnets this value can be very large – typically in excess of 400 amps. As a result, many small brushless d.c. motors are specified by manufacturers as producing very large peak torques. In practice, however, this is not the case. Tomasek (20) explains that whereas in conventional d.c. motors the saturation current is often greater than the demagnetisation rating, for brushless motors the saturation current introduces a *system* peak torque limit which is less than the motor peak limit. In addition, the thermal time constant of the drive circuitry can introduce a system RMS torque limit which is also less than the motor limit. In the applications considered in this project, the thermal time constant of the drive circuitry is in the range 5 to 10 seconds, while the period of the motion is less than 200 msec. Thus for our applications the RMS requirement of the motions and dwells may be considered together and compared against the RMS ratings of the motors.

Since the operating characteristics of brushless d.c. motors are similar to those of conventional d.c. motors, the development of a procedure for the selection of the optimum one could be based on earlier d.c. motor selection work. A large amount of work has been published relating to the selection of drives for incremental motion systems. Tal (21) and Jocz (22) define many of the relationships which form the basis of drive selection calculations, including the selection of optimum drive ratios and motion profiles. However, Tal's work was based on minimising the motor temperature rise, since this is the major limitation in d.c. motor performance. For brushless drives, the improved cooling characteristics enable the emphasis in the selection and profile design to be moved towards the minimising of the total energy consumption of the drive. An analysis based on this is included in Appendix 1. Tal demonstrates that the optimum drive ratio requires inertia matching, in which the drive ratio, N is given by the equation,

$$N = \sqrt{\frac{J_l}{J_m}} \qquad (2.5)$$

This result is confirmed in Appendix 1. Archer and Blenkinsop (23) demonstrate the effect that non–optimal drive ratio selection can have on the temperature rise in the motor. Ratios within the range from on half to twice the optimum value are acceptable, but outside this range unacceptable temperature rise and drive inefficiencies occur.

A further property of Tal's inertia matching relationship is the minimising of the motor's power rate requirement, as stated by Newton (24). Power rate is defined as the rate of change of mechanical power and is demonstrated in Appendix 1 to be a very important measure of a drive's ability to accelerate a load. Power rate enables several different sizes of load to be equally compared for their abilities to drive a load. However, the author is aware of no publications in which this capability is clearly demonstrated. Newton demonstrates that for a motor to be capable of driving a load, its power rate must be at least four times that of the load, but this is not clearly derived or stated, even though it is a fundamental requirement for any drive. Arnold (25) and Floresta (26) both state this requirement more clearly than Newton, but each paper is written from a motor designer's standpoint to define the capabilities of given motors, rather than the servo designers standpoint, where finding a motor to drive a given load is the requirement. Floresta uses motor power rate vs. speed curves to demonstrate that for servodrives, the rotor inertia acts as an impedance to the application of output power and is in effect a parasitic loss. So smaller motors are in fact preferable for a given load requirement. In addition, it is

shown in Appendix 1 that smaller motors, by virtue of their higher natural frequencies, can produce stiffer reponses and that they also use less energy to drive a profile.

A final reference which was used regularly throughout the drives selection and evaluation work is the book by Electro–Craft (27). This contains a great deal of information relating to d.c. and brushless d.c. motors, which has been compiled in a very readable form.

2.5. The design of controllers for the dynamics of the motions.

The development of algorithms for controlling the dynamics of the motions involved extensive use of simulation modelling. Brushless d.c. motors have inherently low damping ratios, typically less than 0.2, so it was necessary to include compensation to produce a stable response. Bell and de Pennington (28) demonstrated the effectiveness of using minor loop acceleration feedback to actively damp electrohydraulic servomechanisms having very low inherent damping ratios. Tal (29) demonstrates the use of minor loop current feedback using a frequency plane analysis. The minor loop introduces a phase lead at the 0 dB gain crossover frequency, thereby enhancing closed loop stability, while not appreciably affecting the bandwidth. Thus current feedback does not have the potential dissadvantage of increased high frequency noise which can result from a wider bandwidth. This technique has been demonstrated by the author (30) to be effective for the control of d.c. servo drives. The author's work involved the use of simulation in the design of a position control servo for a programmable engraving table. Many of the simulations in the earlier work have been extended in the work described in this thesis. The work also contains descriptions of the operation and use of the ACSL simulation language, which was developed by Mitchell and Gauthier (31) and is used throughout this work.

The simulation models enabled the position control algorithms to be developed off–line, which avoided potential damage to the system hardware. Since extensive brushless d.c. motor controller development is being performed elsewhere as part of the specially promoted programme of research, an in–depth investigation and development of optimum control algorithms was not performed as part of this project. Instead, the aim was to develop a controller which would achieve the required servo performance, within the constraints imposed by the drives. The loads to be driven have constant inertias with position and the friction loading is constant and small, so classical control methods were used. Towards this, several textbooks proved to be useful. Raven (32) gives a comprehensive introduction to

control engineering which was referred to extensively throughout this work. The book by Borrie (33) is a more modern control system reference, which includes a more comprehensive description of digital control methods and of proportional plus integral plus derivative (p.i.d.) controller optimisation. Atkinson (34) describes the operation and use of p.i.d. controllers in a book which makes reference to practical engineering aspects of servomechanisms and process controller design. In the chapter on servomechanism dynamics, Atkinson introduces the idea of using velocity feedforward to reduce response phase lag, whilst not altering system stability. Although the method is open–loop in character, for a laboratory implementation where recalibration is not a problem, feedforward can be used to obtain an acceptable response.

The simulation language enabled discrete software control algorithms to be simulated controlling a continuous model of the drives. The finite difference model of a p.i.d. controller is given by Sandoz (35) as,

$$U_k = K_p \cdot \left[e_k + \frac{T_{samp} S_k}{T_i} + \frac{T_d (e_k - e_{k-1})}{T_{samp}} \right] \qquad (2.6)$$

where U_k is the output of the controller, K_p is the controller gain, e_k is the error at sample k, T_{samp} is the sample period, S_k is the sum of e_k, T_i is the integral constant and T_d is the derivative constant. This equation was used as the basis for the controller development work, with finite difference feedforward and software switches on controller functions being evaluated to assess their performances.

2.6. The development of the software coupling controller structure and algorithms.

The software coupling controller must be capable of synchronising the dynamics of groups of actuators which are each independently moving under feedback control. The controller must have a flexible structure which supports machine modularity. A multiprocessor implementation of the software coupling controller is being developed at Aston as part of additional S.P.P. research work. To complement this, the author has investigated three fundamental requirements for the succesful implementation of software coupling:

(i) The method by which independent intermittent dynamic systems may be synchronised.

(ii) The structure of the controller which enables a flexible implementation.

(iii) The verification by dynamic simulation of the software coupling algorithms.

Despite extensive searching, including on–line searches of computer data bases, the author has found no references relating to the problem of synchronising intermittent dynamic motions, or to simulating the software synchronisation of dynamic systems. These two problems have formed an important part of the author's research. The solutions which have been developed for these problems are described in chapter 4.

The development of a flexible structure for the controller was able to make reference to work which has been performed in the development of high performance computing systems. Enslow and Saponas (36) state the goals which have motivated this work as being:

(1) Increased system productivicty
Greater capacity
Shorter response time
Increased throughput
(2) Improved reliability and availability
(3) Ease of expansion
(4) Graceful growth and degradation
(5) Improved ability to share system resources

It is interesting to note that these reasons, if slightly reworded, could have been put forward as justification for the adoption of independent drives in machinery. Increased productivicty is hoped to result from increased machine speed and reduced machine down–time for product changes, improved reliability is hoped to result from the use of less wear parts, ease of expansion and degradation should result from the inherent flexibility of the design method and improved ability to share resources will come from the ability to coordinate and control all levels of the manufacturing process. Enslow and Saponas introduce distributed control as being suitable for achieving some, or all, of the above listed goals.

A distributed control system is described by Sloman (37) as being one in which several processors cooperate to achieve an overall goal – the control of a process or plant. Sloman describes the advantages of distributed control as being reduced costs, increased modularity, improved performance and flexibility, improved reliability and ease of development and maintenance. Again, these could be stated as reasons for introducing independent drives into machinery. There are many

similarities between distributed processing systems and independent actuators machines, with the similarities indicating the suitability of a distributed controller scheme for the synchronisation solution. In distributed control, several processors may be distributed throughout the system, with communications enabling them to interact. The processing configuration is structured to meet the needs of the system and it is the structure which provides the single system goal to all the independent components. In independent actuators, the drives are distributed throughout the machine, with "communications" between actuators taking the form of the passing of products. The actuators are configured to meet the needs of the manufacturing process and it is this which provides the single system goal to all the separate components.

The nature of the passing of products between actuators provides an indicator to the optimum means of interconnecting the control processors. There are two principal methods of connecting processors, which are described by Enslow and Saponas, tight coupling and loose coupling. Tightly coupled processors use shared resources (i.e. shared memory) to achieve coupling. Unfortunately, the access difficulties this introduces in large systems introduces delays, which eventually limit a system's capacity for expansion. These difficulties limit the suitability of tightly coupled controllers for large control problems. The coupling method which has found most favour in large distributed computing systems is loose coupling. Loosely coupled processors have no direct sharing of primary memory. Instead, the processors communicate at the input/output level along hardware channels. In order that a communication be completed, the active cooperation of both processors is required. This is analagous to the passing of products between actuators. In order that a product be passed along a spatial "channel," the active cooperation of all involved actuators is required. In addition, the flexibility of loosely coupled systems makes tham most suitable for independent actuators control. Research is proceeding at Aston University into the development of loosely coupled distributed controllers for independent actuators machinery.

In developing a controller structure which was suitable for the machine control problem, it was most important that flexibility be maintained at all levels of control. The structure which the author has developed is described in detail in chapter 4. This was developed using computer interconnection structures which have been proposed by several authors. Anderson and Jensen (38) published one of the earliest surveys of inteconnection methods, including an introduction to the design decisions which must be made to achieve an operational system. The paper is aimed at computer hardware engineers and as such concentrates on computer

interconnections, rather than the development of the controller scheme itself. However, the paper does introduce the interconnection methods most commonly used at the time and discusses the merits and drawbacks of each.

The controller structure which has been developed is a hybrid of hierarchical and pipeline control. Hierarchical control is described by Weitzman (39). This consists of a ziggurat type structure of processors, as shown for a three level hierarchical controller in figure 2.3(a). Generally, the capability of the processor increases as the top of the ziggurat is reached, with the lowest level processors performing specialised tasks, such as data aquisition or control functions and the highest level having a more general purpose capability.

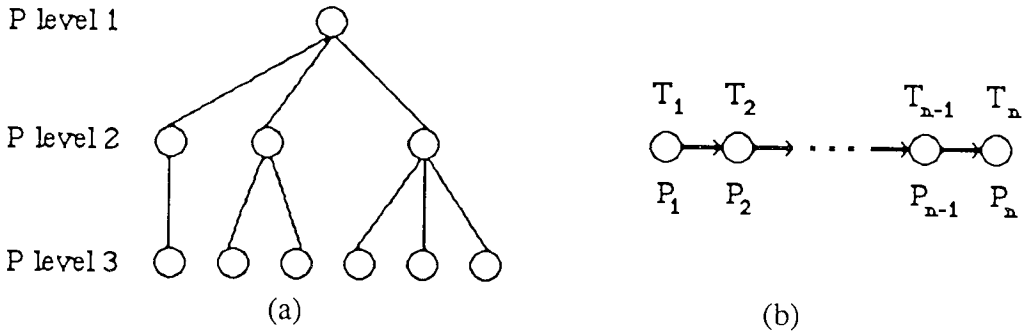


Figure 2.3. Distributed controller structures.
(a) Hierarchical control
(b) Pipeline control.

The principal limitation of hierarchical control is found in the capacity of the higher level processors. In very large systems these have to perform large amounts of processing, which limits the response frequency of the controller. This limits the controllers capacity for expansion. A means of avoiding this limitation was found in introducing pipeline control. This is described by Paker (40) and is represented in figure 2.3 (b). In this a task T is subdivided into consective sub–tasks, T_n, where each sub–task is independent of others and the output of tasks T_i feeds the input of tasks T_{i+1}, for all i. Note that when processor P_i completes task T_i it becomes free to start the replication of a similar task. Thus the pipeline flow is analagous to the flow of products through a machine. The introduction of a pipeline flow to the hierarchical structure removes the processing limitation which otherwise results in large systems. This is described in detail in chapter 4.

2.7. The outline for the areas of investigation.

From the work discussed in sections 2.2 to 2.6, the gaps in the knowledge in the literature reviewed can be summarised as follows.

(a) Independent actuators machinery.

The development of methods for the design of machinery composed of independently driven electromechanical functions. Assessing the suitability of traditionally designed, cam–actuated machinery for independent actuator driving.

(b) Independent drives.

Defining procedures for the selection of high performance electromechanical drives for independent intermittent machine motions. Assessing the suitability and reliability of the drives for processing machine applications.

(c) Coordinating and synchronising independent motions.

Investigating methods for the software coupling of independent, intermittent, interactive machine actuators. Developing procedures for coordinating the interactions of the actuators to enable a functional machine system. Developing a controller structure which will support the software coupling of sets of actuators and does not have an implementation specific design.

(d) Simulating independent actuators.

Developing methods for designing software coupling algorithms with the aid of dynamic simulation models of independent actuators.

It will be the objective of this research project to investigate all aspects of the gaps in knowledge listed in this section.

Chapter 3

Independent Actuators Machine Systems.

3.1. Introduction

This chapter examines traditional product processing machinery to assess the ways in which it is not well suited to modern manufacturing methods. The operational roles of machine parts are identified and methods for implementing these roles which are more suited to modern manufacturing requirements are proposed. Finally the advantages of modular machine designs are discussed and the structure and interconnection of machine modules is defined.

3.2. Traditional high – speed processing machinery.

Figure 3.1 shows the mechanical assembly of a typical high speed processing machine. The example shown is the Molins S.P.1 packaging machine from which the third transfer and arbor drum mechanisms of the demonstrator project described in chapter 6 have been taken.

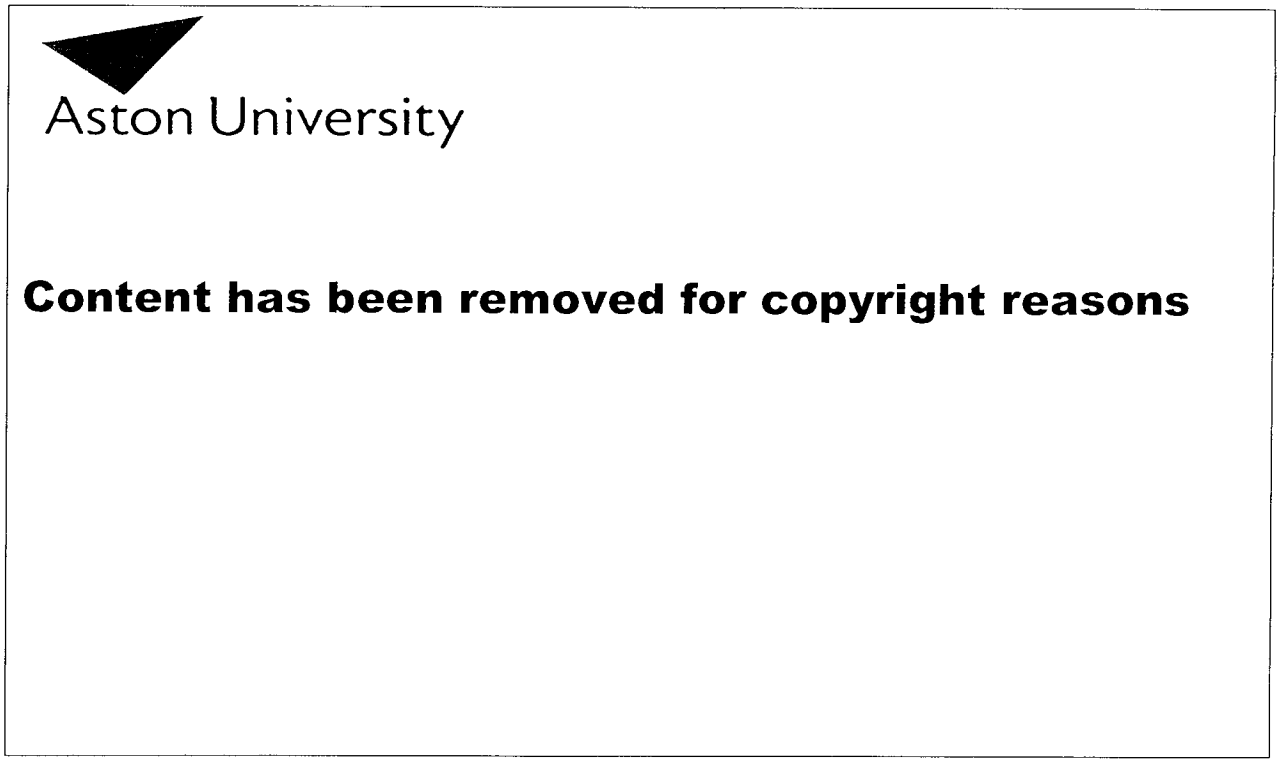


Figure 3.1 High speed processing machine mechanical assembly.

Partially processed materials enter through input stages and are moved through the machine in a sequence of rotary and linear motions as a series of functions

are performed on them. These functions serve to combine and process the materials to form the machined products, which leave the machine through output stages.

Each machine function has a mechanism to perform its work. In figure 3.1 there are ten intermittent mechanisms labelled, including the third transfer and the arbor drum. The motions of the mechanisms are controlled by cams. These provide open loop control, so their motions are coupled by a mechanical transmission system. This coupling serves to synchronise the interactive motions of the mechanisms and to coordinate their actions, so maintaining a flow of materials and products through the machine. An additional property of the coupling is that it re-distributes about the machine the energy dissipated during the braking of the separate mechanisms. As a result of the mechanical coupling, actuation for all the separate functions can be provided by a single electric drive, which for the S.P.1 is a 5.9 Kw induction motor.

The S.P.1 is typical of a wide range of high speed machines that are currently in production for product processing applications. Although advanced materials and computer aided methods are used in their design and manufacture, the underlying design philosophy of using singly actuated, mechanically coupled, cam controlled functions has remained unchanged for many decades. As a result, modern machines have a similar configuration to those produced in the 1920's.

In recent years the trend in manufacturing has been towards high speed, short duration production runs, with the machine being reconfigured between runs to accomodate changes in the product's specification. This trend has identified several distinct limitations in the traditional design of processing machines. Primary amongst these limitations is that the machines are inflexible. Extensive use of mechanical parts means that changes to machine functions can require substantial manual reconfiguring and replacement of parts. In many cases it can take several man weeks work to implement product changes, with significant down time costs being incurred between production runs. This greatly inhibits the machines' suitability for short run manufacturing.

In addition, the machines have a long design and development lead time, which is typically five to seven years. This makes machines very expensive to develop and inhibits the production of machines for specialised applications

Another important limitation of the machines is their unsuitability for high speed operation. Stress and fatigue failures in mechanical links and fretage failures in cams are increasingly common at higher operating speeds. As a result mechanical parts have to be redesigned or reinforced, which increases the complexity and cost of the machines. The resulting machines are physically massive and expensive to produce, while high speed operation of the high inertia moving parts produces large intensity vibrations and acoustic noise.

These machine difficulties led to the realisation that the centrally driven, mechanically coupled approach to machine design needed to be changed if improvements in machine performance were to be achieved. To identify ways in which the design methods could be changed, it was necessary to firstly identify the operational requirements of all the parts of the machine. This then enabled the development of alternative methods by which the operations could be performed in a manner more suited to modern manufacturing trends.

3.3. Re-configuring the elements of product processing machinery to overcome the limitations of the mechanical system.

The elements of processing machinery can be categorised into five divisions:

{i} Functions.
{ii} Mechanisms
{iii} Actuators
{iv} Controllers
{v} Couplings.

Functions are the processing activicties which are performed by the machine on the product. The design of the machine centres on enabling the functions to be performed properly and in the correct sequence. Each function requires a mechanism to perform its work. Each mechanism requires an actuator to provide energy for its motion and a controller to ensure it performs its function correctly. Finally, for all the separate functions to combine to produce a machined product, their motions must be coupled. The methods of application of these elements to a given set of machine functions and mechanisms may be seen by considering the five functions shown in figure 3.2.

The five functions can each be seen in a wide range of processing machines. The sequence shown is based on transfering materials in to and out of the arbor drum. The drum incorporates a set of mechanical sub-functions which are triggered by its motion to process products which are placed in the arbors,

which are mounted about its circumference. The first function in the sequence is a motion of products towards the drum, which is carried out by an intermittent belt mechanism. A transfer slider performs the function of transfering the products from the belt in to the arbor drum. A similar slider transfers the products out of the drum and on to a second intermittent belt, after the product has been processed by the drum. The final function is the moving of the processed products away from the drum, which is performed by the second belt mechanism.

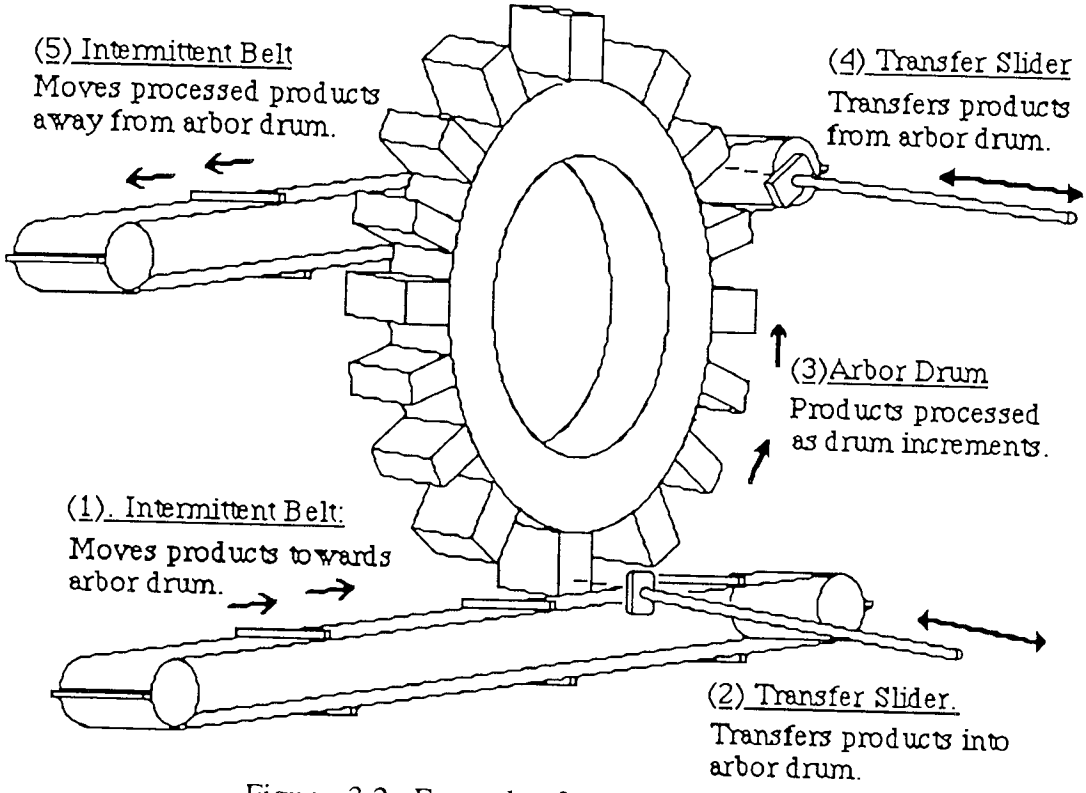


Figure. 3.2. Example of a machine function sequence.

The configuration shown therefore contains two divisions of processing machine elements, functions and mechanisms. For the machine functions to be performed correctly, the elements of the three remaining divisions, actuation, coupling and control, must be supplied. The traditional approach to this would be to use a single motor to actuate the five functions, a mechanical transmission system to couple the separate motions, cam mechanisms to control the linear sliders and geneva cam mechanisms to control the rotary belts and the drum. This arrangement is shown in block diagram form in figure 3.3.

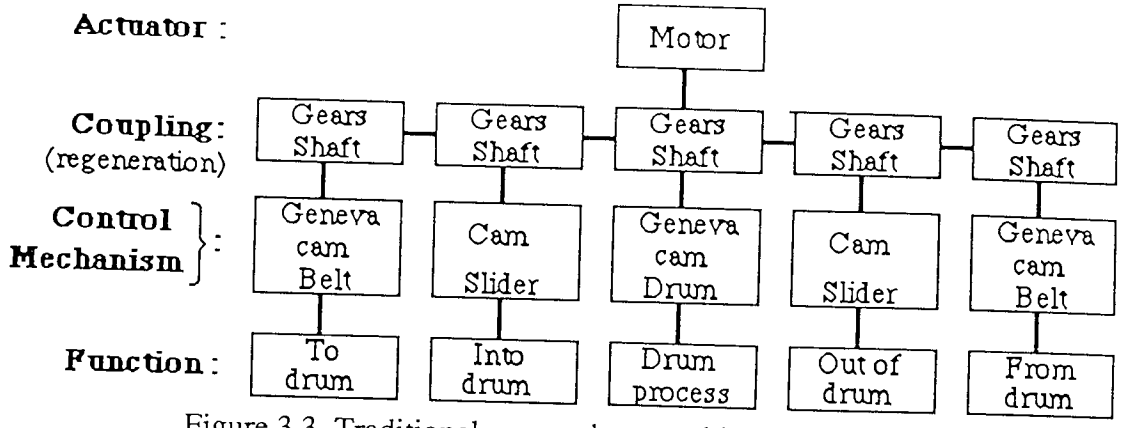


Figure 3.3. Traditional approach to machine design.

Figure 3.3 demonstrates the operational hierarchy as configured in traditional machinery. The five operational levels are formed by the five divisions of machine elements listed above. The operation of each level is directly governed by the level above it, mechanisms govern functions, controllers govern mechanisms, etc. However, the use of cams serves to combine the operations of mechanisms and controllers. A cam is in effect a self controlled mechanism, which when running at constant speed provides open loop control of the function. All levels below the actuator are in motion. Since this forms the upermost level of the hierarchy and the functions form the lowest, it follows that all the intermediate levels are in motion. It is this extensive use of moving parts between actuator and function which inhibit machine performance. To reduce the number of moving parts, it is necessary to reduce the number of levels separating actuator and function, which by definition are both in motion. However it is necessary to perform the operational roles of all five divisions if machine integrity is to be maintained. A method of achieving this which has formed the basis of this research is the use of independent actuation for each function.

Figure 3.4 demonstrates how this may be implemented for the five function system. Each mechanism has its own actuator to provide the energy for it to perform its function. The action of each actuator is controlled using a feedback controller, which can be reprogrammable if software algorithms are employed. Since feedback control is used, the motions of the functions can be software coupled. Thus the mechanical transmission and open loop cam control systems which form the limiting elements of traditional machinery have been replaced in this configuration by software, which introduces flexibility into the machine. The operational integrity of the machine is maintained by each function having upward access to each of the five divisions of machine elements. However, there are only two levels below the actuator, with software coupling and control

forming the upper two levels of the hierarchy. Thus the number of moving elements is substantially reduced.

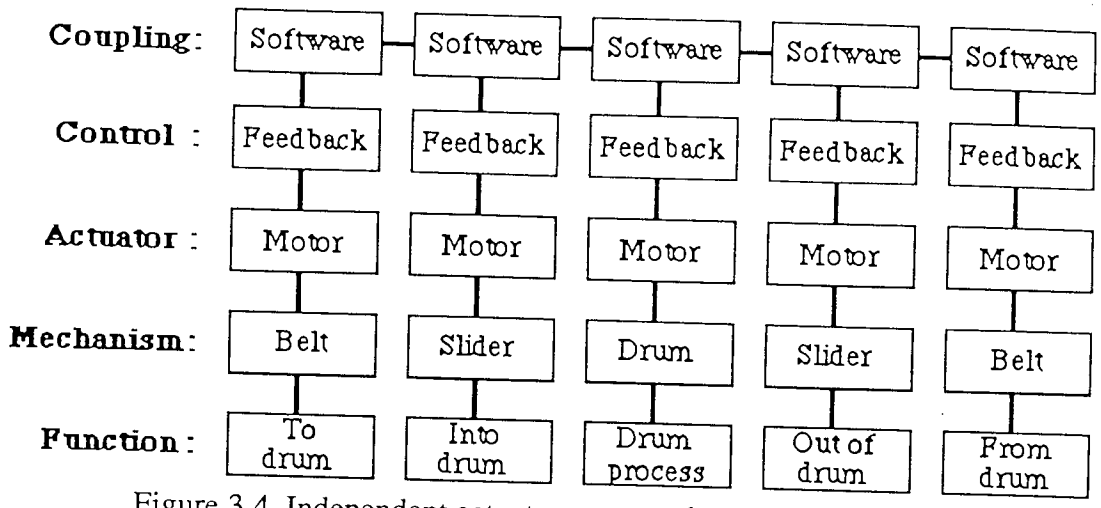


Figure 3.4. Independent actuators approach to machine design.

3.4. The requirements for implementing an independent actuators machine

For independent actuators' machine systems to be made implementable, it was necessary to investigate three main areas of interest:

{1} The development of software methods for coupling the motions of the mechanisms.

{2} The suitability of drives to directly supply the energy for the mechanisms' motions.

{3} The development of algorithms to enable the actuators to control the motions.

The development of software methods for recoupling the actuators is dicussed in chapter 4 of this thesis, including methods by which the coupling algorithms may be tested using continuous simulation models of actuators and mechanisms. The suitability of drives for actuating independent mechanisms is discussed in chapter 5, with procedures which have been developed for selecting the optimum drives for particular applications being included in Appendix 1. Chapter 5 also includes a discussion about the requirements for feedback controllers for independent machine drives. Chapter 6 details the independent actuators test rig which has been developed to test the methods for implementing multiple independent actuators' systems.

3.5 Modular machine systems.

The principal aims of introducing software coupled independent actuators into machines are improving flexibility and high speed performance. An additional aim is the reduction of machine development lead times. A significant aid to this can be found in the development of modular machine systems. A modular approach to machine construction can have several advantages:

{1} Reduced development overheads.: a new machine can be developed by adapting or directly incorporating previously designed and implemented machine modules into the new machine. This should reduce development lead time and costs.

{2} Improved implementation–specific development capability: machines can be more readily adapted to a specialised customer's requirement by concentrating development to those machine modules which do not fit the specialised application.

{3} Reduced production costs: the use of common modules in several types of machines will enable more parts to be mass produced, which will reduce production lead times and costs.

{4} Improved repair and maintenance capabilities: general repair and maintenance of machines can be carried out by concentrating on those modules which require work. More complex maintenance can be carried out off–line by replacing worn or damaged machine modules with good ones, thus reducing maintenance down-times.

{5} Improved machine performance upgrading capabilities: the operating performance of a machine will be limited by the capabilities of the machine module having the poorest performance. Thus uprating machine performance can be achieved by upgrading or replacing the poorest modules.

3.5.1. The components of a standard machine module.

For a modular implementation to be made viable, it was necessary to develop a standard configuration for a machine module. For maximum flexibility it was most important that machine modules were developed as much as possible to be self–contained units with common input/output stages. The input/output

compatability between modules would then enable interconnection topologies to be more readily constructed.

A standard machine module will contain elements from each of the five divisions listed in section 3.3. Thus each module will have a function to perform, such as moving product about the machine or transfering product into an arbor drum. Each function will require a mechanism and a form of actuation for that mechanism, which will be an electromechanical actuator such as a motor. This will require a servocontroller to control its motion and the means of coupling its motions with those of other modules by software. Examples of three standard actuator modules are shown in figure 3.5.

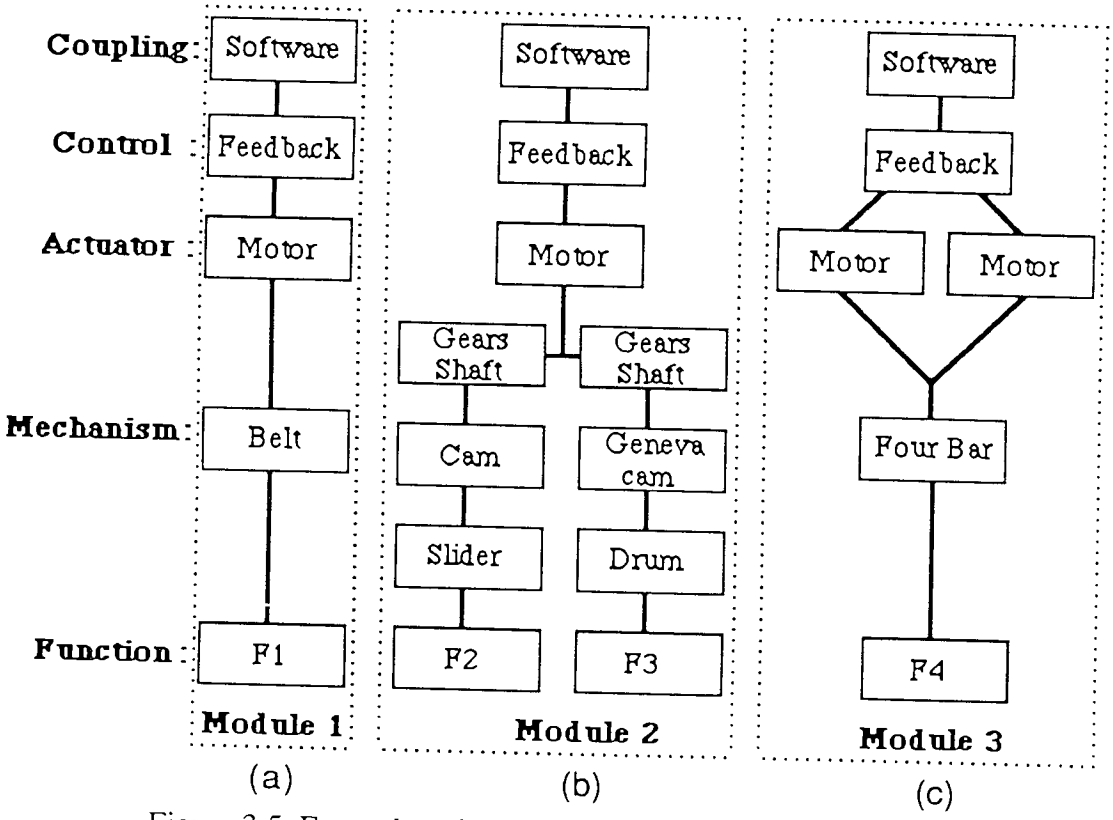


Figure 3.5. Examples of standard machine modules.
(a) Linear belt module
(b) Transfer slider and arbor drum combined module
(c) non-linear four-bar mechanism module

The linear belt module consists of a belt mechanism actuated by an electric motor under software control, with software algorithms for coupling its motions with those of other modules. Thus it contains one element from each of the five operational divisions, which is the minimum requirement for a machine module. A more complex module may be formed by using one actuator to drive two or more functions, as is demonstrated by the transfer slider and arbor drum module

in figure 3.5. Here the slider and the drum both employ cam mechanisms for controlling their motions, with the two mechanisms being hardware coupled to a common drive motor. Thus the two cam mechanisms and their couplings act as a single mechanism under the actuation of the drive module. For the combined mechanism to be software coupled to the host machine, feedback transducers must be employed, with the coupling software forming the upermost level of the two function module.

Modules incorporating cam mechanisms may be employed for driving functions whose motions are too arduous for direct driving by independent actuators. This also enables an independent actuators' implementation which incorporates mechanisms from traditional machine designs, which would be most useful during the change-over period from the traditional to the new machine design methods.

A modular implementation would enable functions having complex non-linear requirements to be included within a machine design. Figure 3.5.(c) shows a non-linear function having a four-bar mechanism with two actuator motors. Feedback control of the two motors ensures synchronisation with other machine motions, while software implementation of the control algorithms would enable this form of module to perform a wide range of machine functions.

The development of software controlled four-bar mechanisms incorporating twin Bru-500 brushless d.c. motors is currently being undertaken by the Mechanisms and Machines group at Liverpool Polytechnic as part of the S.E.R.C. Specially Promoted Programme of Research into High Speed Machinery.

3.5.2 Machine module interconnection

The interconnection of machine modules will take place at two levels. Firstly the electromechanical hardware will be mounted and aligned such that the combined set of modules forms the finished machine. Secondly the software coupling the modules' actions will require communication channels to allow the modules' actions to be coordinated. The form that the two levels of interconnection take will be application dependent, being governed by the physical construction of the modules and machine, the type of software used for controlling and coupling the modules and the processor hardware in which the software runs.The possible interconnection of modules for the five functions introduced in section 3.3 is represented in block diagram form in figure 3.6. In the

implementation scheme shown three modules are used to perform the five functions, with one module being used for each intermittent belt function and one module being used to drive the two transfer sliders and the arbor drum.

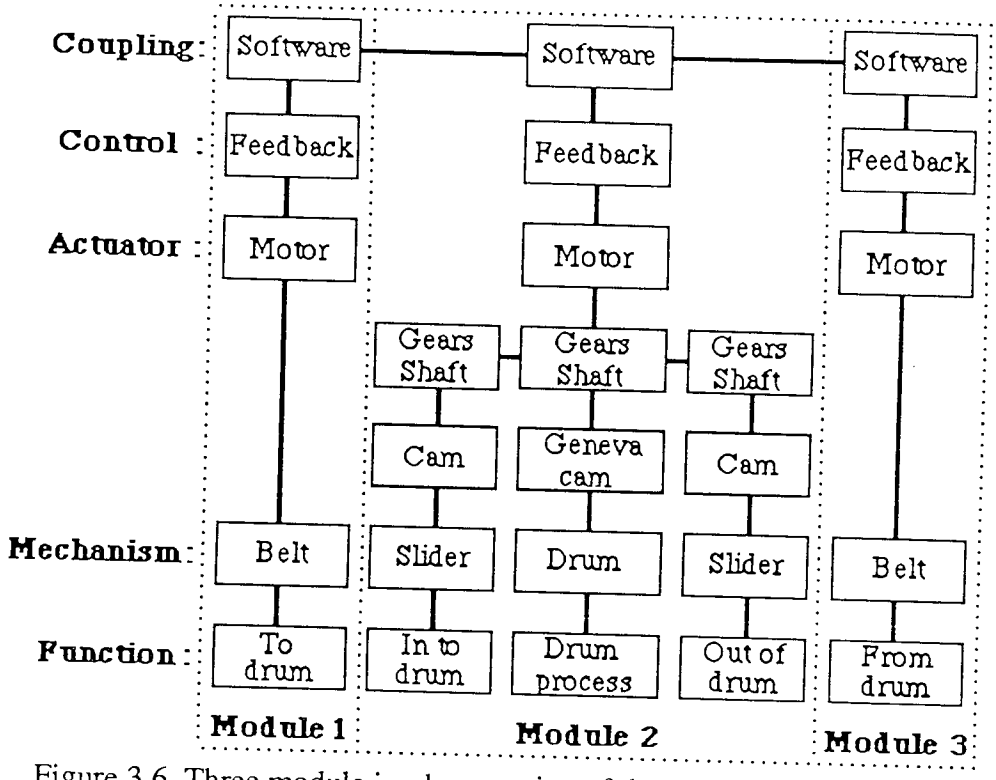


Figure 3.6. Three module implementation of the five function system.

The dotted line in figure 3.6 show the divisions between the actuators. The actuator hardware and software feedback algorithms would be developed to be application independent, while the coupling algorithms would be written to be application dependent. In this way, modules may be developed off-line and be adapted to suit a particular machine requirement. Although the implementation shown has three modules, the five functions may be grouped into up to five modules for the final implementation. Furthermore, it would be possible to later replace a multi-function module with several single function modules, or vice-versa, if performance were required to be improved on-line.

Chapter 4

The Development of the independent actuator controller scheme.

4.1.1. Introduction

This chapter describes the development of the software for coordinating and synchronising the independent machine motions. The software was developed to have a structure which allowed the coordination of the two motions of the demonstrator project while being suitable for expansion to the control of multiple actuator machines and multiple machine lines. To achieve this, extensive use was made of computer simulation to test the ability to synchronise dynamic models of electromechanical actuators. The dynamic models used were obtained in the laboratory tests described in chapter 6, which were simulated with feedback control loops using the algorithms developed in Appendix 3.

The author's research programme has been run in conjunction with software engineering research work at Aston University which is developing distributed control systems for independent actuators machines. The author's work has investigated the software coupling algorithms themselves and the development of a distributed controller structure which supports both flexibility and modularity in the machine.

4.1.2. Understanding the machine control problem.

The introduction of independent drives into a machine greatly expands the number of tasks that the machine's controller has to perform. In order to understand the level of expansion, it is useful to firstly consider the control requirements of a centrally driven, cam-actuated machine.Materials enter the machine through one or more buffered inputs and are moved through the machine in a series of intermittent motions. During the moves, the materials are combined and processed to form the finished product, which leaves the machine through a buffered output.

The dynamics of the machine motions are governed by the cam-actuated, centrally coupled mechanical drive. Since the time constant of the drive is many times the actuation period of the individual motions, no servo control is possible and the machine's dynamic control function merely sets the velocity of the single drive motor to the required process speed of the production plant. Thus in the overall control scheme, the machine may be treated as a single constant speed actuator, with buffered material inputs and a buffered product output.

During its passage through the machine, the material is monitored at discrete points to maintain process control and to enable process fault analysis. This is carried out by sampling the signals from transducers mounted through the machine and checking their states against known required values, with the state analysis being carried out by the central processing unit. The transducer sampling is not central to the machine's operational control functions, but is a secondary function which maintains product quality. Many of the sampling points are installed on-line to help rectify problems which arise during manufacturing runs.

In addition to setting the motor speed and monitoring the product flow, the controller provides the interface by which the operator may communicate with the machine. This includes programming the speed and duration of machine runs, monitoring the quality of the machine process and checking such parameters as seam sealer temperatures, amounts of materials remaining, etc. The controller must also contain service routines to handle events such as emergency stop, which interrupt the normal operation of the machine.

The control functions of the centrally driven machine may then be summarised as :

{i} Setting the velocity of the drive motor to the process speed of the production plant.

{ii} Monitoring the product at discrete points in the machine to provide product control

{iii} Providing a man-machine interface for on-line monitoring of the machine functions.

{iv} Handling service routines for unexpected events, such as emergency stop.

In addition to these prime functions, many machines incorporate task analysis functions, including intelligent analysis of process failures to establish trends and predinct machine faults. However, these functions are carried out by add on units and are not central to the normal operation of the machine. They do not fall within the scope of this analysis.

The same product processing functions and flow of material could be provided in a mechanically decoupled system, with the separate functions being driven by independent actuators under software control. The use of independent actuators greatly expands the level of control required. Whereas in centrally driven machinery the dynamic control of all functions are combined in a single open–loop controlled actuator, each drive of an independent actuators machine must operate under closed–loop control if the dynamics of its function are to be performed properly.

Furthermore, there are no buffers separating the actuator 'sub-machines' as there are in a production line of centrally coupled machines, so the problem of synchronising and coordinating the interactions of the actuators has been introduced. This requires that the motion of each actuator be monitored continuously to detect failures that may lead to destructive hardware collisions. In addition, the design aim of independent drives of introducing flexibility, means that each of the actuator functions must be readily reprogrammable, through an interface that will as a consequence be more complex than that of the centrally coupled machine.

The control functions for each actuator may be summarised as :

{i} Achieving tight control of the dynamics of the actuator.
{ii} Monitoring the actuator continuously to ensure no failures occur which could produce a hardware clash.
{iii} Synchronising the motion of the actuator with those of all interacting actuators.
{iv} Providing an interface between the actuator and the operator to provide monitoring of the actuator's motion and to enable on-line reprogramming of the actuator functions.
{v} Providing service routines for response to unexpected events, such as emergency stop.

Each of these functions must be provided for each actuator in the machine. In addition, there are machine-level control functions, which may be summarised as :

{vi} Setting the speed of the machine process to that specified by the operator.
{vii} Coordinating the flow of products through the machine.
{viii} Monitoring the product at discrete points in the machine to aid product control.
{ix} Enabling the man-machine interface to govern and reprogram the process at the machine level as well as at the actuator level.
{x} Providing service routines for coordinating the response to unexpected events, such as emergency stop.

It should be noted that in a well structured controller scheme, some of the actuator and machine functions would overlap and as such may not need to be implemented at both levels. However, comparison of these requirements with those of the centrally driven machine shows the level of expansion which has occured in the

complexity of the control problem. Clearly a scheme is required which will enable the reliable control of the independent drives machine, while allowing the machine functions to be reprogrammable and the machine cofiguration to be expandable and contractable. The software should thus enable the required machine flexibility and modularity which is central to the independent drives philosophy.

4.1.3. Distributed Control Systems for Independent Drives.

A distributed control system (36) is one in which multiple processors cooperate to achieve an overall goal - the control of a physical process or plant. The processors may be distributed throughout the plant, with communications enabling the computation hardware to interact.

An important aspect of distributed control is that of structuring the processing configuration to meet the needs of the controlled system. It is the structure which provides the single system goal to all the independent components. It is this feature of goal achievement through structure which most emphasises the suitability of distributed control systems for independent drives machinery. In both the controller and the machine, each independent component is working autonomously, but the whole array of components are combining to produce a finished item. Although each component may interact or communicate with only a few neighbouring components, the unifying structure enables the flow of interactions to fulfill the system goal. Furthermore, the localised nature of communications in distributed control enhances both flexibility and modularity in the system, since hardware or software alterations affect only the processor being altered and the few components it interacts with directly. Thus a distributed control scheme may be structured to suit the requirements of the independent drives machine, with the localised communications enabling the structure to be readily expanded or contracted to fit changes in the machine hardware.

The advantages of distributed control for independent drives machinery may be summarised as:

A: Flexibility.

The use of specialised processor units for controlling machine functions mean that alterations to a function will only affect the processors which are local to that function. Thus reprogramming a function's control will not require altering the control of the whole machine. Flexibility is also seen in the ease with which localised communications enable a machine's configuration to be altered.

B: Modularity.

The use of independent processors, which are matched to their independent hardware functions enhances the ability to replace system hardware with improved units to alter or upgrade the performance of the machine. In addition, particularly succesful machine components, such as a sensor or an actuator, may be incorporated into a new machine design along with the processing and software modules that are used to control it. A further advantage of modularity is the ability to introduce new control or analysis functions into the machine. These may be added by the incremental addition of the processors on which they are run, without the risk of having to upgrade the existing processor system.

C: Performance

Since distributed control system performance is readily upgradeable by the incremental addition of processors, there is virtually no limit to the number of tasks the properly designed controller can perform. Specialised processors which handle a particular task will be capable of achieving the required sample speeds without the danger of their actions being slowed by expanding the number of tasks that the control system must perform.

D: Development

The use of specialised processors for controlling the independent machine functions make it easier to develop the functions off-line, away from the machine. In this way both the hardware and the software may be developed, either in isolation or in groups of functions before being incorporated into the machine system.

E: Maintenance

Maintainance is enhanced in distributed controllers by the ability to replace system modules when failures occur. In addition, as with machine function development, units may be tested for faults off-line, thereby reducing the risk of further machine failure resulting from the faulty component.

4.2.1. Developing a distributed controller structure for the independent actuators machine problem.

In developing a structure for the distributed machine controller it is necessary to determine the optimum arrangement for the communications flow through the processor array.A great number of processor interconnection and communication schemes have been proposed, with some of the relevant literature covering processor interconnections being discussed in chapter 2. To decide upon the optimum scheme for the machine controller it is necessary to consider the functional

requirements listed in section 4.1.2, so that these may be incorporated into the control structure.

Figure 4.1 shows the hierarchy of machine control functions for one actuator, with the emergency stop and failure routines forming the highest level and the actuator feedback controller occupying the lowest level.

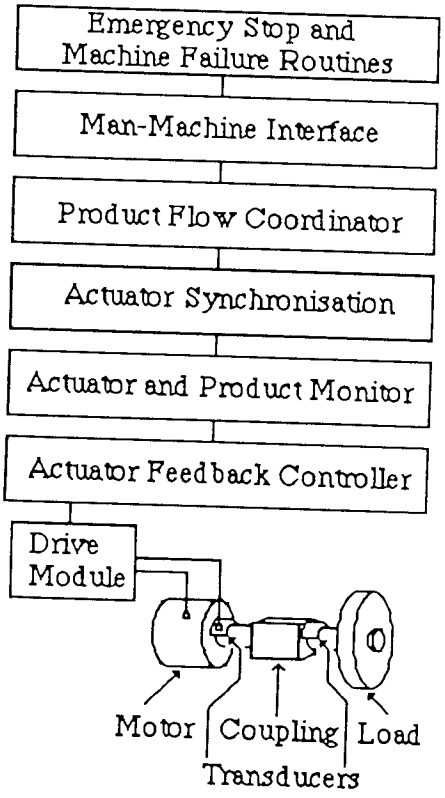


Figure 4.1. Hierarchy of control functions.

The emergency stop and machine failure functions must have the capability to overide all other control actions. These interpret signals from lower levels to trigger their actions. Below these, at the highest level of normal operational control, is the man-machine interface. This receives information from the outside world which governs the operation of the machine. The product flow coordinator uses this information to control the flow of product through the machine. This function instructs each actuator on when to move. The actuator synchronisation function then ensures that movement is safe using information from its actuator and product process monitor and from other actuators. The actuator feedback controller controls the actuator's motion, with the monitor relaying motion information to other control functions.

The fact that there is a clear hierarchy in the control functions points towards the adoption of a hierarchical control scheme for the distributed controller structure. A conventional three level hierarchical scheme (39) is shown in Figure 4.2.

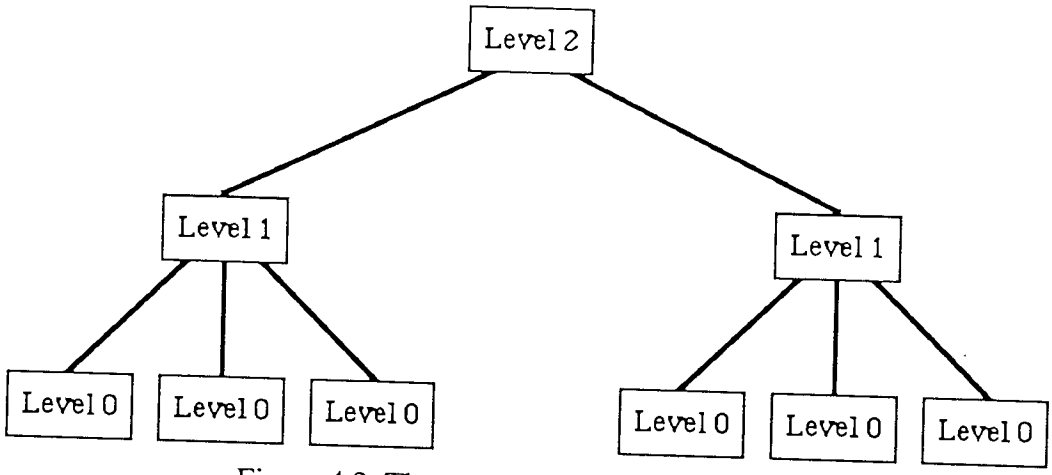


Figure 4.2. Three level hierarchical interconnection scheme.

Generally for a hierarchical scheme level 0 processors are highly specialised, performing well defined control or data logging functions and relaying information to level 1. The capabilities of a level 1 processor will generally be much greater than that of level 0. The duties of this level of processor can be very varied, including controlling level 0 operations, receiving and transmitting level 0 information and performing on-line analysis of level 0 data. The level 2 processor is usually the most powerful unit. Level 2 will provide overall coordination and control of all system tasks and will allocate duties to lower level processors.

Figure 4.3 shows how the conventional hierarchical control scheme could be applied to the independent drives machine control problem. This enables separation of the control functions into levels 0 and 1, actuator level functions and levels 2, 3 and 4, machine level functions. This serves to give the structure.actuator modularity. Each actuator controls and monitors its own motion and the motion of the product as it passes through it. The level 1 control of actuator synchronisation governs the actuator's interactions under the coordination of the level 2 product flow function. Above the flow coordinator are the man-machine interface and, at the highest level, the emergency stop and machine failure functions.

The hierarchical scheme will introduce the required machine modularity, but using this configuration level 2 could be a limiting factor in the capabilities of the machine. Level 2 has to coordinate the concurrent interactions of all machine actuators, so it is clear that the transfer of significant amounts of data will be required between levels 1 and 2. At the same time as handling data transfers, level 2 will have to interpret

the data and make coordination decisions accordingly. It will also have to transmit process information to higher levels. If a large system is to be constructed, with many independent functions requiring coordination, the response time of level 2 will be relatively long. This will inevitably limit the maximum size of machine which can be controlled by this control structure.

A further restriction introduced by level 2 is flexibility. Since level 2 coordinates all machine functions, implementing software changes to a few of the functions will require rewriting the software for the whole machine's coordination. A more flexible approach would be to enable each actuator to coordinate it's own actions with those of other actuators. This enables changes to a function to be implemented independently from other functions. This increases both the flexibility and the modularity of the controller. A method of implementing this is described below.

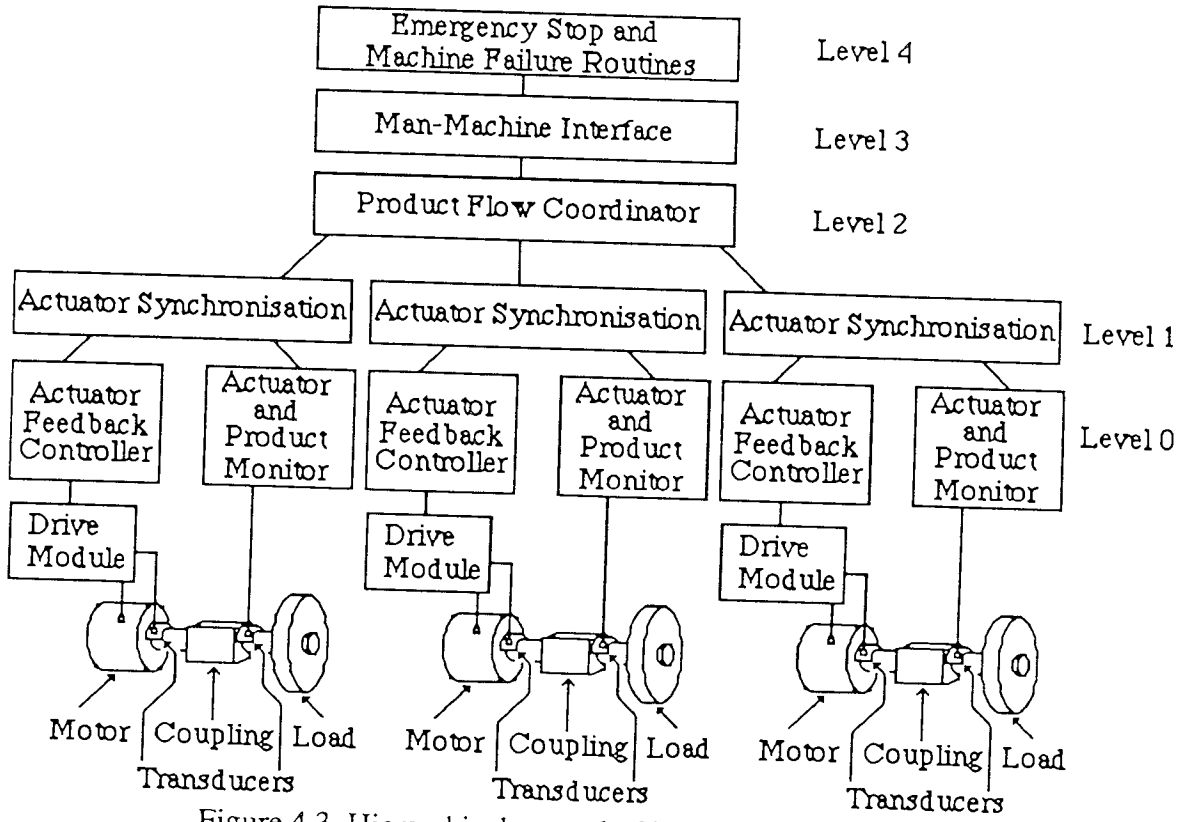


Figure 4.3. Hierarchical control of independent actuators.

4.2.2. Decentralising the controller structure.

The upwardly converging nature of the hierarchical structure tends to centralise the control functions. As described in section 4.2.1., this centralisation can limit the machine performace due to the large amount of processing that can be required at the higher levels. To overcome this problem it is useful to compare the hierarchical structure of Figure 4.3 with the operation of a centrally driven, cam-actuated

machine. The level 2 processor is in effect the 'central prime mover' of the controller, with level 1 forming the hardware couplings and level 0 the cam actuators. From this analogy it is apparent that if the controller structure is to have the same degree of flexibility as the independent drives hardware, level 2 must be 'de-coupled' and its action must be distributed amongst other controller functions, in the same way that independent drives distribute the energy scource about the machine.

