Aston University

Some pages of this thesis may have been removed for copyright restrictions.

If you have discovered material in AURA which is unlawful e.g. breaches copyright, (either yours or that of a third party) or any other law, including but not limited to those relating to patent, trademark, confidentiality, data protection, obscenity, defamation, libel, then please read our Takedown Policy and contact the service immediately

Design Methodology for the Servo Control of High Speed Multi-Axis Machinery

Decoupling and Synchronisation of a Generic Machine by Blockwise Decentralised MIMO Control

By

Andrew Yuen Soun Chan

Doctor of Philosophy

The University of Aston In Birmingham

August 1998

This copy of the thesis has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with its author and that no quotation from the thesis and no information derived from it may be published without proper acknowledgement.

The University of Aston In Birmingham

Design Methodology for the Servo Control of High Speed Multi-Axis Machinery

Decoupling and Synchronisation of a Generic Machine

by Blockwise Decentralised MIMO Control

By Andrew Yuen Soun Chan

Doctor of Philosophy, 1998.

SUMMARY

Traditional machinery for manufacturing processes are characterised by actuators powered and co-ordinated by mechanical linkages driven from a central drive. Increasingly, these linkages are replaced by independent electrical drives, each performs a different task and follows a different motion profile, co-ordinated by computers. A design methodology for the servo control of high speed multi-axis machinery is proposed, based on the concept of a highly adaptable generic machine model. In addition to the dynamics of the drives and the loads, the model includes the inherent interactions between the motion axes and thus provides a Multi-Input Multi-Output (MIMO) description.

In general, inherent interactions such as structural couplings between groups of motion axes are undesirable and needed to be compensated. On the other hand, imposed interactions such as the synchronisation of different groups of axes are often required. It is recognised that a suitable MIMO controller can simultaneously achieve these objectives and reconciles their potential conflicts. Both analytical and numerical methods for the design of MIMO controllers are investigated.

At present, it is not possible to implement high order MIMO controllers for practical reasons. Based on simulations of the generic machine model under full MIMO control, however, it is possible to determine a suitable topology for a blockwise decentralised control scheme. The Block Relative Gain array (BRG) is used to compare the relative strength of closed loop interactions between sub-systems. A number of approaches to the design of the smaller decentralised MIMO controllers for these sub-systems has been investigated. For the purpose of illustration, a benchmark problem based on a 3 axes test rig has been carried through the design cycle to demonstrate the working of the design methodology.

Indexing terms :- MOTION CONTROL, DISTRIBUTED CONTROL, CONTROL STRUCTURE
BLOCK RELATIVE GAIN ARRAY, MULTIVARIABLE CONTROL

Acknowledgements

I would like to express my earnest gratitude to my supervisors Dr. S. D. Garvey and Prof. M. T. Wright for their patient guidance and support through the preparation of this thesis. Likewise, I would also thank Dr. D. Seaward and Mr. M. J. Cahill of Molins Plc. for their kind assistance. Special thanks to the assistant staff of the Machine Control & Drives Laboratory of Aston University whose helps have always been available.

Contents

List of Figures

List of Tables

1. Introduction

1.1 Topology Selection for Servo Control Systems

The current practice of servo design for independently driven multi-axis machinery (Figure 1-1) can be summarised as follows; parallel feedback loops are closed one at a time starting with the fastest (smallest time constant) loop. Once stability is ensured, slower loops are closed subsequently. Stability of the slower loops is ensured by detuning the faster loops. Cross axis interactions are ignored unless they become a problem. It is recognised that more performance can be harnessed for the following reasons:

- the practise of sequential loop closing is not a 'simultaneous' synthesis and the resultant design tends to be too conservative
- the provision of information cross feeding between axes can compensate cross axis disturbance, it also remove the restriction that system axes have to be isolated in design
- synchronisation can be provided by forced coupling

It is therefore clear that improvement can be made with the servo design if cross axes interaction is to be taken into account at an early stage. To handle all the cross interaction, a Multi-Input Multi-Output (MIMO) approach is required. Both the system model and the servo design methodology will have to be in a MIMO paradigm.

A MIMO controller designed to meet the performance specification will satisfy all the synchronisation requirements, which are not guaranteed by sequential loop closing. The cross couplings inherent to the system will also be compensated simultaneously. The realisation of a full MIMO controller, however, may be found to be prohibitive due to the cost of communication and computation. The alternative (Figure 1-2) is a block-decentralised implementation of a centralised designed MIMO controller. Clearly, the determination of a suitable topology for the block structure is the key to such a distributed approach.

In short, the design philosophy can be summarised by the following. Some form of imposed coupling is needed to compensate for inherent couplings and satisfy the

synchronisation requirements simultaneously. Such imposed coupling is designed into a centralised MIMO controller and its performance will be the upper bound of the "achievable" performance with distributed control. A decentralised topology is then determined from this controller by considering the overall level of interactions between subsystems (blocks). As an interaction measure, block relative gain (BRG) array is found to be a useful tool. Finally, a decentralised implementation is applied to the overall system.

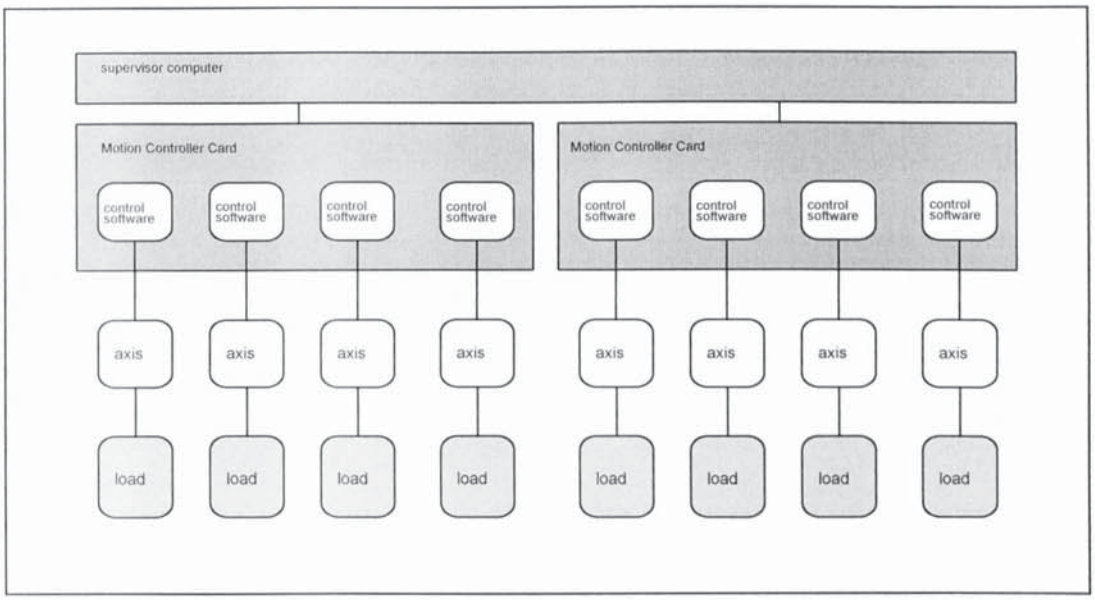

Figure 1-1 Axis with Independent Servos

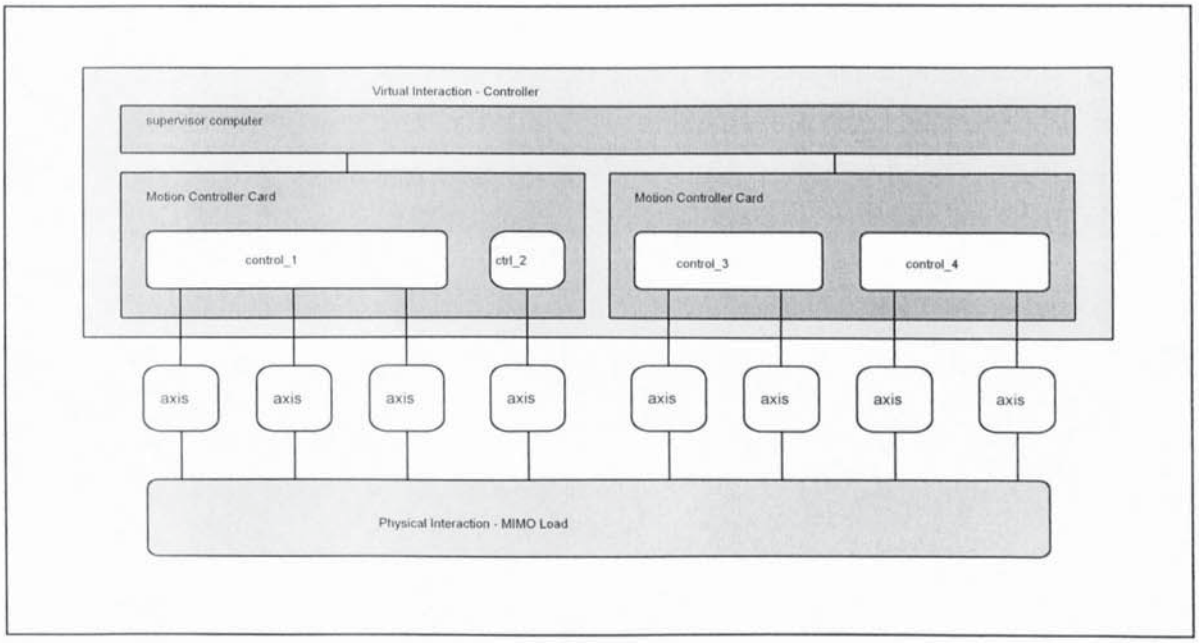

Figure 1-2 Decentralised Control - A Different Servo Architecture

The design methodology is aimed to formalise and automate, as far as possible, the process of design and synthesis of the control strategy of multi-axis high speed machinery. The target would be a distributed computer control system. Special emphasis is placed on the control topology (the way axes are grouped into subsystems) and the main focus is on how to deal with the interaction between subsystems.

A process of system decomposition takes into consideration the electro-mechanical properties (intrinsic constraints) and the target behaviour (specification - imposed constraints) of the machine and suggests possible topologies. A decentralised control scheme is employed which involves deriving local multi-variable controllers for each subsystem resulted in the decomposition process. The resultant control system is then evaluated by simulation.

1.2 Main Contributions

The aim of the project is to improve performance of these machines by adapting a MIMO approach. Attention is focused on the co-ordination of the axes rather than the performance of individual axis. The methodology is constructed with the aim to be implemented as a computed aided design software for servo systems. At the end of the project, the following issues has been considered:

- Formal and structural design methodology with a framework that is easily extendible
- Support for new possibilities in machine design is achieved by a more integrated and modular servo architecture, i.e. generic machine design
- Generic Machine Model - Isolation of various physical components. The model is set in a MIMO paradigm
- Off diagonal terms as imposed coupling imply synchronisation and compensate for inherent couplings. Cross axis interactions are taken into consideration
- LQG optimisation has been extended to deal with synchronisation
- Numerical search methods for decoupling axes were discussed
- A numerical search method for a "balanced" control solution has been achieved
- Block decentralised control for "optimal" cost in communication & computation

- A centralised control design for decentralised implementation
- A custom design toolset in the form of a MATLAB toolbox
- Design and commission of a MIMO test rig
- Implementation of MIMO controllers on a DSP
- Demonstration of all above

1.2.1 Advantages with the Proposed Design Methodology

The proposed methodology offers a design that is based on a MIMO archetecture. The resultant control systems are better integrated abd yet offer more flexibility.

On the Machine Performance Level

It is possible to have :

- less noise
- less power consumption
- better accuracy
- better flexibility for configuration and operation
- lower maintenance
- shorter down time for process modification, reconfiguration and change over
- lower cost for modification
- variable machine speed and hence graceful start up and stop mode, diagnostic jogging mode
- arbitrary motion profiles (electronic gearbox, cam) for active functional shafts
- better sequencing of motions
- better synchronisation between axes, disregard distance involved
- less interference between axes that accelerated at different rate (e.g. shock loading)

On the Design Level

It is possible to have :

- faster development
- lighter weight
- lower machine cost
- non-square systems - external encoders and sensors can be integrated
- less design restrictions - system axes are not necessary isolated in design
- modular design with simple parts replacement
- simplification, enhancement and synthesis of mechanism (e.g. software gearbox)
- possibility of a machine with changing reference - action on demand at different locations

1.3 An Industrial Example -- Saddlebinding Machines

Saddle binders are machines that collate folded paper into stacks, to be stitched and trimmed into books or magazines. A top of the range machine involves sophisticated loader and precision transfer system and can operate up to 12,000 cycles per hour. The actual collating process is illustrated in the following schematic (Figure 1-3).

A series of vertical feeding hoppers are arranged in a line and positioned over a delivery belt. These hoppers will drop folded paper one on top of another to form stacked sets. Individual hoppers can be added, removed or decoupled depending on the requirement. The prestitched magazines are passed to the stitching and trimming unit at the end of the delivery system for binding. The conveyor is actually a gripper shuttle system that positions the paper and detects any incorrect sets. Incorrect sets will pass through the stitcher to a reject tray.

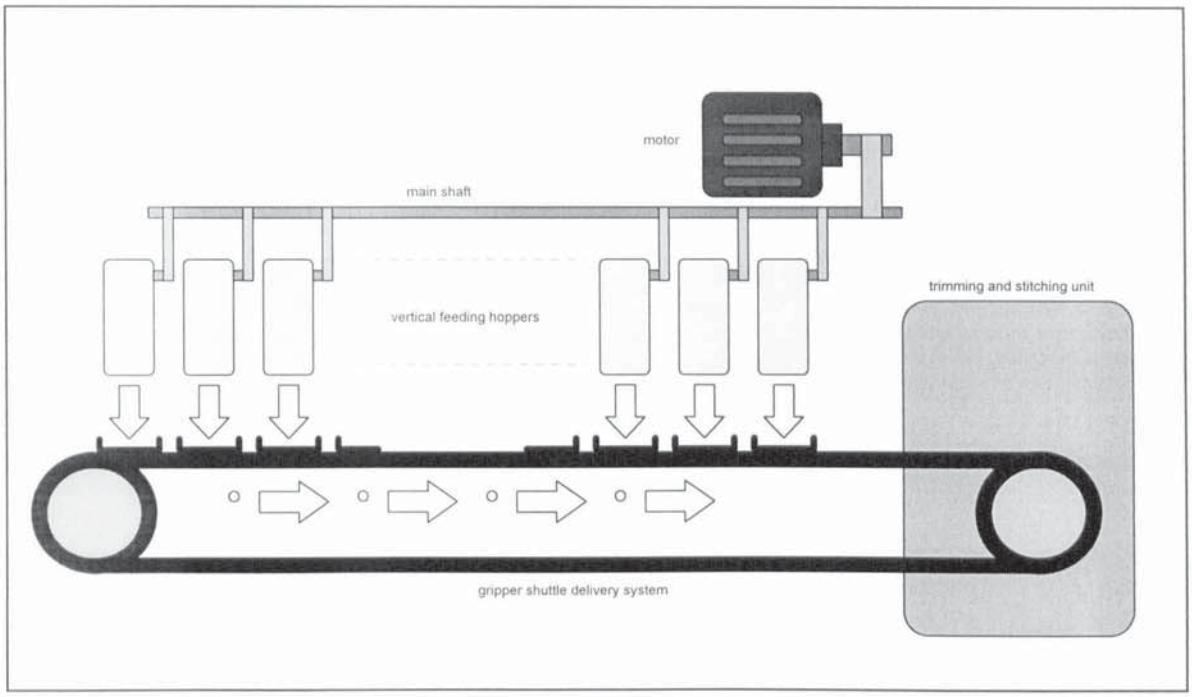

Figure 1-3 Collating Process for a Saddlebindder

The overall process is controlled and co-ordinated by a PLC (programmable logic controller) and all the hoppers derive their power from an AC inverter/motor through a main shaft. A full blown version of such a system involves over 30 hoppers and the main shaft can be up to 20m long. This means that transportation can be a problem as the machine has to be disassembled after testing and be reassembled at customer's location. More importantly, the length and the finite stiffness of the main shaft mean the positions of the two ends will be different. This creates problems especially at start-up and shut down.

A latest trend in the publishing industry is the availability of customised magazine in which the content of a magazine is tailored to the taste of the reader. For example, if an individual is interested in fishing rather than, say, golfing, only the fishing section will be included in that particular copy of the magazine. In addition, ink jets are installed on the hoppers so that specific information related to the customer can be printed on specific pages. Other gimmicks including postcard inserts are also available. These requirements mean that the feeding hoppers no longer perform identical job but instead may require slightly different execution time and power in a cycle.

One possible development is to replace the long collating system with a series of shorter modules (as shown in the schematic Figure 1-4) that are co-ordinated by electronic rather than mechanical means. These modules will have smaller motors and shorter shafts, and

hopefully less moving parts and hence higher accuracy, reduced noise and lower power consumption. Such an approach will be advantageous as it increases flexibility, reduces upgrade cost, set-up time and ultimately higher performance and lower running cost. The challenge will be to design and implement an overall control system that can cope with the modular parts and deliver the required performance.

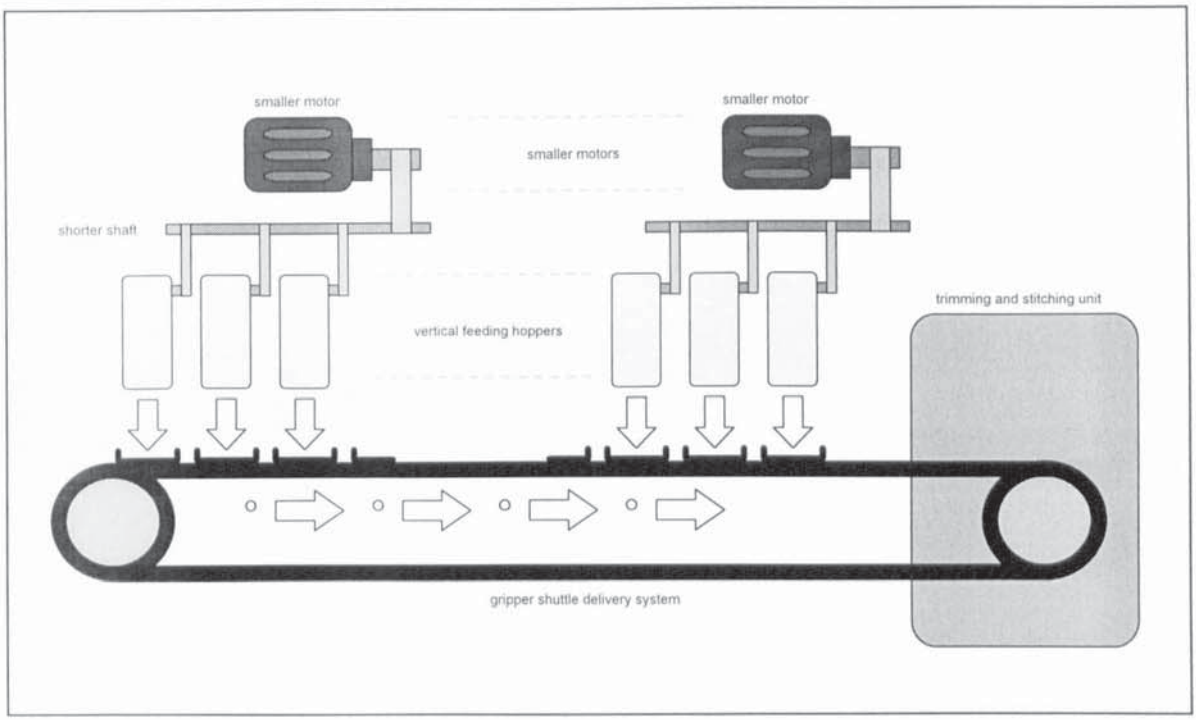

Figure 1-4 Saddlebindder with Collator Modules

1.4 Typical Multi-Axis High Speed Machines

In order to clarify what a multi-axis high speed machine is, a number of examples are collated and shown in Table 1-1. Admittedly there is no formal definition since most of these machines are recognised by their functionality rather than by their constructions. A better understanding of the technology involved, especially the issue of the control system, can be gained by looking at a particular machine in details. The rest of this section will be devoted to the description of a tea bag making machine which is under development in industry.

Table 1-1 Examples of Multi-Axis High Speed Machines

Molins PASSIM Cigarettes Maker and Plug Assembler	The PASSIM produces up to 7000 filter cigarettes per minute. There are many features including automatic paper bobbin change to avoid stoppage, reduced main shaft speed to reduce noise. The maker is managed by PLCs with microprocessors serving an automated inspection and data collection system while the plug assembler is controlled by microprocessor.
Molins SP1 Soft Packer for cigarettes	The SP1 produces 350 soft packets per minute in two and three row collations of 10 to 25 cigarettes per packet. This is done by wrapping cigarettes with foil, labels and stamps at high speed.
Molins HLP5 Hinge Lid Packing Machine for cigarettes	The HLP5 has an output of up to 400 packets per minute and one of the most significant advantage is its flexibility of being able to adjust size and collation changes.
Langston Saturn IV Corrugated Container Maker	The Saturn IV is an automatic high speed corrugated box maker. It offers printing, creasing, cutting, folding and gluing in one modular but integrated machine.
Delta OVSF Applicator for security printing (bank notes)	This applicator applies highly reflective Optically Variable Security Film at a speed of 8000 sheets per hour. It also offers very low wastage of expensive tamper proof material and very low (<0.06%) spoiled rejects.

1.4.1 Description of a Typical Machine - Tea Bag Machine (TBM)

There are numerous machinery in the market that can be loosely categorised as multi-axis machines but only those that are rated towards the high end in terms of performance are under consideration. The tea bag machine, as shown in Figure 1-5 and described in Table 1-2 is a typical high end design. It produces double chambers tea bags (as shown in Figure 1-6) by carrying out a series of mechanical and thermal processes in continuous motion at high speed. High level of performance is obtained by employing in house designed control system with low level customised software instead of off-the-shelf control solutions.

Figure 1-5 High Speed Tea Bag Making Machine (TBM)

Table 1-2 Brief Summary of Tea Bag Machine

size	approx. 2.8(w) x 1(d) x 2(h)m
number of axis	7 brushless 2 brush motors
number of shafts	25 functional 25 idle
number of sensors	35
production rate	1000/min
motion types	unidirectional
sampling rate	1ms
encoder	8000 pulses per rev
accuracy	30 pulses

1.4.1.1 Double Chambers Tea Bag Production Process

The double chamber tea bag is made up of the following parts

- a printed paper tag
- a piece of string which attached the tag to one of the tea chambers
- tea chambers which are made up of filter paper with polypropylene contain which becomes adhesive upon application of heat.
- measured dose of tea

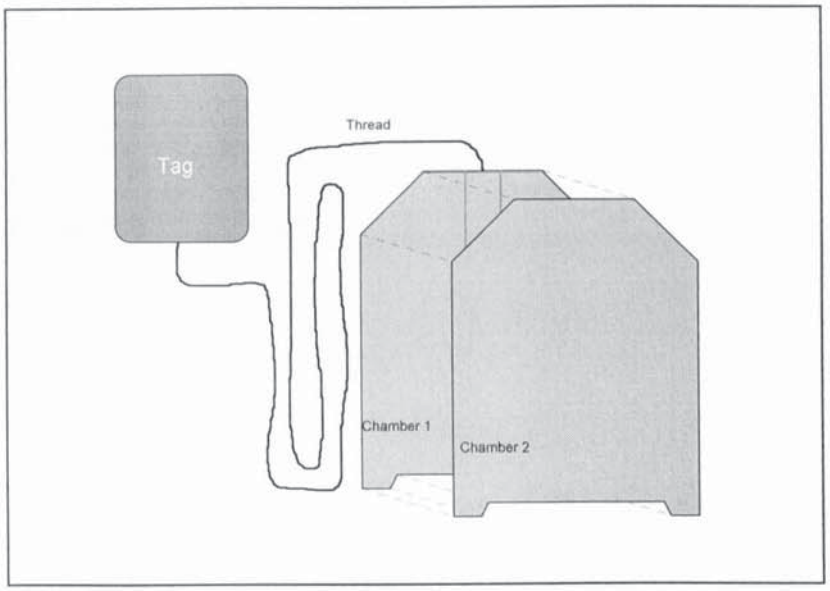

Figure 1-6 Construction of a Double Chamber Tea Bag

The process of double chamber tea bag construction is summarised in flow chart in Figure 1-7. Continuous tapes of filter paper, tag labels and thread are coming off bobbins with an automatic mechanism which changes over to spare bobbins when any of the material is coming to the end and therefore ensure a smooth production without stoppage.

The filter paper tape is first split into two down the middle of its width and go into two different laps. In the left hand (LH) lap, tags are cut out of the label tape feed and a piece of thread is attached, one end to the tag and the other end to the filter paper. The thread is folded neatly twice before pressed onto the filter paper. The filter paper is then folded into a tube and a measured dose of tea is deposited into this tube while it is forming. Heat is then applied to seal the tube and a continuos steam of chambers are formed.

In the right hand (RH) lap, a similar process is performed but without the tag and thread attachment. The two streams of tea chambers are then come together one on top the other.

They go into the tail and top joiners where the four corners are folded, pressed and fused together. Heat is applied again so that the tag adhered to the tea bag properly. This continuous stream of double chamber tea bags are then cut and the individual tea bags are collated into cardboard boxes.

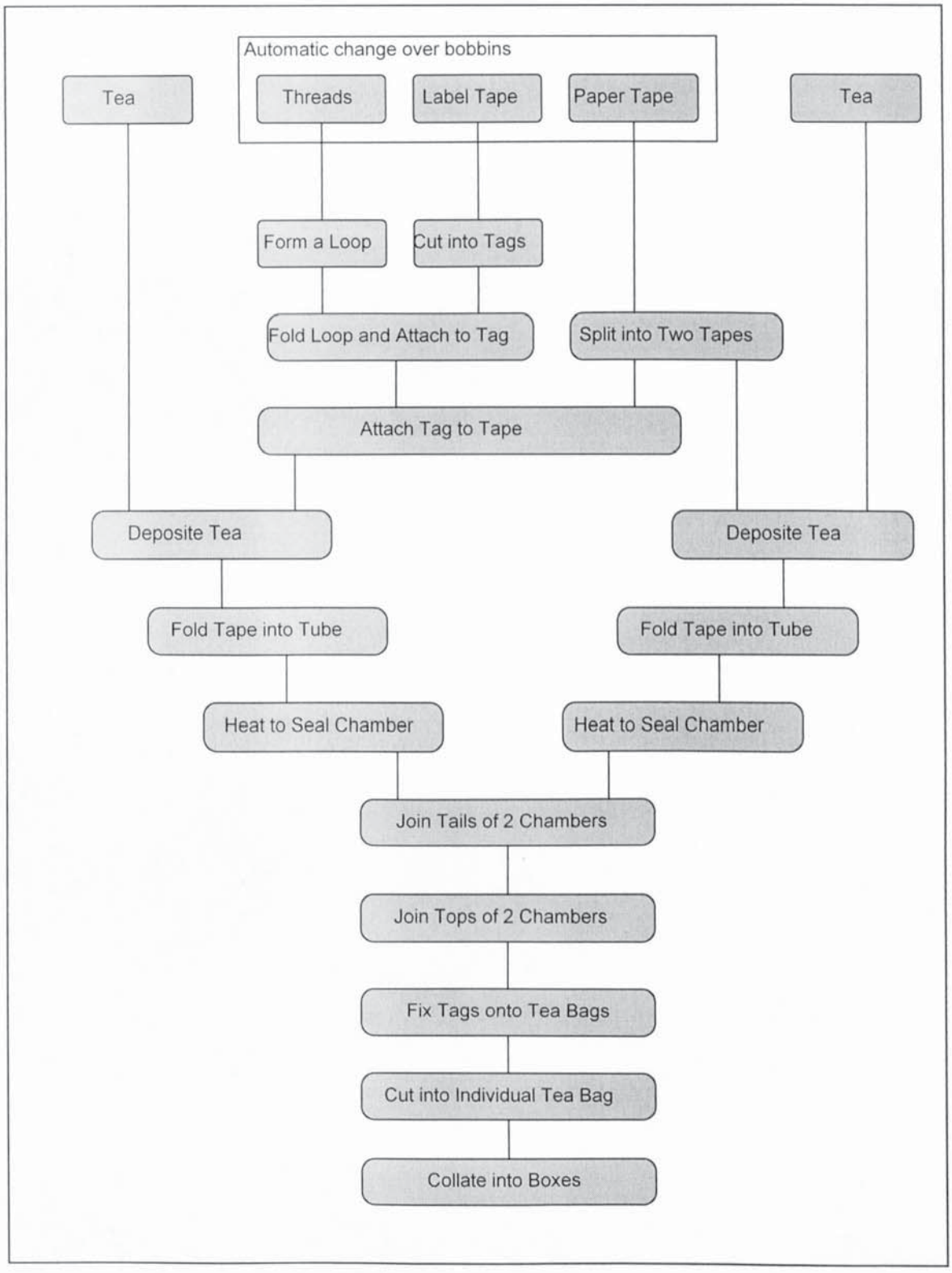

Figure 1-7 Process Chart for Double Chambers Tea Bags

1.4.1.2 Tea Bag Machine Mechanical Linkages

The schematic in Figure 1-8 shows the level of complexity involved in driving a series of functional shafts in the TBM. The inherent inflexibility is apparent. Any modification will be a time consuming undertaking. In theory, however, all shafts can be replaced by independently driven axis, depending on the cost and required accuracy.

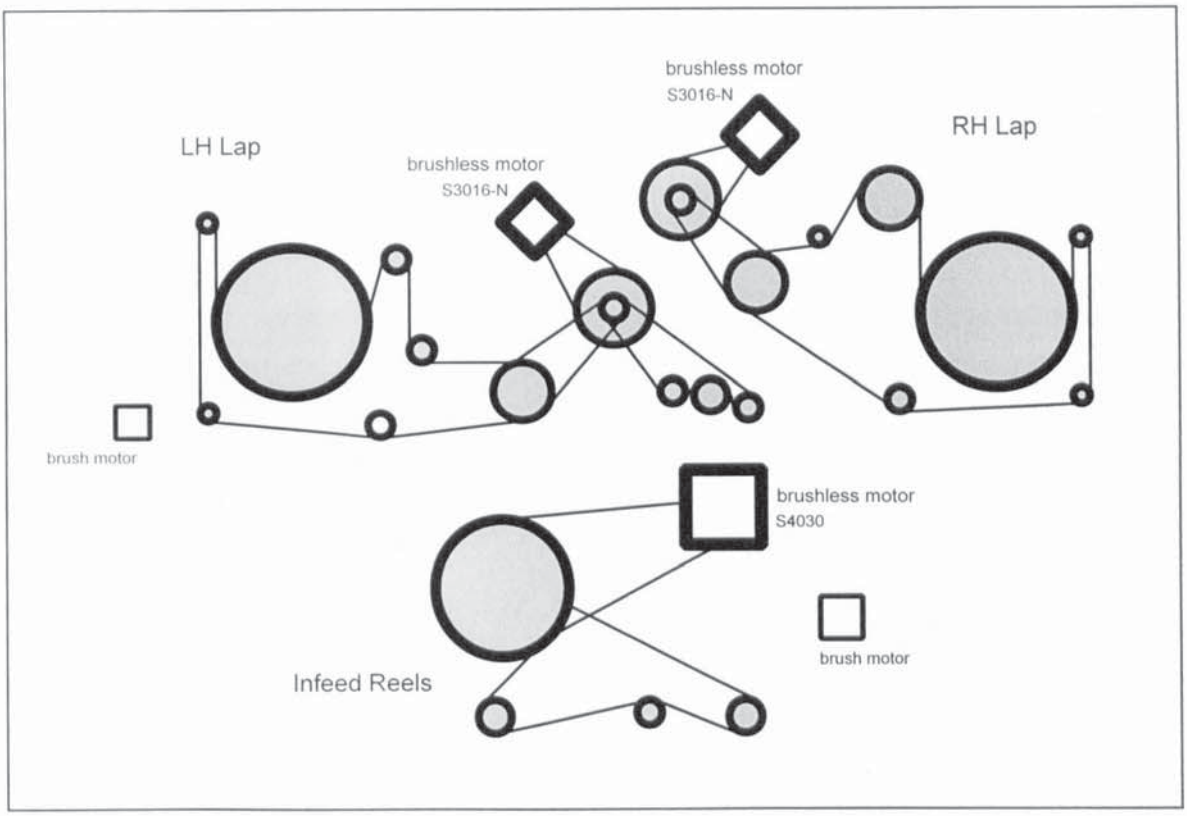

Figure 1-8 Mechanical Linkages for LH Lap, RH Lap & Infeed Reels

1.4.2 Servo Hardware for a Typical Machine

The hardware involved in the servo system of the tea bag machine is depicted in Figure 1-9. Each motion axis consists of a motor and its corresponding drive. These axes are controlled by dedicated controller cards which are mounted on a VME bus and are supervised by a host computer. For development purpose, the host computer is connected to a high performance computer, in this case a SUN workstation, where actual coding and compilation is done. The control structure of these servos are shown in the schematic in Figure 1-10. A summary of the hardware components is given in Table 1-3.

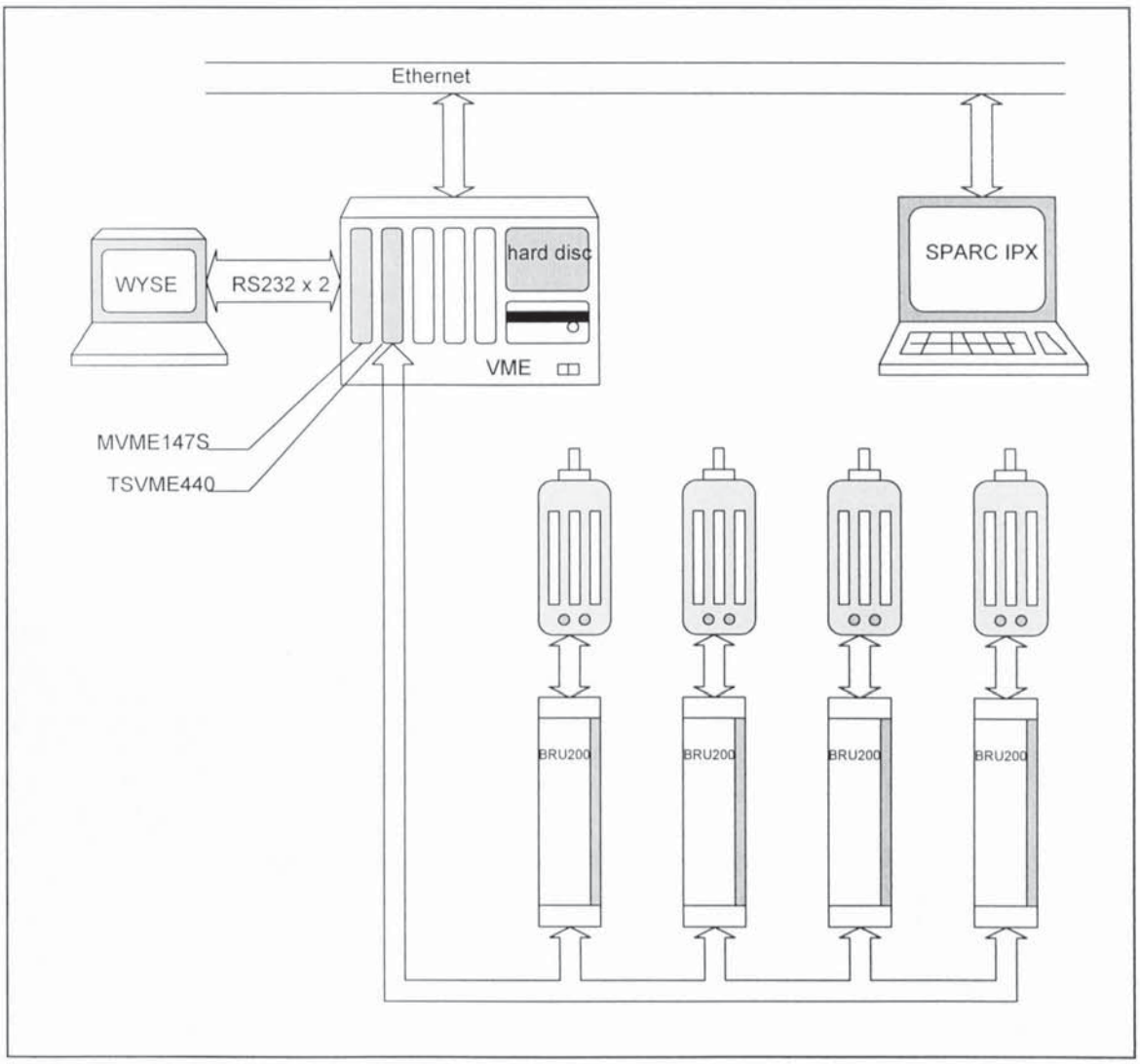

Figure 1-9 Schematic for Servo Hardware Arrangement

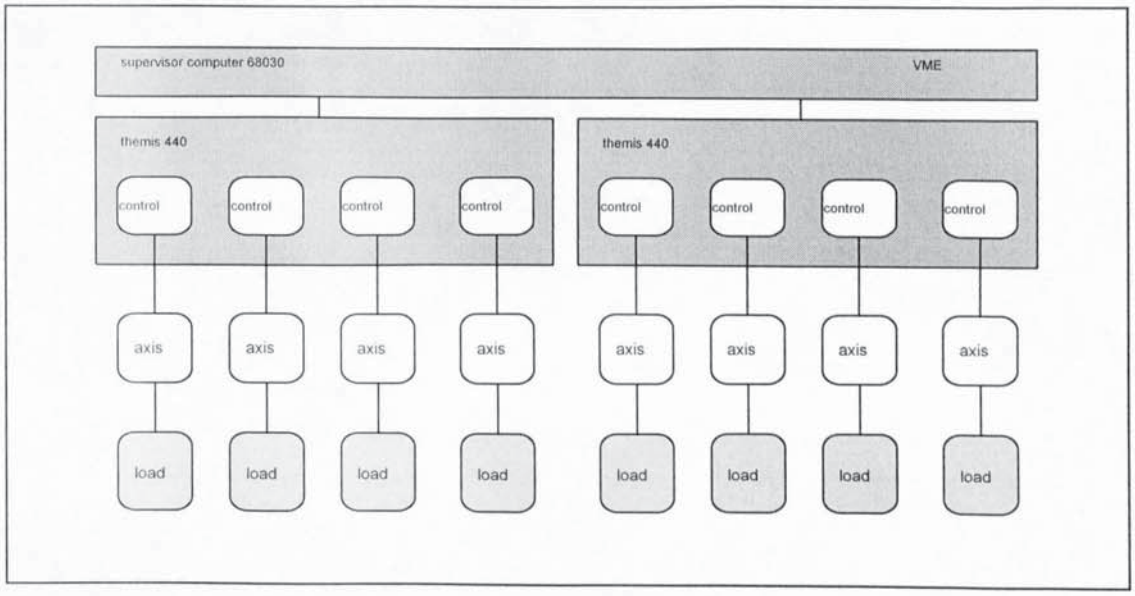

Figure 1-10 Schematic for Servo Control Architecture

Table 1-3 Servo Hardware Specifications

ElectroCraft BRU200 brushless servo system [Electrocraft 1987]	• DM-30 drive module and S-4075 brushless motor • 16-bit microprocessor control • plug-in personality module matches drive module and motor characteristics and fixed parameters for immediate start-up ready for fine tuning of the system to match the application • full tuning and diagnostics are available via an RS-232C or RS-422 serial interface • torque and velocity modes • built in PID control • current and speed limit • high frequency filter • incremental encoder with standard TTL indexed output - 2000 pulses/rev • +/- 10V analogue inputs / outputs • top speed 3000rpm
VME	• 7u 12 slots • 40 MB hard disc
Themis TSVME440 Motion Control Card	• intelligent 4-axis controller for 4 motor axes with position control • 68010 @ 12.5MHz CPU • 64 KB of 0 wait states static RAM (16 KB accessed by the VME bus) • +/- 10V analogue 13 bits command • position with 32 bits incremental encoder • hosted on VME rack
Motorola MVME147S Card	• 68030 @ 25MHz CPU 4MB • OS9 • Microware C cross compiler • Ethernet • Serial Port • WYSE terminal
Load	• coupled with torsionally stiff flexible coupling to reduce the problems of backlash and alignment

In a brushless machine the field winding is replaced by high power, rare earth permanent magnets mounted on the rotor, removing the need for a brushgear. The resulting machine offers advantages in terms of reduced maintenance, increased torque/volume ratio, increased torque at high speeds and simplified protection in comparison with more conventional machines.

Brushless machine can exhibit a ripple in their output torque. This contains two main components: the reluctance ripple caused by the inherent magnetic asymmetry of the machine; and the drive current ripple at the frequency of pn/2, where p is the number of poles and n is the speed in rpm. There is also a component at the speed of the machine which arises from the non-alignment of the rotor within the stator.

The BRU200's consist of a control unit, a power supply module and a brushless Neodymium Boron Iron permanent magnet synchronous motor. The motor has a shaft optical encoder, producing 8000 quadrature pulses per revolution, to provide feedback information. This system is considered to be one of the best servo drive systems available at the time the machine was being designed.

The BRU200 has a number of regulator/control loops within it, in order that the motor can function properly, these loops carry out necessary functions such as determining the phases of the power supply to the motor. In addition, the BRU200 is supplied with a variable parameter Proportional Integral and Derivative (PID) velocity control loop to follow velocity demands. The proportional element of the controller provides a scaled response to velocity errors, in an attempt to improve upon the simple proportional controller performance, the integral and derivative terms are included. In theory, the integral term is used to reduce steady state errors within the system but makes the system unresponsive and decreases the maximum possible system gain before instability occurs. In theory the derivative term increases the speed of response and extends the system bandwidth but in practice it also acts as a noise amplifier, particularly in a sampled data systems.

The BRU200 has two control modes, a velocity mode and a torque mode. The velocity mode uses a velocity control loop which takes an analogue velocity input and converts it to an equivalent digital velocity signal which is compared to a digital feedback velocity taken from the optical encoder on the motor shaft. The difference between the actual and demanded velocity is the error signal which was used in the PID control algorithm in an

attempt to adjust the motor velocity to its required level. In the Torque mode an analogue torque input signal is put into the BRU200. The BRU200 determines the required current magnitude which is input to an internal current regulator which compares the demanded with actual current and uses a feedback arrangement to adjust the current input to produce the required torque.

The digital nature of the BRU200's control algorithm and feedback produces a number of problems. The discrete sampling of the optical encoder pulses produces a ripple/oscillation in the feedback measurement causing fluctuation in speed to occur. The feedback oscillations can produce mechanical resonates known as "bell" resonance, in the motor causing audible noise and a fluctuation in speed to occur. The BRU200 has an internal software filter to counter this effect. The pulse width modulation unit in the power module produces considerable electrical noise which hampers the taking of analogue measurements, such as torque, from the system. When forming the design problem, the problems of noise, quantisation error, mechanical and electrical limits must be included in the design specification.

1.4.3 Control Software for a Typical Machine

Although the functionality of the class of high speed machines concerned varied greatly (see Table 1-1), electro-mechanically each can be seen as a collection of co-ordinated mechanisms driven by motors. Consequently, the control software in each case are very similar in terms of data structure and algorithm. For this reason, an adaptive generic drives software has been written to enable fast development. More specifically, this generic software is a code generator which takes standardised definition files and produces target code for both the supervising computer and the motion control cards. Due to the confidential nature of this software, no source code nor official documentation was released. However, the use of the code generator was made available to the author by industry.

The two most important definition files are described in the following tables to illustrate the working of the code generator. MIG is the machine instant generator which configures the machine through a series of definitions and SAG is the system application generator which provides a mechanism by which run time routines can be defined and linked to become commands in the application interface.

Table 1-4 Machine Instant Generator

MIG - Machine Instant Generator
• bus address for hardware • machine and axis names • sampling period • variables in data structure • PID parameter settings • etc.

Table 1-5 System Application Generator

SAG - System Application Generator
• start/stop • enable/disable • load motion profiles • diagnostics • customised run time routines

1.4.4 Typical Design Cycle for Machine Building

Aided with the engineers from industry, the typical steps in the design cycle of machine building are identified and summarised as follows:

1. Customer identifies a requirement and prepares a (draft) specification. (Mostly, supplier makes customer aware of available technology and prices and supplier calculates that a new machine is appropriate). The specification includes normal expected performance figures together with measurable quality and reliability targets.

2. Sometimes supplier discusses draft specification with customer and determines a refined specification.

3. Supplier examines how the process can be broken down into stages. Particular attention is given to where synchronism is vital and to where process speed is likely to be ultimately limited. Parallelising the process is considered for speed-critical stages. Depending on the inertia of the work-material, an assessment is made as to whether the

speed critical operations should be carried out "on-the-fly" (the tool accelerated to synchronise with the work material speed) or statically.

4. Mechanisms are designed to perform the individual stages (utilising standard modules if possible; otherwise standard design/selection procedures). Often low-speed prototyping is used on the individual stages to assess viability. Dynamic calculations verify that high-speed operation will be acceptable and stable and quantifies the requirements on the input (torques or forces and stiffness). If more than one arrangement is possible, a choice is made on the basis of a weighted merit analysis.

5. Drives are selected for the various inputs. Depending on the nature of the input, it may be appropriate to use a single electromagnetic drive (maximises flexibility) a double or triple clutch-like drive taking power solely from an input shaft (maximises power/volume and bandwidth) or a hybrid machine which takes power predominantly from an input shaft and uses an additional low power drive to superimpose a desired motion profile (maximises efficiency and power/cost). Sizes are chosen on the basis of maximum instantaneous torque (or force) requirements and average allowable heat generation.

6. The structure of the motion controller is then chosen. An initial option is to consider driving all input axes to independent motion profile demands with slow (a time constant of many cycles of individual axis) adjustment of the cycle speed to prevent slack or over-tension from building up. Almost invariably, certain axis which must be tightly coupled will not achieve the desired synchronisation consistently. Present practice is to identify groups of axis which must be tightly coupled. The drive with the longest time-constant is chosen as a master and the other drives in the group are "slaved" to it in master-follower arrangements. (Which means that the output velocity profile of the master axis becomes an input demand profile for the higher-bandwidth drives). This "slaving" can only take one form in a two-axis arrangement but with three or more axes, the slave drives can be connected in parallel or cascaded in order of decreasing time-constants.

7. Preliminary stability checks are made by taking the nearest linearised and time independent models of the individual drives and using these to carry out a check on whether major poles lie in the stable half-plan. This check is further endorsed by time-domain simulation.

8. A machine is then built and tested as a whole unit. Gains are set to the values suggested by the theoretical part of the process and when these invariably fail to deliver the performance which is known to be possible, manual adjustments are made to the settings (occasionally to the topology of the controller) to optimise or at least improve the overall performance of the machine.

1.4.5 Current Practice for Servomechanism

After some discussions with the engineers from industry, a number of issues concerning the design of servo systems are summarised as follows:

1. The prime concern is position accuracy, both absolute and relative, which has to be considered in reference to the operating frequencies of the electric servo systems which are experiencing both varying load and dynamic parameters.

2. The starting point of the design is the specification of hardware and software. Mechanical restrictions/limitations of process are based on simple prototyping and estimation (experience and knowledge of process). Structure and functionality of the core control system has to be considered (typical software architecture with task and sensors data distribution). The motor/servomechanism selection is based on power requirement and process details is altered to retro-fit the servo whenever possible.

3. Servo selection is currently carried out by engineers with a predominantly mechanical engineering background with a knowledge of mechanism design, inertia calculations, acceleration/velocity profiles and machine design, but may not have experience of the use and/or selection of servos.

4. To date Molins have predominantly used PID plus feedforward controllers with each servo following a notional software master. Occasionally master/slave relationships or additional logic within the controllers (e.g. dead-zone, print registration) are implemented. It is likely that the vast majority of the systems will be met by the above situation.

5. Up to 60% of all applications are tuned manually with simple PID control and the remains are controlled by identifying system with transfer function analyser and tune PID accordingly. Occasionally, feedforward control and/or simple alternative (e.g. the addition of constant phase) are also employed.

6. Each servo/axis is controlled independently and no provision is made to deal with interaction between servo/axis.

1.5 Link Project

With the support of the Engineering and Physical Science Research Council (EPSRC) and the Department of Trade and Industry (DTI), of the UK government, a number of LINK projects have been set-up throughout the UK so that an organised effort can be put to the research of high speed machinery. In particular, with the collaboration of Molins PLC, research into the design methodology for servo systems of independently driven multi-axis machinery has been conducted in the Machine Control and Drives Laboratory in Aston University.

Overall, the aim of the project is "to develop a general purpose methodology to aid Molins designers to select servos for their machinery and to aid the optimisation of the corresponding controllers." Specifically, the objective is to formalise and automate the process of design and synthesis of control strategy for the servo system of multi-axis high speed machinery. At the first instant, the deliverable of the project will be a design methodology in the form of a document or software that serves as guidelines. For this reason, the ease of use will be of foremost importance for any proposed solutions.

2. Background

2.1 Introduction

The proposed design methodology is aimed at formalising and automating the process of design and synthesis of the motion control strategy of multi-axis high speed machinery. The target is a decentralised and distributed control system. Special emphasis is placed on the control topology (the way axes are grouped into subsystems) and therefore the interaction between subsystems will be the main focus.

The design process takes into consideration the electro-mechanical properties (intrinsic constraints) and the target behaviour (specification and hence imposed constraints) of the machine and suggests the best topology subject to the design criteria set out. A decentralised control scheme with local multivariable controllers for each subsystem is employed. The overall control system is then evaluated by simulation.

A survey is conducted on the relevant technical areas which includes issues such as system decomposition and decentralised control techniques. Although the system decomposition - decentralised control process should be treated as an integrated whole, the control engineering literature reveals that they are usually considered and discussed as separate issues (see section 2.2.4). As ideas are being drawn from different branches of the literature, attempts are made to clarify the various relevant concepts.

2.2 Methodology

A methodology is an integrated and organised set of guide-lines based on some working principles together with the use of carefully established procedures. It should provide a framework in which a problem can be handled and tackled. The required methodology is aimed at assisting a designer to obtain a viable control strategy. The resultant control system should guarantee the desired system behaviour under the design constraints. Some relevant materials are discussed as follows.

2.2.1 The Mechatronics Principle

This is the integration of electronics and computing technologies into a wide range of primarily mechanical products and manufacturing processes. The performance and their

manufacture depends on the capacity of industry to exploit development of technologies and to introduce them in the design stage into both products and manufacturing processes. The result is systems which are cheaper, simpler, more reliable and with greater flexibility of operation. The old practice of separate mechanical and electronic engineering is increasingly being replaced by the integrated and interdisciplinary approach to engineering design referred to as mechatronics.

The best realisation of a product usually depends on a consideration of the necessary electronics, control engineering and computing from the earliest stages of the design process. Mechatronics is not a distinctly defined discipline but an integrating theme within the design process. The foundations of a mechatronic approach to engineering design are considered to lie in information and control. The feature of a mechatronic approach is that the resulting mechanical systems are often simpler, involving fewer components and moving parts than their wholly mechanical counterparts. This simplification is achieved by the transfer of complex functions such as accurate positioning from the mechanical system to the electronics.

To be successful, a mechatronic approach needs to be established from the earliest stages of conceptual design process, where options can be kept open before the form of embodiment is determined. In this way, the engineer can avoid going too soon down familiar and perhaps less productive paths.

2.2.1.1 Mechatronics & Manufacturing

In the manufacturing industries there is a demand for production systems which are capable of responding rapidly to changing market conditions, accommodating a range of product types with short production runs involving relatively small number of items. Neither manual manufacturing processes nor mass production lines can meet these requirements. The former - though highly adaptable - suffers from low level of productivity. The assembly and transfer lines associated with the latter lack flexibility, with changeovers involving significant time costs.

Within a wide range of manufacturing systems and processes, a mechatronic approach has had as its primary benefit, the ease with which the processes can be reconfigured, while at the same time offering enhanced product quality and consistency.

The process industries have seen the working together of mechanical, electrical and electronic engineering for many years, and the 'wet' process industry in particular has exhibited a significant degree of integration in its approach to system design. Indeed, a chemical plant may be regarded as a mechatronic system in its own right, in which the introduction of microprocessor and associated communications is having a significant impact though the opportunities offered for distributed control incorporating local decision making capability. The introduction of decentralised systems has also brought about significant changes in procedures in areas of plant optimisation, diagnostic based maintenance and data handling.

2.2.1.2 Mechatronics and Technology

Mechatronics is concerned with the bringing together and integration of certain key technologies, particularly:

- sensors and instrumentation systems
- embedded microprocessor systems
- drives and actuators (Motion Control)
- engineering design

Figure 2-1 shows the conceptual/technological components in typical mechatronics.

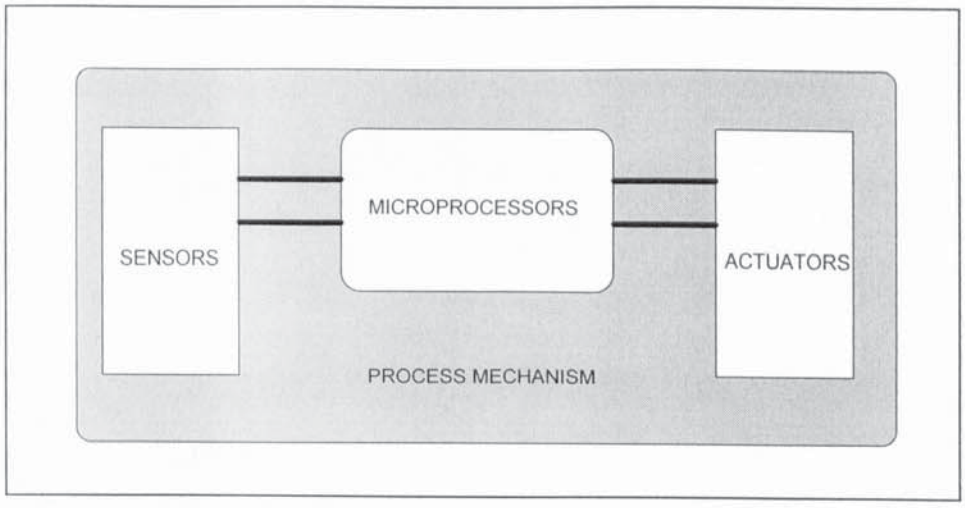

Figure 2-1 The Basic Mechatronic Approach

2.2.1.3 Machine

A machine is defined as an assembly of linked parts or components, at least one of which moves. It includes the appropriate actuators, control and power circuits joined together for a specific application, in particular for processing, treatment, moving or packaging of a material.

2.2.2 Large Scale Systems

Large scale systems are characterised by their high dimensionality and complexity. In dealing with such systems, one naturally looks for the inherent structure. The next logical step is to approximate by imposing a structure to the system. If possible, the system is partitioned into smaller subsystems such that both the dimensionality and complexity are reduced.

A detailed account of the modelling methods and model reduction of large scale systems in both time domain and frequency domain is given by [Jamshidi 83] and a useful survey on the decentralised control of large scale systems is given in [Sandell, Varaiya, Athans & Safonov 78]. In general, these methods require a complete system description as a starting point.

2.2.3 Interconnected Dynamic Systems

Once the inherent structure of a complex system is identified, it can be decomposed accordingly into a number of interconnected subsystems. The decomposition process is non-trivial and will be discussed further in section 2.3.

If restrictions are imposed on the flow of information between interconnected subsystems, subsystems can act on the full local information and on the partial (or none of the) information from other subsystems. Such arrangement demonstrates a decentralised information pattern. As only partial external information is required by the local subsystem (or none at all) and computation (involving a relatively small subsystem) can be done locally, decentralised control schemes can be devised to take advantage of the lesser communication and computation requirement.

To investigate the characteristics of the overall system, a set of analytical tools similar to the classic control theory may be useful. This is discussed further in section 2.4.

2.2.1.3 Machine

A machine is defined as an assembly of linked parts or components, at least one of which moves. It includes the appropriate actuators, control and power circuits joined together for a specific application, in particular for processing, treatment, moving or packaging of a material.

2.2.2 Large Scale Systems

Large scale systems are characterised by their high dimensionality and complexity. In dealing with such systems, one naturally looks for the inherent structure. The next logical step is to approximate by imposing a structure to the system. If possible, the system is partitioned into smaller subsystems such that both the dimensionality and complexity are reduced.

A detailed account of the modelling methods and model reduction of large scale systems in both time domain and frequency domain is given by [Jamshidi 83] and a useful survey on the decentralised control of large scale systems is given in [Sandell, Varaiya, Athans & Safonov 78]. In general, these methods require a complete system description as a starting point.

2.2.3 Interconnected Dynamic Systems

Once the inherent structure of a complex system is identified, it can be decomposed accordingly into a number of interconnected subsystems. The decomposition process is non-trivial and will be discussed further in section 2.3.

If restrictions are imposed on the flow of information between interconnected subsystems, subsystems can act on the full local information and on the partial (or none of the) information from other subsystems. Such arrangement demonstrates a decentralised information pattern. As only partial external information is required by the local subsystem (or none at all) and computation (involving a relatively small subsystem) can be done locally, decentralised control schemes can be devised to take advantage of the lesser communication and computation requirement.

To investigate the characteristics of the overall system, a set of analytical tools similar to the classic control theory may be useful. This is discussed further in section 2.4.

2.2.4 Decomposition vs. Decentralisation

In [Sandell 76], an interesting discussion is given on the potential benefits of decentralisation and decomposition. It can be summarised briefly as the following. Decentralisation is concerned with reduction of the on-line communication and computation requirements implicit in a mathematical defined control law whereas decomposition is concerned with the off-line computation required to obtain the given control law.

2.2.5 Control Strategy

After a suitable control topology has been decided, a decentralised control scheme can be devised. However, such a scheme can only be properly evaluated when the individual subsystems, which are multivariable in general, are under control. Therefore multivariable controllers have to be considered at an early stage and the full control solution should consist of both the control structure and the control laws. The design of a multivariable controller is an involved subject and a discussion is given in section 2.5.

If one takes the view that each process of modelling, system decomposition and decentralisation represents some form of approximation, one can see that the overall design process should be integrated vertically as far as possible to minimise the degree of approximation.

2.2.6 Distributed Computer Control System

Without loss of generality, the control strategy provided by the design methodology can utilise the control hardware architecture that is currently used in the typical high speed multi-axis machines. The generic control hardware consists of:

- axis - motor, sensor, encoder, amplifier, drive
- controller card - on board computer dedicated to real time control
- host computer - supervisory

In a typical application, the individual axes are controlled independently, as shown in Figure 2-2, and it is suggested, in section 3.2, that improvement can be made if interaction between the axes are taken into account by the controllers, as shown in Figure 2-3. Such an arrangement should harness the full potential of the distributed computer control

system. Note that an effective multivariable control strategy can only be achieved if detailed knowledge of the full interaction within the plant is available.

In [Nader 79], an outline of the design process of distributed computer control systems is given and the issues of decentralised control and distributed processing (concurrent programming) are emphasised.

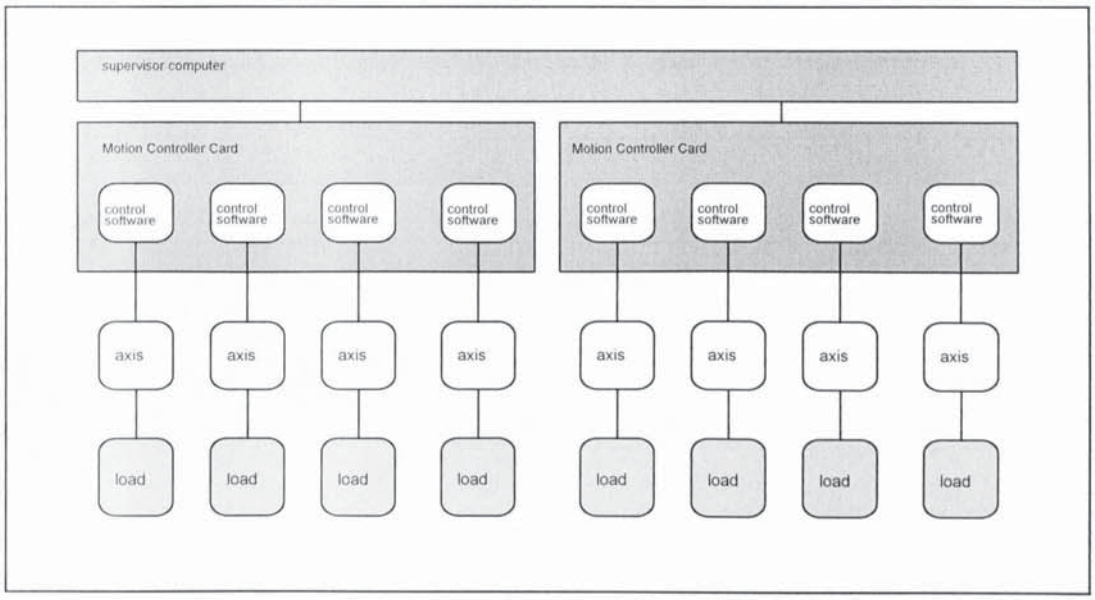

Figure 2-2 Distributed Computer Control System - Independent Axes and Load

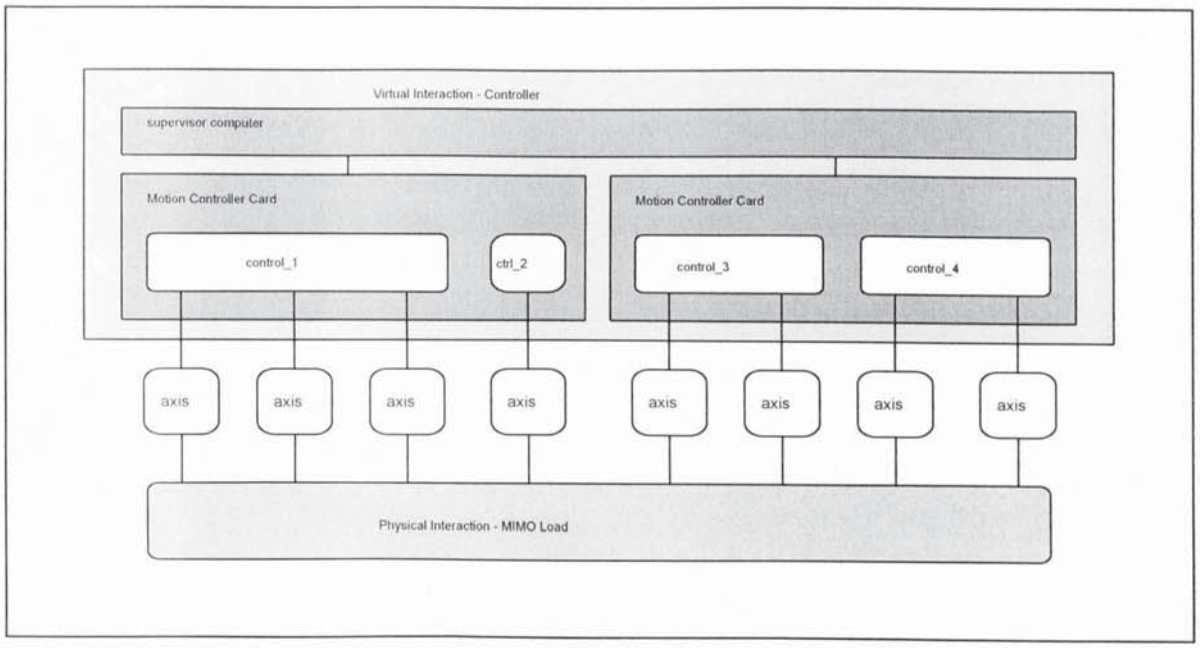

Figure 2-3 Distributed Computer Control System - Multivariable Subsystems

2.2.7 Computer Aided Control System Design

A natural development to the automation of a design process is the use of computer. Computers are particularly useful in dealing with complexities whereas humans are better at dealing with uncertainties. A computer environment can be created so that the design process can be carried out in an interactive manner. A computer aided design package typically contains three levels of abstraction (Figure 2-4).

- *The conceptual level* : consists of guide-lines which assist the designer in making trade-off of resources and performance against various requirements.
- *The synthesis level* : consists of procedures which involve symbolic manipulation for achieving the appropriate control strategy (control structure and control law).
- *The computation level* : consists of routines for the evaluation of the various numerical results.

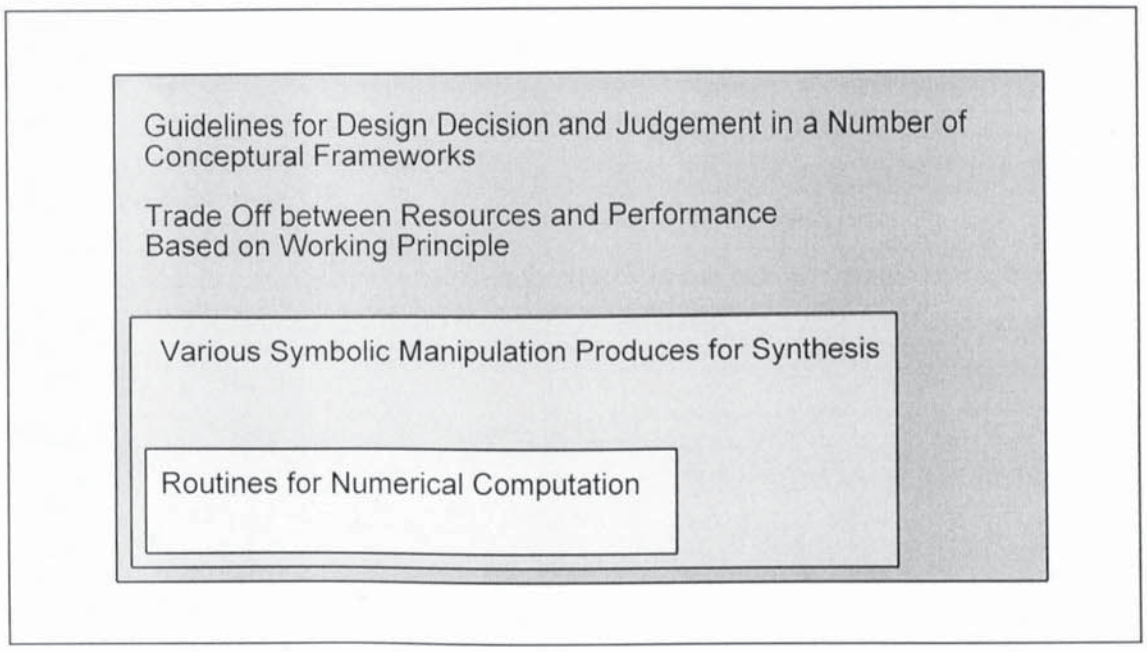

Figure 2-4 Computer Aided Design Methodology - 3 Levels of Abstraction

The aim of a computer aided control system design methodology is to guide a designer along the best possible path, most likely to be iterative, through the various relevant procedures. The designer will have to make a number of trade-off decisions whereas the repetitive computation and simulation will be done by the computer.

In [MacFarlane, Gruebel & Ackermann 89], some ideas of the design environment for control system are discussed. In [Lieslehto, Tanttu & Koivo 93], an expert system

approach to the design of multivariable controller is described. The most interesting aspect is the use of a hypertext package as an interface to the knowledge base. Some of the concepts in the multi-objectives optimisation approach [Ng 89, 93] in computer aided control system design may be relevant to the problem in hand.

2.2.8 Simulations and Design Parameters

Effective simulations can be carried out only if a detailed model of the plant is available. For the application in hand, it is assumed that the following is available.

- characteristics of motors [Pillay & Krishnan 91], drives, amplifiers, encoders etc.
- characteristics of load with full description of the interaction between axes.
- characteristics and capacities of the communication and computation hardware.
- specification in terms of tolerance of absolute/relative errors and the required degree of imposed coupling and decoupling.

In other words, all intrinsic electro-mechanical characteristics are known or can be estimated and an appropriate model can be built. One of the most common form of interaction is the material web around several axes and an in-depth discussion is given in [Young & Reid 93]. Another form of interaction is the inertial coupling through the structure of the machine and general discussion can be found on standard texts on mechanical vibration.

2.2.9 Implementation Issues

A design process will not be completed until the control system solution is implemented and evaluated. However, the implementation of such a solution is not trivial as there are many issues involved in moving from theory to practice. [Anderson 93] points out the problems of order reduction, discretisation and digital realisation of continuous-time controllers.

2.3 Control Structure Selection

In the context of the proposed design methodology, decentralisation is the employment of a block diagonal control structure on the full system transfer function matrix. The

selection of this block structure is essentially a top-down process starting with a system description that takes into account of all input/output interactions. The extent of decentralisation can vary depending on the design constraints. A full decentralisation turns the system into a set of parallel SISO (Single Input Single Output) subsystems whereas a partial decentralisation turns the system into a set of smaller MIMO (Multi Input Multi Output) subsystems.

To a large extent, the control structure selection process is a screening process. To illustrate the size of the problem, a brief study of the number of possible control structure is given in section 3.4. It is shown that by imposing block diagonal constraints, the dimension of the problem can be reduced to a more manageable size.

It is worth noting that whatever decomposition criteria are taken in the selection process, proper results can only be obtained by considering the system in closed-loop with the controllers in place. For this reason, the selection of control structure and the synthesis of decentralised (multivariable) controllers should be integrated somehow. However, it appears that most of control engineering literature discusses them as separate issues.

In the following discussion, it should also be borne in mind that the decomposition processes described take only into account of the inherent structure of the plant dynamics whereas in the context of the design methodology, provision has to be made to accommodate the imposed structure (coupling and decoupling) demanded by the designer. This represents a degree of bottom-up design freedom in the methodology.

2.3.1 Graph-Theoretic Approach

A comprehensive treatment of the decentralised control of complex systems with a graph-theoretic approach is given by [Siljak 91] and some of its ideas are summarised in sections 2.3.1.1 to 2.3.1.4. The main focus of the approach is the properties associated with the generic structure of the system.

2.3.1.1 Structured Systems

Analysis of dynamic systems is carried out in the rigorous framework of graph theory. A graph-theoretic approach is set up for the existence of suitable structures for decentralised control. The end product is a collection of efficient and reliable algorithms for testing candidate schemes for decentralised control and estimation.

The main problem is to determine if there is a path to every state from at least one input and that is if the system is input reachable. The dual concept of output reachablility is defined likewise. Such concepts are used in the decomposition of large dynamic systems. Together with the concept of generic rank, the notion of structural controllability is introduced. To study the existence problem of control laws under structure constraints, structurally fixed modes are considered.

2.3.1.2 Hierarchical LBT Decompositions

In many cases, systems are decomposed into subsystems based on physical insight, intuition or experience. For system with a large number of states and variables, a systematic method for decomposition is desired. A graph-theoretic algorithm is developed to partition a linear dynamic system into a Lower Block Triangular (LBT) form. The utility of LBT decompositions by a sequential LQG optimisation of hierarchically structured systems is demonstrated. The information structure for distributed control of LBT forms are discussed and the interrelationship between the physical constraints and the computation capabilities of the system is explored.

2.3.1.3 Nested Epsilon Decompositions

It has long been recognised that most variables in a complex system are weakly coupled, if coupled at all and it has been argued that the overall system behaviour is dominated by the strongly coupled variables.

A decomposition-clustering process is discussed. The underlying principle is to associate a graph with a given system, disconnect the edges corresponding to the interconnections with strength smaller than a prescribed threshold epsilon, and identify the disconnected subgraph (components) of the resulting graph. The obtained components correspond to the subsystems with mutual coupling smaller than or equal to ε. Such an algorithm has the advantage of requiring only binary computations.

Note that the epsilon decompositions produce nested hierarchies which, unlike the "vertical" LBT hierarchies, required testing for stability. A construction of a hierarchical Liapunov Functions is presented [Siljak 91] which is well suited to establishing nested connective stability for nested structures.

2.3.1.4 Overlapping Decompositions

It is advantageous to recognise the fact that many natural and man-made systems share common parts and decentralised control and estimation schemes should be built using overlapping information sets. It has been known for some time that overlapping subsystems can be treated by standard decentralised control methods by expanding the state space. The expanded laws are then contracted for implementation for the original system. Such expansion-contraction process has been organised into the general mathematical framework of the Inclusion Principle. These results are of central importance for building reliable multiple controller schemes.

2.3.2 Block Diagonal Dominance

The subject of the control of large scale systems is dominated by techniques for model simplification and order reduction and a natural trend is the philosophy of decentralisation. A natural partitioning of a system into weakly interacting subsystems can be obtained by matrix transfer function (frequency domain) considerations.

In [Bennett & Baras 80], a decentralised compensator design similar to the Inverse Nyquist Array method is developed. A condition to ensure the desired performance of the overall system is proposed. Better flexibility is provided in this design because the performance requirement of each subsystem may be of different nature and it is possible to employer different design approaches for each of them. In addition, the reduction of the dimension of the overall problem leads to significant reduction in the computational complexity of the design procedure.

2.3.2.1 Generalised Gershgorin Theorem

In [Feingold and Varga 62], the Gershgorin Theorem is generalised to partitioned matrices. By allowing different norms for different subspaces, a variety of alternative estimates of the eigenvalues are available. Since the inclusion regions depend on the partitioning of the matrix and the vector norms that are used on the subspaces, it is suggested that tighter estimates can be obtained by different combinations of norms.

2.3.2.2 Disjoint and Overlapping Decomposition

A generalisation of the Nyquist array method to blockwise decomposition is presented in [Ohta, Siljak & Matsumoto 86]. In the purposed method, the system matrices are allowed

to be partitioned into disjoint as well as overlapping sub-matrices. Controllers for individual subsystems can be designed independently of each other. The first objective of the paper is to show a generalisation of the Nyquist array method to the design of disjoint decentralised controllers. The second objective is to introduce an input-output inclusion framework for transfer function matrices. It is also shown that overlapping decompositions are superior to standard disjoint decompositions when decentralised control strategies are considered.

The Nyquist and Inverse Nyquist array methods became two of the most effective methods for designing MIMO controllers centrally because of the conceptual and numerical simplicity of the procedures. It is however difficult to apply them to MIMO subsystems. The notion of block diagonal dominance and the re-formulation of the Nyquist methods to blockwise decomposition is therefore necessary so that decentralised information structure constraints can be taken into account.

2.3.3 Comments

The graph-theoretic approach provides a framework for the natural decomposition of complex systems according to their inherent structures. At this stage, some method is needed to superimpose the external structure (imposed coupling and decoupling) required by the designer. Once this is achieved, either the BRG or the Block Gershgorin bands can be employed as interaction measures to assess how isolated the subsystems can be under a decentralised feedback scheme. This is matched against the design criteria for the viability of that particular decomposition.

2.4 System Analysis

Decentralised control is a multivariable control problem with structural constraints. In order to investigate and understand the system better and to quantify the system properly, a number of system analysis techniques are developed in the literature. Some of these approaches are summarised in the following sections. The concept of block relative gain (BRG) array as an interaction measure is found to be particularly useful and is discussed further in section 2.7.

2.4.1 Decentralisation, Structure Constraints and Fixed Modes

One particular approach which put its emphasis on structural constraints is given by [Trave, Titli & Tarras 89]. It explores a number of concepts including structural controllability/observability and fixed modes and they are introduced in the sections 2.4.1.1 to 2.4.1.3.

2.4.1.1 Structural Controllability and Observability

Large scale systems are often characterised by structurally constrained feedback patterns. [Trave, Titli & Tarras 89] presents an overview of the well known results concerning stabilisation and pole assignment of time-invariant dynamic systems subjected to centralised control (no structural constraints). It starts with the introduction of the concepts of structural controllability and observability, which are independent of parameters' values. By following a graph-theoretic approach, problems are reduced to those of binary nature. The necessary and sufficient conditions for the existence of solutions to the stabilisation and pole assignment problems are then stated for both centralised state and output feedbacks in terms of the controllability and observability properties of the system.

The study of the structural properties can be achieved by either an algebraic or a graph-theoretic approach. Nevertheless the easiest way to check controllability and observability is to use a combination of the two approaches. That is to examine reachability by graph association and generic rank determination. By using an appropriate decomposition, the system global conditions can be expressed in terms of several conditions referring to smaller dimensioned systems and the checking can be carried out in a sequential way.

2.4.1.2 Structurally Constrained Control

Stabilisation and pole assignment of large scale systems are characterised by their high dimensionality and when the assumption of centralised information pattern does not hold, constraints lead to economical and reliability problems related to information transfer. Whenever possible, control system should make use of local controllers which require only part of the whole information. When no transfer of information is allowed between local controllers, a decentralised control scheme can be employed. When some but not all information transfers are allowed, a non-standard information pattern presents itself.

The advantages for decentralised control schemes are clear and the study of the stabilisation and pole assignment problems of such schemes are given in [Trave, Titli &

Tarras 89], which also extends the results to the more general case of arbitrary structurally-constrained control. Specified restricted information pattern constraints the feedback structure of a control system. The conditions for the existence of solution to the stabilisation and pole assignment problems are also given. These results are expressed in terms of fixed modes and are related to the concept of controllability under decentralised information structure.

In addition, the general framework of arbitrary structural constraints on control is considered together with the concept of fixed modes. The importance of fixed modes is pointed out as the existence of unstable fixed modes indicates that stability is impossible while the presence of any fixed modes rules out arbitrary pole assignment. Fixed modes appear as the generalisation of uncontrollable and unobservable modes in the non-constrained control.

2.4.1.3 Fixed Modes

To obtain some insight into the behaviour of the fixed modes and an understanding of their occurrence, it is helpful to refer to [Trave, Titli & Tarras 89] where a study of their characterisation is given. In addition to the characterisation of the fixed modes with a time-domain and a frequency-domain system representation, it also points out the condition for their existence. It appears that the fixed modes have either a structural or a parametric origin. Furthermore, it also provides an overview of the graphical characterisation of fixed modes. Graph-theory provides an effective tool to explore the various structural properties.

The characterisation of fixed modes can be classified into four groups:

[1] Characterisation in terms of transmission zeros

[2] Characterisation in the time-domain

[3] Characterisation in the frequency-domain. This approach shows that fixed modes arise from origins related to the interconnection pattern and can be classified into structurally fixed modes (depend on structure) and non-structurally fixed modes (depend on parameters).

[4] Graph-theoretic characterisation. Graph theory is used to study the structural properties such as the fixed modes but quantitative data can also be dealt with if graphs are weighted.

In [Patel & Misra 91], a numerically reliable algorithm for the computation of decentralised fixed modes is given.

2.4.2 Stabilizability and Connective Stability

In [Siljak 91], a comprehensive treatment of issues concerning decentralised control of complex systems is given and this treatment is summarised in this section. In the stabilisation of complex systems, decentralised schemes arise due to the inherent non-classical constraints on information structure as well as the imperfect knowledge of the model of interconnections among the subsystems.

It has been known for some time that decentralised control structures guarantee robust stability and tolerates a wide range of non-linear uncertainties in the interconnections among the subsystems. That is, closed-loop interconnected systems that are stabilised by local feedback laws are connective stable. A natural way to confirm connective stability of decentralised controlled complex systems is to apply the vector Liapunov functions.

Together with the non-existence of decentralised fixed modes, necessary but not sufficient stability test can be applied to a decentralised control structure. By describing certain classes of linear interconnected systems that are decentrally stabilisable by state feedback, it is possible to identify classes of decentralised systems that can always be stabilised by local feedback.

2.4.3 Geometric Theory of Linear Multivariable Systems

A comprehensive treatment of the geometric theory of linear control system is given by the authoritative text [Wonham 79] and a tutorial is given by [Commault & Dion 92]. The geometric theory of linear multivariable system is significant in solving control problems such as disturbance rejection, decoupling and model following. Necessary and sufficient conditions can be expressed in terms of relationships between specific state space subspaces.

This approach places its emphasis on the following :

- the trajectory interpretation of subspace invariance;

- the discrete-time interpretation of the basic subspaces and their associated constructive algorithms;
- disturbance decoupling problem (DDP)

2.4.4 Algebraic Theory of Full and Decentralised Feedback Compensators

A unified algebraic approach [Gundes & Desoer 90] to the study of linear time-invariant MIMO systems with full feedback and decentralised feedback is given. This approach applies to both continuous and discrete time lumped-parametric system and much of it can be extended to distributed parameter systems. It also utilises a factorisation approach based on ring theory and presents the conceptual tools and key results on the use of right, left and bicoprime factorisations. The main issues considered include :

- closed-loop stability
- parametrisation of all stabilising compensators
- parametrisation of all achievable stable closed-loop I/O maps

and the following feedback configurations are considered :

- unity-feedback system, standard MIMO
- general interconnected systems, MIMO with I/O not in feedback loops
- decentralised control systems, block diagonal structure
- two-channels and multi-channels

2.5 Design of Multivariable Controllers

2.5.1 Multivariable Feedback Design

A comprehensive text on the design of multivariable controllers is given by [Maciejowski 89]. In addition, a good discussion on the feedback properties of SISO and MIMO system is given by [Doyle & Stein 81] and a survey on the limitation of static output feedbacks is given by [Syrmos et al. 97]. It is identified that the method of sequential loop closing has the following drawbacks:

- very limited class of controller
- must proceed in a very ad hoc manner

- the only means for interaction reduction is high loop gain
- can be applied only if the loop assignment problem is solved

A number of standard MIMO design methods and some control issues are discussed in the following sections:

2.5.2 Sequential Loop Closing

This is the simplest approach to multivariable design, the method based on the idea of reducing the MIMO problem into a sequence of scalar problems. A SISO controller (usually PID), is successively designed for each pair of input and output variables. The procedure is often adopted and is appropriate when there is to be no interactions between loops. An advantage of the method is that it can be implemented by closing one loop at a time, ensuring that the system remains stable at each step.

There are a number of disadvantages namely, it only allows a very limited class of controllers to be designed, and the design process is very unstructured. The effects on the behaviour of the remaining loops after closing the first one or two loops may be to degrade performance. The only way to reduce interaction is by having high loop gains.

In terms of standard scalar design problem, each time a loop is closed, a set of poles are moved to new positions. However, the poles associated with the scalar systems that are moved in the context of sequential loop closing do not in general correspond to poles of the MIMO system and it is therefore clear that this method does not offer enough control of system behaviour to the designer.

2.5.3 Characteristic-Locus Method

This method is based on the manipulation of open loop characteristic (eigenvalues) loci as if they were ordinary Nyquist loci. The design usually involves computing separate compensators for high, medium and low frequencies respectively: these are then connected in series. Each locus is compensated until the loci exhibit sufficient stability margin, suitable 0dB cross-over frequency and satisfactory gain behaviour.

It is necessary to check the complete design by some other method (i.e. closed loop principal gains and Nyquist arrays) as the characteristic loci can be unreliable indicators of stability margins and performance. Large discrepancies between characteristic loci and

principal gains indicate untrustworthy characteristic loci. Indeed, the eigenvalues λ and singular values σ are related by the bounding relationship

$$\underline{\sigma}[A] \leq |\lambda[A]| \leq \bar{\sigma}[A] \tag{2-1}$$

(for rational matrix A) and the feedback properties deduced from the characteristic loci represent only an upper bound to the true feedback properties of the resulting MIMO system. Therefore, this method is reliable only if the system concerned has a tight singular value/eigenvalue bound over the frequency range in question.