The flow of materials through the machine may be compared with the flow of liquid through a pipeline. Actuators process the product as they move it through the pipe, while more complex functions serve to combine materials at pipeline 'junctions'. If this pipeline structure is introduced into the controller network, the controller may be given a decentralised structure which matches the flow of materials through the machine. In this way, the product flow coordination can be handled at the actuator level, with local groups of actuators combining through communications to coordinate the flow. Figure 4.4 shows how the pipeline philosophy may be incorporated into the hierarchical control structure.

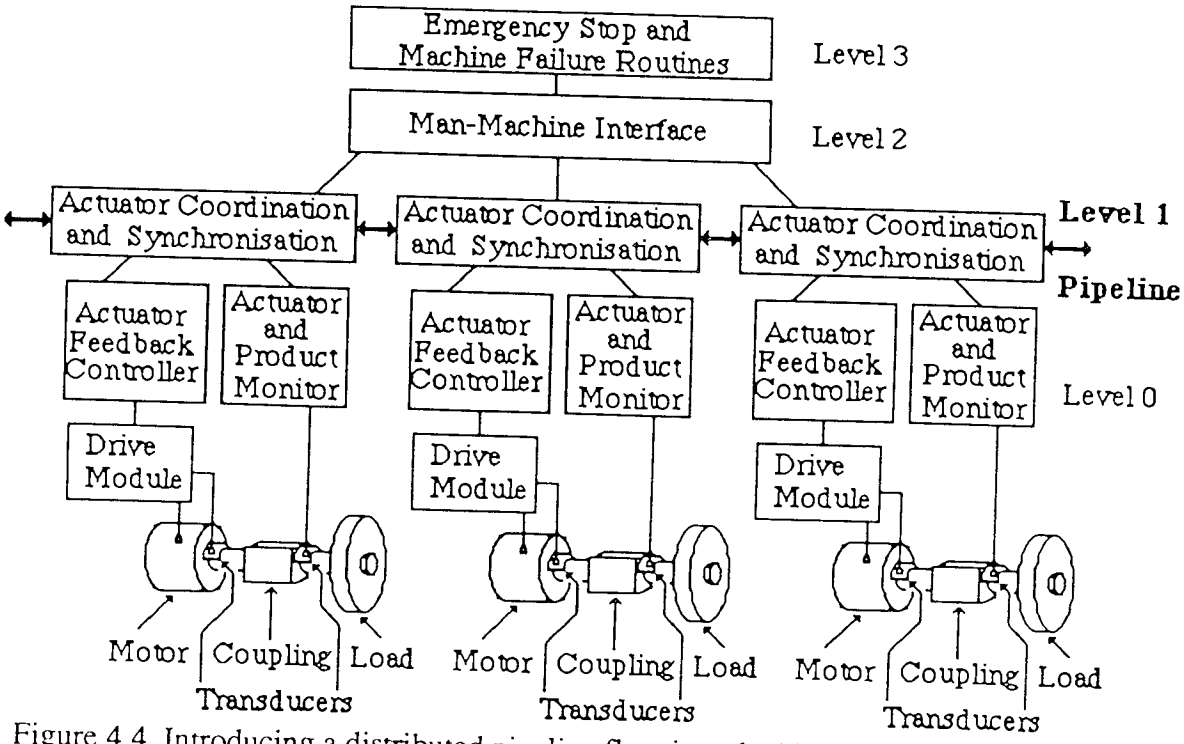


Figure 4.4. Introducing a distributed pipeline flow into the hierarchical structure.

The product coordination and actuator synchronisation functions are combined at level 1, with inter-actuator communications taking place at this level, without utilising the centralised intermediary processor as is the case in conventional hierarchical control. The level 1 processors therefore form the control pipeline, through which communications relay state of the actuators and the flow of products

through the machine. Level 1 processors then decide upon their own actuators' actions using the communicated information and send control communications to the level 1 functions.

In the hierarchical/pipeline structure of Figure 4.4, all normal operational machine control functions take place at the actuator level. During production runs, the machine level functions carry out monitoring and analysis processing, with intervention only occuring when an unexpected event occurs, such as an actuator or product failure, or an operator command interrupting normal operation. Since the actuator levels are so important to the machine control and since they are of modular construction, it is clear that all actuator control modules must be compatable, while the module interactions must be well specified to enable the reliable flow of products through the machine. The structure of the actuator control modules will now be discussed and the specification of their communications will be developed.

4.2.3. The independent actuator control module structure.

The tasks to be carried out by the controller for each independent axis may be broadly separated into three sections. Firstly the dynamic performance of the actuator must be tightly controlled using feedback control. Secondly the controlled motion must be monitored to ensure product control and failure detection. Thirdly the motion of each actuator must be closely synchronised with the motions of its interacting actuators, while all the motions must be coordinated to ensure a flow of product through the machine. These three functions will be considered under the headings:

(i) Actator feedback control (AFC)
(ii) Actuator process monitoring (APM)
(iii) Actuator coordination and synchronisation (ACS)

The operational tasks of each of the three functions will now be discussed, then the interconnection of the functions to form an actuator module will be developed.

4.2.3.1. Actuator feedback control

AFC is the control function which most directly affects the quality of the finished machine product. This takes the form of a classical feedback controller which controls the dynamics of the machine parts which come into contact with the product during the manufacturing process. Stiff servo control is therefore very important. The AFC characteristics will be designed to obtain the optimum performance from the actuator being driven. Therefore each actuator will have its

own AFC algorithm which will be adapted to suit the application for which the actuator is being used. The overall software structure must therefore not be AFC algorithm dependent, but must allow new developments in, for example, intelligent, predictive or adaptive controllers to be incorporated to improve machine performance.

An exercise was carried out to develop the optimum feedback control algorithm for controlling the BRU-500 motors that were selected for use in the demonstrator project. The reader should refer to Appendix 3 for details of this. The algorithms developed in Appendix 3 were used during the development of the control structure described in this chapter. The implementation of these in the ACSL simulation models is described in section 4.4.

The AFC contains the information about the motion profile that the actuator is required to follow. When it receives the command to move, it will follow the profile within its memory until either it has completed the motion or until it receives an emergency stop command. In the latter case the AFC function will contain the emergency stop procedures to be followed. These will depend on the actuator being driven and the method of implementation of rapid braking for the actuator. Upon completion of the motion the AFC function maintains the actuator in its holding state until it receives the next command to move.

4.2.3.2. Actuator process monitoring–detecting processing failures.

The monitoring of the operation of the machine functions is an essential task for the machine controller to perform. Controlling and synchronising the motions of perfect machine elements is a relatively straight forward task, but real systems do not always provide perfect operational performance. It is the potential for non–perfect operation of the machine elements which adds to the complexity of the control problem. The APM function is required to monitor the performance of the actuator and the state of the product at all times during the machine's operation. It must determine whether the function is being properly performed and must convey this information to the ACS control function. To enable this to be carried out, it is necessary to consider the potential failure modes of independent machine functions and how such an occurence may be quickly identified by the APM.

One of the biggest difficulties in failure detection analysis is that it must be assumed that it is not possible to predict every type of failure and resulting performance change that could occur. If this is not assumed, then the occurence of any failure mode which is not predicted could result in extensive damage occuring in the

machine. Even if every possible type of failure were to be predicted, the development of a controller capable of handling service routines for each possible failure would be extremely complex and would result in an inflexible implementation. Therefore, it is necessary to consider possible failure modes together and to develop the APM function to be able to detect, though not necessarily identify, failures. This can then give a similar general response to similar failure modes.

Failures in machine functions can take place at several levels of operation:

(i) Power failure
(ii) Coupling or controller failure
(iii) Actuator failure
(iv) Mechanism failure

Power failure can take the form of either a total loss or a fluctuation. Several existing machines incorporate solenoid operated mechanical brakes which are released when power is applied to the machine. A loss of power causes the brakes to lock and stop the machine. It is proposed that these be incorporated in independent actuators to stop their motions in the event of a total loss of power and to serve as emergency stop brakes. A power fluctuation could cause the performance of the actuator to be degraded, with a fluctuation occuring in the dynamics of the function causing potential product damage or a collision. The APM must therefore be capable of detecting such a fluctuation.

A failure in the software coupling or controller could cause machine performance to be degraded or, worse, could cause collisions to occur. This could be caused by either a fault in the algorithms themselves, or in the hardware of the controller. To overcome the potential of a control failure, research is being performed in the Department of Electrical Engineering at Aston University into the design of reliable distributed controllers for independent actuators machinery. This includes the formal specification and verification of software, the design of fault tolerant software which is capable of properly responding to controller failures and the design of fault tolerant hardware to support the control functions, including hardware redundancy and reliable communication chanelling.

An actuator failure can take many forms. During the laboratory testing work, several drive module failures occured which caused the motors to freewheel. Another motor suffered from bearing seizure, which caused the motor to stop. Several other small failures caused the drives' dynamic characteristics to change suddenly, resulting in performance fluctuations. In each case, if the drives had been

operating machine functions, product damage or collisions could have occured. Therefore it is important that the APM be capable of detecting such failures so that resulting damage can be minimised.

As with actuator failures, mechanism failures can also take several forms. These could cause either a sudden change in characteristics, as would result from a brakeage or seizure, to a gradual change in characteristics, as could occur due to general wear. A sudden change could result in knock–on failures in the actuator or in other actuators, while a gradual change could cause the product quality to be degraded. Thus the APM must be capable of detecting changes in the mechanism's characteristics so that the correct machine response can be given.

The requirements for the APM are therefore that it be capable of detecting power fluctuation effects, actuator failures and mechanism failures. Since it is a fundamental assumption that each particular failure mode can not be predicted, it was necessary to determine a generalised method of detecting and responding to failures which included all potential power, actuator or mechanism failures. Therefore a common effect of the three sources of failure was sought. It was a further requirement that the method of detection be implementable at low cost and with as little complexity as possible. A method which satisfied all these requirements was found in the sampling of the dynamic performance of the actuator.

The APM monitors signals from actuator mounted transducers and checks their states against known performance parameters. Each transducer poll is registered as an event. The APM must be able to register three types of event :

1.Normal operation: All events occur within specified tolerances. The process monitor allows the process to continue normally.

2 Damaged Product.: An event occurs whose parameters are such that they indicate that the product will have been damaged, but not in a way that will endanger the operation of the machine. Examples might be too large an actuator acceleration resulting in compression of the product, or a transfer mechanism contacting the product at too high a speed, thus causing crumpling. For this type of occurence, the process monitor passes information to other functions that the product is damaged.

3. Emergency Stop: An event occurs which indicates that damage may occur to the machine if the process continues. Examples may be break down of the product through too large contact velocities or accelerations, causing blockages, or

mechanical failure of a machine part, or a power failure. These events require the actuator motions to be stopped immediately, by mechanical braking if available. The process monitor triggers the emergency stop function to stop all motions immediately.

A failure of power, actuator or mechanism will cause a fluctuation in the dynamics of the actuator. The APM monitors dynamic transducers and checks the actuator's state against that of the required motion profile, which is known. The determination of event types is achieved by introducing tolerance bands into the required motion profile. Figure 4.6 shows the profiles for a typical incremental move, plotted against a common time axis. Each profile is shown with two tolerance bands, which for clarity are not shown to scale. The inner band is the specified tolerance band for the production of an acceptable product., while the outer band is the specified tolerance band for minor product damage to be expected.

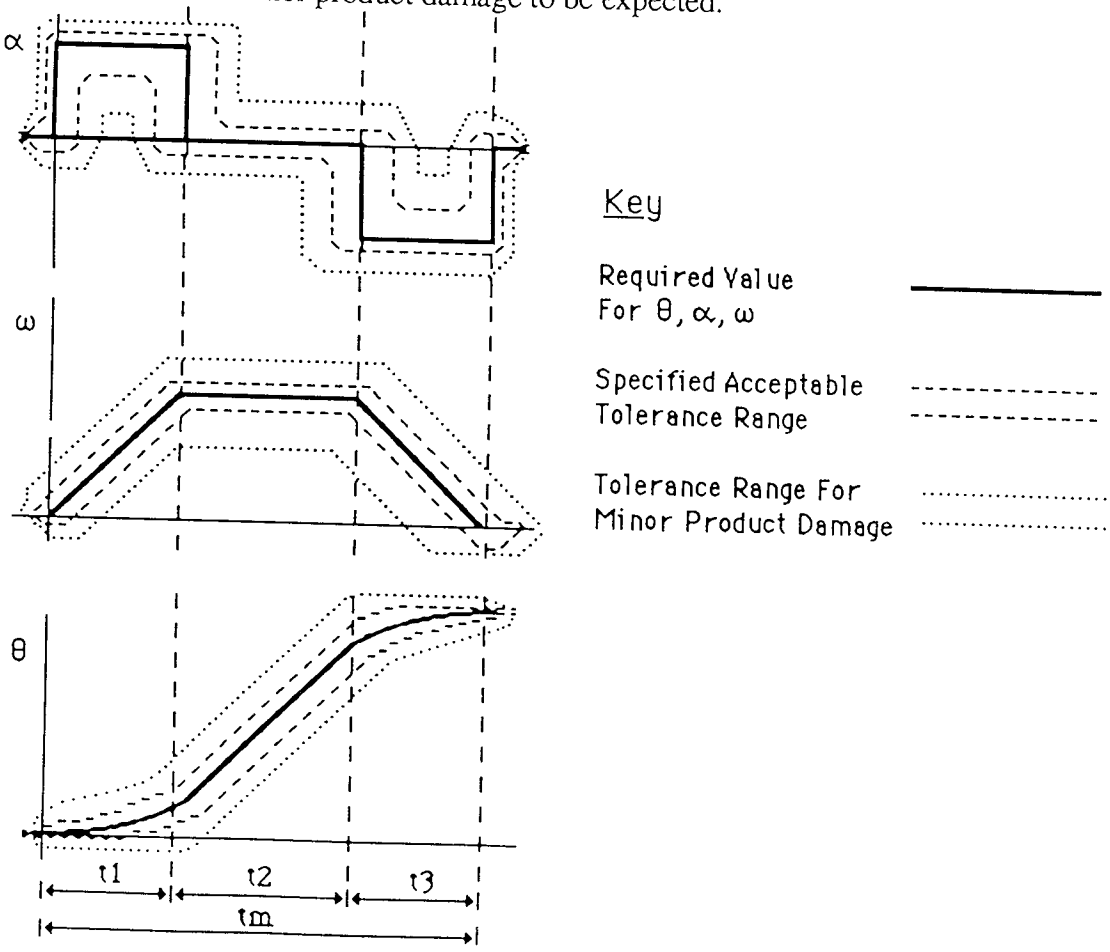


Figure 4.5. Trapezoidal increment showing profile event tolerance bands.

When an event is sampled which falls inside the inner region, the process monitor registers a type 1, normal operation event. If an event occurs in which the actuator moves outside of the normal operation band but inside the outer band, the actuator's motion is such that it may cause damage to the product. This is then registered as a

type 2, damaged product event. For example, if the third transfer slider of the demonstrator project contacts the product at greater than 1.0m/s, the packet will crumple and the product will be damaged, but the machine will still be able to continue operating if the contact speed is less than 1.2 m/s. Thus if the sampled event corresponding to the packet contact has a velocity in the 1.0m/s to 1.2m/s band, it will register as a type 2 event.

If an event is sampled whose position, velocity and acceleration parameters are such that it falls outside of either of the tolerance bands, then either the product will be damaged sufficiently to cause a blockage or moving parts will be in danger of contacting each other. Therefore the event will register as a type 3, emergency stop, event and all motions will be stopped immediately.

The tolerance bands may be specified as having either spatial or temporal dependance. For example, the spatial definition of the event that represents the third transfer actuator contacting the product would be that at 53mm $^{+}_{-}$ 1.0mm into the stroke the actuator must have constant velocity in the range 0.8 m/sec to 1.0 m/sec. The same event could be given a temporal definition by adding the requirements that it occurs 19 msec into the motion. The choice of spatial or temporal tolerance definition will depend on the function of the machine or actuator being controlled.

The profile monitoring scheme described here allows event interpretation to be performed using position, velocity or acceleration transducers. The same method may be extended to product state detection devices, such as sealer temperature or material weight transducers. The exact configuration of transducers will be dependent on the function of the actuator, but interpreting the signals as event types will introduce compatability between actuator sensor types.

During the periods between motions, when in intermittent systems the actuators are required to remain stationary, the APM must ensure that the actuator does not move outside of its holding state. It must therefore continue to relay actuator condition information to the ACS. This also serves to inform the ACS that the actuator is ready to commence the next move.

4.2.3.3. Actuator coordination and synchronisation.

The ACS function must make the decision to move the actuator. To do this it must determine :

(i) Is movement required ?

(ii) Is movement safe ?

The performance of the ACS is crucial to the operation of the machine. The ACS is the principal element of machine modularity, it being the function which coordinated the actuators functions within the machine. It communicates with its own APM and AFC, with other actuators and with the machine level functions. The nature of communications to enable safe synchronisation and the reliable flow of products will be discussed in detail in section 4.3.

4.2.3.4. Configuring the communications within the actuator control module.

The three functions making up the actuator control module each have specialised duties which they carry out autonomously, with inter function communications serving to coordinate their activicties within the module. The interconnection topology of the module will be dependent on the communications requirement of each actuator. Table 4.1 summarises the duties of the three functions and lists the communications input each one requires to carry out its duties during normal machine operation.

Function	Duties	Communications Input Required
AFC	Controls actuator motion along self-contained profile using feedback control	(i) Commence motion command (ii) Transducer feedback signal
APM	Monitors performace of actuator and product. Checks this against pre-programmed profile tolerances. Deduces actuator state	(i) Commence motion command (ii) Transducer signals
ACS	Synchronises actuators interactions. Determines when actuator should move. Coordinates product flow through actuator	(i) State and content of actuator (ii) State and content of local actuators

Table 4.1. Actuator function operational duties and communications requirements.

The duties and communications requirements are the minimum requirements for normal operation. It should be noted that all functions need to be capable of communicating with the man-machine interface and the emergency stop and machine failure functions. However, inter-actuator communications need only take place at the ACS level, which reduces the number of communications channels the machine requires. Based on the requirements in Table 4.1, the actuator module structure of Figure 4.6 was developed.

The module structure maintains the hierarchical configuration and pipeline capabilities, but has a well defined communications topology. The linking of chains of these modules to form the pipeline controller requires the actuator synchronisation and product coordination schemes to be well defined. These will

now be described and the use of simulation models to develop methods of testing the machine controller structure will be explained.

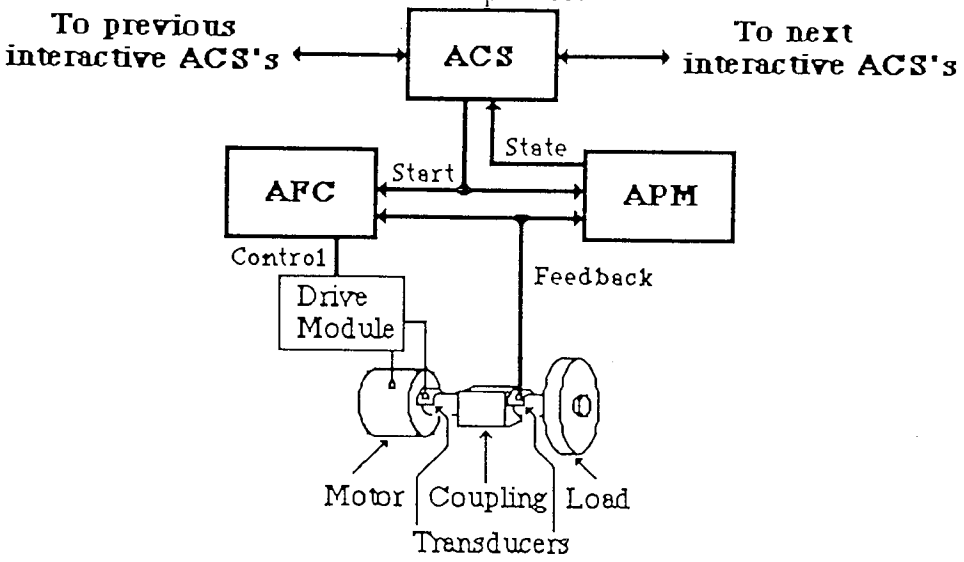


Figure 4.6. Single axis control module schematic showing inter-function communications requirements.

4.3. The development of the independent actuators interaction scheme.

The hierarchical/pipeline controller topology enables a modular controller which matches the structure of the machine system. In section 4.2.2 it was explained how the controller was analogous to a centrally driven cam actuated machine. The level 0 AFC and APM functions represent the cam actuators, the synchronisation functions the mechanical couplings and the coordination functions represent the central prime mover. The control functions were decentralised by introducing the ACS pipeline, while the actuator interconnection scheme described in section 4.2.3.4 enabled each actuator controller to be considered as an independent module. In this section, the ACS pipeline will be considered and the control logic developed to enable the pipeline to perform the function of recoupling the actuators' interactions.

The decentralisation of the coordination of the actuators means that each actuator module has autonomous control over its own motions. An ACS decides when its actuator should move based on the information it has available from communication inputs. These come from its own level 0 functions and from the ACS's of all interactive actuators. Since the communications handling serves to recouple the actuator motions, the specification and implementation of the communications are very important. This will be considered first because the communications form the basis for the development of the coordination and synchronisation scheme.

4.3.1 The inter-function communications scheme.

Each actuator control module is a self contained unit, with an AFC containing details of its own functional requirements, an APM having the ability to analyse its own performance and an ACS ensuring it moves when it is required to. This arrangement allows machine flexibility in the ease of altering actuator performance by reprogramming that actuator alone, without having to alter the modules of surrounding actuators. Thus each control module will contain detailed information about its own functional requirements, but not about the functional requirements of other actuators. The communications scheme should support this, with the information being transmitted between modules taking a form that does not require specialised analysis to interpret its meaning. This can be readily achieved using a logic state approach. Each module defines its own condition in terms of logic states and relays this via the ACS pipeline to other modules. Standardisation of the communications logic ensures compatability between modules and thus enhances modularity.

To define the communications logics, it is necessary firstly to define the actuator interactions information which needs to be transmitted. Motion decisions are made according to the state of variables which affect the need and the ability to make the motion. There are two variables that affect an actuators decision to move, these are the dynamics of the actuators it interacts with and the state of the product it is processing. It will only move if there is no danger of collision during the motion. It will only be required to move if there is product in the machine pipeline which requires it to move. Thus each actuator must be capable of defining its dynamic state and the state of the product it is processing in terms of two multi-state logics, which it can relay through the pipeline. All modules use the logic states of their local interactive actuators and of their own actuator to decide whether they are able to move and whether they are required to move. To do this it is necessary to define the two logic states that are to be communicated.

1.MOTION condition.

During normal machine operations, several actuators may follow motion profiles whose envelopes overlap spatially. These spatial overlap regions are of significance in the synchronisation of actuator motions. If two actuators follow profiles which put them in the same overlap region at the same time, they will collide. Thus when making a decision to move, an actuator needs to know if any interactive actuators occupy an overlap region, or if they are moving towards an overlap region with the intention of occupying it.

An actuator's intermittent motion profile may be defined as being in one of five states :

{i} READY :- the actuator is in the stationary dwell position, able to receive product or commence a move.

{ii} INTENTION :- the actuator is moving within tolerance. It is outside the region of spatial overlap, but it is following a profile that will pass through the overlap region.

{iii} OCCUPATION :- the actuator is moving within tolerance inside the region of spatial overlap.

{iv} CLEAR :- the actuator is moving within tolerance. It is outside the region of spatial overlap and will not return during that motion cycle.

{v} DANGER :- the actuator is moving outside of tolerance and may cause a collision. This is an emergency stop event.

The five motion states can be determined by the APM. In section 4.2.3.2 the APM's event detection function was introduced. The APM handles all transducer inputs from the product and the actuator. It also contains information relating to the actuator's required profile and hence its intention. The APM function can therefore use these features to interpret the actuator's MOTION state, which it relays to its ACS.

It should be noted that within an independent drives machine incorporating a pipeline flow, many actuators will have envelopes which encompass more than one overlap region. The APM therefore must be able to interpret one MOTION state for each overlap region it passes through, with the ACS communicating the relevant state to each of the interactive control modules.

2. PRODUCT condition.

The APM's event monitoring function is used to set the state of the product by means of registering event types, as described in section 4.2.3.2. The PRODUCT condition can have three states:

{i} GOOD :- the product is being processed within tolerances. All processing events have been type 1, normal operation.

{ii} DAMAGED :- the product has been damaged during processing, but is in a state which enables it to continue through the machine to a point where it may be discarded.

{iii} EMPTY :- the actuator has no product to process.

The two logic conditions enable the ACS functions to communicate via the pipeline details about actuators and products which may be readily interpreted and which entail small communcications overheads. For example, transmitting the MOTION condition is CLEAR is more readily interpretable and requires less communications than, for instance, stating position = 38mm, velocity = -3.1 m/s.

The two logic conditions and their states are summarised for reference in Table 4.2.

Condition	States
PRODUCT	GOOD DAMAGED EMPTY
MOTION	READY INTENTION OCCUPATION CLEAR DANGER

Table 4.2. Actuator MOTION and PRODUCT logic conditions.

4 3.2. Coordinating and synchronising actuator interactions by interpreting the multi state logic communications.

Having defined the actuator's communication requirements as multi state logic conditions, it is necessary to develop the product flow coordination scheme which utilises the MOTION and PRODUCT states to decide when the actuators should move. Each ACS decides when to commence motion based on the information it receives through the communications. Since there is no centralised product coordination, the product state information must move with products as they are moved through the pipeline. Therefore it is necessary to define the point at which an actuator takes control over a product from the actuator which precedes it in the pipeline, so that the PRODUCT condition may be transfered with the product.

An actuator may be considered to take control of a product when the preceding actuator ceases to have an influence on its condition. In many cases the exact point at which this occurs may not be readily definable, however the spatial overlap regions may be used to define the handover point. When the actuator passing the product leaves the spatial overlap region between it and the receiving actuator, with no intention of returning, control of the product has definately been transfered. Thus when the passing actuator's APM informs the ACS that its MOTION condition has changed from OCCUPATION to CLEAR, the ACS must pass the PRODUCT condition information to the receiving ACS, which then changes its

PRODUCT state from EMPTY to that of the accepted product. The passing actuators PRODUCT state then changes to EMPTY.

Generally, within intermittent motion machines, an actuator will not move unless it has product to process. Thus it will be input driven. However, if a multi product actuator such as the arbor drum was purely input driven, it would not clear itself at the end of a materials run. This then may be input/output driven, with motion depending on both its input stage and on the contents of all of its arbors. Since each actuator in a machine is independent, the decision mechanism defining the motion stimulus for each actuator must be independently specified. The flexible nature of independent actuators allows the decision priorities of each actuator to be matched to the requirements of the pipeline flow.

The coordination problem therefore involves defining the points at which an actuator accepts and relinquishes control of the product and defining the flow priorities of the actuator's section of the pipeline.This then decides when each actuator is required to move in order to maintain a reliable product flow. It does not however define when each actuator is safe to move. Since each actuator makes autonomous decisions about its own motion requirements, the decision to move must satisfy synchronisation logic which will not allow the actuator to move unless its envelope is clear. This can be achieved by interpreting the MOTION condition communications.

Actuators are in danger of colliding if they occupy a common spatial overlap region concurrently. The general principle of synchronisation must therefore be that no actuator may move unless all other actuators satisfy conditions which determine that they will not use the actuator's regions of spatial overlap while it passes through them. This is implementable using the MOTION conditions of all interactive actuators. The synchronisation mechanism for an actuator is that motion can only commence if the MOTION states of all overlapping actuators are not OCCUPATION or INTENTION. Once the commence motion command has been issued, the ACS relays the actuator's MOTION state as INTENTION, which prevents interactive actuators from moving.

4 4.1. Simulating independent interactive actuators using ACSL.

The ACSL simulation package (30) was developed for modelling and evaluating the performance of continuous sytems described by time dependent, non-linear differential equations. In chapter 6 it is described how ACSL has been used to develop reliable models of the brushless d.c. motors being used in the demonstrator

rig. In Appendix 3 ACSL is used to evaluate feedback control algorithms using these motor models. In each of those sections of work the ACSL models are for one axis only, with the simulations being carried out to optimise the performance of each actuator in turn.

The structure of an ACSL model allows the components of the simulated system to be separated into their constituent parts - real poles, complex poles, integrators, comparators, etc. The ACSL sorting routine combines the separate parts to form the system model. The flexibility that this allows is demonstrated in the modelling work of chapter 6. The separation into parts also allows several axes to be simulated in the same model, with each axis having its own required profile and so the complete set of axes following independent trajectories. Thus the operation of multiple independent continuous systems may be simulated concurrently, with their motions being recorded in a common temporal frame.

The ability to simulate several axes concurrently offers the potential to model the independent actuators' machine. To achieve this it was necessary to develop methods in which the logic of discrete controllers could trigger the actions of the continuous models asynchronously. A method of simulating the controller action is found in PROCEDURAL blocks. Code within PROCEDURAL blocks is not sorted by the ACSL translator and is executed in the sequence written. Thus standard FORTRAN code may be inserted into the ACSL model to simulate the action of the controller. If a PROCEDURAL block is inserted into the DERIVATIVE section of the model, it is evaluated at every integration step. If the PROCEDURAL block is in the DISCRETE section, it is evaluated at intervals given by the sample period of the model. In either case, the simulated evaluation period of the code does not vary with the amount of code in the block, the code sequence is considered to execute instantaneously.

In a distributed control system for the independent actuators, each actuator will have its own control processors, so the execution period of the controller will not change appreciably with the number of axes being controlled. The distributed controller action may therfore be designed with a sample speed that is not dependent on the size of the overall control task, but instead is dependent on the processing period of the slowest function. This effect of the sample speed being independent of machine size is similar to the evaluation action of the ACSL PROCEDURAL block code, which facilitates simulation of the expandable and contractable parallel controller action sequentially in the PROCEDURAL block.

Figure 4.7 shows how the AFC, APM and ACS controller functions for two actuators may included with the continuous system elements in an ACSL simulation model.

```
PROGRAM TWO AXIS SIMULATION MODEL STRUCTURE
  INITIAL
    Initialisation Routines
  END

  DERIVATIVE
    Continuous Model 1 Definitions
                                        } Code Sorted
    Continuous Model 2 Definitions
  END

  DISCRETE
  INTERVAL TSAMP=0.001
    PROCEDURAL
      Model 1 AFC
      Model 1 APM
      Model 1 ACS
                                        } Code Executed Sequentially
      Model 2 AFC
      Model 2 APM
      Model 2 ACS
    END
  END
END
```

Figure 4.7. Two axis simulation model structure.

The functions' instructions are grouped in sections within the PROCEDURAL block, with the sections being implemented sequentially. By including the PROCEDURAL block within the DISCRETE section, the sampling action of the distributed controller is reproduced. The three sub-sections for each actuator controller are grouped together for neatness of the model structure. This also serves to give a modular feel to the model. By structuring the sequential code in this way, the procedures and logic of each functional sub-section of the controller may be developed independently.

In the distributed controller topology, the ACS's communicate in a pieline scheme. By means of their communications they decide which actuator should move and the chosen ACS sends its AFC a command to move. The AFC then follows its self contained intermittent motion profile, while the APM monitors the motion and deduces the state of the actuator and product, which it relays to the ACS. The AFC follows the profile until either it is completed or it receives an emergency stop command. When the motion is completed, the APM informs the ACS and thus the pipeline is updated on the state of the product flow.

The PROCEDURAL block enables the simulated AFC's, APM's and ACS's to interact, but a method was required to enable the ACS's to trigger the motion of the actuators. This can be achieved with the aid of ACSL function generation operators. In ACSL a step input at one second into the simulation run may be generated using the instruction:

```
INPUT = STEP (1.0)
```

A ramp input of magnitude five/second beginning at 0.5 seconds into a run may be generated by:

```
INPUT = 5.0 * RAMP (0.5)
```

Complex motion profiles may be defined by combining these functions. For example, a trapezoidal velocity profile may be defined as:

```
INPUT = AMAG*(RAMP(T0)-RAMP(T1)-RAMP(T2)+RAMP(T3))
```

Where AMAG is the required acceleration, T0 is the start time of the profile, T1 is the time the acceleration ends, T2 is the time the braking begins and T3 is the time the braking ends. An alternative defintion of the same profile is:

```
ACCEL = AMAG*(STEP(T0)-STEP(T1)-STEP(T2)+STEP(T3))
INPUT = INTEG(ACCEL,0.0)
```

The same profile may be defined in terms of a start time TSTRT, and three segment times:

```
CONSTANT T1=1.0 , T2=2.0 , T3=3.0
ACCEL=AMAG*(STEP(TSTRT)-STEP(TSTRT+T1)-
STEP(TSTRT+T2)+STEP(TSTRT+T3))
INPUT=INTEG(ACCEL,0.0)
THETA=INTEG(INPUT,0.0) + THETA0
```

Where THETA is the required position and THETA0 is the reference position, which initially is zero. By setting TSTRT to a large value which is outside the normal run time of the simulation, say 1000 during the INITIAL section, the profile will not have an automatic effect during the normal run period of 1 to 2 seconds. Instead it will be a line of code that describes a motion profile relative to a start time, TSTRT. This can act as the trigger by which the ACS can start the actuator motion. When the ACS logic determines that the actuator must move, the mechanism for starting the motion will be:

```
TSTRT = T
```

This will then move the temporal origin of the profile to the time at which the move decision was made. After the profile has been completed, i.e. at T > TSTRT + T3, the APM must then reset the profile for the next increment :

TSTRT = 1000

THETA0 = THETA0 + INCRMT

Where INCRMT is equal to one incremental distance. Using this method, the ACS can trigger the required motion profile, which the AFC will drive the continuous model to follow and the APM will reset at the end of the motion.

Having developed a method to trigger the continuous motions from the PROCEDURAL block controller model, it was then possible to use this mechanism to develop the actuator synchronisation and product flow coordination logic for systems of drives. A five axis system will now be described and this will be used as an example of how the distributed pipeline control scheme operates.

4.4.2. Simulating a five axis intermittent actuator system.

The test rig which is described in chapter 6 was developed to facilitate testing of methods for the driving and control of independent intermittent actuator motions. The rig consists of two actuators, a linear slider and an angular drum, but it does not have facilities for the introduction of a flow of product. To test methods for controlling the product flow and actuator synchronisation it was therefore necessary to develop simulated multiple actuator systems, through which a theoretical flow of product could be coordinated. Figure 4.8 shows an example of such a system containing examples of most elements of the independent drives control problem. This is a five actuator system consisting of two incremental drums, two transfer sliders and a packet feed actuator having an in-out intermittent action.

The second transfer slider transfers products from a buffer and into an arbor of the arbor drum. The drum has a cyclic increment-dwell motion., which moves the full arbor away from the second transfer slider and so clears the path for the next product input. The packet feed is a twin roller actuator which feeds packets around the arbor containing the product during the dwell in the drum's motion. When the product and packet has been further incremented to the third transfer position, the slider moves the product and the packet into an arbor in the drying drum. During this transfer, both the arbor drum and the drying drum must be stationary and in position. The drying drum then increments to move the packeted product to its next position. Thus the product flow is :

(i) Second transfer out of buffer into arbor drum.
(ii) Increments around arbor drum.
(iii) Packet feeds from buffer around arbor containing product.
(iv) Increments around arbor drum.
(v) Third transfer out of arbor drum into drying drum.
(vi) Increments around drying drum.

The removal of the packet from the drying drum is not considered in this model.

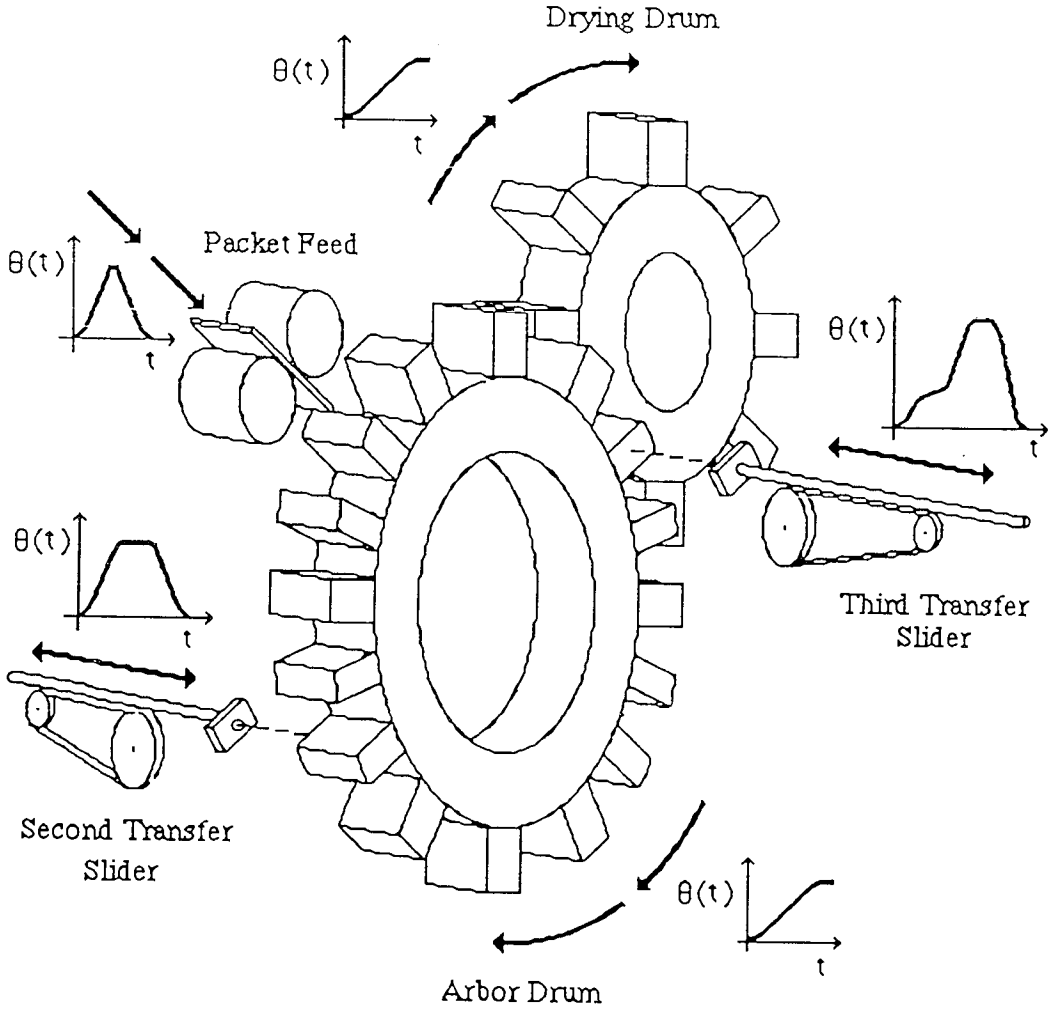


Figure 4.8. Five axis independent actuator system.

To achieve an accurate representation of continuous system performance, the five actuator models are all based on the two actuators of the test rig. The third transfer and arbor drum use the models and profiles for the same actuators that were developed in chapter 6. The simulated arbor drum and third transfer slider are driven by S-6200 and S-4050 brushless d.c. motors respectively. So all parameters of the two motor models have been tested in the laboratory. The simulated drying drum uses the same S-6200 model as the arbor drum, but it is considered to have eight arbors whereas the arbor drum has sixteen. The second transfer slider and the packet feed both use the S-4050 motor and controller model of the third transfer slider. Each of the five axes therefore uses the continuous model shown in Figure 4.9, with the motor and controller parameters set to match the brushless d.c. motor being modelled.

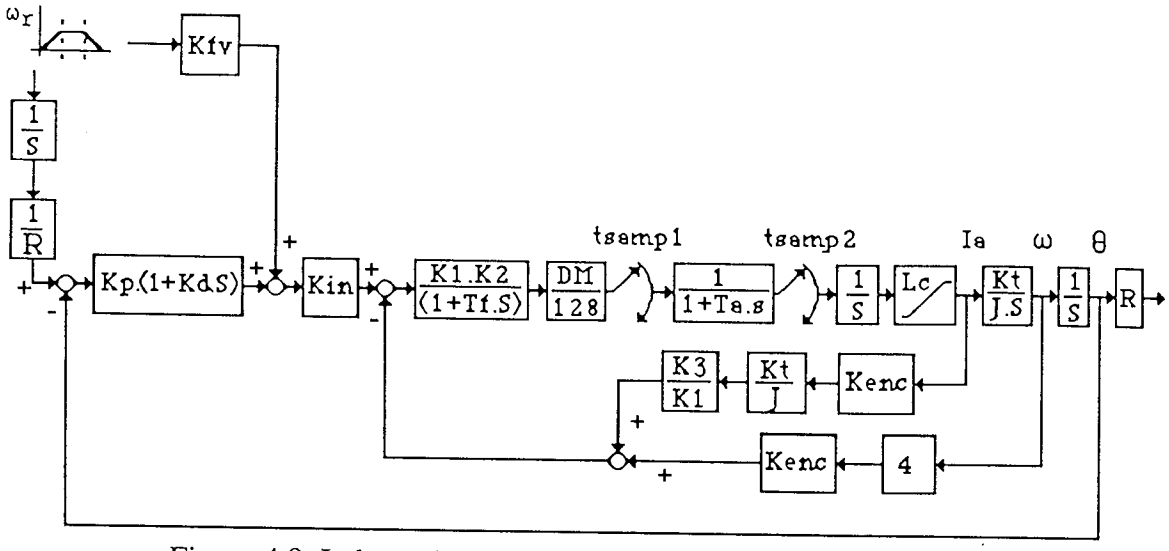


Figure 4.9. Independent actuator contiuous system model.

4. 4.3 Synchronising the motions of the five axis system.

The five actuators each share regions of spatial overlap with one or more other actuators. The ACS functions must ensure that such regions are never occupied by more than one actuator at a time. An actuator can only commence a motion if the MOTION states of all overlapping actuators are not OCCUPATION or INTENTION. Therefore for the five actuator system each region of spatial overlap must be determined and each actuator's use of overlap regions must be defined.

Figure 4.10 shows the relative positions of the five actuators. These show the four regions of spatial overlap : the second transfer slider and the arbor drum, the third transfer slider and the arbor drum, the third transfer slider and the drying drum, the packet feed mechanism and the drying drum. These overlap regions will now each be considered and the conditions of occupation and intention to occupy for each actuator will be defined.

1. Second transfer slider and arbor drum.

The second transfer slider contacts the product 5mm into the insertion stroke. At 15mm into the stroke the product enters the drum, which is the entry of the slider and product into the overlap region. The total length of stroke is 123mm, at which the slider begins the retraction stroke. The slider leaves the drum and the overlap region 100mm from the retracted postion. Thus the second transfer slider occupies the overlap region from 15mm into the insertion stroke until the point on the retraction stroke that is 100mm from the origin. In addition, it has the intention to enter the overlap region from receipt of the move command until it is 15mm into the insertion stroke.

The arbor drum has a dwell position tolerance of $^{+}_{-}$ 0.35 ^{0}. When it is not within this tolerance zone it occupies the overlap region. It has the intention of occupying the overlap region during the period from receipt of the move command until it leaves the $^{+}_{-}$ 0.35 ^{0}.zone.

The required motion profiles of the second transfer slider and the arbor drum are plotted against a common time axis in Figure 4.11 (a). This shows the relative periods of occupancy and intention to occupy the spatial overlap region of the two actuators.Safe synchronisation requires that either actuator can only have occupancy or intention to occupy the overlap region when the other actuator is clear.

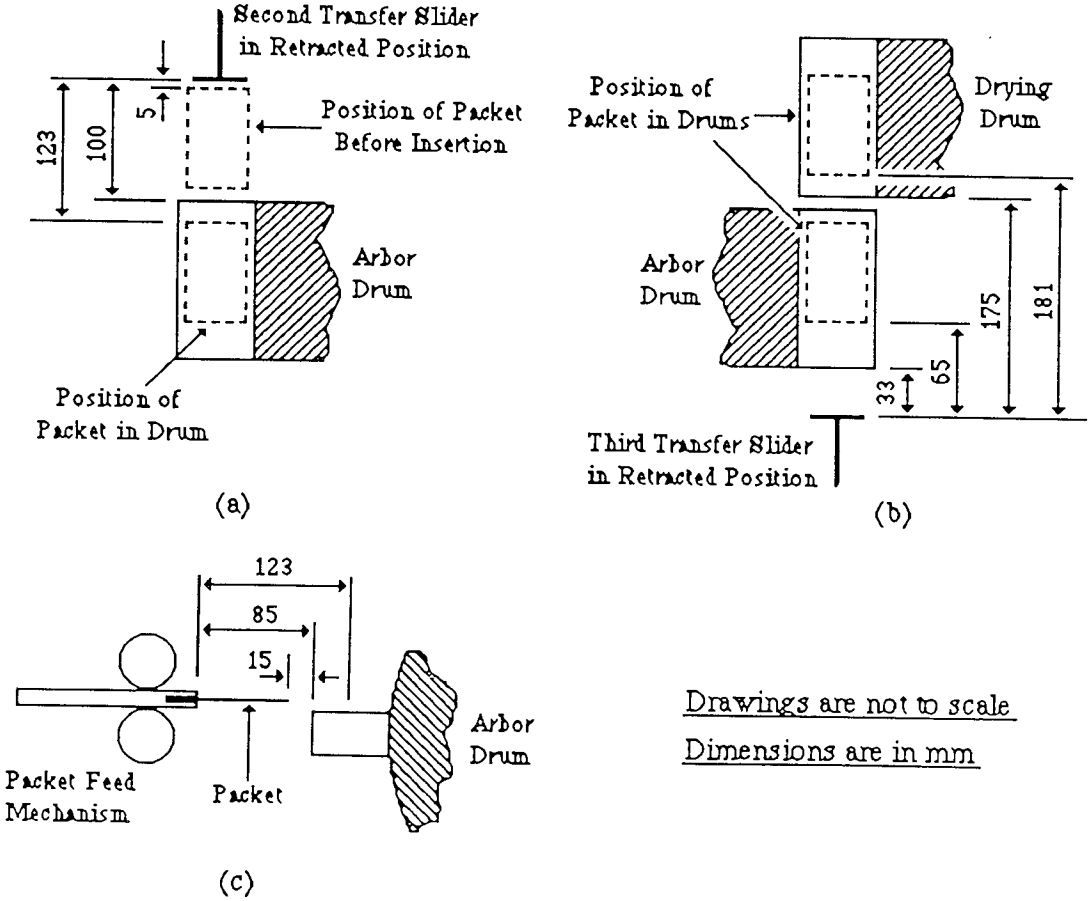


Figure 4.10. Relative positions of the five intermittent actuators.
(a) Second transfer slider and arbor drum.
(b) Third transfer slider, arbor drum and drying drum.
(c) Packet feed mechanism and arbor drum.

2.Third transfer slider and arbor drum.

The third transfer slider enters the arbor drum and the region of overlap 33mm into the insertion stroke. It then occupies the overlap region until the point on the retraction stroke that is 33mm from the origin. It has the intention to occupy the

overlap region during the period form receipt of the move command until it is 33mm into the insertion stroke.

The overlap conditions for the arbor drum are the same as given above for the arbor drum/second transfer slider overlap region.

The required motion profiles of the third transfer slider and arbor drum are shown against a common time axis in Figure 4.11 (b). This shows the relative periods of occupancy and intention to occupy the overlap region of the two actuators.

3. Third transfer and drying drum.

The third transfer slider contacts the packet in the arbor drum 65mm into the insertion stroke. The product enters the drying drum when the slider is 90mm into the insertion stroke. The total stroke length is 181mm and the slider leaves the drying drum on the retraction stroke when it is 175mm from the retracted position . Thus the third transfer slider occupies the region of overlap from 65mm into the insertion stroke until it is 175mm from the origin on the retraction stroke. It intends to occupy the overlap region from receipt of the move command until 65mm into the insertion stroke.

The overlap conditions for the drying drum are identical to those of the arbor drum in the previous two cases, with the same tolerance of $^{+}_{-}$ 0.35 ^{0}.being in force.

The required motion profiles of the third transfer slider and the drying drum are plotted against a common time axis in Figure 4.11 (c). This includes the periods of accupancy and intention to occupy the overlap region of the two actuators.

4. Packet feed and arbor drum.

The packet feed mechanism moves the packet into the overlap region 15mm into the insertion stroke. The peak insertion of the mechanism is 123mm and the mechanism leaves the overlap region on the retraction stroke when it is 85mm from the fully retracted position. In addition it intends to occupy the overlap region during the period from receipt of the command to move until it is 15mm into the insertion stroke.

The overlap conditions for the arbor drum are the same as for the arbor drum/second transfer overlap region.

The required motion profiles of the packet feed and arbor drum are shown in Figure 4.11 (d) against a common time axis. This includes the periods of occupancy and intention to occupy the overlap region of the two actuators.

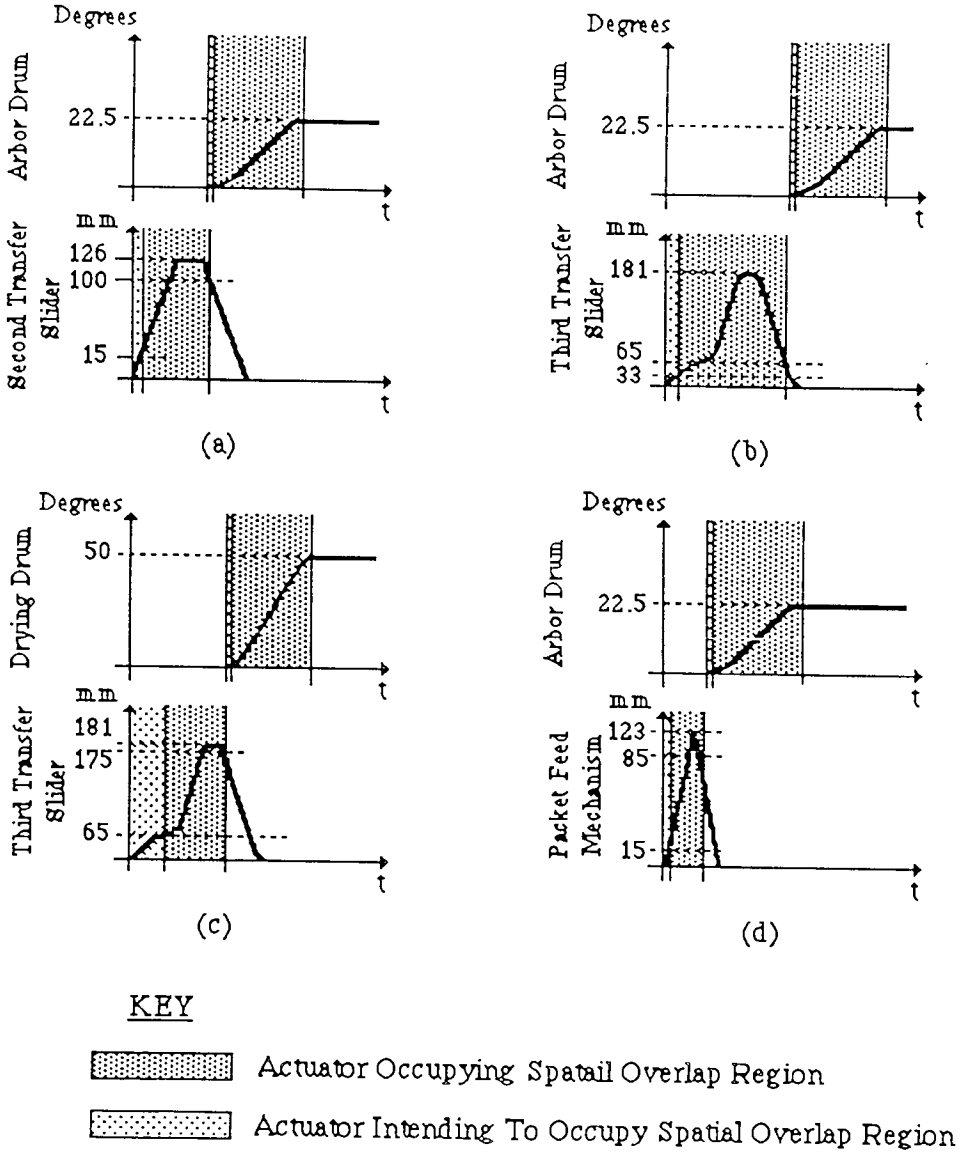


Figure 4.11. Relative actuator motion profiles showing periods of occupation and intention to occupy spatial overlap regions.

(a) Second transfer slider and arbor drum overlap region.

(b) Third transfer slider and arbor drum overlap region.

(c) Third transfer slider and drying drum overlap region.

(d) Packet feed mechanism and arbor drum overlap region.

The four spatial overlap regions each enclose a volume in which two actuators could collide if their motions are not synchronised. It can be seen that some actuators have envelopes which encompass more than one overlap region, notably the arbor drum and the third transfer slider. In an independent drives machine incorporating a

pipeline product flow, it would be expected that this would be the case for many of the actuators. The fundamental synchronisation rule is that only one actuator at a time may have occupation or intention to occupy an overlap region. The implementation of this means that when an actuator is given the command to move, it effectively disables all other actuators it shares overlap regions with. Furthermore the actuator can not be issued the move command unless all overlap regions it passes through are clear.

4 4.4 Example of an ACSL actuator control module implementation.

As an example of how the synchronisation works, consider a move of the third transfer slider. The coordination elements of the controller determine that there is a product in the arbor drum which needs to be transferred out, the receiving arbor of the drying drum is empty and the third transfer slider is retracted and stationary. The slider's envelope passes through two spatial overlap regions, one with the arbor drum and one with the drying drum. The slider's ACS must check the ACS's of the arbor drum and drying drum to check that their overlap states are CLEAR and not OCCUPATION or INTENTION to occupy. The slider's ACS then sends the move command to its AFC and APM functionsand sets the slider's two overlap states to INTENTION. The APM monitors the actuator and informs the ACS when firstly the slider/arbor drum and secondly the slider/drying drum overlap states change to OCCUPATION. As the slider leaves the drying drum, the APM informs the ACS which changes the second overlap state to CLEAR. This clears the drying drum to increment. Similarly, when the APM detects the slider leaving the arbor drum, the ACS changes the first overlap state to CLEAR, which allows the arbor drum to move if it is able.

The FORTRAN code describing the third transfer slider's control module is listed in Figure 4.12, which has been taken from the ACSL simulation model of the five actuator system. The software is separated into its three component parts of AFC, APM and ACS. The first component to be executed is the AFC. This is a finite difference position controller having a propotional plus integral plus derivative plus velocity feedforward algorithm. The required position and velocity profiles are defined within the sorted code of the continuous section of the model, with the motion trigger being of the form described in section 4.4.1

The second component of the controller is the APM. This has three functions: setting the MOTION condition of the system by interpretation of the actuator's dynamics, defining the PRODUCT condition of the actuator, also by interpreting the dynamics and resetting the commence motion mechanism at the end of the

transfer cycle. The first line of APM coding serves to update the PRODUCT condition, TTPROD to match that set by the ACS. This resets the condition at the controller sample following the receipt or passing of products. The next APM action is the interpretation of Type 3, emergency stop events. This action checks that the slider's positiion, TTPOS is within a tolerance band about the required position, $TTPS^{+}_{-}$ TTPDAN. A similar check is carried out for the velocity, TTVEL being within the tolerance band $TTVS^{+}_{-}$ TTVDAN. Should either the position or velocity be outside the tolerance bands, the program jumps to label TTP5 where the MCHINE flag is set to the value STOPTT, which is recorded on the simulation output file to show that the APM has detected a fault. A discussion of machine failure simulation is included in section 4.4.6.

The next lines of coding detect Type 2, damaged product events. These concentrate on the detection of position error at the extreme insertion point and velocity error at the point at which the slider contacts the product on the insertion stroke. If at the extreme insertion position the slider is outside the position tolerance, TTPTOL, or if on packet contact it is outside the velocity tolerance, TTVTOL, a type 2 event is registered and the PRODUCT condition TTPROD is set to DAMGED. The next lines of APM coding set the MOTION condition, TTMOT. This is separated into motion on the insertion stroke and motion on the retraction stroke. TTMOT is set to one of six states for interpretation by the ACS into the slider/arbor drum and slider/drying drum MOTION conditions. These are: INTENT, the slider intends to occupy both drums; OCC1, the slider is inside the arbor drum on the insertion stroke but has not contacted the product; OCC2, the slider's MOTION condition for both drums is OCCUPATION; CLEAR1, the slider has cleared the drying drum on the retraction stroke but still occupies the arbor drum; CLEAR2, the slider has cleared both drums; READY, the motion cycle is complete and the slider is ready to commence the next cycle. The READY state is set at the end of the cycle when the position, velocity and acceleration are within the holding tolerances. The setting of READY is accompanied by resetting of the motion trigger, TTSTRT to NOGO.

The third component of the actuator controller is the ACS. The majority of ACS coding is concerned with interpreting the MOTION information from the APM. The first line checks whether the APM has detected a Type 2 event. If this is the case, the PRODUCT condition, TTPRDT is set to DAMGED. The lines up to TTCS1 set the slider/arbor drum MOTION condition, TTADM and the slider/drying drum MOTION condition, TTDDM to INTENT if the APM condition, TTMOT is INTENT. The lines from TTCS1 to TTCS2 cover the slider's motion within the arbor drum before contacting the product. From TTCS2 to TTCS3 the coding

interprets the taking of the product and OCCUPATION of both overlap regions. Tha taking of the product is accompanied by setting TTPRDT from the output arbor of the arbor drum,ADOPRD. The lines from TTCS3 to TTCS4 cover the clearing of the drying drum on the retraction stroke and the resetting of TTPRDT to EMPTY. Lines TTCS4 to TTCS5 cover the clearance of the arbor drum on the retraction stroke.

The coding beginning at TTCS5 handles the ACS's decision to commence actuator motion. The APM condition TTMOT is READY, so the ACS sets TTADM and TTDDM to READY. It then checks that the arbor drum and drying drum MOTION conditions are not INTENT od OCCPTN, the arbor drum output product ADOPRD is not EMPTY and the drying drum input product, DDIPRD is EMPTY. This indicates that it is safe and necessart to move, so the command TTSTRT = T is issued and the two MOTION conditions are set to INTENT.

```
        "Third Transfer AFC"
        TTP = TTEROR $" Proportional"
        TTI = (TTEROR * TSAMP) + TTI $" Intergral"
        TTD = (TTEROR - TTNM1) / TSAMP $" Derivative"
        TTVFF = TTV1 $" Velocity Feedforward"
        TTOUT = TTKP*(TTP + (TTKI*TTI) + (TTKD*TTD)) + (TTKFF*TTVFF)
        TTNM1 = TTEROR

        " Third Transfer APM"
        IF(TTPROD.NE.TTPRDCT) TTPROD = TTPRDT $"PRODUCT Condition
Update"
        IF(TTPOS.GT.TTPS+TTPDAN.OR.TTPOS.LT.TTPS-TTPDAN) GO TO TTP5
$"Position Danger"
        IF(TTVEL.GT.TTVS+TTVDAN.OR.TTVEL.LT.TTVS-TTVDAN) GO TO TTP5
$"Velocity Danger"
        IF(TTPOS.LT.TTPS+TTPTOL.AND.TTPOS.GT.TTPS-TTPTOL) GO TO TTP1
        IF(TTPS.GT.0.179) TTPROD = DAMGED $"Position Overshoot Damage"
        TTP1..CONTINUE
        IF(TTVEL.LT.TTVS+TTVTOL.AND.TTVEL.GT.TTVS-TTVTOL) GO TO TTP2
        IF(TTPOS.GT.0.06.AND.TTPOS.LT.0.07) TTPROD = DAMGED $"Velocity
Contact Damage"
        TTP2..CONTINUE $" PRODUCT Condition Has Been Set"
        IF(TTSTRT.EQ.NOGO) GO TO TTP6 $"If Between Motion Cycles"
        IF(T.GT.TTSTRT + TTT2) GO TO TTP4 $"If At End Of Transfer"
        IF(T.GT.TTSTRT + TTT1) GO TO TTP3 $"If On Retraction Stroke"
        IF(TTPOS.LT.0.033) TTMOT=INTENT $"Outside Both Drums"
        IF(TTPOS.GE.0.033.AND.TTPOS.LT.0.09) TTMOT=OCC1 $"In Arbor
Drum"
        IF(TTPOS.GE.0.09) TTMOT=OCC2 $"In Both Drums"
        GO TO TTP6 $"Insertion Condition Has Been Set"
        TTP3..CONTINUE $"Retraction Stroke"
        IF(TTPOS.LT.0.175) TTMOT=CLEAR1 $"Clear of Drying Drum"
        IF(TTPOS.LT.0.033) TTMOT=CLEAR2 $"Clear of Both Drums"
        GO TO TTP6 $"Retraction Condition Has Been Set"
        TTP4..CONTINUE $" Look For End Of Transfer"
        IF(TTPOS.GE.TTPTOL.OR.TTPOS.LE.-TTPTOL) GO TO TTP6 $"Position"
        IF(TTVEL.GE.TTVTOL.OR.TTVEL.LE.-TTVTOL) GO TO TTP6 $"Velocity"
        IF(TTACC.GE.TTATOL.OR.TTACC.LE.-TTATOL) GOTO TTP6 $"Acceln"
```

```
        TTMOT = READY
        TTSTRT = NOGO $"Reset Trigger for Next Cycle"
        GO TO TTP6
        TTP5..CONTINUE $"Set Danger Flag"
        MCHINE = STOPTT $ Stop Indicated by Third Transfer"
        TTP6..CONTINUE $"End Of APM"

        "Third Transfer ACS"
        IF(TTPROD.EQ.DAMGED.AND.TTPRDT.EQ.GOOD) TTPRDT=DAMGED $"Update
From APM"
        IF(TTMOT.NE.INTENT) GO TO TTCS1  $If Not First Segment"
        TTADM = INTENT $"Third Transfer/Arbor Drum MOTION Condition = INTENT"
        TTDDM = INTENT $"Third Transfer/Drying Drum MOTION Condition = INTENT"
        GO TO TTCS6 $"Condition Set"
        TTCS1..CONTINUE
        IF(TTMOT.NE.OCC1) GO TO TTCS2 $"If Not Second Segment"
        TTADM = OCCPTN
        TTDDM = INTENT
        GO TO TTCS6 $"Condition Set"
        TTCS2..CONTINUE
        IF(TTMOT.NE.OCC2) GO TO TTCS3 $"If Not Third Segment"
        TTADM = OCCPTN
        IF(TTDDM.EQ.OCCPTN) GO TO TTCS6 $"Look For Taking of Product"
        TTPRDT = ADOPRD $" Product Taken From Arbor Drum Output"
        TTDDM = OCCPTN
        GO TO TTCS6 $"Condition Set"
        TTCS3..CONTINUE
        IF(TTMOT.NE.CLEAR1) GO TO TTCS4 $"If Not Fourth Segment"
        TTADM = OCCPTN
        IF(TTDDM.EQ.CLEAR) GO TO TTCS6 $"Look For Passing of Product"
        TTDDM = CLEAR
        TTPRDT = EMPTY $"Product Passed"
        GO TO TTCS6 $"Condition Set"
        TTCS4..CONTINUE
        IF(TTMOT.NE.CLEAR2) GO TO TTCS5 $" If Not Fifth Segment"
        TTADM = CLEAR
        TTDDM = CLEAR
        GO TO TTCS6 $"Condition Set"
        TTCS5..CONTINUE $"Between Motion Cycles"
        TTADM = READY
        TTDDM = READY
        IF(ADTTM.EQ.INTENT.OR.ADTTM.EQ.OCCPTN) GO TO TTCS6 $"Check
Arbor Drum Safety"
        IF(ADOPRD.EQ.EMPTY) GO TO TTCS6 $"Check Arbor Drum Has Product"
        IF(DDTTM.EQ.INTENT.OR.DDTTM.EQ.OCCPTN) GO TO TTCS6 $"Check
Drying Drum Safety"
        IF(DDIPRD.NE.EMPTY) GO TO TTCS6 $"Check Drying Drum Needs Product"
        TTSTRT = T $"Trigger Motion"
        TTADM = INTENT
        TTDDM = INTENT
        TTCS6..CONTINUE $"End Of ACS"
```

Figure 4.12. Third transfer slider actuator control module software.

4.4.5 Implementing the five axes simulation.

Each of the axes in the ACSL model of the five actuators system has an AFC, APM, ACS control module of the form described above and a dynamic model of the form described in section 4.4.2. The communications between modules takes place

at the ACS level, with communications occuring between actuators which interact directly. The communications interconnection for the five actuators system are shown in Figure 4.13.

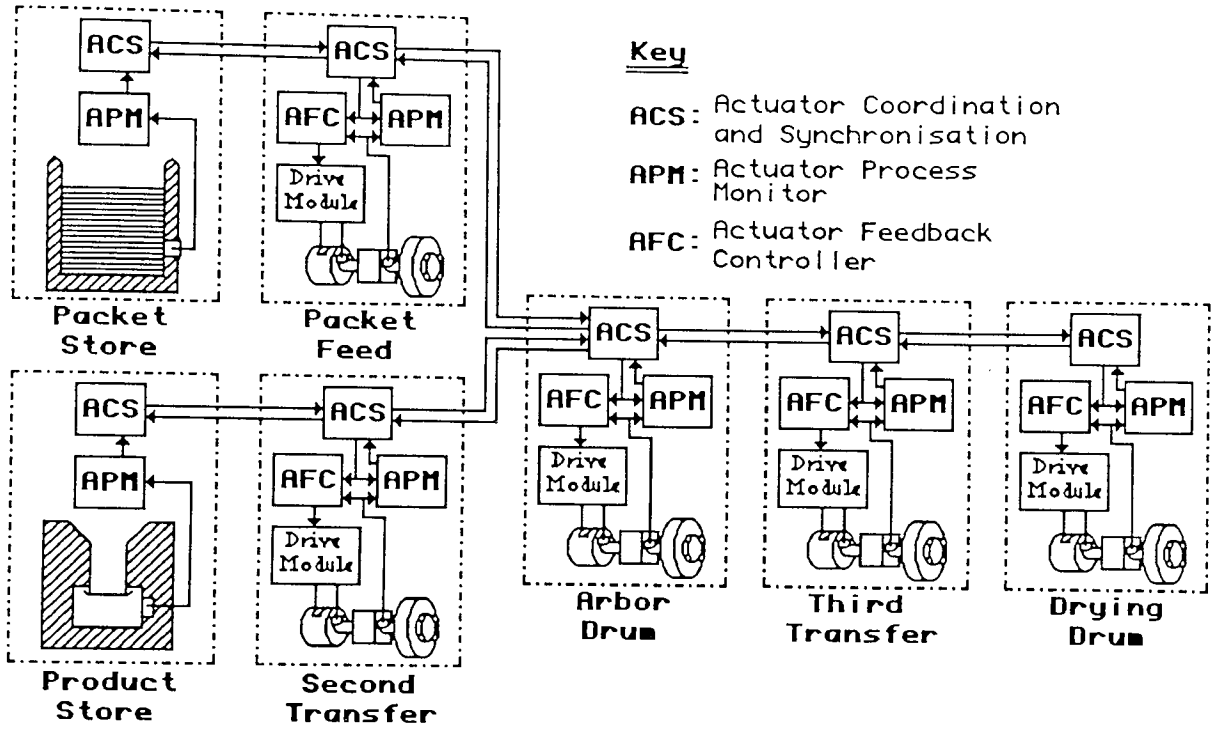


Figure 4.13 Multi actuator control modules communication links.

For the purposes of the simulation, the products are considered to be input to the arbor drum in position 0 by the second transfer from a buffer store. The packet is fed around the arbor in position 1, with the packets being fed from a second buffer store. The packets and products are removed from the arbor which is in incremental position 3 by the third transfer slider. The simulated position responses of the five actuators are shown in Figure 4.14. The operation of the product buffer is simulated by incrementing through an array at each increment of the arbor drum. The first 15 elements of the array are 1,1,0,1,1,1,0,0,1,1,0,1,1,1,1, with 1 representing a full buffer and 0 representing an empty or partially full buffer. This enables the flow of products to be readily observable, since the transfer mechanisms and packet feed only operate when there is product in the pipeline. The three linear mechanisms are triggered by product input, with the packet feed lagging the second transfer by one increment of the arbor drum and the third transfer lagging the second transfer by three increments.

The motion stimulii for each independent actuator are defined according to the requirements of the product and the machine. The ACSL simulations enable the stimulii for each actuator to be altered and tested to assess the effect of such changes on machine performance. The resolution of the five plots is not sufficient to enable

detailed comparison of the effects of such changes, but the ACSL simulation allows any variable or logic condition to be output to a separate file, or drawn as a full page plot. This enables the effect of motion profile or actuator hardware changes to be assessed both at the single actuator and machine pipeline levels, since the simulation is a detailed model of the complete software-coupled, multi-actuator machine system. By software-coupling proven models of independent actuators, a simulation model has been formed which enables all levels of machine performance to be examined, from the highest level of material input - product output studies, down to the lowest level of examining the current in the windings of an individual actuator and hence evaluating the energy consumption under given operational conditions.

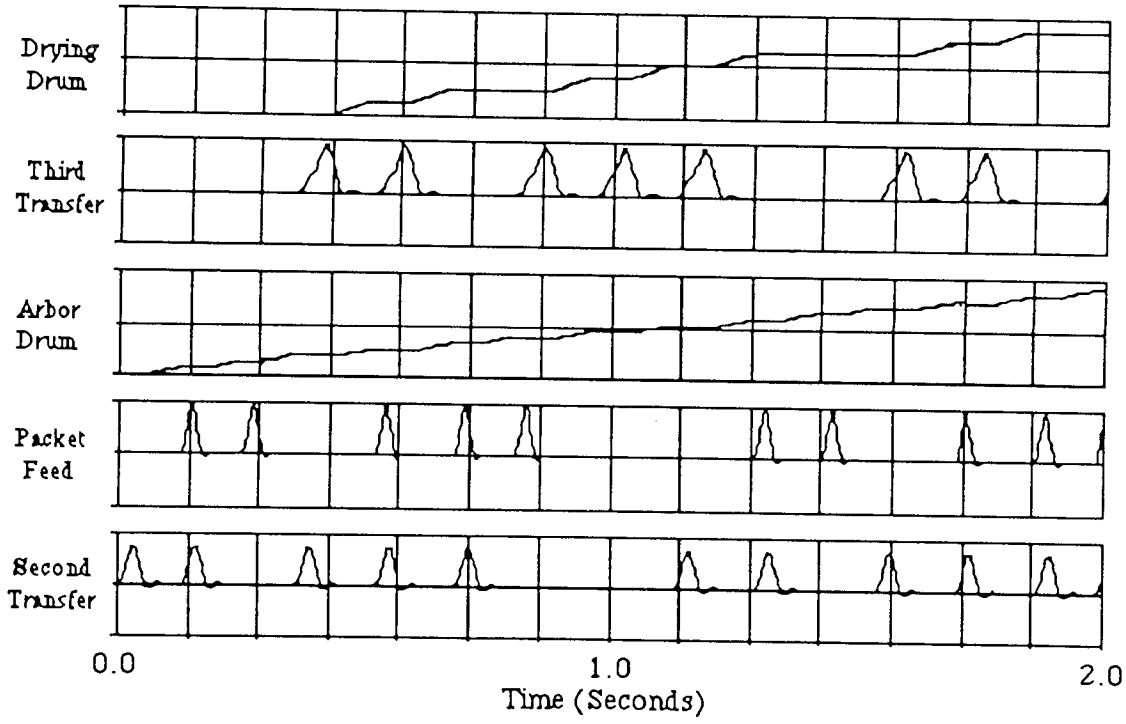


Figure 4.14. Simulated performance of the software coupled five actuator system

4.4.6 The simulation of the machine's response to unpredictable events.

The simulation of mutiple, software-coupled independent actuators enables a tool for the design and development of flexible machine systems. This capability may be extended to the analysis of the machine's response to unpredictable events, such as time dependent deteriation, hardware failures or externally triggered events such as emergency stop. The actuator's APM's detect events which result from unexpected machine performance. The machine must be designed to properly respond to such events. The mutiple actuator simulation capability enables machine designs to be

tested by simulating hardware failures and checking that the machine responds proerly and safely.

The five actuator simulation described above records the detection of unexpected events in an output file, but since the machine is a relatively simple intermittent actuator pipeline, it does not alter its operations as a result of such detection. The testing of the response to unexpected events, particularly Type 2 failures, requires that the machine model has alternative courses of action which can be taken other than to merely stop the simulated motions. Figure 4.15 shows a system that was developed to investigate both the handling of unexpected events and the suitability of the modular controller software for controlling systems of intermittent and continuous motions.

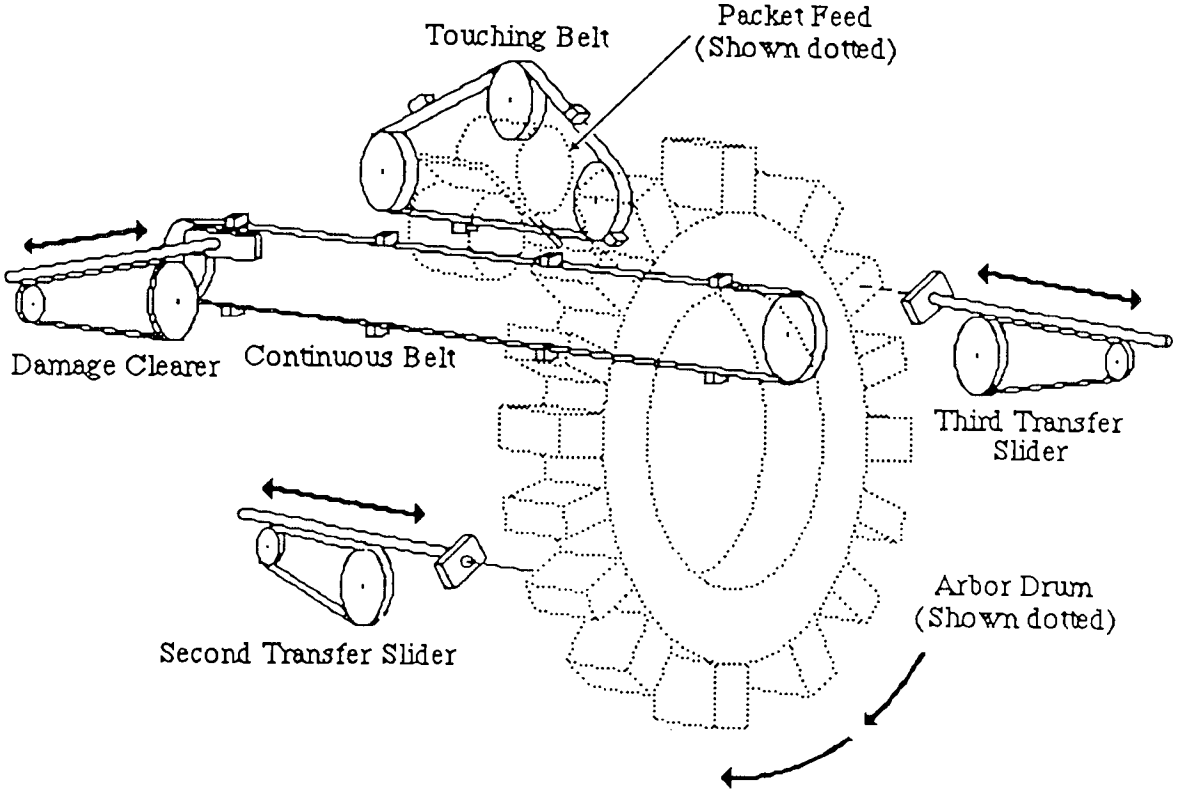


Figure 4.15. Seven independent actuators system incorporating intermittent and continuous actuators and alternative product flow options.

The seven actuator system was developed by replacing the drying drum of the five actuator system with two continuous belts and an intermittent slider mechanism. In the new system the third transfer slider transfers products out of the arbor drum, which is shown dotted in Figure 4.15, onto the continuous belt. The belt has eight equally spaced stops on its surface and the packet is placed against one of the stops. The third transfer slider must therefore match velocity and position with the stop to ensure a smooth transfer. A second belt, called the touching belt, is mounted above and parallel to the continuous belt. The touching belt has four equally spaced stops

on its surface, which represent packet sealers. The touching belt matches velocity and position with the continuous belt with the stops on the touching belt lagging 100mm behind those on the continuous belt. When a product is placed against a stop on the continuous belt, the relative position of the touching belt advances by 15mm, so that the one of the sealer stops touches the packet. Contact is maintained for 100ms, before the touching belt retards to the 100mm relative lag position.

The seventh independent actuator in the system is an intermittent slider called the damage clearer. Its motion passes directly across the path of the continuous belt at a position after the product path has cleared the touching belt. If a product passes along the belt whose condition is not GOOD, the damage clearer moves to clear the product from the belt. If the product's condition is GOOD, the damage clearer takes no action. To maintain the principal of using proven actuator models in the simulations, the damage clearer uses the same contiuous slider model as the second and third transfer mechanisms. The slider is required to follow an in-out position profile of magnitude 75mm in a total time of 50ms.

The seven actuator system includes synchronised intermittent and continuous motions and the introduction of alternative product flow choices for quality control. The continuous models of the two belts employ the Bru-500 S-4050 model described in section 4.4.2, with the inertias of both belts being set equal, which simplifies the task of synchronising the motions. The required velocity and position of the continuous belt are defined as:

```
CBVR=CBMAG*(RAMP(0.0) - RAMP(CBT1)) $"Required Velocity"
CBSR=INTEG(CBVR,0.0) $"Required Position"
```

Using these profiles, the required position profile of the touching belt is define as:

```
TBSR=CBSR-LAG+BULGE+(KTB*(CBPOS-TBPOS))
```

where TBSR is the required position of the touching belt, LAG = 100mm, BULGE is the 15mm position advance trigger, CBPOS is the actual position of the continuous belt, TBPOS is the required position of the touching belt and KTB is the proportional relative position gain. It is necessary to put a lower limit of zero on TBSR to avoid a negative required position profile during start-up. The gain KTB is used to remove any position error which exists between the two belts. However in practice since the two simulation models are identcal, they give identical responses. So in all simulation runs KTB was set to zero. It should be noted that in an actual application of the system, the two drives would be very unlikely to give identical responses, so KTB would be used bring the belts into synchronism.

The position advance trigger BULGE is initially set to zero. The touching belt's ACS sets BULGE to 0.015 to trigger the position advance in the same way that the third transfer slider's ACS was earlier shown to set TTSTRT to T to trigger its motion. When the touching belts APM detects that the position has been properly advanced for 100ms, it resets BULGE to zero for the next packet seal operation.

Figure 4.16 shows the ACSL output plots of the motions of the seven actuators during normal machine operations. The second transfer, packet feed, arbor drum, third transfer and damage clearer plots are of position against time and the touching belt and drying belt plots are of velocity against time.

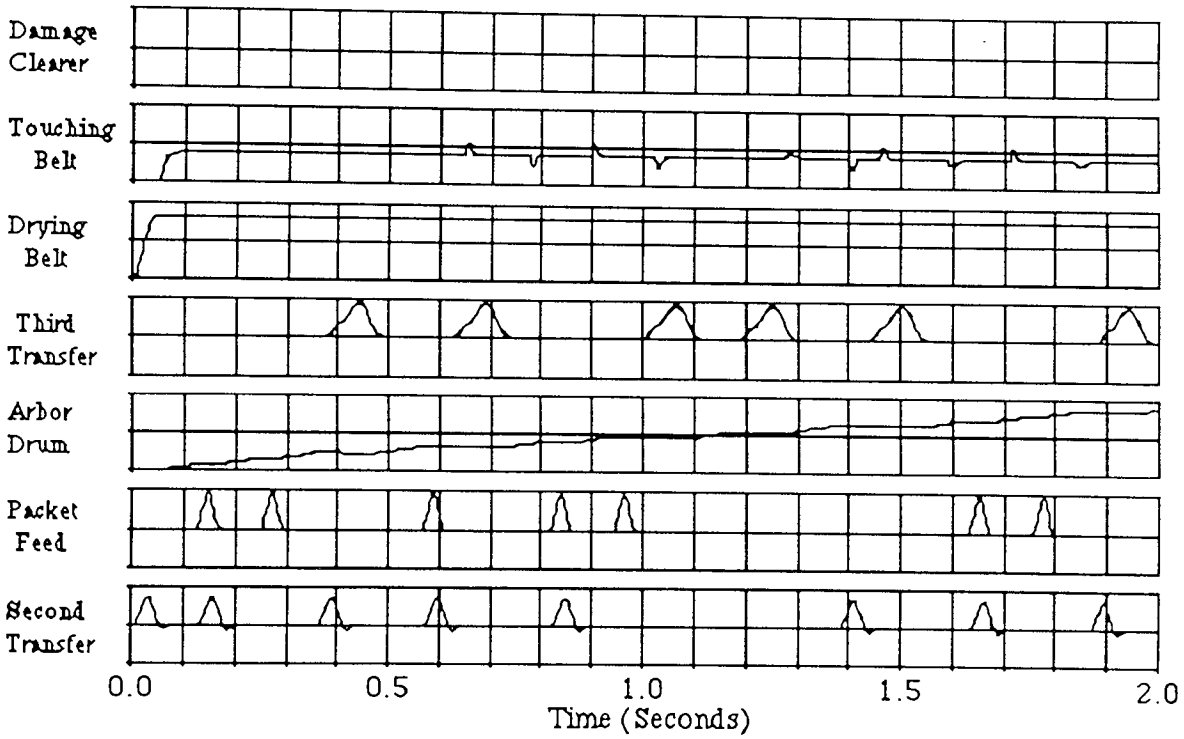


Figure 4.16 Simulated synchronised motions of seven actuator system.

The product input through the second transfer slider follows the same patern that was used for the five actuator system. The flow through the seven actuator system is approximately 20% slower than that through the five actuator system, with the cause of the slow-down being the third transfer onto the continuous belt. The transfer slider has to match its position and velocity to a stop on the continuous belt, so the slider can not begin a transfer until a stop is in the correct position for the matching of dynamics to be performed. This causes the slider to wait for the proper time to commence motion. In addition, the belt's velocity is 3m/sec, which is therefore the peak velocity of the transfer slider after it has contacted the product at 1m/sec. This compares with the peak insertion velocity of 5.5 m/sec for the transfer of the five actuator system.

After the products are placed on the drying belt, the touching belt accelerates and decelerates to advance the sealer to touch the packet for the 100 msec contact duration. This action can be seen in the velocity plot of the touching belt.

The damage clearer can be seen to have not moved during the simulation run. The damage clearer only operates if an event has caused a product to be damaged. In the simulation run of Figure 4.16 all actuators have performed properly, which resulted in all products being correctly handled. Actuator failures can be introduced into a simulation run to assess the machine's ability to give the proper response. Any failure which can be described mathematically can be included in the simulation model of one or more of the actuators. The exact details of the type of failure depend upon the configuration of machine being modelled, with the type of function being used to describe the event being defined by the nature of the failure.

In section 4.2.3.2 it was described how the APM must be capable of detecting power fluctuations, actuator failures or mechanism failures. The method of implementing these was by the detection of tolerance bands to indicate the occurence of dynamic fluctuations. Changes in power, actuator characteristics or mechanism characteristics can be introduced into the seven actuator simulations Simulation runs have been performed with the system which have included the three general sources of failure. As an example of such a simulation, consider a temporary 20% loss of current in the windings of the third transfer driver during the period from 0.9 seconds to 1.1 seconds into the simulation run. This may be readily modelled in ACSL using the instructions:

```
TTV7 = TTCRT * 0.2 * (STEP(0.9) - STEP(1.0))
TTV8 = TTCRT - TTV7
```

Where TTCRT is the current in the windings. The effect of this operator is to introduce a temporary forward path gain of 0.8 into the third transfer current loop model, which serves to reduce the closed loop natural frequency of the drive and increase the damping ratio. So for the period from 0.9 to 1.1 seconds, the drive is less stiff. Figure 4.17 shows the effect that this has on the performance of the seven actuator system.

The plots in Figure 4.17 do not have sufficient resolution to provide a detailed assessment of the performance of the third transfer slider during the partial failure. The purpose of the multiple actuator plots are to investigate the 'knock-on' effects of single actuator failures. It can be seen from the plot that the loss of current causes the third transfer slider to overshoot at peak insertion. This causes the ACS to register the PRODUCT condition as DAMAGED, because the product would be

expected to be crushed against the stop on the continuous belt. The knock-on effect of the product being damaged can be seen by comparing the plots with those of Figure 4.17. The product that has been placed on the continuous belt is damaged. This is detected by the touching belt's ACS, which has as one of its commence motion conditions that the PRODUCT condition must be GOOD. So the touching belt does not advance its position to perform the sealing operation. When the damaged product enters the region of the damage clearer, the damage clearer the PRODUCT as DAMAGED, so the commence motion signal is triggered and the damage clearer removes the product from the continuous belt.

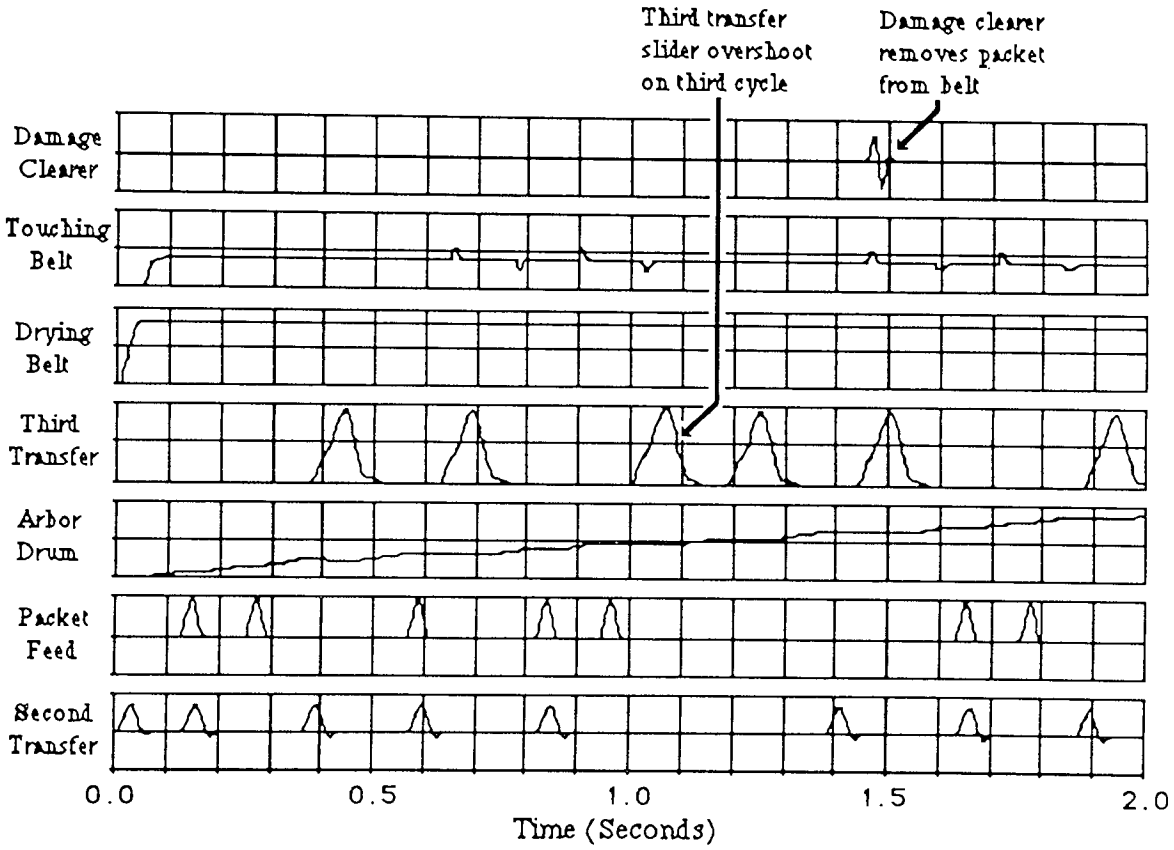


Figure 4.17. Response of seven actuator system to a partial actuator failure.

In addition to the power fluctuation simulation, several similar runs have been performed to assess the response to actuator and mechanism failures these include:

(i) An actuator (the second transfer) running into an obstruction, represented using the BOUND function to limit the motion of the actuator. This resulted in the MCHINE flag being set to STOPST (second transfer danger).

(ii) A gradual increase in friction loading in a mechanism (the drying belt), represented using the RAMP function to introduce a negative torque to the motor. For small magnitude friction torques, this resulted in product being

rejected by the damage clearer, while for large torques, the MCHINE flag was set to STOPDB (drying belt danger).

(iii) A sudden failure in an actuator (the arbor drum) causing freewheeling, represented using the RSW software switch to set the required drum profile to constant velocity during the run, whilst not affecting the profile read by the APM as the required profile. This resulted in the MCHINE flag being set to STOPAD (arbor drum danger).

In each case the controlled system gave the proper response to the machine failure. The simulations described here have enabled the controller's response to power fluctuations, actuator failures and mechanism failures to be tested. The method of detecting and responding to the failures has been shown to work properly, with either the emergency stop being triggered for type 3 failures, or the complete set of actuators giving a coordinated response to type 2 failures.

Chapter 5

The design of servomechanisms for intermittent machine functions

5.1. Introduction.

In traditional intermittent processing machinery, the separate machine functions are performed by cam mechanisms. These convert the transmitted actuation of the driver into the controlled motion of the machine functions. The fixed dynamic properties of the cam mechanisms are utilised to ensure accuracy and repeatability of the function, whilst their open loop nature gives simplicity to the machine's control, with the coordinated production of products being assured by the coupling action of the mechanical transmission system.

In independent actuators machinery, the aim of introducing reprogrammability into the machine requires the replacement of fixed function actuators by ones having variable dynamic characteristics. For maximum flexibility throughout the manufacturing process, each function should be autonomous, with its own mechanism, actuator and dynamic controller, but such autonomy can only be reached if each actuator can repeatedly achieve the dynamic accuracy required by its function. In high speed intermittent machinery, these dynamic requirements impose exacting specifications on the drives and their controllers, so the design of the optimum servomechanism for each machine function is most important.

The servomechanism is the machine component which most affects the product being produced. It was shown in chapter 4 that the operation and interactions of sets of autonomous servomechanisms could be properly coordinated and monitored. The monitoring action of the APMs ensures that no faulty products are produced, but it can not ensure that good products are produced. If the servomechanisms can not perform within the manufacturing tolerances, then the APMs will reject the products which the machine is producing. Good products will only result if all servomechanisms in the machine can achieve their required performance specifications. The failure of any one servomechanism to perform within its tolerance band can result in the product being damaged.

This chapter investigates the design of flexible high performance servomechanisms for use in independent actuators machinery. Most particularly, the driving and controlling of intermittent machine functions is considered, with a discussion of recent developments in electric servodrive technology and the requirements for controllers which enable the new drives to actuate intermittent machine functions,

such as the third transfer mechanism and the arbor drum which are described in chapter 6.

5.2. The elements of an independent machine servomechanism.

Figure 5.1 is a representation of the elements of a machine servomechanism. The servo consists of the classical control system elements of controller and plant. Within the independent actuators scheme, the controller must handle the AFC functions which were introduced in chapter 4, while the plant is a servodrive and mechanism which performs the machine function.

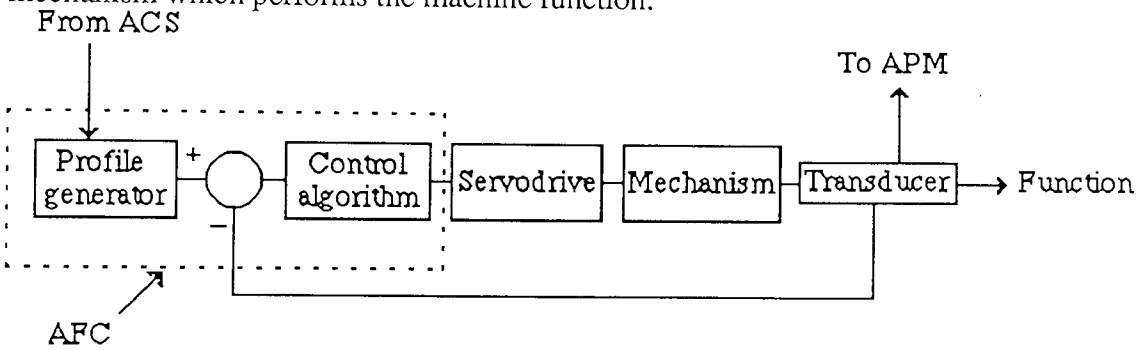


Figure 5.1. Independent actuators machine servomechanism.

The AFC receives the commence motion command from the ACS. It then controls the servodrive through a self–contained motion profile, with the control algorithms being developed to suit the requirements of the servodrive and load being controlled. The servodrive must be selected to be capable of driving the mechanical load through the required motion profile of the machine function. The configuration of the mechanism will depend upon the function being performed, but the controller and servodrive must be capable of operating within the manufacturing tolerance.

The development of servomechanisms for intermittent machine functions therefore requires the selection of the optimum servodrive for each function and the design of the optimum control algorithms for each drive.

5.3. The selection of servodrives for intermittent machine functions.

This section describes procedures which have been developed for the selection of servo drives to provide actuation for independent machine functions. The arbor drum and transfer slider are intermittent functions, so the procedures specifically address drive selection for intermittent and incremental motions. These are particularly arduous requirements for independent drives, since energy must be supplied both to accelerate and decelerate the motions.

A large amount of work has been published from the mid 1970's to the present date relating to the problem of drive selection for intermittent motions, most particularly in the proceedings of the annual symposiums on incremental motion control systems and devices, which have been running in Illinois, U.S.A. since 1972. However each selection procedure is based on a particular set of load or drive requirements. In 1973 Armin Jocz (22) defined high performance incremental motion requirements as being characterised by one or more of the following criteria:

{i} 300 cycles/minute for fractional power drives up to 6 per minute for 33.5 kw drives.
{ii} A duty cycle in excess of 30% of the operating cycle.
{iii} Positional accuracies greater than 95%
{iv} Torque to inertia ratios greater than 100,000 rad/sec ^{2}.
{v} Life greater than 10 million cycles.

The requirements for the arbor drum are 450 cycles per minute at peak instantaneous load power (acceleration . inertia . velocity) of 16 Kw, a duty cycle of 42% of the operating cycle, positional accuracies greater than 99.9% and peak load accelerations of 800 rad/sec ^{2}, while an operating life of 10 million cycles would give less than 16 days of continuous operation. This demonstrates the increase in performance requirements which have taken place since much of the incremental motion selection work was first published. For the class of machine functions considered in this project it was necessary to develop a selection procedure which is based on work which has been published for other classes of motions.

5.3.1. The development of the drive selection procedure.

As part of this research programme, a set of relationships has been developed which can be used to select the optimum electric servodrives for intermittent machine functions. The derivation of these relationships and their use as a drive selection procedure is described in Appendix 1. The selection procedure enables the optimum drive to be obtained on the basis of its capability to drive a specified load, the stiffness of the drive and its energy consumption. The application of the procedure to the selection of drives for the third tansfer mechanism and arbor drum is described in chapter 6.

The definition of the requirements for a drive to actuate a function will include the specification of a linear mass or angular inertia which must be moved and a motion profile which it must follow. The profile will include the peak and RMS accelerations and the peak velocity of the motion. These values can be used with

equations A.1.4 and A.1.12 for angular loads and with equations A.1.10 and A.1.14 for linear loads to define the fundamental requirements for a driver. Since each driver is considered to be coupled to the load through its own optimum ration, this enables different sizes of driver to be similarly evaluated for driving the load.

The two fundamental requirements of power rate and inertia:velocity can be combined in plots which indicate the driveability rating of motors. A representation of such a plot is shown in figure 5.2. The horizontal axis is the motor velocity:inertia rating and the vertical axis is the motor power rate. The ratings of the motors are represented by points on the plot. The location of each motor's point will be governed by its inertia and torque–speed performance. Generally, two driveability plots will be required for each application to cover the motors' peak and RMS power rate ratings.

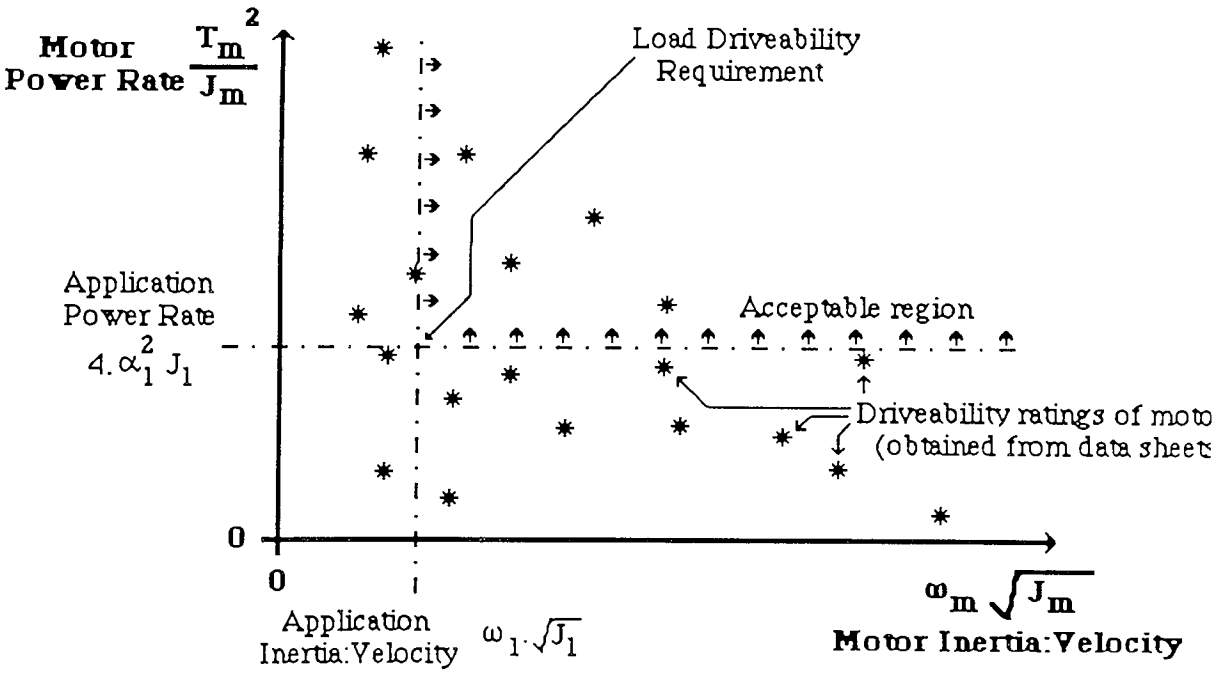


Figure 5.2. Representation of a driveability chart.

The suitability of motors for driving a given load may be quickly assertained from the driveability chart using the fundamental requirement equations A.1.4, A.1.10, A.1.12 and A.1.14. The mass, or inertia and peak velocity of the load can be used to plot the load requirement along the horizontal axis, with the value that is four times the power rate of the load being plotted against the vertical axis. An example of a load requirement is given in figure 5.2. The motors which are suitable for driving the load will be those whose points on the plot lie to the upper right of the load requirement. In the representation plot there are four motors shown whose driveability rating makes them suitable for driving the load.

5.3.2. High driveability rating servodrives.

The driveability rating enables different types of servodrive to be compared for assessing their suitability for independent actuator applications. Earlier research work at Aston University by Firoozian and Foster (10) performed a comparative evaluation of servomotor systems. It was found that for the high load inertia, high power rating applications considered in this research project, electrical motors produce a faster speed of response than hydraulic motors. Since that evaluation was performed, significant improvements have taken place in large electric motor technology, particularly in the high torque, low inertia brushless motors which are used in robotics and numerically controlled machine tool applications. It was these developments in electric motors which enabled the initiation of this research project, so further development of the selection procedure concentrated on determining which of the new motors best suited the particular applications being considered.

The new types of motors incorporate one or both of two improvements which have been developed during the 1970's and 1980's. The first of these is the use of high current, high speed switching transistors which are used to produce the switched current waveforms for brushless commutation. The use of feedback transducers mounted on the rotor enables the current waveform to be generated to give maximum motor efficiency with rotor position, while the absence of brushes further enhances commutation efficiency. The second improvement which has been introduced is the use of high coercivity rare–earth magnets. These enable small framed, low inertia motors to generate large torques, while the high demagnetisation current ratings of the magnets enable extremely high peak torques to be obtained. Samarium cobalt is the most widely used rare–earth magnet, but recently nedymium–iron–borun magnets have been introduced, which have greater coercivity, are more robust and are cheaper to produce (17).

The earliest applications of rare–earth magnets were in conventional direct current (d.c.) motors. Early samarium cobalt magnets were expensive, so the motors were mainly used for aerospace and defence applications, as well as in computer hardware. More recently though, rare–earth magnet d.c. motors have spread into wider use. There are many publications available which describe the construction and operation of d.c. motors. The book by Kenjo and Nagamori (14) includes explainations of d.c. motor and rare–earth magnet characteristics. The reader should refer to this for a more detailed understanding of the operation of the motors. The structure of a d.c. motor is shown in figure 5.3 (a). The rotor carries windings whose many ends terminate at a commutator, which is in effect a switch to control

the direction of current in the windings according to rotor position. The stator (the stationary part of the motor) carries the permanent magnets.

Since rare–earth magnets are of high coecivity, the rotor designs are different from those motors with ferrite magnets, so direct comparison between the two types of motors are difficult. However, for similarly sized motors with similar inertias, a samarium cobalt motor would generate 150% torque and 200% power of a ferrite motor, while the mechanical time constant of the samarium cobalt motor would be approximately 50% and the electrical time constant approximately 70% of that of the ferrite motor.

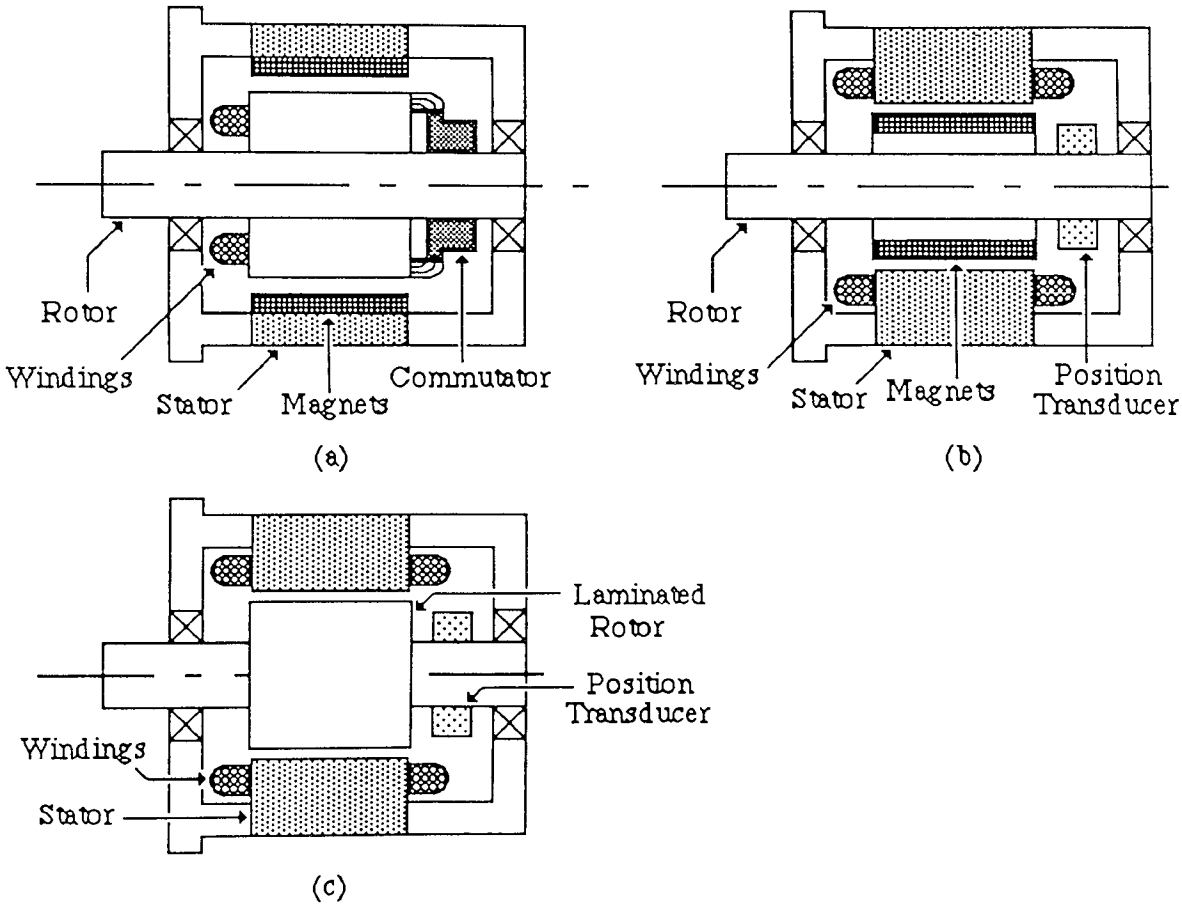


Figure 5.3. Servomotor construction

(a) d.c. motor

(b) brushless d.c. motor

(c) switched reluctance motor.

A type of motor which incorporates both rare–earth magnets and switched commutation is the brushless d.c. motor, which is shown in figure 5.3 (b). Here the permanent magnets are mounted on the rotor and the electric windings are mounted on the stator. The rotor mounted position transducer provides commutation switching information. Detailed descriptions of brushless d.c. motor

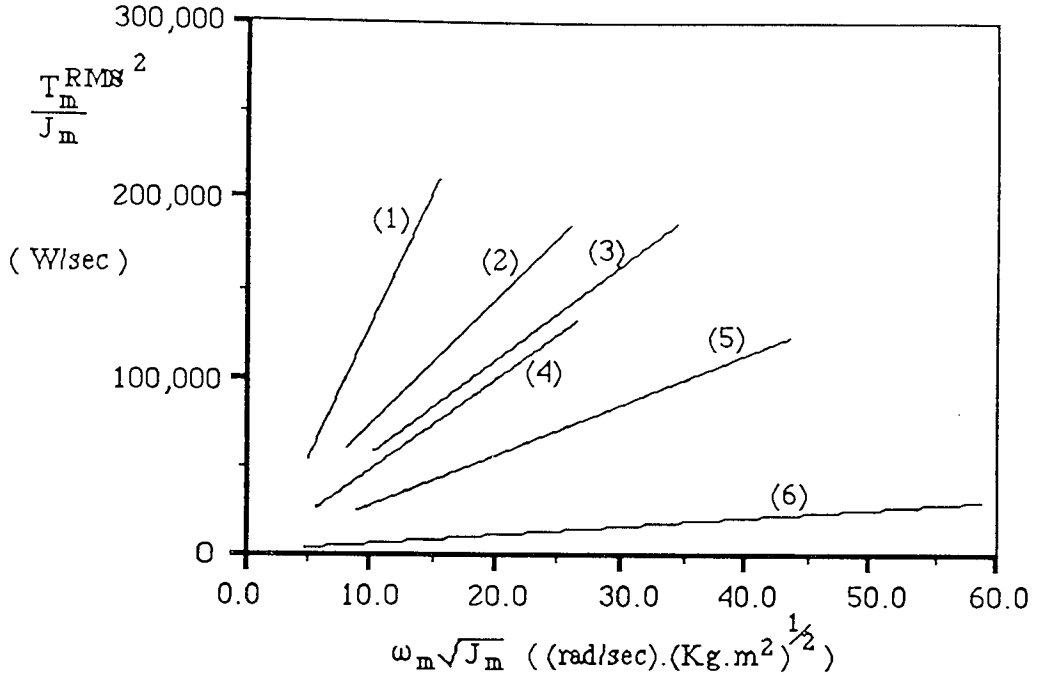
operation are contained in Kenjo and Nagamori's book (14) and in the book by Tal (18). By mounting the windings around the perimeter of the motor, very efficient cooling characteristics result. This is enhanced by the lack of commutation brushes, so very large peak currents can be tolerated.

Direct comparisons between brushless d.c. motors and traditional ferrite magnet d.c. motors can be misleading, because of the differences in their construction and operation. In addition, the improvement in torque performance is not linear through the size range of motors. However, some idea of the differences in performaces can be seen in two Electro–Craft motors which are in the University laboratories. The S–586 is a ferrite magnet d.c. motor of inertia 3.7 x 10^{-5} Kg.m^2 and volume 0.00146 m^3. This is of similar size to the S–3007 neodymium–iron–borun brushless d.c. motor of inertia 3.4 x 10^{-5} Kg.m^2 and volume 0.00121 m^3. The S–3007 generates a continuous torque which is approximately 400% that of the S–586, has a mechanical time constant which is only 23% of the S–586's and an electrical time constant which is 80% of the S–586's. Thus the brushless d.c. motor represents a significant increase in performance over traditional ferrite magnet designs.

A third type of motor which incorporates the new technologies is the switched reluctance motor, which is shown in figure 5.3.(c). Like the brushless d.c. motor, this incorporates stator mounted windings and switched commutation, however, the switched reluctance motor has no magnets. Instead it has a laminated rotor, which gives it a low inertia and a large power rate. Detailed descriptions of the construction and operation of switched reluctance drives are given by Lawrenson and others (13). Switched reluctance motors have larger volumes per rotor inertia than brushless d.c. motors. However, their power rate ratings are very similar to those of samarium cobalt brushless d.c. motors.

The driveability ratings of servodrives are a combination of their power rate, inertia and velocity characteristics. The types of motor described here are all available in high torque, medium speed or medium torque, high speed options. The selection of which option to use will be decided by the velocity:inertia rating of the application, given by equations A.1.12 and A.1.14. Figure 5.4 shows the average driveability ratings of the new types of servodrives. Each type of drive is represented by a line which is a best fit to the scatter produced by plotting the individual motors from the manufacturers' ranges. Each line is labelled with the year in which the range of drives was released for sale. The plot demonstrates the improvements in performance which have been obtained using the new drive technologies. The

brushless d.c. motors incorporating rare–earth magnets have the highest driveability ratings per size of any drive currently available.



Key to plots

(1) Electro-Craft Neodymium-Iron-Boron Brushless d.c. (1987)
(2) Inland Samarium Cobalt Brushless d.c. (1985)
(3) Tasc Switched Reluctance (1986)
(4) Moog Samarium Cobalt Brushless d.c. (1986)
(5) Contraves Samarium Cobalt d.c. (1983)
(6) Lucas Ferrite d.c. (1980)

Figure 5.4 Comparative driveability ratings of servomotors.

5.3.4. The suitability of brushless d.c. motors for intermittent independent actuators applications.

The high driveability ratings of brushless d.c. motors indicate that they can drive inertia loads at high rates of acceleration. In the development of the drive selection procedure, it was found that drives having small electrical and mechanical time constants are very stiff, while small mechanical time constants also give the drives low energy consumption properties. For high performance intermittent machine drives, large stiffness and low energy consumption are very important, so drives having both these properties will be most suitable.

The use of rare–earth magnets in motors gives them large torque constants, due to the magnets' high flux densities. In addition, the ability to orient the magnets radially gives the motors low inductances. In brushless d.c. motors, the magnets are mounted in the rotors, where their low densities and small sizes serve to give the motors low inertias. Examination of equation A.1.18 shows that small inductances,

high torque constants and low inertias give brushless d.c. motors very small electrical and mechanical time constants. This in turn gives the motors the high stiffness and low energy consumption properties which are required for intermittent machine drives.

In addition to their dynamic and energy consumption properties, brushless d.c. motors are favoured for processing machinery for several other reasons. They are very clean, having no exposed bearings or surfaces which have coatings of contaminants which might prove harmful to the manufactured products. The lack of brushes gives them low maintenance requirements and enables them to operate in potentially explosive atmospheres, which can exist when inflammable products are being processed. In addition, developments in neodymium–iron–boron magnet rotor designs have made the motors very much cheaper to produce.

A most important characteristic of brushless d.c. motors for all applications is the ease with which they can be controlled. The operating characteristics of the motors are the same as those of conventional d.c. motors, with the motor velocity being proportional to the reference voltage. Current feedback may be readily implemented to control the damping of the servo, while many types of inexpensive tachogenerators and position transducers are available for closing velocity and position loops using either hardware or software controllers.

5.4.1. The independent servomechanism controller requirements.

The drive selection procedure enables the optimum servomotor to be obtained for intermittent drives. For the servomotor to form part of an independent actuators machine, it requires a controller which is compatable with the control scheme developed in chapter 4 and which is capable of controlling the motion within the manufacturing tolerances. For intermittent motions, this requires accurate control of the actuator's position.

The controller's orientation in the independent actuators control scheme is shown in figure 4.6. The AFC receives actuator "go", "no–go" or "emergency stop" commands from the ACS. These commands ensure that its motion does not clash with that of another actuator. The AFC is required to contain the motion profile to be followed and the control algorithms, both of which are designed to suit the application and motor being controlled. The AFC employs feedback control, as shown in figure 5.1, to control the motor through the required profile after receiving the "go" command from the ACS. At the end of the motion, the AFC holds the motor in its dwell state until it next receives the command to move. The AFC must

also contain service routines to handle emergency stop commands if these occur. The minimum requirements for the controller can therefore be listed as:

(i) The ability to receive, interpret and act upon commands from the ACS.
(ii) The ability to store and read a motion profile.
(iii) The ability to interpret transducer signals from the actuator or load.
(iv) The capability to control the motion of the servomechanism through the motion profile within the manufacturing tolerances.
(v) The ability to store emergency stop routines and to act upon them when instructed.

In addition to these minimum requirements, there are additional properties which would enhance the controller's flexibility and suitability for application. For flexibility there must be the ability to alter the motion profile being followed. This could be attained either by introducing several self–contained profiles which the controller can access, or by profile generation algorithms which calculate the profile to be followed from a set of motion requirements.