2.5.4 Nyquist Array Methods

There are two types of Nyquist method; the direct and the inverse methods. Both of the methods are based on the idea of reducing the multivariable problem to a sequence of scalar problems. This is done by constructing a set of scalar transfer functions to which the classical Nyquist method can be applied individually. Pre- and post-compensators are designed to make the system sufficiently close to a diagonal system (i.e. diagonal dominant) and the diagonal elements of the loop transfer functions are taken to be scalar functions. In fact, the diagonal elements approach the eigenvalues as the system becomes more diagonal dominant and the Nyquist methods aim at shaping feedback properties by manipulating the (near) eigenvalues rather than the principal gains of the true MIMO system. The limitation is therefore clear.

2.5.4.1 Direct Nyquist Array Method

Once a satisfactory degree of dominance has been achieved, the design of compensator for each loop is carried out using the diagonal elements of the transfer function matrix on which are superimposed Gershgorin (eigenvalues) bands. The designs can use either Nichol or Nyquist diagrams. It is important in this method to try and obtain as large a degree of dominance as possible so as to minimise the effect of closing the other loops. The advantages of this method is that the designed compensator does not have to be inverted, the plant does not have to be square, since it need not have an inverse and it is easier to keep control over the compensator pole locations. Overall, this can be seen as an improved sequential loop closing method but not yet a truly simultaneous synthesis.

2.5.4.2 *Inverse Nyquist Array*

This method is used because it allows a more accurate prediction of closed loop behaviour from open-loop characteristics. Ostrowski circles, (which are gain dependent) form narrower bands than the Gershgorin bands and they allow the behaviour of individual loops to be predicted, taking into account the effects of interactions with other loops. There is no such relationship available for the direct method. Ostrowski bands can also be used to indicate how the stability changes when the gain in that loop changes. In particular, it can be used to indicate whether the system would fail if one loop failed while all other loops remained at their design gains. However, the Gershgorin bands can be used to predict stability when the gains in all the loops change simultaneously, whereas the Ostrowski bands cannot.

2.5.5 Reversed Frame Normalisation

This is a synthesis method which attempts to obtain a normal return ratio. The characteristic gains and the principal gains then coincide, and the characteristic loci give reliable indications of stability margins. The synthesis is achieved approximately by optimisation.

In the reversed-frame method, the open loop characteristic loci are specified to provide the required stability margins and performance. A compensator is then designed by an optimisation algorithm to approximate the required open loop system. The compensator structure where its singular vectors approximately match those of the system being compensated but in the reversed order.

2.5.6 Quantitative Feedback Theory

QFT assumes that the plant uncertainty is represented by a set of templates on the complex plane, each of which encloses within it all the possible frequency responses from the inputs to the outputs at a particular frequency and that the design specification is in the form of bounds on the magnitudes of the frequency-response matrices. The QFT method leads to a design which satisfies these specifications for all permissible plant variations, while minimising the transmission of output sensor noise. A feature of the method is that only the diagonal elements are designed and all the off-diagonal elements are set to zero. As with the sequential loop closing approach, success in designing a suitable compensator may depend on finding the best pairing of plant inputs and outputs.

2.5.7 LQG Optimisation

A brief account of the development of optimal control is given by [Bryson 96] and a review paper on the optimal control of linear multivariable systems is given by [Johnson & Grimble 87]. The Linear Quadratic Guassian method allows the designer to shape the principal gains (extremal of singular values) of the return ratio at either the input or the output of the plant, to achieve required performance or robustness specifications. Stability is obtained automatically, so the characteristic loci do not need to be examined. The problem addressed by the theory is to devise a feedback control law which minimises a particular cost function. The solution to the LQG problem is prescribed by the separation principle which stated that the optimal result is obtained by first determining an optimal estimate of the state and using this estimate as if it were an exact measurement of the state to solve the deterministic linear quadratic problem. The procedure reduces the problem to two sub-problems, the solutions of which are known.

The solution to the first sub-problem is that of estimating the state is given by Kalman filter theory. The second problem is to find a control signal which minimises the cost function. The designer is required to select stochastic models for sensor noise, commands and disturbances and defined a weighted mean square error criterion as the standard of goodness. The solution to this is to let the control signal be a linear function of the state. Often, iterations are required to adjust weights in the performance index or to change the stochastic disturbance and noise models etc. before an acceptable solution is obtained. Guidelines for these iterations are provided by procedures such as loop transfer recovery which forces the loop transfer function estimated by the observer approaches the loop transfer function in feedback control.

In spite of its success, LQG controllers are known to have robustness problems. They are sensitive to variations in plant parameters and to unmodelled high frequency dynamics. The development of H^{∞} optimisation addresses this particular issue. However, since the main objective is the determination of a suitable topology rather than the pursuit of controller quality, the robustness issue has only secondary importance.

2.5.8 H^{∞} Optimal Control

A recent tutorial paper on H^{∞} optimisation is given by [Kwakernaak 93]. H^{∞} optimisation is a form of worst-case design and may be regarded as the steady state of differential game between the controls and the disturbances with integral quadratic constraints. An attempt

is made to minimise the quadratic performance index for the worst case of parameter deviations within a bounded range of such deviations and thus guarantee stability robustness. This is a "minmax" problem since the compensator parameters depend on the worst plant parameters and vice-versa. In its most basic form, the design procedure allows the designer to combine the achievement of a particular level of sensitivity (performance) with a particular level of robustness. A related topic is the technique of μ-analysis which is the study of structured uncertainty in control systems. The combination of μ-analysis and H^{∞} optimisation produces the so called μ-synthesis which has the great advantage of providing specified levels of performance robustness.

2.5.9 Parameter Optimisation - Edmunds' Method

In this method [Edmunds 79], the control system design is obtained by adjusting parameters within a previously chosen controller structure so as to optimise the closed loop performance. The performance is optimised in that the designer chooses some target MIMO transfer function which represents a desirable closed loop transfer function. The structure of the controller is also determined and fixed by the designer. An algorithm then adjusts the free parameters in the controller to approximately minimise the difference between the closed loop transfer function actually achieved and the target transfer function. The algorithm allows a flexible problem statement; the controller structure can be fully determined by the designer and the relative importance of achieving each element of the transfer function can be adjusted by specifying appropriate weights. The weights may be frequency dependent.

2.5.10 Graph-Theoretic Approach to the Synthesis of Multivariable Controllers

A brief history of the development of multivariable control shows the relevance of a graph-theoretic approach to the synthesis problem.

1950 State Space Representation

multivariable control system using modern control theory

1960 Controllability and Observability

Problem - differential matrix equations are not robust against parametric perturbation and designer loses the desirable feeling and visual insight. The dimension of large scale system presents another obstacle.

1970 Geometric Approach

Design by co-ordinate free representation of linear vector spaces but does not correspond to traditional control engineering.

The graph-theoretic approach focus on the structural nature of system that holds generically (independent of numeric parameter values) and offers certain advantages over the previous approaches. In [Reinschke 88], the multivariable control synthesis problem is treated in such a framework. It might prove to be a convenient method as information from a graph-theoretic decomposition can be re-utilised.

2.5.11 Drawbacks of the Block Decoupling Methods

Some key results in the development of the Block Decoupling Methods are summarised below.

- [Morgan 64] shows a decoupling synthesising technique using state feedback control.
- [Gilbert 69] gives necessary and sufficient conditions for existence of decoupling state feedback law.
- [Wonham & Morse 70] proved that all closed loop poles of a decoupled system can be arbitrary assigned if the controllability spaces of the decoupled system were independent.
- [Kamyama & Furuta 76] discussed the decoupling of multivariable system by restricted state feedback law.

It has been identified that the methods mentioned above have the following drawbacks. Firstly, there are no bounds on control amplitude. Secondly, no consideration of design objectives such as optimality has been taken into account. Moreover, robustness required exact model and errors lead to poor performance or even instability.

2.6 Decentralised Control

With the control structures suggested by the decomposition process and the information generated by the system analysis in hand, a decentralised control scheme (together with the appropriate controllers) can be applied and the closed-loop system performance evaluated.

There are many ways that decentralisation can be accomplished. Control loops can be closed by a combination of static or dynamic and state or output feedbacks under block diagonal constraints. In [Linnemann & Wang 93], block decoupling by unity output feedback is studied. The use of state feedback suggests some form of decentralised estimation or observers are needed [Siljak 91].

It is also possible to obtain decentralised optimal control scheme although it might involve some complications. The development of the use of H^{∞} optimisation in decentralised control has also been reported [Wu & Mansour 88,90].

2.6.1 Decentralised Control Systems

The concept of high performance systems driven by central computer has become obsolete as larger and more sophisticated systems are built. Notions such as subsystems, interconnections, distributed computing, parallel processing have emerge and it becomes apparent that "well-organised complexity" is the way for the future. The prime concern of dimensionality, uncertainty and information structure constraints of complex systems serve as the main driving force of the development of decentralised control theory.

By considering a dynamic system as an interconnection of subsystems and taking advantage of the special structural features of the decomposed system, a substantial reduction in dimensionality can be made. The subsystems can then be controlled individually and these solutions can be combined in some way to give an overall solution. A comprehensive presentation of this approach is given in [Siljak 91], in which the uncertainties, the interconnections, the information channels, the control configuration of the subsystems and the superior robustness of decentralised control laws are emphasised.

2.6.2 Decentralised Output Control

Decentralised control is attractive because of the reductions in communication requirements. This advantage disappears when state of subsystems are not accessible at the local level. The building of observers might force the exchange of states between subsystems and thus violate the constraints on information structure. A way to over come this problem is to consider static or dynamic output feedback to satisfy both the global (decentralised) and local (output) information constraints simultaneously.

The class of interconnected systems that are always stablisable by decentralised output feedback is identified in [Siljak 91]. Each decoupled subsystem is stabilised by a dynamic controller. In the presence of interconnections, stability is retained by making sure that the gains of the loops in the overall system are sufficiently small. In this way, the compensated system consists of linear plants interconnected by memoryless non-linearities.

The approach to build connectively stable systems is to use decentralised feedback and treat interconnections as a disturbance of subsystems dynamics. Both the concept of almost invariant subspaces [Willems 81] and the small gain theory can be utilised for this purpose. In [Siljak 91], chapters are devoted to the following topics:

- Decentralised Stabilisation in Presence of Non Structurally Fixed Modes
- Choice of Feedback Control Structure to Avoid Fixed Modes
- Design Techniques for Parametric Robustness
- Structural Robustness

2.6.3 Decentralised Optimisation

Attempts to formulate decentralised control strategies by extending standard optimisation techniques have not been successful because of the non classical decentralised information structure. Nonetheless a number of ad hoc methods exist. The presence of essential uncertainties in the interconnections among the subsystems complicate the notion of optimality further. Robustness in complex systems has to be considered as part of the problem rather than part of the solution.

In [Siljak 91], by imitating the concept of the decentralised structure of competitive market systems, complex systems comprising interconnected subsystems with distinct inputs are considered. Linear state feedback is used to optimise a cost function for the local subsystem, ignoring other subsystems. The suitability of the decentralised Linear-Quadratic (LQ) control law is established by showing stability of the overall closed-loop interconnected system. A local optimal decentralised control is not optimal in general for the interconnected system and an index of sub-optimality is defined. This index can be used to measure the cost of robustness to structural perturbation.

2.6.4 Decentralised Servomechanism

In a series of papers, [Davison 76a,b,c] gave a treatment of the robust control of a general servomechanism problem. Specifically, in [Davison 76a], the robust decentralised servomechanism problem, that is to find a decentralised controller so that some outputs of the system asymptotically track given reference inputs independent of any external disturbance, is discussed. The paper finds necessary and sufficient conditions for the existence of the solution. Complete characterisation of all decentralised controllers is also given. It is also shown that under certain mild conditions, there "almost always" exist a solution for any composite system. The main result of the paper is given in terms of the properties of the fixed modes of the system. In [Vaz & Davison 89], the scope of the problem considered is extended.

2.7 The Concept of Block Relative Gain Array

2.7.1 Introduction

One of the underlying principles for the development of the Design Methodology for Multi-Axis Machinery is the design of distributed controllers. First, a full MIMO controller for the overall plant is designed and its performance will be the upper bound of the "achievable" performance with distributed control.

The inherent couplings, which are part of the plant dynamics, will be compensated by this full MIMO controller. If all performance specifications are met, this controller will also satisfy all the synchronisation requirements, which are imposed by the designer. Also, the relative importance of inherent couplings and synchronisation requirements are all captured by the off diagonal terms of the controller. Thus the centralised designed controller provides a unified approach that can deal with the two types of cross axes interactions, namely, the inherent couplings and the synchronisation requirements, simultaneously.

Due to various practical limitations, it is not possible to implement the above controller. For example, a ten axis system will require an update of a 10 by 10 matrix for each sampling period. An alternative is to opt for a block decentralised implementation. That is to find a set of controllers (blocks) equivalent or as close as possible to the full controller. This set of controllers will be distributed physically across the system and has the advantage of greatly reducing the complexity and thus the computation and

communication cost of the system. However it will also suffer from some degradation from the best "achievable" performance of the centralised controller. Such an approach allows the designer to exploit a broader class of control structures and are no longer restricted to the extremes of either complete decentralisation (SISO loops) or complete centralisation (full MIMO). Clearly, careful consideration should be given to the block structure to which the controller is decentralised.

For a system with a particular block structure, interactions exist between the various blocks. To select the "optimal" structure, some form of measure of the relative magnitudes of these interactions within the structure will be required. The Block Relative Gain (BRG) array is found to have the right properties for this purpose.

2.7.2 Interaction Measures

The Relative Gain Array (RGA) has been widely used as a measure of interaction and as a tool for control structure selection for single-loop controllers. RGA is originally defined at steady state [Bristol 66] but can be extended to higher frequencies [Bristol 78]. It has the advantages of depending on the plant model only and it is scaling independent. It is also easy to generalise to the Block Relative Gain (BRG) introduced by [Manousiouthakis, Savage & Arkun 86].

Based on its definition, it seems reasonable that RGA should have some use as a performance or stability measure for decentralised control. A thorough review of the use of the RGA is given by [Grosdidier, Morari & Holt 85]. In [Arkun 87,88], DBRG (Dynamic Block Relative Gain) and Relative Sensitivity which include the controller are proposed.

The development of the BRG concept can be summarised by the following papers.

- relative gain array [Bristol 66]
- block relative gain [Manousiouthakis, Savage & Arkun 86]
- dynamic block relative gain [Arkun 87]
- relative sensitivity matrices [Arkun 88], [Lee & Park 92]
- non-linear block relative gain [Manousiouthakis & Nikolaou 89]
- non-square block relative gain [Reeves & Arkun 89]

2.7.3 Definition of BRG

Consider a *n x n* transfer function matrix G(s) that is partitioned as follows:

$$G = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix}$$

(2-2)

and

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = G \cdot \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

(2-3)

where G_{11} is *m x m* and G_{22} is *(n-m) x (n-m)*. The plant is to be controlled under the decentralised feedback structure in the figure below with compensator K and filter F.

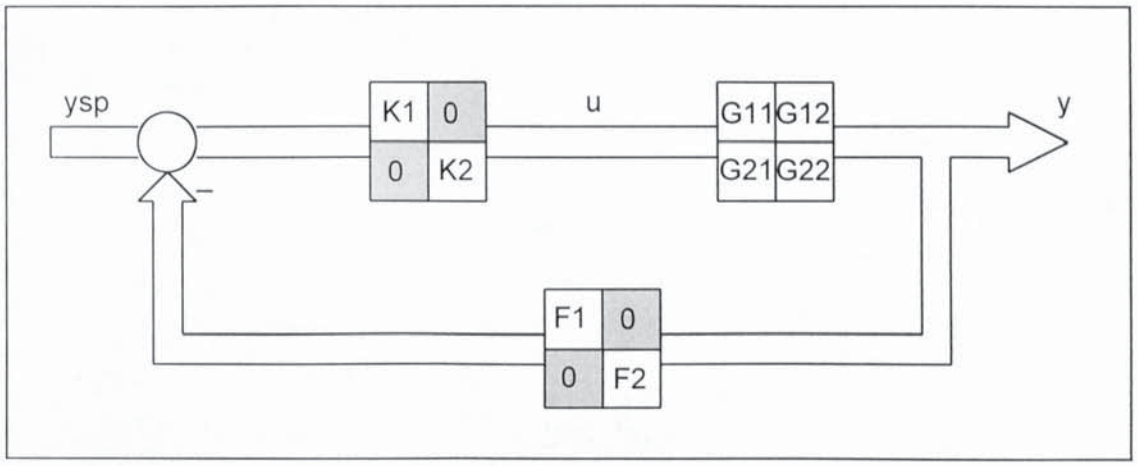

Figure 2-5 Decentralised Feedback Structure

Assuming G^{-1} exists,

$$u = G^{-1} \cdot y$$

(2-4)

then

$$\frac{\partial y_1}{\partial u_1}_{\substack{u_2 = 0 \\ F = 0}} = G_{11}$$

(2-5)

where G_{11} denotes the block gain between y_1 and u_1 when all loops are open (F=0). Similarly,

$$\left.\frac{\partial y_1}{\partial u_1}\right|_{\substack{y_2=0\\F_1=0\\F_2=I}} = \left(\left[G^{-1}\right]_{11}\right)^{-1} = G_{11} - G_{12}G_{22}^{-1}G_{21}$$

(2-6)

if G_{22} is non singular. $([G^{-1}]_{11})^{-1}$ is the block gain between y_1 and u_1 when the first *m* loops are open (F_1=0) and the last *n* - *m* loops are closed (F_2=I) and under perfect control (y_2=0) by some compensator K_2. The *m*-dimensional block relative gain (left and right) is defined as follows:

$$BRG_L = \left[\left.\frac{\partial y_1}{\partial u_1}\right|_{\substack{u_2=0\\F=0}}\right] \cdot \left[\left.\frac{\partial y_1}{\partial u_1}\right|_{\substack{y_2=0\\F_1=0\\F_2=I}}\right]^{-1} = G_{11} \cdot [G^{-1}]_{11}$$

(2-7)

$$BRG_R = \left[\left.\frac{\partial y_1}{\partial u_1}\right|_{\substack{y_2=0\\F_1=0\\F_2=I}}\right]^{-1} \cdot \left[\left.\frac{\partial y_1}{\partial u_1}\right|_{\substack{u_2=0\\F=0}}\right] = [G^{-1}]_{11} \cdot G_{11}$$

(2-8)

2.7.4 Significance of BRG

Consider the closed loop performance of the system in Figure 2-5, the significance of BRG can be illustrated by the following cases:

- F_1=0 & F_2=0 (no feedback)

$$\left.\frac{\partial y_1}{\partial y_1^{sp}}\right|_{\substack{y_2^{sp}=0\\F=0}} = G_{11}K_1$$

(2-9)

- F_1=0 & F_2=I (perfect control in y_2 by K_2)

$$\left.\frac{\partial y_1^{sp}}{\partial y_1}\right|_{\substack{y_2=0\\F_1=0\\F_2=I}} = K_1^{-1}\left[G^{-1}\right]_{11}$$

(2-10)

- F_1=I & F_2=I (full feedback with y_2 under perfect control by K_2)

$$\left.\frac{\partial y_1^{sp}}{\partial y_1}\right|_{\substack{y_2=0\\F=I}} = I + K_1^{-1}\left[G^{-1}\right]_{11}$$

(2-11)

with these relations, we can derive the followings:

$$\left.\frac{\partial y_1}{\partial y_1^{sp}}\right|_{\substack{y_2=0\\F=I}} = \left[I + \left(BRG_L^{-1}\cdot\left.\frac{\partial y_1}{\partial y_1^{sp}}\right|_{\substack{y_2^{sp}=0\\F=0}}\right)^{-1}\right]^{-1}$$

(2-12)

Thus BRG_L^{-1} is the frequency dependent matrix factor by which the open-loop gain $G_{11}K_1$ must be pre-multiplied so that the effect of the other loops is taken into account in the closed-loop response of y_1 to its set points y_1^{sp}. If $BRG_L = I$,

$$\left.\frac{\partial y_1}{\partial y_1^{sp}}\right|_{\substack{y_2=0\\F=I}} = \left[I + \left(G_{11}K_1\right)^{-1}\right]^{-1}$$

(2-13)

which is the closed loop response of y_1 to its set points y_1^{sp} when no other loops exist. Thus for the *m x m* block under consideration, when the other *n - m* outputs are under perfect control, the closed loop performance is a continuous function of BRG_L. If BRG_L=I, the block operated as if it is isolated from the rest of the plant. In this sense, BRG can be treated as an interaction measure.

2.7.5 Properties of BRG

The following properties of BRG are stated without proof , detailed discussion can be found in [Manousiouthakis et.al. 86].

- BRG_L (BRG_R) is not affected at all by the ordering of the first m inputs (outputs) and the ordering of the last n - m inputs and outputs. For all the transfer functions that have the same first m inputs and outputs but in different arrangements, their BRG's turn out to be trivial rearrangements of each other and can be considered as equivalent. Thus only one BRG is needed to be examined for a unique group of inputs and outputs.

- BRG_L (BRG_R) does not depend on the scaling of the first m inputs (outputs) and the last n - m inputs and outputs but it does depend on the scaling of the first outputs (inputs).

- BRG has well-defined diagonal elements that they remain on the diagonal but not necessarily at the same locations when G(s) is trivially rearranged. This implies that the designer will only need to examine m diagonal terms for all the BRG's corresponding to a particular group of m inputs and outputs.

- The eigenvalues of BRG, have the property that they are independent of scaling and that

$$\lambda_i(BRG)_L = \lambda_i(BRG)_R \; \forall i = 1 \ldots m \qquad (2\text{-}14)$$

2.7.6 An Example of Topology Selection

The following is an example to illustrate the use of BRG arrays for a simulated system. This simulated system consists of 3 independent axes labelled as axes-A, B & C. A material web is going around axes-B & C and axis-A is running on its own. The axes are mounted on perfectly rigid platform so no vibration is present. Interaction is therefore expected between axes-B & C but not between axis-A and the others.

In the next step of topology selection, a full MIMO controller is designed for this plant (see section 2.7.1). Assuming for whatever reason that a 3-by-3 MIMO controller is not desirable and that implementation is restricted to 2-by-2 controllers. The possible configurations for block decentralisation on the controller are:

- [A B] & [C]

- [A] & [B C]

- [A], [B] & [C] - fully decentralised

The BRG arrays obtained on a full MIMO controller for the first two configurations are shown below for comparison.

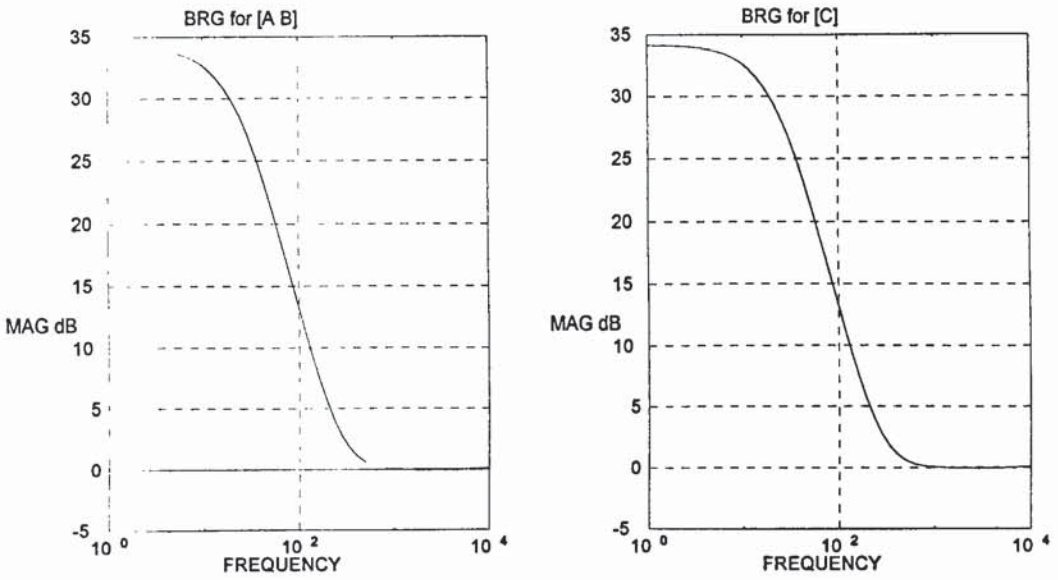

Figure 2-6 Block Relative Gain Array for [A B] & [C]

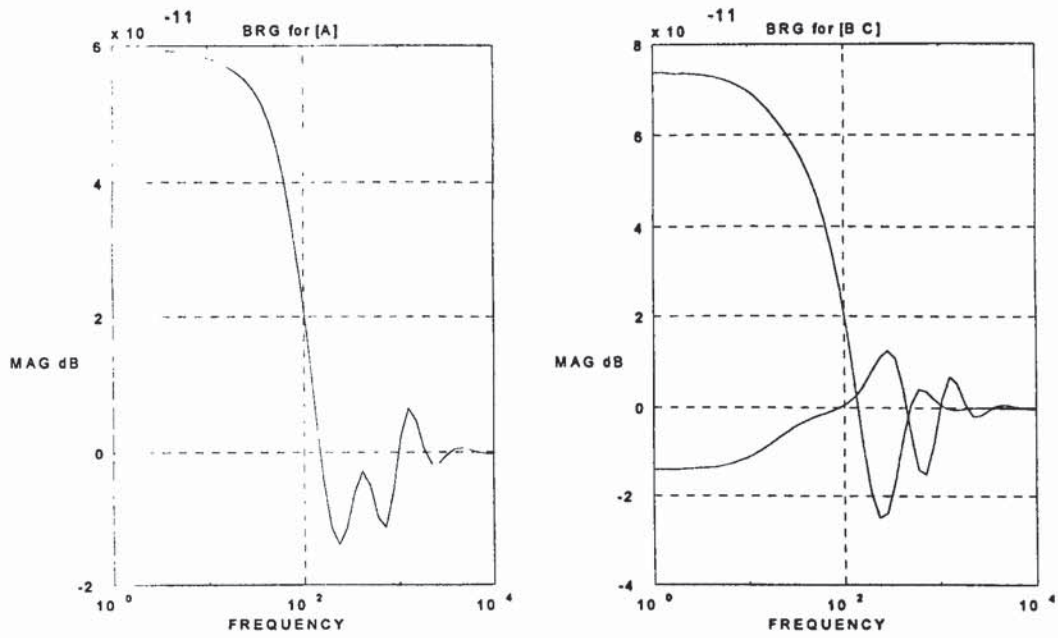

Figure 2-7 Block Relative Gain Array for [A] & [B C]

Consider the block relative gain for axis-C in Figure 2-6. Over the range of frequencies in question, the open loop transfer function of the isolated axis-C will have to be pre-multiplied by this gain for it to behave as if the other loops are present. In other words, the

BRG represents the combined influence of the other loops on axis-C normalised to its gain. The BRG of axes-A & B can be interpreted in a similar way.

Recall that there is a material web around axes-B & C and axis-A is on its own. This is clearly shown in Figure 2-6 where axis-B and axis-C belong to different subsystems. The relative size of the external (to the respective subsystem) influence can be as high as 34 dB at low frequencies. This external influence becomes less significant as the frequency approaches the system bandwidth.

By comparison, in Figure 2-7, where axes-B & C belong to the same subsystem, the only cross axes interaction in the form of a web around axes-B & C does not span across subsystems. The corresponding BRG's are of magnitudes no larger than 10^{-10} dB. It is therefore clear that the configuration represented by Figure 2-7 is the preferred topology for block decentralisation.

The system described above is a trivial example in which only web coupling is present and no synchronisation requirement was considered. However, the advantage of the design methodology actually lies in the fact that inherent couplings (part of the plant model) and synchronisation requirement (part of the performance specification) can be measured and dealt with simultaneously. Further simulation will be made to demonstrate this point.

3. Design Methodology

3.1 Introduction

The design methodology is aimed at formalising and automating the process of design and synthesis of the control strategy of servo system for multi-axis high speed machinery. The target is a distributed computer control system. Special emphasis is placed on the control topology (the way axes are grouped into subsystems) and the main focus is on how to deal with the interaction between subsystems. Also of importance is that the methodology is intended to be implemented as a computed aided design software for servo designs.

A process of system decomposition takes into consideration the electro-mechanical properties (intrinsic constraints) and the target behaviour (imposed constraints) of the machine and suggests possible topologies. A decentralised control scheme is employed which involves deriving local multivariable controllers for each subsystem resulted in the decomposition process. The resultant control system is then evaluated by simulation.

3.2 Justification for MIMO Control

Currently, a typical multi-axis machine is treated as a set of parallel axis with each axis treated as an independent Single Input Single Output (SISO) system. The servos are designed for each of these axis independently by closing the parallel loops one at a time starting from the fastest loop. At each stage, the faster loops are detuned to accommodate the slower loops. This approach is often termed "successive loop-closure" or "sequential loop closing" which can easily give results that are far from optimal because poorly co-ordinated controls fight each other and thus wasting control authority. Conceptually it is clear that further performance can be harnessed if some form of simultaneous synthesis of controllers is employed.

A more general approach is to apply Multi Input Multi Output (MIMO) control. Such an approach requires an alternative look at the plant being controlled. That is, the plant has to be considered as a full MIMO system. A closer look at the interaction between the axes (loops) is required for its potential to be exploited.

On the axis-to-axis level, the availability of information from other axes on the same machine provides assessment of their influence and thus possible improvement in

(synchronisation and decoupling) performance. By taking into account the cross axes interaction in the control system model, it also remove the restriction that system axes have to be isolated in design. The drawback is the increase of complexity in the design of controller and possibly the implementation of control software.

3.3 Justification for Decentralised Control

The ideal control system will be the use of a full MIMO controller based on the complete MIMO description of the plant. Such a controller model can be described by a matrix of transfer function with the diagonal elements representing the axes' local characteristics and the off diagonal elements representing the cross axis responses. In practical terms, a system with a large number of axes involves large matrices and its implementation will be restricted by both computation and communication cost.

If the system model is divided into subsystems by grouping different numbers of individual axes together and these subsystems are treated as isolated systems, the theoretically achievable performance, disregarding implementation issues, of the overall system will be degraded compared with a full MIMO controlled system but the size and cost of the system will be greatly reduced. To select the correct grouping of axes and thus the 'optimal' topology, two main issues have to be dealt with. By looking at some appropriate measure of the relative strength of the interaction between any two subsystems against a design threshold, the small interaction in the model can be ignored and a decentralised control topology is obtained. Further consideration of synchronisation requirements imposed other constraints and the topology will be altered accordingly.

In the extreme, when the system is fully decentralised, each subsystem consists of only one axis and the MIMO controller is reduced to a set of SISO controllers. As these controllers are obtained from some simultaneous synthesis, they should have every chance to perform just as well as those obtained by sequential loop closing. Between the full MIMO and the fully decentralised control, there are topologies of different degrees of complexity. One possibility is the use of master-follower configuration within the decentralised subsystems.

To satisfy the performance requirements and the various practical considerations, the problem becomes one of searching for the right topology which balances the achievable performance against the system complexity and hardware limitations.

3.4 Size of the Topology Selection Problem

This section is an exercise to illustrate the size of the topology selection problem in relation to block diagonal structures. Although only square systems are considered, the methodology also cover non square (number of inputs and outputs are not equal) systems.

3.4.1 The Number of Possible Interconnection

Define interconnection matrix

$E = e_{ij}$ where for $i \neq j$

$e_{ij} = 1$ if output of j influences input of i

$e_{ij} = 0$ otherwise

Consider a system of n axis. When there is no constraint on the pattern of interconnections between them, there can exist several different types of cross influence:

[1] master/slave configuration

output of axis i influences input of axis j but not vice versa

$e_{ij} = 1$

$e_{ji} = 0$ $i \neq j$

[2] full coupling

output of axis i influences input of axis j and vice versa

$e_{ij} = e_{ji} = 1$ $i \neq j$

[3] feedforward

input of axis i influences output of axis i directly

$e_{ii} = 1$ (note an abuse of definition)

When all three types of influences are allowed, there are $2^{n\hat{}2}$ different possible interconnection patterns. When feedforward are excluded, there are $2^{n\hat{}2-n}$ and if only full couplings are allowed, there are $2^{(n\hat{}2-n)/2}$. Examples are given as:

Table 3-1 Number of Possible Interaction Patterns

Number of Axis	Influence Type [1] & [2] & [3]	Influence Type [1] & [2]	Influence Type [2]
1	2	1	1
2	2^4	2^2	2
3	2^9	2^6	2^3
4	2^{16}	2^{12}	2^6
5	2^{25}	2^{20}	2^{10}
6	2^{36}	2^{30}	2^{15}
7	2^{49}	2^{42}	2^{21}

3.4.2 The Number of Block Diagonal Structure

Restricted by a block diagonal structure, axes are grouped into subsystems of various sizes. For example, a 4 axis system can be grouped into subsystems of size (1,1,1,1), (2,1,1), (2,2), (3,1) and (4). With respect to these 5 block structure configurations, the 4 axis provides altogether 15 different combination of groupings.

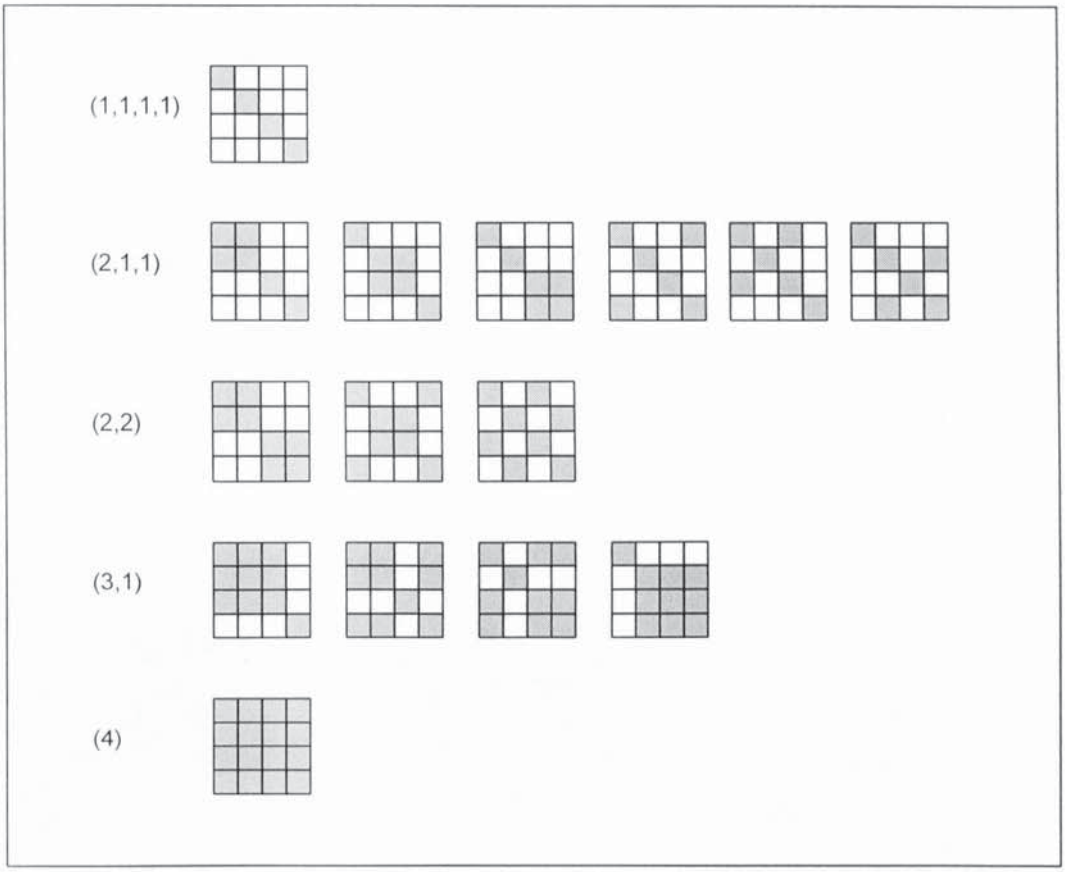

Figure 3-1 Possible Patterns for Block Decentralisation

With reference to the above table, the total number of possible interconnection patterns (as restricted by the block structures excluding (4)) can be obtained as follows: (2,1,1) has 6 axis combination for which, if all 3 types of interconnections are considered, there are 2^4x2x2 interconnection patterns for each combination, and hence 384 patterns for this configuration.

Table 3-2 Number of Interconnection Patterns for a 4 Axis System

Block Configuration	No of Axis Combination	Influence Type [1] & [2] & [3]	Interconnection Patterns
(1,1,1,1)	1	2x2x2x2	16
(2,1,1)	6	2^4x2x2	384
(2,2)	3	2^4x2^4	768
(3,1)	4	2^9x2	4096

The sum of all the configurations comes to a total of 5264 interconnection patterns as compared with the unconstrained (2^{16}=) 65536 patterns. This represents a 92% reduction.

Examples indicating the block structure constraint are shown as below:

Table 3-3 Reduction of Interconnection Patterns by Block Structures

Number of Axis	Block Structure Configurations	Axis Combinations
1	1	1
2	2	2
3	3	5
4	5	15
5	7	52
6	11	204
7	15	879

These numbers indicate the dimensionality of the configurational problem of the control system can be greatly reduced by imposing a block diagonal structure.

3.5 Design Framework

At the moment, much of the servo control system of the multi-axis machinery is designed by an trial-and-error approach. The difficulty lies in the fact that at the early stages of design, the hardware involved often only exist as drawings on paper and therefore the control system design can only rely on the designer to draw on his/her experience. This is increasingly the case as efforts are being put to reduce development cycle time and the concept of concurrent engineering is taken aboard. As a result, control systems are over designed initially and then fine tuned at a later stage. As the building of the machine develops, more of the physical parts become available and better design of the control system is obtained by an on-going, iterative refinement process.

Based on the current practice, the objective is to construct a design framework by which a formalised and streamlined design cycle for the control system can be carried out with ease. As the design process is highly iterative and draws heavily on experience, system information should be stored and updated readily within the framework. Design and analysis procedures should be automated as far as possible. This calls for a generic system model in which system information is easily updated and manipulated. An important element is the capability of the retrieval of past designs which provides examples and indications of how current design should proceed. As more accurate information becomes available, the conceptual model converges to the actual machine and the control system designed becomes better.

The main objective of the design methodology is to be able to select the best suited topology early on in the design cycle. That is to say trying to determine a topology that will be most cost effective, often with minimal information and when no physical parts are available. The approach will be based on consideration on the relative sizes of the dynamic interaction between the axes. In practice, it means the topology is likely to be obtained from a simulation model.

Clearly, the quality of any solution depends ultimately on the accuracy of the model. The objective of the project is to provide a design framework for the selection of a suitable topology for decentralised control but not the design of a superior controller. With a particular topology, different types of controllers can be designed. Evaluation can only be made on the same footing. For example, comparing control systems of different topology but synthesised by the same method.

3.6 Design Cycle

System information is categorised into various model components and incorporated into the generic system model. For visual and conceptual clarity, these are represented by block diagrams. In the background, however, standard state space and transfer function representation will be generated.

Design objectives such as performance requirements as well as practical restrictions are translated into design criteria. An emphasis is placed on their implication on the axis/shaft level topology. By looking at the interaction between different groups of axes, a topology can be selected.

A set of decentralised MIMO controllers is designed and their performance is evaluated by simulation. With a suitable test rig, these controllers can also be implemented and tested. At the end of one design cycle, an indication of how well the control system performs determines whether further iteration is necessary.

3.7 A Unified Approach

Consider a design problem where no synchronisation requirement is imposed and the only coupling between the axes derives from cross axes interaction. Information on the plant alone will be sufficient to determine a suitable topology for block decentralisation. Consider also a design problem where the plant has no inherent couplings but synchronisation is required. Again, the synchronisation requirements alone will be sufficient to determine the appropriate control topology. However, when inherent couplings are present and synchronisation is required, it is less obvious how a suitable topology can be obtained.

Consider a multi-axis plant which is represented by a transfer function matrix with its diagonal terms representing the axes local dynamics and the off-diagonal terms representing the inherent couplings. The performance specification can be translated to the target response of the overall system. However, embedded in this specification will be both the synchronisation requirements and the tolerated level of inherent coupling.

To achieve the target response, a full MIMO controller can be designed which will represent the best achievable theoretical solution. The diagonal terms of such a controller will be control loops closed locally for each of the axes. The off-diagonal terms will however represent the cross axis coupling needed to be imposed to achieve the various performance requirements.

If all performance specifications are met, the controller will compensate for the inherent couplings and satisfy the synchronisation requirement simultaneously. Also, their relative importance is captured by the off diagonal terms of the controller. Thus the centralised designed controller provides a unified approach that can deal with the two types of cross axes interactions, namely, the inherent couplings and the synchronisation requirements, simultaneously.

3.8 Practical Issues for Distributed Control

One of the underlying principles for the development of the Design Methodology for Multi-Axes Machinery is the design of distributed controllers. Consider a ten axis system under full MIMO control. The controller will require updates of the equivalent of a 10 x 10 matrix for each sampling period. The realisation of such a controller may be found to be prohibitive due to the cost of communication and computation.