For system development purposes, it is necessary for control variables to be accessible during operation. In analogue controllers this may be readily carried out by a development engineer using an oscilloscope or other form of analogue signal recorder, but in digital and software controllers, control variables such as required position or position error are not always accessible. The controller should therefore contain the ability to output control variables to an external device, such as a terminal screen or, via a digital to analogue convertor to a recorder.

An additional property which is important for the development engineer is the ability to tune the system on–line. This requires that controller constants such as gains or break frequencies be capable of being altered.

The additional controller requirements may therefore be listed as:

(vi) The ability to change the motion profile being followed.
(vii) The ability to output control variables to external devices.
(viii)The ability for on–line tuning to be performed.

5.4.2. The development of control algorithms for independent actuators.

The control algorithms must be developed to suit the actuator being controlled. Standardisation of algorithms would only be possible for similar actuators and for

optimum performance these would need to be tuned to suit the particular application being controlled. The actuator controller scheme developed in chapter 4 is not AFC algorithm dependent, which enables developments in adaptive, predictive or intelligent control methods to be incorporated into the machines if necessary.

In related S.P.P. research work at Liverpool Polytechnic and Birmingham University, detailed investigations are being performed relating to the design of control algorithms for brushless d.c. motors positioner systems. The object of the work performed in the author's research programme was to develop feedback control algorithms for use in the laboratory testing and in the simulation of multiple actuator systems. This did not include an in depth analysis of brushless d.c. motor characteristics and position control design methodology, since this was being performed elsewhere.

In appendix 2 the development of an 8086 microprocessor based controller for a Moog brushless d.c. motor is described, including the design of the control algorithms using a w–plane analysis. Unfortunately, due to reliability problems with the motor which are described in chapter 6, it was not possible to use the controller in the laboratory. The selection of alternative motors required that a new controller be obtained. The evaluation of control algorithms for the alternative motors is described in Appendix 3. The testing and implementation of the controller in the laboratory is described in chapter 6.

Chapter 6.

The Laboratory Tesing.

6.1 Introduction

This chapter describes work which has been carried out in the laboratory to assist the development and verification of methods for the design of independent actuator systems. The laboratory work, which has been carried out over a three year period, has been complimentary to the theoretical analysis and simulation work, with the different aspects of the research requiring feedback from each other to enable the project to be completed.

The given aim of the research project was to investigate methods for the control and synchronisation of independently driven intermittent mechanisms. Two intermittent cam mechanisms from the Molins S.P.1. packaging machine were considered with a view towards driving their motions independently. These were the incremental arbor drum and the third transfer slider. These are represented in figure 6.1.

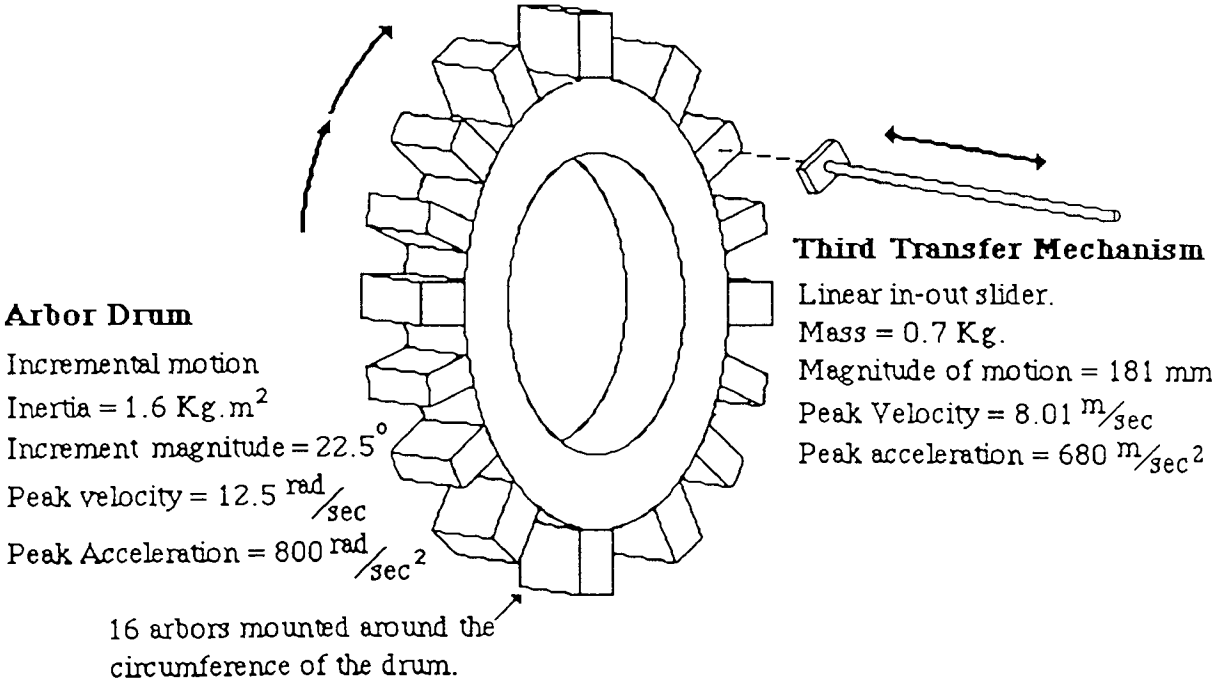


Figure 6.1. The third transfer slider and the arbor drum.

The two mechanisms were selected as being particularly arduous examples of their types. The arbor drum has 16 arbors mounted equispaced around its circumference. The drum is required to increment in steps and to remain stationary between steps. During the dwells in the drum's motions, the third transfer slider is required to move in-to and out-of one of the arbors. The acceleration and velocity values given

in figure 6.1 correspond to the machine's peak operating speed of 450 transfer and increment cycles per minute. It is required that the operating speed be variable up to this rate.

In the S.P.1. machine the drum is operated by a geneva cam mechanism and the slider by a cam and follower mechanism. The two mechanisims are coupled by a mechanical transmission system to a common drive motor, as is described in chapter 3. The drum was considered to be an arduous application for independent driving because of its large inertia and acceleration requirements, which are given in figure 6.1. The transfer slider has a mass–acceleration requirement that is less arduous than the arbor drum. However it is required to track to a specified position profile during part of its motion cycle. The slider has a maximum in–out stroke of 181mm, with a fully retracted position which is 33mm clear of the drum. Thus it penetrates the boundary of the drum by 148mm. The required position profiles of the slider and the drum are represented against a common time axis in figure 6.2. The time scale corresponds to a machine speed of 450 cycles per minute.

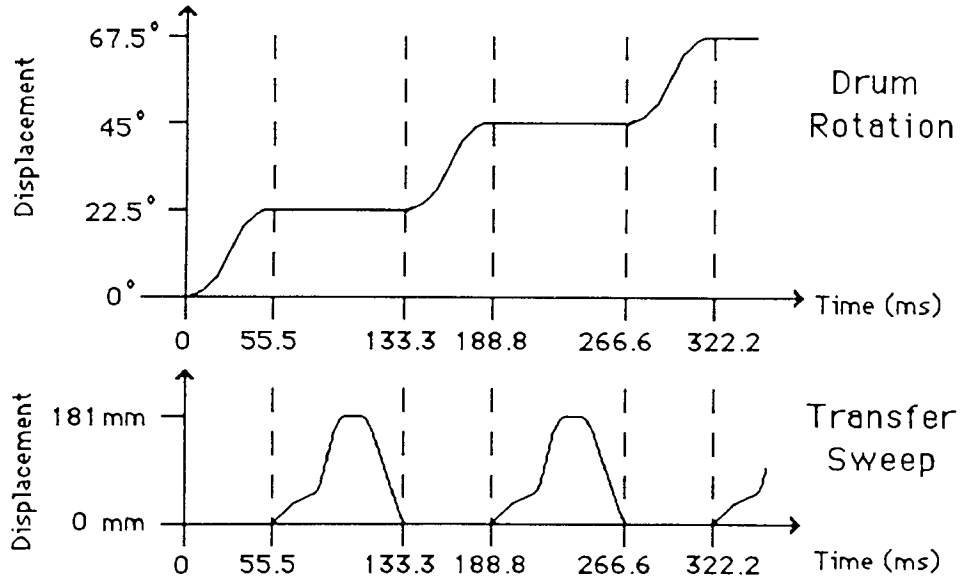


Figure 6.2. Required synchronised motion profiles of the third transfer slider and the arbor drum.

The requirement of the laboratory testing work was that the slider and the drum were required to be independently driven under software coordination and control, with the construction of the system being such that the operation of the two functions could be readily altered by reprogramming. The problems that needed to be solved to fulfill these requirements included:

(i) The selection of drives for the motions.

(ii) The design and development of mechanisms to enable the drives to drive the motions.

(iii) The development and implementation of algorithms for controlling the drives.

(iv) The development and implementation of methods for synchronising the motions.

The solution of these problems enabled the two motions to be independently actuated, which was the given aim of the laboratory work and was valuable in that it induced confidence in the independent actuators machine concept. An aspect of the work which had more practical importance was the verification that the simulation models described throughout this thesis were accurate representations of the hardware. These models were then fed back into the multiple actuator synchronisation work described in chapter 4, which was then fed back into the laboratory to enable the physical hardware systems to interact. By linking the theoretical and practical work in this way, it was possible to use the knowledge gained in independently driving two mechanisms to develop methods for independently driving complete machine systems.

6.2 Selecting and modelling drives for the two motions

The procedures which have been developed for selecting drives for intermittent motions are described in Appendix 1. The application of these procedures to the test rig development has been hampered by inaccurate characteristics and performance data supplied by manufacturers. The two motions to be driven are at the limits of the capabilities of current servo drive technology. As a result, the drives which have been selected for testing have utilised very new technology, with several having clearly been released for sale before sufficient development work has been carried out. The accompanying data sheets have been hurriedly put together and contain many errors. The under-developed nature of the drives has resulted in many failures occuring in the laboratory during testing, several of which have involved extensive damage to drive circuitry, because the high frequency, large inertia, large magnitude intermittent motions put demanding energy requirements on the drives.

During the three years of the test programme, the development of the test rig has provided a good insight into developments in brushless servo drive technology over the same period. Drives were initially selected and tested on the basis of manufacturers data sheet information, but these had to be rejected after tests found reliability problems and data sheet inaccuracies. Second generation drives were then

purchased, but extensive testing found these to be similarly unreliable. During this time, third generation drives had been released and these were then fitted to the test rig, but although these were more reliable than other drives, these too suffered from several drive failures and mechanical limitations.

This section of the chapter therefore catalogues the selection and modelling of drives, including details of the failures which occured and caused drives to be rejected and newer models to be investigated.

6.2.1 The initial selection of drives for the transfer slider.

The transfer slider performs an in-out linear transfer of 181mm stroke length. The exact motion proflie to be followed was not specified, but there is a requirement that at the point on the insertion stroke which is 67mm from the retracted position the slider's velocity must be less than 1m/sec and constant. In addition, the position tolerance at the fully inserted position is $\pm$ 0.2 mm. Since the exact profile to be followed was not specified, initial drive selection was based on the characteristics of the cam driven slider in the S.P.1 machine. At the peak required machine speed of 450 cycles/minute this gave load and acceleration requirements for the drive of:

Peak linear acceleration : 680 m/sec^2

RMS linear acceleration : 188 m/sec^2

Peak linear velocity : 8.10 m/sec

Tolerance at peak insertion position : $\pm$ 0.2 mm

Slider mass : 0.75 Kg

The nature of the in–out action of the transfer slider would make it suitable for driving with a linear actuator. However, there are no linear actuators with stroke lengths and performances in the ranges required which are capable of being servo controlled. Therefore the selection concentrated on high performance rotary servomotors, with the intention of using a mechanism to provide linear motion. Using equations A.1.10 and A.1.14 from Appendix 1 and the given requirements for the slider, the requirements for the drive are :

(i) Peak power rate $\frac{T_m^{P\,2}}{J_m} = 4.(680)^2.(0.75) = 1{,}387{,}200$ KW/sec

(ii) RMS power rate $\frac{T_m^{RMS\,2}}{J_m} = 4.(188)^2.(0.75) = 106{,}032$ KW/sec

(iii) Velocity.$\sqrt{\text{Inertia}} = (8.01).\sqrt{0.75} = 6.937$

In 1986, when the initial selection was made, the highest performance servo drives available were samarium cobalt magnet brushless d.c. drives incorporating trapezoidal switched commutaion current waveforms, as supplied by Inland, Moog

and Electro-craft (B.2000 series). The peak and RMS available power rates of these drives are plotted against their velocity : inertia rates in figures 6.3 and 6.4. Also plotted are the similar ratings for the Tasc switched reluctance drives and the Electro-craft Bru-500 brushless d.c. drives, which were not available at the time of the initial selection. The two graphs show the required power rates and velocity : inertia ratings of drive which are suitable for driving the transfer slider. Suitable drives for the slider are those which can be found in the upper right quadrant from the point where the requirement ratings cross, as described in chapter 5.

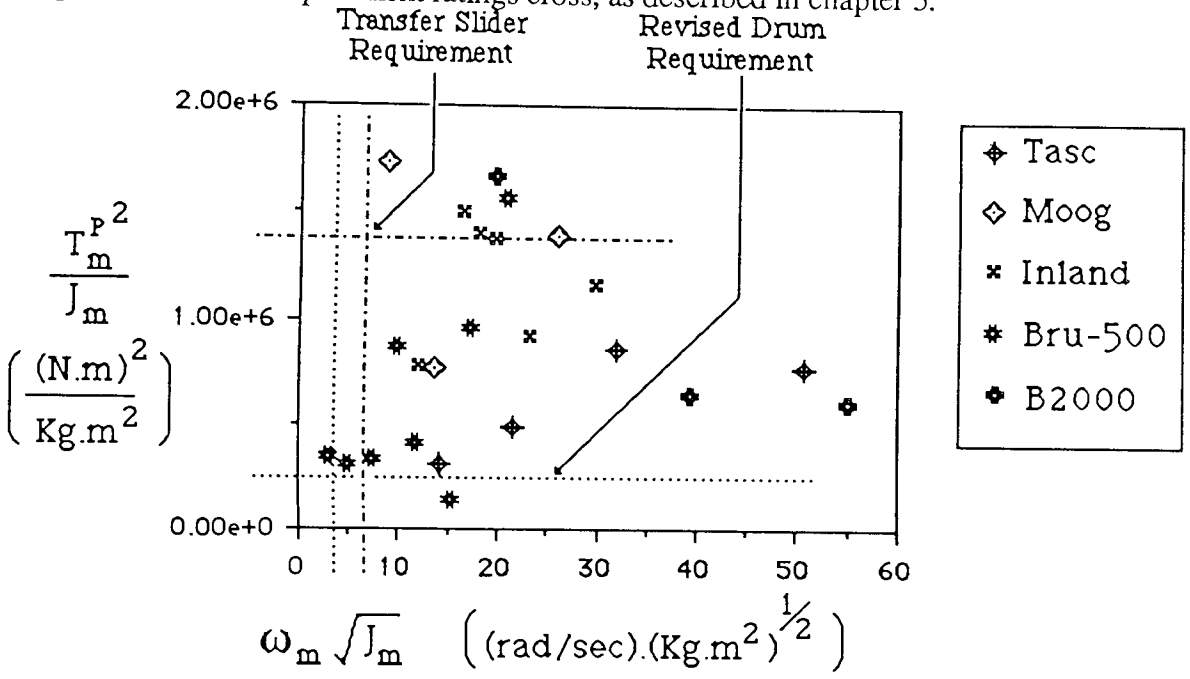


Figure 6.3. Available peak power rate ratings of brushless servo drives.

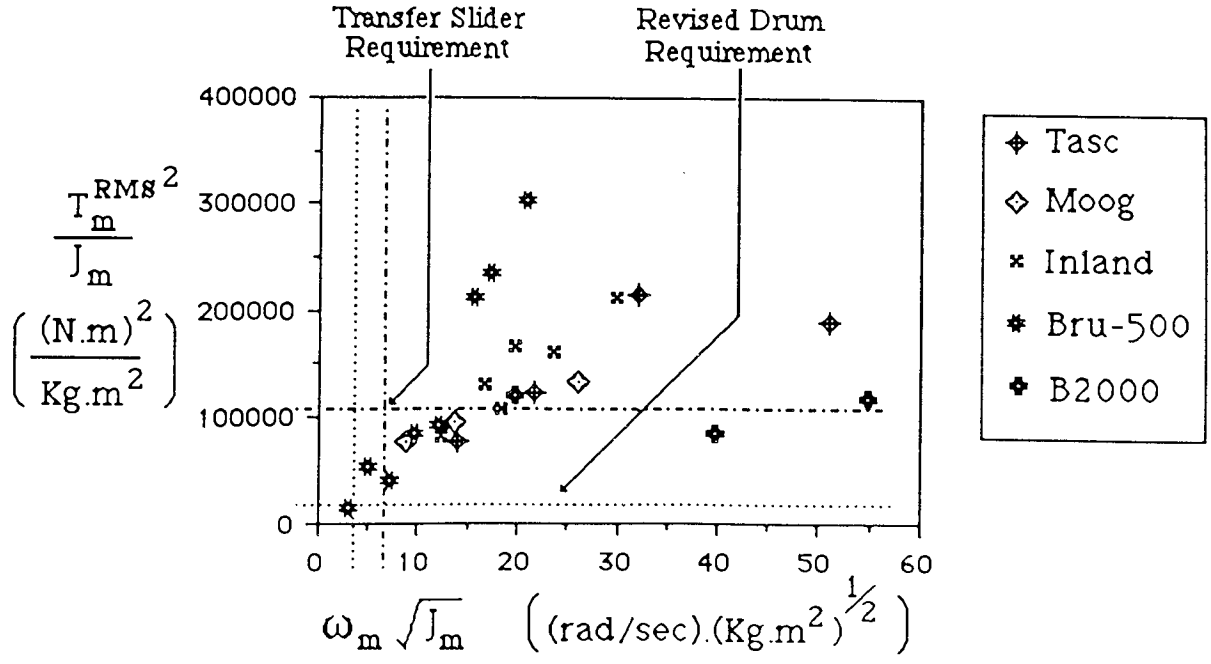


Figure 6.4. Available RMS power rate ratings of brushless servo drives.

The drives which fulfilled both the peak and RMS requirements were:

(1) Moog 306-023

(2) Electro-Craft B-2024

(3) Inland BHT 2204

(4) Inland BHT 3302

This gave four drives to be considered for the motion. However, testing work at Molins' Saunderton plant had found that the Inland drives had a poor reliability record. The analogue drive modules and the encoder were subject to repeated failures. In addition, the trapezoidal current waveforms for the commutation created large torque ripples, particularly at low speeds. Therefore it was decided that Inland drives were not suitable for independent drive applications.

At the time the selection was made, the Electro-craft B.2000 motors were also being tested at Molins. Initial tests had shown reliability to be better than Inland, but that torsional resonances could occur in the drive due to an insufficiently robust rotor. Therefore the Moog 306-023 motor was selected for the drive.

6.2.2. Modelling and testing the Moog 306-023 brushless d.c. servomotor.

The Moog brushless d.c. servomotor has a specification which indicates it to be very suitable for driving tightly controlled intermittent motions. It is a 14 pole motor, which indicates it should generate a low torque ripple. It incorporates a 4096 count absolute resolver for position sensing, with the absolute position data being output to a 12–bit parallel interface for communications with other devices. The motor's drive module has a specified peak current rating of 120 amps, which is greater than that of any other motor which was available at the time of its selection and is the reason it has a high peak power rate rating. The specification of low torque ripple, readily available absolute position information and high power rate ratings made the motor the optimum selection for the transfer slider drive.

The motor which was supplied was the first Moog brushless motor to be supplied in Europe. It was very much a development model and had very poor documentation and non–precise set up and operating instructions. The motor and controller's characteristics were specified as:

$J_m = 0.00848$ Kg.m ^{2}

$K_t = 0.893$ N.m /a

$\tau_m = 10.5$ msec

$\tau_e = 1.5$ msec

No values were given for R_a, L_a or K_b. These were interpreted from Tal's (18) brushless motor model to be:

$R_a = 0.154\ \Omega$

$L_a = 1.336$ mH

$K_b = 0.893$ Volts / (rad/sec)

The values for R_a & L_a were confirmed with a resistance–capacitance–inductance (R.C.L.) meter. The characteristics for the motor were combined with the model which Moog supplied for the controller to form the given servo model shown in figure 6.5. It should be noted that the given model does not include current feedback. Moog confirmed at the time that this is not employed in their drives.

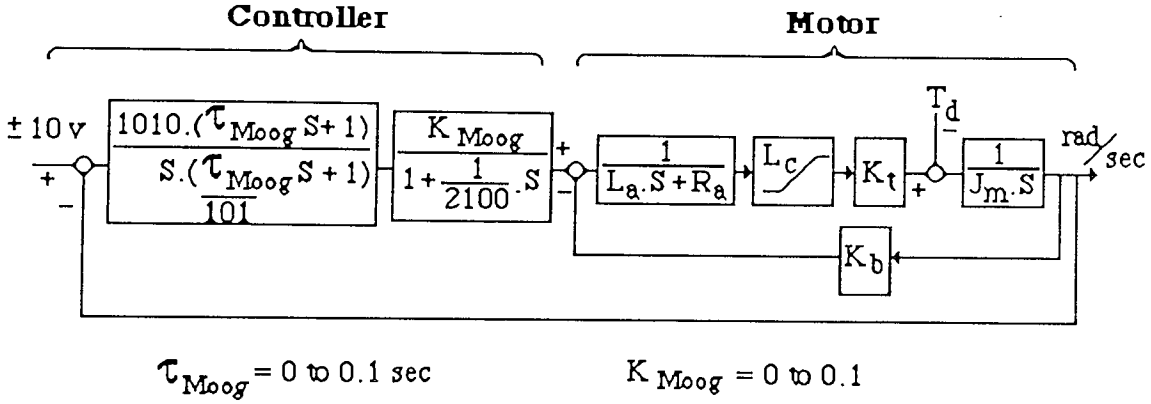


Figure 6.5 The Moog motor, given model.

The simulated frequency response of the given model driving an inertia load equal to the rotor inertia is plotted in figure 6.6. This was obtained using ACSL. Also plotted on the same axes is the frequency response of the actual system driving a similar inertia load,which was obtained in the laboratory. Both plots were obtained for an input demand signal of $^{+}_{-}$ 4.0 rad/sec and with velocity loop parameters K_{Moog} and τ_{Moog} set to mid range values. Examination of the plot shows that the simulation model is not a good representation of the actual system. Some differences were expected because the parameters K_{Moog} and τ_{Moog} on the actual system are set using 10–turn potentiometers, which give imprecise setting of given values. However, K_{Moog} and τ_{Moog} affect the gain peak in the region 200 rad/sec to 400 rad/sec and have little effect on the break frequency. So the differences between the simulated and actual responses in figure 6.6 were not due to this effect. The given model is just not a true representation of the system. In fact the higher bandwidth of the actual system indicates that shorter transient response settling will occur for the actual system than the Moog model predicts.

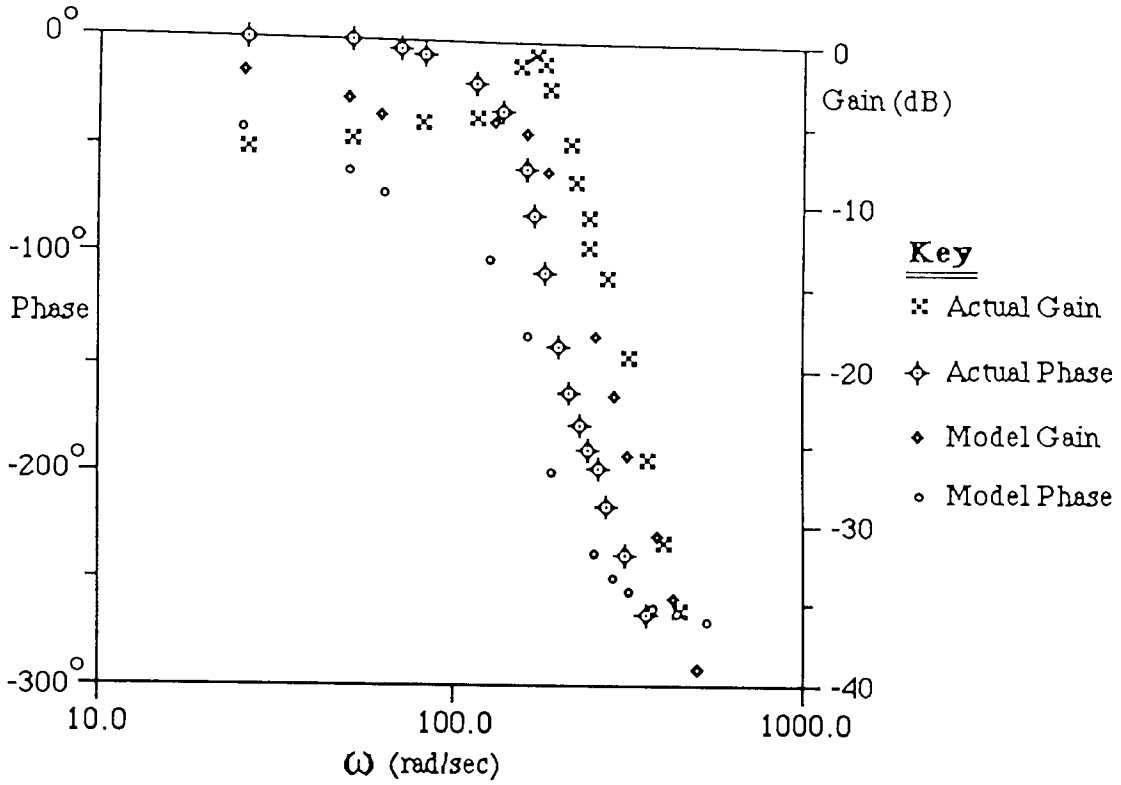
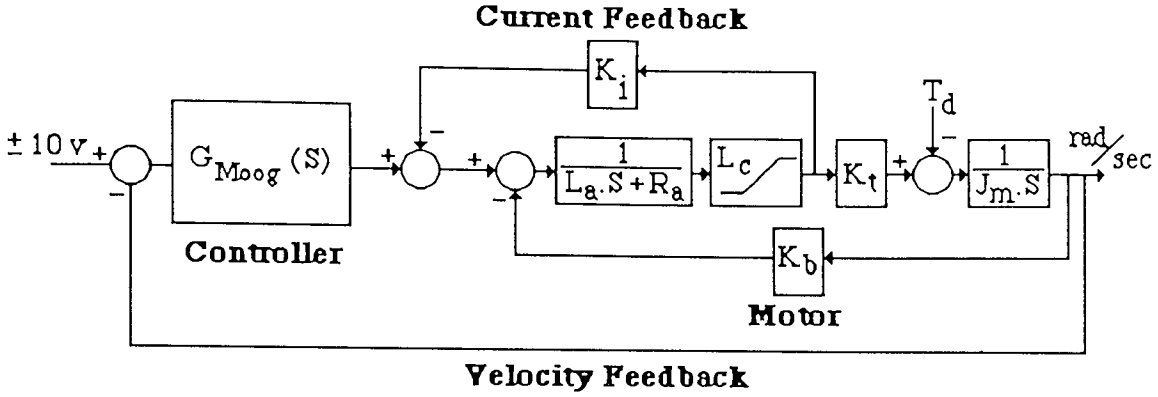


Figure 6.6 Frequency response of given model and actual system.

During early frequency response testing some indications arose of the poor construction of the motor drive circuitry. It was found that vibrating or lightly tapping the driver's housing would change the characteristics of the system. In addition, the system's response altered as the driver warmed up during use, due to inadequate temperature compensation circuitry. This made predicting the motor's performance very difficult. Finally, a small explosion and fire occured in the driver during a frequency response test. This was due to insufficient protection on the current drive circuitry. Frequency response tests can induce large current demands on the drive. The drive circuitry contained a 10 second time–out function to ensure that the R.M.S. rating was not exceeded, but no protection to ensure that peak current demands did not overstrain the current switching circuitry. Two of the high frequency current switching transistors failed due to peak current overload, which caused the fire. Peak current protection was introduced to the driver by mounting thermocouple switches directly onto the current switching transistors. These disabled the driver if the temperature of the transistors exceeded 80 ^{0} C.

It was most important that an accurate simulation model of the system be obtained. This was needed for the development of the software coupling algorithms for the slider and the drum. Communications with Moog determined that the controller model supplied was not the model of the driver. In addition, it was not known whether current feedback was employed by the driver and they were unwilling to

release detailed circuit diagrams or transfer function information. Therefore it was necessary to develop a new model based on the characteristics which were known to be correct. The model which was developed is shown in figure 6.7.



$$G_{Moog}(S) = \frac{K_1.(1+t1.S).(1+t2.S)}{(1+t3.S)(S^2+2.\zeta_1.\omega_{n1}.S+\omega_{n1}^2).(S^2+2.\zeta_2.\omega_{n2}.S+\omega_{n2}^2)}$$

Figure 6.7. The developed Moog servodriver simulation model.

The simulation model was developed using a best fit to the laboratory frequency response. This incorporates the model of the motor itself and a complex model of the controller, $G_{Moog}(S)$. Current feedback has been introduced to set the damping ratio of the motor, as described in section A.1.2.5.1. The controller function has five poles and two zeroes, which give the simulated system a similar response to that of the actual system in the laboratory. Figure 6.8 shows the simulated frequency response of this model, with the actual system response plotted on the same axes. The model gives a good match to the system through most of the frequency range, particularly in the frequency range up to the bandwidth. The high frequency phase error of the model did not affect the transient response match between the two systems.

Although the developed model was able to produce a response which matched that of the actual system, it did not contain any detailed information relating to the function of the driver. It was therefore considered that the model was insufficient to form the basis of multi actuator machine simulations and further model development was required. Unfortunately, soon after the preliminary model had been developed, several failures occured in the driver which led to the abondonment of the Moog system. These will be summarised in section 6.2.3. After the failures occured no further use was made of the Moog motor during the project.

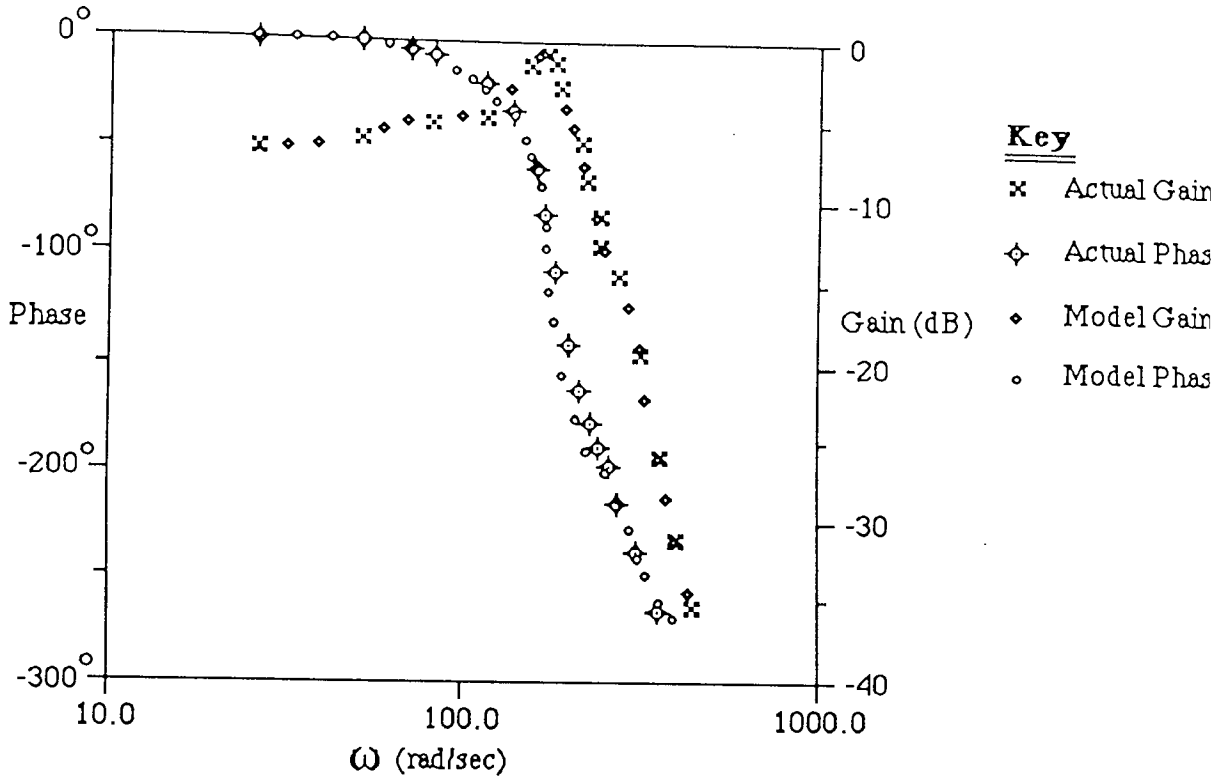


Figure 6.8. Developed model and actual system frequency responses.

6.2.3. Failures and subsequent abandonment of the Moog motor.

The destructive failure during frequency response testing which is described in section 6.2.2 was the first of several major and minor failures which were to occur in the system and cause it to be abandoned. This failure and subsequent limitations and failures of the system which occured over a 12 month period are listed below.

{i} Explosive failure of two current switching transistors in drive module during frequency response test due to current overload. Thermocouple switches fitted to avoid repeat failure.

{ii} Drive module characteristics drift with temperature and when module is tapped or vibrated.

{iii} No accurate detailed model available for system. Model required to be interpreted from laboratory tests.

{iv} Velocity loop control circuitry found to be analogue whereas manufacturers stated it to be digital.

{v} Failure of parallel I / O port in drive module. Replacement required.

{vi} Failure of power supply module causing repeated blowing of fuses under low load conditions. Replacement required.

{vii} Explosive failure occured in replacement power supply. Believed to be due to build up of residue on insufficiently robust terminals. Possibly similar reason for failure of earlier power supply unit.

{viii} Explosive failure of drive module causing total write–off of unit. Cause unknown but possibly related to earlier failures in power supply module.

After the write–off of the drive module, the replacement cost was found to be greater than the cost of the newer Electro-Craft Bru–500 drives. Therefore it was decided not to replace the drive module of the unreliable Moog system, but instead to purchase a Bru-500 system to drive the third transfer slider.

6.2.4. Selection of a Bru–500 motor for the transfer slider.

During the 12 months between supply and final destruction of the Moog system, several newer drives became available. One of these was the Bru-500 range produced by Electro–Craft. Initial tests by Molins' Saunderton division had found these to have improved reliability over the Inland and Moog drives, with similarly improved documentation, modelling information and drive electronics. Therefore it was decided to investigate the suitability of these drives for intermittent motions.

The Bru-500 servo drives have high coercivity neodymium iron borun magnet rotors. Position sensing is carried out by an 8000 count incremental encoder. Motors are available in 4,6 and 8 pole versions, with the switched commutation using a sinusoidal waveform which increases commutation efficiency and reduces torque ripple. Velocity and current control algorithms are performed by software and several algorithms are available.

The specified power rate and velocity·: inertia characteristics of the Bru–500 drives in 1987, at the time of the second selection of transfer slider drive, are listed in table 6.1. Comparison of this with the requirements for the transfer slider given in section 6.2.1 shows that there were three Bru–500 drives which satisfy the initial selection requirements. In Appendix 1 it is shown that the drive which will use the least energy for the motion will be that having the smallest mechanical time constant, while the drive which will be expected to give the stiffest response will be the one having the smallest value of the product of its electrical and mechanical time constants. For both cases the S–4050 motor was the optimum selection. It was also

the least expensive of the three drives at that time. Therefore this was selected for the transfer slider drive.

Drive	Peak Power Rate	R.M.S. Power Rate	$\omega_m \sqrt{J_m}$
S-3007	282,647	18,356	3.05
S-3016	544,444	53,778	4.97
S-4030	1,098,286	87,500	4.96
S-4050	2,542,042	155,042	7.01
S-6100	686,429	106,314	11.76
S-6200	2,023,158	290,658	13.7
S-8350	1,324,413	297,794	26.85
S-8500	2,160,900	384,400	20.94

Table 6.1. 1987 specified performance of Bru-500 drives.

6.2.5. Modelling and testing the Electro-Craft S–4050 motor.

The S–4050 brushless servomotor was supplied with a DM–50 velocity drive module and PSM–50 power supply module. The characteristics of the motor were specified as:

$J_m = 2.4 \times 10^{-4}$ Kg.m ^{2}

$K_t = 0.5$ N.m /a

$R_a = 0.8\ \Omega$

$L_a = 5.5$ mH

$K_b = 0.57$ Volts / (rad/sec)

Figure 6.9 shows the reduced system model supplied by Electro-Craft for the motor and driver.

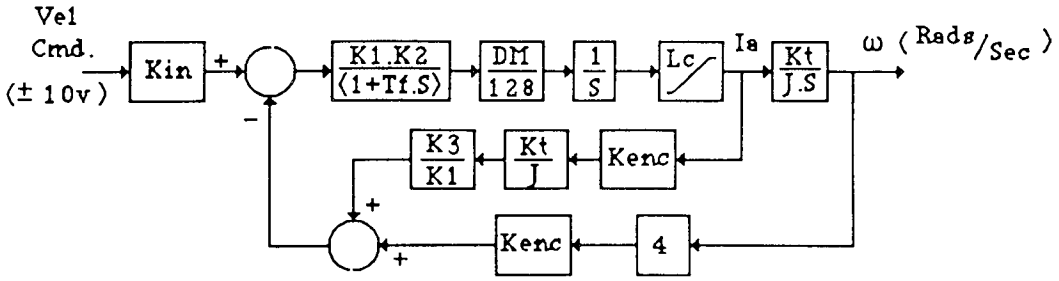


Fig.6.9 Bru-500 Velocity Regulator: Given Reduced Model.

Kin = 0.2667 . (No. of Encoder Lines) . ((Rev/min)/volt)

Kenc = 2 . (No. of Encoder Lines) / π

K1 = (Band . G1) / (Damp . 65536)

K2 = (Band . Damp . G2) / 16777216

K3 = 0.001

DM = Drive Module (25, 50, 100 or 150)

Kt = Torque Constant (Nm / a)

J = System Intertia

$Tf = 1 / (2 . \pi . FBW)$

$G1 = 2376$

$G2 = 3775$

Band, Damp and FBW are set interactively from the terminal. To understand the effect that these have, it is useful to consider the acceleration and velocity loops separately. Figure 6.10 shows the given velocity regulator model reconfigured to highlight the two loops.

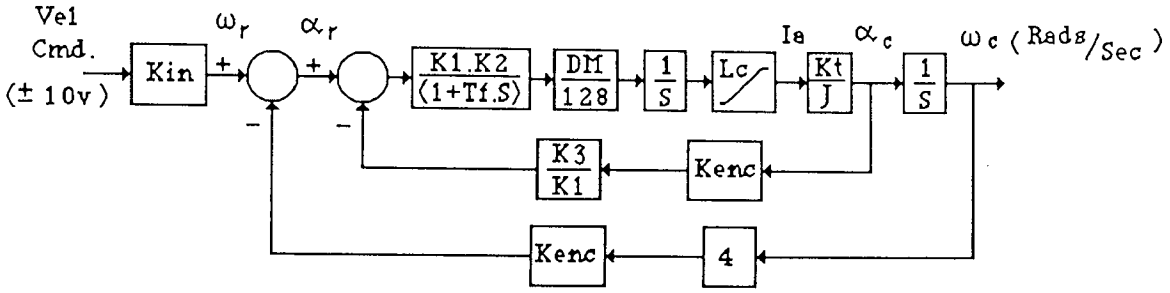


Figure 6.10. Bru-500 acceleration and velocity loops.

The controller characteristics are functions of the acceleration feedback loop. The closed loop transfer function of the acceleration loop is

$$\frac{\alpha_c}{\alpha_r} = \frac{\left\{\frac{2.\pi . G1 . G2 . DM . K_t}{(128).(J).(65536).(16777216)}\right\}.BAND^2.FBW}{S^2 + 2.\pi.FBW.S + \left\{\frac{2.\pi .G2 .DM .K_t.Kenc.K3}{(128).(J).(16777216)}\right\}.BAND .DAMP .FBW}$$

This can be related to the classical second order system transfer function:

$$\frac{\alpha_c}{\alpha_r} = \frac{K_{system} .\omega_n^2}{S^2 + 2.\zeta.\omega_n.S + \omega_n^2}$$

which gives:

$$\omega_n = \sqrt{\left\{\frac{2.\pi.G2.DM.K_t.Kenc.K3}{(128).(J).(16777216)}\right\}.BAND .DAMP .FBW}$$

$$\zeta = \frac{\pi .FBW}{\sqrt{\left\{\frac{2.\pi.G2.DM.K_t.Kenc.K3}{(128).(J).(16777216)}\right\}} . BAND .DAMP .FBW}$$

$$K_{system} = \left\{\frac{G1}{(Kenc).(K3).(65536)}\right\} . \frac{BAND}{DAMP}$$

Thus the relationships between the controller parameters BAND, DAMP, FBW and the classical second order system parameters ω_n, ζ and K_{system} are:

(i) BAND: Proportional to ω_n^2
Proportional to K_{system}
Inversely proportional to ζ^2

(ii) DAMP: Proportional to ω_n^2
Inversely proportional to K_{system}
Inversely proportional to ζ^2

(iii) FBW: Proportional to ω_n^2
No effect on K_{system}
Proportional to ζ^2

Thus BAND can be related to the system bandwidth, since increasing BAND increases ω_n and reduces ζ. However BAND should be increased with caution, because the reduction in ζ is accompanied by an increase in the current loop gain. DAMP can be related to the system damping, because although an increase in DAMP reduces the current loop damping ratio, it also reduces the gain. FBW has a similar effect on both ζ and ω_n. It can be used to filter out high frequency noise. If noise in the system requires FBW to be reduced, the loss of high frequency gain is compensated for by the reduction in acceleration loop damping, which at lower frequencies increases the gain and reduces the phase shift.

Bru-500 systems are delivered with default controller settings of BAND=50, DAMP=50, FBW=300. The constants G1 and G2 are pre–set according to the drive module and motor configuration being run. These settings give the system an underdamped velocity response under no–load conditions. The addition of a load reduces ω_n and increases ζ, so BAND and DAMP must be increased to compensate for this. These should both be given similar values to maintain a constant gain. Finally, if noise is present in the system, FBW can be reduced to compensate.

Early testing of the S–4050 Bru-500 system identified discrepancies between the performances of the actual system and the simulated given model. Meetings with Electro-Craft determined two main causes for this. The principal problem was that the supplied motor's characteristics were not as specified in the literature. Early Bru–500 motors suffered from torsional vibrations due to insufficiently stiff rotors, so the entire range had been fitted with reinforced rotors. The supplied motor had an inertia:

$$J_m = 5.6 \times 10^{-4} \text{ Kg.m}^2$$

This change had a significant effect on the suitability of the motor for the transfer slider, since the power rate ratings of the drive were halved. Thus the supplied drive

was not capable of driving the transfer slider at peak machine speeds. However, the lack of alternative drives and the need to complete the project within a fixed time frame led to the decision to continue the work using the S–4050 motor.

It should be noted that the Bru–500 ratings used in the production of figures 6.3 and 6.4 are May 1988 values for drives having reinforced rotors.

In addition to the inertia change, the second improvement to the given model was the inclusion of a first order pole to represent the current loop lag and two zero order hold functions to represent the software sampling action of the controller. The improved Bru–500 model is shown in figure 6.11.

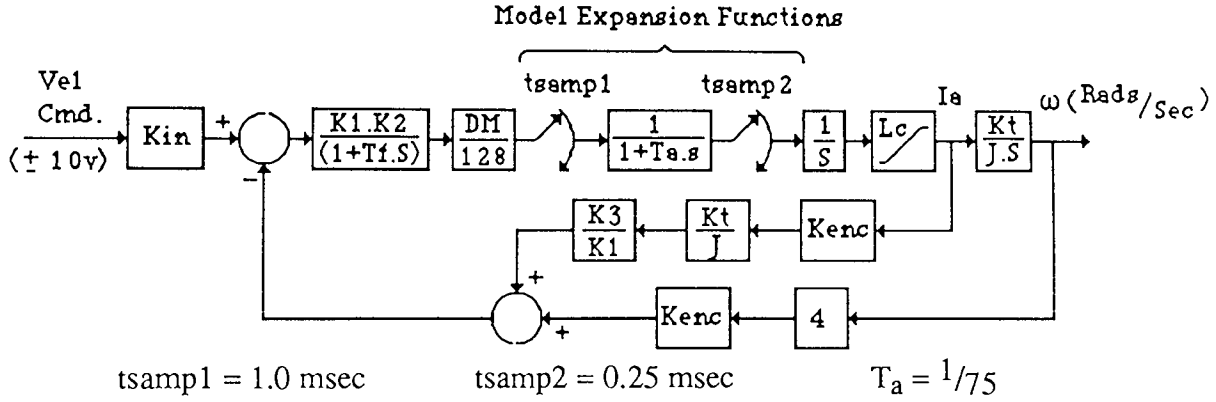
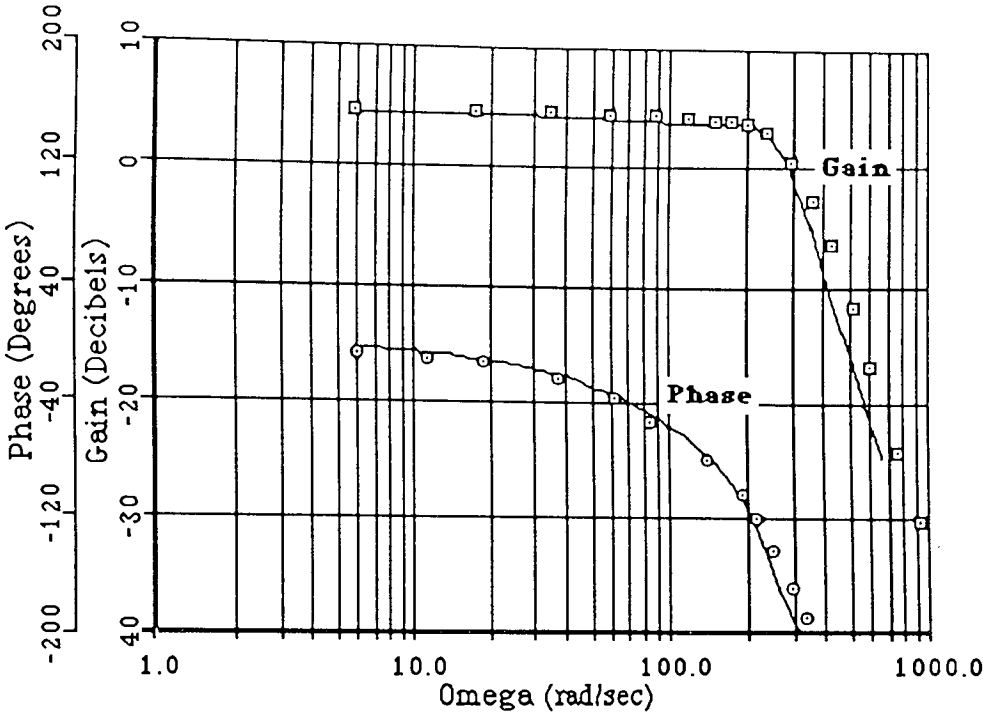


Figure 6.11 Bru–500 velocity regulator, expanded model.

The simulated frequency response of the S–4050 expanded model is shown in figure 6.12. Also plotted on the same set of axes is the actual frequency response obtained in the laboratory. Both plots are for drives fitted with an inertial load of 5.6 x 10^{-4} Kg.m^2, with an input signal equivalent to $^{+}\_$ 21 rad/sec superimposed onto an offset velocity of 40 rad/sec and controller settings BAND=60, DAMP=70, FBW=300. The –90 ^{0} phase crossover occurs at approximately 175 rad/sec, while the –3 dB gain occurs at approximately 275 rad/sec. It is often argued that the -3 dB gain point represents the bandwidth of the system. However, for the design of high frequency acting servos the useful bandwidth should be taken as the lower of the two values, because a large phase lag makes it difficult to achieve a stiff servo when a feedback loop is closed about the system.The model is a good representation of the system in the frequency band up to 340 rad/sec, with a small divergence between the responses occuring at higher frequencies.



Key

Simulated Phase and Gain (Labelled on Plot)

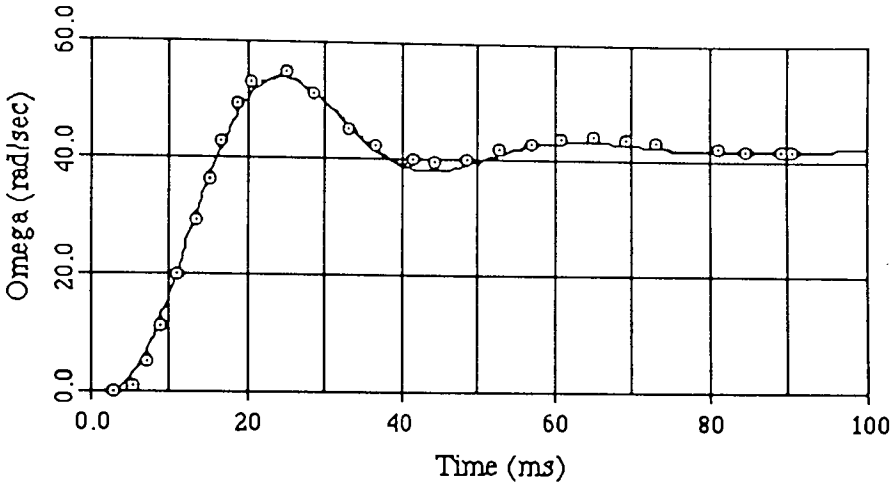
System Gain

System Phase

Figure 6.12 S–4050 drive, simulated and actual frequency responses.

The transient responses of the actual system and the model are shown in figure 6.13. In this plot, both systems had similar matched inertias fitted to those used in the frequency reponse tests, but the default controller settings of BAND=50, DAMP=50, FBW=300 were used. It can be seen that the simulated response is a very close reproduction of the actual system reponse, with a discrepancy between the two response which is at all time less than 2.5%. However, the simulation model is a simplified single input single output reproduction of the six–pole motor system. The actual system in the laboratory has variable characteristics with angular position, due to the pole configuration in the motor. This is demonstrated in the transient response plots of figure 6.14. The two responses are both for the motor driving the matched inertia in response to a demand signal of 400 rev/min from a stationary start. The controller settings for the two responses were the same, with BAND = 80, DAMP = 80, FBW = 300. However the initial position of the drive for response 1 was displaced by 60^{0} from that of response 2. The peak overshoot of response 1 is 11.5% greater than that of response 2. This is the result of the variable motor characteristics with rotor position, which are caused by the winding configuration of the motor. The pattern of large and small overshoots varies with

the number of pole pairs of the motor. Thus for the six–pole S–4050 motor there are six equispaced positions which alternately give maximum and minimum overshoots to the fixed demand signal. The two reponses in figure 6.14 represent the maximum and minimum response overshoots which the system gives under the conditions described.



Key

—— Simulated Response

o o Actual Response

Figure 6.13 S–4050 motor, simulated and actual transient reponses.

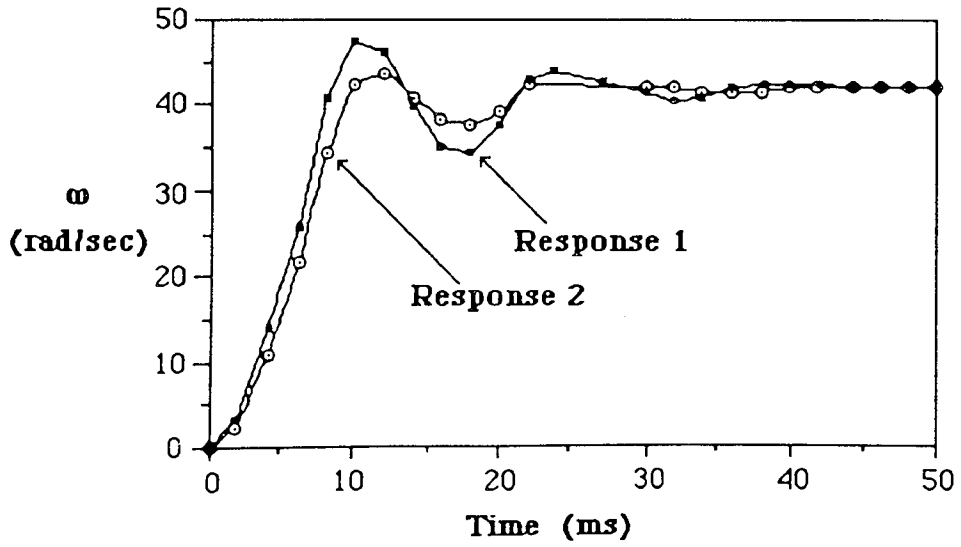


Figure 6.14 S-4050 motor laboratory responses showing variable characteristics with position.

A detailed investigation of the operation and modelling of the Bru–500 system is being carried out in the Department of Electrical and Electronic Engineering at Aston as part of the second Aston/Molins S.E.R.C. S.P.P. research project . This includes examination of the variable characteristics with position problem. It was

decided that to avoid replication of work, detailed multi–pole modelling of the motor would not be carried out as part of this project. Instead the expanded model of figure 6.1.1 would be used for further research and development work. The aim of the multi actuators system work was to develop methods for the coupling and control of independent actuator systems, not specifically Bru-500 systems. The expanded model's reponse has an overshoot which is mid way between the maximum and minimum overshoots which the actual system gives for similar controller settings. It was therefore felt that the expanded model was sufficient for further research work.

6.2.6. The selection of a drive for the arbor drum.

The arbor drum is required to increment in 22.5 ^{0} steps, with pauses between increments. The exact motion profile of the drum was not specified, but a position tolerance was given for the stationary dwell periods. Initial analysis of the system for drive selection was based on the characteristics of the cam driven drum in the S.P.1 machine, which has a modified sinusoid motion profile. At peak machine speeds, this gave load and acceleration requirements of:

Peak angular acceleration: 800 rad/sec ^{2}

RMS angular acceleration: 210 rad/sec ^{2}

Peak angular velocity: 12.44 rad/sec

Drum inertia: 1.6 Kg.m ^{2}

Tolerance during dwells: $\pm$ 0.05 mm at 336mm radius.

It should be noted that the tight position tolerance is not met by the cam driven drum on the SP1 machine. A backlash of approximately $\pm$ 0.3 mm at 336mm radius was measured in an arbor drum mechanism at Molins' Deptford division early in the research project.

Using equations A.1.4 and A.1.12 from Appendix 1 and the given requirements for the drum, the requirements for the drive are:

(i) Peak power rate $\frac{T_m^{P\,2}}{J_m} = 4.(800)^2.(1.6) = 4{,}096{,}000$ KW/sec

(ii) RMS power rate $\frac{T_m^{RMS\,2}}{J_m} = 4.(210)^2.(1.6) = 282{,}240$ KW/sec

(iii) Velocity.$\sqrt{\text{Inertia}} = (12.44).\sqrt{1.6} = 15.74$

Extensive searches for drives found these requirements to be beyond the capabilities of drive technology at that time. At the time of writing this report, three years later, these requirements are still beyond the capabilities of available drives. However,

developments in drive technology are continually improving the available performance. Table 6.1 shows that the Electro–Craft S–8500 drive, which has not required a reinforced rotor, is capable of meeting the RMS requirement for the drum. A 200 amp peak current drive module for this drive is expected to be available during 1989. This will give the drive a peak power rate rating of over 4,000,000 KW/sec. Therefore it is expected that a suitable drive for the drum will be available soon after completion of the project.

It was decided to continue the arbor drum work with a low inertia model of the drum. A full scale model of the drum was designed. This had an inertia of 0.1 Kg.m ^{2}, which gave revised power rate and velocity requirements for the drive of:

(i) Peak power rate $\frac{T_m^{p\,2}}{J_m} = 4.(800)^2.(0.1) = 256{,}000$ KW/sec

(ii) RMS power rate $\frac{T_m^{RMS\,2}}{J_m} = 4.(210)^2.(0.1) = 17{,}640$ KW/sec

(iii) Velocity.$\sqrt{\text{Inertia}} = (12.44).\sqrt{0.1} = 3.94$

These revised requirements for the drive are plotted in figures 6.3 and 6.4. This demonstrates that a wide range of brushless servomotors were suitable for driving the model drum. The failures of the Moog motor and the succesful early testing and modelling of the Bru–500 motor led to Bru–500's being selected for the drive. The drum and slider test rig had by this time been designed to suit the Moog motor, so it was necessary to adapt the rig to suit the Bru–500's. To simplify this it was decided to obtain a Bru–500 which was capable of driving the drum by direct drive, without employing a coupling ratio. This gave three suitable drives, of which the S-6200 had the smallest mechanical and electrical time constants, which indicated it would give the stiffest response and use the least energy. Therefore the S–6200 motor was selected for driving the model drum.