An alternative is to opt for a block decentralised implementation. That is to find a set of controllers (blocks) equivalent to or as close as possible to the full controller. This set of controller will be distributed physically across the system (on different controller/DSP cards) and has the advantage of greatly reducing the complexity and thus the computation and communication cost of the system. However, it will also suffer from a degradation from the best "achievable" performance of the centralised controller. This approach allows the designer to exploit a broader class of control structures and are not restricted to the two extremes of complete decentralisation (SISO loops) or complete centralisation (full MIMO). Clearly, careful consideration should be given to the block diagonal structure to which the controller is decentralised. Note that the plant is taken as a whole and only the controller is decentralised.

3.9 Outline of the Topology Selection Procedure

In general, for a plant with n axes, there are $2^{n\hat{}2}$ possible patterns or topologies for interconnection (see section 3.4). However, some practical considerations, such as the knowledge of the mechanical structure and the geography of the machine will limit the number of possible topologies. Furthermore, the size of the MIMO controller that can be implemented within the sampling period greatly reduces the number of possibilities. It is expected that under most circumstances, the number of serious candidate topologies will be manageable by manual comparison.

When a MIMO system is decentralised subject to a certain block diagonal structure, each block will become a subsystem. Since each subsystem will be treated independently, it is desired that the interaction between any two subsystems to be minimal. In fact, if these subsystems are perfectly isolated, there will be no loss of information and hence no degradation of the performance by the process of decentralisation. Therefore, to determine the "goodness" of a topology, one can consider the level of interactions between the various subsystems. The more isolated the subsystems are, the better will be a topology

which does not contain connections between those subsystems. For the purpose of measuring the level of interaction between a subsystem and the rest of the overall system, it is found that the Block Relative Gain Array to be a useful tool. A more detailed description is given in the next section.

To summarise, the principle of the proposed decentralised control approach is to start with the best achievable theoretical solution which represents the best possible trade-off between the various conflicting requirements. The performance requirements are then relaxed such that some of the cross axes interaction, subjected to a suitable topology, can be ignored. The resultant system can then be considered as a collection of smaller sub-systems with lesser degree of complexity that can be handled with fewer difficulties. Once the 'optimal' topology is identified, MIMO controllers can be redesigned for each of the subsystems, as if they are isolated systems. Figure 3-2 summarises the design methodology. In actual application, the design will be a continuous iteration from all the stages and the paths for iterations are omitted from the diagram for clarity.

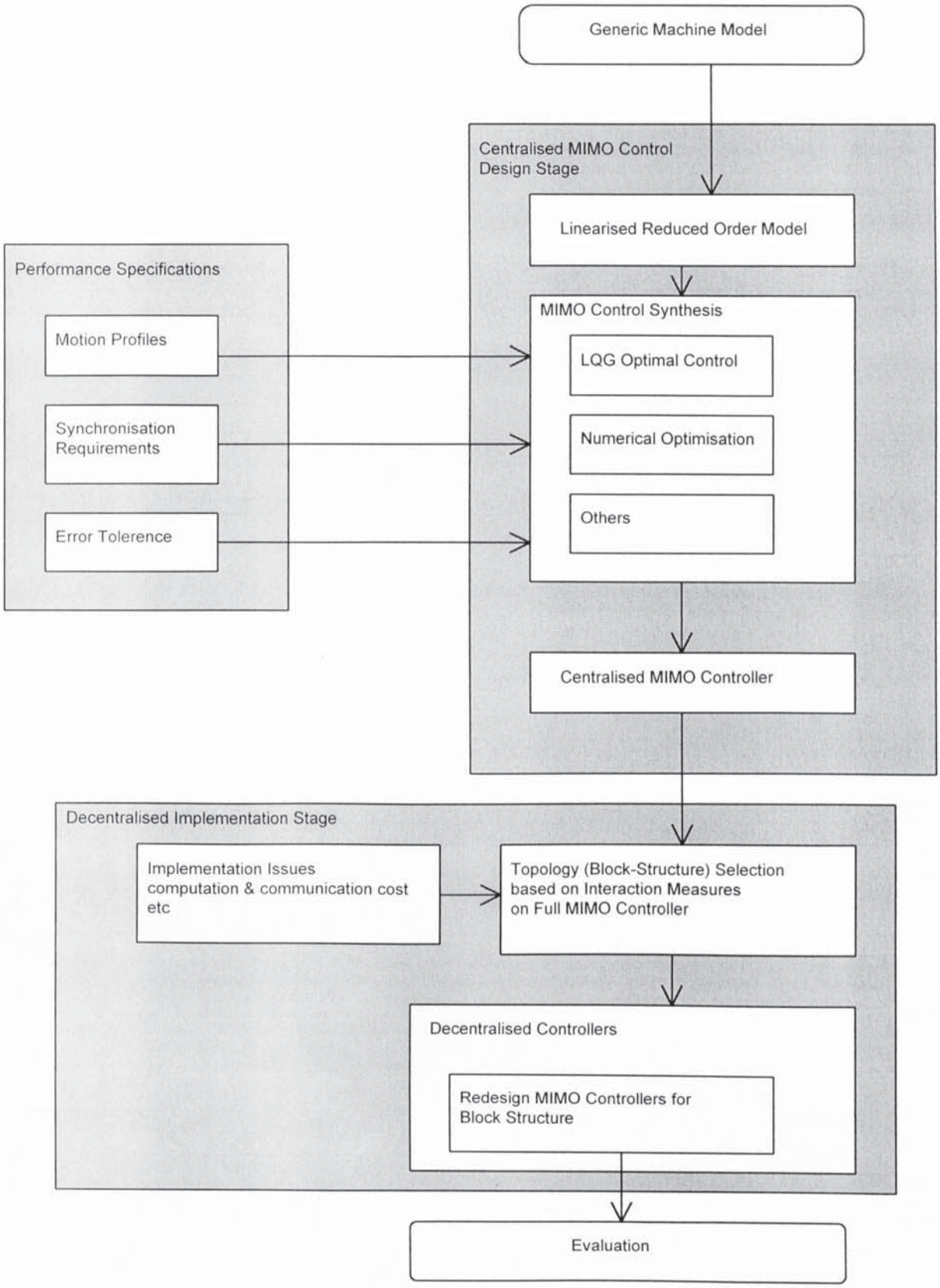

Figure 3-2 Design Methodology

4. Generic Machine Model

4.1 Generic Machine Model

The generic machine model is a combination of the following components in a block diagram representation:

1. MIMO servo block
    - Individual servo
    - DC bus coupling
2. MIMO load block
    - Individual load
    - Web coupling
3. Support structure block
    - Structure vibration modes
    - Structural coupling
4. Stiffness matrices
5. Torque distribution block
    - Individual shafts inertia
    - Linkage stiffness matrix
    - Input shaft stiffness vector
    - Output shaft stiffness vector

The overall model has a modular structure which, so far as possible, provides a one-to-one correspondence between a physical part and a conceptual component. This is advantageous because different combinations of physical parts can be modelled easily and therefore different ideas can be tried out rapidly. The models of individual components, such as servos or typical mechanisms can be gathered into libraries and made available to future designs. As physical parts become available during the building of the machine, actual dynamic responses of these parts can be measured and substituted into the model hence improving the validity of the model.

The generic model also has a layered or hierarchical structure where similar components are grouped together. Such an arrangement clarifies the overall structure and the signal pathways of the model. It also provides a general layout for all designs. Another important feature is the use of stiffness matrices in the feedback paths to represent coupling (web and distribution) between different shafts. These matrices have entries determined only by the actual configuration and the stiffness of the linkages. If any changes have to be made to the design, simulation tests can easily be conducted because the model will be relatively easy to be modified.

4.2 Servo Axis -- Motor & Drive Pair

The servo axes used are usually off the shelf units. A number of the Electrocraft BRU200 servo systems were available to this project and the modelling of a servo axis is based on this. Similar approaches can be taken to model other servo systems. The BRU200 servo axis consists of a drive which converts 3-phase AC current to drive a S-4075 DC brushless motor, as depicted in Figure 4-1.

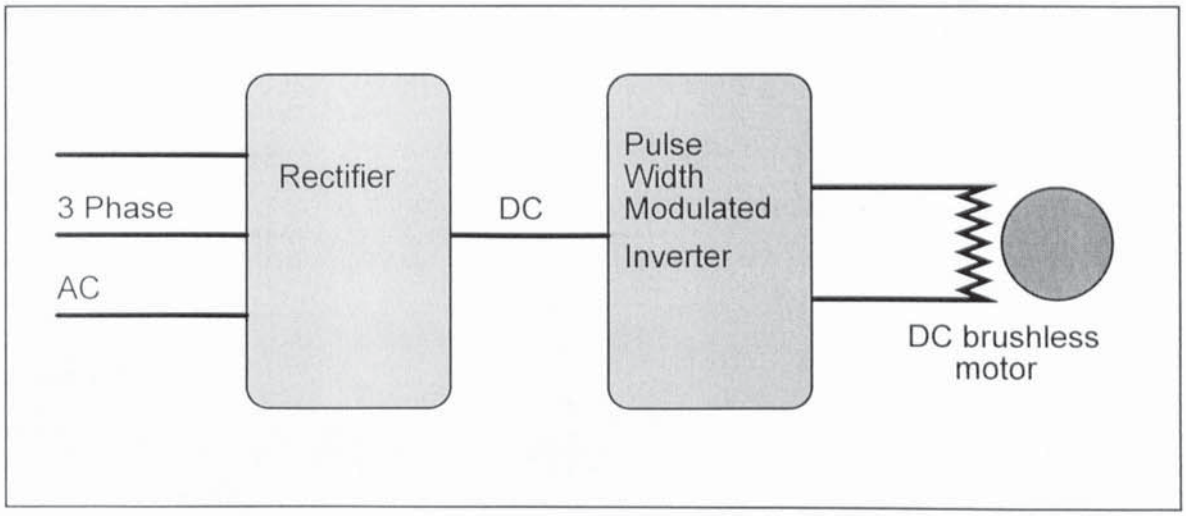

Figure 4-1 Schematic of a DC Motor and Drive

For all intended purposes, a servo axis on its own (driving no load) can be modelled by a DC equivalent as shown in Figure 4-2, which is a block diagram drawn in SIMULINK. In a conventional servo drive system, velocity is usually the variable to be controlled and thus a velocity control loop is usually present as part of the self contained unit. That is, the servo axis usually operates in velocity mode. However, one of the aims of the methodology is to replace the individual control loops of the multiple axes with a MIMO controller and therefore only the very core of the servo system, without the velocity loop,

needed to be modelled. The velocity loops will effectively be closed by the MIMO controller and the servo axes will be operated in "torque mode".

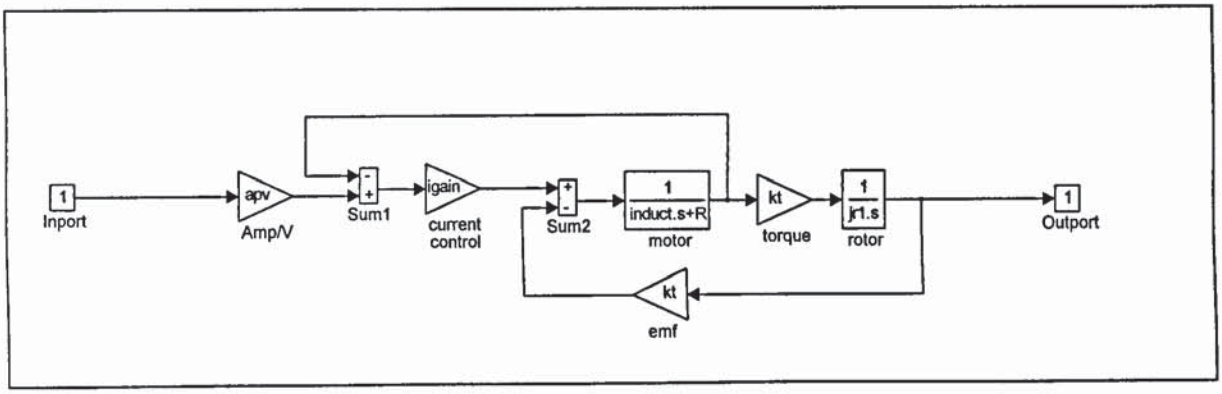

Figure 4-2 DC Equivalent Model for the BRU200 Servo Axis

In essence, the motor part of the servo (Figure 4-3) is represented by a 2nd order model

$$\frac{V_t \cdot K_t}{J_r s(Ls + R) + K_t^2} \tag{4-1}$$

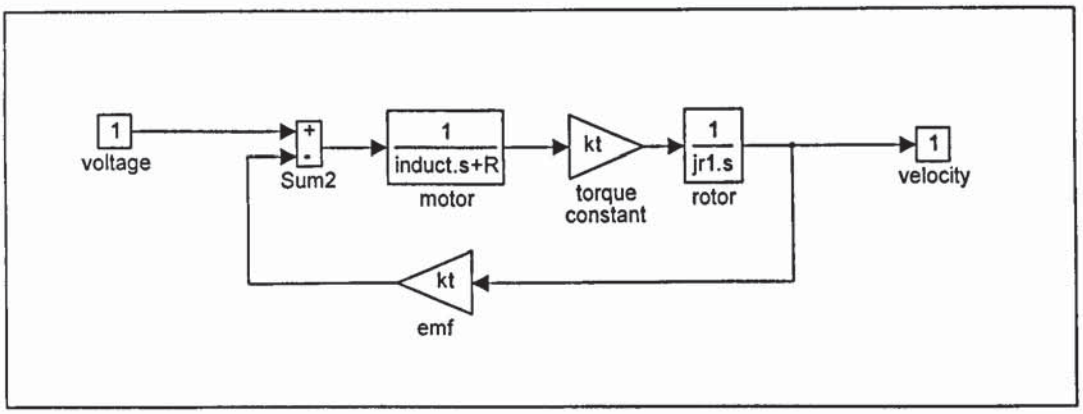

Figure 4-3 2nd Order Motor Model

which relates voltage across the motor terminals to rotor velocity. In this model, K_t is the torque constant, L is the winding inductance, R is the winding resistance, J_r is the rotor inertia and s is the Laplace operator. A back e.m.f. loop is also present. The function of the drive is modelled by the ampere per voltage amplifier together with a fast current loop (Figure 4-2). Nominal values for these constants can be obtained from the manufacturer or measured with simple experiments.

For the actual servo hardware, there is +/- 10V analogue input which represents the full range speed demand of 3000rpm in either direction. There is also a monitor output on the front panel of the drive which gives again an analogue +/- 10V full range motor speed.

The servo axis, for a given purely inertia load, in this regard, can be treated as a black box with a unique velocity to voltage frequency response. A bode plot and a step response plot of the above model are given in Figure 4-4 and Figure 4-5 below:

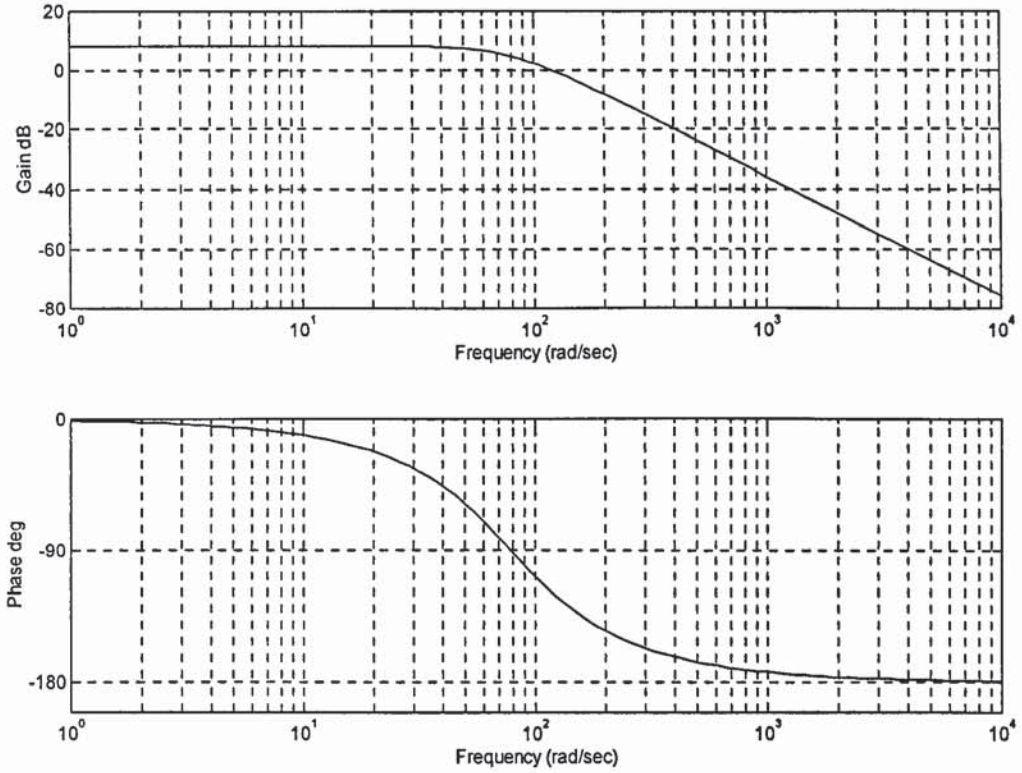

Figure 4-4 Bode Plot for the Tentative BRU200 Model*

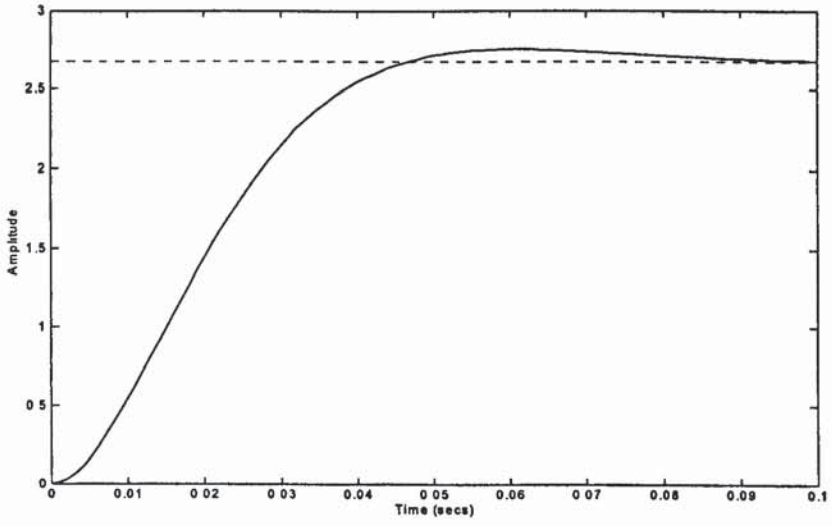

Figure 4-5 Step Response of the BRU200 Model

* Frequency will be measured in unit of rad s^{-1} in all subsequent plots.

With some fine tuning, this model can be adjusted to resemble closely the measured behaviour of the actual hardware. In terms of the design process, there is no reason why the servo model cannot be replaced by actual measured response altogether to produce a better overall model. For the same reason, components of the generic model can be replaced when actual hardware measurements become available.

For most intended applications, the objective is a servo that follows a position rather than a velocity profile. An integrator is appended to the model to reflect this. Another important feature that has to be added in the model is a channel for the application of external torque which allows the axis to deliver torque to a load and allows interactions between axes.

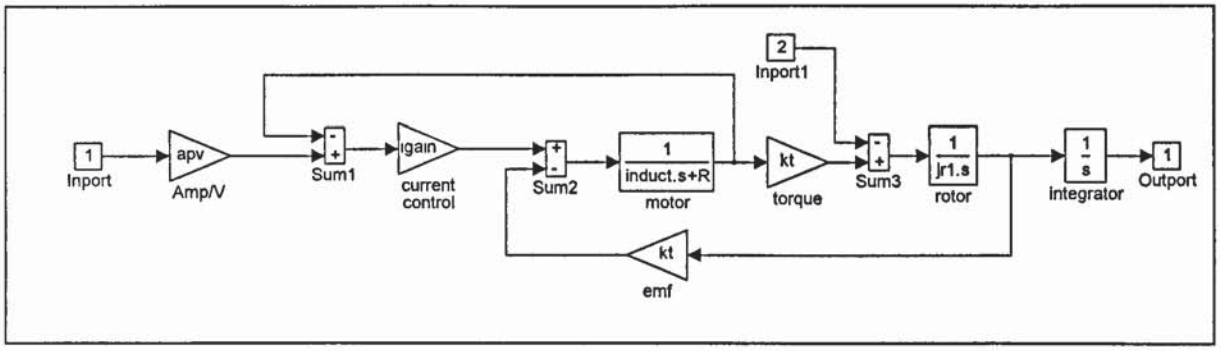

Figure 4-6 Servo Axis Model with External Torque Taken into Account

This model of a servo axis can be encapsulated into a 2 inputs 1 output block component as depicted in Figure 4-7. In fact, a library of different servo axes can be built up with members all conforming to the same input/output signal pattern. During the development of the control system design, different servos can easily be incorporated in the simulation to be tested for suitability. If the actual dynamic response data of the servo is available, its model can be replaced by a lookup table, having as many dimensions as there are states in the model.

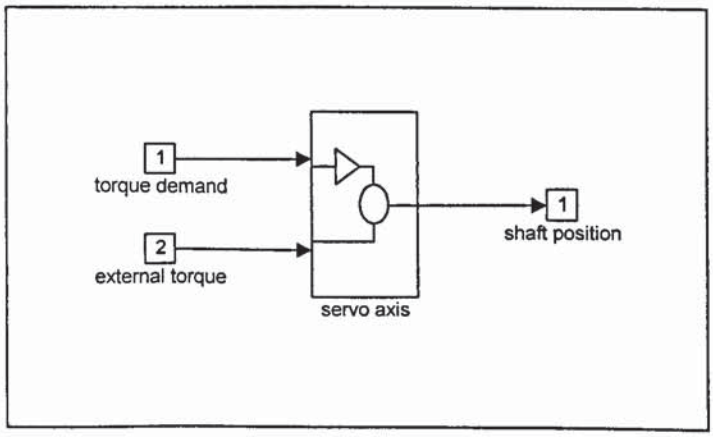

Figure 4-7 Block Representation a Servo Axis

4.3 Load Description

To a mechanical engineer, the loading on the axes can be classified into the static, dynamic and shock types, depending on the function of the corresponding mechanisms. In most cases, during the design of these mechanisms, the time motion of the servo axes can be translated into the torque and power requirements. These relationships can usually be represented or approximated closely by differential equations or the equivalent transfer functions. Often, a computer aided design package, such as CAMLINKS, is used. With some simple manipulations, these in turn can be put into a block representation. For example, a simple inertia J_L is given by the following transfer function relating angular position to applied torque;

$$\frac{\theta}{\tau} = \frac{1}{J_L s^2} \qquad (4\text{-}2)$$

and the block representation:

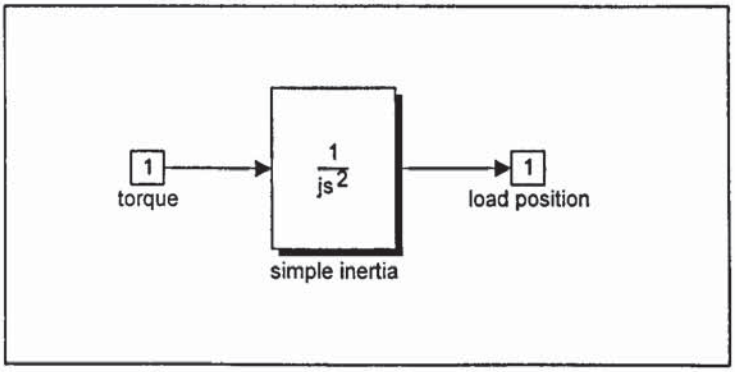

Figure 4-8 Block Representation of a Simple Inertia

In general, one can expect the load inertia to be a function of position (angle) rather than a function of time and this is assumed to be the case although simple inertia is used for the rest of the discussion for the sake of simplicity.

4.3.1 Servo Axis Driving a Simple Inertia

Consider a servo axis driving a simple load with inertia J_L. When the simulation model for such a system is constructed, the rotor inertia and the load inertia are usually lumped together as one item for simplicity, as in the transfer function

$$\frac{1}{(J_r + J_L)s^2} \tag{4-3}$$

However, for a flexible generic system model, it is desirable to separate these two inertias as they really are properties of distinct and replaceable physical parts. In other words, they are partial description of different classes of "objects". In order to reflect this modular nature of the physical hardware and establish a one-to-one correspondence between the hardware and the model components, a finite shaft stiffness (k_{shaft}) connecting the axis and the load is introduced.

As an example, in the block diagram shown below (Figure 4-9), the torque required to drive the load to the demanded position is represented as a (negative) external torque to the servo axis. As long as the stiffness is large, the shaft position and the load end position can be considered to be the same. The extra degree of freedom introduced will not manifest itself in the frequency range concerned. However, the extra number of states introduced may become a burden for computation and accuracy. Specifically, a very stiff shaft introduces two very short time constants into the system and thus degrades the efficiency of any simulation. A model order reduction procedure will have to be introduced to keep this under control, especially when a number of servo axes are involved.

Note also that the shaft motor end position rather than the load end position is taken as the shaft output position. This is because the encoder is mounted at the non drive end inside the motor, in the case of the servo (BRU200) under consideration.

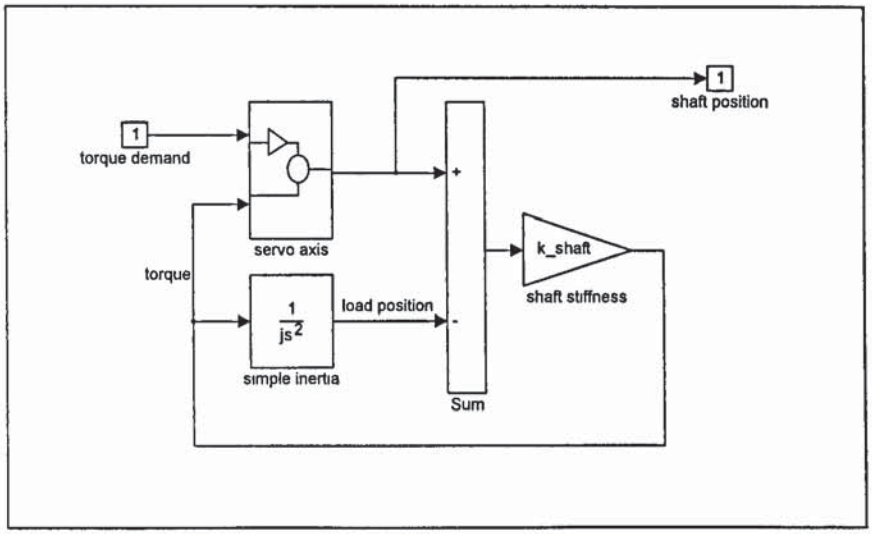

Figure 4-9 Servo Axis Driving a Simple Inertia

4.4 Multi-Input Multi-Output System

In the previous section, the basic structure of a SISO system was established. The next step is to consider a MIMO system. For illustration, consider a two-axis system with axis A and axis B each driving its own load of inertia J_A and J_B respectively. The corresponding stiffness of the shafts are k_A and k_B. The same basic structure as the SISO model is employed but with some of the scalar signal lines (thinner) replaced by vector signal lines (thicker). Multiplexers and demultiplexers are used to convert between scalar and vector signal lines as shown in the block diagram. Another layer of structure is introduced as now the servo, the load and the stiffness blocks each contain 2 components of the same type. This structure is shown in the block diagrams in Figure 4-10, Figure 4-11, Figure 4-12 & Figure 4-13.

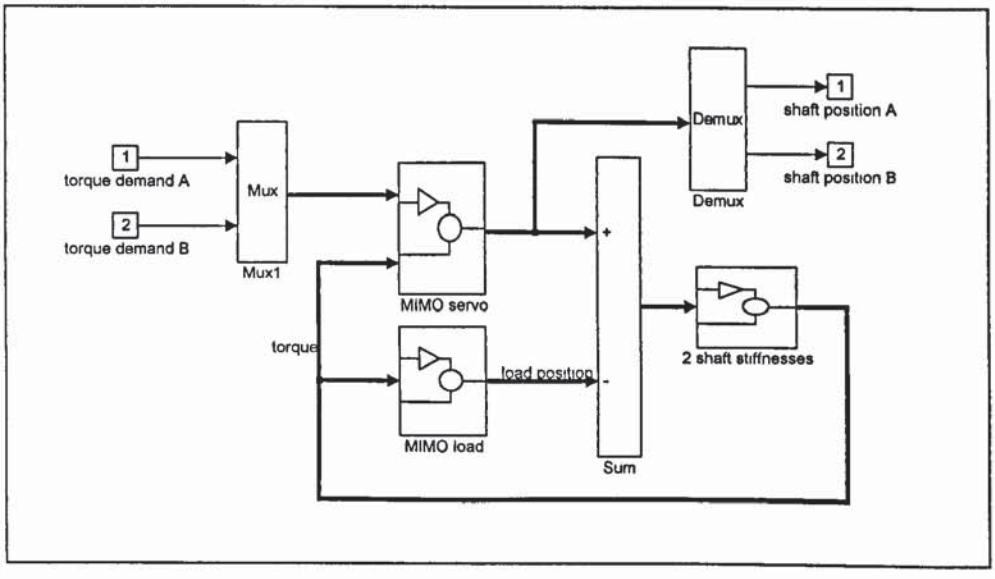

Figure 4-10 Example of a System with Two Parallel Axes

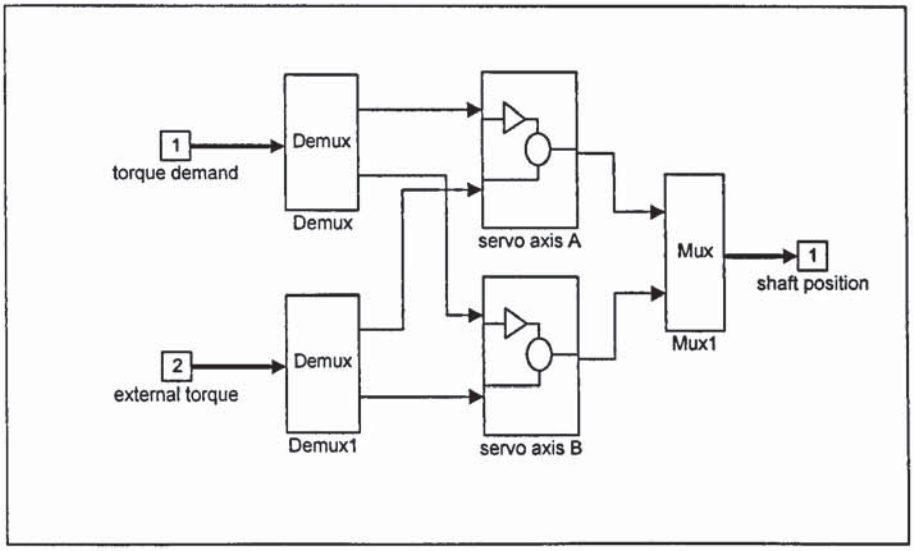

Figure 4-11 MIMO Servo Block with Two Parallel Axes

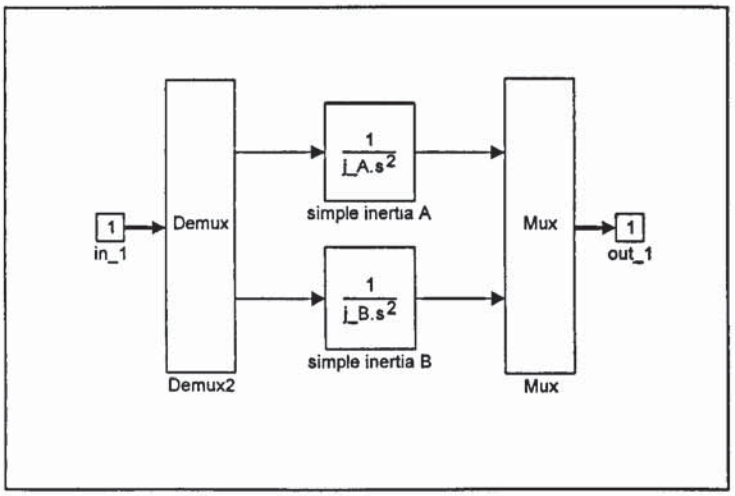

Figure 4-12 MIMO Load Block with Two Simple Loads

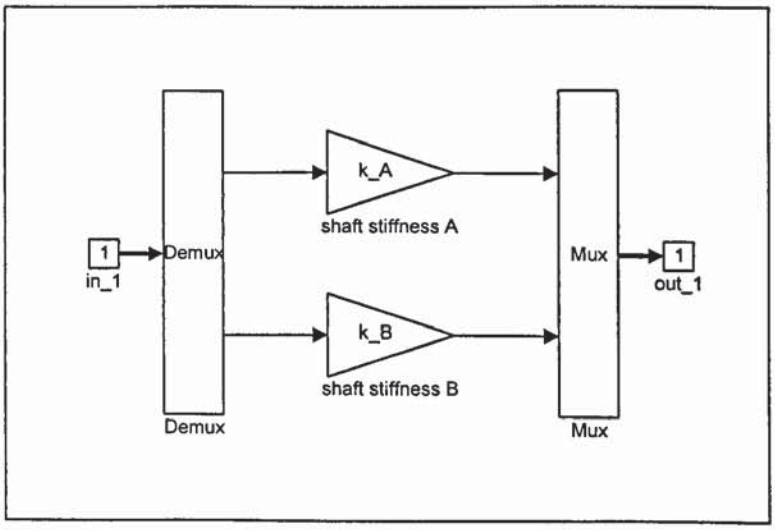

Figure 4-13 Shaft Stiffness Block for Two Shafts

4.5 Cross Axis Interactions -- Inherent Coupling

4.5.1 Classification of all Inherent Coupling

As explained in the previous chapter, in order to take advantage of multivariable control methods, a complete MIMO description of the plant is required. At the present time, most of the physical components are modelled as single-input single-output (SISO) dynamic systems under SISO control. The important piece of information that is missing in the description is the representation of the various types of coupling inherent to the type of machinery under consideration. In current practice, these interactions are treated as disturbance to the axes when they are being controlled independently. Inherent couplings between axes are usually undesirable interactions and can be compensated by suitable decoupling MIMO controllers.

Inherent couplings are any electrical or mechanical interactions which exist between any two axes on a machine with no controller action present. Four different types of inherent coupling have been identified. Two of these are found commonly in machines. The other two will be encountered in the near future, if the current trend of machine design prevails. [Tokuz & Rees Jones 91]

4.5.2 Structural Coupling

Consider a machine in which several motors are mounted on the same mechanical structure. Apart from the acceleration of the rotor, any changes in the air gap torque will also act on the stator. As the stator is mounted rigidly, it can be considered as part of the overall structure. The finite stiffness of the structure means that this reaction torque will in turn excite the mechanical resonance modes of the structure. This causes the stators of the other motors, which are mounted on the same structure, to deflect. Since rotor "positions" are measured as the difference between the angle of the stator and the angle of the rotor and the changes of air gap torque in one motor influences the stator angles of other motors, these motors are in effect coupled. The dynamics of the coupling will be that of the mechanical resonant modes of the backbone structure of the machine. Similarly, there are also cases where a motor is driving an eccentric load and transverse forces come to act on its bearing and hence couple to the other motors through the structure.

4.5.2.1 Mathematical Representation of Structural Coupling

A structure is completely characterised by its natural modes. Assume that a machine structure has L significant natural modes and associated resonance frequencies indexed by i. Often, these mode shapes can be estimated reasonably well from experience. Furthermore, reasonable estimates of the resonance frequencies can be obtained by performing a Ritz type calculation.

Each motor has a stator that is effectively part of the mechanical structure. Let there be N motors on a structure indexed by the variable n. Each mode shape will be characterised by a set of N angles. Note that although mode shapes can be arbitrary scaled and still remain mode shapes, there is a normalisation procedure, which is extremely useful for providing a reference scaling. This normalisation procedure is called 'mass normalisation'. In a mass normalised mode shape:

$$\iiint \rho(x,y,z)\cdot\left(\delta_x^2+\delta_y^2+\delta_z^2\right) dx\cdot dy\cdot dz = 1 \qquad (4\text{-}4)$$

where ρ is density and δ's are particle translations in the x, y, z directions and the integration is taken over the entire structure. Thus, assume that we can estimate L mass-normalised modal vectors U_i, with i=1...L and each of these vectors has N components. The resonance frequencies are ω_{ni} with i=1...L. In general, ω_{ni} are the solutions to the differential equation:

$$J\ddot{\theta} = -K\theta \qquad (4\text{-}5)$$

where J and K are the inertia and stiffness matrices of the structure and θ is the angle vector with respect to the appropriate frame of reference. A modal matrix U can be formed by collating the modal vectors U_i

$$U = [U_1, U_2, \cdots, U_i, \cdots, U_L] \qquad (4\text{-}6)$$

With each U_i corresponding to its characteristic value $\lambda_i=\omega_{ni}^2$. Together, a diagonal matrix Λ can be formed

$$\Lambda = \begin{bmatrix} \omega_{n1}^2 & 0 & \cdots & 0 \\ 0 & \omega_{n2}^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \omega_{nL}^2 \end{bmatrix}$$

(4-7)

The response of the structure to a set of N torques is a vector of angle

$$v(t) = U \cdot p(t)$$

(4-8)

Here p(t) represents the instantaneous 'amount' of each mode that is in operation (angle in modal direction) and this is determined by

$$\ddot{p}(t) = P(t) - \Lambda \cdot p(t)$$

(4-9)

where P(t) is the net instantaneous modal 'force' acting on each mode and

$$P(t) = U^T \cdot V(t)$$

(4-10)

relates P(t) to the actual torques V(t) applied to the structure. Combining the above expressions leads to

$$v = U \cdot \left(\Lambda + s^2 I\right)^{-1} U^T \cdot V$$

(4-11)

which is shown as an equivalent block diagram in Figure 4-14. It is relatively easy to measure or estimate both the natural frequencies and the mass normalised modes of a structure.

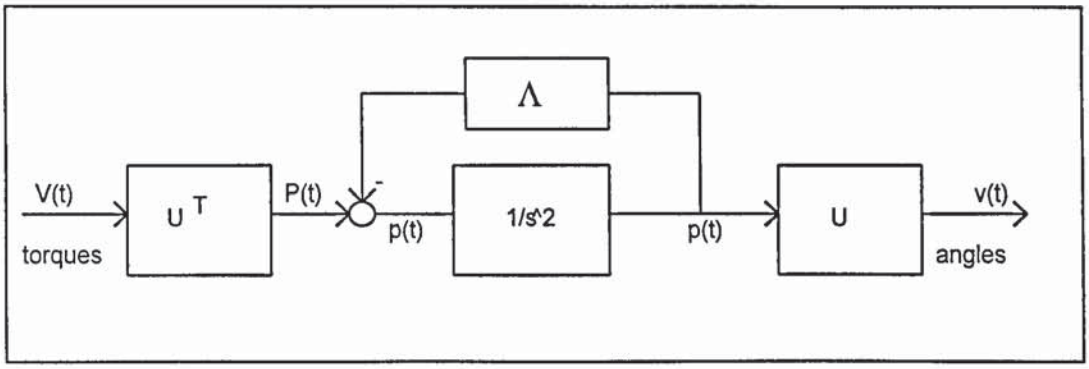

Figure 4-14 Modal Transformation

4.5.2.2 Incorporating Structural Coupling

As shown in the previous section, the modal transformation is a formulation which can be utilised to relate the airgap torques of the motors to the angles or positions of their stators via the natural frequencies of the structure. To introduce this formulation, the model of the servo axis is modified such that apart from the inertia of the rotor (J_r), the inertia of the stator (J_{st}) is also present (Figure 4-15). The airgap torque generated in the motor is now allowed to drive both "inertias" (in opposite directions). Accordingly, the MIMO servo block has to be modified as shown in Figure 4-16. In order to keep the physical components as separate entities in the model, a fictitious coupling with very high stiffness between the stators and the support structure is introduced in the overall generic system model, as shown in Figure 4-17. With reference to Figure 4-14, the mechanics of the structural coupling is encapsulated in the support structure block which is shown in details shown in Figure 4-18.