6.2.7 Modelling and testing the S–6200 motor.

The S–6200 brushless servomotor was supplied with a DM–100 drive module and PSM–125 power supply module. The characteristics of the supplied motor were:

$J_m = 0.0024$ Kg.m ^{2}
$K_t = 0.62$ N.m /a
$R_a = 0.18\ \Omega$
$L_a = 2.4$ mH
$K_b = 0.71$ Volts / (rad/sec)

The drive is essentially a larger, 8–pole version of the S–4050 drive, employing similar motor construction and drive circuitry. The characteristics of the S–6200 drive can be represented by the expanded model of figure 6.11. This requires changing the motor characteristics K_t ,J_m and the drive module constants G1 and G2 to those of the DM–100:

$$G1 = 2145$$
$$G2 = 4876$$

In addition, the current limit, which for the DM–50 module is 50a, must be changed to the DM–100's peak limit, L_c= 100 a.

The S–6200 system has a similar variable response with position characteristic as the S–4050. However, being an 8 pole motor there are 4 equispaced peaks and troughs in the overshoot vs position characteristic. As with the S–4050 drive, the expanded model gives a response which is mid way between the maximum and minumum overshoots obtained by the actual system for simular controller settings. Since detailed analysis of both 6 pole and 8 pole drives was being performed by the second Aston/Molins S.E.R.C. S.P.P. research project, it was decided that the expanded single degree of freedom model of the S–6200 system would be sufficiently accurate for further multi actuators system development work.

6.3 Designing the slider and drum mechanisms.

The slider and drum mechanisms were designed to provide the capability to test and prove the synchronisation of two intermittent actuators. The two mechanisms were designed to run with no spatial overlap, or to be moved together so that overlap occurs. This facility enabled extensive testing of the synchronisation software to be carried out under safe conditions, in which loss of synchronisation would not cause physical contacts to occur. After testing with no overlap was completed, the two mechanisms were then moved together to demonstrate that the system operated succesfully.

The two mechanisms were each originally designed to suit the optimum drive requirements of the Moog 306–023 brushless servomotor, which was under test at the University at that time. After the destruction of the Moog motor and adoption of Bru–500's, it was necessary to adapt the mechanisms to suit the new motors. This required compromises to be made between the optimum requirements for the new drives and the physical construction of the parts which had been built. The development of the two mechanisms is described below, including the optimum design for the Moog motor and the adaptations made to suit the Bru–500's.

6.3.1. The transfer slider mechanism.

The slider mechanism was designed to enable the rotary motion of the motor to perform the linear in out transfer function. In intermittent synchronised systems, the decision to move is asynchronous, with the time of commencement of each cycle not being temporally regular. Therefore it was not possible to use a slider–cank type mechanism, in which the motor is in continuous motion, because it is necessary to have the ability to start or stop the motion at any time. In addition, the slider's action must be reprogrammable through software. A mechanism which had been specifically designed for a given stroke length or profile would not have the inherent flexibility to allow functional changes to be readily implemented. Finally, for ease of control of the system, it was desirable to have a mechanism in which the inertia and ratio were constant with position. This would enable the optimum drive conditions to be maintained throughout the slider's stroke.

Figure 6.15 shows the arrangement of the transfer slider mechanism.

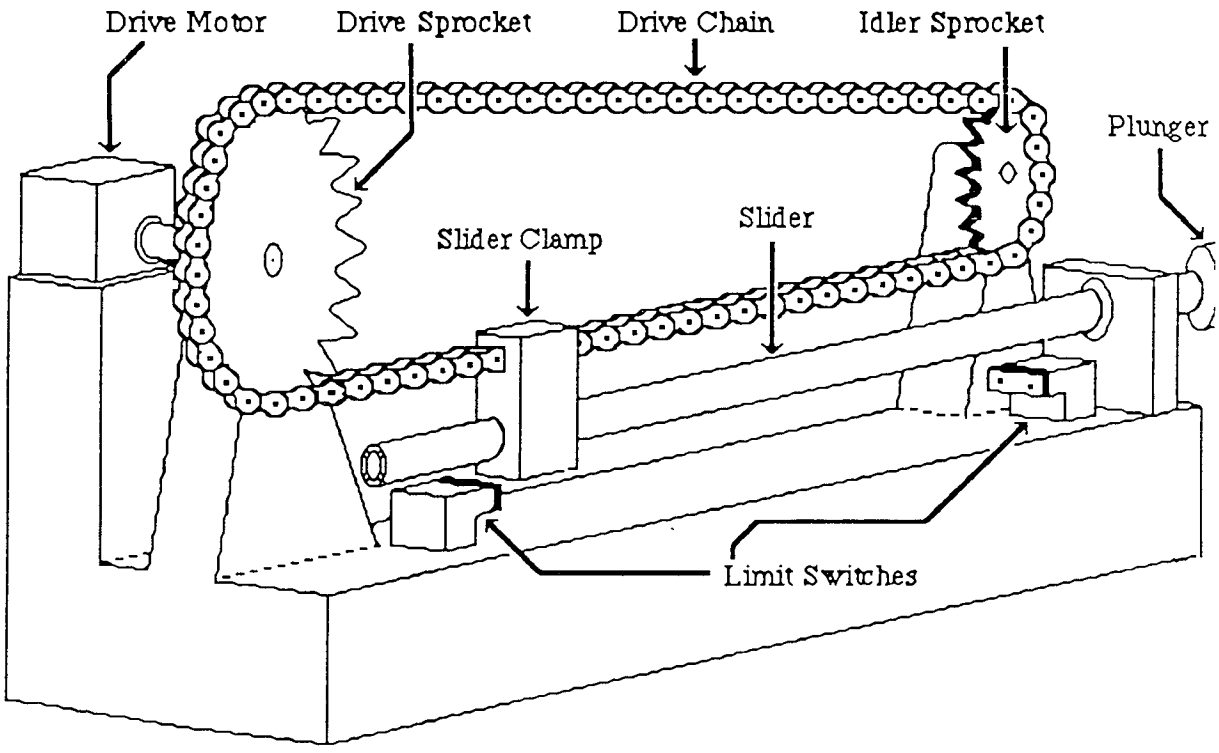


Figure 6.15 The transfer slider test assembly.

The transfer slider mechanisms has a straight forward design. For ease of development, the slider was driven by the motor using a bicycle chain and sprockets. This gave flexibility because replacement parts were readily available, including a good range of sprockets giving a choice of drive ratios at low cost. The motor drives the mechanism through the drive sprocket. The slider is clamped to the lower section of the chain, with the chain and slider being aligned so that they are

parallel. Electrical contact switches are used to ensure that the slider does not overshoot it's 200mm working region and collide with the sprockets.The slider passes through a plane bearing and a plunger is mounted on the end of the slider which moves in the slider–drum overlap region.

For the selection of the optimum drive sprocket radius, it was required that the mass of all moving parts at the radius of the slider be known. These were:

{1} Slider shaft + plunger: 0.29 Kg

{2} Chain, length 910mm: 0.35 Kg

{3} Slider clamp assembly: 0.1 Kg

From equation A.1.8, the optimum drive radius for the Moog 306–023 servomotor was therefore:

$$R = \sqrt{\frac{(0.00848)}{(0.29 + 0.35 + 0.1)}} = 107 \text{ mm}$$

A drive sprocket of radius 105 mm was used for the mechanism, After failure of the Moog motor the mechanism had been partially constructed, so it was decided to complete the assembly using the 105mm radius and to substitute a sprocket to suit the Bru–500 motor after initial testing had been performed. However, as discussed in section 6.2.5, initial tests of the S–4050 motor identified the rotor inertia to be greater than that specified, which made the motor incapable of driving the given profile at full machine speed. It was therefore decided not to modify the mechanism by changing the sprocket to suit the S–4050 motor, but to use the existing drive radius for synchronisation tests. The recurring problems with the drives led to the feeling that to achieve the required machine speeds, the Bru–500's would have to be later replaced by newer drives, which would require further modification of the mechanisms. Thus the slider mechanism was driven by the S–4050 motor through the 105mm radius drive sprocket.

6.3.2. The arbor drum mechanism.

Section 6.2.6 described the difficulties in obtaining a drive for the arbor drum which led to the decision to build a full scale low inertia model of the drum for use in the laboratory testing work. The use of a model drum provided the capability to test the control and synchronisation algorithms within the time frame of the research programme, since suitable drives for the actual drum would not be available until after the project completion date. Thus the arbor drum model was designed with the same drum–slider spatial overlap dimensions as occur in the actual system.

Figure 6.16 shows the arrangement of the arbor drum mechanism. As with the transfer slider, the mechanism was originally designed to suit the Moog 306–023 servodrive and was later adapted for a Bru–500 drive. The drum wheel itself is constructed from polypropylene. The 16 openings around the circumference of the drum are dimensioned and positioned to the same tolerances as the arbor openings on the actual drum. Thus the positional resolution and synchronisation requirements of the model are identical to those of the actual drum. The complete assembly was designed to have several configuration options. The drum and motor each have individual mountings. This enabled the two to be coupled directly or through a drive ratio. In addition, it is possible to remove the drum from its mounting and couple it directly to the motor shaft.

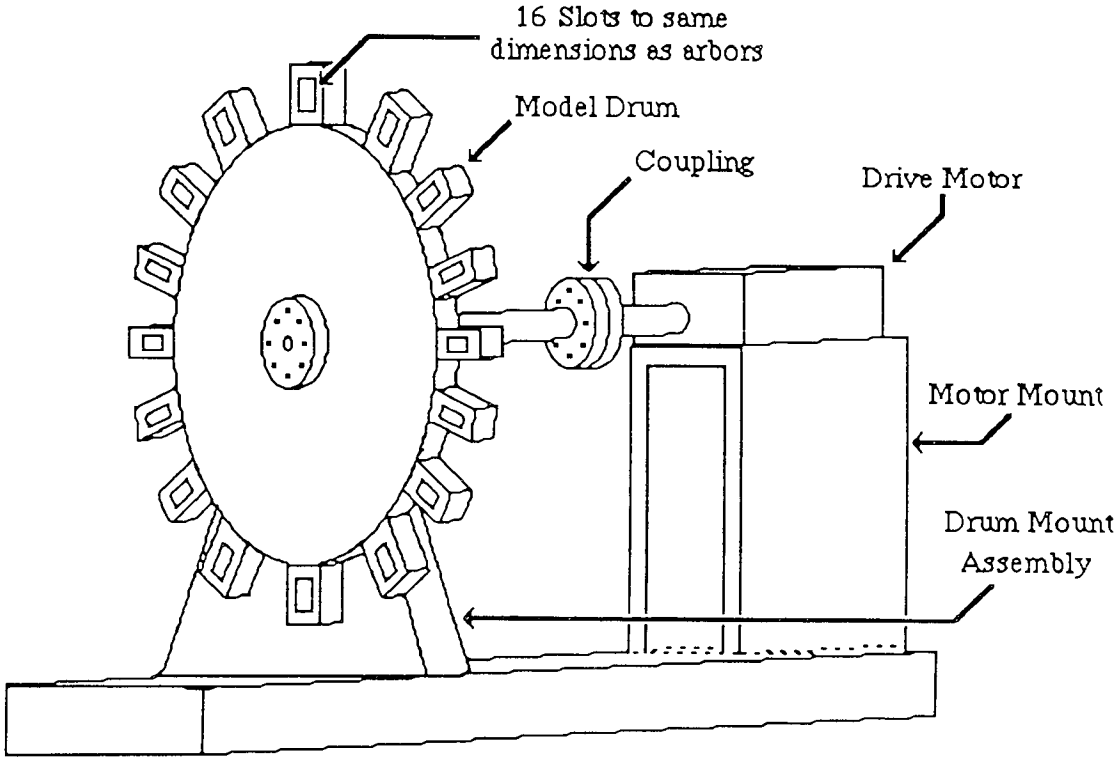


Figure 6.16 The arbor drum mechanism assembly

6.4 The two actuators' system controllers.

This section of the work describes the hardware and software implementation of the transfer slider and arbor drum synchronisation and control algorithms. The development of the multi–actuator synchronisation and coordination algorithms is described in chapter 4 , with the position control algorithms for the Bru–500 drives being developed in Appendix 3. The aim of this part of the laboratory work was to prove that the control techniques developed using simulation could be succesfully implemented in a real system.

Each of the actuators required the three control functions of actuator coordination and synchronisation (ACS), actuator feedback control (AFC) and actuator process monitoring (APM) to be performed. The laboratory implementation of the functions is discussed below.

6.4.1. Actuator Feedback Control.

The requirements for the AFC are discussed in section 5.4.1. The minimum requirements were stated as being:

(i) The capability to receive, interpret and act upon commands from the ACS.

(ii) The ability to store and read a motion profile.

(iii) The capability to interpret transducer feedback signals from the actuator or load.

(iv) The capability to control the motion of the servomechanism through the motion profile within the manufacturing tolerances.

(v) The ability to store emergency stop routines and to act upon them when instructed.

Additional requirements for system flexibility and ease of development were:

(vi) The ability to change the motion profile being followed.

(vii) The ability to output control variables to external devices.

(viii)The ability for on–line tuning to be performed.

In the initial plan for the laboratory implementation it was intended to use Moog motors for the two drives and to develop controllers for the motors which fulfilled all the above requirements. A flexible controller was developed for the Moog motor. However, the failure of the Moog motor and the adoption of Bru–500 motors introduced difficulties because the developed position controller was not compatable with Electro–Craft hardware. The control algorithms were based on the Moog motor's frequency response performance, the input–output (i.o.) algorithms were written to interpret the 4000 count absolute resolver feedback data and the controller hardware was configured to be compatable with the Moog hardware.The Bru–500 communication signal levels were smaller, the drive enabling routines were different, the position feedback information was from an 8000 count incremental

encoder which required interpretation hardware to be developed and the control software itself did not suit Bru–500 characteristics. Thus a new position controller would have to be developed.

The Moog controller had taken a considerable amount of time to develop. At the time of selection of Bru–500 motors, they were the fourth type of brushless d.c. motor to be evaluated by Molins or Aston University, with the three previous ranges from Inland, Moog and Electro–Craft each suffering from severe performance and reliability problems. Therefore there was not sufficient confidence in Bru–500's to justify developing a new position controller which might have had to be abandoned. Instead it was decided to purchase a proprietory controller for motor testing and, if necessary, to adapt the control algorithms to suit the requirements of the research programme.

The design of the Moog controller hardware which fulfilled the controller requirements is described below, along with the controllers which were purchased for the Bru–500 motors and their implementation in the laboratory.

6.4.1.1. The Moog motor AFC.

The controller for the Moog motor was developed to fulfil all of the above requirements. A schematic drawing of the controller hardware is shown in figure 6.17.

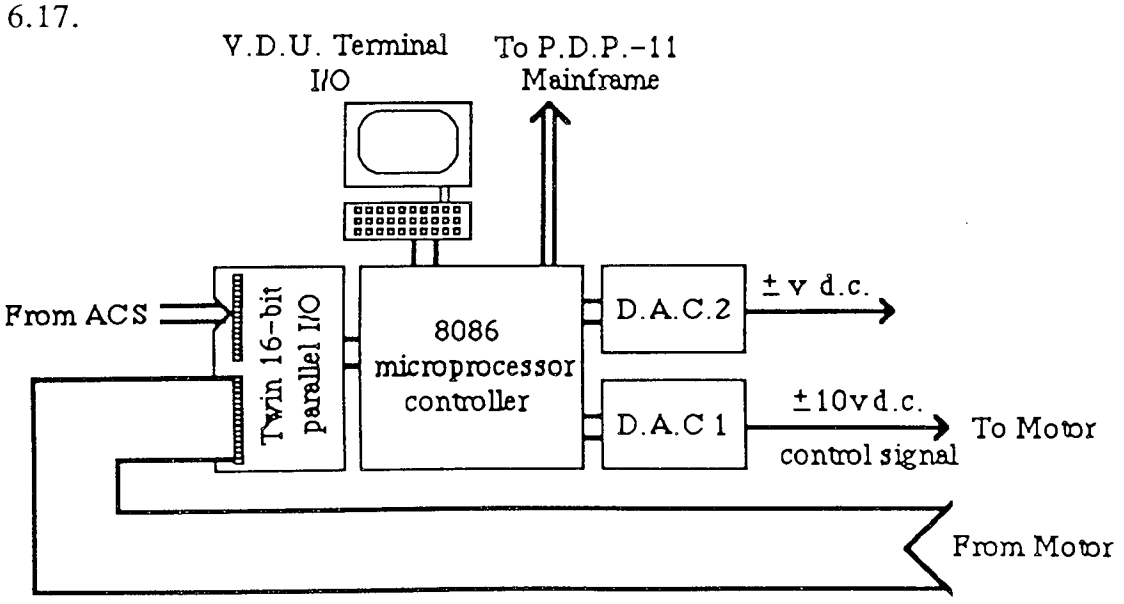


Figure 6.17. Configuration of controller for the Moog motor.

The control software and the profiles to be followed are stored in EPROMs within the controller. These were capable of storing several motion profiles and the emergency stop routines in look–up table form. Communications from the ACS and

the motor, including the resolver feedback information, took place through two 16–bit parallel interface ports. The control signal to the Moog velocity regulator took the form of a +_10 v d.c. signal, which was supplied by the controller using a 12–bit real time digital to analogue convertor (d.a.c.). Thus the five minimum requirements for the controller were fulfilled.

The capability to alter the motion profile being followed came from the ability of the controller to read several profiles, which were stored in the EPROM. In addition, there was the capability to access a PDP–11 mainframe computer for on–line development of software, which enabled profiles to be altered, control algorithms to be developed and controller constants to be tuned. Finally, controller variables and other information could be output to a terminal screen or through a second d.a.c. to an oscilloscope or recorder. This then enabled the additional requirements for the controller to be fulfilled.

The controller hardware described here enabled a very flexible controller to be developed which fulfilled all the requirements for an independent actuator controller. Unfortunately, although the controller was built and the interfaces were tested, the failure of the Moog motor during initial controller testing meant that the controller was never used.

6.4.1.2. The Bru–500 motor AFC.

In purchasing a controller for the Bru–500 motors, it was necessary to consider the requirements listed above to assess how these may be applied to the selection of and operational, pre–tested controller. Commercial controllers are purchased as developed units and suppliers are not often willing to release detailed information about their operation and software. Several controllers were available which fulfilled the five minimum requirements discussed in section 5.4.1., but none could be obtained which fulfilled all the additional flexibility and development requirements as well. Therefore the selection of controller was to be the one which could best fulfill the complete set of requirements.

A simulation analysis was performed to determine the optimum control algorithm for Bru–500 position control. This analysis is described in Appendix 3. It was determined that a proportional plus derivative plus velocity and acceleration or velocity feedforward control algorithm gave the optimum response in both the linear and current saturated motor performance bands. Therefore this algoriithm requirement was included in the controller selection.

The controllers which were selected and implemented were PROPOS–E position controllers built by Electro–Craft. These employ proportional plus integral plus derivative plus velocity feedforward control algorithms and are compatable with Bru–500 hardware. The motion profiles these employ are programmable trapezoidal velocity blends, which are shown in Appendix 3 to be suitable for intermittent motions. The programming capability includes wait–for–input, branch–on–input and output commands, so the controller is capable of receiving and acting upon external commands and can return communications to external devices. Additional interupt ports enable emergency stops to be triggered externally and service routines can be included in the programming to give the propper response to such interupts. Position feedback information is taken from the motor's encoder. Thus the five minimum requirements for the controller are fulfilled.

The PROPOS–E has the capacity for storing up to 15 motion profiles, which introduces the capacity for on–line functional changes. In addition, the controller constants can be tuned on–line, so two of the three additional controller requirements can be fulfilled. However, the capacity to output control variables is only partially fulfilled. All control variables are software generated and as such are not continually available. The only control variable which can be output is position error, which can be output to a terminal screen at discrete points in the motion profile. This has some limitations for the development engineer, since it is not possible to obtain plots of the motor's required position, or other variables such as position error. The system's performance must therefore be assessed using plots of the motor's velocity as it moves through the position profile. This information can be combined with the position error at discrete points in the profile to compare against simulation plots of the system moving through similar motion profiles.

6.4.2. Actuator Process Monitor.

The APM must be capable of monitorsing actuator transducer inputs and detecting if the feedback information falls outside of pre–defined tolerance values. The action of the APM is described in section 4.2.3.2. In the initial implementation scheme, it was intended to perform the APM functions using the controllers which were developed for the Moog motor. Since all profile and response information would be known, the APM routines could be readily included within the controller software.

With the selection of Bru–500 motors to replace the Moog motors, it was necessary to develop a new method of implementing the APM to suit the new hardware. Many commercial position controllers include emergency stop flags which can be set to trigger if the system's performance moves outside a programmed tolerance band.

Such a facility is available in the PROPOS–E and this was the final justification for its selection. A position error flag can be set to trigger if the error exceeds a programmed value. This enables the motor's performance to be monitored continuously through the motion and dwell cycles and so performs the APM function. Although this introduces only one tolerance band, for development purposes this was sufficient to ensure no hardware clashes could occur.

6.4.3. Actuator Coordination and Synchronisation.

The operation of the ACS function for each actuator is described in chapter 4. The ACS's were required to communicate with the PROPOS–E AFC and APM functions of the two actuators. The ACS functions for the two actuators are both performed by an Epson personal computer (PC), with the software being written in C. A parallel interface card enables the PC to communicate with the two Propos–E units. Each communication between the Propos–E's and the PC is duplicated using separate i.o. lines, ensuring hardware redundancy on all communications. In addition, each communication is tripple polled to filter out electrical noise. Thus for an ACS to trigger motion commencement it must send the GO command along two independent communication lines, with the AFC double checking each line to ensure the GO command is true. In return, all APM communications to the ACS utilise redundant back–up lines, with the ACS double checking each time it reads the input.

C software was employed to ensure compatability with Molins' control software. Since C is a sequential language containing all the constructs used in the Fortran coded multi actuator simulation control software, the algorithms developed in the simulation work were readily transferable to the laboratory. The flow chart showing the operational sequence of the PC software is given in figure 6.18. The program begins with an initialisation routine, which sets the MOTION and PRODUCT conditions to their start up states. Since neither actuator actually moves products, a simulated product is considered to be moved in the same menner as that moved in the multi actuators simulations, with the product considered to have been passed when the passing actuator's MOTION condition changes from OCCPTN to CLEAR.

The continuous sampling loop begins by reading the communications inputs from the drum and slider. The error flags are checked first. If an error flag has been set by and APM, the program moves to a stop routine to halt both motions, then displays the types of error which has occured to the PC screen. If the error flag has not been set, the program updates the two sets of PRODUCT and MOTION

conditions from the APM communications. The slider and drum ACS routines are then performed. The coding is very similar to the Fortran coding in section 4.4 for the simulated system controller, with the same IF–THEN operators being used to interpret the sets of PRODUCT and MOTION conditions to determine the action the actuator should take. Each ACS has autonomous control over its actuator's actions and sets its own PRODUCT and MOTION conditions according to its decisions, which the other ACS then uses to make its own control decisions.

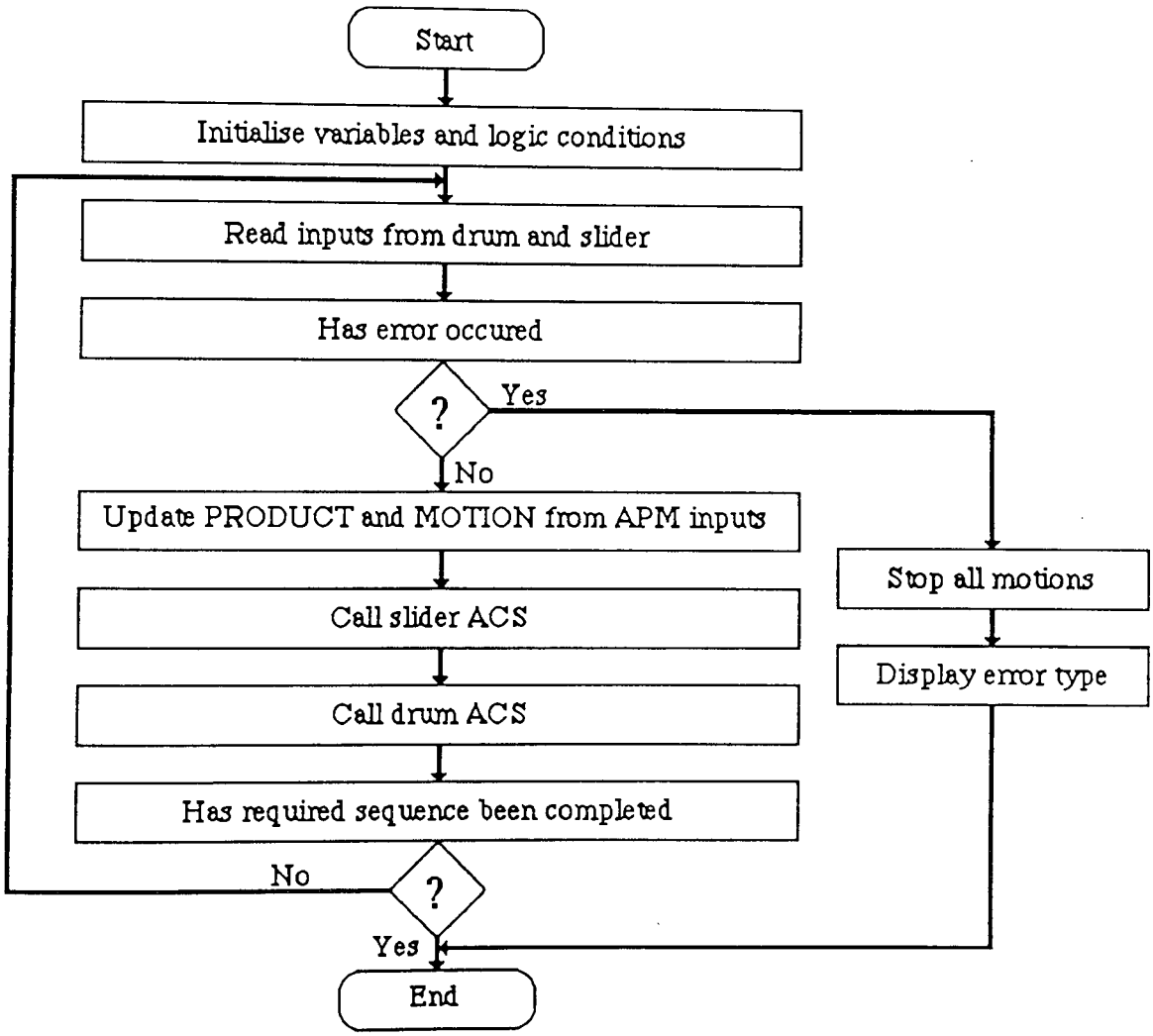


Figure 6.18 Epson PC software structure.

The main loop ends with a counter, which increments each time a drum increment has been completed. Thus the number of motion cycles of a run pre–programable, with the system halting when the desired sequence has been completed.

6.5 Implementing the two actuators system.

The two actuators implementation in the laboratory is represented in figure 6.19. The transfer slider and arbor drum each have a mechanism to perform their

functions, each mechanism has a motor to actuate its motion, each motor has a controller to control the motion and the two controllers are software coupled. Thus operational integrity has been introduced to the two actuators system, with the five categories of machine element described in chapter 3 being provided.

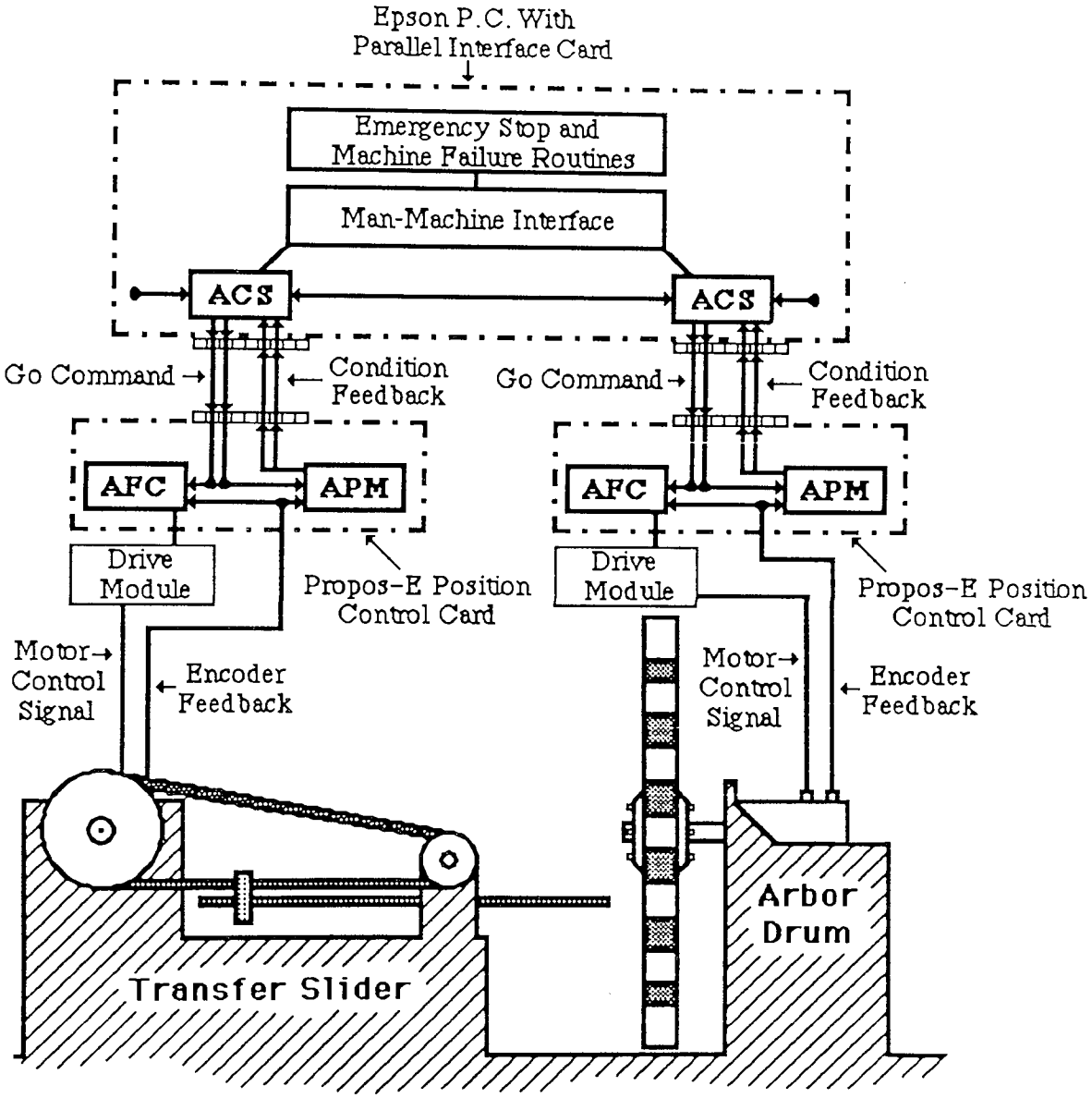


Figure 6.19. Transfer slider and arbor drum controller implementation scheme.

The implementation of the system in the laboratory was limited by failures in the drives which restricted the maximum machine speed to 210 transfer and increment cycles per minute when running with the given motion profile requirements. This was increased to 315 cycles per minute by running the slider with a less arduous profile, but this was the maximum attainable due to drive restrictions introduced by Electro–Craft to prevent further drive failures.

A summary of the failures which occured and the corresponding restrictions these imposed is given below. This is followed by a description of the synchronised system's performance in the laboratory. Finally the implications that the system's performance have on the design of independent actuators machinery is discussed.

6.5.1. Failures and limitations of the Bru–500 drives.

{1} Increased inertias.

The increased inertia problem of the supplied S–4050 motor described in section 6.2.5 was also a problem with the S–6200 motor. The specified inertia of the drive was 80% of the value of the actual motor's inertia, given in section 6.2.7. This placed peak and RMS power rate restrictions on both actuators' performances.

{2} Bearing seizure.

The increased inertia modification was accompanied by increased bore bearings to accomodate the increased rotor diameter. In the case of the S–4050 drive, it was found that when the transfer slider was run at high cycle rates, the bearing overheated and gradually seized. This introduced a friction loading, which in turn caused an increased position error. This was detected by the APM, which triggered the error flag and stopped the system. If the error tolerance was set to a large value, the bearing overheating eventually caused the motor to shut down due to current overload. After cooling, the motor again ran freely.

The effect of the bearing weakness was to limit the cycle speed at which the transfer slider could be run for long periods.

{3} Software induced limit cycle.

The original current drive software contained a fault which under certain conditions could cause the drive to enter a positive to negative peak current limit cycle. This induced a violent non–linear oscillation in the motor. The conditions which triggered the oscillation were never precisely determined. Improved software was supplied by Electro–Craft which did not suffer from this problem. However, the improved software included a solution to a problem which had been encountered elsewhere, though not at that time in this research project. The problem was that if the system ran into a limit stop, a current build up could cause an explosive failure in the drive module. The "quick–fix" solution which Electro–Craft had implemented to solve this was to limit the peak current to the RMS limit when the motor was running at less than 30 rev/min. In addition, the RMS current limit was reduced by approximately 15%. These limits were included in the replacement software supplied.

In intermittent motions the largest current demands can occur at low speeds, during starting, settling and changes of direction. The low speed current limit and reduced RMS limit therefore resticted the operation of the transfer slider, particularly at high cycle rates, when an increased initial error and a final position overshoot on settling occured.

{4} Velocity loop limit cycle.

When the Bru–500 motors were fitted with inertia loads, it was necessary to adjust the BAND and DAMP settings to give high current loop gains and bandwidths in order to produce acceptable responses. This introduced a limit cycle to the velocity controlled system, which was measured accurately to have a magnitude of less than 0.5mm at 550mm radius from the from the centre of the rotor. This corresponded to a magnitude of 1 encoder count. The oscillation could be removed using the software filter FBW, but this was at the expense of increased response overshoot and transient settling time.

The effect of this oscillation was to limit the position resolution which could be obtained during high speed operation of the two mechanisms.

{5} Displacement forcing failures.

Tests were carried out on the arbor drum system to determine the drive's response to displacement forcing. The intention was to investigate the Knock–on effect of a failure in a machine element, which would be expected to induce a displacement forcing in surrounding actuators.The arbor dum was diplaced by approximately $^1/_{16}$ revolution, to be released and the response recorded. The displacement caused an explosive failure in the drive module. The limit stop current build–up fault described in {2} above had been triggered by the displacement. The replacement software included the same reduced RMS and low speed peak current limits that were supplied in the replacement S–4050 software. Similar operating restrictions were therefore introduced to the S–6200 drive, with reduced initial acceleration, increased settling time and reduced machine cycle speed capability resulting.

{6} Electrical noise.

When operating the two motors were found to generate large amounts of electrical noise. If the tripple polling function on the control communications was removed, the noise caused occasional collisions between the slider and drum by triggering motions to commence. The tripple polling removed this, but the noise interfered with other equipment in the laboratory. The noise levels increased as the machine speeds increased, due to the magnitude of the current peaks being supplied to the motors.

The electrical noise did not inhibit system performance, but it would restrict the ability to implement brushless servodrives outside of the test environment. The phenomenon is being investigated as part of the second Aston/Molins S.P.P. project and as part of a separate S.P.P. investigation at Bristol University.

6.5.2 The synchronised actuators performance.

The combined effect of the limitations imposed by the failures described above was to limit the operating speed of the combined motions. The transfer slider was able to meet the given insertion position tolerance and the 1 m/sec constant velocity requirement in a profile of 175msec minimum duration. The velocity response of the slider following this profile is shown in figure 6.20.

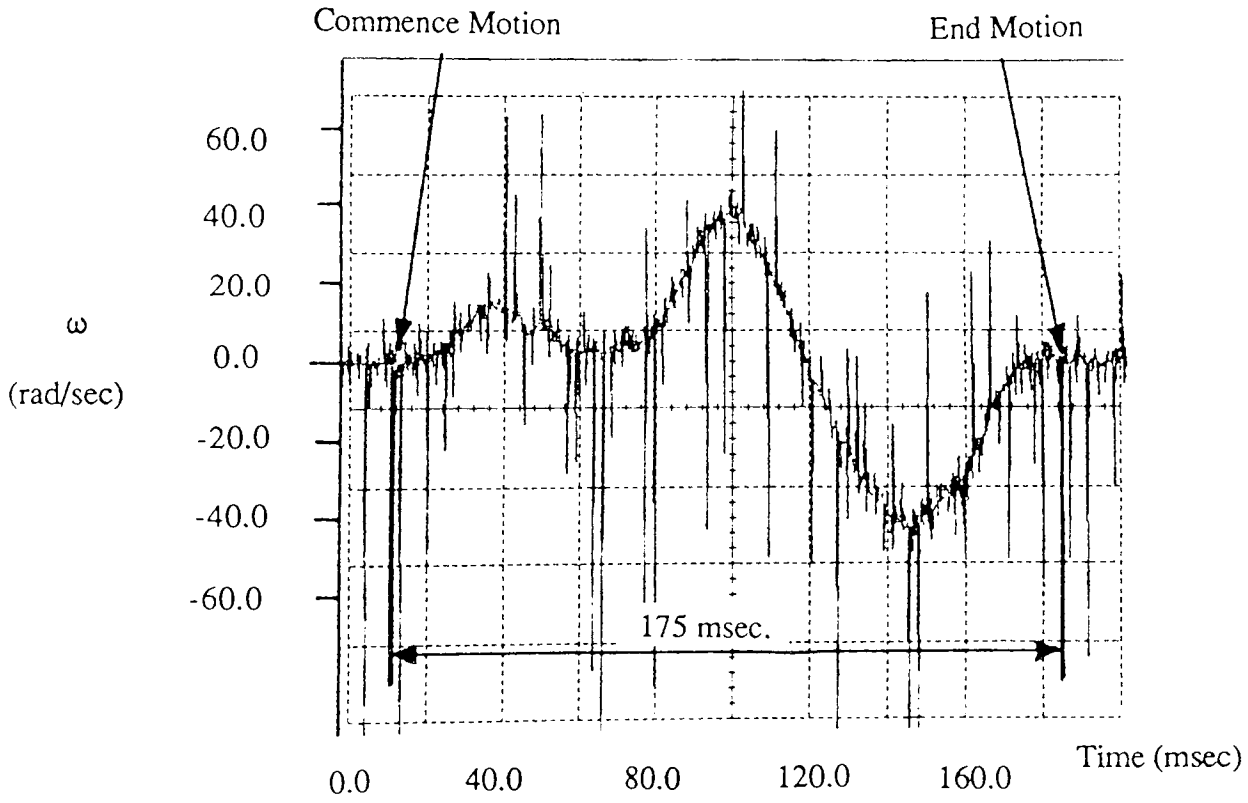


Figure 6.20 Velocity response of transfer slider following 175 msec position profile

Figure 6.20 clearly demonstrates the electrical noise generated by the drive under high load conditions. The spikes in the plot are the electrical interference caused by the current switching action of the electronic commutation. The level of the noise signal varies with the amplitude of the current. The 175msec profile places very large current demands on the drive, which causes large levels of electrical noise to be created by the current switching transistors. The velocity plot demonstrates the constant velocity segment on the insertion stroke. The peak insertion position

tolerance obtained was +_ 2 encoder counts, which was equivalent to a linear stroke tolerance of +_ 0.165 mm.

One of the main aims of the laboratory work was to verify that the models and techniques used in the multi actuators simulations were correct. Figure 6.21 shows the simulated velocity responses of the closed position loop system. The overlay plot is the response of the actual system in the laboratory following the same transfer profile, but showing two motion cycles whereas the simulated response shows only one. The two plots were obtained with the same controller settings and are drawn to the same scale. The close match between the plots confirmed the accuracy of the position servo models used in chapter 4.

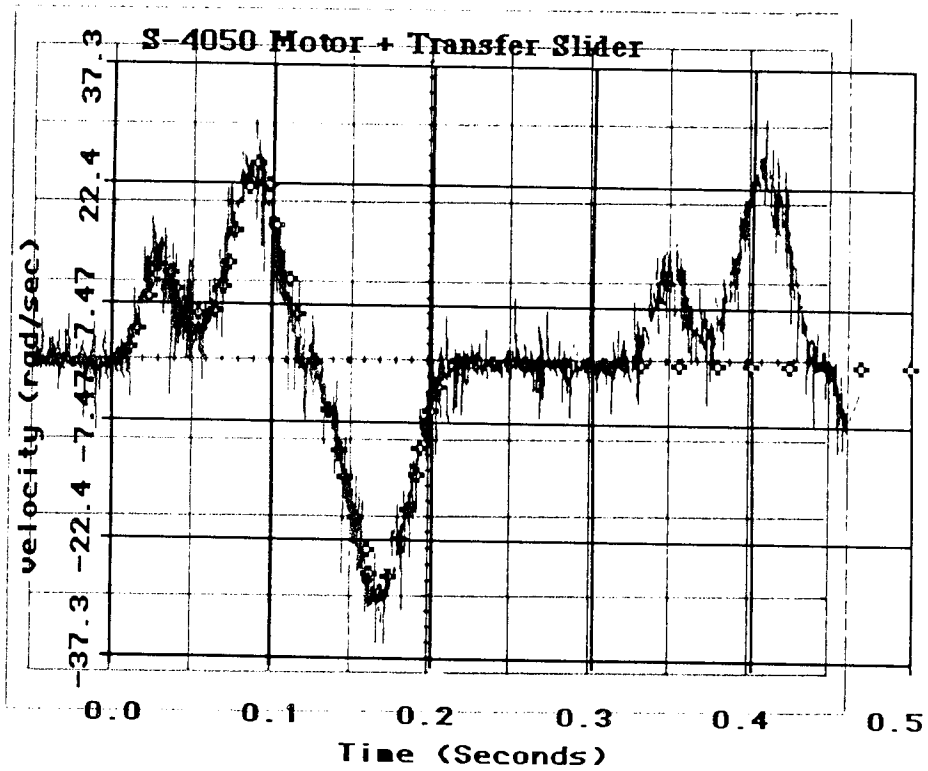


Figure 6.21. Comparative simulated and actual responses of the transfer slider.

When the transfer profile was run at cycle periods of less than 175msec, current saturation occured. This inhibited either or both the peak position and constant velocity requirements. In addition, overheating in the motor bearings degraded the profile followed. However, it was possible to run shorter period profiles for high speed synchronisation tests by removing the constant velocity segment and decreasing the length of the insertion stroke. The minimum transfer period employed in the combined system synchronisation tests was an 85msec in–out motion of magnitude 150mm. However, the shortest transfer period which could be driven within specification was 175 msec.

The arbor drum attained a position to the nearest encoder count of the required value during the dwell periods between motions. This corresponded to 0.132 mm at 336 mm radius. When stationary, the limit cycle oscillation described in section 6.5.1 had a magnitude of 1 count, centred on the required dwell position. Although the tolerance does not meet the given specification, it is within that attained by the existing arbor drum, with the encoder resolution being equivalent to the backlash of the cam–driven mechanism. The minimum increment period attainable was limited by the software limits discussed in 6.5.1 to 107 msec. Shorter period profiles were not possible due to the RMS current sensor disabling the drive.

The variable characteristics with position discussed in section 6.2.7 were visible in the incremental motions of the drum. The motor is an 8–pole motor and has 4 peaks and 4 troughs in its characteristic with position. The arbor drum moves in increments of $^1/_{16}$ revolution, so there were four sets of four response characteristics. However, the final position after each increment settled was within the same 1 encoder count tolerance.

When the two actuators were synchronised, the maximum combined speed at which both actuators could meet the specified tolerances was 210 transfers and increments per minute. For this, the drum's cycle time was 107 msec and the transfer slider's cycle time was 175 msec. By running the transfer slider with the 85 msec profile described above, the cycle speed was increased to a maximum of 310 cycles/minute. This was the maximum speed attainable due to the software RMS current limit imposed by Electro–Craft on the S–6200 drive.

6.5.2.1. Testing the two actuators response to failures.

In chapter 4 the response of systems of independent actuators to failures was tested using simulation models. Since the models have been demonstrated to be correct, the simulation tests verify that the controlled system responds properly to failures. In the laboratory tests, it was not intended to perform many tests of the reponse to failures because of the unreliability of the drives and the potential for extensive damage being caused by high speed collisions. However, a beneficial effect of the unreliability of the drives was that genuine machine failures enabled the process monitoring function to be tested on–line.

If current overload was detected by either Bru–500 drive, it would disable the drive and the motor would free wheel. This occured several times in both the transfer slider and the arbor drum. On each occasion an error was detected by the APM and

the actuator's danger flag was set, which stopped the unaffected mechanism's motion, so no collisions occured.

When the transfer slider's motor suffered bearing seizures, the APM detected the performance fluctuation and stopped the motor, which in turn stopped the unaffected motor.

When peak current limit cycles occured in either drive, the APM detected this and the unaffected drive remained stationary.

On no occasion did a failure in either drive cause a collision to occur. The APM's detected errors and the unaffected actuator remained stationary. In addition, push–button switches were introduced to each actuator to enable the APM dynamic error flags to be manually triggered. This caused the actuator to stop, which in turn caused the untriggered actuator to hold a stationary position.

6.5.3.Improving the performance of synchronised actuator systems.

The given aim of the laboratory work was to independently drive and software couple the third transfer and arbor drum mechanisms. As has been described, no drive was available for the drum, but one will be available after the project has been completed, so the given aim was altered to enable a model drum to be synchronised to with the third transfer slider. Throughout the work which has been performed towards achieving this aim, the system failures and performance limitations which have been described have all been related to the brushless d.c. motors being tested. The Bru–500 motors which were used for the implementation have been found by the author and by Molins' development staff to have the best performance and reliability of all brushless d.c. motors tested, but despite this the failures and limitations described in section 6.5.1 served to limit the system speed to 210 cycles per minute when running with the specified profiles, which is less than one half the peak required operating speed. At the time of publication of this thesis, Bru–500s are widely considered to be the best brushless d.c. motors available, with the large motors which are expected to be available for the arbor drum drive also being Bru–500 designs. Therefore continuing S.P.P. work will incorporate Bru–500 drives, so it is important that areas be identified in which these must be improved, so that the ongoing research programmes can achieve improved machine performance.

Primary amongst the areas for improvement is that of general reliability. It is clear that Bru–500s are not sufficiently tested before being released for sale. Several of

the author's drive failures occured due to errors in drive software, with each requiring a "quick–fix" software correction by Electro–Craft to enable work to proceed. These would have been avoided if the drive software had been adequately tested before delivery.

The performance constraints introduced by the re–inforced higher inertia rotors could also have been avoided if the motors'performances when driving large loads had been pre–tested. The tendency for rotor bearing seizure under high load conditions should also have been corrected before delivery. Both of these weaknesses are the result of poor mechanical design and inadequate testing. The method of testing adopted by Electro–Craft is to run a drive system at constant speed for varying periods of time. Although this method serves to prove that the motor and controller will function as a system, it only tests one area of the drive's performance characteristics. It does not identify potential drive faults which can occur during transient response performance. It is the transient performance which induces the highest energy loading and which is most likely to identify weaknesses in the system, such as torsional resonances or current overloads. Therefore it is recommended that the Bru–500 systems be more rigorously tested before sale, with emphasis being placed on testing all performance characteristics of the systems.

The drive module control software must be rewritten and improved. The BAND–DAMP control sofware does enable the effect of parameter adjustment on system natural frequency and damping ratio to be understood, but at high bandwidth settings an encoder quatization limit cycle oscillation is introduced. This needs to be compensatable by an improved digital filter. The FBW filter simply reduces the system bandwidth, which degrades the transient perfomance when used to compensate for the encoder limit cycle. The should be replaced by a more discriminating filter, such as a digital lead–lag. The introduction of this should be accompanied by a re–write of the control software to remove the faults which have been identified and to enable a linear system perfomance throughout the motor speed range.

The redesign of the software and the improvement of the drive testing procedure should be readily implementable. More complex electromechanical redesign is also required to improve the performance of the motors. The variable characteristics with rotor angle effect is detrimental to intermittent motion control systems. In continuous systems this effect requires the redesign of the winding configuration, or reprofiling of the current waveform, or a combination of both. Although this can

involve major redesign work, it will be necessary if the motors are to be implemented in high speed machinery.

The laboratory work described in this chapter has combined with the simulation and theoretical work to determine a means by which flexibility may be introduced into product which is suited to independent actuators machinery application. Since Bru–500s are expected to figure prominently in ongoing research work, it is most important that each of the improvements is investigated. With their existing standards of performance and reliability, the drives are not yet suitable for machine application.

Chapter 7.

Flexible high–speed machinery.

7.1. Introduction

The author Arthur C. Clarke once suggested that revolutionary technology passes through three stages of development:

1. " It's impossible – dont waste my time."

2. " It's possible, but not worth doing."

3. " I said it was a good idea all along."

During the three years that this research programme has been running, the idea of developing independent actuators machinery has moved from Clarke's stage 1 to the point where it is now well on the way to stage 3. At the beginning of the project, the idea met with a cold reception at mechanical engineering seminars and conferences. It was said that the dynamics of the motions could not be repeatedly achieved by independent drives, that the energy requirements would cause the drives to burn out, that synchronisation could not be maintained at high machine speeds and anyway, why change a method of design which has been used for many years?

At more recent conferences, demonstrations and videos of synchronised drive systems in operation have created a much warmer response to the new design approach. Demonstrations of machine function reprogrammability have answered the question of why the design method should be changed, while the fact that the servos are repeatedly achieving the required profiles without burning out and without colliding have shown that it can actually be done. In addition, the second S.P.P. research project at Aston University has found that the drives can be coupled to a common d.c. bus, which enables regeneration to take place and reduces the energy inefficiencies of mechanically decoupled functions. However, although independent actuators machines have been demonstrated in principal to be feasible, it has also been shown that much additional work must be performed before an operational system can be produced and marketed.

The aim of this research programme has been to investigate methods for the design of flexible high–speed machinery. This has been an S.P.P. phase 1 investigation, with several phases of investigation being planned, so the identification of areas in which further research must be performed has been an important aim of the work.

The results of the work which has been performed and how it may be applied to flexible machinery design are discussed below, along with a discussion about areas in which further investigation should be carried out if operational machinery is to result.

7.2. The independent actuator controller scheme.

The controller scheme described in chapter 4 was developed to enable each actuator to take autonomous control of its own actions. Inter–actuator communications enable each actuator to decide what its own actions should be, while the ACS pipeline communications avoid the potential performance limitation which could arise in a centralised hierarchical controller. The pipeline communications enhance both structural flexibility and actuator modularity, but maintain the control hierarchy which enables the upper control levels to have access to all lower level functions. This is necessary for the overall machine control and external interface to be readily implementable.

This method of machine control has been demonstrated to be operable, both using simulated multi–actuator systems and for the two actuators implementation in the laboratory. The control structure can be flexibly implemented, with a single sequential program being used to control the simulated systems and the laboratory implementation incorporating three communicating microprocessors. However, neither the simulated or the laboratory systems involve modular implementations of the controller. Both were specifically designed as development tools to test the control functions and employed methodologies which suited their application environments.

A modular implementation of the control structure within a manufacturing environment will have a form similar to that shown in figure 7.1. Each actuator module has a single transputer to handle its ACS, AFC and APM control functions. The ability of the transputer to concurrently process parallel sequential functions makes it most suitable for this form of implementation. The transputers communicate with each other along the actuator pipeline and with the machine module, which serves as a framework for the actuator module interconnections and performs the machine level functions, such as the man–machine–interface and emergency stop coordination and recovery. The machine module will be able to communicate with other machine modules via the machine pipeline to coordinate the machine's actions with those of the rest of the production plant. Thus the actuator control module structure could be upwardly and outwardly extended to encompass sets of communicating machines. At the machine level, intelligence may be

introduced to enable the machine to coordinate its own actions to suit the flow of materials through the production plant. Intelligent machine functions such as failure prediction or product quality trend analysis, both of which are implemented in Molins' traditional machinery range, can be introduced to enhance the machine's function.

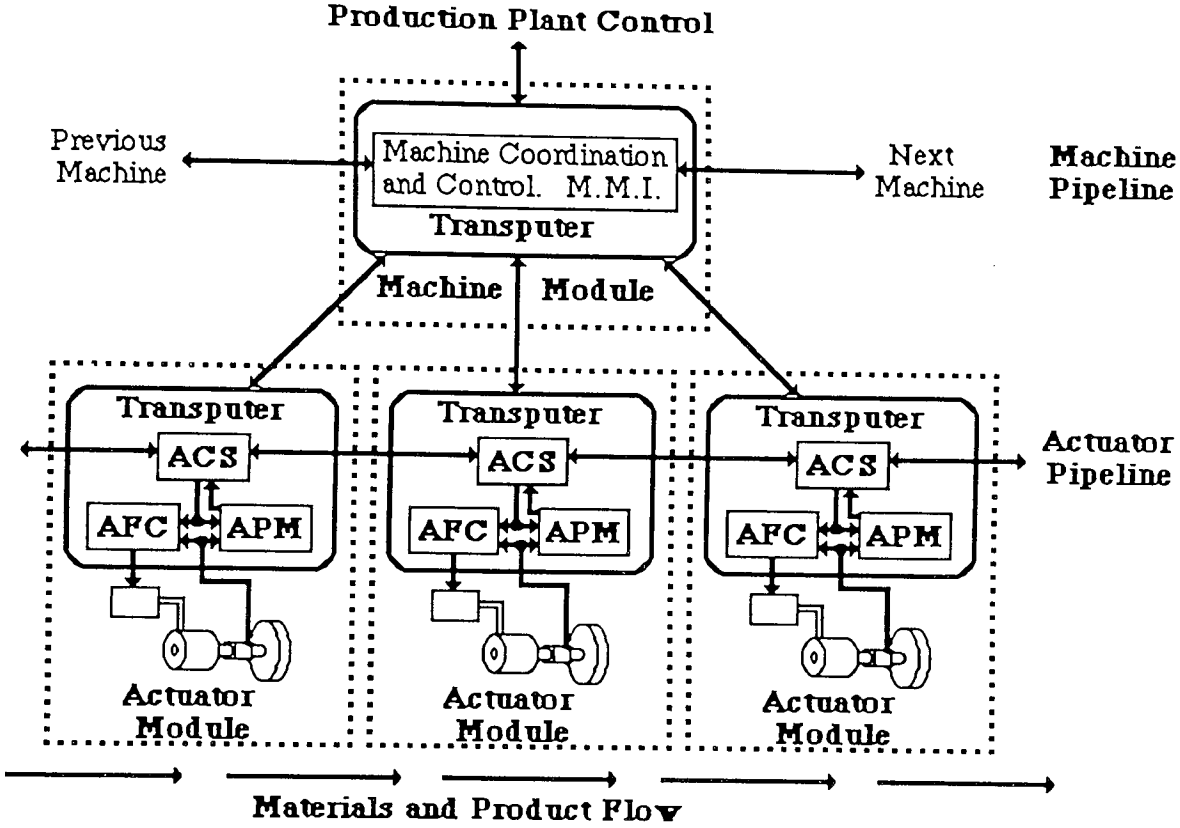


Figure 7.1. Modular implementation of production plant control.

The achievement of the implementation described will require much additional research work to be carried out, some of which is already underway at Aston University. Real–time interface cards have recently become available which are compatable with Intel transputer hardware and work is proceeding towards using these to enable transputers to control actuator hardware. Current research is investigating the use of the transputer's external event interrupt facility, linked to a hardware clock timer, to trigger the sampling action of the feedback controller. It is then intended to use two transputers to control the third transfer slider and arbor drum mechanisms, with each transputer handling the ACS, AFC and APM functions of one actuator. This should then enable a transputer based implementation of the author's two actuators controller described in chapter 7.

When a multi–processor or multi–transputer machine controller is finally implemented, the programming of the machine functions will present some

difficulties. In a production plant made up of communicating independent actuators machines, the manufacturing process control hierarchy will contain a large number of communicating processes. A large machine could have up to 20 actuators, while a single production unit would have at least three machines with the machines being linked by robots or buffers, each of which will have its own communicating processes. A production plant will ordinarily contain several production units, so the implementation of product changes could require several hundred autonomous processe to be reprogrammed. Thus it will be necessary to develop programming methods which enable a centralised, user friendly production plant interface to be used to implement manufacturing function changes in the distributed controller array. It would not be a flexible manufacturing system if functional changes had to be independently calculated and reprogrammed for each actuator. So a control and interface language must be developed which will enable process changes to be flexibly implemented.

Proposals are currently being developed for software engineering research at Aston University to investigate methods for programming and controlling independent actuators machinery plants. This will incorporate experience which has been gained at the University in the use of distributed real–time control systems, in software fault tolerance and in the formal specification and verification of distributed control software.

7.3. The independent actuator drives

Brushless d.c. servodrives have been demonstrated to be capable of being flexibly controlled and implemented as part of synchronised actuator systems. Their characteristics can be accurately modelled and their performance predicted by numerical simulation. However, the laboratory work has demonstrated that even the highest perfomance neodymium–iron–boron motors, which are currently the highest state–of–the–art in electric motor design, clearly do not yet have sufficient standards of reliability and performance to enable their inclusion in operational machinery. Section 6.5.3. described the improvements which must be carried out to make them suitable for machine application. If these improvements are implemented, then the arrival of the larger drives, which are due in early 1990 will give a large range of suitable drives. Brushless d.c. servodrives are still very new technology, but the large demand for such drives means that a great deal of development work is being performed by their manufacturers. This is unlikely to produce another leap in performance, as occured with the introduction of rare–earth magnets and high current electronic commutation, but the quality and reliability of

existing designs will without doubt improve. Research is currently proceeding amongst manufacturers into the use of magnets constructed from sintered magnetic conductors in a plastic matrix. Unfortunately, due to commercial secrecy, it has not been possible to obtain details of this work, so it is difficult to predict its effect on motor performance. However, if thermal and structural strength difficulties can be overcome, it would be expected to produce very low inertia motors with very large power rate ratings and low mechanical time constants, which would be most suited for our applications.

In addition to the use of brushless d.c. servomotors, it may be useful to consider the use of other types of servodrive. The independent actuators machine scheme was developed to enable alternative types of drives to be incorporated. One type of drive which is worthy of consideration is the electrohydraulic servodrive. These have not been considered in this research work because Molins' engineers do not feel that they are clean enough for use in processing machinery. However, examination of their specifications demonstrates that their dynamic performances could make them suitable for machinery drives. In chapter 5, driveability plots were introduced. Figure 7.2 shows such a plot for a range of Moog electrohydraulic servodrives. Plotted on the same axes are the peak and RMS driveability ratings of the Electro–Craft Bru–500 servodrives which were used for the laboratory implementation.

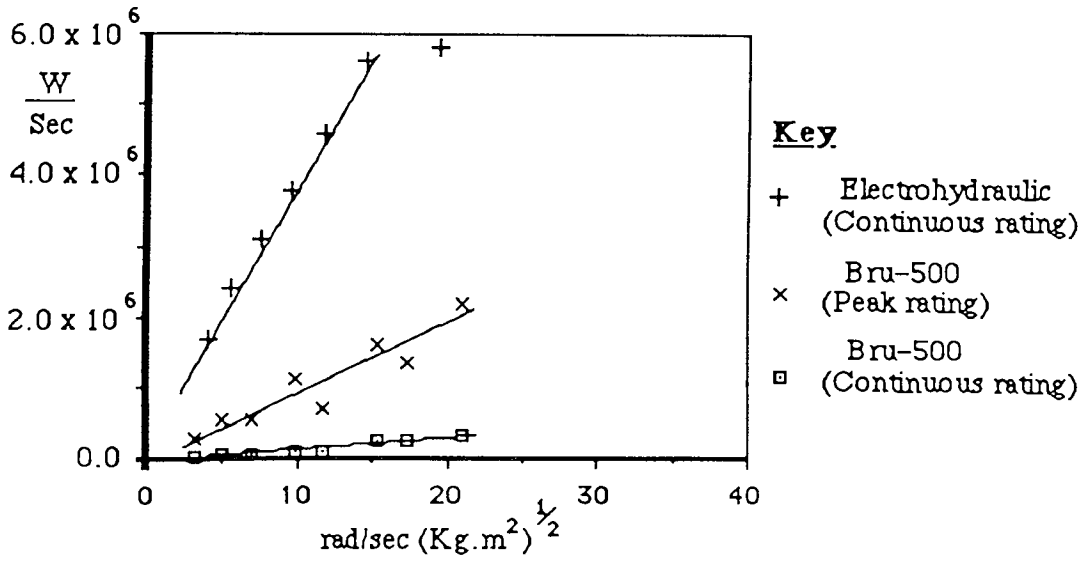


Figure 7.2. Driveability ratings of electrohydraulic and brushless d.c. servodrives.

Figure 7.2 demonstrates that the driveability rating of electrohydraulic servodrives is significantly superior to that of the brushless d.c. motors. Electrohydraulic motors have fallen out of favour because of the difficulties of implementing large

bandwidth satisfactorily damped servos, due to the need to provide pressure or acceleration feedback. In addition, the maintenance costs can be significant, particularly if the motors operate in an environment which can contaminate the hydraulic fluid or the moving parts. However, electrohydraulic drives should not be completely disregarded and efforts should be made towards improving their suitability for processing machinery application.

7.4. The simulation of software coupled dynamic systems.

The multiple actuator simulation models described in chapter 4 enable the software coupling and control algorithms to be developed and tested without endangering actuator hardware. This form of simulation could be extended to become a powerful design tool for the designer of machinery. Simulation models can be developed of actuator performance, which can then be combined to give very detailed models of independent actuators machines, which themselves can then be combined to give detailed models of integrated manufacturing units. This form of detailed plant model could be accessed by the plant controller to enable the effect of product or process changes before implementing such changes on–line. In addition, such a detailed simulation could serve as a reference model for manufacturing process control and analysis.

In the form described in chapter 4, the synchronised actuator models are not user–friendly. It requires an experienced user of ACSL to develop the simulation models, which can involve many hours work. However, it is the ACSL programming language which is complex to use. This is a control system simulation package which enables the user to have access to the numerical integration and sorting algorithms which perform the simulation run. The user of an independent actuators machinery design package would not require such access to the numerical simulation algorithms, so it would be a relatively straight forward task to develop a more user friendly machine design package. Simulation blocks could be built up of actuator modules, which could be combined using, for example, an icon–driven software package, with which the customer could develop models of his production plant. The implementation of such a package would require very little additional research to be carried out, since icon driven software and numerical simulation methods are readily available. Such a package would be most useful for the designers and users of independent actuators machinery.

7.5. Achieving high production speeds with flexible machinery.

The design of flexible high–speed machinery can be broken down into two main problems. The first is the introduction of flexibility, the second is the achievement of high production speeds. Traditionally designed machinery can be classified as high–speed, but it is inflexible. The independent actuators application that has been implemented in the laboratory is very flexible, but it can not operate at high speeds, particularly if the model arbor drum were to be replaced by the actual mechanism. The reason for this lower speed operation is that the drives are being retrofitted to mechanisms and motions which were originally designed within the constraints of cam drives. These do not produce the optimum conditions for independent actuator driving.

The energy which must be delivered to an electromechanical actuator to enable it to drive a function is proportional to the root of the load, directly proportional to the magnitude of the motion and proportional to the square of the frequency of the motion. The cam driven intermittent functions which have been considered in this work have large inertias and are required to be moved through large magnitude perturbations at high frequencies. The resulting energy demands that are placed on indpendent drives when attempting to actuate the motions are very large, which is the reason for the current overload and saturation problems which have occured during this research programme. As a result of these problems, the drives are not capable of actuating the functions at high operating frequencies.

To overcome these problems, it will be necessary to redesign the machine functions to suit the capabilities of the drives. The mechanisms should be designed to have smaller inertias. This should include the use of materials other than steel and aluminium, which are the principal constituents of existing machines. The profiles which the drives follow should be designed to have small magnitude and low frequency perturbations. High product throughputs can be achieved by using mechanisms and profiles which require the motions to have unidirectional motion, with small magnitude perturbations superimposed onto a constant velocity profile. An example of such a motion can be seen in the Continuous Belt and Touching Belt system in section 4.6. The Touching Belt has a low inertia and follows a low magnitude asynchronous profile which is superimposed onto a continuous velocity motion. The energy requirement of the disturbance is low, but a high product throughput is maintained by the speed of the belt. This mechanism suits the requirements of the drives, whereas the retrofitted intermittent motions of the laboratory work do not.

By configuring the machine functions to suit the capabilities of the actuators, higher machine speeds will be attainable than can be reached by retrofitted intermittent functions. These can be controlled by a flexible controller of the form described above and developed in chapter 4, which will enable flexible high–speed machinery to be produced.

7.6. The conclusions of the work.

The principal conclusion of this research programme is that flexible high–speed machinery incorporating independently actuated mechanisms under software control can be built, but that additional work must be performed before such machinery can form part of a viable manufacturing system. The additional work which must be performed does not involve the development of new technology, but existing technology must be improved and new methods of applying the technology must be developed.

Specific statements may be made relating to the work which has been performed and the additional work which is required. These are:

(1) The use of independenlty driven, software couple actuators to perform product processing functions introduces flexibility into product processing machinery. However, the performance of the machinery will be restricted if a retrofit design approach is adopted, whereby the independent actuators are required to perform functions which were originally developed to suit cam–driven functions. The operating speed of the machine will be limited by the energy demands which high frequency operation places on the drives. Higher operating speeds will be attainable if the machine functions are designed to suit the constraints of the drives. Mechanisms should have low inertias and dynamic perturbations should have small magnitudes and low frequencies.

(2) A distributed controller network which provides actuator autonomy can be used for a flexible controller implementation. However, it will be necessary to develop user–friendly methods of programming large distributed machine controller networks if the controller is to be expandable to large machine and production plant control systems.

(3) Brushless d.c. motors can be used to control flexible independent actuators. Their specifications are very suitable for the requirements of independent function actuation. However, the technology has not yet been sufficiently developed for machine drive implementation. Improvements are required in the

areas of reliability, driver control algorithms and software, electromechanical construction and manufacturers testing procedures.

(4) Simulation models of actustor systems, which have been proven to be accurate in system identification tests, can be used to test software coupling and control algorithms. These can enable very detailed models of independent actuators machines and production plants to be developed. If further developed, this technique could form a very powerful tool for the design and off–line analysis of flexible machine systems.

Independent actuators machinery is revolutionary technology. It incorporates the most modern electric servodrives and distributed control hardware. For the technology to be succesfully implemented, it will be necessary to further develop the servodrives, the control methods employed and the physical construction and operation of the machine functions. However, although the technology is revolutionary, it is attainable. Great benefits of manufacturing flexibility will result when it is finally implemented. The study which is described in this thesis has demonstrated the basic concept of independent actuators machinery is sound, with a degree of flexibility being introduced which can not be found in traditional machine designs. Despite the many failures of drive hardware which occured in the laboratory, no major problem has been identified which it is believed will prevent the independent actuators concept being succesfully implemented.

Appendix 1.

Brushless d.c. motor selection procedure development.

A.1.1. Introduction.

The selection procedure was developed to enable the optimum brushless d.c. motor to be obtained for actuating intermittent mechanical functions. The requirements are that a specified load is required to follow a specified intermittent motion profile, with pauses between motions. The drive must be selected which will obtain the stiffest response to the profile. In addition, weighting should be given in the selection towards drives which will use least energy in actuating the machine function. This will minimise the energy requirements of the complete machine system.

A.1.2. The basic motor power rate requirement.

A representation of a motor driving an angular load is given in figure A.1.1. The motor, of inertia J_m, drives the inertia load J_l, through a drive ratio N.

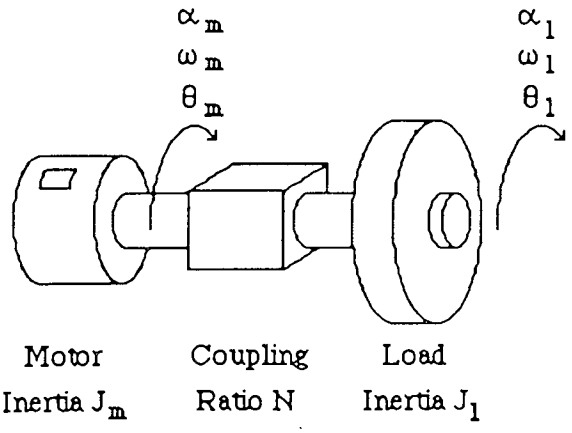


Figure A.1.1. Angular motion drive system.

The torque which is required to be generated by a motor to drive the load J_l with an acceleration α_l is given by:

$$T_m = N.\alpha_l.\left\{ J_m + \frac{J_l}{N^2} \right\} \qquad \{A.1.1\}$$

Where T_m is the required motor torque. The moment of inertia of the drive ratio is assumed to be small, compared to the load, or to be included within the load inertia, the friction torque is assumed to be small compared to the motor torque. The drive ratio should be selected to minimise the motor torque requirement. The optimum value can be found by differentiating equation A.1.1 with respect to N:

$$\frac{d.T_m}{d\,N} = \alpha_l.J_m - \frac{\alpha_l.J_l}{N^2}$$

The turning points of this equation can be found by equating the derivative to zero. This leads to:

$$N = \sqrt{\frac{J_l}{J_m}} \qquad \{A.1.2\}$$

To check that this is a minimum point, the second derivative of equation A.1.1 is:

$$\frac{d^2.T_m}{d\,N^2} = \frac{2.\alpha_l.J_l}{N^3} \qquad (= \text{POSITIVE})$$

Thus the value for N given by equation A.1.2 yields the minimum torque requirement for a motor driving the load. By a similar analysis to this, Tal (21) has have shown that this value for N also minimises the heat generation within a direct current (d.c.) motor driving incremental loads.

The torque equation, A.1.1, may be combined with the optimum drive ratio equation, A.1.2, to give:

$$T_m = 2.\alpha_l.\sqrt{J_l.J_m} \qquad \{A.1.3\}$$

squaring equation A.1.3 and rearranging gives the power rate requirement for the drive:

$$\frac{T_m^{\,2}}{J_m} = 4\,.\,\alpha_l^{\,2}\,.\,J_l \qquad \{A.1.4\}$$

The terms to the left of equation A.1.4 describe the power rate requirement for a motor driving the angular load. Power rate is defined as the rate of change of mechanical power with time

$$\text{Power Rate} = \frac{T^2}{J} = \frac{d.(P_m)}{dt} \qquad (\text{Kw / sec}) \qquad \{A.1.5\}$$

Arnold (25) and Floresta (26) describe the importance of power rate in incremental motion drive selection. A drive's power rate indicates its ability to apply power to drive a load. Equation A.1.4 demonstrates that for a motor to be capable of driving a load, the motor's power rate must be greater that four times that of the load:

$$\frac{T_m^{\,2}}{J_m} = \alpha_m^{\,2}\,.\,J_m = 4\,.\,\alpha_l^{\,2}\,.\,J_l \qquad \{A.1.6\}$$

This result agrees with Newton's (24) torque condition. However, Newton's derivation adopts a normalised drive ratio, with the normalisation base being that

given by the equation A.1.2, without deriving the optimum ratio expression itself. The relationship between the drive and load power rates under optimal drive conditions can be more clearly seen in the derivation given here.

Equation A.1.6 is one of the two fundamental requirements for assessing a motor's suitability for driving a load. A similar power rate relationship may be derived for linear loads. Figure A.1.2 shows the arrangement for a motor driving a linear load through a carriage drive.

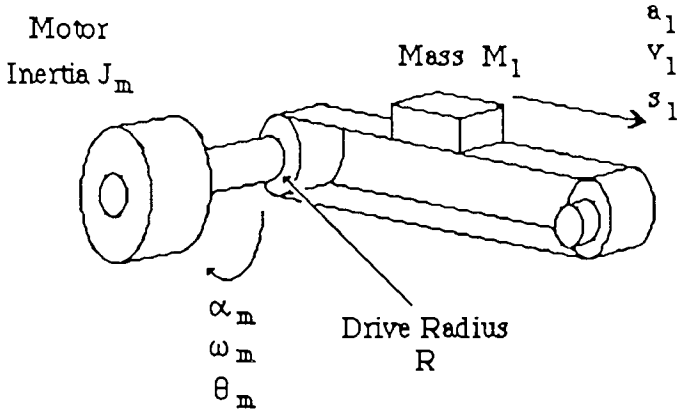


Figure A.1.2. Linear motion drive system.

The motor with inertia J_m drives the load of mass m_l through the drive radius R. The torque that is required to be generated by the motor to accelerate the load at an acceleration a_l is given by:

$$T_m = .\frac{a_l}{R}.\left\{J_m + M_l.R^2\right\} \qquad \{A.1.7\}$$

By a similar analysis as is given above for the drive ratio N, the optimum value of R to minimise the torque required to be generated by the drive motor is given by:

$$R = \sqrt{\frac{J_m}{M_l}} \qquad \{A.1.8\}$$

This expression can be combined with equation A.1.7 to give:

$$T_m = 2.a_l\sqrt{J_m.M_l} \qquad \{A.1.9\}$$

Rearranging and squaring gives:

$$\frac{T_m^{\,2}}{J_m} = 4\,.a_l^{\,2}.\,M_l \qquad \{A.1.10\}$$

Thus, as with angular loads, for a motor to be capable of driving the load it must have a power rate rating which is greater than four time that of the load.

A.1.3. The basic motor velocity requirement.

For angular loads the motor velocity ω_m, will be related to the load velocity ω_l through the drive ratio N:

$$\omega_m = N.\omega_l \qquad \{A.1.11\}$$

Substituting for N from equation A.1.2 and rearranging:

$$\omega_m.\sqrt{J_m} = .\omega_l.\sqrt{J_l} \qquad \{A.1.12\}$$

Since the inertia and angular velocity of the load are specified, equation A.1.12 sets the velocity:inertia requirement of the drive motor. For linear loads the motor velocity is related to the load velocity V_l through the drive radius R:

$$\omega_m = \frac{V_l}{R} \qquad \{A.1.13\}$$

Substituting for R from equation A.1.8 and rearranging:

$$\omega_m.\sqrt{J_m} = .V_l.\sqrt{M_l} \qquad \{A.1.14\}$$

Thus, as with angular loads, since the mass and required linear velocity of the load are specified, equation A.1.14 sets the velocity:inertia requirement of the drive motor.

A.1.4. The stiffness requirement.

The term stiffness, when applied to a drive, describes the speed of response to a disturbance load or to a motion profile. The drive's stiffness can be related to spring stiffness. A stiff spring responds quickly to a disturbance and has a high natural frequency. A drive's transient reponse to a step input is governed by the equation:

$$\omega(t) = a.\, e^{-\zeta.\omega_n.t}.\, \sin(\omega_d t - \phi) \qquad \{A.1.15\}$$

Thus a drive having a high natural frequency will give a fast response to the input. It was useful, therefore, to determine the parameters of brushless d.c. drives which indicate their natural frequencies, so that these could be used to indicate which drives would give the stiffest reponse.

Figure A.1.3 shows the block diagram of a brushless d.c. motor (14).

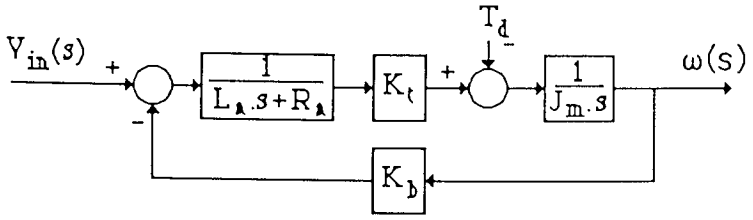


Figure A.1.3. Brushless d.c. motor block diagram.

The motor's transfer function is

$$\frac{\omega(s)}{V_{in}(s)} = \frac{K_t}{J_m.L_a.s^2 + J_m.R_a.s + K_t.K_b} \qquad \{A.1.16\}$$

This can be expressed in terms of the natural frequency, ω_n and damping ratio, ζ of the motor

$$\frac{\omega(s)}{V_{in}(s)} = \frac{K.\omega_n^2}{s^2 + 2.\zeta.\omega_n.s + \omega_n^2}$$

ζ and ω_n can be expressed in terms of the motor's electrical time constant, τ_e and mechanical time constant, τ_m.

$$\omega_n = \sqrt{\frac{1}{\tau_e.\tau_m}} \qquad \zeta = \frac{1}{2}\sqrt{\frac{\tau_m}{\tau_e}} \qquad \{A.1.17\}$$

where

$$\tau_m = \frac{J_m.R_a}{K_t^2} \qquad \tau_e = \frac{L_a}{R_a} \qquad \{A.1.18\}$$

The use of rare–earth magnets in the motors, giving small inertias and large torque constants, has led to the development of large drives in which τ_e is very much larger than τ_m. As a result, many large brushless d.c. drives, of rating 6–8 Kw and above, have inherent damping ratios in the range 0.2 to 0.3, or less. Therefore controllers for brushless motors include current compensation to increase the damping ratio, as shown in figure A.1.4. In this case, the compensated motor's transfer function is

$$\frac{\omega(s)}{V_{in}(s)} = \frac{K_t}{J_m.L_a.s^2 + J_m.(R_a + K_i).s + K_t.K_b} \qquad \{A.1.19\}$$

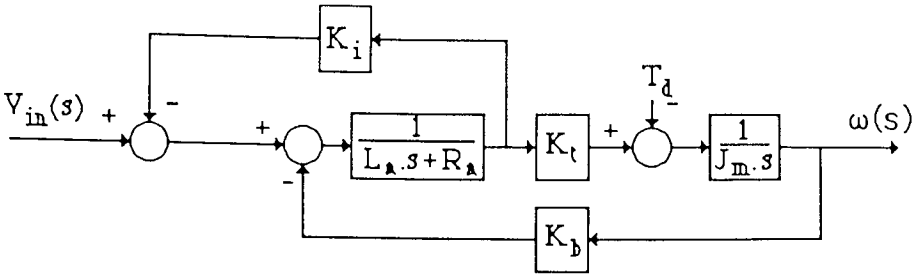


Figure A.1.4. Brushless d.c. motor with current feedback.

Comparison between equation A.1.19 and equation A.1.16 shows that the current feedback term, K_i affects the resistance term in the transfer function.Increasing K_i increases τ_m.and decreases τ_e.by equal proportions, since τ_m.is proportional to R_a and τ_e.is inversely proportional to R_a. Thus K_i affects the damping ratio of the motor, but has no effect on the natural frequency. Therefore the expression for natural frequency given in equation A.1.17 enables a ready pointer to the stiffness of the motor. Motors having small values of the product $\tau_m.\tau_e$.will have high natural frequencies, so will be expected to give a stiff response to a motion profile.

A.1.5. The energy consumption requirement.

The energy required by a motor to drive a load is given by

$$W_m = \int_0^t P_m . \, dt \qquad \{A.1.20\}$$

Where W_m is the energy consumption of the motor and P_m is the instantaneous power consumption of the motor. The motor which requires the least energy to move the load will be that which consumes the least power. The instantaneous power consumption consists of three principal parts: P_{load} , the power required to drive the motor and the load, P_f , the power required to overcome friction and P_{diss} , the power which is dissipated from the motor windings.

$$P_m = P_{load} + P_f + P_{diss} \qquad \{A.1.21\}$$

Each of these terms may be expressed in terms of load and motor parameters. The power required to drive the motor and an angular load is given by

$$P_{load} = T_m.\omega_m = \left\{ N.\alpha_l.\left\{ J_m + \frac{J_l}{N^2} \right\} + \frac{T_f}{N} \right\}.\{N.\omega_l\} \qquad \{A.1.22\}$$

Where T_f is the load friction torque. The drive ratio, N can be substituted from equation A.1.2 for the optimum value

$$P_{load} = \left\{ \sqrt{\frac{J_l}{J_m}}.\alpha_l.\{2.J_m\} + T_f\sqrt{\frac{J_m}{J_l}} \right\}.\left\{ \sqrt{\frac{J_l}{J_m}}.\omega_l \right\}$$

This leads to

$$P_{load} = \left\{2.J_1 . \alpha_1 . \omega_1\right\} + \left\{T_f . \omega_1\right\} \qquad \{A.1.23\}$$

The power which is required to overcome friction is the product of the friction torque and the load speed

$$P_f = T_f . \omega_1 \qquad \{A.1.24\}$$

The power which is dissipated in the windings is given by

$$P_{diss} = I_a^2 . R_a = \left\{\frac{T_m}{K_t}\right\}^2 . R_a$$

$$P_{diss} = \frac{R_a}{K_t^2} . \left\{ N.\alpha_1 . \left\{ J_m + \frac{J_1}{N^2} \right\} + \frac{T_f}{N} \right\}^2$$

Substituting for N from equation A.1.2 and expanding this expression leads to

$$P_{diss} = \frac{J_m . R_a}{K_t^2} . \left\{ 4.\alpha_1^2 . J_1 + \frac{T_f^2}{J_1} + 4.T_f \alpha_1 \right\} \qquad \{A.1.25\}$$

Equations A.1.23, A.1.24 and A.1.25 can be substituted into equation A.1.21, to give an expression for the power consumption of the motor in terms of motor and load parameters

$$P_{load} = 2.J_1 . \alpha_1 . \omega_1 + 2.T_f \omega_1 + \frac{J_m . R_a}{K_t^2} . \left\{ 4.\alpha_1^2 . J_1 + \frac{T_f^2}{J_1} + 4.T_f \alpha_1 \right\} \qquad \{A.1.26\}$$

All but three of the terms to the right of equation A.1.26 are parameters of the specified load. The three motor parameters can be seen in equation A.1.18 to be the mechanical time constant of the motor. Thus a motor having a small mechanical time constant will consume less power in moving a load. So the optimum motor selection in terms of energy consumption will be that having the smallest mechanical time constant, τ_m.

A.1.6. The intermittent function brushless d.c. motor selection procedure.

The intermittent machine drive selection procedure was developed to determine which brushless d.c. motor would be the optimum selection for a specified application. It should be noted that the procedure was developed to enable a selection to be made by a user with no previous drives experience. In practice, it would be expected that the user would introduce further criteria into the selection, such as hands–on experience of reliability, availability of motors, the environment of the application or the cost of the motors.

The selection procedure requires that the load requirements be specified. These are:

(i) The inertia of the load

(ii) The peak and RMS accelerations of the load

(iii) The peak velocity of the load.

The procedure for selecting a motor for the drive is as follows:

1) Determine required motor power rate.

The required motor power rate is given by equation A.1.4, for angular loads and by equation A.1.10 for linear loads. These should be determined for both the peak and RMS acceleration requirements of the load.

2) Determine motor velocity:inertia requirement.

For angular loads the required velocity:inertia performance of the motor is given by equation A.1.12. For linear loads the velocity:inertia requirement is given by equation A.1.14.

3) Plot motor requirement on driveability chart.

The power rate and velocity:inertia requirements should be plotted onto dirveability charts using both the peak and RMS power rate requirements. The dives should be plotted on the chart using manufacturers data. Using this, the initial selection should determine which motors are physically capable of driving the load.

4) Selecting a drive on the basis of stiffness.

The drive which will give the stiffest response will be that having the smallest value of the product of the electrical and mechanical time constants, $.\tau_e.\tau_m$.

5) Selecting a drive on the basis of energy consumption.

The drive which will have the smallest energy consumption will be that having the smallest mechanical time constant, τ_m.

A.1.7. The minimum energy trapezoidal velocity profile.

The trapezoidal velocity profile is widely used for incremental motion control systems. It is straight forward to implement using a digital controller and requires very little computation time for particular trapezoids to be calculated. As a result, many commercial controllers use trapezoidal motion profiles for incremental drives.

Throughout Molins' machine range there are many examples of incremental motions, the arbor drum being an example of such a system. One of the design objectives in independent actuators' machines is the minimising of the total energy consumption of the machine. Therefore it is necessary that the motion profiles

followed by the machine functions be optimised for minimum energy consumption. It has been shown by Tal (21) that the optimum trapezoidal profile for minimum energy dissipation is one in which the acceleration time, t_1, the slew time, t_2 and the braking time, t_3 are all equal. As a result, the $t_1 = t_2 = t_3$ trapezoidal profile has been widely adopted as the standard profile for incremental motion systems. However, equation A.1.26 demonstrates that the dissipated energy is only part of the total energy consumption of the drive. In this section it is shown that the $t_1 = t_2 = t_3$ trapezoidal profile is not the optimum for minimum energy consumption and that the optimum trapezoidal profile can be determined using load and motor parameters.

Figure A.1.5 shows the acceleration, velocity and position profiles of a trapezoidal velocity increment.

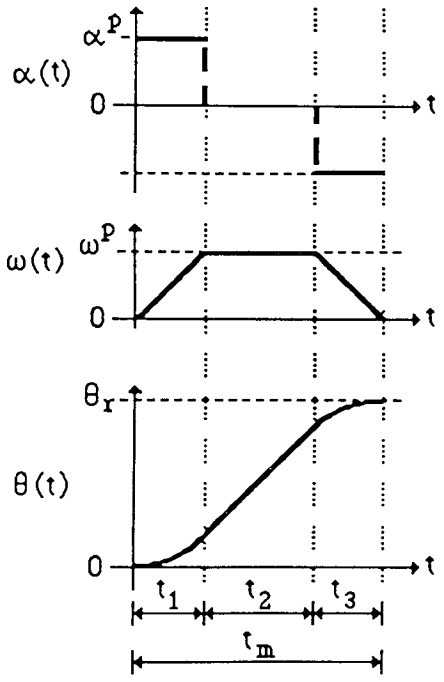


Figure A.1.5. Trapezoidal velocity incremental profile.

The specification for an incremental motion is that the load is required to be moved through a distance, θ_r in a time, t_m with a dwell period, t_d between increments. The design of a trapezoidal profile involves determining the magnitude of the acceleration, α^p and the profile segment acceleration, slew and braking times, t_1, t_2 and t_3 which will give the required displacement in the required time. For designing the optimum profile for minimum energy consumption, it is necessary to define the profile segment times in terms which enable substitution into equation A.1.26 for power consumption. This can then be integrated in accordance with equation A.1.20, within the time limits of the profile, to determine an expression for the energy consumed in terms of t_1, t_2 and t_3, which can then be evaluated to determine the optimum segment times relationship.

To perform the increment, the drive must supply sufficient kinetic energy to accelerate the system to the peak velocity and to deccelerate the system to the incremented stationary position. Therefore, a high peak velocity will result in a high kinetic energy requirement. The total area swept out by the velocity profile gives the distance moved, which is fixed by the specification. So the profile must be designed to encompass the specified area while limiting the maximum velocity attained. To accomplish this, the profile should be symmetrical, with the acceleration and braking times equal and the magnitude of the acceleration and braking should be large, to limit the maximum velocity attained

The total displacement of the profile is given by

$$\theta_r = \left\{\frac{\alpha^p.t_1^2}{2}\right\} + \left\{\alpha^p.t_1.t_2\right\} + \left\{\alpha^p.t_1.t_3 - \frac{\alpha^p.t_3^2}{2}\right\} \qquad \{A.1.27\}$$

By symmetry, $t_1 = t_3$ and $t_2 = (t_m - t_1)$. Therefore this expression becomes

$$\theta_r = \alpha^p.t_1.\left\{t_m - t_1\right\} \qquad \{A.1.28\}$$

This can be rewritten for α^p

$$\alpha^p = \frac{\theta_r}{t_1.\left\{t_m - t_1\right\}} \qquad \{A.1.29\}$$

The peak velocity is attained at the end of the acceleration segment

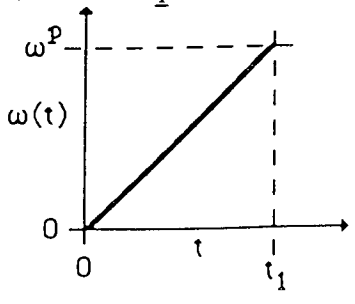
$$\omega^p = \alpha^p.t_1 \qquad \{A.1.30\}$$

The expression for the energy consumed in performing the increment is obtained by integrating equation A.1.26 in accordance with equation A.1.20

$$W_m = 2.J_1\int_0^t \alpha_1.\omega_1.dt + 2.T_f\int_0^t \omega_1.dt + 4.\tau_m.J_1\int_0^t \alpha_1^2.dt + \frac{\tau_m.T_f^2}{J_1}\int_0^t dt + 4.\tau_m.T_f\int_0^t \alpha_1.dt \quad \{A.1.3$$

Equation A.1.31 will be equated between the limits $0 \le t < t_1$, $t_1 \le t < (t_1 + t_2)$ and $(t_1 + t_2) \le t \le t_m$, to obtain the expression for the energy consumption.

(1) $0 \le t < t_1$



$\omega(t) = \alpha^p.t$

$\alpha(t) = \alpha^p$ = Constant

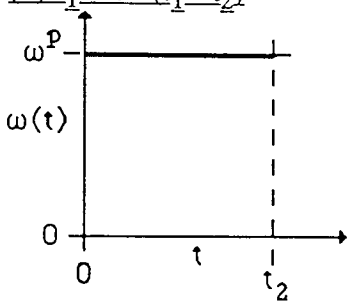
$$W_m = 2.J_1.\alpha^p\left[\frac{\alpha^p.t^2}{2}\right]_0^{t_1} + 2.T_f\left[\frac{\alpha^p.t^2}{2}\right]_0^{t_1} + 4.\tau_m.J_1\alpha^{p^2}[t]_0^{t_1} + \frac{\tau_m.T_f^2}{J_1}[t]_0^{t_1} + 4.\tau_m.T_f.\alpha^p[t]_0^{t_1}$$

$$W_m = J_1.\alpha^{p^2}.t_1^2 + T_f.\alpha^p.t_1^2 + 4.\tau_m.J_1\alpha^{p^2}.t_1 + \frac{\tau_m.T_f^2.t_1}{J_1} + 4.\tau_m.T_f.\alpha^p.t_1$$

Substituting for α^p from equation A.1.29 leads to

$$W_m = \frac{\left[J_1.\theta_r^2 + T_f.\theta_r.t_1(t_m - t_1)\right]}{(t_m - t_1)^2} + \tau_m.\frac{\left[2.J_1.\theta_r + T_f.t_1(t_m - t_1)\right]^2}{J_1.t_1(t_m - t_1)^2} \qquad \{A.1.32\}$$

(2) $t_1 \le t < (t_1 + t_2)$



$\omega(t) = \alpha^p.t_1 = \text{Constant}$

$\alpha(t) = 0$

Since t_2 is a segment length and not an absolute value, the integral may be equated between the limits $0 \le t < t_2$, where the origin has been shifted to the end of the first segment.

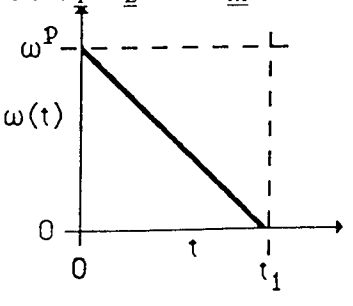
$$W_m = 2.T_f\left[\alpha^p.t_1.t\right]_0^{t_2} + \frac{\tau_m.T_f^2}{J_1}[t]_0^{t_2}$$

$$W_m = 2.T_f\alpha^p.t_1.t_2 + \frac{\tau_m.T_f^2.t_2}{J_1}$$

Substituting for α^p from equation A.1.29 leads to

$$W_m = \frac{2.T_f\theta_r.t_2}{(t_m - t_1)} + \frac{\tau_m.T_f^2.t_2}{J_1} \qquad \{A.1.33\}$$

(3) $(t_1 + t_2) \le t \le t_m$



$\omega^p = \alpha^p.t_1$

$\omega(t) = \omega^p - \alpha^p.t = \alpha^p(t_1 - t)$

$\alpha(t) = \alpha^p = \text{Constant}$

Note: Braking energy must be supplied by the motor. Therefore, the acceleration terms in the energy equation must be treated as positive. However, the friction torque assists braking, so the friction terms must be treated as negative.

Since t_3 is a segment length, not an absolute value and since by symmetry $t_3 = t_1$, the integral may be equated between the limits $0 \le t \le t_1$.

$$W_m = 2.J_1.\alpha^{p^2}\left[t_1.t - \frac{t^2}{2}\right]_0^{t_1} - 2.T_f\left[t_1.t - \frac{t^2}{2}\right]_0^{t_1} + 4.\tau_m.J_1\alpha^{p^2}[t]_0^{t_1} + \frac{\tau_m.T_f^2}{J_1}[t]_0^{t_1} - 4.\tau_m.T_f\alpha^{p}[t]_0^{t_1}$$

$$W_m = J_1.\alpha^{p^2}.t_1^2 - T_f\alpha^{p}.t_1^2 + 4.\tau_m.J_1\alpha^{p^2}.t_1 + \frac{\tau_m.T_f^2.t_1}{J_1} - 4.\tau_m.T_f\alpha^{p}.t_1$$

Substituting for α^p from equation A.1.29 leads to

$$W_m = \frac{\left[J_1.\theta_r^2 - T_f\theta_r t_1 (t_m - t_1)\right]}{(t_m - t_1)^2} + \tau_m.\frac{\left[2.J_1.\theta_r - T_f t_1 (t_m - t_1)\right]^2}{J_1.t_1 (t_m - t_1)^2} \qquad \{A.1.34\}$$

The equation describing energy consumption per trapezoidal increment is obtained by adding equations A.1.32, A.1.33 and A.1.34. This equates to give

$$W_m = \frac{2.J_1.\theta_r^2 + 2.T_f\theta_r t_2 (t_m - t_1)}{(t_m - t_1)^2} + \tau_m.\frac{\left[8.J_1^2.\theta_r^2 + T_f^2.t_1.t_m.(t_m - t_1)^2\right]}{J_1.t_1 (t_m - t_1)^2} \qquad \{A.1.3.5\}$$

For most independent actuators incremental applications, the friction torque is very small. Therefore this expression can be approximated by

$$W_m = \frac{2.J_1.\theta_r^2}{(t_m - t_1)^2} + \frac{8.\tau_m.J_1.\theta_r^2}{t_1 (t_m - t_1)^2} \qquad \{A.1.36\}$$

Of the two terms to the right of equation A.1.36, the first gives the total energy required to move the system through the trapezoidal profile and the second gives the amount of energy dissipated through the windings during the increment. This expression is to be used to design the optimum trapezoidal profile, therefore it is important that it be shown to be correct. Since the profile assumes constant, equal magnitude acceleration and deceleration torques, it is straight forward to check equation A.1.36 using an energy approach. The total kinetic anergy and dissipted energy during the acceleration and deceleration segments is given by

$$W = 2.\left\{\frac{1}{2}.J.\omega_m^2\right\} + \left\{I_a^2.R_a\right\}.(t_1 + t_3)$$

Where $\omega_m = N.\omega^p$, J is the driven inertia reflected to the motor, I_a is the current in thc motor windings and for a symetrical profile, $t_1 = t_3$. Thus

$$W = \left\{J_m + \frac{J_1}{N^2}\right\}.N^2.\omega^{p^2} + \left\{\frac{T_m^2}{K_t^2}.R_a\right\}.(2.t_1)$$

Substituting for N from equation A.1.2 and for T_m from equation A.1.3 gives

$$W = \{2.J_m\}.\left\{\frac{J_l}{J_m}\right\}.\omega^{p^2} + \left\{4.\alpha^{p^2}.J_l.\frac{J_m.R_a}{K_t^2}\right\}.(2.t_1)$$

$$W = 2.J_l.\alpha^{p^2}.t_1^2 + 8.\alpha^{p^2}.J_l.\tau_m.t_1$$

Substituting for α^p from equation A.1.29 leads to

$$W = \frac{2.J_l.\theta_r^2}{(t_m - t_1)^2} + \frac{8.\tau_m.J_l.\theta_r^2}{t_1\,(t_m - t_1)^2}$$

Which confirms equation A.1.36. Tal demonstrated that the dissipation term is a minimum when $t_1 = t_2 = t_3$. The kinetic energy term will be a minimum when $t_1 = 0$, since t_m is fixed and t_1 can not be negative. The optimum value for t_1 will be a compromise between minimum dissipation and minimum kinetic energy. An expression for this value can be obtained by differentiation.

$$\frac{d.W_m}{d.t_1} = \frac{4.J_l.\theta_r^2.(t_1^2 - 2.\tau_m.(t_m - 3.t_1))}{t_1^2.(t_m - t_1)^3}$$

The turning points of this equation are found by equating to zero. i.e. maximum and minimum points occur when

$$t_1^2 + 6.\tau_m.t_1 - 2.\tau_m.t_m = 0$$

This expression has positive and negative roots. The acceleration time,t_1 can not be negative. The second derivative of W_m shows that the positive root is the minimum value. Therefore the minimum energy consumption profile has acceleration and braking times given by,

$$t_1, t_3 = \frac{-6.\tau_m + \sqrt{36.\tau_m^2 + 8.\tau_m.t_m}}{2} \qquad \{A.1.37\}$$

Thus the optimum value for t_1 depends on the period of the motion and the time constant of the motor, so the profile must be designed to fit the particular motor being used for the drive if optimum energy usage is to be obtained.

The percentage energy savings that equation A.1.37 can introduce have been tested using a profile design program, written in pascal running on an Apple Macintosh desk top computer. The program solves equation A.1.36 for several values of t_1, including Tal's optimum dissipation value and the optimum value given by equation A.1.37. It is required that J_l , θ_r , t_m and τ_m be input. The percentage energy saving given by the optimum profile varies with the load and motor characteristics being considered. As an example, consider the case of an Electro–Craft B–2024C brushless d.c. motor driving the model arbor drum described in chapter 7. The inertia load of 0.1 Kg.m ^{2} is required to be incremented through $\pi/8$ radian in 55.5

msec, driven through an optimum ratio. Equation A.1.36 indicates that the minimum energy consumption profile has $t_1 = t_3 = 10.3$ msec. The minimum dissipation profile has $t_1 = t_2 = t_3 = 18.5$ msec.

Table A.1.1 shows the kinetic energy, dissipated energy and total energy consumption of the system following 55.5 msec, $\pi/8$ radian trapezoidal proflies, calculated using equation A.1.36. It can be seen that, although the $t_1 = 10.3$ msec profile has 20% greater energy dissipation than the $t_1 = 18.5$ msec profile, the total energy consumption is 19% smaller. Thus the profile given by equation A.1.37 introduces significant energy savings over the standard $t_1 = t_2 = t_3$ profile

t_1, t_3 (msec)	t_2 (msec)	Energy (Joules)		
		Kinetic	Dissipated	Total
8.2	39.1	13.8	14.5	28.3
9.5	36.5	14.6	13.2	27.8
10.3	34.9	15.1	12.6	27.7
12.3	30.9	16.5	11.6	28.1
15.0	25.5	18.7	10.8	29.5
17.7	20.1	21.5	10.5	32.0
18.5	18.5	22.5	10.5	33.0
20.4	14.7	24.9	10.6	35.5
23.1	9.3	29.2	10.9	40.1

Table A.1.1. Energy consumption of brushless d.c. motor driving trapezoidal profiles.

It should be noted that the increased energy dissipation of the optimum energy consumption profile will cause greater temperature rise in the motor. However, the construction of brushless d.c. motors gives them very efficient cooling characteristics. It will be necessary to ensure that the profile specification is within the peak and RMS driveability ratings of the motor.

Appendix 2

Development of the Moog motor position controller.

A.2.1. Introduction.

This Appendix describes the development of a position controller for the Moog 306–023 brushless d.c. servomotor. The decisions which led to the selection of the Moog motor and the initial testing of the motor in the laboratory are described in Chapter 6. The position controller described here was developed after the testing work of Chapter 6 had been performed, but was never implemented as part of the synchronised actuators system due to the unreliability of the Moog hardware. The work described below includes the design of the position control algorithms which would enable the motor to achieve the required motions, the testing of the algorithms using simulation and the design of the hardware for the position controller implementation.

A.2.2.Designing the position control algorithms.

The design of the position control algorithms was based on incorporating the continuous closed loop Moog motor/controller system in a unity negative feedback digital control system using a zero order hold. The control system block diagram is shown in figure A.2.1.

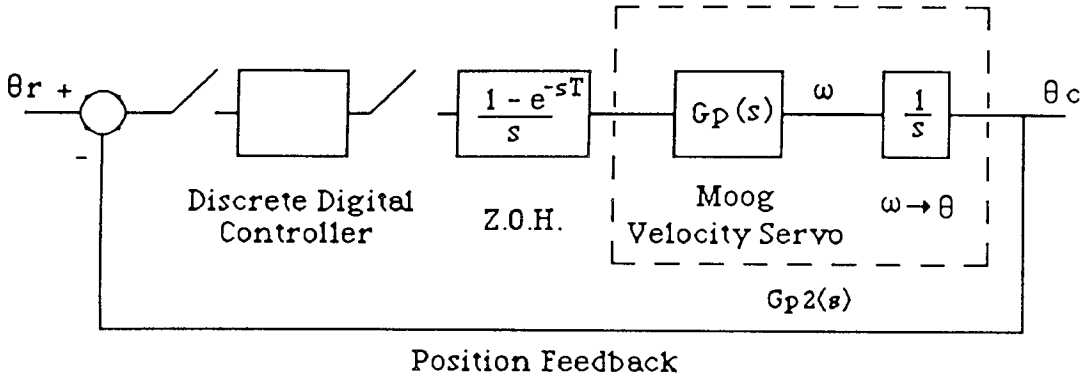


Figure A.2.1. Discrete Position Control Of Moog Velocity Servo.

The design of the compensation elements was carried out in the w–plane to allow ease of implementing the final controller as a discrete system. W–plane design procedures are covered in several reference texts, with books by Borrie (33) and Katz (41) including the frequency response design methods used here.

A.2.2.1.Deriving the w–plane response of the Moog system.

For the purposes of the controller design, the frequency response of the Moog motor driving a load inertia equivalent to the rotor inertia was used to represent the continuous system, Gp(jω). This was obtained in the laboratory and is shown in figure A.2.2. Beginning with this system frequency response, the steps towards obtaining the w–plane frequency response for use in the control algorithm design were as follows:

(i) Incorporate integrator into forward path.

Since the Moog frequency response is of the velocity loop, it was necessary to incorporate an integrator into the system reponse to form a plant frequency response. This then formed the frequency reponse of the Moog velocity servo and integrator The combined frequency response of the Moog motor plus integrator is,

$$\frac{G_{p}.(j\omega)}{(j\omega)} = G_{p2}.(j\omega)$$

$G_{p2}(j\omega)$ represents the plant to be controlled. The frequency response of $G_{p2}(j\omega)$ is shown in figure A.2.3.

(ii) Incorporate the effect of sampling.

It is necessary to introduce the effect of the sampling action of the controller. For the purposes of the design, a worst case sample length of 3.141 msec was used.The frequency response of such a sampler, $G_h(j\omega)$, is shown in figure A.2.4. Comparison of this plot with figure A.2.3 shows that the sampling effect is approximately that of a constant gain within the bandwidth of the Moog servo, $G_{p2}(j\omega)$, with more de-stabilising phase and gain effects beginning at frequencies of approximately four times the bandwidth. The combined frequency response of the Moog servo with integral effect plus sampling, $G_{p2}(j\omega).G_h(j\omega)$, is shown in figure A.2.5.

(iii) Obtain the sampled data frequency response.

The frequency response of the sampled system, $G^*(j\omega)$, is obtained from the expression

$$G^{*}.(j\omega) = \frac{1}{T}.\sum_{-\infty}^{\infty} G.(j\{\omega + K.\omega s\})$$

Given the choice of sample frequency being four times the bandwidth of the system, this expression can be approximated by fundamentals and the first upper and lower sidebands to,

$$G^{*}.(j\omega) \approx \frac{1}{T}.\left[G.(j\omega) + G.(j\{\omega - \omega s\}) + G.(j\{\omega + \omega s\})\right]$$

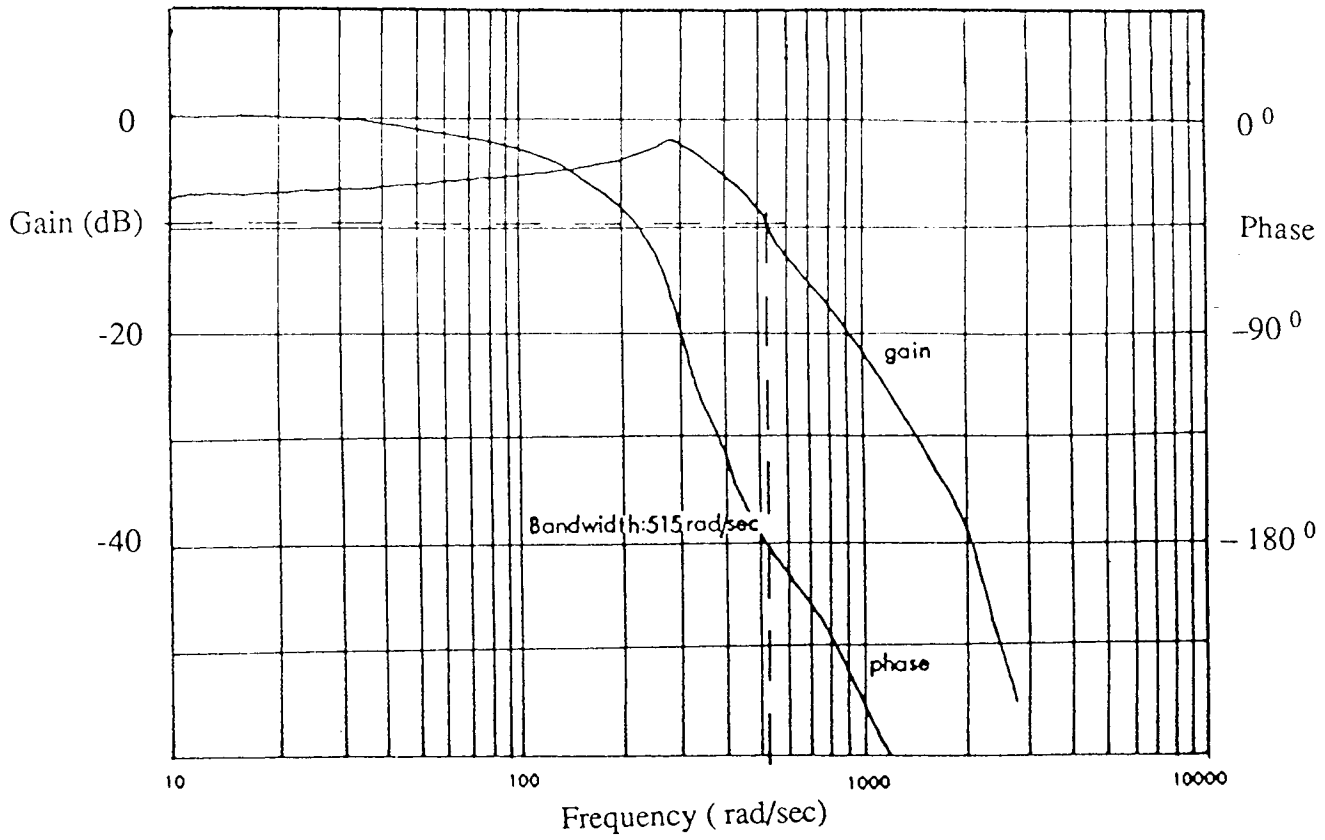


Figure A.2.2. Moog system frequency response, $G_p(j\omega)$.

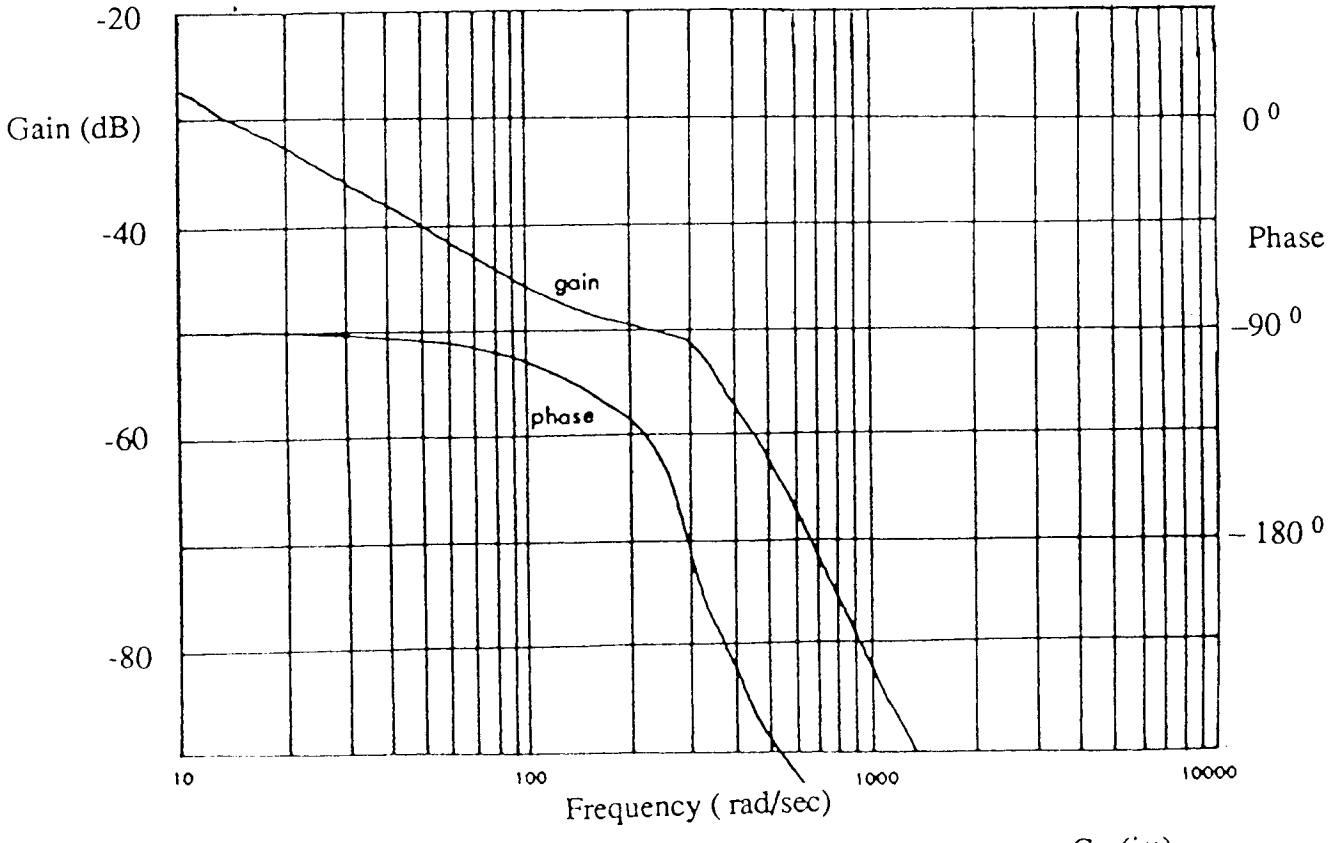


Figure A.2.3. Moog system plus integrator frequency response, $G_{p2}(j\omega)$.

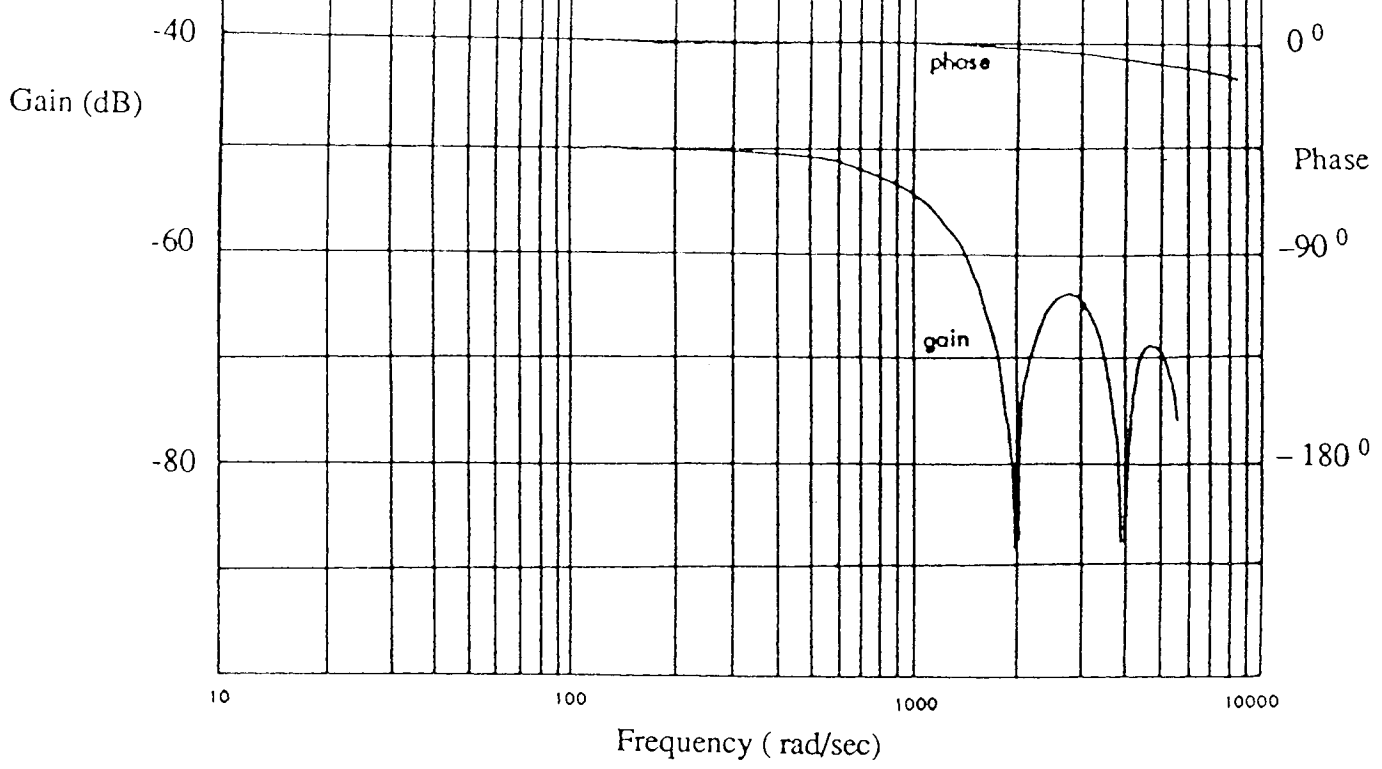


Figure A.2.4. Frequency response of zero order hold, $G_h(j\omega)$.