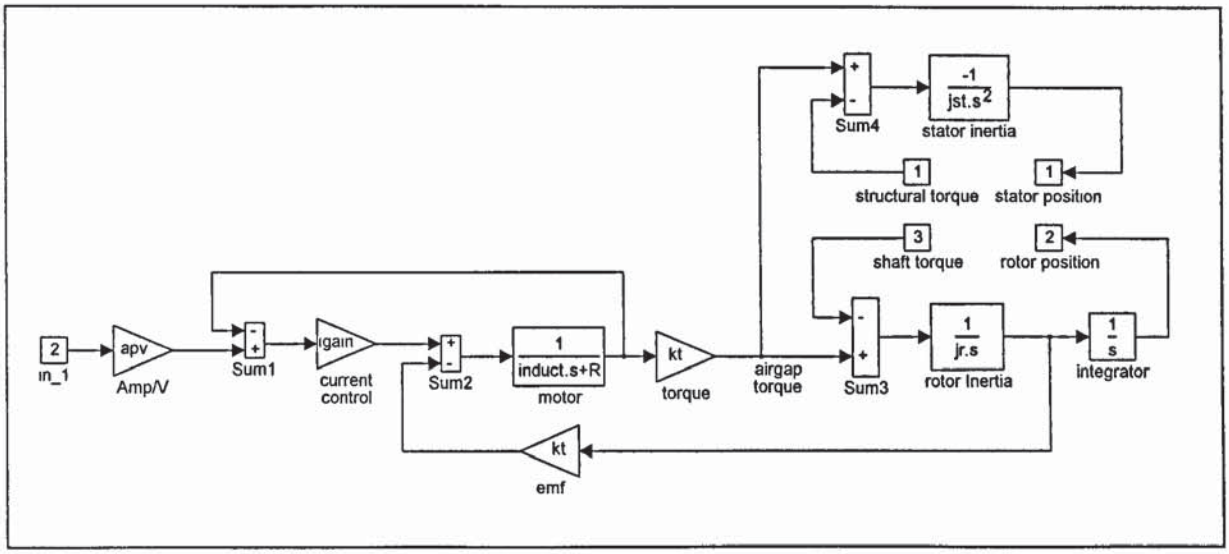

Figure 4-15 Modified Servo Model

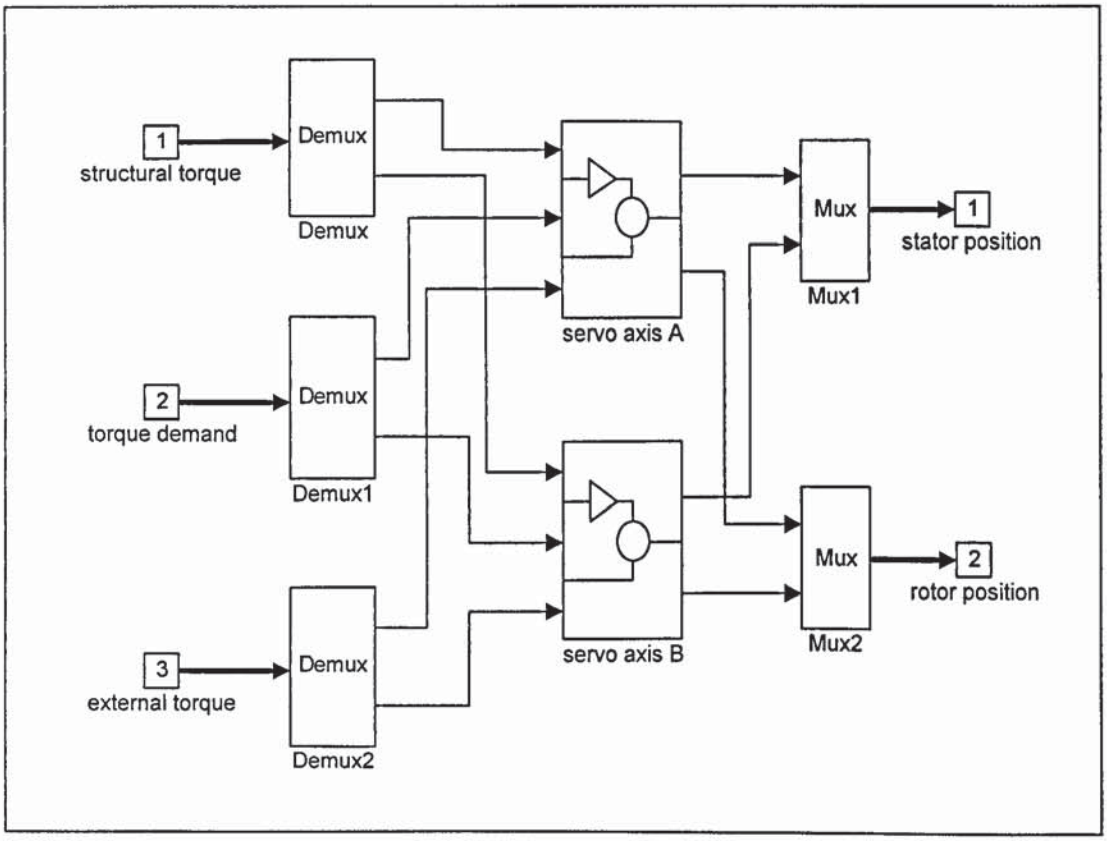

Figure 4-16 Modified MIMO Servo Block

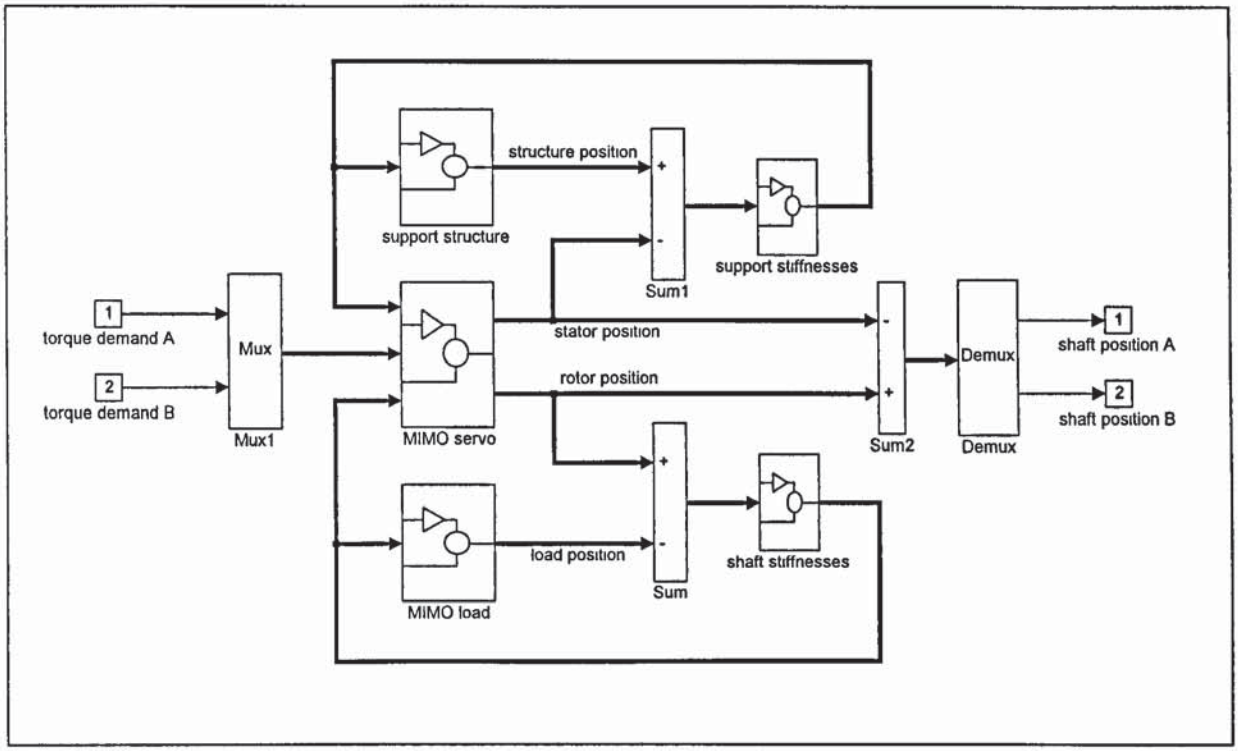

Figure 4-17 Generic System Model Feature Structural Coupling

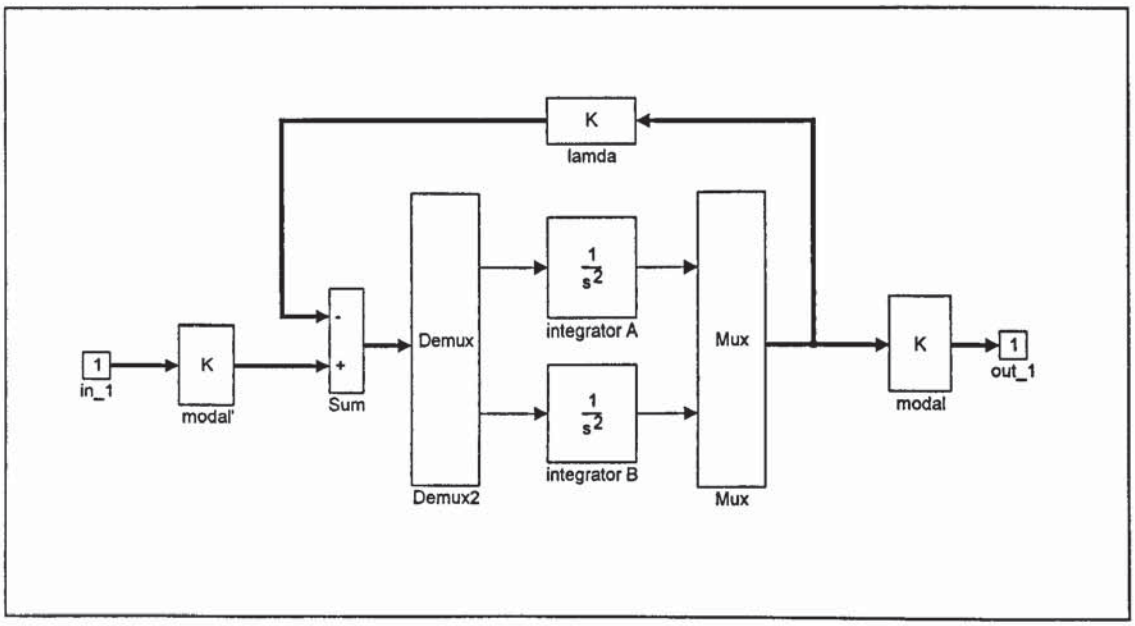

Figure 4-18 Support Structure Block

4.5.2.3 Example : 3 Motors on a Flexible Platform

As part of the MIMO test rig (Figure 4-19, details in appendix), 3 identical motors are mounted on a flexible platform that is supported at 4 points with small rubber blocks.

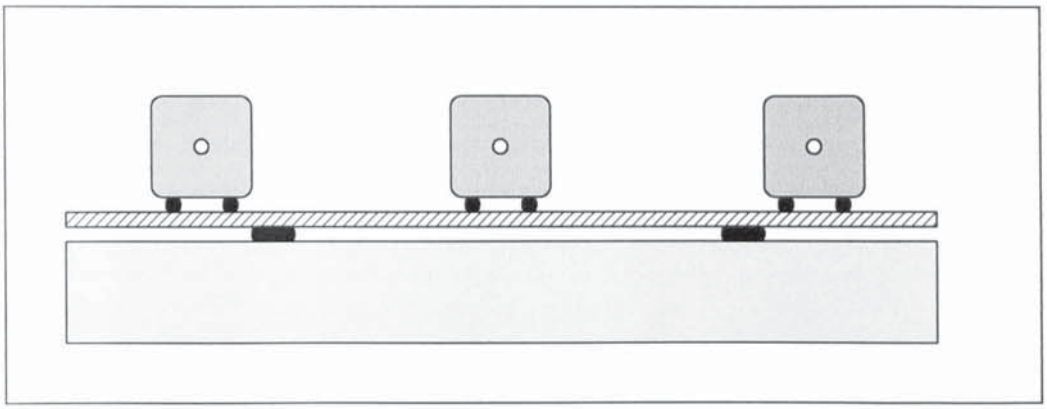

Figure 4-19 Motors on Flexible Platform

4.5.2.3.1 Estimation of the Resonance Frequencies

Each motor weighs around 4 kg, the stator inertia can be taken as $8x10^{-3}$ kgm^2 and the platform has a mass of 25 kgm^{-2}. With the aid of a finite element analysis on the mode shapes (Figure 4-20), the frequencies are estimated as around 7.2, 17.2 & 30.0 Hz with respect to the first 3 relevant modes.

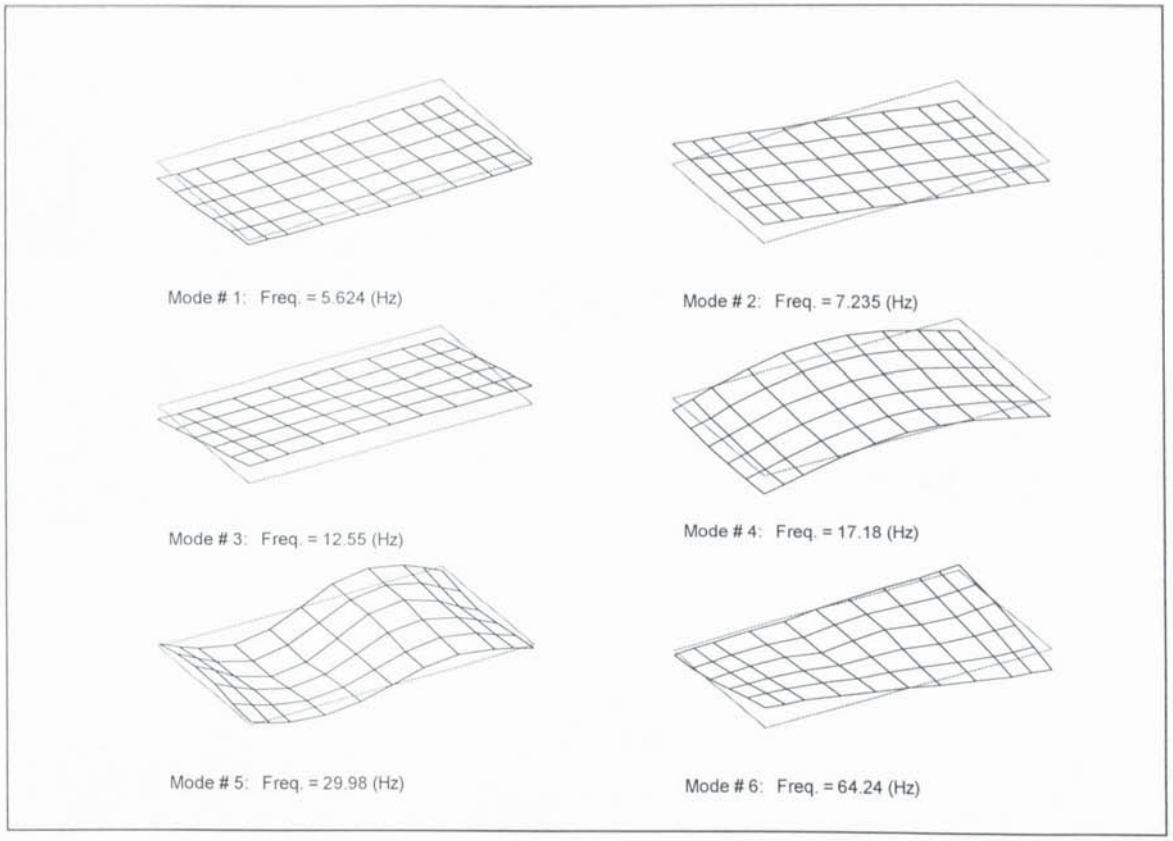

Figure 4-20 Mode Shapes of the MIMO Test Rig

4.5.2.3.2 Measurement of the Resonance Frequencies

To measure the resonant frequencies, the 3 motors are set to run at the same constant speed (80 counts per ms or 10 Hz). On one of the motors, a disturbance of the form of an impulse function is added and the responses are recorded. The power spectra of the 3 axes, measured in arbitrary units against frequency (Hz), are shown below. In Figure 4-21, the impulse is applied to the first channel and a peak of about 7 Hz can be observed. The middle channel has a spike at 30 Hz while the last channel has spikes at both 10 and 17 Hz which corresponds to the motor speed and the 4th resonant mode. In Figure 4-22, the impulse is applied to the middle channel which induced a wide spread of frequencies in the power spectra. This can be explained by its relative rigidity with respect to the structure (Figure 4-19). Disregarding the 10 Hz spikes due to the motor speed, the resonant of 7 and 30 Hz can again be observed from the two outside channels.

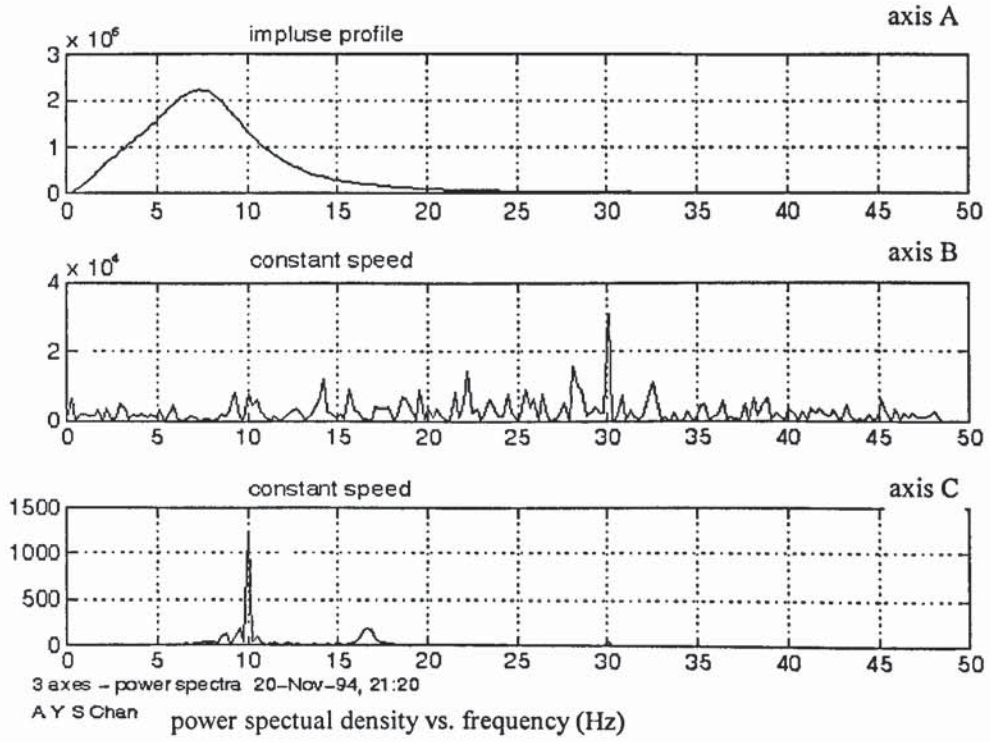

Figure 4-21 Power Spectra of Motors on Flexible Platform -- Impulse in First Channel

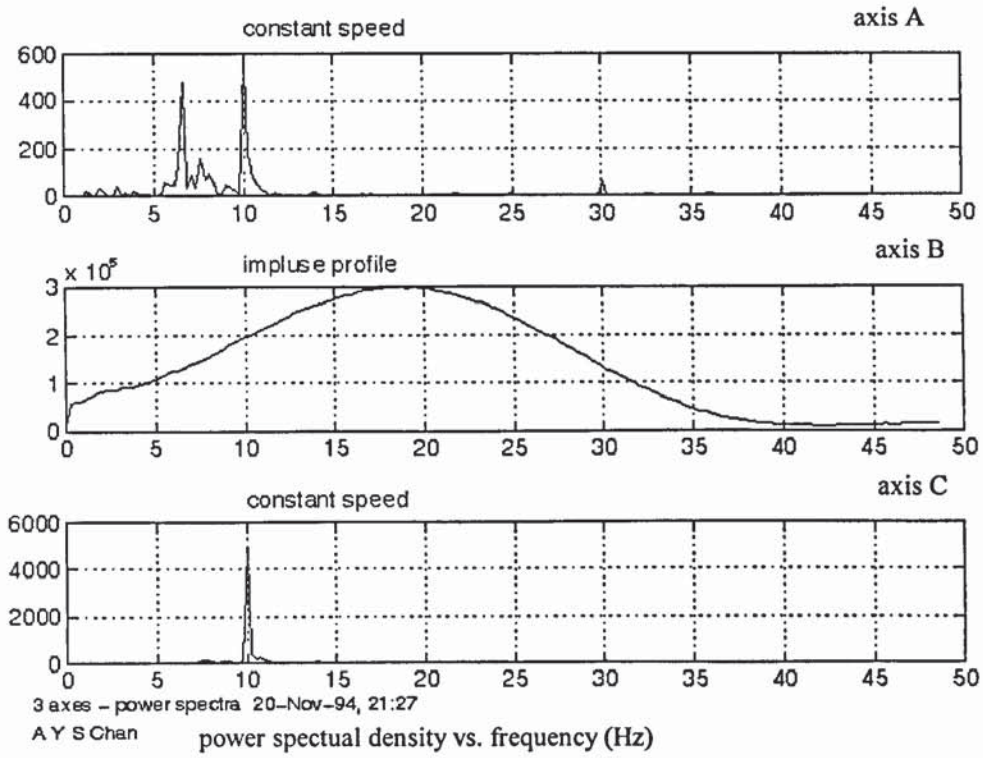

Figure 4-22 Power Spectra of Motors on Flexible Platform -- Impulse in Second Channel

4.5.3 Web Coupling

On a typical multi-axis machine, very often, a continuous flow of material will be processed by a number of mechanisms driven by the corresponding axes. This web of material thus interacts with a number of shafts and provides a mean of coupling between these shafts. Assuming that the web has negligible mass, is under tension and no slippage occurs when it runs through the machine, the interaction can easily be incorporated into the model by a simple feedback matrix. Consider the examples in Figure 4-23:

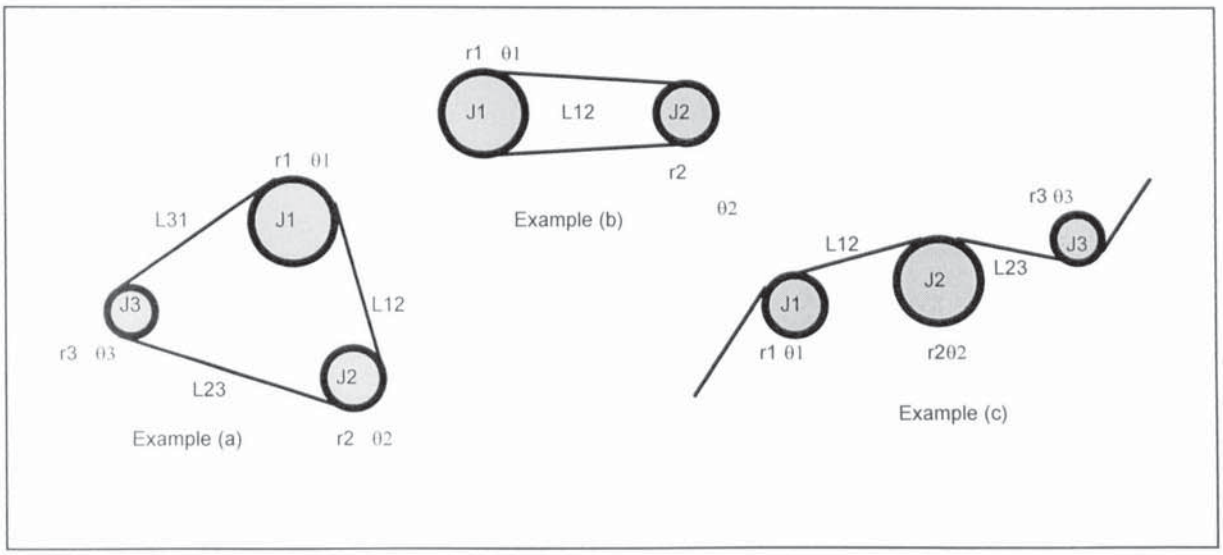

Figure 4-23 Example of Different Web Configurations

In each case the J's represent the total inertia on that axis, the *L*'s represent the lengths of the web segments and the r's and θ's represent the equivalent radii and the angles of the mechanisms. Assign a stiffness of K_{web} to the web material, the following Newtonian equations can be derived for example (a):

$$J_1\ddot{\theta}_1 = \tau_1 - K_{web}r_1\left(\frac{r_1\theta_1 - r_2\theta_2}{L_{12}} + \frac{r_1\theta_1 - r_3\theta_3}{L_{31}}\right)$$

$$J_2\ddot{\theta}_2 = \tau_2 - K_{web}r_2\left(\frac{r_2\theta_2 - r_1\theta_1}{L_{12}} + \frac{r_2\theta_2 - r_3\theta_3}{L_{23}}\right)$$

$$J_3\ddot{\theta}_3 = \tau_3 - K_{web}r_3\left(\frac{r_3\theta_3 - r_1\theta_1}{L_{31}} + \frac{r_3\theta_3 - r_2\theta_2}{L_{23}}\right)$$

(4-12)

and put into matrix and Laplacian notation:

$$Js^2\theta = \tau - K_{web} \cdot H \cdot \begin{bmatrix}\theta_1\\ \theta_2\\ \theta_3\end{bmatrix}$$

(4-13)

or in terms of a block diagram

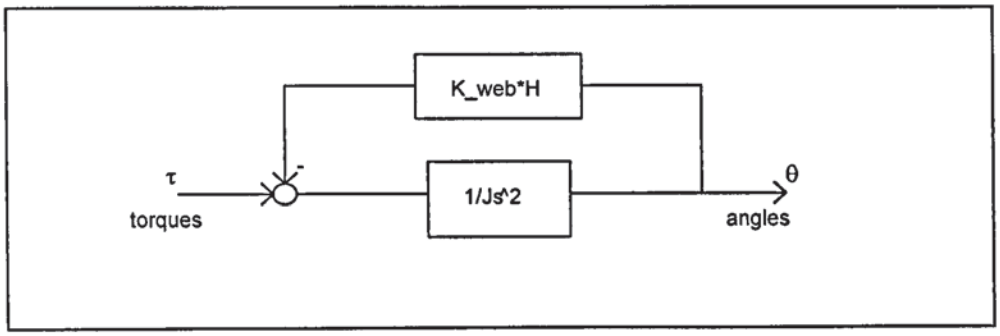

Figure 4-24 Web Coupling as Stiffness Feedback

in which

$$H = \begin{bmatrix} r_1 & 0 & 0 \\ 0 & r_2 & 0 \\ 0 & 0 & r_3 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{L_{12}} + \frac{1}{L_{31}} & -\frac{1}{L_{12}} & -\frac{1}{L_{31}} \\ -\frac{1}{L_{12}} & \frac{1}{L_{12}} + \frac{1}{L_{23}} & -\frac{1}{L_{23}} \\ -\frac{1}{L_{31}} & -\frac{1}{L_{23}} & \frac{1}{L_{31}} + \frac{1}{L_{23}} \end{bmatrix} \cdot \begin{bmatrix} r_1 & 0 & 0 \\ 0 & r_2 & 0 \\ 0 & 0 & r_3 \end{bmatrix} \tag{4-14}$$

Similar derivation shows that H takes the following values in example (b)

$$H = \frac{2}{L_{12}} \begin{bmatrix} r_1 & 0 \\ 0 & r_2 \end{bmatrix} \cdot \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} r_1 & 0 \\ 0 & r_2 \end{bmatrix} \tag{4-15}$$

And for example (c)

$$H = \begin{bmatrix} r_1 & 0 & 0 \\ 0 & r_2 & 0 \\ 0 & 0 & r_3 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{L_{12}} & -\frac{1}{L_{12}} & 0 \\ -\frac{1}{L_{12}} & \frac{1}{L_{12}} + \frac{1}{L_{23}} & -\frac{1}{L_{23}} \\ 0 & -\frac{1}{L_{23}} & \frac{1}{L_{23}} \end{bmatrix} \cdot \begin{bmatrix} r_1 & 0 & 0 \\ 0 & r_2 & 0 \\ 0 & 0 & r_3 \end{bmatrix} \tag{4-16}$$

The matrix in the feedback path has entries depending only on the material stiffness and the geometry of the web - that is which and in what order the web goes around the shafts. At the moment, only static stiffness is considered but dynamic stiffness can easily be included when necessary.

To incorporate web coupling into the generic system model is just a matter of putting the right feedback matrix between the output position (load end of the shaft) and the shaft torque. The MIMO load block is modified as in Figure 4-25:

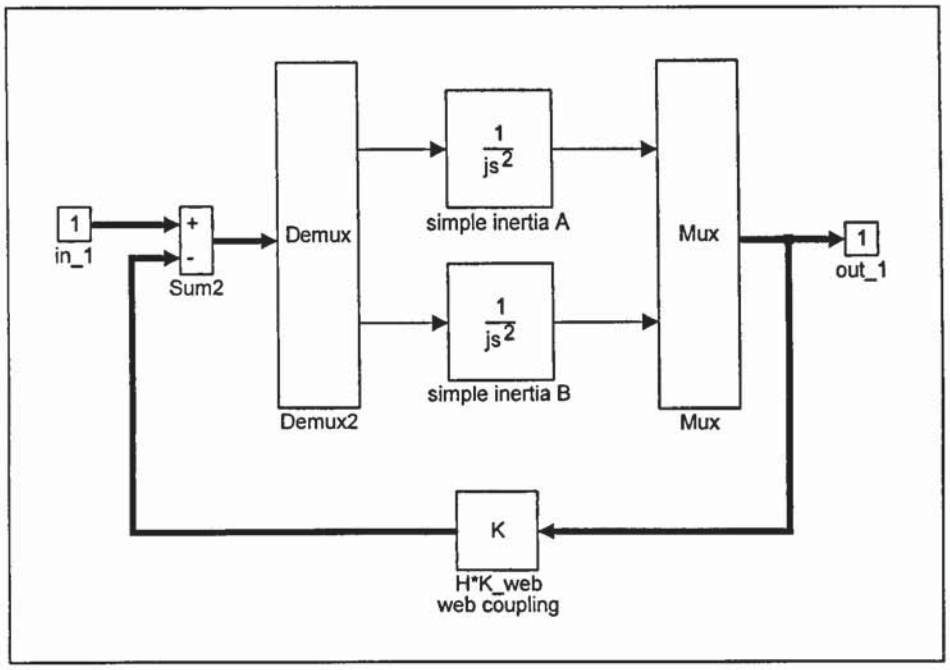

Figure 4-25 MIMO Load Block with Web Coupling

4.5.4 DC bus coupling

Consider a situation where a number of servo axes draw their power from a common DC bus. When one of the motors increases its airgap torque, the increased demand of current causes an instantaneous drop of voltage on the DC bus. The PWM inverter in the drive produces a current gain that is proportional to the DC bus voltage. As a result, errors occur on other drives if the motors are turning or if current is flowing through them. A finite amount of time will pass before these errors are recovered. Thus this is another mechanism by which the servo axes will affect each other. Assuming a certain constant impedance DC supply, at any moment the actual supply will be V_{dc} volt which is

$$V_{dc} = DC - R_{bus} \sum_{n} i_n \qquad (4\text{-}17)$$

where i_n is the current demand of the n'th axis and R_{bus} is the impedance of the DC bus. Accordingly, a DC coupling block is constructed and the servo model is modified. The block diagrams shown in Figure 4-26, Figure 4-27 & Figure 4-28 are an example of a 2

axis system. This mode of coupling has been assigned with a low priority and is not tested on the experimental rig.

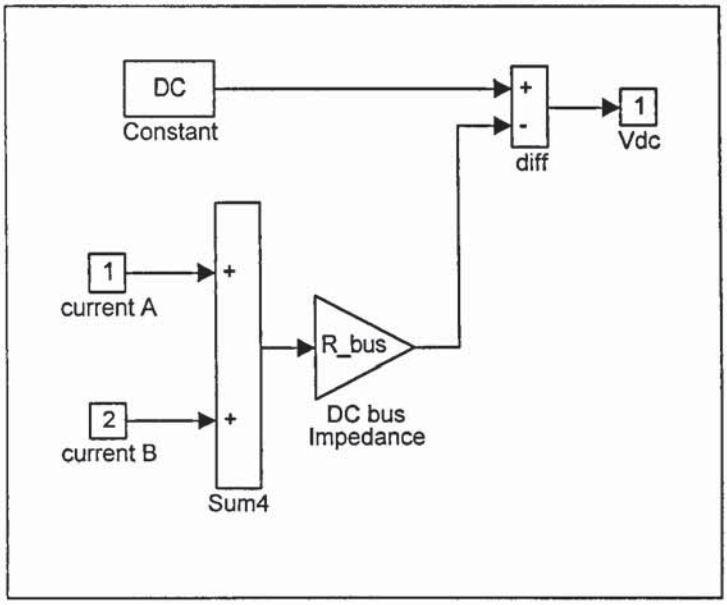

Figure 4-26 DC Coupling Block

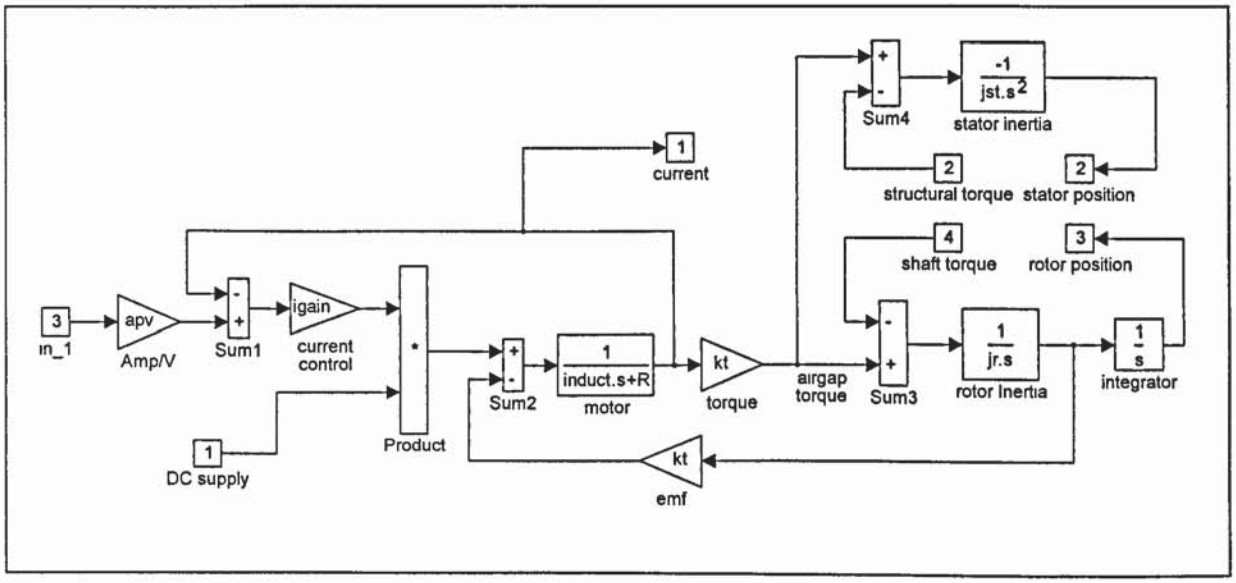

Figure 4-27 Servo Model with DC Bus Coupling

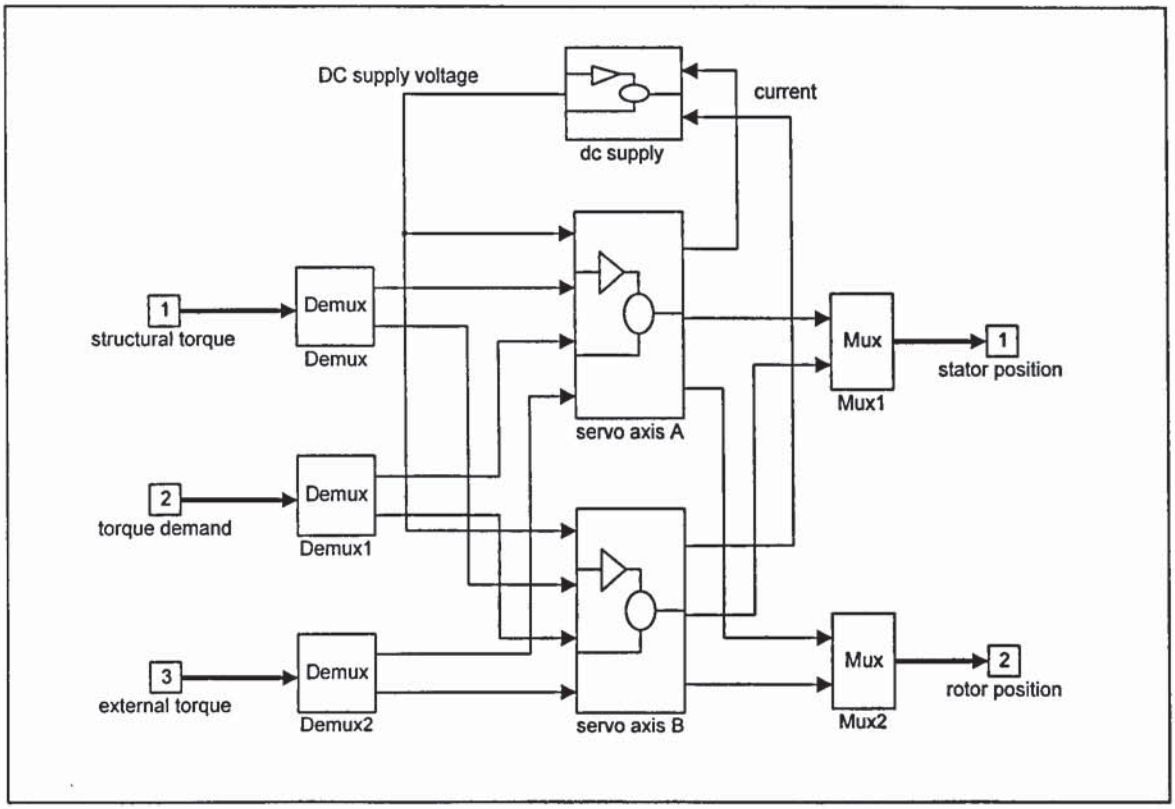

Figure 4-28 MIMO Servo Block with DC Coupling

4.5.5 Hybrid Coupling

Hybrid coupling is the mechanical equivalent of DC bus coupling. Instead of a common DC bus, the axes are powered by a hybrid mechanical transmission. This coupling happens when the primary power source for a multiplicity of hybrid mechanisms is loaded by one and this causes an error in the others connected to that shaft. As such a facility is not currently available to the project, a lower priority is assigned and it is not currently implemented. However, modification can easily be made to accommodate this in the model.

4.6 Cross Axes Interaction -- Imposed Coupling

4.6.1 Synchronisation Requirements

Inherent couplings are any electrical or mechanical interactions which exist between any two axes on a machine in the absence of closed loop control. These are undesirable interactions and can be compensated by suitable decoupling MIMO controllers. Often, a number of isolated (independently driven) axes are required to be synchronised electronically in order to perform certain tasks. Hence, the synchronisation requirement

implies some "desirable" cross axes interaction has to be imposed on the system. There are several ways to achieve synchronisation. For example:

- All axes are slaved to the same software master and synchronisation is ensured by holding the absolute errors of these axes within bounds. This strategy is unnecessarily stringent and there is no information exchange between axes.
- Assigning the slowest loop to be the master and the second slowest loop to be the first follower. The third slowest loop will be second follower and subsequently faster loops then follow one after another. Such master/follower configuration makes use of a one-way information flow but stability is not guaranteed as the phases of the followers accumulated.
- Full MIMO control that involves two ways information exchange between axes.

Note that between any two axes, there can be both (undesired) inherent coupling and synchronisation requirement and their effects can be enhancing or counteracting each other.

4.6.2 Imposed Coupling

The principle of our particular decentralised control approach is to start with the best achievable theoretical solution (full MIMO control for the overall system). The performance requirements are then relaxed such that some of the cross axes interaction can be ignored. The resultant system can then be considered as a collection of smaller sub-systems with lesser degree of complexity that can be dealt with fewer difficulties.

Consider a multi-axis plant which is represented by a transfer function matrix with its diagonal terms representing the axes local dynamics and the off-diagonal terms representing the inherent couplings. The performance specification can be translated to the target response of the overall system. However, embedded in this specification will be *both* the synchronisation requirements and the tolerated level of inherent coupling.

To achieve the target response, a full MIMO controller can be designed which will represent the best achievable theoretical solution. The diagonal terms of such a controller will be control loops closed locally for each of the axes. The off-diagonal terms will however represent the cross axes coupling needed to be imposed to achieve the various

performance requirements. Therefore, imposed coupling is the interaction necessary to ensure synchronisation and to compensate for the undesirable inherent coupling, simultaneously. Note that the plant is taken as a whole and remains intact, only the controller is decentralised.

4.7 Axes to Shafts Distribution

In general, the mechanism driven by an axis has more than one turning shaft. In most cases, these shafts are coupled by various gears and synchronised in the event cycles and the torque demand presented to the servo axis depends only on the angle of the axes. However, when these shafts are not synchronised to the event cycle, the power demand on the servos will depend on the positions of the individual shafts. A mechanism has to be provided to model these dynamics. Consider an example (Figure 4-29) of a system with 2 servo axes (denoted by A and B) driving 5 active shafts. Each active shaft in turns drives its local load. The dynamics of the 2 servo axes and the 5 loads are already modelled in the MIMO servo and MIMO load blocks. To make the connection, a torque distribution block is introduced. This distribution block represents a conceptual component with 2 input shafts (of inertia J_{ip1} and J_{ip2}) and 5 output shafts (of inertia J_{op1} ... J_{op5}) and it captures the torque and angular position relationship of these 7 shafts.

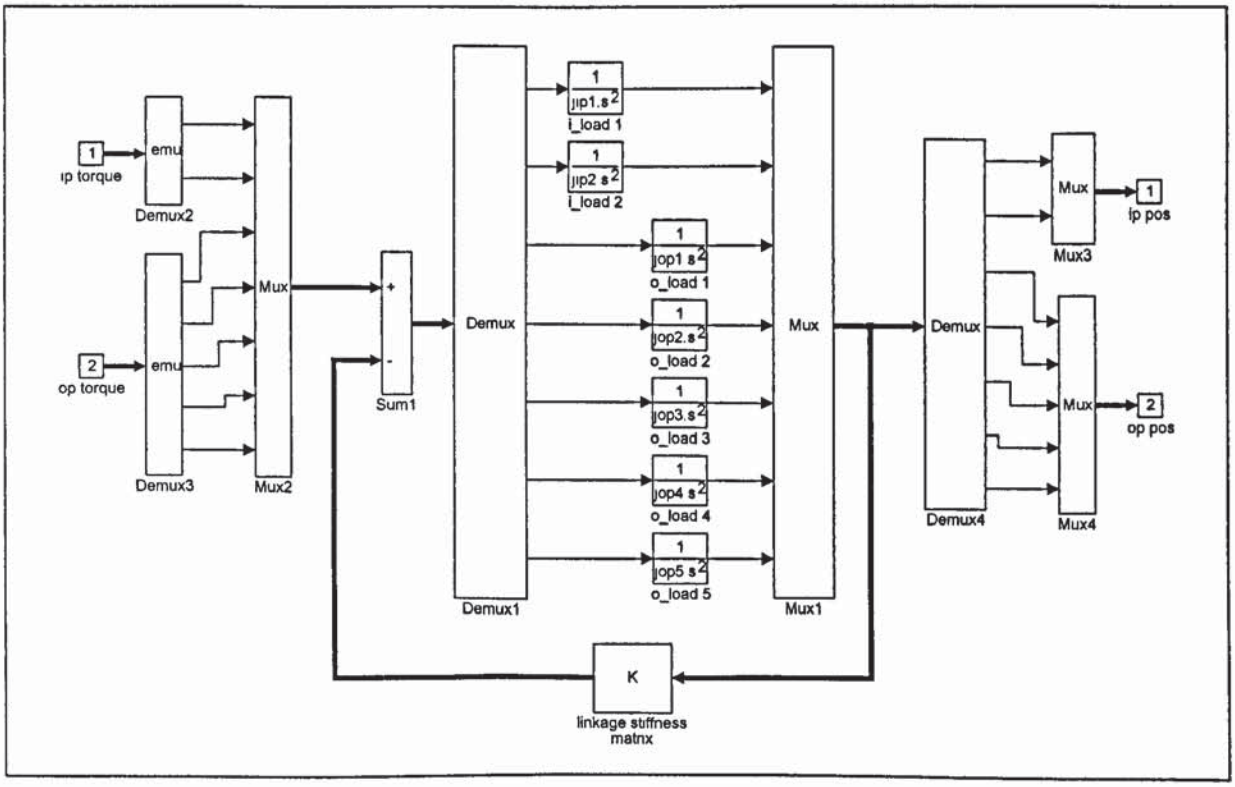

Figure 4-29 Torque Distribution Block for 2 Axes 5 Active Shafts

Analogous to the incorporation of web coupling, the linkages between these shafts are represented by a feedback matrix which depends only on the configuration and the stiffness of the linkage. To develop the example further, assume the model has a configuration as shown in Figure 4-30:

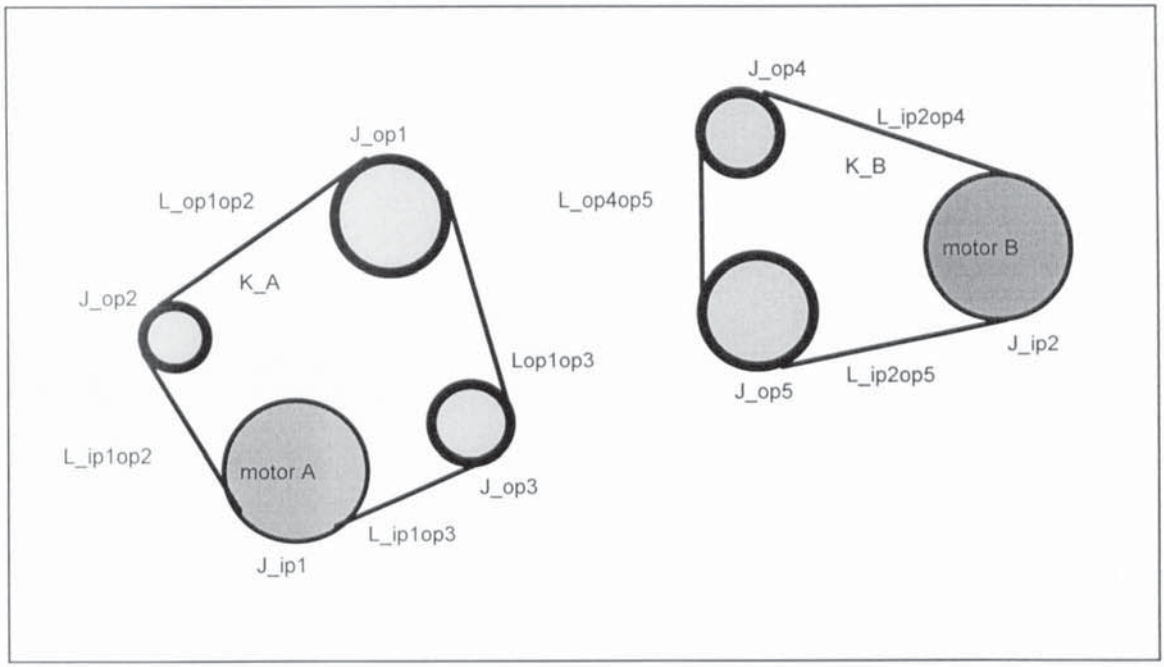

Figure 4-30 2-Motors Driving 5-Inertia

In this configuration, motor A powers the shafts 1, 2 and 3 by means of a belt of stiffness K_A, while motor B powers shaft 4 and 5 with a belt of stiffness K_B. The following Newtonian equations can be written:

$$J_{ip1}\ddot{\theta}_{ip1} = \tau_{ip1} - K_A r_{ip1}\left(\frac{r_{ip1}\theta_{ip1} - r_{op2}\theta_{op2}}{L_{ip1op2}} + \frac{r_{ip1}\theta_{ip1} - r_{op3}\theta_{op3}}{L_{ip1op3}}\right)$$

$$J_{ip2}\ddot{\theta}_{ip2} = \tau_{ip2} - K_B r_{ip2}\left(\frac{r_{ip2}\theta_{ip2} - r_{op4}\theta_{op4}}{L_{ip2op4}} + \frac{r_{ip2}\theta_{ip2} - r_{op5}\theta_{op5}}{L_{ip2op5}}\right)$$

$$J_{op1}\ddot{\theta}_{op1} = \tau_{op1} - K_A r_{op1}\left(\frac{r_{op1}\theta_{op1} - r_{op2}\theta_{op2}}{L_{op1op2}} + \frac{r_{op1}\theta_{op1} - r_{op3}\theta_{op3}}{L_{op1op3}}\right)$$

$$J_{op2}\ddot{\theta}_{op2} = \tau_{op2} - K_A r_{op2}\left(\frac{r_{op2}\theta_{op2} - r_{ip1}\theta_{ip1}}{L_{ip1op2}} + \frac{r_{op2}\theta_{op2} - r_{op1}\theta_{op1}}{L_{op1op2}}\right)$$

$$J_{op3}\ddot{\theta}_{op3} = \tau_{op3} - K_A r_{op3}\left(\frac{r_{op3}\theta_{op3} - r_{ip1}\theta_{ip1}}{L_{ip1op3}} + \frac{r_{op3}\theta_{op3} - r_{op1}\theta_{op1}}{L_{op1op3}}\right)$$

$$J_{op4}\ddot{\theta}_{op4} = \tau_{op4} - K_B r_{op4}\left(\frac{r_{op4}\theta_{op4} - r_{ip2}\theta_{ip2}}{L_{ip2op4}} + \frac{r_{op4}\theta_{op4} - r_{op5}\theta_{op5}}{L_{op4op5}}\right)$$

$$J_{op5}\ddot{\theta}_{op5} = \tau_{op5} - K_B r_{op5}\left(\frac{r_{op5}\theta_{op5} - r_{ip2}\theta_{ip2}}{L_{ip2op5}} + \frac{r_{op5}\theta_{op5} - r_{op4}\theta_{op4}}{L_{op4op5}}\right)$$

(4-18)

in which the J's, θ's and τ's are the inertia, positions and torques of the respective shafts, the *r*'s are the (equivalent) radii of pulleys acted as linkages and the *L*'s are the web segment lengths. In matrix and Laplacian notation, this becomes:

$$Js^2 \cdot \begin{bmatrix} \theta_{ip1} \\ \hdashline \theta_{ip2} \\ \hdashline \theta_{op1} \\ \theta_{op2} \\ \theta_{op3} \\ \hdashline \theta_{op4} \\ \theta_{op5} \end{bmatrix} = \tau - K_{linkage} \cdot \begin{bmatrix} \theta_{ip1} \\ \hdashline \theta_{ip2} \\ \hdashline \theta_{op1} \\ \theta_{op2} \\ \theta_{op3} \\ \hdashline \theta_{op4} \\ \theta_{op5} \end{bmatrix}$$

(4-19)

in which the linkage stiffness matrix is

$$K_{linkage} = \left[\begin{array}{cc:ccccc} r_{ip1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & r_{ip2} & 0 & 0 & 0 & 0 & 0 \\ \hdashline 0 & 0 & r_{op1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & r_{op2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & r_{op3} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & r_{op4} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & r_{op5} \end{array}\right] \cdot \left[\begin{array}{c:c} K_{11} & K_{12} \\ \hdashline K_{12}' & K_{22} \end{array}\right] \cdot \left[\begin{array}{cc:ccccc} r_{ip1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & r_{ip2} & 0 & 0 & 0 & 0 & 0 \\ \hdashline 0 & 0 & r_{op1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & r_{op2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & r_{op3} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & r_{op4} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & r_{op5} \end{array}\right]$$

(4-20)

with

$$K_{11} = \begin{bmatrix} K_A\left(\frac{1}{L_{ip1op2}} + \frac{1}{L_{ip1op3}}\right) & 0 \\ 0 & K_B\left(\frac{1}{L_{ip2op4}} + \frac{1}{L_{ip2op5}}\right) \end{bmatrix}$$

(4-21)

$$K_{12} = \left[\begin{array}{ccc:cc} 0 & -\frac{K_A}{L_{ip1op2}} & -\frac{K_A}{L_{ip1op3}} & 0 & 0 \\ \hdashline 0 & 0 & 0 & -\frac{K_B}{L_{ip2op4}} & -\frac{K_B}{L_{ip2op5}} \end{array}\right]$$

(4-22)

$$K_{22} = \left[\begin{array}{ccc:cc} K_A\left(\frac{1}{L_{op1op2}} + \frac{1}{L_{op1op3}}\right) & -\frac{K_A}{L_{op1op2}} & -\frac{K_A}{L_{op1op3}} & 0 & 0 \\ -\frac{K_A}{L_{op1op2}} & K_A\left(\frac{1}{L_{ip1op2}} + \frac{1}{L_{op1op2}}\right) & 0 & 0 & 0 \\ -\frac{K_A}{L_{op1op3}} & 0 & K_A\left(\frac{1}{L_{ip1op3}} + \frac{1}{L_{op1op3}}\right) & 0 & 0 \\ \hdashline 0 & 0 & 0 & K_B\left(\frac{1}{L_{ip2op4}} + \frac{1}{L_{ip2op5}}\right) & -\frac{K_B}{L_{op4op5}} \\ 0 & 0 & 0 & -\frac{K_B}{L_{op4op5}} & K_B\left(\frac{1}{L_{ip2op4}} + \frac{1}{L_{ip2op5}}\right) \end{array}\right]$$

(4-23)

The axes to shafts linkage arrangement can easily be modified by changing this matrix. Finally, analogous to the coupling of the MIMO servo and MIMO load block, a further (output) shaft stiffness matrix is introduced so that the distribution block can be inserted in between the MIMO servo and load blocks. As shown in Figure 4-31, the product of the

input shaft stiffness and the difference between rotor and input shaft positions gives the input shaft torques, which are fed back to both the MIMO servo and the distribution block. The product of the output shaft stiffness and the difference between load end and output shaft positions gives the output shaft torques, which are fed back to both the MIMO load and the distribution block.

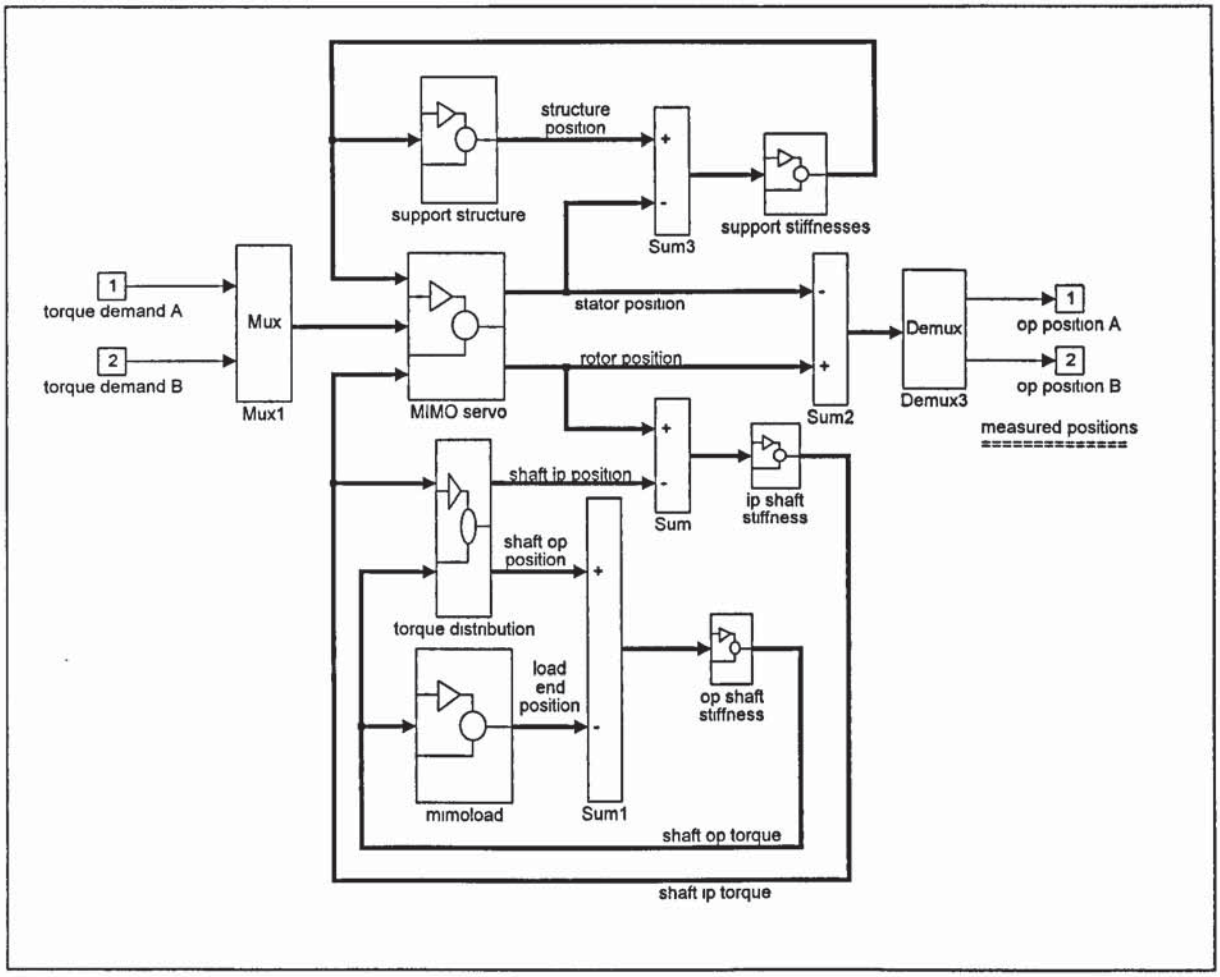

Figure 4-31 Redistribution of Torque at Shaft Level

4.7.1 Non Square System Model

For most applications, the use of off-the-shelf servo system meant that inputs and outputs to the system are brought into the design one pair at a time and hence the number of inputs equals to the number of outputs in most machines. However, it can be envisaged that for some critical applications, more sensors (both encoders and others) will be needed to monitor closely the state of the machine. In such cases, the number of outputs or signals to be fed back, will be larger than the number of inputs and the machine will have a non-square model.

For example, consider the 2-axis 5-shafts model from the pervious section. Instead of using the encoders on the 2 rotors, measurements are made directly with external encoders on the 5 active shafts (as shown in Figure 4-32). This system is obviously non square and the fact that there are more degrees of freedom at the outputs than the inputs means controllability may be a problem.

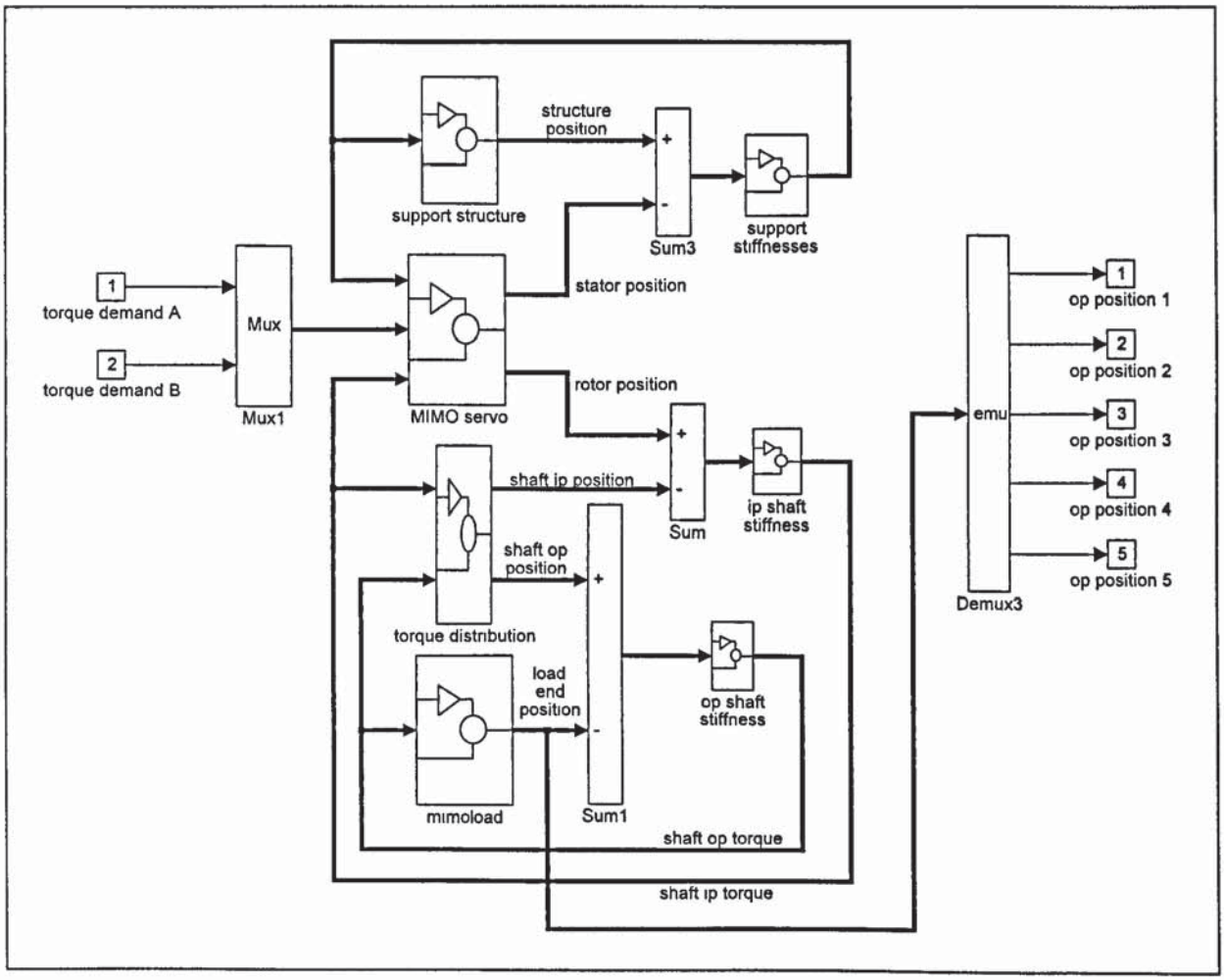

Figure 4-32 Non-Square System Model with Position Sensors on Shaft Level

4.8 Generic System Model in State Space Representation

Essentially, a model constructed in SIMULINK is stored as a set of connected transfer function blocks, which is a graphical representation. When a dynamic simulation is called for, a selection of integration routines is available to trace the signals and solve the set of equivalent differential equations. However, for further design and analysis work, the model has to be put into a proper mathematical representation such as state space equations or matrix transfer functions.

In general, the machines under consideration are complex and non-linear in nature. However, for simplicity and for the application of the most commonly available design and analysis techniques, a linearisation process has to be employed to obtain a linear model. "*linmod*" is a built in routine in SIMULINK which perform such a task. With reference to a pre-specified operating point, *linmod* produces a linearised state space representation of the model in standard notations:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned}$$

(4-24)

Thus the generic model is made available to the MATLAB workspace and conversion to other formats, such as transfer function, can be done easily. An *n*-inputs *m*-outputs system can be represented by a matrix of transfer functions

$$H(s) = \begin{bmatrix} H_{11}(s) & H_{12}(s) & \cdots & \cdots & H_{1n}(s) \\ H_{21}(s) & H_{22}(s) & \cdots & \cdots & H_{2n}(s) \\ \vdots & \vdots & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ H_{m1}(s) & H_{m2}(s) & \cdots & \cdots & H_{mn}(s) \end{bmatrix}$$

(4-25)

where

$$H_{ij}(s) = \frac{y_i(s)}{u_j(s)}$$

$$\forall i, j \in (i=1\ldots m, j=1\ldots n)$$

(4-26)

is the transfer function from the j'th input to the i'th output. This representation highlights the cross-axes interactions, which does not show up explicitly in a state space representation. In practice, the conversion from state equations to transfer functions

$$H(s) = C(sI - A)^{-1}B + D \tag{4-27}$$

involves a numerical inversion which is potentially problematic and the actual transfer functions obtained may not be minimal. That is to say the numerator and denominator contain common factors that are not cancelled and hence are of higher order than needed be.

The state space representation offers a more compact and algebraically simpler format. However, SIMULINK does not provide a facility to tag the individual state and hence there is no obvious way to control the order of the states. i.e. It is not possible to tell which state corresponds to which system variable without tracing through the model manually. At first sight, this might be a problem. However, majority of the machine models constructed would have a large number of states and hence the order of the models would be so high that they become numerically ill conditioned. A number of procedures including model order reduction, minimal realisation and input-output balancing of states may have to be applied to obtained a well enough conditioned model.

Indeed, the choice of state variables for a model is not unique and often it is advantageous to work with a set of states other than the system's physical variables. Moreover, analytical tools such as the Bode plot arrays, can be employed to investigate the responses between any pair of input and output. The particular set of state variables chosen to represent the model can be made invisible to the user; which is consistent with the methodology's aim of automation.

Work has also been carried out to test the validity of the SIMULINK routine *linmod*. With reference to the block diagram of a relatively complex (4th order) linear servo, its model is translated to the corresponding state space models by *linmod*. Then a set of algebraic equations relating the input and output signals of each block is written down. With the help of a symbolic calculator (in this case REDUCE), this set of equations is solved. The parameters are only substituted at the end to minimise the accumulation of numerical

errors. The end result is a transfer function of the overall model, in the form of a quotient of two polynomials;

$$H(s) = \frac{num(s)}{den(s)} = \frac{s^m + b_1 s^{m-1} + \cdots + b_{m-1} s + b_m}{a_0 s^n + a_1 s^{n-1} + \cdots + a_{n-1} s + a_n} \tag{4-28}$$

The corresponding bode diagrams of the transfer function and the state equations are plotted using the MATLAB routine *bode* for comparison. It is found that the two models agree closely up to a reasonably high frequency. In effect, this demonstrates that the state space equations generated by *linmod* give a faithful representation of the model up to a certain order for a specified bandwidth.

4.9 Transfer Function Measurements

As it is difficult to ensure that a linearised model based on conceptual design can truly represent the final physical machine, it is suggested that each component in the model be replaced by the transfer function of the real part when it becomes available so that continuous reassessment can be made. An exercise has been conducted on the MIMO rig in Aston University (see Appendix). Frequency response measurements were made and the appropriate transfer functions were fitted to them. A vibration test has been conducted on the tea-bag machine in Molins PLC and some typical values have been determined for the structural coupling on the machine.

4.10 Time Simulation

Although great care has been taken to select the appropriate design criteria, be it time domain or frequency domain specifications, the performance of the control system can only be evaluated properly by running a time simulation with actual motion profiles as input. One such exercise is shown in Figure 4-33. Essentially, a model constructed in SIMULINK is stored as a set of connected transfer function blocks, which is a graphical representation. When a dynamic simulation is called for, a selection of integration routines is available to trace the signals and solve the set of equivalent differential equations.

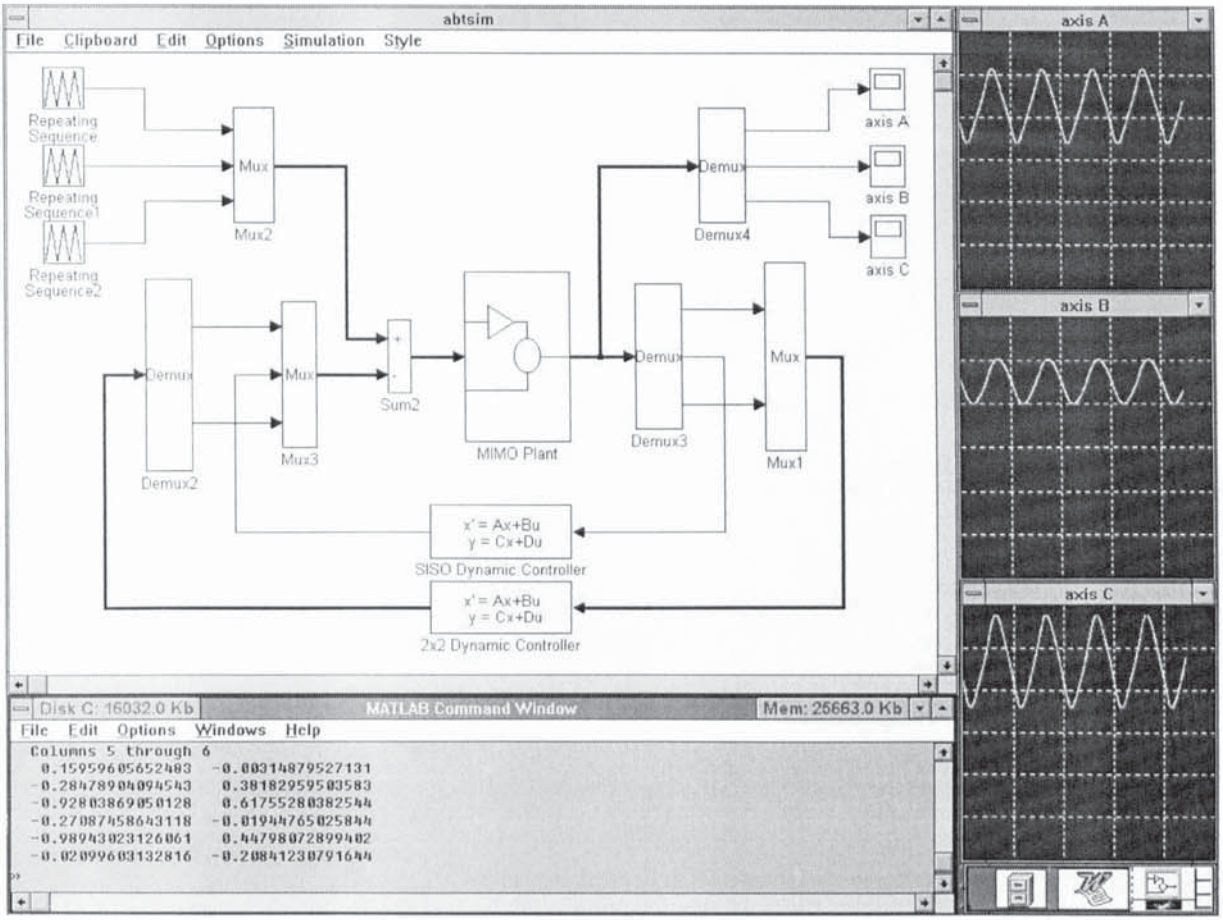

Figure 4-33 Time Simulation in SIMULINK

5. Centralised MIMO Controller Design (I)

5.1 Introduction

5.1.1 Requirements for Centralised MIMO Controllers

In most MIMO control problems, the decentralisation topology is determined by looking for minimal interaction between possible sub-systems in the open loop plant. This approach is not adequate for the problem at hand. Firstly, the topology corresponding to minimal interaction in an open loop plant may not have minimal interaction if some or all of the loops are closed. More importantly, this approach does not take into consideration any possible synchronisation requirements. One may find that axes belonging to different sub-systems, which were determined from the inherent coupling of the open loop plant, may have to be highly synchronised and thus, a topology determined from the open loop plant may be quite unsuitable.

Therefore, it was proposed that the topology should be determined from a full MIMO controller instead. This MIMO controller will satisfy all the performance criteria especially

1. undesired interactions between axes are compensated
2. specified degrees of synchronisation between specified axes are observed

Only when both of the above are satisfied simultaneously does the controller truly reflect the tolerated and required level of interaction between the axes in the closed loop system. Essentially, the controller has to provide a decoupling effort to compensate for the undesired interaction which are some of the inherent couplings in the plant. At the same time, the controller must provide a coupling effort for synchronisation, which can be viewed as a layer of electronic coupling that has to be imposed on the plant. When overlapped, these two layers of efforts may be found to be conflicting and thus make the controller design problem difficult, especially when a simple and highly automated method is required.

The principal reason for a decentralisation topology is the prohibitively high cost (in computing and communication) of implementing a full MIMO controller and it is clear that this full MIMO controller will exist only in simulation. It is for this reason, at least at

the first instance, that the purpose of the full MIMO controller is to facilitate the determination of a topology rather than to provide implementable controls. Therefore less effort is given to ensure the quality of the controller, such as robustness and disturbance rejection, at this stage. Once a topology is determined, individual smaller MIMO controllers can be designed for each of the sub-systems with appropriate attentions. However, in a restricted sense, the full MIMO controller does provide a theoretical best performing controller and thus an upper bound of possible performance for the system. Note that in arriving at the topology, only the controller is "decentralised" and the overall plant is left intact and thus retains maximum information.

To illustrate the potentially conflicting nature of the requirements, an example is given in Figure 5-1. A plant with 8 axes has a certain inherent coupling structure representing by the line connecting the axes. Say synchronisation is required between axes 1-2 and axes 4-5. This requirement implies an imposed structure. A solution in this case may involve a compromise between these two structures. One possible topology is depicted in Figure 5-1 which implies that decoupling is required between axes 2-4, axes 2-5, axes 5-6.

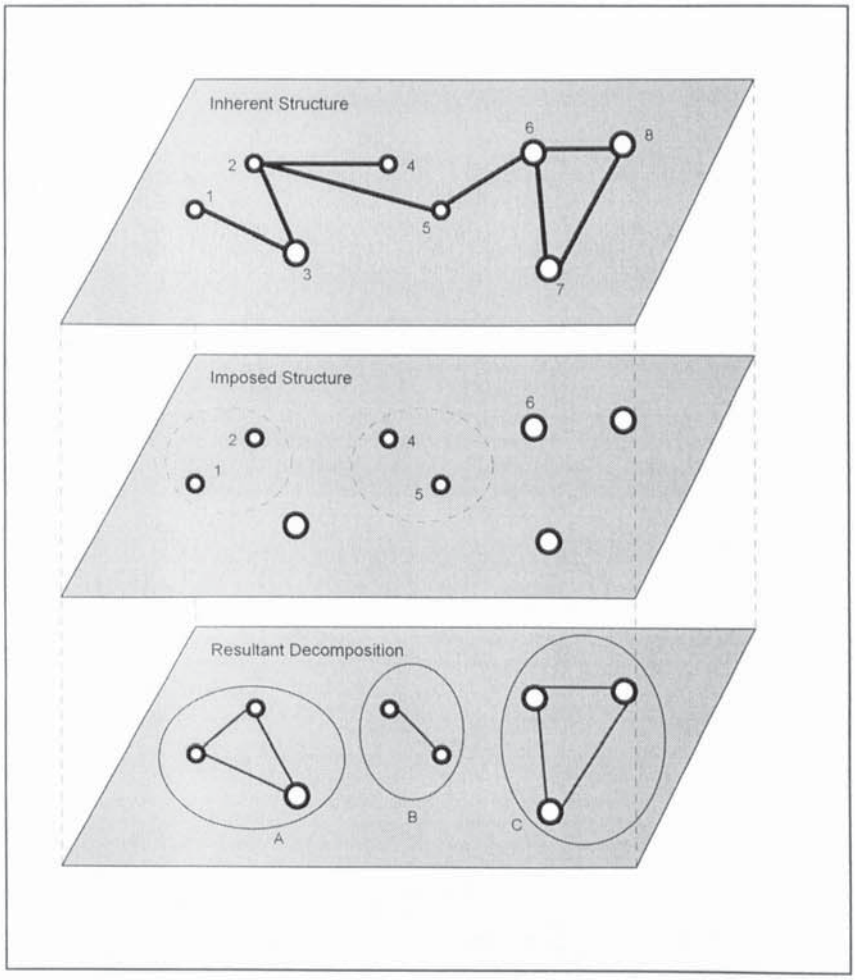

Figure 5-1 Decoupling vs. Synchronisation Requirements

5.1.2 Use of a Simple Benchmark Model for Evaluation

For the purpose of investigating the various controller design methods, a simple benchmark model is constructed so that evaluations can be made comparatively easily and efficiently. Apart from being continuous linear time invariant and minimal, the benchmark model will have web coupling but no structural nor DC coupling. This model, as shown in Figure 5-2 and Figure 5-3, will have significant inherent coupling to illustrate the performance requirements and yet be simple enough to have only 9 state variables - a relatively small number. Another reason for this construction is that close reference has been made to a MIMO test rig (see appendix) that was built in the laboratory in parallel with this simulation work. The model will therefore serve as both a benchmark for the development of the methodology and also a design example that can be implemented eventually.

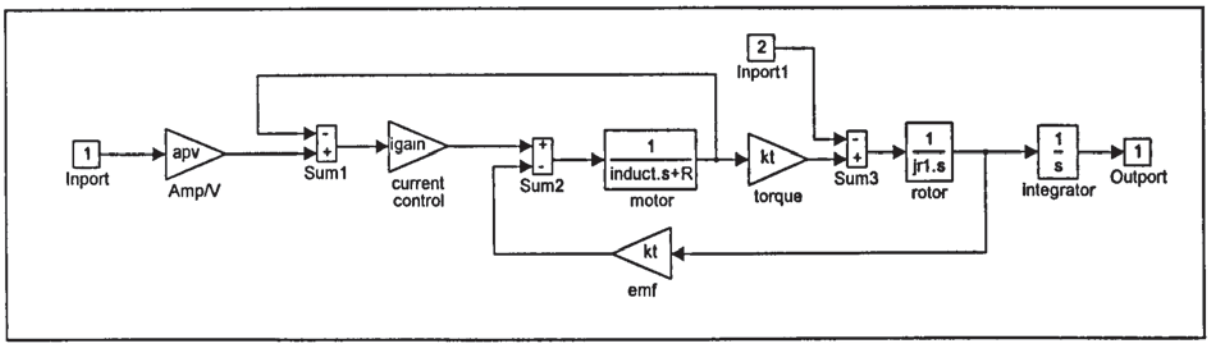

Figure 5-2 Simple Servo Axis

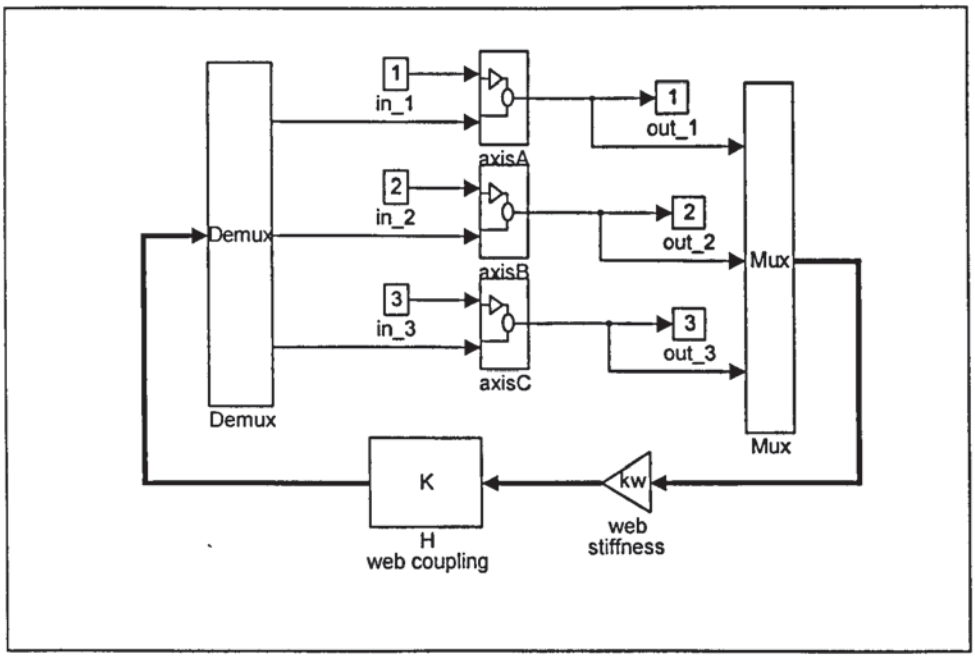

Figure 5-3 3 Axes Benchmark Model

In the benchmark model, the total inertias of the axes are $j_{r1} = j_{r3} = 5.367\text{x}10^{-3}$ kgm^2 and $j_{r2} = 2.521\text{x}10^{-2}$ kgm^2 and the web coupling matrix is

$$H = \begin{bmatrix} 2 & 0 & -2 \\ 0 & 0 & 0 \\ -2 & 0 & 2 \end{bmatrix}$$

(5-1)

and the web stiffness k_w =538.5 +26.92j with a complex part to mimic some degree of damping. These parameters are assigned to the appropriate blocks in the SIMULINK benchmark model (Figure 5-3). With the use of the "*linmod*" routine, a non-linear model can be linearised about a pre-defined operating point. In the case of the benchmark model, which is already linear, the routine translates it into the state space matrices in equation (5-2). Unless otherwise stated, these state space matrices are used when the benchmark model is refereed to.

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0.0397 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.1863 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1863 \\ 0 & 0 & 0 & -0.1164 & -0.1144 & 0 & 0 & 0 & 0 \\ 0 & -1.0769 & 1.0769 & 0.0529 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.1164 & -0.0244 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0529 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.1164 & -0.1144 \\ 0 & 1.0769 & -1.0769 & 0 & 0 & 0 & 0 & 0.0529 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1.6445 \\ 0 & 0 & 0 \\ 0 & 1.6445 & 0 \\ 0 & 0 & 0 \\ 1.6445 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

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

$$D = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

(5-2)

The step and frequency responses of the benchmark model are given in Figure 5-4 and Figure 5-5 for reference. In MATLAB, the computational kernel "*ltitr*" for linear time-invariant time response is used in the step response function "*STEP(A,B,C,D,IU,T)*" which plots the time responses of the system (A,B,C,D) towards input *IU* in time steps specified by the time vector T. By using time vectors of different time step sizes, the accuracy of the "*ltitr*" routine has been checked and found to be satisfactory. In general, the time steps are chosen so that they are short enough to reflect the finer details of the time response concerned.