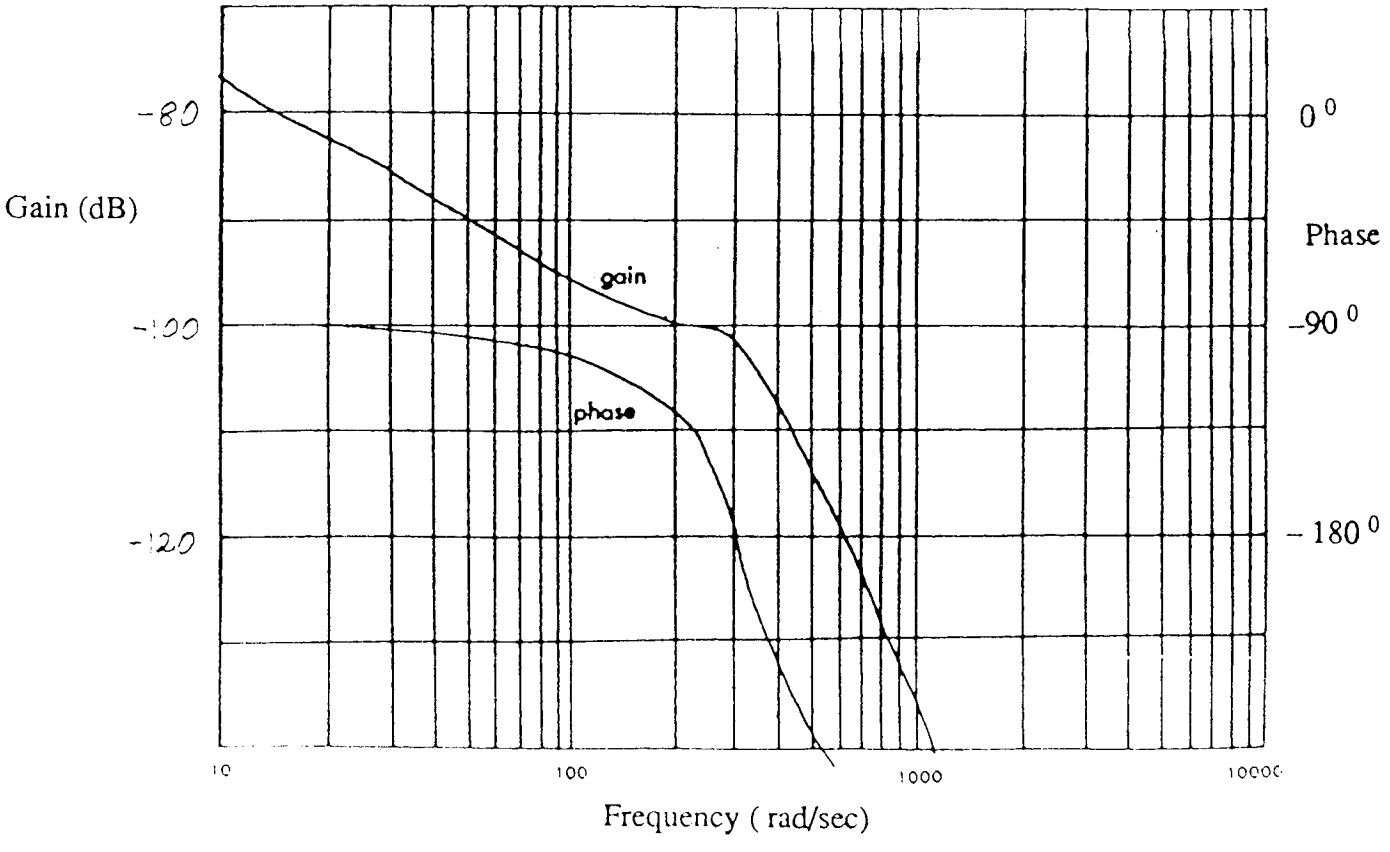


Figure A.2.5. Combined $G_{p2}(j\omega)$. $G_h(j\omega)$ frequency response.

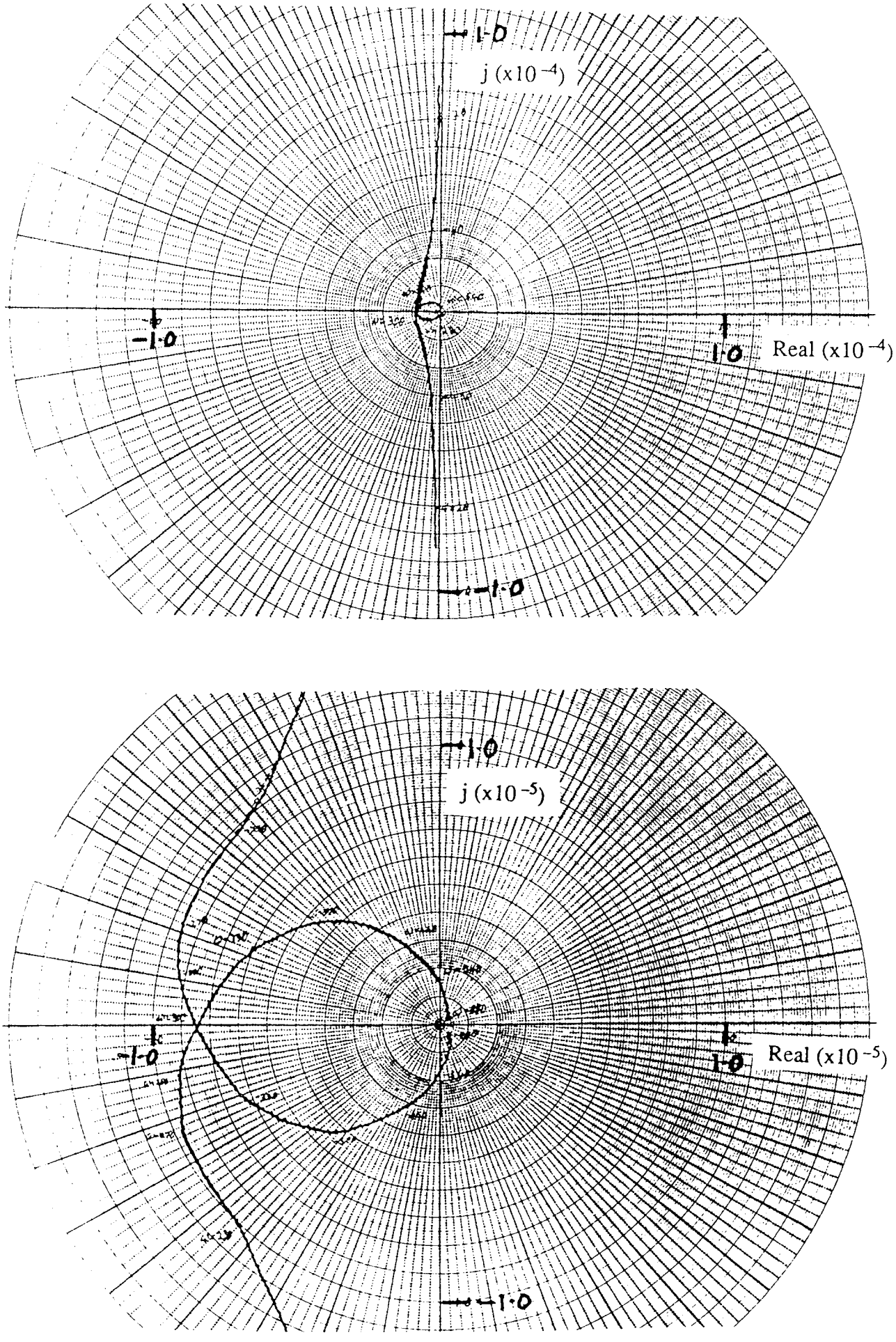


Figure A.2.6. Nyquist charts of $G_{p2}(j\omega)$. $G_h(j\omega)$.

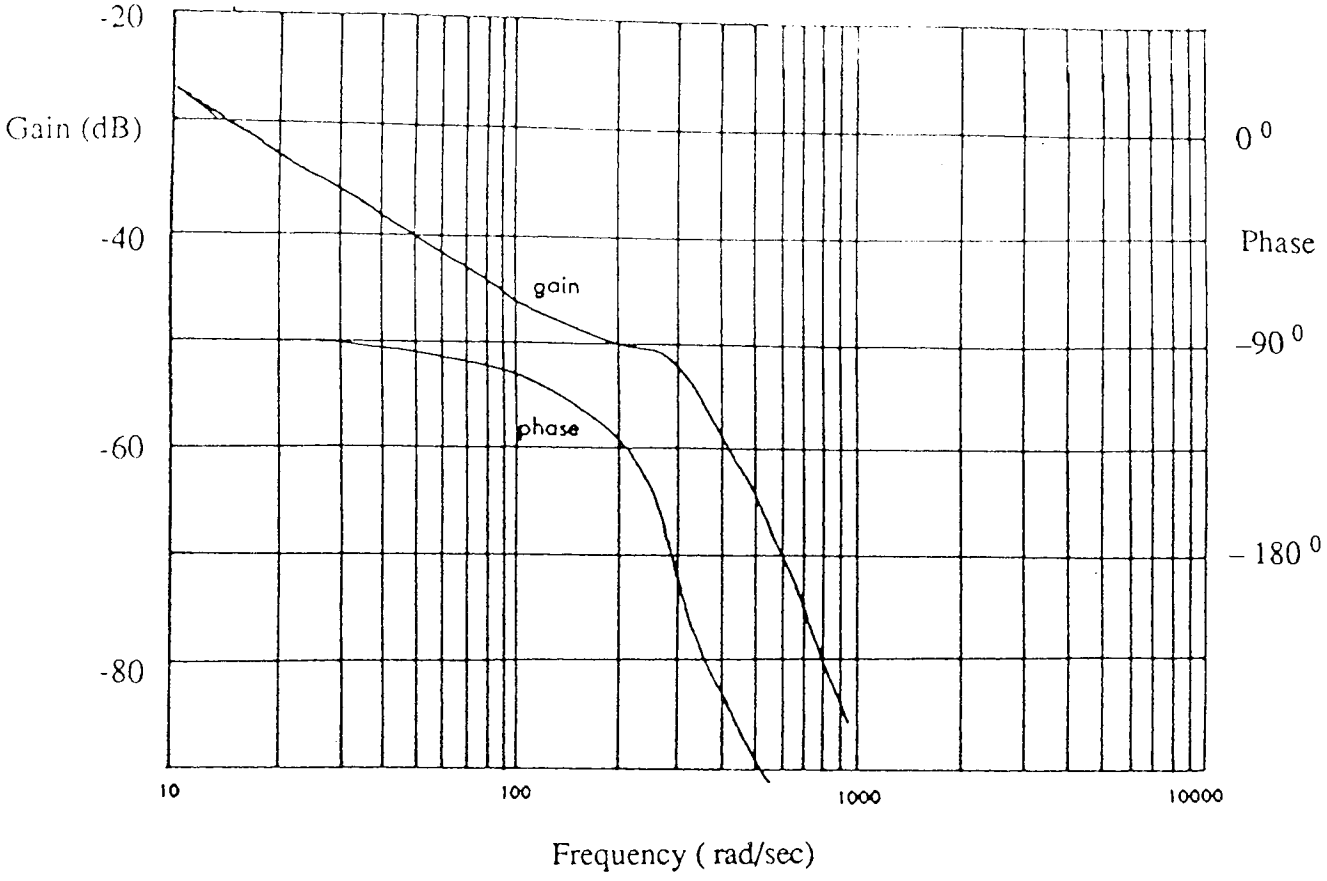


Figure A.2.7. Sampled system frequency response, $G^*(j\omega)$.

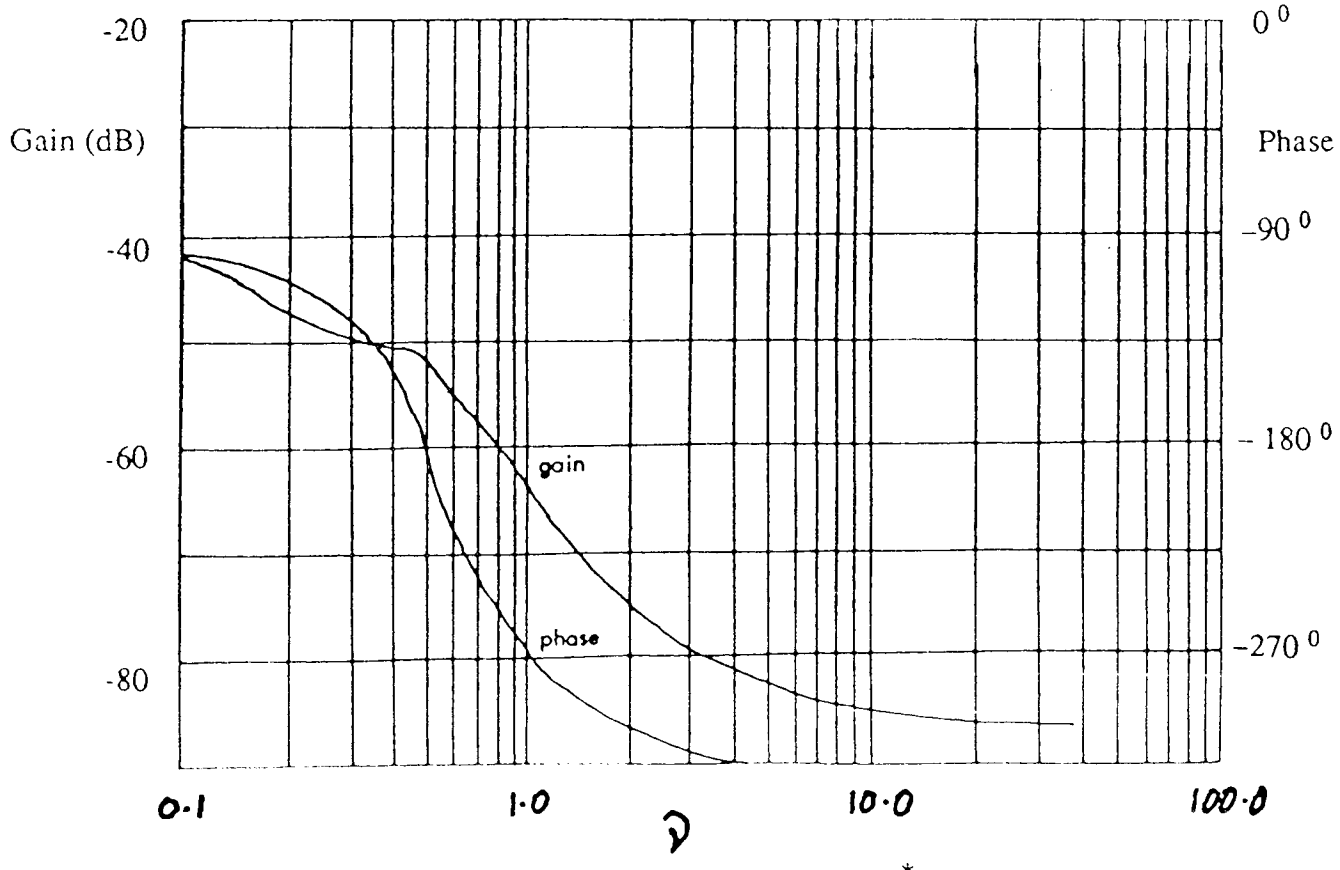


Figure A.2.8. W-plane frequency response, $G^*(j\upsilon)$.

Since $G^*(j\omega)$ is a vector sum, it may be calculated by taking the real and imaginary components of the absolute values of $G(j\omega)$ (Note that in our case, $G(j\omega) = G_{p2}(j\omega).G_h(j\omega)$). This process was aided by plotting the Nyquist chart of $G_{p2}(j\omega).G_h(j\omega)$, which is shown in figure A.2.6 (a) and (b). The frequency response of $G^*(j\omega)$ is plotted in figure A.2.7.

(iv) transpose the sampled data frequency response to the w–plane.

The frequency response of $G^*(j\omega)$ may be used for designing the controller, but it is usually more convenient to design controllers in the w-plane. The w-plane frequency response of our system, $G^*(j\upsilon)$ was obtained from $G^*(j\omega)$ by "warping" the frequency axis using the expression

$$j\upsilon = j.\ \mathrm{Tan}\left\{\frac{\omega.T}{2}\right\}$$

This gave the w-plane frequency response of figure A.2.8.

A.2.2.2. Defining the w–plane design objectives.

The design specification required the position controlled system to achieve the following performance criteria:

i) System stable.

ii) System to have zero steady state error to a step input.

iii) Maximum Overshoot < 1.3 %.

iv) 4% Settling Time <20 msec.

The w–plane characteristics which would enable these performance criteria to be met are described below.

i) Stability:

For stable system in the w-plane, ensure gain < 0dB for all phase < -180 °.

ii) Final Value:

For zero steady state error to a step input, require type 1 (zero step error, constant ramp error) system. System is already type 1, due to integrator in forward path.

For type 2 system (zero ramp error), we would require a $1/(z-1)$ term in the controller.

iii) Overshoot:

We require < 1.3 % overshoot. Thus at 180 ° phase we require a system open loop gain, G_{180} that ensures

$$\frac{G_{180}}{1 - G_{180}} < 1.013$$

Thus we require G_{180} = -6 dB gain margin.

iv) Settling:

The required phase margin to give a 20 msec settling time may be estimated by considering the phase response of the classical second order continuous system with open loop transfer function

$$\frac{\omega_o^2}{s.(s + 2.\zeta.\omega_o)}$$

and closed loop transfer function

$$\frac{\omega_o^2}{s^2 + 2.\zeta.\omega_o.s + \omega_o^2}$$

For 1.3 % overshoot, the second order system requires

$$1.3 = 100.e^{-\frac{\zeta.\pi}{\sqrt{1-\zeta^2}}}$$

which requires $\zeta = 0.81$. For 20 msec settling time with ζ=0.81

$$\frac{e^{-\zeta.\omega_o t}}{\sqrt{1-\zeta^2}} < 0.04$$

which requires $\omega_o = 230$ rad/sec. By substituting these values for ζ and ω_o into the classical second order open loop transfer function, with $s = j.\omega$, the frequency at which the forward path gain is unity (0 dB) can be determined. This gives ω = 134 rad/sec (which corresponds to $\upsilon = 0.213$).

Re-substituting in the open loop transfer function to find the corresponding phase leads to

$$\text{angle } G.(j\upsilon) = -90^\circ - 19.8^\circ \approx -110^\circ$$

Thus, using the classical second order system as an approximation, for 20 msec settling time and 1.3% overshoot we require 70 ° phase margin and – 6 dB gain margin at υ =0.213.

A.2.2.3. Applying the design objectives to the Moog frequency reponse.

Close study of the w-plane frequency response of the Moog system,figure A.2.8, shows that at υ= 0.213 the system has approximately 67 ° phase margin and –51 dB gain margin. Thus the phase margin is within 5 ° of that found to be required for an overshoot and settling time that falls within the required design criteria. Therefore a controller consisting of a gain of approximately 175 (44.9 dB), with

no additional compensation should produce a response that is close to that required. This also indicates that it should be possible to achieve an improved response by including additional compensation elements within the controller.

In order that this be tested before implementation of the controller, the Moog system model shown in figure 6.7 was included within the discrete position loop of figure A.2.1 in an ACSL simulation of the step and ramp responses of the system. Figure A.2.9 shows the response of the system to a step input under discrete proportional control with a controller gain of 175, as predicted in the design work above. It can be seen from the plot that the system does not fulfil the position requirements set out for overshoot or settling time. The peak overshoot is approximately 31% and the 4% settling time is approximately 80 msec. The reason for this may be seen in the velocity response that is included in the plot. This shows that the peak velocity gradient (acceleration) is limited, which is due to current saturation within the model. This has a detrimental effect on both the rise time of the system and the braking torque, which causes the overshoot.

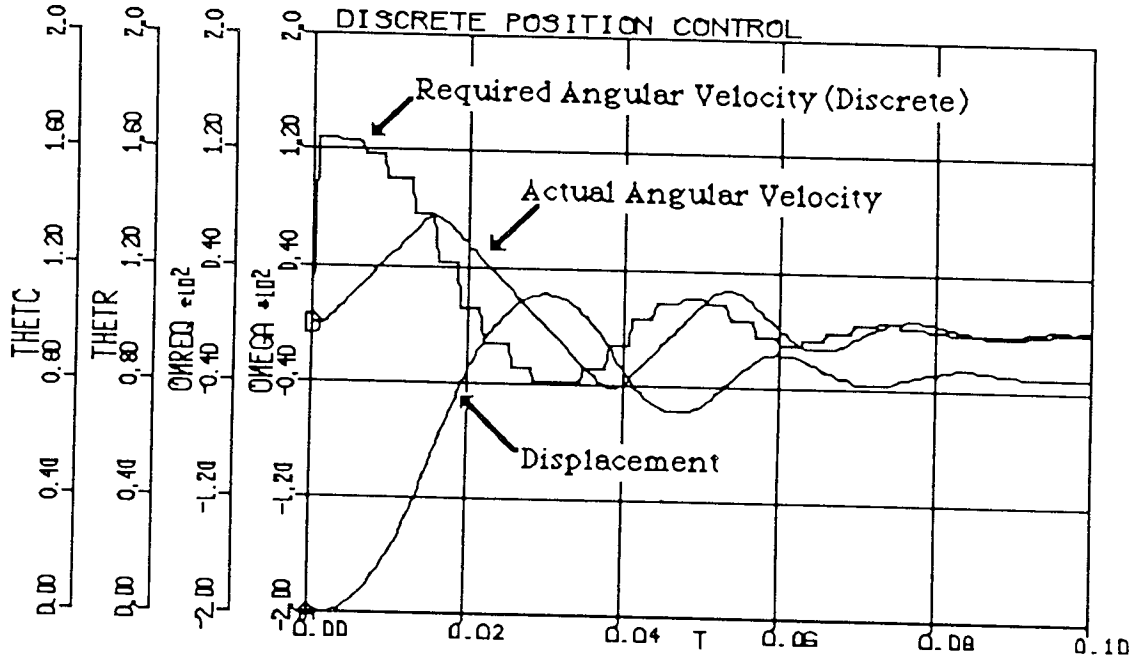


Figure A.2.9. Step response of the simulated position controlled system.

Current saturation clearly limits the performance of the simulated closed loop position controlled system. However, it was not known whether the model of the Moog system which is developed in chapter 6 was accurate, so one of the purposes of implementing the position controller was to perform more system identification tests on the motor. Therefore, since current saturation was seen as a potential problem in the design of control algorithms, it was decided to initially

implement the controller with a proportional control algorithm, then to develop an improved algorithm after further model identification had been performed.

As a final step in the simulation work before implementation of the position controller, the proportionally controlled model was simulated to assess it's ability to drive the third transfer profile, which is the application for which the Moog motor was initially selected. The results of this are plotted in figure A.2.10. From this it is apparent that the model is achieving the working stroke of the slider within the arbor drum in 72 msec, which would correspond to a packaging speed of 480 packets/minute. The model is also succesfully achieving the constant velocity segment of the profile. However, the model displays a significant phase lag between the required position signal and the actual displacement profile. Although this would not prevent the motor's use in a synchronised actuator system, it would limit the peak operating speed of an asynchronously triggered machine, since the motion would not commence until several milliseconds after the commence motion signal had been received. It was intended that further testing of the system should be performed to examine this phenomenon.

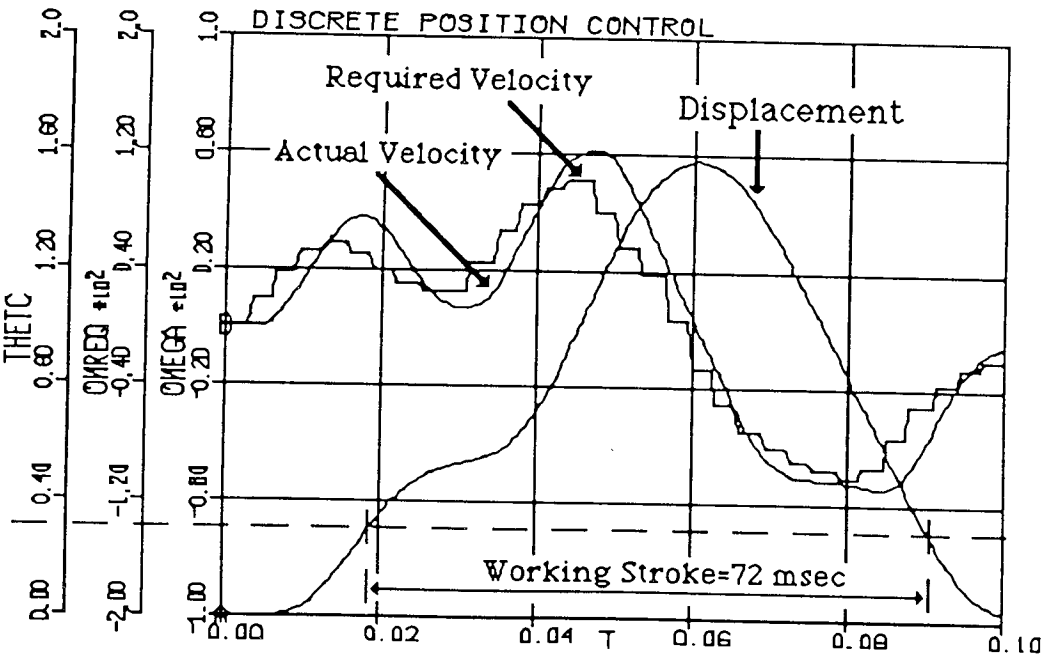


Figure A.2.10. Simulated response of the closed position loop to the third transfer motion profile.

Appendix. 3.

Evaluation of Bru–500 position controller algorithms.

A.3.1. Introduction.

The position controller described in Appendix 2 was developed to suit the Moog brushless d.c. servomotor. After the failures of the Moog motor, which are described in chapter 6, it was decided to purchase a controller for the replacement Electro–Craft motors. For this it was necessary to determine the optimum control algorithms which the controller should have. This appendix describes simulation work which was performed to evaluate position control algorithms for Bru–500 motors

In related S.P.P. research work at Liverpool Polytechnic and Birmingham University, detailed investigations are being performed into the design of axis position controllers for brushless d.c. motors. The object of the work described here was to develop suitable algorithms for use in the laboratory testing and in the simulation of multiple actuator systems. This does not include an in–depth analysis of brushless d.c. motor characteristics and performance, since this was being performed elsewhere.

A.3.1.1. Brushless D.C. Motor: Transient Response Performance.

The actuators are required to be controlled through two types of motion profile:

(i) Incremental motion, e.g. arbor drum

(ii) Complex intermittent profile, e.g. third transfer

For incremental motions, the actuators are required to achieve high speed point-to-point moves, where positional accuracy is required at the final point but not along the trajectory. The complex intermittent profiles are separated temporally into linked segments, with each segment having differing requirements for following error and finishing point accuracy. Both types of motion require very short settling times with small or zero overshoot. The large drive ratios employed for the drives to achieve inertia coupling mean that the profiles that the motors are required to follow have large magnitudes. The combined effect that the stiff response and large magnitude requirements have on the position controller specification may be seen in the brushless d.c. motor transient response performance.

Figure A.3.1 is the block diagram of a brushless d.c. motor with current feedback compensation.

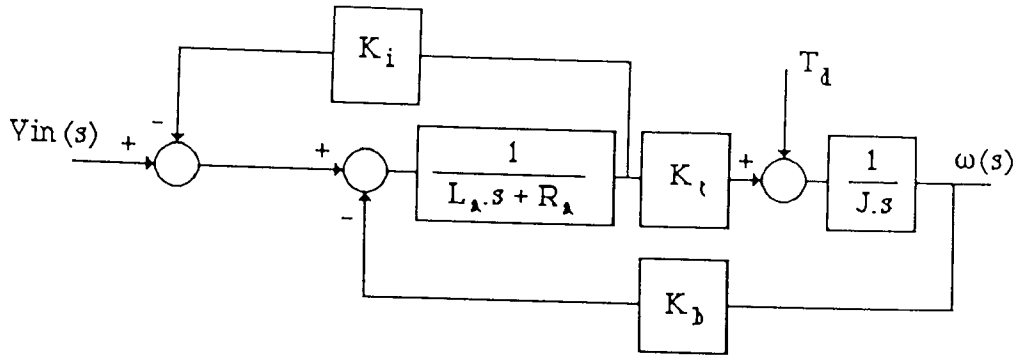


Figure A.3.1. Brushless D.C. Motor With Current Feedback.

The inclusion of the current feedback term, K_i, allows control of the motor's damping ratio, ζ, which for most brushless d.c. motors is small, often 0.2 or less. The transfer function of the motor is,

$$\frac{\omega(s)}{Vin(s)} = \frac{K_t}{J.L_a.s^2 + J.(R_a+K_i).s + K_t.K_b} \qquad \{A.3.1\}$$

This may be expressed as,

$$\frac{\omega(s)}{Vin(s)} = \frac{\frac{1}{K_b}.\omega_n^2}{s^2 + 2.\zeta.\omega_n.s + \omega_n^2} \qquad \{A.3.2\}$$

where

$$\omega_n = \sqrt{\frac{K_t.K_b}{J.L_a}} \qquad \zeta = \frac{R_a + K_i}{2}\sqrt{\frac{J}{L_a.K_t.K_b}} \qquad \{A.3.3\}$$

In order that techniques for controlling the position of a B.L.D.C.M. driven servo be evaluated, it is necessary to understand the performance limitations of the motor itself. The transient response of the motor to a step change, $Vin = R0/s$ is,

$$\omega(t) = \frac{R0}{K_b}.\left\{1 - e^{-\zeta.\omega_n t}.(\sin\omega_d t + \cos\omega_d t)\right\} \qquad \{A.3.4\}$$

with

$$\omega_d = \omega_n\sqrt{1-\zeta^2}$$

From this it can be shown (32) that the percentage overshoot of the motor is

$$P.O. = e^{-\frac{\zeta.\pi}{\sqrt{1-\zeta^2}}} \qquad \{A.3.5\}$$

Thus the current feedback gain, K_i, should be adjusted to set the motor damping ratio that will give the required velocity overshoot. The settling time of the motor may be obtained from

$$\frac{e^{-\zeta.\omega_n.t}}{\sqrt{1-\zeta^2}} \leq \frac{P.S.}{100} \qquad \{A.3.6\}$$

Where P.S. is the required percentage overshoot. Equation A.3.6 shows that the motor should have a large natural frequency, ωn, in order that it be able to achieve a fast settling time. It would also seem from A.3.6 that if the motor has a sufficiently high natural frequency, and hence a large bandwidth, then a settling time can be achieved that is within the specified requirement for the system. However, this does not take account of the non-linear effect of current saturation. The current reponse of the motor during the velocity step may be investigated by rearranging the motor block diagram of figure A.3.1 to highlight the current loop, as shown in figure A.3.2.

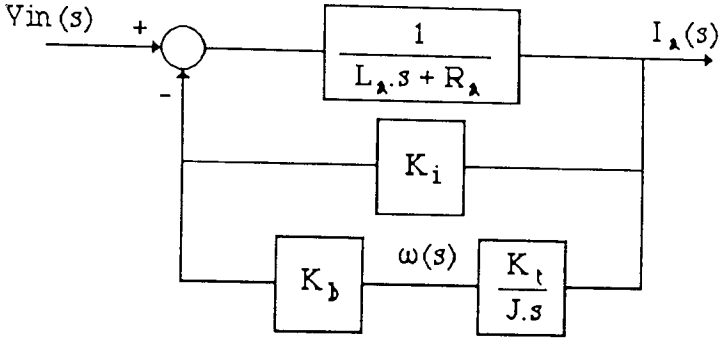


Figure A.3.2. Brushless d.c. Motor Current Loop.

The current transfer function of the motor is,

$$\frac{I_a(s)}{Vin\ (s)} = \frac{J.s}{J.L_a s^2 + J.(R_a+K_i).s + K_t.K_b} \qquad \{A.3.7\}$$

The response of the current to the step velocity input, $R0/_s$, is

$$I_a(t) = \frac{R0}{L_a \omega_n} . \left\{ e^{-\zeta.\omega_n t} . \sin \omega_d t \right\} \qquad \{A.3.8\}$$

Equation A.3.8 shows that the magnitude of the current response is directly proportional to the magnitude of the input and inversely proportional to the damped natural frequency, ω_d, which itself varies with the inverse root of the inertia, J. Thus the large magnitude movements of large inertias that are required in the machine axes require very large currents to be delivered to the motors. When the machine is running at high speed, many of the axes have regions within their motion profiles in which the current demands created by the magnitude and inertia requirements are so large that they can not be met by any of the available motor and drive module configurations. In these cases the system is driven into current saturation. Therefore the control algorithm for high speed machine axes should be able to drive the system in both the linear and saturated performance bands.

A.3.1.2.Non-Linear Position Controller Design Requirements.

1) All controller actions should be software driven. The software must be capable of handling communications that ensure compatability with the hierarchical control configuration. Transducer feedback into the controller is to be position data only. No additional transducer hardware is to be incorporated into the system.

2) The controller algorithm is to be developed to overcome current (acceleration) saturation. The controller must produce a good response to both saturating and non-saturating profile demands without external adjustment.

3) The control algorithms should give optimum response to a ramp position input and to a position incremental input based on a trapezoidal velocity profile. The ramp position response requirements are for minimum maximum position error, minimum settling time and small steady state error. The trapezoidal increment requirements are for minimum incremental position error overshoot and small incremental settling time.

A.3.2.1 Evaluating Position Control Algorithms by Simulation.

The evaluation by simulation of software algorithms for controlling the position of a brushless d.c. motor requires the formation of a system model that includes the linear and non-linear elements of the continuous plant and the discrete sampled data controller. The ACSL simulation language (31) contains operators which enable a discrete position loop model to be closed around a model of a brushless d.c. motor velocity servo to form a system model of the form shown in figure A.3.3.

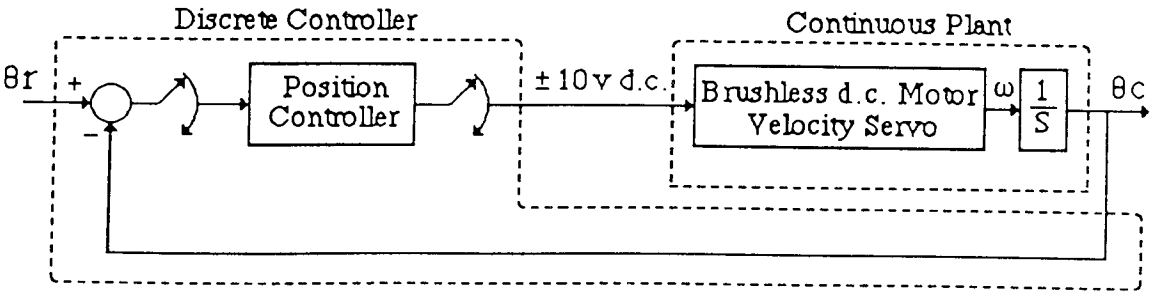


Figure A.3.3. Discrete Position Control of Continuous System.

In the ACSL model definition of this system the DYNAMIC block consists of two parts, the DERIVATIVE block containing the continuous system elements, including the non-linear saturation functions and the DISCRETE block containing the sampled data controller. The structure of the ACSL model is shown in Figure A.3.4.

The execution of the model begins by evaluating all statements within the INITIAL section. This initialises variables and calculates system parameters, such as time constants, which form part of the system model. The DYNAMIC block is then executed. The DERIVATIVE elements are evaluated at every step of the simulation and the DISCRETE elements are executed periodically at times given by the controller sample interval. The values of variables in the DISCRETE block are held constant between samples and data may be transfered between the DERIVATIVE and DISCRETE blocks during the simulation run. Thus the DISCRETE block is able to reproduce the action of a sampled data controller.

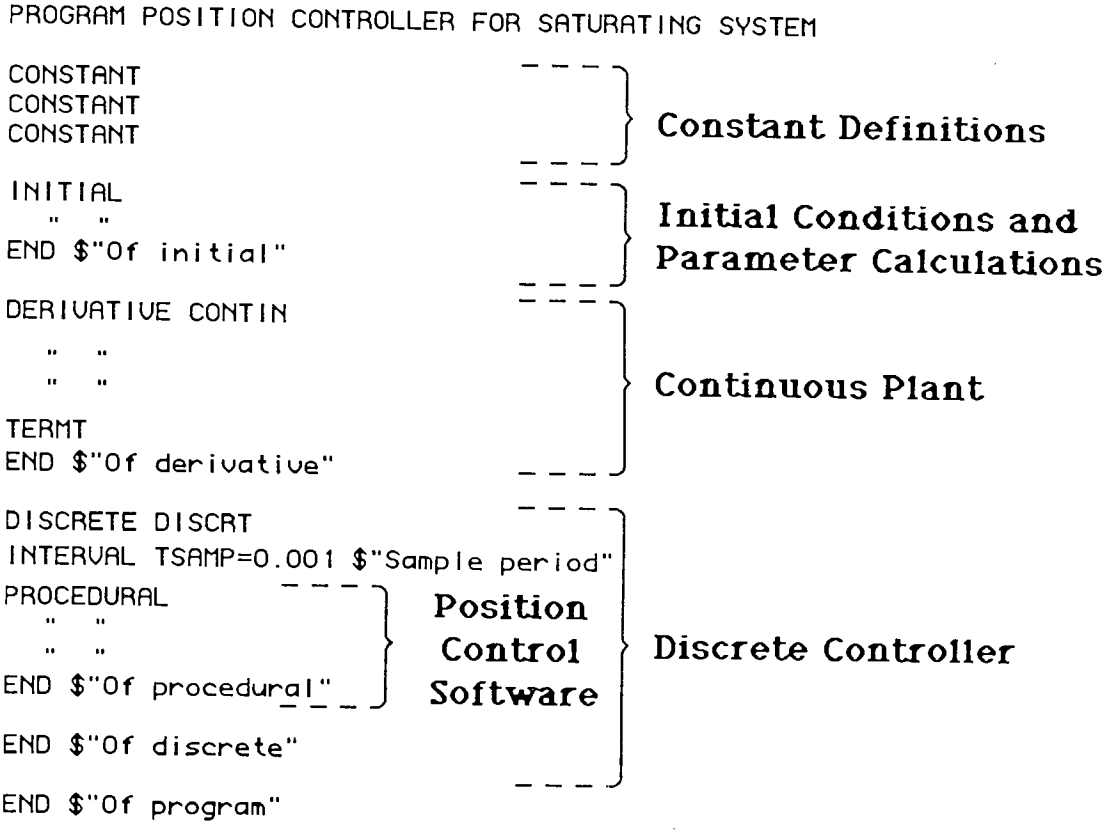


Figure A.3.4. ACSL Controller Evaluation Model Structure.

Code contained within the DISCRETE block is executed sequentially by line from top to bottom. It is written as FORTRAN code and therefore constitutes a good reproduction of control software written in a conventional sequential high level programming language. This makes for ease of writing and understanding the controller software used in the simulation. It also means that the software in the DISCRETE block can be used as a basis for the control software of the real system after the simulation evaluation has been completed.

A.3.2.2. Controller Evaluation Procedure.

The brushless d.c. motor velocity servo used in the controller evaluation is a reduced BRU-500 model which has been shown in laboratory tests to be a good reproduction of the BRU-500 system. The discrete position loop was closed around the BRU-500 model to form the system model in figure A.3.5. This shows that the BRU-500 velocity regulator includes current and velocity feedback loops. In the position control system these act as minor loop compensation filters. Tal (29) shows how this form of compensation may be used to introduce a phase lead at the crossover frequency, thereby enhancing closed loop stability, while not having an appreciable effect on the bandwidth. Therefore this method of compensation does not have the potential dissadvantages of increased high frequency noise or torsional resonance that can result from a wider bandwidth. All brushless d.c. motor velocity systems have some form of feedback compensation and in the independent actuators machinery scheme it is intended to use these systems as supplied. Therefore the position controller evaluation used the BRU-500 velocity regulator model within the minor feedback loop.

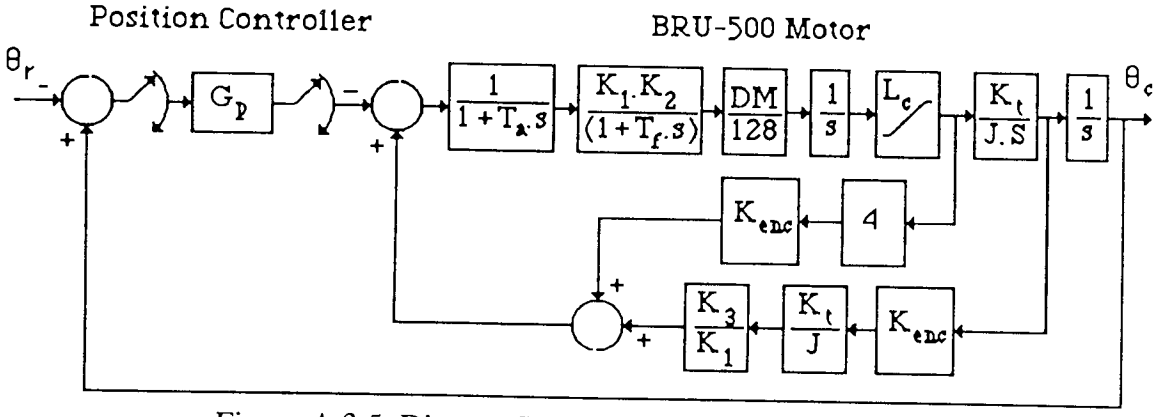


Figure A.3.5. Discrete BRU-500 Position Control.

The parameters of the BRU-500 model were preset to give a velocity overshoot of approximately 4.3 %, which corresponds to a closed loop damping ratio, ζ, of 0.707. This was then held constant for all controller evaluation simulation runs. The discrete position loop was then closed around the BRU-500 model.

The sample rate of the position controller was set at 1 ms. This rate is used in a wide range of commercial controllers and the sample frequency of 1000 Hz is approximately 25 times the bandwidth of the BRU-500 system, so the sampled data phase and gain effects would not affect the loop stability of the system.

Position control algorithms were developed to drive ramp position profiles and incremental position profiles based on velocity trapezoids. The two types of motion

were found to have differing controller requirements, so the evaluation of controllers for the two motion types are described seperately below.

A.3.3.1. Position Ramp Controller Development.

Several of the intermittent motions of the SP1 machine may be most simply described by profiles consisting of several linked ramps. The third transfer slider, which is described in chapter 6, is typical of this type of motion. A position ramp corresponds to a velocity step profile, which was shown above to create a current demand that drives a brushless d.c. motor system into saturation if the step has a large magnitude. This is also the case for the position ramp. When an intermittent motion is required to take place at high frequencies, the segments of a linked ramp profile are of large magnitude and the system can be driven into current saturation during the motion. The purpose of this work is to develop a controller that is capable of driving profiles having both saturating and non-saturating magnitudes.

The controllers were compared by evaluating their abilities to drive the BRU-500 model through required position ramps of magnitudes 1, 200 and 250 rad/sec. The 1 rad/sec ramp evaluated the systems' responses in their linear performance bands, while the 200 rad/sec and 250 rad/sec ramps evaluated the responses to profiles which drove the BRU-500 into current saturation. Thus a comparison was obtained of controller performance in both the saturating and non-saturating performance bands. The performances were evaluated by recording the percentage peak position error and the 10 % and 5 % settling times. The steady state error was recorded as the position error at 40 ms into the motion. This was chosen because it is approximately one half the time of the transfer sliders' motion in the S.P.1 when operating at 450 packet cycles per minute.

The ramp position controller algorithms are explained in detail below, along with listings of the software used for the simulation evaluation. Table A.3.1 lists the performances of each controlled system to the saturating and non-saturating ramp inputs. Reading from left to right, the columns of Table A.3.1 include the controller algorithm; the magnitude of the ramp input, the percentage peak position error; the time at which the peak error occured; the 10% settling time; the 5% settling time; the percentage position error at 40 ms into the transient response; whether current saturation occured during the motion. The error and time values listed in the table were obtained from the ACSL print outs of the transient performances of each system. These are given as nine significant figure values in the ACSL listing, but in Table A.3.1 have been rounded to 4 significant figures.

A.3.3.2 Proportional Control.

In the proportional controller the output of the controller is related to the error signal by the transfer function,

$$\text{Ctrout} = K_p . \text{Error}$$

where K_p is the proportional gain of the controller. The FORTRAN code for implementing this controller in an ACSL DISCRETE block is given in figure A.3.6.

The position control system is type 1, which gives a steady state error to a ramp input. The ACSL simulation run of the system gives a 40 ms error of 11.9% to the non saturating 10 rad/sec ramp requirement, with the proportional gain K_p set at 4. The peak error can be seen in Table A.3.1 to be 43%, which occurs at 10.7 ms into the response.

```
DISCRETE DISCRT
INTERVAL TSAMP=0.001 $"Sample period"
PROCEDURAL
ERRN = ERROR
CTROUT = KP * ERRN
END $"Of procedural"
END $"Of discrete"
```

Figure A.3.6.Proportional Control Software.

When the system is required to follow the 200 rad/sec and 250 rad/sec ramps current saturation occurs. This has only a small effect on the 40 ms error, since the system is by then operating in the linear band, but the peak error is increased and occurs later in the transient. In addition, although 10% and 5% settling does not occur for this system, the settling time does increase as the amount of saturation increases. The effect of inducing saturation is therefore analogous to the effect of decreasing the system's natural frequency. This analogy follows into the frequency response, since introducing saturation into the linear band decreases the gain and increases the phase shift, thereby reducing the bandwidth, which is similar to the effect of decreasing the systems natural frequency.

A.3.3.3.Proportional plus integral control.

In the proportional plus integral (p-i) controller, the ouput of the controller is related to the error by the transfer function

$$\text{Ctrout} = K_p . \left[\text{Error} + K_i . \int_0^t \text{Error}.dt \right] \qquad \{A.3.9\}$$

Where K_i is the integrator gain. The FORTRAN code for implementing this controller is given in figure A.3.7.

```
DISCRETE DISCRT
INTERVAL TSAMP=0.001 $"Sample period"
PROCEDURAL
ERRN = ERROR
CV1 = KP * ERRN  $" Proportional"
CV2 = (ERRN * TSAMP) + CV2  $" Integral of Error dt."
CV3 = KP * KI * CV2 $"Integral"
CTROUT = CV1 + CV3   $" Proportional + Integral"
END $"Of procedural"
END $"Of discrete"
```

Figure A.3.7. Proportional plus Integral Control Software.

The p-i controller is generally implemented in a system to increase the type number and thus reduce the steady state error to zero. The p-i position controlled brushless d.c. motor is type 2, which gives zero steady state error to a ramp input. Table A.3.1 shows that with $K_p = 4$ and $K_i = 47.5$, the integrator reduces the ramp response 40ms error to 1.63% for a non saturating input, while increasing the peak position error to 62% at 9.1ms into the motion. A linked ramp motion profile to be driven by this system would require a minimum segment time length that is longer than the transient settling time, which in this case is 22.4ms for 5% settling

When the p-i controlled system is required to respond to ramps of saturation inducing magnitude, the response error that this produces is increased by the integral action to give a non-linear oscilliatory response with large overshoots and no 40 ms settling. This oscillation inducing effect prohibits the p-i controller's use for the independent drives of high speed machinery.

A.3.3.4. Proportional plus switched integral.

It was found in A.3.3.2. that the proportionally controlled system has a large steady state error to both saturating and non saturating requirement profiles. In A.3.3.3. it was found that the addition of an integrator reduces the steady state error when used to control a non saturating system, but that the integrator induces a non-linear oscillation when controlling a saturating system. A controller was therefore developed that would combine the beneficial saturation and non saturation effects of the proportional and p-i controllers by switching off the integrator when the system entered saturation.

The distributed system fault tolerance requirement of minimum hardware meant that the controller should detect saturation through interpretation of the position transducer input. Thus it was necessary to define the conditions in the position response that indicated that current saturation was occuring.

It was stated in A.3.3.3 that the output of the p-i controller is given by

$$\text{Ctrout} = K_p \cdot \left[\text{Error} + K_i \cdot \int_0^t \text{Error}.dt \right]$$

The output of the controller, Ctrout, is the input to the velocity loop

$$\text{Ctrout} \equiv \omega_r \qquad \{A.3.10\}$$

where ω_r is the required velocity. The brushless d.c. motor system will be expected to enter current saturation if it is required to accelerate at a rate larger than that given by the saturation current. i.e. saturation occurs if

$$\alpha_r \geq \left\{ \frac{k_v . K_t}{J} \right\} . I_a^{sat} \qquad \{A.3.11\}$$

where α_r is the required acceleration, K_v is the velocity loop gain, K_t is the torque constant of the motor, J is the inertia of the system and I_a^{sat} is the saturation current of the driver. Since α_r is the derivative of ω_r, then equations A.3.10 and A.3.11 can be combined

$$\frac{d.\text{Ctrout}}{dt} \geq \left\{ \frac{k_v . K_t}{J} \right\} . I_a^{sat} \qquad \{A.3.12\}$$

This expression gives the conditions under which the output of the position controller may be expected to induce current saturation in the brushless d.c. motor system. This may be related to the error by differentiating A.3.10 and combining with A.3.12.

$$k_p \left[\frac{d.\text{Error}}{dt} + k_i . \text{Error} \right] \geq \left\{ \frac{k_v . K_t}{J} \right\} . I_a^{sat} \qquad \{A.3.13\}$$

The error signal is obtained by subtracting the position transducer output from the required position, which is contained within the position controller. Therefore equation A.3.13 may be used by the position controller to interpret the occurence of current saturation from the position transducer input.

The FORTRAN code for implementing the switched p-i controller as an ACSL DISCRETE block is given in figure A.3.8. At the beginning of each control sample the error signal is tested to see whether the condition of equation A.3.13 is true. If true, then the interpretation is that saturation is occuring, so the controller resets the integrator to zero and acts with a proportional action using the software from section A.3.3.2 If the condition is false then it is interpreted that saturation is not occuring, so the controller acts with a p-i action, using the software developed in section A.3.3.3.

The system was simulated with the controller settings $K_p = 4$ and $K_i = 47.5$. The results in Table A.3.1 show that the switched p-i controlled system does not suffer from the non-linear oscillation during current saturation that occured with the p-i controller of section A.3.3.1.2. The switched p-i response has a 40 ms settling of

approximately 2 % for both the 200 rad/sec and 250 rad/sec saturating ramp profiles, while in both cases the peak position errors are within 1 % of those of the proportionally controlled system. Figure A.3.9 shows the plots of the position, velocity and current responses to the 250 rad/sec saturating ramp input of the proportional, p-i and switched p-i controlled systems.The proportionally controlled system shows the steady state position error that the system settles to within the 40 ms time band, while the p-i controlled system shows how the integral error signal builds up during saturation to give a large position overshoot at 40 ms. The response of the switched p-i controlled system can be seen to be similar to that of the proportional system during current saturation, but after the system exits the saturation band the position error of the switched p-i system settles towards zero.

```
DISCRETE DISCRT
INTERVAL TSAMP=0.001 $"Sample period"
PROCEDURAL
ERRN = ERROR

TV1 = KP * (ERRN-ERRNM1) / TSAMP                    ---
TV2 = KP * KI * ERRN                                  |
TV3 = TV1 + TV2                                       |  Test Error Signal for
IF (TV3.GE.KV*KT*IASAT/J) GO TO SATN                  |  Saturation Condition
IF (TV3.LE.-KV*KT*IASAT/J) GO TO SATN                 |
                                                    ---

CV1 = KP * ERRN  $" Proportional"                   ---
CV2 = (ERRN * TSAMP) + CV2  $"Integral of Error"      |  P-I Control when
CV3 = KP * KI * CV2 $" Integral                       |  System not Saturatin
CTROUT = CV1 + CV3   $" Proportional + Integral"      |
GO TO SETNEXT                                       ---

SATN..CONTINUE $"Output during saturation"          ---
CV2 = 0.0 $" Reset integrator to zero"                |  Proportional Contro
CTROUT = KP * ERRN $"Proportional only"               |  when System Saturati
                                                    ---
SETNEXT..CONTINUE $"Set values for next sample"
ERRNM1 = ERRN $"Last error for saturation test"

END $"Of procedural"
END $"Of discrete"
```

Figure A.3.8. Proportional plus Switched Integral Control Software.

The three system responses in figure A.3.9 each have a similar initial response, during the time up to about 20 ms, since each enters saturation at approximately the same time. Each has an error peak of about 57% at 12.5 ms into the response, while the switched p-i system takes nearly 25 ms for the error to settle to within 10%. Further controller design was therefore aimed at reducing the peak error and the settling time of the switched p-i system, for both saturating and non saturating requirement profiles. Achieving this would require increasing the area under the initial current response curve, which in turn would increase the area under the initial position response curve and thereby reduce the error. For saturating responses, reducing the peak error could only be achieved by causing the system to saturate

earlier in the initial transient, since the saturation band is the upper limit of the current response curve, while increasing the rate of settling would require increasing the area under the later part of the current reponse curve, either by reducing the rate of decrease of current after leaving the saturation band or by holding the current in saturation for a longer period.

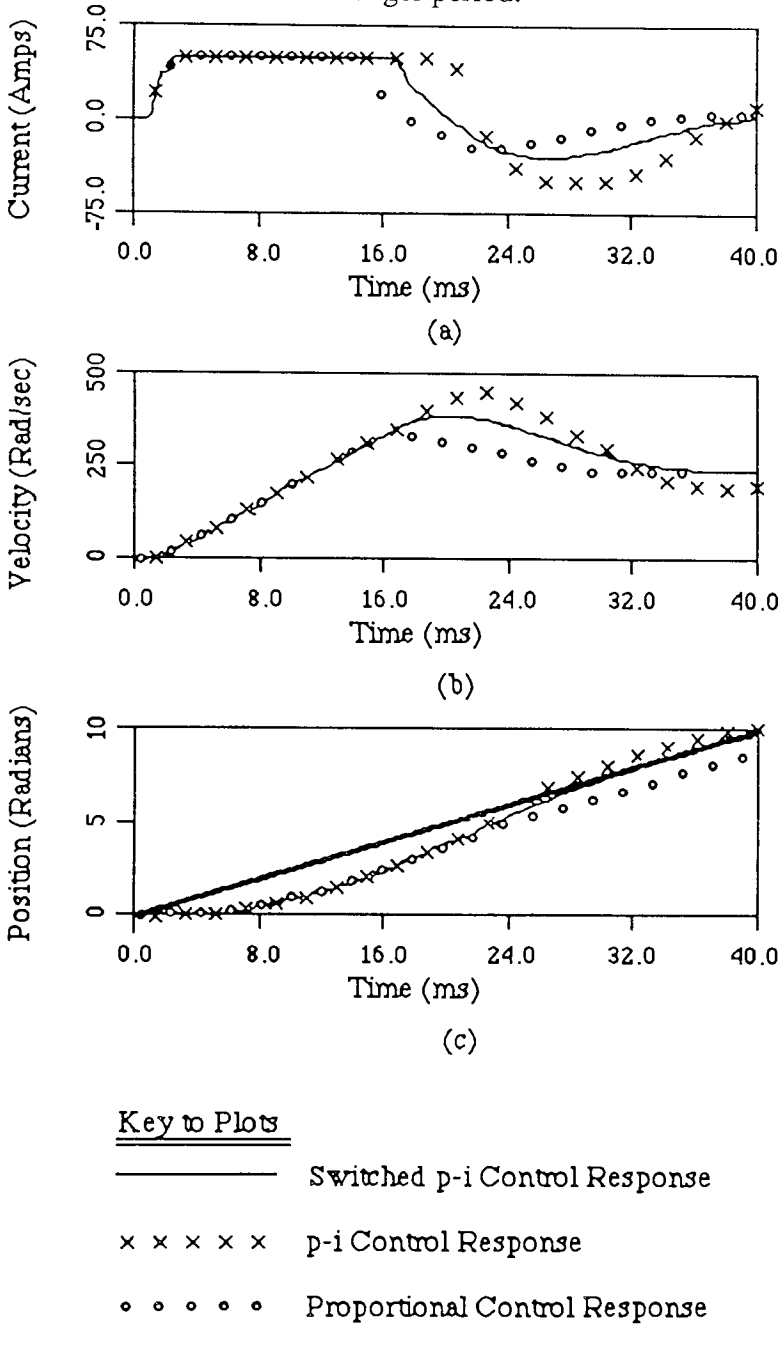


Figure A.3.9 Saturating transient reponse of proportional, proportional plus integral and proportional plus switched integral controlled systems.