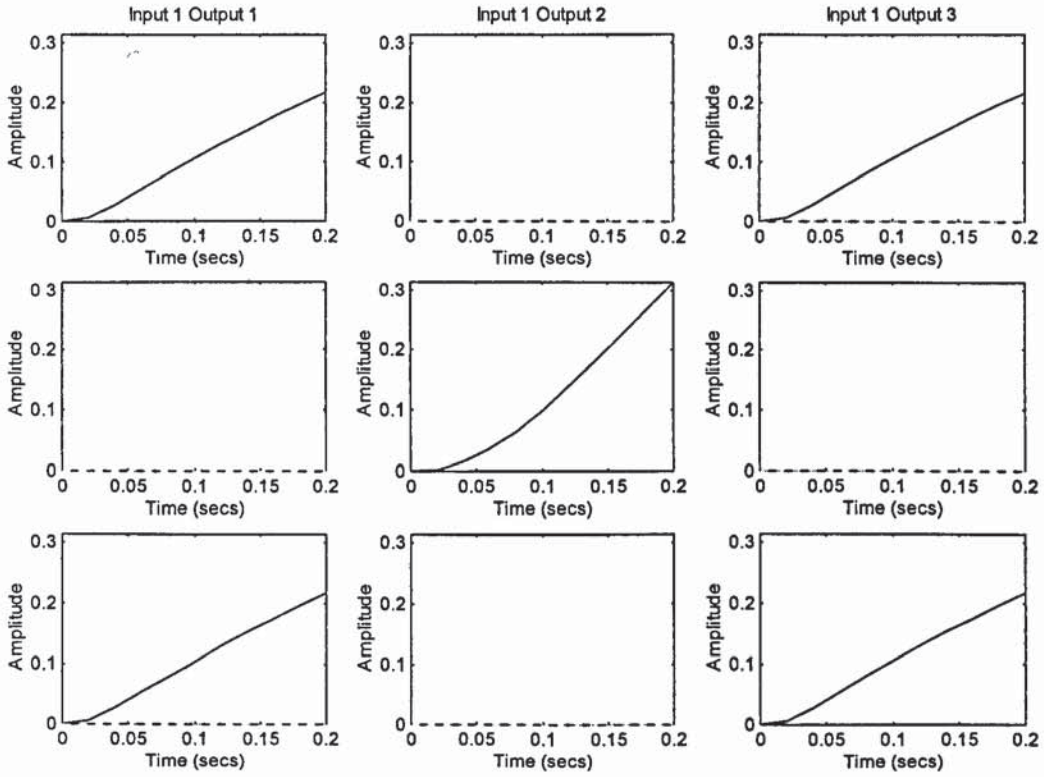

Figure 5-4 Step Response of Benchmark Model (Torque - Position)

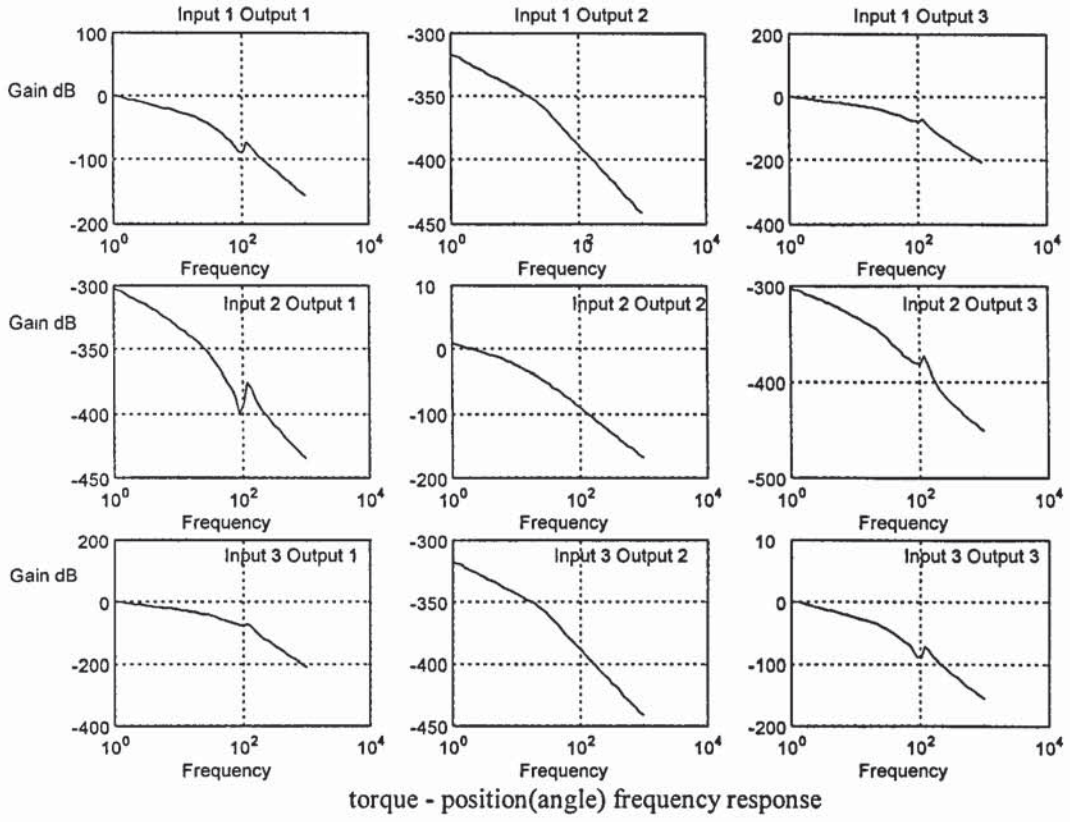

Figure 5-5 Frequency Response (Torque - Angle) of Benchmark Model (with Web Coupling)

5.1.2.1 Relative Importance of Structural and Web Coupling for the Benchmark Model

In section 4.5.2, structural coupling has been identified and incorporated as one form of inherent interaction between the motion axes. In the case of the 3 motors mounted on a flexible platform, the examples in section 4.5.2.3.1 and 4.5.2.3.2 also demonstrate that close agreement can be obtained between estimated resonance frequencies and the actual resonance frequencies measured by external excitation, namely, air gap torque in the motors. With reference to the MIMO test rig described in the appendix, which has built into it both the structural and web couplings, a comparison of their relative importance can be made. Based on the relevant mass-normalised mode shapes calculated in section 4.5.2.3.1, the frequency responses of the structure are estimated and shown in Figure 5-6. Notice the symmetry in the response of axis A and axis C which are in agreement with the mode shapes shown in Figure 4-20. More importantly, comparing the frequency responses of the flexible platform structure (Figure 5-6) and that of the plant with only web coupling (Figure 5-5), it is clear that structural coupling has a gain that is, on the whole, negligible

compare to the sensitivity of the encoder (approx. -62 dB). This is a further justification for including web but not structural coupling in the benchmark model.

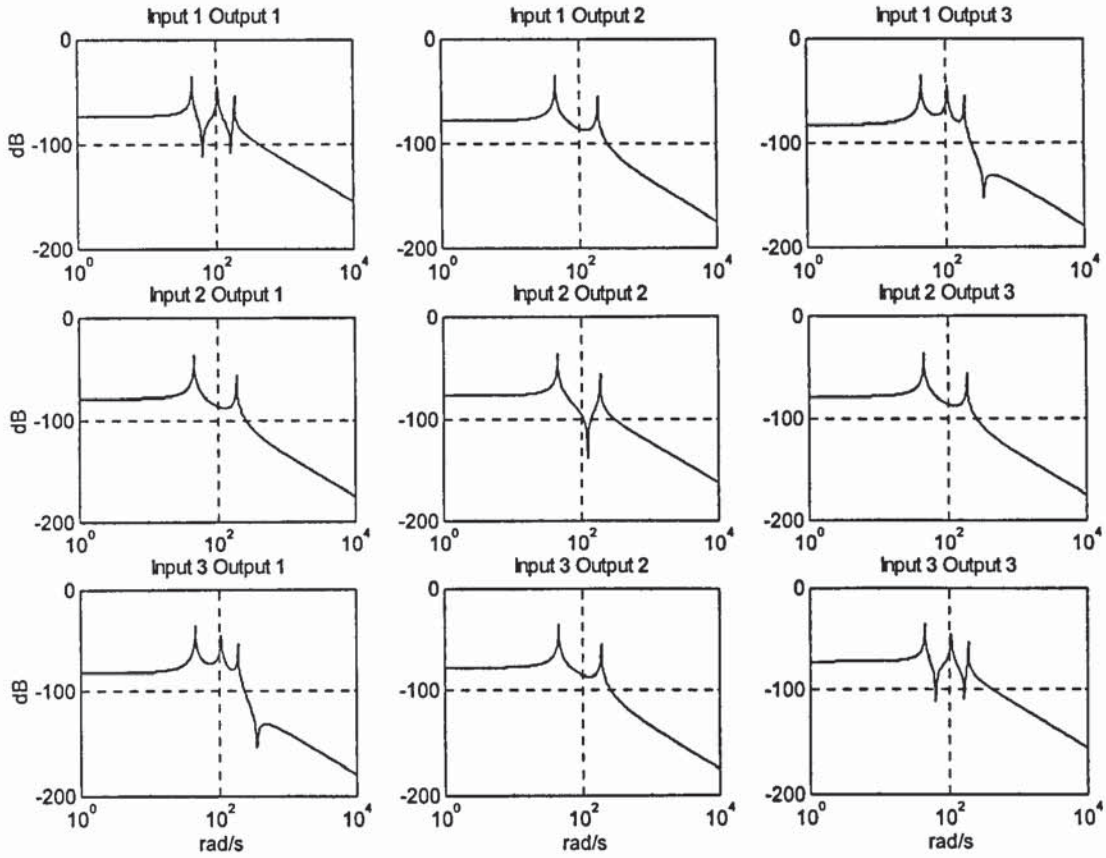

Figure 5-6 Frequency Response (Torque - Angle) for 3 Motors on the Flexible Platform

5.1.3 Criteria for the Evaluation of Design Methods

For the design of MIMO controllers, there are a considerable number of established methods exist in the literature, each catering for a different set of problem structures and requirements. To select an appropriate approach and customise accordingly, these methods must first be evaluated in their capabilities to deliver certain system behaviour. Based on the benchmark problem and the general requirements described in the previous sections, the following is demanded of the controlled system, in behavioural terms

> *For the benchmark model, as shown in Figure 5-3, the 3 axes are denoted by A, B, and C. Axes A and C are inherently coupled through a web whereas axis B is totally independent. It is demanded that axes A and B are to be synchronised but axis C is to be decoupled from the influence of axis A.*

In simulations, the degree of synchronisation and decoupling that is being delivered can be probed by feeding the same motion profile to each input, one at a time, so that the cross axes interactions can be measured. In operational terms, however, it is unlikely for axes that are coupled by a web, and equally unlikely for axes that are to be synchronised, to be required to follow wildly different profiles. Therefore, in experimentation, the degree of synchronisation and decoupling is probed by feeding the same profile to all 3 channels simultaneously but with an addition of a disturbance on different channel one at a time. A number of indicators can be used to assess the performance of each method based on the benchmark problem but the main tools will be the BRG arrays (section 2.7) and the step responses in time domain.

Demanding two axes to be decoupled so that they can follow totally different profiles is unreasonable, if these axes are coupled by a web. This requirement becomes reasonable and valid, however, if it is directed to the axes that are coupled by the other types of inherent interactions, such as structural coupling. The choice of including web coupling rather than the other types of inherent coupling is made because its exaggerated effects on system behaviour is easier to monitor, and perhaps more importantly, that its behaviour can be more conveniently controlled and adjusted.

5.1.4 Design Methods to be Investigated

A brief review of some established methods for MIMO controller design is given in section 2.5. These methods can roughly be put into two categories:

1. Frequency domain methods : These are usually extensions of the classical SISO methods which involve shaping the open loop responses in the frequency domain after some form of diagonal dominance is achieved. These approaches have a qualitative flavour but the need for transfer function manipulations makes them numerically less attractive. System behaviour is often measured by the principal gains which are the maximum singular value of a matrix or the largest possible output induced by the unity input vector. Therefore, the objective is a form of worst case design and can be conservative. Also they seem suitable for tackling the decoupling part of the problem but it is less obvious how they can be used for addressing synchronisation.

2. Optimisation methods : These methods are characterised by their quantitative nature. The approach usually involves manipulation in state space and time

domain specifications can be accommodated with relative ease. However, the use of arbitrary weighting functions and the non uniqueness of the states, which means they do not necessary correspond to any real physical quantities, render the methods less intuitive. Nonetheless, if performance specifications are well defined, one can imagine the construction of an optimal controller which forces the system to approach the demanded output and state trajectories asymptotically at an arbitrary rate.

With the ease of design automation in mind and with the various reasons that are detailed in the subsequent sections, the following approaches were studied and investigated.

- Linear Quadratic Guassian (LQG) optimisation.
- Decoupling by pseudo diagonalisation.
- Decoupling by linear state feedback.
- Edmunds' method - optimising controller parameters to match target frequency responses.
- Numerical search - keeping the performance of the controller within certain tolerance, the trajectory of the controller parameters are followed while the model is perturbed from zero to full inherent coupling.
- Downhill simplex optimisation - the Amoeba algorithm

5.2 Synchronisation by LQG Optimisation

5.2.1 LQ Regulator with Synchronisation Extension for a MIMO Plant

The method of Linear Quadratic Guassian optimisation seems to be the easiest to automate as the level of interaction with the user is relatively low. For the user, the design in its most basic form, is a matter of adjusting weighting functions, whereas most other methods will require some detailed knowledge of control theory. The standard LQG method is well documented. However, to specify the synchronisation requirement and incorporate it into the LQG design, some explanation is needed.

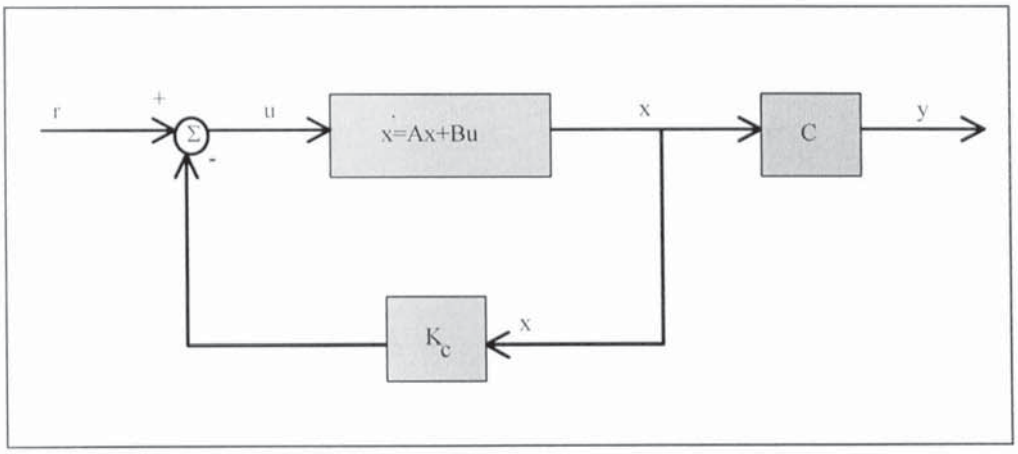

Figure 5-7 Standard LQ Full State Feedback Control Structure

By virtue of the separation principle, the standard LQG problem can be treated as two sub problems - the design of a linear quadratic regulator (LQR) based on linear feedback of the state variables and the design of a linear quadratic estimator (LQE or Kalman filter) which provides an optimal state estimate. First, by assuming all states are available, with the state feedback control law

$$u = -K_c x \tag{5-3}$$

a state feedback matrix K_c can be obtained by minimising the integral square error cost function

$$J = \int_0^T (x' Qx + u' Ru) dt \tag{5-4}$$

subjected to the constraint of the system equation

$$\dot{x} = Ax + Bu \tag{5-5}$$

where the weighting matrix Q is positive semi-definite and R is positive definite. Note that arbitrary scaling of Q and R together do not affect the end result of the LQ optimisation and their importance are only relative to each other. To incorporate the synchronisation requirements, a fictitious spring is introduced between the relevant outputs of the motion

axes. Accordingly, the stiffness of this fictitious spring produces a restoring torque as followed

restoring torque = -(stiffness) x (size of relative error)

or

$$\tau = -\xi\alpha y \tag{5-6}$$

where ξ is the required stiffness, α is a matrix specifying several linear combinations of the outputs which are to be kept near zero and y is the output vector which relates to the state vector by

$$y = Cx \tag{5-7}$$

The weighting on the states in the cost function can therefore be modified as follows to account for synchronisation.

$$Q' = C^T \alpha^T \xi \alpha C + Q \tag{5-8}$$

The solution to the (infinite time) LQ regulator problem is given by

$$K_c = R^{-1} B^T P_c \tag{5-9}$$

where P_c satisfies the algebraic Riccati equation

$$A^T P_c + P_c A - P_c B R^{-1} B^T P_c + Q = 0 \tag{5-10}$$

Similarly, a Kalman gain K_f can be obtained via the Riccati equation for the linear model

$$\begin{aligned} \dot{x} &= Ax + Bu + \Gamma w \\ y &= Cx + Du + v \end{aligned} \tag{5-11}$$

with noise covariance E[w]=E[v]=0, E[ww']=Q_f, E[vv']=R_f and E[wv']=0, such that the continuous, stationary Kalman filter

$$\dot{\hat{x}} = A\hat{x} + Bu + K_f(y - C\hat{x} - Du)$$

(5-12)

produces an LQG optimal estimate of the state. Replacing the state with its estimate in equation (5-3), the overall LQG control structure is shown in Figure 5-8.

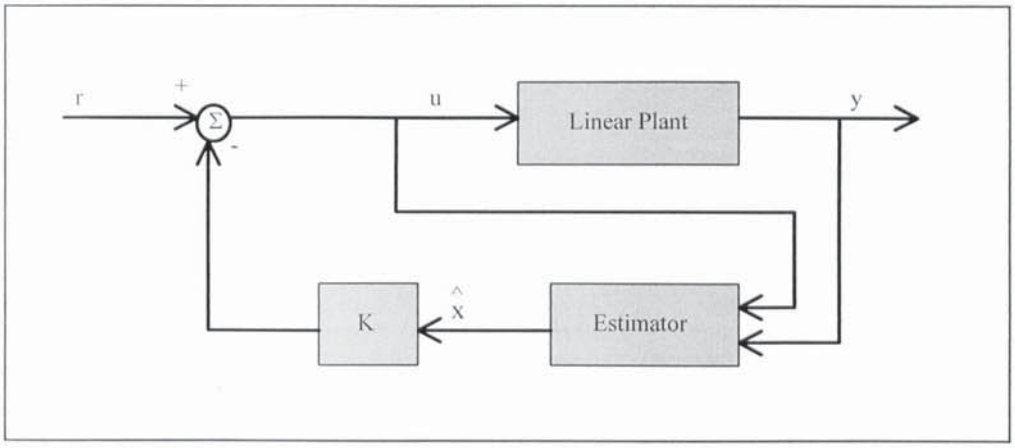

Figure 5-8 LQG Control Structure

With some manipulation, this can be put into the more familiar form

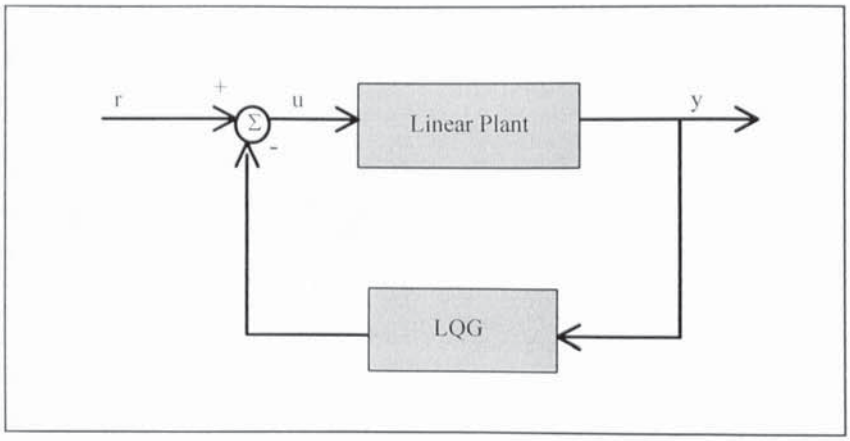

Figure 5-9 Equivalent LQG Control Structure

in which the LQG controller has the following internal structure

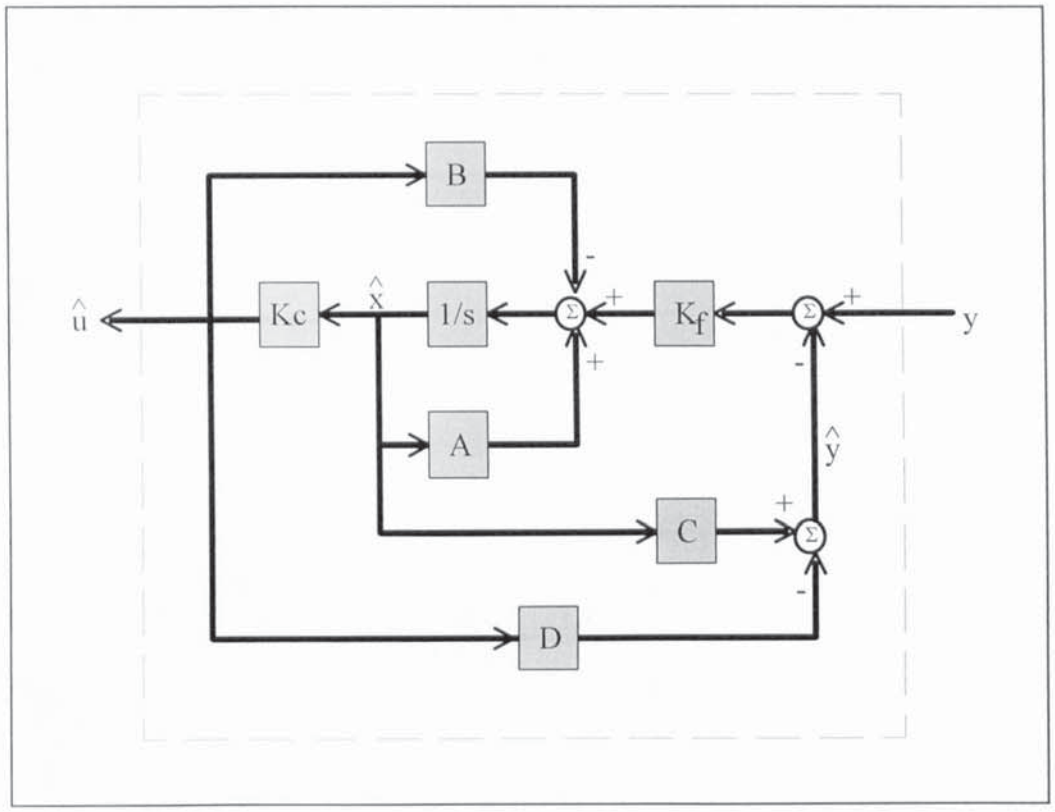

Figure 5-10 Internal Structure of the Continuous LQG Controller

5.2.1.1 Example -- Standard LQG Control for a MIMO Plant

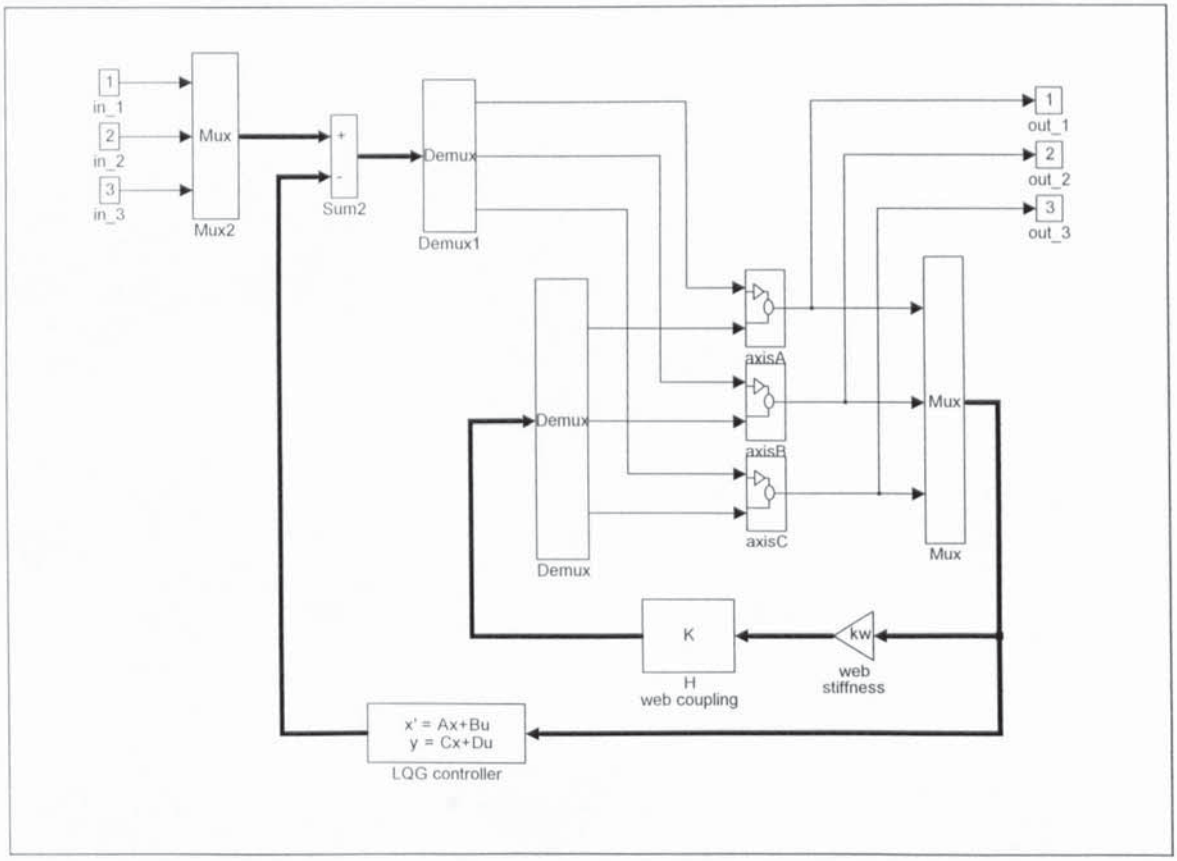

Figure 5-11 MIMO Plant under LQG Control

The routines "*[kc,s,e]=lqr(A,B,q,r)*" and "*[kf,pp,ee]=lqe(A,B,C,v,w)*" from the MATLAB control system toolbox were used to solve the respective Riccati equations to produce the optimal control gains and optimal estimator gains. The routine "*[ac,bc,cc,dc] =reg(a,b,c,d,K,L)*" uses the state space model together with the gains to form a dynamic feedback controller.

As a reference, the web stiffness k_w is set to zero so that the interactions between the axes are "switched" off and that the axes can be considered independent. A LQG controller is designed for this plant and its step responses are shown in Figure 5-12. In fact, the LQG controller designed over the 3 axes is the same as 3 individual LQG controller designed over each axis in turn.

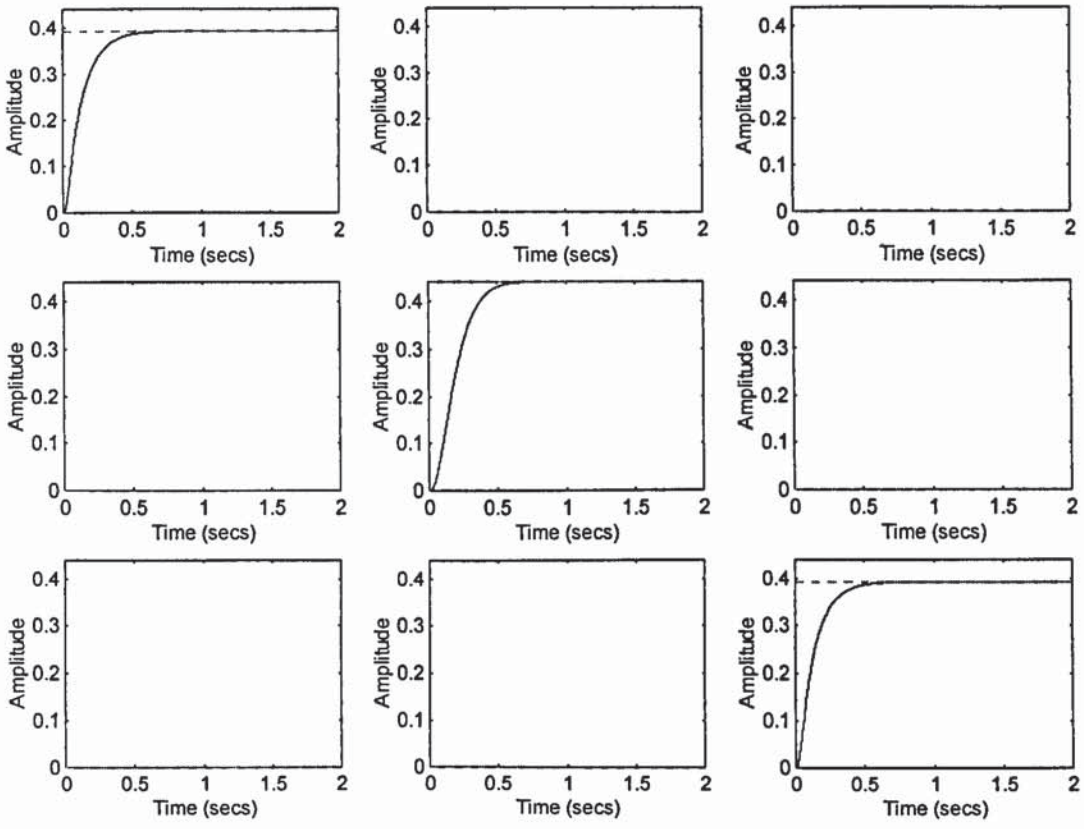

Figure 5-12 MIMO LQG Control on Non Interacting Plant

With the fully interacting benchmark model, (i.e. k_w reset to nominal value,) a new LQG controller is designed and the step responses become those shown in Figure 5-13, which reflects the effect of web coupling between axes A and C.

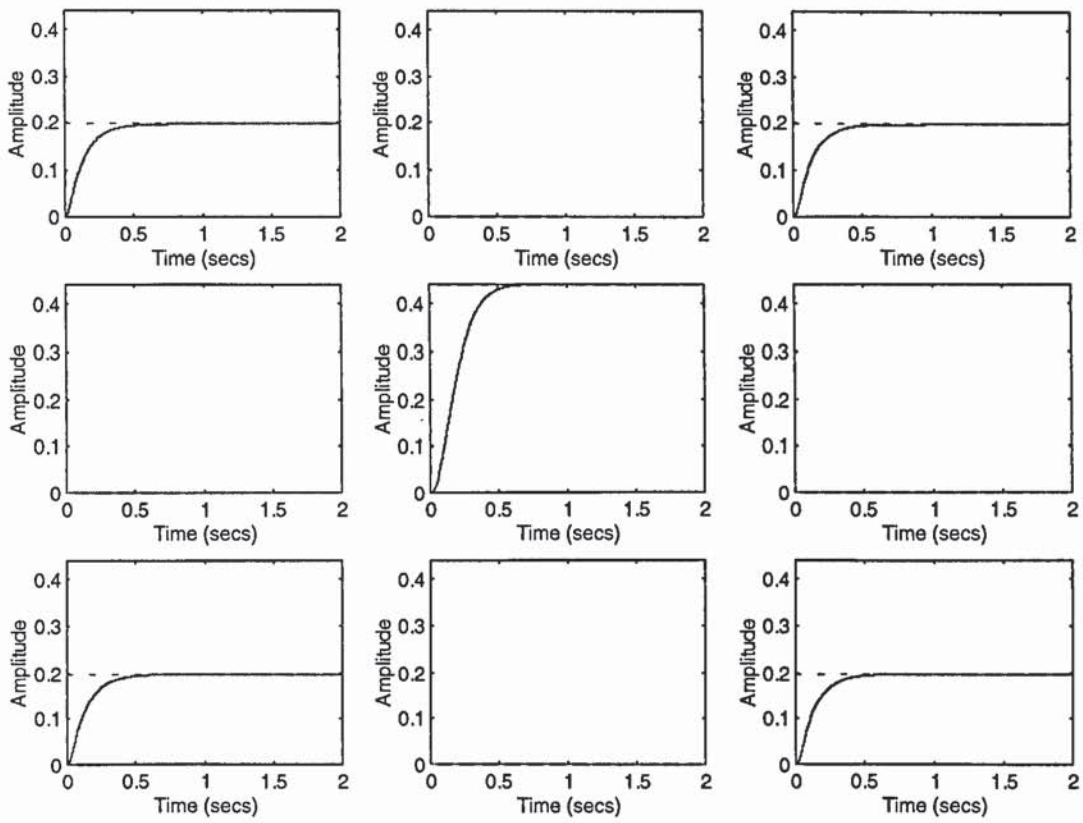

Figure 5-13 LQG Control of the interacting plant

5.2.1.2 Example -- LQ Control with Synchronisation Extension

With reference to the benchmark problem, since the synchronisation is applied only to axes A and B, the overall LQ controller is equivalent to a combination of an individual LQ controller designed over axis C and a 2x2 controller designed over axes A and B. For clarity, this is the route taken to produce the required overall controller in this particular example. The parameters used for the synchronising 2x2 controller are

$$\xi = 100$$
$$\alpha = \begin{bmatrix} 1 & -1 \end{bmatrix}$$

(5-13)

and the step responses of the LQ state feedback and the LQG output feedback are shown in Figure 5-14 and Figure 5-15 respectively.

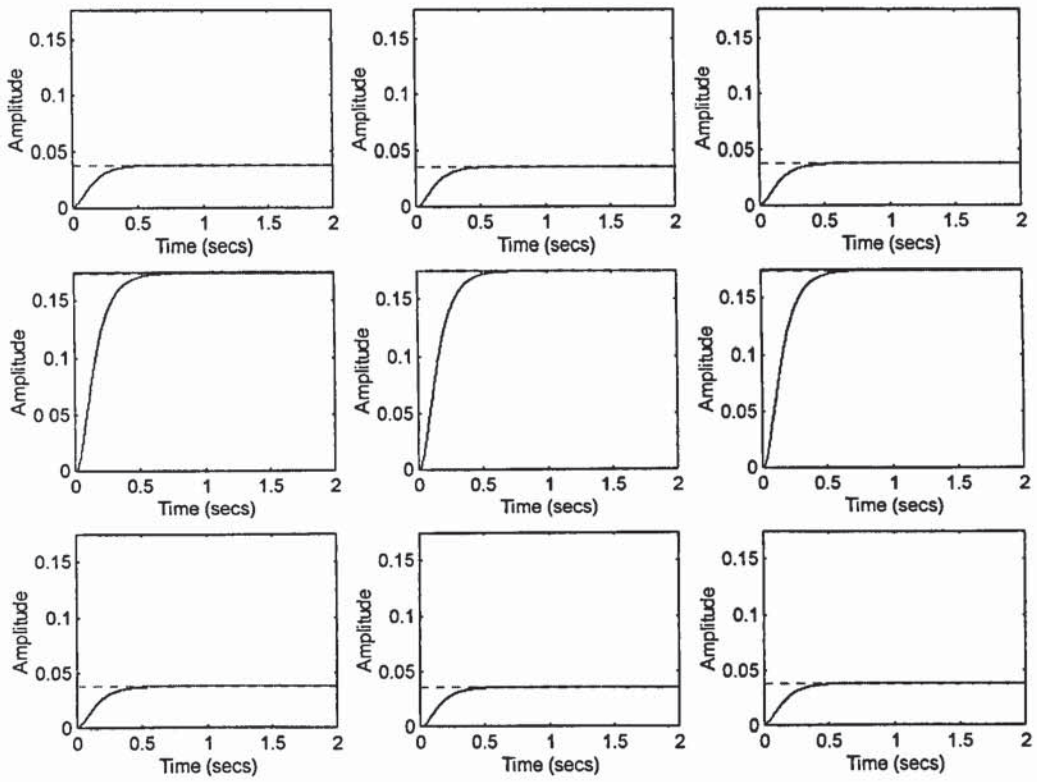

Figure 5-14 Step Responses of Plant under LQ State Feedback with Synchronisation

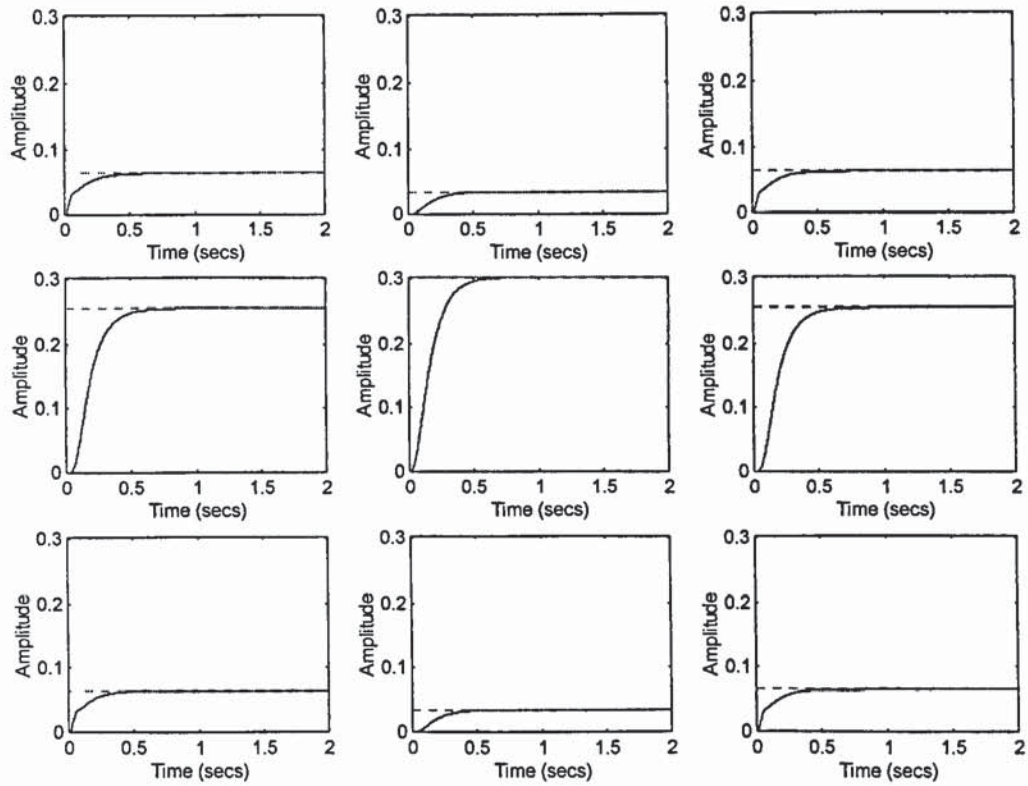

Figure 5-15 Step Responses of Plant under LQG Output Feedback with Synchronisation

This particular formulation of LQ regulator which implicitly optimises the relative error for synchronisation, performs well for the specified synchronisation requirements. However, unlike the synchronisation problem, which can be specified by the relationship between states, the decoupling problem requires the *non*-existence of relationship between certain states. The standard LQ regulator is already in the business of optimising the individual axis so they each follows their input profiles as close as possible. The obvious step of assigning higher weights to the axes needed to be decoupled undermined the performance of the axes that are not involved and no acceptable decoupling controller has been obtained this way. Another possibility, however, is the formulation of LQ controller with feedforward which with 2 degrees of freedom may provide the needed performance.

5.2.2 LQ With Feedforward -- The Tracking Problem

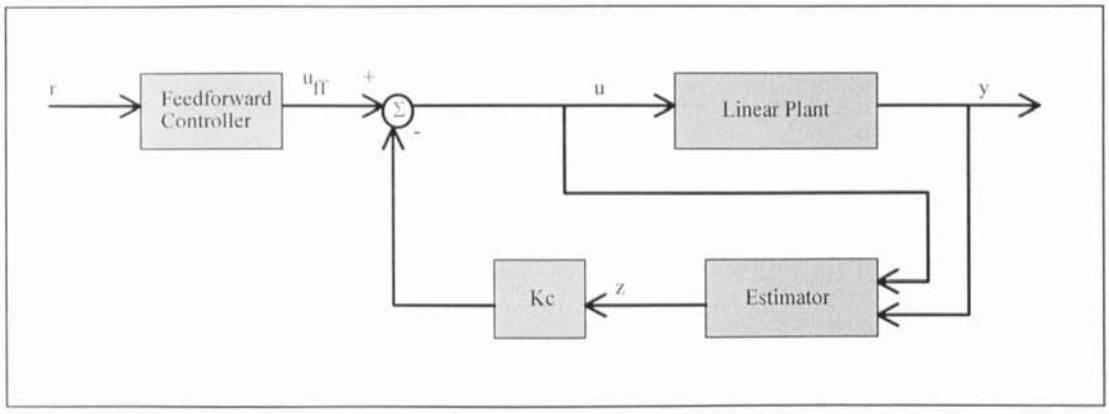

Figure 5-16 LQG Control with Feed-forward for Tracking

The standard LQ formulation is aimed at the design of regulators which return the system to zero state. If a finite trajectory in the form of a time function is specified for the system to follow, the tracking property can be improved by the addition of a feedforward controller [Anderson & Moore 71] in the form of

$$u_{ff} = -R^{-1}B^{T}b \qquad (5\text{-}14)$$

such that the control signal becomes

$$u = -R^{-1}B^{T}(Px + b) \qquad (5\text{-}15)$$

where P(.) is the solution of the matrix Riccati equation

$$-\dot{P} = A^T P + P_c A - P_c B R^{-1} B^T P_c + Q$$
$$P(T) = 0$$

(5-16)

and b(.) is the solution of

$$-\dot{b} = (A - BR^{-1}B^T P)^T b + QC^T (CC^T)^{-1} r_t$$
$$b(T) = 0$$

(5-17)

in which r_t is the trajectory to be followed. In the limiting case where the terminal time T approaches infinity and all matrices are time invariant, part, if not all, of the optimal controller becomes time invariant. However, difficulties may arise for this case because it may not be possible for the plant to track the desired trajectories so that the error approaches zero as time becomes infinite with a finite control law. Moreover, even if it is possible to track in this sense using a finite control law, unless the control approaches zero as time approaches infinity, other difficulties arise due to the fact that the performance index (cost) would be infinite for all controls and therefore minimisation would be meaningless.

It is for the above reason that although in the type of applications concerned, the motion profiles are infinite but periodical varying infinite series, only finite segments (one period) of the profiles are considered. In [Brogan 91], the LQ tracking problem in finite time is solved by integrating two coupled differential equations to produce two sets of time dependent and profile dependent gains. First, the Hamilton-Jacobi-Bellman condition of dynamic programming

$$\frac{\partial g}{\partial t_r} = \min_{u(t)} \left\{ L(x(t), u(t), t) + (\nabla_x g)^T \dot{x} \right\}$$

(5-18)

with optimal control

$$u^*(t) = -\frac{1}{2} R^{-1} B^T \nabla_x g(x(t), t_r)$$

(5-19)

is applied to the LQ cost to produce

$$\frac{\partial g}{\partial t_r} = \min_{u(t)}\left\{\left[x(t)-\eta(t)\right]^T Q\left[x(t)-\eta(t)\right]+u(t)^T Ru(t)+(\nabla_x g)^T\left[Ax+Bu\right]\right\}$$

(5-20)

with $\eta(t)$ as the desired state trajectory and $\nabla_x g$ as the "sensitivity of cost function towards changes in the states". The boundary condition is

$$g\left[x(t),t_r\right]_{t_r=0}=\left[x(t_f)-x_d\right]^T M\left[x(t_f)-x_d\right]$$

(5-21)

where M is a symmetric positive semi-definite matrix, $t_r=t_f-t$ is the reverse time counting backwards from the terminating time t_f and x_d is the terminating state. The HJB equation is solved by assuming a solution

$$g\left[x(t),t_r\right]=x^T(t)W(t_r)x(t)+x^T(t)V(t_r)+Z(t_r)$$

(5-22)

substituting (5-22) into (5-20) and equating terms of the same order in states leads to the following differential equations

$$\frac{dZ}{dt_r}=\eta(t)^T Q\eta(t)-\frac{1}{4}V^T(t_r)BR^{-1}B^T V(t_r)$$

(5-23)

$$\frac{dV}{dt_r}=-2Q\eta(t)+A^T V(t_r)-W(t_r)BR^{-1}B^T V(t_r)$$

(5-24)

$$\frac{dW}{dt_r}=Q+WA+A^T W-WBR^{-1}B^T W$$

(5-25)

These coupled equations are solved in reversed order and normally $Z(t_r)$ is not needed unless the actual cost is needed to be determined too. Differentiating equation (5-22) and treating $x(t)$ and t as independent variables gives

$$\frac{\partial g}{\partial t_r} = x^T(t)\frac{dW}{dt_r}x(t) + x^T(t)\frac{dV}{dt_r} + \frac{dZ}{dt_r}$$
$$\nabla_x g = 2W(t_r)x(t) + V(t_r)$$

(5-26)

together with equations (5-19) and (5-15) determine the actual control profiles. However, for a specified finite time trajectory, the gains of both the feedback and the feedforward controllers become time dependent and have to be determined off line. This undesirable requirement implies that the gains will have to be updated for each sampling period. The implementation of such a controller can be achieved either by downloading the entire gains table to the controller card at initialisation or passing the gains from the host computer to the controller card at each sampling interval. In both cases, it is estimated that the demand would exceed the system capability. Therefore, it was decided not to pursue this approach any further.