(a) Current response

(b) Velocity response

(c) Position response.

A.3.3.5 Proportional plus switched integral plus derivative.

In a proportional plus integral plus derivative (p-i-d) controller the output from the controller is related to the error signal by the transfer function,

$$Ctrout = K_p.\left[Error + K_i.\int_0^t Error.dt + K_d.\frac{d.Error}{dt} \right] \quad \{A.3.14\}$$

where K_d is the derivative action gain. The integral action of the controller is used to reduce the steady state error to zero,whereas the derivative action generally enables higher speeds of response than would be possible with p-i control while maintaining the same relative stability.

The derivative action was added to the switched p-i controller of section A.3.3.4. to increase the initial rate of change of current and thereby reduce the initial error peak. In this case, the integral and derivative actions were switched off when the saturation condition of equation A.3.13 was true, to leave proportional control during saturation. The results in Table A.3.1 were obtained with controller settings $K_p = 4$, $K_i = 47.5$ and $K_d = 2.5 \times 10^{-3}$. Comparison of the results for the switched p-i and switched p-i plus derivative controllers shows that for non saturating responses the derivative action reduces the peak position error by 7%, with the peak error time reducing by $2^1/_4$%. However, the derivative increases the 10% and 5% settling times by 7.6% and 15.6% respectively. For saturating responses the derivative action has almost negligible effect upon the peak error, but the increased rate of drop off of current after saturation that the derivative causes increases the settling times in a similar manner to the non saturating response.Thus the derivative action has only a small effect on the peak response of non saturating systems, while creating an undesirable increase in the settling time of both saturating and non saturating responses.

A.3.3.6. Proportional plus switched integral plus deliberately induced saturation.

It was stated in section A.3.3.4. that to decrease the peak position error it is necessary to increase the area under the initial current response curve. To facilitate this, the deliberately induced saturation (d.i.s.) algorithm was formulated. This is a non linear algorithm which has the following action during the initial rise time

TIME $0 \leq T \leq T_{RISE}$

IF ERROR IS POSITIVE AND INCREASING INDUCE POSITIVE SATURATION

IF ERROR IS NEGATIVE AND DECREASING INDUCE NEGATIVE SATURATION

The aim of the algorithm was to induce the system into saturation at the beginning of the transient response, which increased the area under the initial part of the

current response curve and thereby reduced the peak error.The system was simulated with controller settings $K_p = 4$ and $K_i = 47.5$.

Table A.3.1 shows that for saturating responses, the addition of d.i.s. to the switched p-i controller has a similar effect to the addition of derivative action, in that it achieves small decreases in the peak error and the peak error time, but does so at the expense of an increased settling time and an increased 40 ms error. However, if the magnitude of the required ramp is small, the d.i.s. causes saturation to occur when it would not normally be expected and a non linear limit cycle oscillation results. This effect and the increased settling time for saturating responses mean that the switched p-i plus d.i.s. controller is not suitable for the independent machine drives.

A.3.3.7. Proportional plus switched integral plus deliberately maintained saturation.

Deliberately maintained saturation (d.m.s.) was added to the switched p-i controller as an attempt to decrease the settling time of saturating system responses. The aim was to increase the area under the part of the current response curve that occured after the first peak by maintaining current saturation, when it occured, for a number of controller samples after the system would ordinarily have ceased saturating. This was facilitated in the switched p-i controller by the condition test that is carried out to check whether saturation is occuring, as described in section A.3.3.4. A flag was added to the condition test that was set during each sample according to whether saturation was TRUE or FALSE. The d.m.s. algorithm then acted according to the state of the flags of the required number of previous samples to hold the controller output high and maintain saturation.

Table A.3.1 includes the results for the switched p-i controller with 1, 2 and 3 samples d.m.s. The proportional and integral gains were set at $K_p = 4$ and $K_i = 47.5$. For the saturating responses, the d.m.s. does not achieve the desired reduction in settling times, which in fact increase. This occurs because the current response drops very quickly and undershoots after the d.m.s. samples. The negative undershoot current tends to slow the position rise, which increases the settling time. The d.m.s. algorithm is therefore detrimental to the control of independent machine drives.

A.3.3.8 Proportional plus switched integral plus velocity feedforward.

Feedforward compensation adds a direct branch from the input signal to the drive amplifier. This feedforward branch can then be used to give the amplifier an additional bias to supplement that of the loop controller. Since the feedforward elements do not form part of the closed loop, they do not affect the loop stability. The block diagram of the proportional plus switched integral with velocity feedforward controller is shown in figure A.3.10.

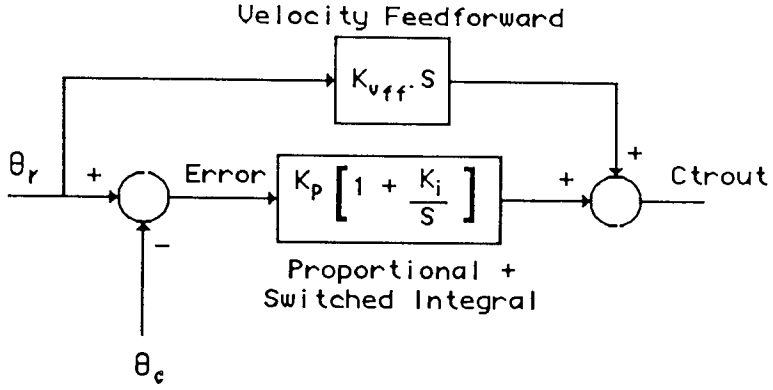


Figure A.3.10. Introducing velocity feedforward into a proportional plus switched integral controller.

Feedforward compensation is open loop in nature, it is not capable of responding to changes in system characteristics, such as an increase in load friction. However, since the object of this exercise was to develop a controller for use in the laboratory, the need to recalibrate for changes in system characteristics was not a problem. When velocity feedforward is used with the switched p-i controller, the feedforward is switched out with the integrator, since during saturation this compensation has no effect. This means that velocity feedforward is straight forward to include in the existing switched p-i control software of figure A.3.8, with the only change being that the line

```
CTROUT = CV1 + CV3 $"Proportional + integral"
```

should be changed to

```
CTROUT = CV1 + CV3 + (KVFF * VELN) $"P-i + velocity feedforward"
```

where VELN is the required velocity at sample n, the details of which are contained within the profile data. The results in Table A.3.1 were obtained with $K_p = 4$, $K_i = 47.5$ and $K_{vff} = 9 \times 10^{-3}$. These show that velocity feedforward reduces the peak error and settling times of both saturating and non saturating responses. For non saturating responses the peak error is reduced by 9% and the 5% settling time is

reduced by 19.3%. In the case of the 250 rad/sec saturating profile, the velocity feedforward reduces the peak error by 2% and the 5% settling time by 8.3%.

A.3.3.9 Position ramp controller conclusions.

The optimum algorithm for controlling the brushless d.c. motor system through saturating and non saturating ramp position profiles is the proportional plus switched integral with velocity feedforward controller. This produces good peak error and settling performance when following large and small magnitude ramps, with no controller parameter readjustment, as is required by the control scheme. The non linear switching algorithm is straight forward to implement and serves to succesfully introduce the integrator to reduce the steady state error when the system leaves the current saturation band.

Controller Action	Inmag (rad/sec)	Peak Error		Settling Time (ms)		S.S.Error (% at 40ms)	Sat'n ?
		%	(ms)	10 %	5 %		
Proportional	1	43	10.7	----	----	11.9	No
	200	58	10.8	----	----	12.05	Yes
	250	57	12.5	----	----	12.05	Yes
P-I	1	62	9.1	19.7	22.4	1.63	No
	250	Large Overshoot. No 40ms Settling					Yes
Switched P-I	1	62	9.1	19.7	22.4	1.63	No
	200	59	10.6	22.6	26.1	2.06	Yes
	250	56	12.5	24.4	27.4	1.98	Yes
Switched P-I-D	1	55	8.9	21.2	25.9	1.16	No
	200	59	10.6	23.6	28.2	1.49	Yes
	250	57	12.5	25.3	29.4	1.49	Yes
Switched P-I Plus D.I.S.	1	Non-Linear Diverging Oscillation					Yes
	200	56	9.8	25.5	32.4	2.51	Yes
	250	55	11.9	25.3	30.4	2.50	Yes
Switched P-I + 1 Sample D.M.S.	1	62	9.1	19.7	22.4	1.63	No
	200	59	10.6	23.3	27.5	2.21	Yes
	250	56	12.5	25.0	28.7	2.29	Yes
Switched P-I + 2 Samples D.M.S	200	59	10.6	24.1	29.2	2.39	Yes
	250	56	12.5	25.7	30.2	2.58	Yes
Switched P-I + 3 Samples D.M.S.	200	59	10.6	25.9	33.7	2.98	Yes
	250	56	12.5	27.2	34.4	3.37	Yes
Switched P-I + Velocity Feedforward.	1	53	7.5	16.1	18.3	0.97	No
	200	56	10.6	19.7	21.4	0.96	Yes
	250	54	12.0	22.4	25.3	2.05	Yes

Table A.3.1. Comparison of Non-Linear Controller Performances.

A.3.4.1 Trapezoidal increment controller development.

Many of the motion requirements for high speed machine actuators are for point-to-point increments, where positional accuracy is required at the final point, but not along the trajectory. The arbor drum is typical of this sort of actuator. In chapter 5 a procedure was developed for designing the optimum incremental motion profile for minimum energy consumption by a motor moving through a trapezoidal velocity profile. To supplement this it was necessary to develop an algorithm for controlling the motor through trapezoidal increments. Figure A.3.11 shows the position, velocity and acceleration profiles of a trapezoidal incremental profile.

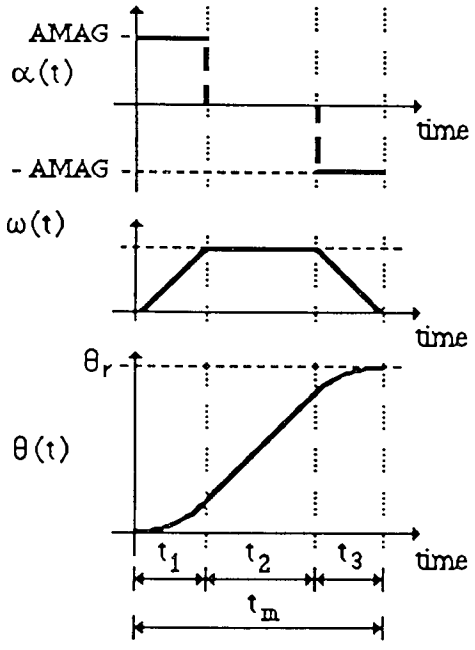


Figure A.3.11. Trapezoidal increment requirement profile

The trapezoidal profile can be defined in an ACSL model file by combining STEP functions to give the acceleration, which is then integrated to give the velocity and the position.

ACCEL = AMAG * (STEP (0.0) - STEP (T1) - STEP (T2) + STEP (TM))

VLCTY = INTEG (ACCEL, 0.0)

POSTN = INTEG (VLCTY, 0.0)

The magnitude of the acceleration, AMAG, may be obtained from the expression

$$AMAG = \frac{\theta_r}{t_1.(t_m - t_1)} \qquad \{A.3.15\}$$

where θ_r is the required angular increment, t_1 is the acceleration time and t_m is the period of the motion profile, as shown in figure A.3.11. Equation A.3.15 is only valid for symmetrical profiles, where the acceleration and braking times are equal.

During the acceleration and deceleration segments the position control system would need to be type 3 to achieve zero position error. However, for time $\geq t_m$ the position profile is equivalent to a step input, so the system need only be type 1 to achieve zero steady state error. Thus the controller requirements are different from those of the ramp position controller, where a type 2 system was required to achieve zero following error.

The development of controllers for trapezoidal increments was carried out by evaluating each algorithm's ability to control the simulated system of figure A.3.5 through three trapezoidal motions.

{i} Increment through π radians in 50 ms, with AMAG = 5655 rad/sec^2

{i} Increment through π radians in 30 ms, with AMAG = 15708 rad/sec^2

{i} Increment through π radians in 22.5 ms, with AMAG = 27925 rad/sec^2

For each case the acceleration, slew and braking times were set equal at $t_m/3$. The evaluation therefore represented a machine actuator drive in which the required incremental distance was constant, but the incremental period varied with machine speed. The brushless d.c. motor model used in the evaluation had a saturation current which corresponded with an acceleration of 23,500 rad/sec^2. Thus the three profiles had acceleration requirements which were approximately 25%, 66% and 119% of the motor's saturation acceleration, which gave an indication of the controllers' abilities to drive non saturating, marginally saturating and saturating motions. The sample period of the position controllers were set at 1 ms, as in section A.3.3 for the ramp position controller evaluation.

The incremental position control algorithms are explained in detail below. Their performances were evaluated by comparing their abilities to achieve fast increments, with minimum overshoot at the finishing point and short settling times. Table A.3.2 lists the results for each system's responses to the three incremental motion. Reading from left to right, the columns of Table A.3.2 include the controller algorithm; the period of the incremental requirement profile; the magnitude in radians of the peak position error during the motion; the time at which the peak error occured; the percentage peak overshoot on completion of the motion; the time at which the peak overshoot occured; the 5% settling time to completion of the increment, measured from the start of the profile; the 2% settling time to completion of the increment, measured from the start of the profile; whether saturation occured during the motion. The values listed were obtained from the ACSL print outs of the systems' responses. These have been rounded to 4 significant figures.

A.3.4.2. Proportional control.

The proportional controller described in section A.3.3.2, with the software listed in figure A.3.6, was used as a basis for the development of the incremental motion controller. The proportional position control system is type 1, which should give a diverging error to the constant acceleration segments, a constant error to the constant velocity segment and zero error to the steady state constant position segment of the profile. Thus the position response should be expected to lag the requirement profile, with a peak error shortly after the end of the first acceleration segment, but the motion should achieve the increment with zero steady state error.

The simulated response of the proportionally controlled system is summarised in Table A.3.2 for the three trapezoidal profiles with $K_p = 8$. When driven through the 50ms profile, the system had a peak position error 1.7 ms after the first acceleration segment was completed, with zero steady state error at the end of the position increment and a peak overshoot of 1.33%. The 30ms profile caused a larger peak error, which occured 2.8 ms after the end of the first acceleration segment, with a larger peak overshoot of 4.4% and a 2% settling time of 36.8ms, which was 6.8ms after the incremental requirement profile was completed. When the system was simulated driving the 22.5ms increment, current saturation occured in the driver. This created a large peak error and a large overshoot, with 2% and 5% settling times which were in excess of those of the 30ms and 50ms profiles. Thus the proportionally controlled system is not suitable for driving trapezoidal incremental profiles which induce current saturation in the driver.

A.3.4.3 Proportional plus switched integral control.

The switched p-i controller was introduced to the incremental motion system to assess whether the integrator would reduce the long settling time to the saturating profile which occured with proportional control. The controller algorithm was developed in section A.3.4, using the integral switch condition given in equation A.3.13. The FORTRAN code of the controller software is given in figure A.3.8.

The p-i position control system is type 2, which should give a constant error to the constant acceleration segments of the profile and zero error to the constant velocity and constant position segments. Table A.3.2 includes the simulation results for the p-i controlled system with $K_p = 8$ and $K_i = 47.5$, both with and without the saturation switch for the integrator. For the 50ms profile saturation did not occur and the p-i action produced a smaller peak error than that of the proportionally controlled system. However, the p-i controlled system had a larger position

overshoot at completion of the increment and a longer settling time, which is more important in point-to-point systems than the peak error. When the system was simulated driving the 30ms profile, the integral action served to drive the system into current saturation, which did not occur in the proportionally controlled system. Without the integral switch this produced a large overshoot and long settling time. However, introducing the integral switch reduced the settling time to less than that of the proportionally controlled system, with the peak errors of the proportional and p-i controlled systems being almost equal. For the 22.5ms saturating profile, the p-i controller without the saturation switch induces a 400% position overshoot in the system, with no settling within the 100ms of the simulation run. Introducing the saturation switch on the integrator reduces the overshoot to 41.1%, but this is larger than that of the proportionally controlled system and the settling times are longer.

A.3.4.4 Proportional plus derivative control.

In the proportional plus derivative (p-d) controller, the output of the controller, Ctrout, is related to the error by the transfer function

$$\text{Ctrout} = K_p\left[\text{Error} + K_d\frac{d.\text{Error}}{dt}\right] \qquad \{A.3.16\}$$

The FORTRAN code for implementing the p-d controller in an ACSL DISCRETE block is given in figure A.3.12. The addition of derivative action generally enables a faster system response, but does not alter the type number of the system. The p-d controller is type 1, which should give the required zero steady state error to the incremental motion whilst not increasing the overshoot or settling time, which the integral action in section A.3.4.3 was seen to do.

```
DISCRETE DISCRT
INTERVAL TSAMP=0.001 $"Sample period"
PROCEDURAL
ERRN = ERROR
CVAR1 = KP * ERRN $"Proportional"
CVAR2 = KP * KD * (ERRN-ERRNM1)/TSAMP $"Derivative"
CTROUT = CVAR1 + CVAR2 $"p-d"
ERRNM1 = ERRN $"This error = next last error"
END $"Of procedural"
END $"Of discrete"
```

Figure A.3.12. Proportional plus derivative control software.

The simulated responses of the p-d controlled system are summarised in Table A.3.2 for controller settings $K_p = 8$ and $K_d = 0.03$. For all three profiles the p-d controller gives a better response than either the proportional or switched p-i controllers, with the p-d controller giving smaller peak errors, peak overshoots and settling times. In addition, the p-d controller did not induce current saturation in the driver when controlling the 30ms increment, which occured with the switched p-i

controller. However, although the p-d performance was superior, it had a similar limitation with controlling the 22.5ms saturating profile as did the other two controllers, in that the settling time for the 22.5ms profile was longer than that for the 50ms profile. Thus the p-d controller needs further compensation to be used for controlling profiles which are capable of driving the system into current saturation.

A.3.4.5 Proportional plus derivative plus profile superposition.

The p-d controller gave a good response when used to control non saturating incremental motions, but a poor response for profiles which induced saturation. For both types of profile, the motion included a large transient response error, which peaked at 16% for the 30ms profile and 46% for the 22.5ms profile. The following error during the motion is not critical in point to point systems, but reducing its value reduces the incremental overshoot and the settling time. Profile superposition compensates for the following error by superimposing a signal on to the requirement profile. The system then responds to a profile which is made up of the original requirement profile plus the superimposed signal. If the superimposed signal has a profile which is similar to the original error profile, then the system's new performance profile will be similar to the original requirement signal.

Implementing profile superposition requires an understanding of the error function of the closed loop system. Figure A.3.13 is the block diagram of a p-d position controlled brushless d.c. motor system.

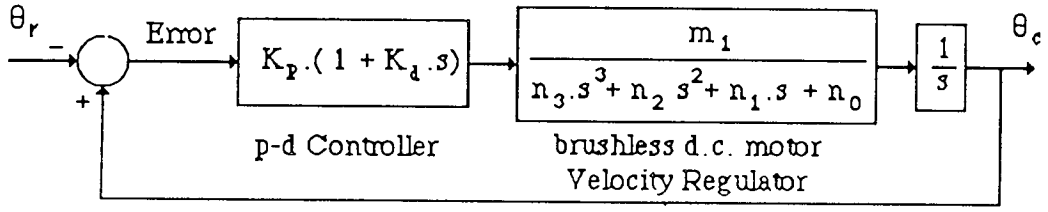


Figure A.3.13. Proportional plus derivative control of a brushless d.c. motor.

The closed loop transfer function of the system is

$$\frac{\theta_c}{\theta_r} = \frac{m_1.K_p K_d s + m_1.K_p}{n_3.s^4 + n_2.s^3 + n_1.s^2 + \left[n_0 + m_1.K_p K_d\right].s + m_1.K_p} \qquad \{A.3.17\}$$

The error function is found by subtracting $\theta_r - \theta_c$

$$\text{Error} = \frac{\left[n_3.s^4 + n_2.s^3 + n_1.s^2 + n_0.s\right].\theta_r}{n_3.s^4 + n_2.s^3 + n_1.s^2 + \left[n_0 + m_1.K_p K_d\right].s + m_1.K_p} \qquad \{A.3.18\}$$

Thus the error signal is made up of components of the first, second, third and fourth derivatives of the requirement signal, θ_r. The ratio of these components of

the error are given by the values n_0 ,n_1, n_2 and n_3, which for the BRU-500 system in the evaluation simulations have the values,

$n_0 = 11240$

$n_1 = 172.7$

$n_2 = 1$

$n_3 = 0.00053$

From these values it is apparent that the dominant component of the error signal is the first derivative of θ_r, with the second derivative component being approximately 65 times smaller, the third derivative component 11240 times smaller and the fourth derivative component approximately 21.2 million times smaller. Based on these ratios, the superposition function to be added to the trapezoidal increment requirement profile to compensate for the error signal is made up of the first and second derivatives of θ_r, which are the required velocity and acceleration respectively. The ratio of the components of the velocity and acceleration to be added should be similar to the ratio of n_0 to n_1, with adjustment being carried out on line to compensate for the third and fourth derivative components of the error signal and for discrete sampling errors not included in the above analysis. Figure A.3.14 shows how the profile superposition was implemented.

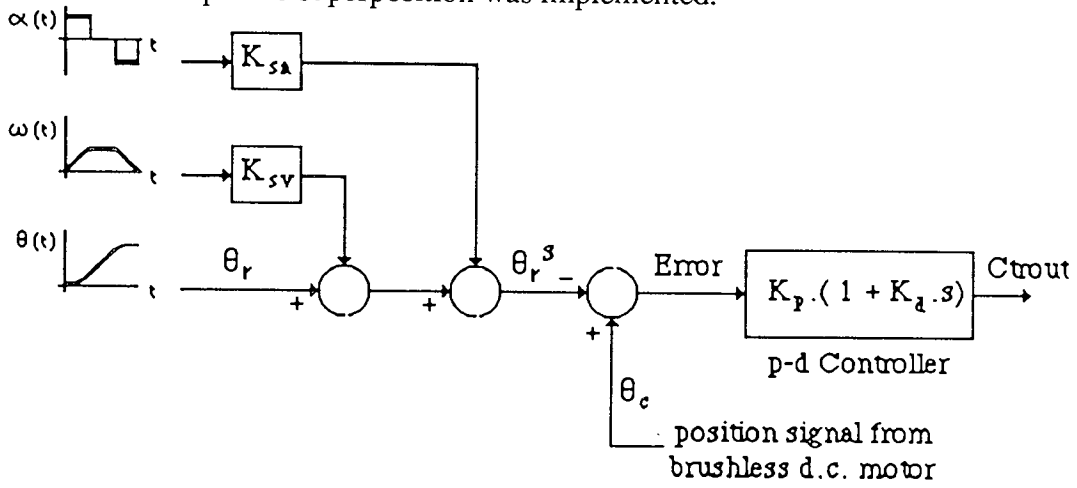


Figure A.3.14. Profile superposition implementation schematic.

The required position incremental profile was defined along with the corresponding velocity and acceleration profiles. The velocity and acceleration information was added to the required position signal after being multiplied by the respective gains K_{sv} and K_{sa}, to give the superposed position signal, θ_r^s. The additions were made outside the position loop. The superposed position signal was then used as the required position for the closed loop control system.

The p-d controlled brushless d.c. motor system was simulated with profile superposition. The results listed in Table A.3.2 were obtained with K_{sv} set at 2.25

x 10^{-3} and K_{sa} set at 3.0 x 10^{-5}, giving a ratio between the velocity and acceleration components of 75:1. This compares with the ratio between n_0 and n_1 of 65:1. The forward path controller gains were maintained at $K_p = 8$ and $K_d = 0.03$. For the 50ms and 30ms requirement profiles the following errors were smaller than those obtained with the p-d controller of section A.3.4.4, the peak following error to the 30ms profile being 16 times smaller than that obtained under p-d control. The incremental position overshoots for both profiles were also smaller with the p-d plus profile superposition algorithm. The improved response to the 30ms profile included the system being driven into current saturation, whereas in the p-d controlled system saturation did not occur. When the 22.5ms profile was driven by the system, the resulting current saturation did not have the same degree of effect as was present with the p-d controller. The peak overshoot was 23% less than that obtained by the p-d controlled system, although the system performance was inferior to that obtained when driving the 30ms profile.

The addition of profile superposition is able to reduce the error to non saturating profiles to very small values. When the period of the profile is reduced to small values, the resulting current saturation in the driver does not serve to produce the same degree of position overshoot that occured with the previous controllers. Further compensation for current saturation by profile superposition is limited because during saturation the error function changes.

$$\text{Error}^{sat} = \theta_r - \frac{K_t . I_a^{sat}}{J.s^2} \qquad \{A.3.19\}$$

Where Error ^{sat} is the error function during saturation. The superposition constants K_{sv} and K_{sa} could be adjusted to compensate for saturating profiles, but this would require re-calibrating the system every time the incremental period was altered. One of the requirements for the controller was that this should not be necessary, because the profile period varies with the machine speed, which must be readily variable over a wide speed range. Therefore feedforward compensation, an alternative form of error superposition, was assessed for its ability to drive saturating profiles.

A.3.4.6 Proportional plus derivative control with feedforward compensation

Feedforward compensation is similar to error superposition in that elements of the error function are used to compensate for the error. However, with feedforward the compensation elements enter the loop after the forward path controller. This is shown for p-d control with velocity and acceleration feedforward in figure A.3.15.

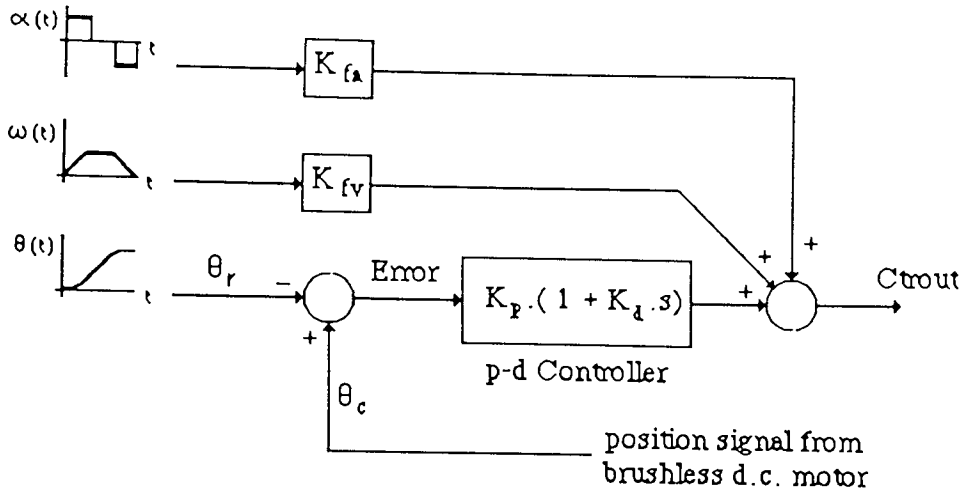


Figure A.3.15. Proportional plus derivative controller with velocity and acceleration feedforward.

The transfer function of the controller is

$$\text{Ctrout} = K_p.\left(\text{Error} + K_d\frac{d.\text{Error}}{dt}\right) + K_{fv}.\frac{d.\theta_r}{dt} + K_{fa}.\frac{d^2.\theta_r}{dt^2} \qquad \{A.3.20\}$$

where K_{fv} and K_{fa} are the velocity and acceleration feeforward gains respectively. The brushless d.c. motor was simulated with the control action of equation A.3.20. Table A.3.2 summarises the results for p-d plus velocity feeforward, with K_p= 8.0, K_d= 0.03, K_{fv}= 9 x 10^{-3} and K_{fa}= 0 and for p-d plus velocity and acceleration feedforward, with K_p= 8.0, K_d= 0.03, K_{fv}= 9 x 10^{-3} and K_{fa}= 3.5 x 10^{-5}.

For all three of the simulated profiles, the incremental overshoot and settling performance of the velocity feedforward system was superior to that of the p-d system, but inferior to that of the p-d plus profile superposition system. The p-d plus velocity and acceleration feedforward system had an incremental performance that was superior to all the other controllers evaluated. A plot of the simulated performance of the system driving the 22.5ms profile is shown in figure A.3.16. The peak overshoot to the 22.5ms profile was 2.7% and the 2% settling time was 27.2ms. This was the only system to achieve a shorter settling time to the 22.5ms profile than to the 30ms profile, although the 22.5 ms profile did induce current saturation in the driver.

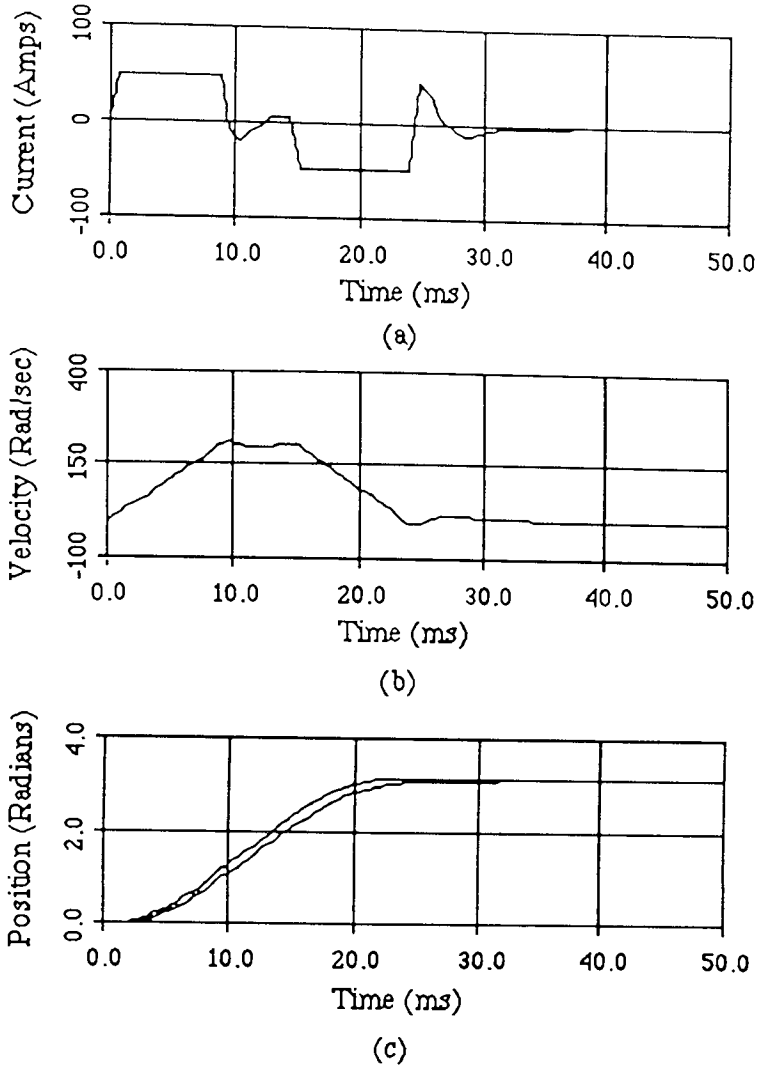


Figure A.3.16. Simulated reponse of proportional plus derivative controlled system with velocity and acceleration feedforward to 22.5ms trapezoidal profile.

(a) Current response

(b) Velocity response

(c) Position response.

A.3.4.7 Trapezoidal velocity increment controller conclusions.

The optimum algorithm for controlling the brushless d.c. motor system through saturating and non saturating trapezoidal velocity profiles is the proportional plus derivative with velocity and acceleration feedforward controller. This produces good overshoot and settling performance to varying duration profiles with no controller parameter adjustment being required. The algorithm is straight forward to implement and satisfies the requirements of the control system scheme.

Controller Action	t_m (ms)	Peak Error (rad)	Peak Error (ms)	Peak Overshoot %	Peak Overshoot (ms)	Settling Time (ms) 5 %	Settling Time (ms) 2 %	Sat'n
Proportional	50.0	0.265	18.4	1.33	52.0	----	----	No
	30.0	0.539	12.8	4.4	36.0	----	36.8	No
	22.5	0.9224	12.8	36.9	31.2	62.4	71.2	Yes
P-I	50.0	0.156	12.0	5.5	45.6	46.1	53.6	No
	30.0	0.443	11.2	11.8	29.6	34.4	61.6	Yes
	22.5	400 % Overshoot. No 100 ms settling.						Yes
Switched P-I	50.0	0.156	12.0	5.5	45.6	46.1	53.6	No
	30.0	0.5386	12.8	2.7	30.0	----	32.8	Yes
	22.5	0.9188	12.8	41.1	31.2	66.4	93.6	Yes
P-D	50.0	0.2227	25.6	0.6	50.0	----	----	No
	30.0	0.3713	20.0	1.6	30.0	----	----	No
	22.5	0.867	12.8	30.8	30.4	51.2	60.8	Yes
P-D + Profile Superposition	50.0	0.017	38.0	0.3	51.0	----	----	No
	30.0	0.023	12.0	1.2	32.0	----	----	Yes
	22.5	0.2115	9.6	7.48	24.8	28.0	31.2	Yes
P-D + Velocity Feedforward	50.0	0.1373	17.6	0.6	51.2	----	----	No
	30.0	0.246	11.2	1.47	31.2	----	----	No
	22.5	0.534	11.2	10.6	26.4	31.2	36.0	Yes
P-D + Velocity + Acceleration Feedforward	50.0	0.1185	33.6	0.2	50.0	----	----	No
	30.0	0.1947	20.0	0.54	30.4	----	----	No
	22.5	0.341	9.6	2.7	24.8	----	27.2	Yes

Table A.3.2. Trapezoidal increment controller performance summary.

A.3.5.1 Complex intermittent profiles.

Many of the requirements for high speed intermittent machines have complex profiles consisting of linked segments, with each segment having differing following error and settling time requirements. The third transfer slider is typical of this type of motion. Most of the composite profiles may be most simply defined as a series of linked position ramps, but this simple definition does not necessarily produce the optimum actuator motion. In this section the position control system's responses to composite position ramp and velocity ramp profiles are evaluated to assess the optimum composite profile design technique.

The position and velocity curves of the two types of composite requirement profile are shown in figure A.3.17. In the linked position ramp profile step discontinuities occur in the first derivative requirement. In the linked velocity ramp the discontinuities are second order, acceleration, steps.

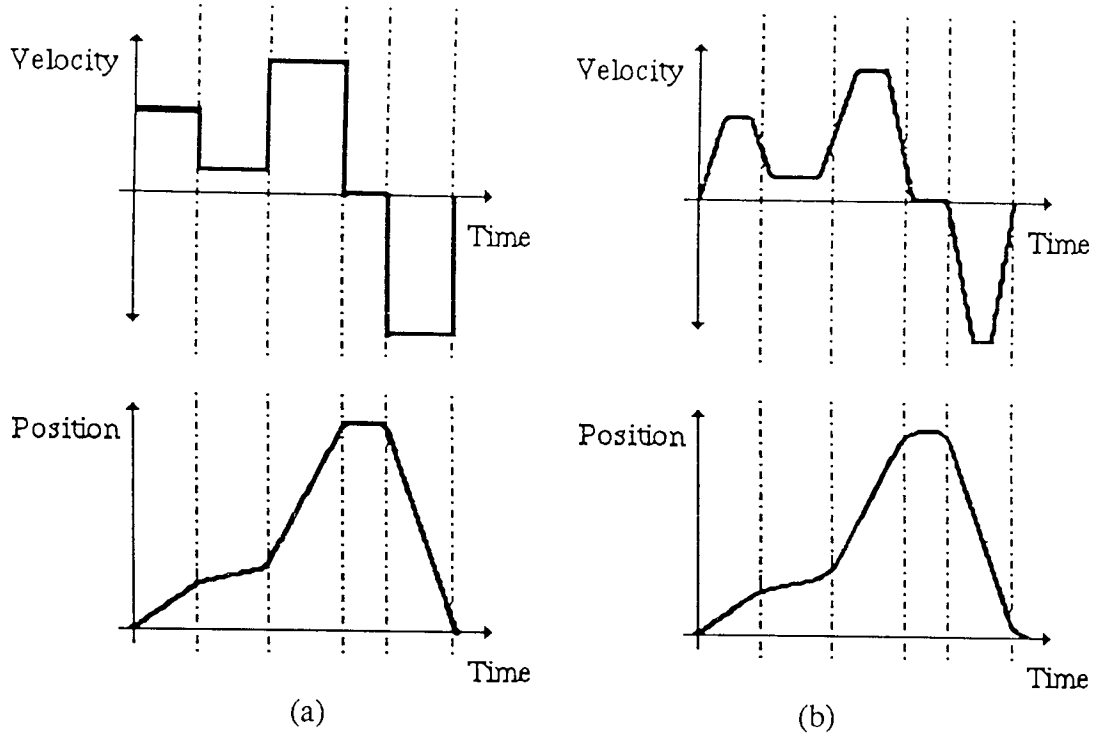


Figure A.3.17.Profile linking to form composite incremental profile.
(a) Linked position ramp profile.
(b) Linked velocity trapezoid profile.

Increasing the order of the discontinuities from first to second reduces the magnitude of the acceleration demand to a finite value, which produces a smoother position requirement curve and reuces the likelyhood of current saturation occuring in the system. The current demand would reduce further if the discontinuity were moved to the third order, producing acceleration ramp profiles. However, in practice the 1ms sample period of the position controller results in acceleration and deccleration ramp segment lengths of less than two samples, which in cffcct produces acceleration step discontinuities. In addition, the design and implementation of position profiles based on constant acceleration segments is relatively straight forward. As a result, commercial axis controllers are available having both position ramp and velocity ramp profiling functions. Therefore this analysis concentrates on linked position ramp and velocity ramp profiles.

A.3.5.2 Evaluating linked profile performance by simulation.

The profiles used in the analysis were based on the required motion of a motor driving the third transfer slider through a drive radius of 100mm. The motor was simulated driving a matched inertia load in an in-out motion of peak magnitude 2 radians. During the inward stroke the motor was required to slow to a constant velocity of not more than 10 rad/s between the positions 0.75 radians and 0.9

radians into the stroke. At the extreme 2 radians point of the motion the motor was required to dwell for the period from 0.55.TM to 0.65.TM, where TM was the total period of the motion.

Two types of profile were developed according to this specification, the linked position ramp and linked velocity ramp. The ACSL model definition information for the position ramp profile is given in figure A.3.18.

```
CONSTANT TM=0.1, TRAVEL=2.0 $"Profile constants"

INITIAL
    "       "
M1=2.3333*TRAVEL/TM
M2=(2*TRAVEL)/(3*TM)
M3=2.2*TRAVEL/TM
M4=2.8571*TRAVEL/TM
    "       "
END $"Of initial"

DERIVATIVE CONTIN
    "       "
OMEGA1 = M1 * (STEP(0.0)-STEP(0.15*TM))
OMEGA2 = M2 * (STEP(0.15*TM)-STEP(0.3*TM))
OMEGA3 = M3 * (STEP(0.3*TM)-STEP(0.55*TM))
OMEGA4 = M4 * (STEP(0.65*TM)-STEP(TM))
VSLIDE = OMEGA1 + OMEGA2 + OMEGA3 - OMEGA4 $"Required velocity"
RSLIDE = INTEG(VSLIDE,0.0) $"Required position"
    "       "
END $"Of derivative"
```

Figure A.3.18. Linked position ramp profile definition.

The magnitude and duration of each profile segment was governed by the total distance to be moved by the slider, TRAVEL and the period of the in-out motion, TM. The magnitude of each velocity step is evaluated in the INITIAL block of the model, with the duration of the steps being defined as functions of TM in the DERIVATIVE block.

The ACSL model definition of the linked velocity ramp profile is given in figure A.3.19 As with the linked position ramp profile, the magnitude and duration of each segment vary with the profile parameters TRAVEL and TM, with the magnitudes of each acceleration step being evaluated in the INITIAL block and the steps being defined in the DERIVATIVE block.

The two profiles were modelled with five periods, TM, 200ms, 150ms, 100ms, 80ms and 70ms, which corresponded to simulated machine speeds of 175, 235, 350, 440 and 500 cycles per minute respectively. The linked velocity ramp profile was simulated with the p-d plus velocity and acceleration feedforward controlled system developed in section A.3.4.6. The linked position ramp profile was

simulated with the switched p-i plus velocity feedforward controlled system developed in section A.3.3.8. This allowed a comparison to be made between the optimum position ramp and velocity trapezoid controllers. In addition, the p-d with velocity feedforward controlled system was simulated driving the linked position ramp to provide a further comparison of system performance.

```
CONSTANT TM=0.1, TRAVEL=2.0 $"Profile constants"

INITIAL
    "       "
ACC1 = 100 * ((0.9*TRAVEL)-(0.016666)/(TM * TM)
OM1P = ACC1 * TM/20
OM1F = 2/(3 * TM)
ACC2 = 20 * (OM1P - OM1F)/TM
ACC3 = 78.125 * ((0.55 * TRAVEL) - 0.13333)/(TM * TM)
OM2P = OM1F + (ACC3 * 0.08 * TM)
ACC4 = OM2P/(0.08 * TM)
ACC5 = TRAVEL/(0.0242 * TM * TM)
    "       "
END $"Of initial"

DERIVATIVE CONTIN
    "       "
ALPHA1 = ACC1 * (STEP(0.0) - STEP(TM/20))
ALPHA2 = ACC2 * (STEP(TM/10) - STEP(3*TM/20))
ALPHA3 = ACC3 * (STEP(0.3 * TM) - STEP(0.38 * TM))
ALPHA4 = ACC4 * (STEP(0.46 * TM) - STEP(0.54 * TM))
ALPHA5 = ACC5 * (STEP(0.67 * TM) - STEP(0.78 * TM))
ALPHA6 = ACC5 * (STEP(0.89 * TM) - STEP(TM))
ASLIDE = ALPHA1 - ALPHA2 + ALPHA3 - ALPHA4 - ALPHA5 + ALPHA6
VSLIDE = INTEG(ASLIDE,0.0)
RSLIDE = INTEG(VSLIDE,0.0) $"Required position"
    "       "
END $"Of derivative"
```

Figure A.3.19. Linked velocity ramp profile definition.

The results of the simulation evaluation are summarised in Table A.3.3. Reading from left to right, the columns in Table A.3.3 include the position controller action and type of profile; the period of the profile; the magnitude of the peak following error during the insertion stroke in millimetres, based on a 100mm drive radius; the time at which the peak insertion error occured; the pak overshoot at maximum insertion, in millimetres; the time at which the peak overshoot occured; the peak following error during the retraction stroke, in millimetres; the time at which the peak retraction error occured; the pak overshoot at maximum retraction, in millimetres; the time at which the peak retraction overshoot occured; whether current saturation occured in the driver during the motion. The values listed in Table A.3.3 were obtained from the ACSL print outs of the systems' responses. These have been rounded to 4 significant figures.

A.3.5.3 Comparison of profile driving performances.

The linked ramp performance evaluation simulations demonstrated that for all five profile periods the switched p-i with velocity feedforward controlled system following the linked position ramps produced the least satisfactory performances.The peak insertion overshoot varied from 4.5mm for the non saturating 200ms period profile to 12.95mm for the 70ms saturating profile. The following errors during the motions were larger than those of either of the other two systems, as were the overshoots at the end of the retraction strokes.

The system having the best position overshoot performance was the p-d with velocity and acceleration feeforward controlled system following the linked velocity ramp profile. For all five profile periods the system had no insertion overshoot, the values given in Table A.3.3 being the peak recorded error during the maximum insertion dwell. The position response of the system to the 100ms profile is shown in figure A.3.20. This demonstrates the good response obtained, with the following error, which peaks at 5.66mm, having the effect of introducing a phase shift between the required and actual position profiles.

The p-d plus velocity feedforward controlled system driving the linked position ramp profiles produced responses that fell between those of the other two systems. Peak insertion position overshoots occured during the three shortest period profiles, when the system was driven into current saturation.

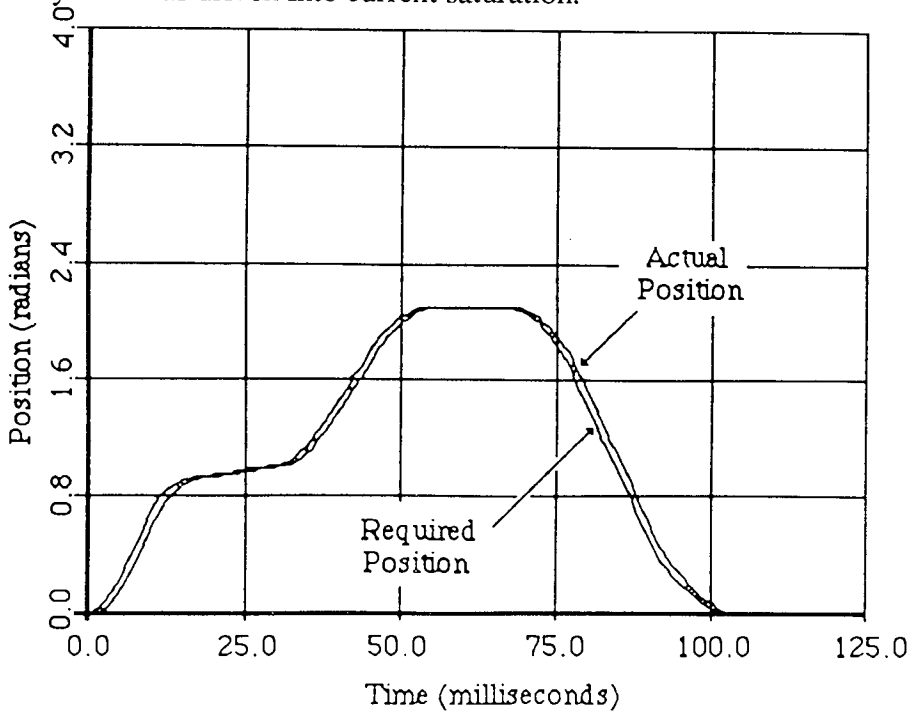


Figure A.3.20. Response of proportional plus derivative with velocity and acceleration feedforward controlled system to 100ms linked velocity ramp profile.

A.3.5.4 Linked ramp profile conclusions.

The proportional plus derivative with velocity and acceleration feedforward controlled system driving the linked velocity ramp profile produces the best actuator performance for both saturating and non saturating profile periods. The system has the smallest incremental overshoot for all profile durations and has very good tolerance of current saturation within the driver, with no adjustment of controller parameters being necessary for changes of the period of the profile.

When driving short period linked position ramp profiles, the proportional plus derivative with velocity feedforward controlled system produces smaller following errors and smaller peak insertion overshoots than the proportional plus switched integral with velocity feedforward controlled system. This is because the segments of the profile are shorter than the relatively long time constant of the p-i controlled system, so the integrator does not have sufficient time to reduce the following error.

Controller Action	TM (ms)	Peak Insertion Error		Insertion Overshoot		Peak Retraction Error		Retraction Overshoot		Sat'n?
		mm	ms	mm	ms	mm	ms	mm	ms	
Switched p-i + velocity feedforward (Linked position ramp profile)	200	4.425	7.2	4.15	117.6	5.35	137.6	5.375	207.2	No
	150	5.9	7.2	5.51	89.6	7.4	104.0	7.621	156.1	No
	100	8.85	7.2	8.92	61.6	13.7	70.4	9.835	108.0	No
	80	11.05	7.2	11.8	51.0	20.05	56.8	12.6	87.2	No
	70	12.65	7.2	12.95	45.6	27.4	50.61	14.92	76.8	Yes
p-d + velocity feedforward (Linked velocity ramp profile)	200	2.2	3.2	*0.95	110.0	2.68	132.8	¥0.85	200.0	No
	150	2.9	3.2	*1.85	82.5	3.675	100.8	¥1.615	150.0	No
	100	5.435	4.0	2.165	57.2	7.185	68.8	¥4.15	100.0	Yes
	80	6.23	3.2	3.2	48.0	10.2	56.0	5.285	84.0	Yes
	70	7.53	4.0	2.9	42.4	10.7	49.6	6.865	74.4	Yes
p-d + velocity + acceleration feedforward. (Linked velocity ramp profile)	200	2.79	20.0	*0.03	110.4	2.875	177.0	¥0.185	200.0	No
	150	3.645	15.2	*0.185	84.0	3.815	132.8	¥0.234	150.0	No
	100	5.445	8.0	*0.343	56.8	5.66	88.8	¥0.65	100.0	No
	80	7.14	6.4	*0.445	44.0	7.09	71.2	¥0.98	80.0	Yes
	70	9.29	6.4	*0.445	38.6	7.985	62.4	¥0.80	70.0	Yes

* No overshoot. Error is peak value during maximum insertion dwell period

¥ No overshoot. Error is value at end of requirement profile

Table A.3.3. Comparison of Linked Position Ramp and Linked Velocity Ramp Profile Performances.

Appendix 4.

Modular Machine Systems

Paper published in the proceedings of the conference on high speed machinery, organised by the Solid Mechanics and Machines Group of the Institution of Mechanical Engineers in collaboration with S.E.R.C.

24th November 1988

Modular machine systems

L FENNEY*, MSc,
C M DRAPER*, BSc
K FOSTER**, MA, PhD, CEng, FIMechE
D J HOLDING*, BSc, PhD, CEng, FIEE, MinstMC, MBCS
*Aston University, **Birmingham University

SYNOPSIS

Aston University

Appendix 5

Characteristics and Dynamic Performance of Electrical and Hydraulic Servo-Drives

Paper published in the proceedings of the first JHPS Symposium on fluid power.

Tokyo, Japan

13th – 16th March 1989.

CHARACTERISTICS AND DYNAMIC PERFORMANCE OF ELECTRICAL AND HYDRAULIC SERVO-DRIVES

Foster, K.* and Fenney, L.**

*University of Birmingham
Head of School of Mechanical Engineering
Birmingham
England

**University of Aston
Department of Mechanical and Production Engineering
Birmingham
England

Aston University

References

(1) **Sweeney, G.**; 1988; "Trends in the design of food packaging machinery."; Proc. ImechE. seminar into The Engineering Contribution To Food Packaging In The 90's.; October 1988.

(2) **Seaward, D.R.** and **Johnson, R.C.**; 1988; " Technology transfer from academia to industry of a phase synchronised drives project." Proc. ImechE. Conf. on High Speed Machinery.; November 1988; pp 11–18.

(3) **Jones, B.** and **Leighton, N.J.**; 1987; " Enhanced performance of sheet feed systems using electrical drives." Proc. ImechE. Seminar on High Speed Machinery.; May 1987.

(4) **Leighton, N.J., Jones, B.** and **Fletcher, H.**; 1988; " Precision press feed systems using digitally controlled drives." Proc. ImechE. Conf.on High Speed Machinery.; November 1988; pp 3–9.

(5) **Stamp, K** and **Rees Jones, J.**; 1988; " High speed planar guidance and coordination with programmable motion control." Proc. ImechE. Conf.on High Speed Machinery.; November 1988; pp 29–33.

(6) **Rees Jones, J.**; 1988; " Some dynamic performance issues in programmable continuous and logically controlled motions in machines." Proc. ImechE. seminar into The Engineering Contribution To Food Packaging In The 90's.; October 1988.

(7) **Bruno, C.L.**; 1988; " The replacement of mechanical drives by electro–mechanical units."; Proc. ImechE. seminar into The Engineering Contribution To Food Packaging In The 90's.; October 1988.

(8) **Marks, C.J.**; 1988; " Packaging machinery design."; Proc. ImechE. seminar into The Engineering Contribution To Food Packaging In The 90's.; October 1988.

(9) **Weston, R.H., Harrison, R., Booth, A.H.** and **Moore, P.R.**; 1989; "A new concept in machine control."; Dept. Manufacturing Engineering, Loughborough University of Technology.

(10) **Firoozian, R.** and **Foster, K.**; 1985; " The choice of a servo motor for a specific application." ImechE.; C276/85.

(11) **Carlisle, B.H.**; 1985; " High tech motors revolutionise motion control."; Machine Design.; July 25th 1985, pp 73–79.

(12) **Barber, N.**; 1985; " Coming to terms with brushless servo drives." Electric Drives and Controls.; April/May 1985.

(13) **Lawrenson, P.J., Stephenson, J.M., Blenkinsop, P.T., Corda, J.** and **Fulton, N.N.**; 1980; " Variable–speed switched reluctance motors."; I.E.E. Proc., Vol. 127, Pt.B, No. 4; July 1980; pp 253–265.

(14) **Kenjo, J.** and **Nagamori, S.**; 1985; "Permanent magnet and brushless d.c. motors."; Clarendon Press, Oxford.

(15) **Liska, M.** and **Ulrich, B.**; 1976; "Magnetic steppermotors and brushless d.c. motors: principles and applications in drive units." Proc. I.E.E. Conf. on Small Electrical Machines.; March 1976; pp 15–18.

(16) **Horner, G.R.** and **Lacey, R.J.**; 1983; "High performance brushless p.m. motors for robotics and actuator applications."; Hightech components limited, servo house, Tadley, Hampshire.

(17) **Christensen, L.**; 1988; " Permanent magnet brushless servo motor development." Proc. MMPA Permanent Magnet Users Conf.; University of Dayton.; Dayton Ohio.

(18) **Tal, J.**; 1984; "Motion control by microprocessors."; Galil motion control Inc.; U.S.A.

(19) **Barber, N.T.**; 1984; ' Benefits of using brushless drives on high–performance robots."; Proc. The British Robotics Assoc. Conf.; May 1984; pp 96–105.

(20) **Tomasek, J.**; 1983; " Analysis of torque–speed performance limits in brushless d.c. servo systems." Proc. 12th annual symposium on incremental motion control systems and devices.; University of Illinois.

(21) **Tal, J.**; 1973; " The optimal design of incremental motion servo systems."; Proc. 2nd annual symposium on incremental motion control systems and devices.; University of Illinois.

(22) **Jocz, A.**; 1973; " Selecting high performance incremental motion devices."; Proc. 2nd annual symposium on incremental motion control systems and devices.; University of Illinois.

(23) **Archer, J.R.** and **Blenkinsop, P.T.**; 1986; " Actuation for industrial robots."; Proc. Inst. Mech Engrs; Vol 200, No. B2; pp 85–89.

(24) **Newton, G.C.**; 1981; " Selecting the optimum electric servomotor for incremental positioning applications."; Proc. 10th annual symposium on incremental motion control systems and devices.; University of Illinois.

(25) **Arnold, F.**; 1984; " Power rate – a most important figure of merit for the incremental motion designer.; Proc. 13th annual symposium on incremental motion control systems and devices.; University of Illinois.

(26) **Floresta, J.**; 1985; " Designing incremental motion systems with power rate."; Proc. 14th annual symposium on incremental motion control systems and devices.; University of Illinois.

(27) " D.C.motors, speed controls, servo systems."; Engineering handbook; Electro–Craft Corp.

(28) **Bell,R.** and **de Pennington, A.**; 1970; " Active compensation of lightly damped electrohydraulic cylinder drives using derivative signals."; Proc. ImechE., Vol.184, pt. 1.

(29) **Tal, J.**; 1981; " Performance improvement in band limited systems."; Proc. 10th annual symposium on incremental motion control systems and devices.; University of Illinois.

(30) **Fenney, L.**; 1985; " The use of computer aided methods in the design of a position and velocity servo for a computer controlled coordinate table."; M.Sc thesis; Aston University, Birmingham.

(31) Advanced Continuous Simulation Language Reference Manual.; 1986; Mitchell and Gauthier Associates.

(32) **Raven, F.H.**; 1978; " Automatic control engineering."; McGraw–Hill International Book Company.

(33) **Borrie, J.A.**; 1986; " Modern control systems."; Prentice/Hall International.

(34) **Atkinson, P.**; 1983; " Feedback control theory for engineers."; Heineman Educational Books, London.

(35) **Sandoz, D.J.**; 1982; " A survey of computer control."; Published in " Computer control of industrial processes."; I.E.E. Control Engineering, 21.; Editors S. Bennett and D.A. Linkens.

(36) **Enslow, P.H.** and **Saponas, T.G.**; ; "Parallel control in distributed systems - a discussion of models."; published in "Parallel processing system"; Editor D.J. Evans; Cambridge University Press. pp 43-72.

(37) **Sloman, M.S.**; 1982; "Communications for Distributed Control."; published in "Computer Control of Industrial Processes." Editors S.Bennett and D.A. Linkens; Peter Peringus Ltd for I.E.E.; pp 112-130

(38) **Anderson, A.** and **Jensen, D.G.**;1975; "Computer Interconnection Structures: Taxonomy, Characteristics and Examples."; Computing Surveys 4, Dec.1975, pp 197-213.

(39) **Weitzman, C.**;1980;"Distributed Micro/Minicomputer Systems, Structure, Implementation and Application."; Prentice-Hall Inc; pp 81-84.

(40) **Paker, Y.**; 1983; "MultiiMicroprocessor Systems"; Academic Press Inc.; pp 42–46.

(41) **Katz, P.**; 1981; "Digital Control Using Microprocessors."; Prentice/Hall International.