5.3 Decoupling Methods

5.3.1 Decoupling Controllers

Two methods for obtaining decoupling controllers have been investigated. The method of pseudo diagonalisation based on optimising the column dominance of the closed loop system responses and the method of full state feedback. It is found that the method of full state feedback produces a superior level of decoupling. However, both methods become less effective when they are applied to higher order models. As with synchronisation, which is usually demanded for a limited number of axes, a specific level of decoupling may only be critical for a limited set of axes. Thus the full state feedback method may still be applicable if applied selectively.

5.3.2 Decoupling By Pseudo Diagonalisation

Within the MFD (multivariable frequency domain) toolbox for MATLAB, a routine is available for pseudo diagonalisation. By specifying the frequency responses of the plant and a suitable set of controller forms (order of controller), the gains of these controllers are calculated by optimising the column dominance of the *closed loop* system. The 'level' of decoupling can be assessed by plotting the corresponding inverse column dominance of each diagonal response. Some degree of decoupling is achieved if its inverse column

dominance is below zero dB. This is only "partial" decoupling in the sense that the interactions between any two outputs are minimised rather than "eliminated".

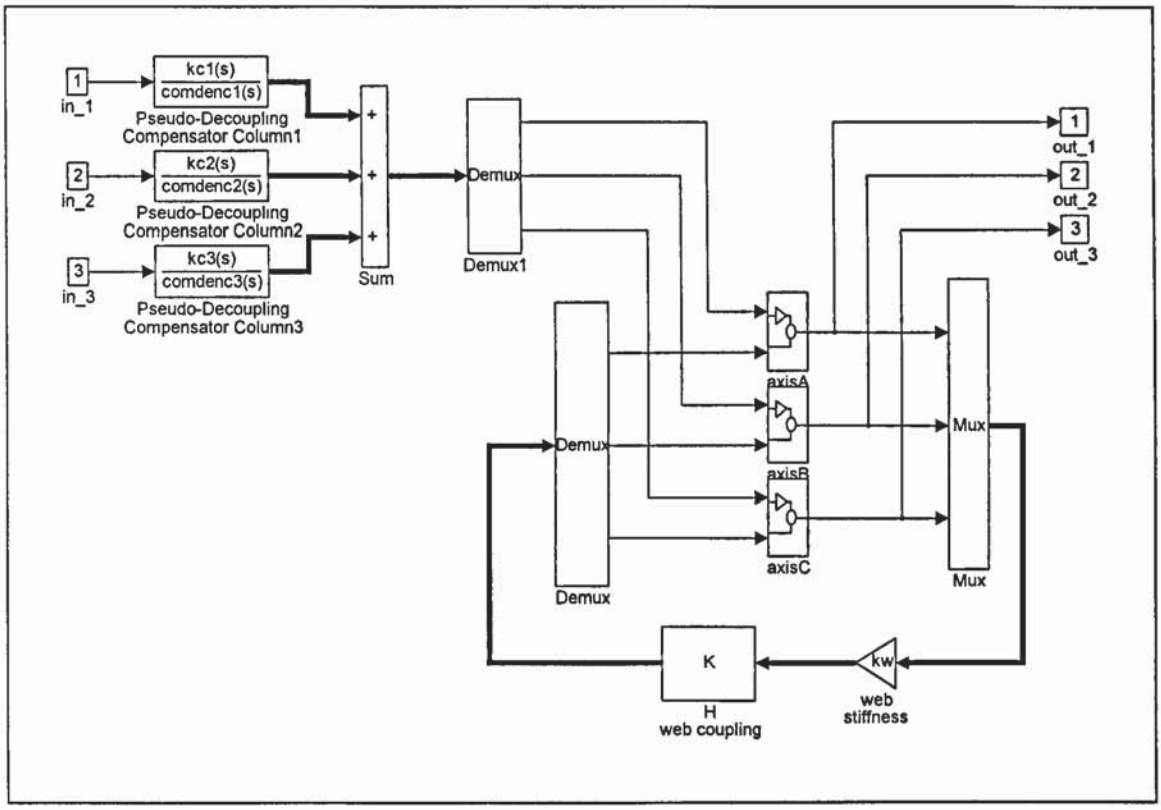

Figure 5-17 Pseudo-Decoupling Pre-Compensators

The routine "*[K,Comden]=fpseudo(ω,G,j,KFORM)*" is used to obtain the controller transfer function matrix *K(s)/Comden(s)* for the j'th column. *G(s)* is the set of open loop frequency responses of the plant over the frequency vector ω and KFORM is a matrix which specifies the order of the compensator for each row. Using the open loop response at the high ($s \to \infty$) and the low ($s \to 0$) end of the frequency range in interest, the routine "*[UPPER,LOWER]=fadj(ω,F)*" gives a crude estimate of the relative order of the required compensator for each column and based on this a trial *KFORM* is formed. The coefficients of the j'th column of the decoupling pre-compensator *K(s)* is calculated to minimise the j'th column inverse dominance

$$\frac{\int_{\omega} \sum_{\substack{i=1 \\ i \neq j}}^{m} |q_{i,j}| \cdot W(\omega)}{\int_{\omega} |q_{i,j}| \cdot W(\omega)} \qquad (5\text{-}27)$$

where $W(\omega)$ is the weight at frequency ω and $q_{i,j}$ are elements of the compensated system

$$Q(s) = G(s) \cdot K(s)$$

(5-28)

Taking the benchmark model described in section 5.1.2 as an example, the design procedure is applied. However, for the reason of clarity, the input and output of the non-participating axis B are removed while leaving the state vector intact, i.e. setting $y=[y_1\ y_3]'$, $u=[u_1\ u_3]'$ instead of $y=[y_1\ y_2\ y_3]'$, $u=[u_1\ u_2\ u_3]'$ in the state model. The "*fadj*" routine from the MFD toolbox suggests a pre-compensator of the form for the resultant 2 input 2 output model.

$$K = \left[\begin{array}{c:c} k_{11,2}s^2 + k_{11,1}s + k_{11,0} & k_{12,0} \\ \hdashline k_{21,0} & k_{22,2}s^2 + k_{22,1}s + k_{22,0} \end{array}\right]$$

(5-29)

and the following compensator is obtained

$$K = \begin{bmatrix} -3.166 \times 10^{-4} s^2 - 67.99 & 67.97 \\ 67.97 & -3.166 \times 10^{-4} s^2 - 67.99 \end{bmatrix}$$

(5-30)

which produces the following inverse dominance plots

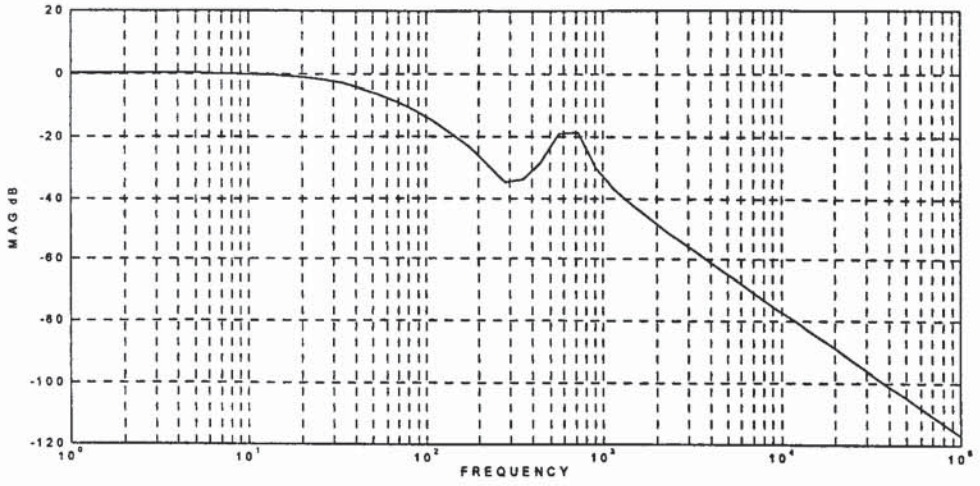

Figure 5-18 Inverse Dominance of Axis A

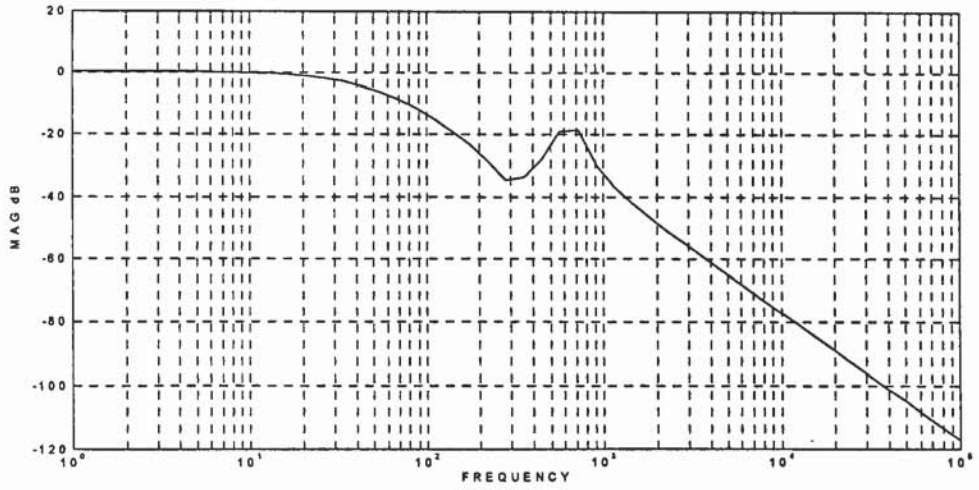

Figure 5-19 Inverse Dominance of Axis C

The new compensator is relatively simple in form and provides a reasonable degree of dominance over the frequencies. Nonetheless, when applied to higher order models, the pre-compensators produced are less than satisfactory. Further investigation will be carried out to look for a more suitable method for decoupling.

5.3.3 Total Decoupling By Linear State Variable Feedback

A more general approach to the decoupling problem is the use of linear state feedback. In the construction of the generic system model and the study of LQ optimisation methods, it was found that the examples considered are observable. Although this may not be true in general, it may be reasonable to expect this is the case at this point and therefore state estimators are assumed to be realisable when required. In the classic paper [Falb & Wolovich 67], conditions for a linear state variable feedback compensator are given for linear multivariable systems and the set of all decoupling controller defined. An alternative and extended treatment is given by [Gilbert 69]. Consider the linear system S

$$\begin{aligned} x &= Ax + Bu \\ y &= Cx \end{aligned} \qquad (5\text{-}31)$$

denoted by the matrix triple {A,B,C} of sizes $n \times n$, $n \times m$, $m \times n$ respectively and with a transfer function

$$H(s) = C(Is - A)^{-1} B$$

(5-32)

and assume the control takes the form

$$u = Fx + G\omega$$

(5-33)

where ω is the external input. The corresponding transfer function for the closed loop system {A+BF,BG,C} is

$$H(s, F, G) = C(Is - A - BF)^{-1} BG$$

(5-34)

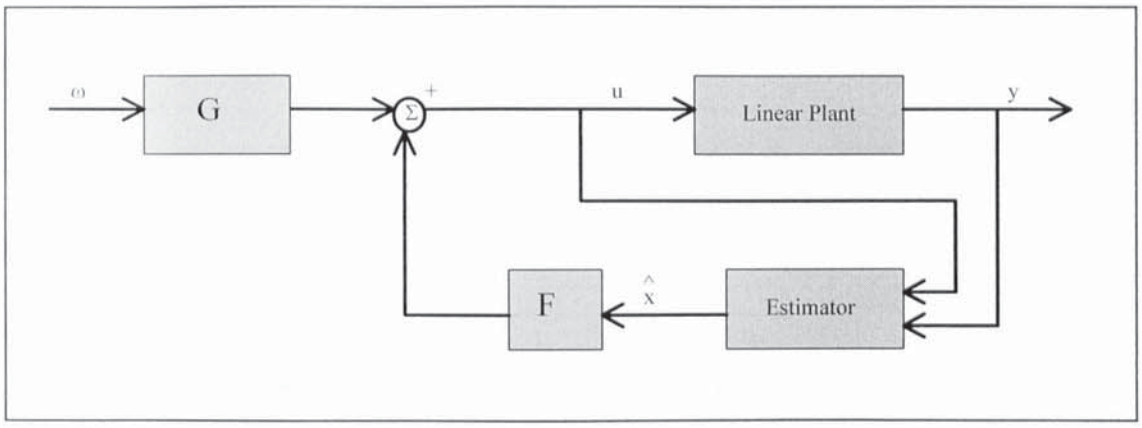

Figure 5-20 Decoupling by Linear State Feedback

The system S(F,G) is defined to be decoupled if H(.,F,G) is diagonal and non-singular. This is a total decoupling design in the sense that the closed loop system will have input-output pairs that are independent of each other. In addition, an F-invariant of S is defined to be any property of S(F,G) which does not depend on F, for any fixed G. Two such invariants d_i and B^* are defined as follows

$$d_i = \begin{cases} \min\{j : C_i A^j B \neq 0,\ j = 0,1,\ldots,n-1\} \\ n-1 \end{cases} \quad \begin{array}{l} \\ \text{if} \quad C_i A^j B = 0 \quad \text{for all } j \end{array}$$

(5-35)

$$B^* = \begin{bmatrix} C_1 A^{d_1} B \\ C_2 A^{d_2} B \\ \vdots \\ C_m A^{d_m} B \end{bmatrix}$$

(5-36)

B^* is a m x m matrix. It can be shown that the necessary and sufficient condition for the existence of a decoupling pair (F,G) is

$$\det(B^*) \neq 0$$

(5-37)

It is suggested that the decoupling property of a system can be classified as follows

det(H)=0	strong inherent coupling
det(B^*)≠0	no inherent coupling
det(B^*)=0 and det(H)≠0	weak inherent coupling

in which the strongly coupled system can not be decoupled by any control laws whereas the weakly coupled system can not be decoupled by state feedback but only by other control laws. Also define the m x n matrix

$$A^* = \begin{bmatrix} C_1 A^{d_1+1} \\ C_2 A^{d_2+1} \\ \vdots \\ C_m A^{d_m+1} \end{bmatrix}$$

(5-38)

$$F = B^{*-1}\left[\sum_{k=0}^{\delta} M_k C A^k - A^*\right]$$

$$G = B^{*-1}$$

(5-39)

where $\delta=\max(d_i)$. It is also shown that by an appropriate selection of the matrices M_k,

$$m+\sum_{i=1}^{m} d_i \le n$$

(5-40)

of the closed loop poles can be assigned by varying M_k. However, the decoupling problem is only part of the overall control objective and it is sufficient to consider, at the first instant, the decoupling pair (F^*,G^*) which is only one particular solution given as part of the proof for (5-37).

$$F^* = -B^{*-1}A^*$$
$$G^* = B^{*-1}$$

(5-41)

The following has been obtained from the 3-axes benchmark model with and without the decoupling controller. The levels to which they are decoupled are assessed by their BRG * plots and the step response of the compensated system is also shown.

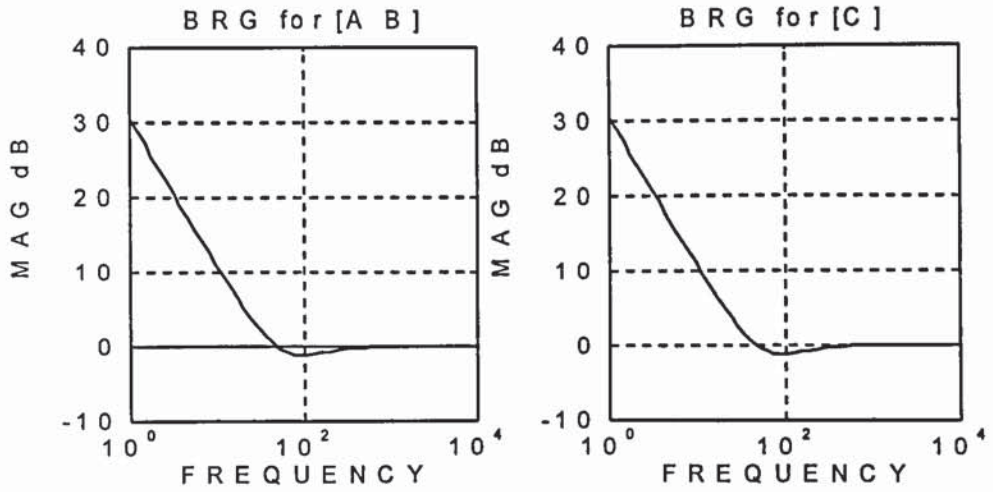

Figure 5-21 BRG for Plant

* Essentially the BRG matrix can be understood in the following way: Consider a full MIMO transfer function matrix. By imposing a selected block diagonal structure, a decentralised system is obtained by discarding the non block diagonal elements. BRG is the transfer function matrix by which a block has to be multiplied to compensate for the discarded elements. (i.e. when there is no off diagonal interaction, the BRG is the identity matrix). It therefore serves as a good measure of interaction between each block and the rest of the overall system.

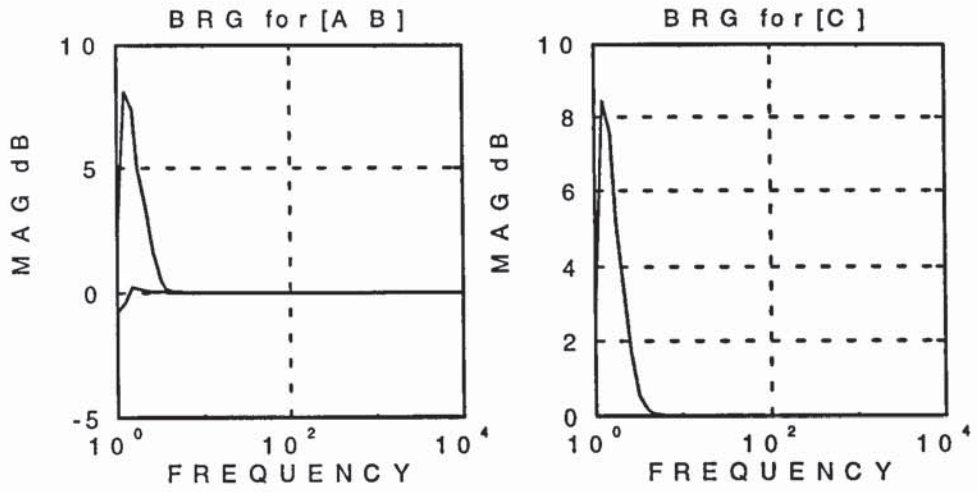

Figure 5-22 BRG for Decoupled Plant

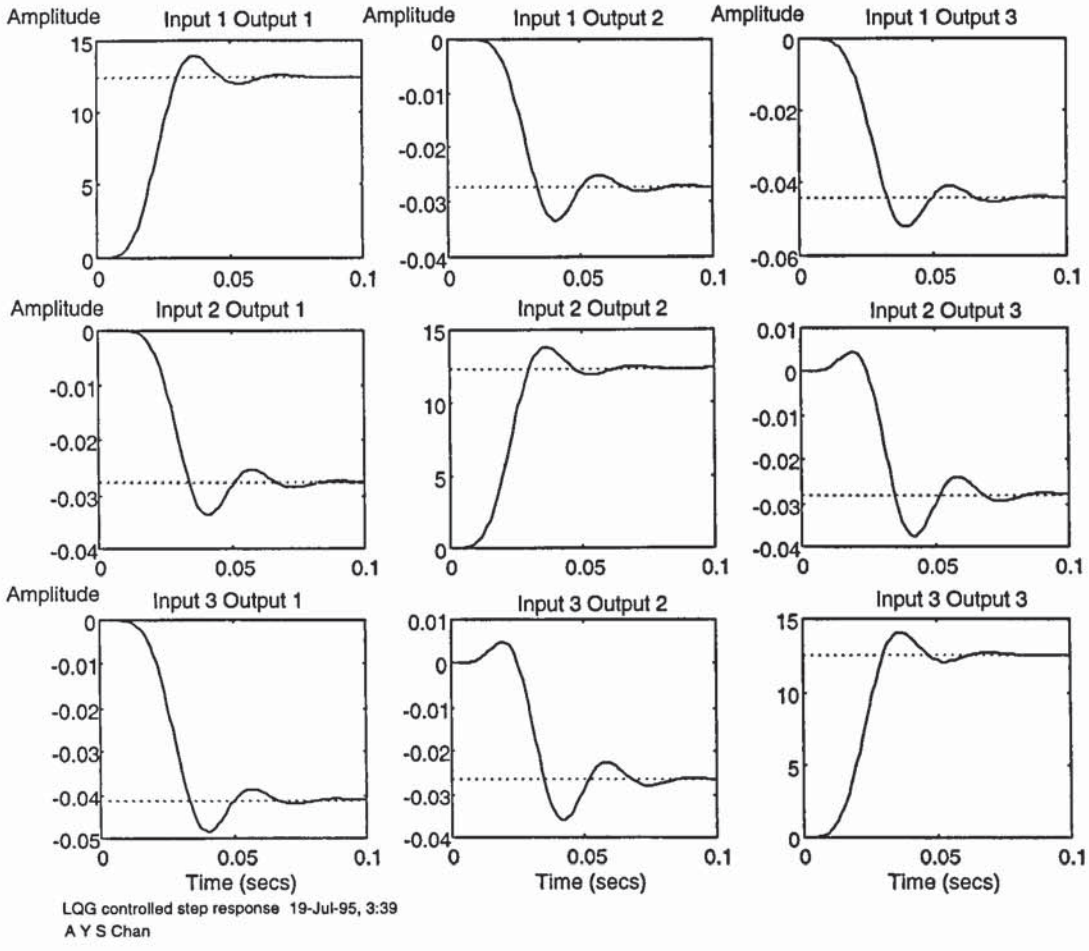

Figure 5-23 Step Response for the Decoupled System

Working along these guidelines, there has been some success with the models of smaller dimension but models of higher order appear to be less successful.

5.3.4 Edmunds' Method

The Edmunds' method [Edmunds 79] is a procedure that attempts to match the frequency response of the closed-loop compensated system to a target frequency response T [Maciejowski 89]. The designer need to provide the frequency responses for the MIMO open loop plant G and pre-define the controller structure in terms of numerators and common denominator of a transfer function array K. The matching is achieved by the optimisation of the coefficients of the numerators of K over the frequency range specified, such that

$$T = GK(I + GK)^{-1} \tag{5-42}$$

By demanding T to have zero response in the diagonal elements, it is possible, in theory, to obtain a decoupling controller. Moreover, by specifying frequency responses of the relevant off diagonal elements of T, it is also possible to obtain a synchronising controller. On the surface, the procedure has the potential to be the one-hit design method that produces a controller which synchronise and decouples simultaneously. However, the main difficulty is that of selecting an appropriate controller structure, which requires plenty of experience and knowledge of the system concerned.

For the decoupling problem, taking an n'th order polynomial, (where n varies between 2 to 9, since there are 9 states in the benchmark model), as the common denominator of the transfer function array K, controllers were designed. It was found that the controllers with higher order dynamics do not necessary outperform the lower ones, possibly due to the sensitivity of coefficients of high order polynomials. Overall, there is not enough guidance given to the designer through this procedure. Another drawback worth mentioning is that this method does not guarantee closed loop stability and a separate check has to be made.

6. Centralised MIMO Controller Design (II)

6.1 MIMO Controller Design - Non Analytical Approaches

In view of the need to guarantee at least a workable solution to the centralised controller synthesis problem, namely, the design of a controller that simultaneously synchronises and decouples axis to specification, some non analytical approaches are to be investigated:

1. *Iterative approach with incremental constraints:* Based on the solution obtained by setting the inherent coupling to zero (i.e. system with independent axes), the controller is modified iteratively with the incremental introduction of inherent coupling and synchronisation requirements, while keeping the tracking property at each stage. Numerical search has the advantage that it is relatively easy to automate once the structure of the problem is understood and set up properly. However, this approach does not guarantee the existence of a solution.

2. *Numerical optimisation:* Some form of hill descending strategy will be employed to optimise the controller parameters over the performance space. The focus will be the construction of a suitable performance index that reflects the control objectives. An important point is that at least a local minimal solution is guaranteed although it may not satisfy all the performance requirements.

6.2 Search Along The Path Of Increasing Degree Of Inherent Coupling

6.2.1 Decoupling Controller -- Keeping Performance within Specification Boundary

Based on the benchmark problem, an iterative routine has been set up to search for a valid MIMO decoupling controller by the following steps (Figure 6-1 and Listing 6-1) and synchronisation consideration will be omitted for the time being:

1. *Start with a plant G with no inherent coupling* - this is done by setting all the coupling matrix in the target model to zero.

2. *Obtain controller K for G with performance P(G,K)<S (S=specification)* - an LQ state feedback controller K is designed for the plant G. Performance vector P

is measured as the root-mean-square error of the step responses for a pre-defined time range which will be adjusted to cover the settling time after a few trials. The specification vector S is defined as the performance of the non-interacting plant with an additional prescribed tolerance. This specification can be relaxed progressively to allow for an "elastic" boundary if it turns out to be too stringent. Note that in general S is a high dimensional vector. Relaxing the whole vector by multiplying it with a scaling factor may not be conservative enough while relaxing individual element one at a time may be computationally too intensive.

3. *Introduce coupling incrementally until $P(G',K)>S$ for plant G'* - a series of new plants G' are obtained by multiplying all the coupling matrices with an increasing fraction in the target model. New performances based on the current controller K are calculated for these plants until it exceeded the specification.

4. *Obtain gradient $\partial P(G',K)/\partial K$*

5. *Obtain new K in the direction of $P<S$* - performance has been pushed across the specification boundary by the new level of inherent coupling in the plant. Attempt to pull performance back inside the boundary by adding to K with (λ is a scaling constant)

$$\Delta k = \lambda \left(\frac{\partial P(G', K)}{\partial K} \right)^{-1} [S - P(G', K)] \qquad (6\text{-}1)$$

6. *Iterate steps 1 to 5 until target plant is reached.*

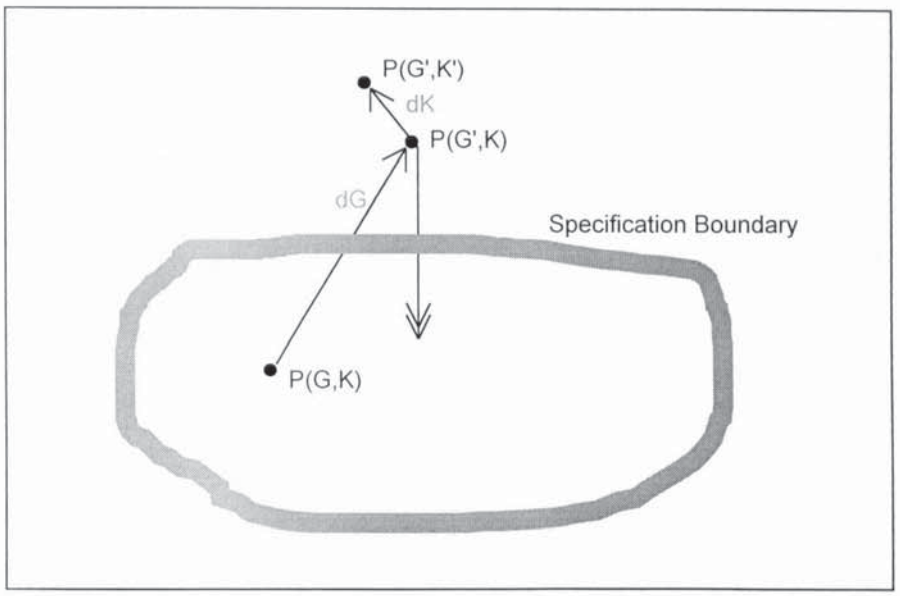

Figure 6-1 Search Strategy

No proof of the existence of a solution is given here. However, it has been shown that controllers can be constructed separately for synchronisation, and to a certain degree, decoupling. The nature of the design methodology should provide opportunities for the designer to discover if the specification is unreasonable and relax them accordingly in an iterative manner.

In geometrical terms, the suggested method starts with a plant (G) consisting of a number of axes each under single input single output control (K). The plant description is moved towards the fully interacted plant gradually with the control law changing accordingly so that the specification is met all the time. It has the advantage of taking the 'factory' solution as the starting point and produces an improved solution based on information on inherent couplings. The position of the performance vector is monitored and attempts made to keep it on the acceptable side of the specification boundary.

For the decoupling part of the benchmark problem, where the controller K has 27 parameters, a performance vector of at least 27 dimensions is needed to determine a unique solution. For the trial runs, various combination of the root-mean-square errors of the step responses of all 3-by-3 channels over different time ranges were used. It soon becomes clear that the iterative process produces a solution that oscillates just outside the specification boundary but never quite converges to a stable value. i.e. the vector (S-P) has elements that are small and non negative. Apart from the fact that no solution was guaranteed, this behaviour can be explained by the fact that this algorithm does not take

into account the sensitivity of the performance vector towards variations in parameter changes and therefore convergence is not guaranteed either.

Listing 6-1 Pseudo Program for Search Along Specification Boundary

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%%      Search for centralised controller:
%%      by incrementally approach required level
%%      of inherent coupling and synchronisation
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

initialisation(model,time,frequency,interval,etc...);
level_of_interaction=0;
plant=plant(level_of_interaction);
kc=LQ_controller(plant);
specification=performance(plant,kc)tolerance;

for level_of_interaction=0 to 1                          %% for loop
        plant=plant(level_of_interaction);
        while flag=true                                  %% while loop
                performance=performance(plant,kc);
                distance_out=performance-specification;
                if (performance~<specification)          %% if not within spec
                        dpdk=jacobian(performance,kc);
                        dkdp=inverse(dpdk);
                        dk=-1.5*dkdp*distance_out;
                        kc=kc+dk;                        %% form new kc
                else                                     %% if within spec
                        flag=flase;
                end_if;                                  %% test spec
        end_while;                                       %% while loop
end_for;                                                 %% change level
```

6.2.2 Decoupling Controller -- Search on the Specification Boundary

Taking account of its drawbacks, the search strategy in the previous section is modified as described below. First of all, the performance vector P is defined as a function of the plant G and the controller K.

$$P = P(G, K) \tag{6-2}$$

It is helpful to consider the partial derivative

$$\left.\frac{\partial K}{\partial G}\right|_P = \left(\frac{\partial P}{\partial K}\right)^{-1} \frac{\partial P}{\partial G} \tag{6-3}$$

together with the relationship

$$\Delta K = \frac{\partial K}{\partial G}\Delta G$$

(6-4)

which points to a more refined approach to updating the controller during each iteration. Again starting with the solution for the non interacting plant, the strategy is to look at the deviation of performance caused by the changes ΔG in the plant G, as it approaches the target plant incrementally, and attempts to compensate these changes at each stage by modifying the controller K with ΔK. However, for this to work, consideration will also have to be given to the range of linearity to which these perturbations apply. Probing along all directions will produce the acceptable range of linearity $\Delta G_{max,n}$ and $\Delta K_{max,n}$ for each iteration. The new perturbations are truncated unless the following holds

$$\Delta G_n < \Delta G_{\max,n}$$
$$\Delta K_n < \Delta K_{\max,n}$$

(6-5)

and the following update is made for the n'th iteration.

$$K_{n+1} = K_n + \Delta K_n$$
$$G_{n+1} = G_n + \Delta G_n$$
$$P_{n+1} = P(G_{n+1}, K_{n+1})$$

(6-6)

Ideally, the performance vector should remain stationary, if there exists a continuous path along which the perturbation in K can compensate for the perturbation in G perfectly. In actuality, the performance vector gyrates on (or very close to) the surface of the specification boundary. The determination of the valid range of sensitivities and the evaluation of the partial derivatives are computationally intensive and convergence of the routine is expected to be slow, if indeed a solution does exist. Moreover, it turns out that the movement of the performance vector is too restrictive and the range of linearity is often too small for the iteration loop to make any significant modification to the controller parameters. Some elements of ΔG_{max} correspond to the non zero elements of G are often as

small as 10^{-4} compare to the expected target magnitudes of 10's or 100's. After some considerations, this approach is found to be unfavourable.

6.3 Downhill Simplex Method

6.3.1 Downhill Simplex Optimisation Method

The downhill simplex method [Press, Flannery, Teukolsky & Vetterling 92] is a multi-dimensional optimisation of performance index in N variables. This is a self contained strategy in which one dimensional minimisation is not required. This method requires only function evaluation and not derivatives. It is not the fastest method but may be the best if the aim is to get something to work quickly and the computational burden is not too great.

A simplex is a polyhedron with $N+1$ vertices in the N dimensional vector space, not necessarily regular. For example, a simplex in 2 dimensions will be a triangle and in 3 dimensions will be a tetrahedron. In general, only non-degenerate simplexes (those that enclose finite N-dimensional volume) are of interest. If any one of the vertices of a non-degenerate simplex is taken as the origin, the other N vertices define vector directions that span the N-dimensional vector space.

To begin with, the algorithm requires a starting point K_o which denotes an n dimensional vertex. A further n arbitrary vertices can be obtained by

$$K_i = K_o + \lambda e_i \qquad (6\text{-}7)$$

where $i=1.....n$, e_i are the unit vectors and λ is a constant vector. An initial simplex is constructed by joining K_o and the K_i's together. The entries in λ are the (guessed) characteristic length scales of the problem.

The simplex will now "feel" its way downhill by taking one of the following operations (Figure 6-2):

(a) *reflection away from the highest point* : Move the vertex where the function has the highest value through the opposite face of the simplex to a lower point of equidistance so that the volume (and hence non degeneracy) is conserved.

(b) *reflection and expansion away from the highest point* : If a reflection produces a lower point, the simplex is expanded in the same direction so that a larger downhill step can be taken.

(c) *contraction along one dimension from the highest point* : When the simplex reaches a "valley floor", it contracts in the transverse direction and tries to ooze down the valley.

(d) *contraction along all dimensions from the lowest point* : If there is a situation where the simplex is trying to "pass through the eye of a needle", it contracts itself in all directions, pulling itself in around its lowest (best) point.

The termination criteria can be delicate and one suggestion is to terminate when the decrease in the function value is fractionally smaller than some tolerance f_{tol}. It may require a few trial runs before a suitable criterion can be chosen.

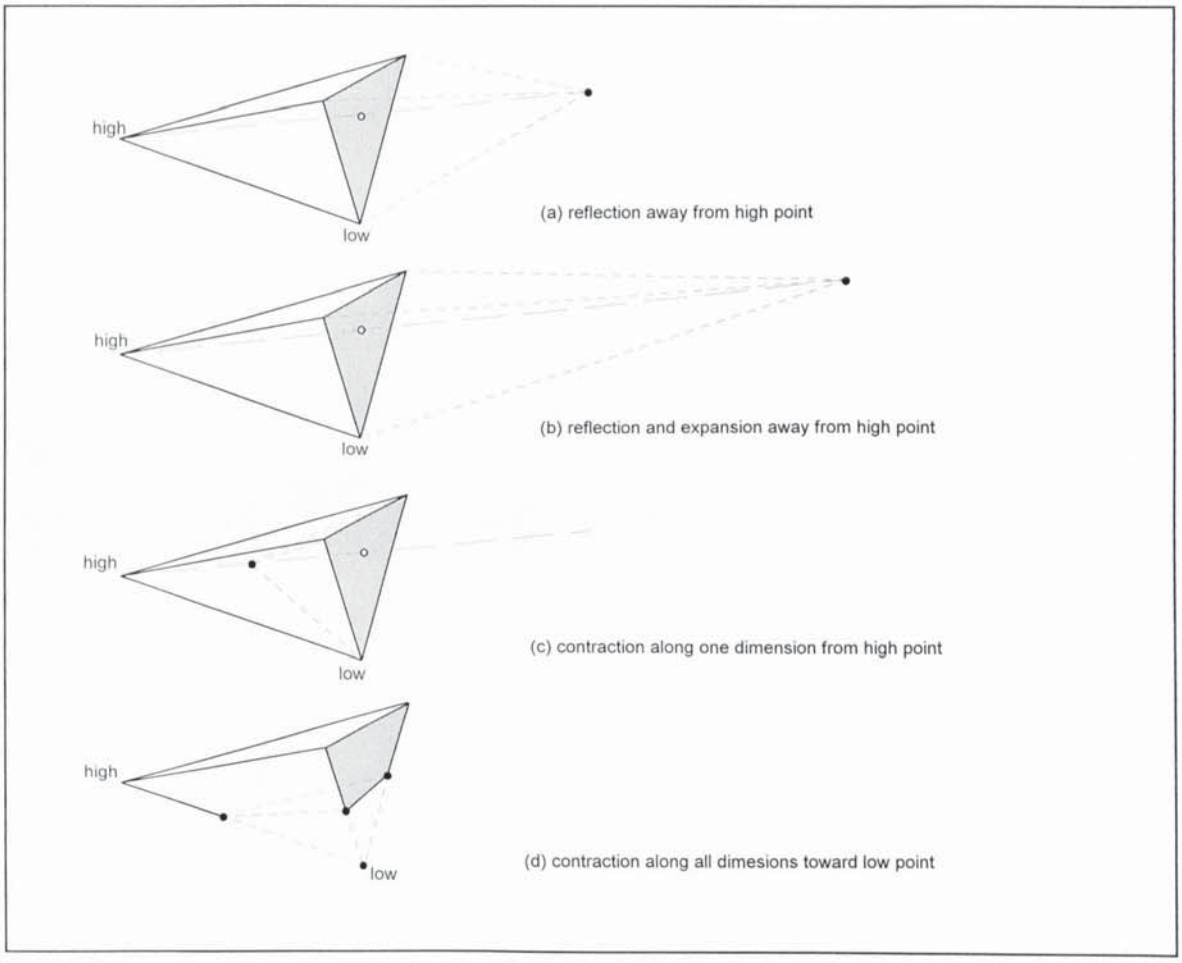

Figure 6-2 Behaviour of Simplex in 3 Dimensions

6.3.2 The Amoeba Algorithm

A stand alone routine which implements the downhill simplex method is given in [Press, Flannery, Teukolsky & Vetterling 92]. However, this routine is coded in 'C' and it is translated into MATLAB script, with the necessary modifications, to ensure a better control over its behaviour (see Appendix).

As in many of the optimisation problems, the performance index is a scalar but the performance specification is multidimensional. A weighting matrix is therefore required to balance the emphasis on the different aspects of the requirements. In fact, the definition of the performance index is the most important input that the designer can make. This is where the designer's insight of the engineering properties of the system becomes very important. The main advantage is that the process guarantees a (local) "optimal" solution subjected to the definition of the performance cost.

6.3.3 A State Feedback Decoupling Controller by Amoeba

As a starting point of the optimisation process, an LQ state feedback controller is designed for the benchmark plant with

- all inherent coupling set to zero so that the plant has only non interacting axes
- the uninvolved input and output of axis B discarded but all the states of the model retained so that the optimisation is performed on an 18 rather than a 27 dimensional manifold

and the result is a 2-by-9 matrix which has a transpose of:

```
fstkc' =  1.0e+003 *
  0.00000000000251  0.00000000000303
  0.00000000000000  3.16227766016837
  3.16227766016839  0.00000000000000
  0.00000000000000  0.38194205291771
  0.00000000000000  3.10590831123194
  0.00000000000409  0.00000000000494
  0.00000000000900  0.00000000001086
  0.38194205291771  0.00000000000000
  3.10590831123196  0.00000000000000
```

This will be referred to as the *non interactive solution* and will be taken as the first vertex of the initial simplex. The optimisation will now be performed over the benchmark model with full interactions (i.e. all inherent couplings set back to the nominal values).

The performance vector, p(G,K) is defined as follows

$$p(G,K)=\left[\frac{1}{N}\sum_{t=0}^{T}\left[\left[\frac{y(t,G,K)}{y_{ss}}-r_h(t)\right]\cdot t^3\right]^2\right]^{\frac{1}{2}}$$

(6-8)

where

- t is a discrete time series with N equidistant elements in the interval [0,T]
- T is the typical settling time of the step response of the corresponding independent axis
- $r_h(t) = [r_1, r_2]$ is a step reference signal vector
- $y(t)$ is a 4-vector $[y_1, y_2, y_3, y_4]^T$ with y_1, y_2 as the output response to the reference input r_1 and y_3, y_4 the output responses to the input r_2 for a particular set of plant G (which remains constant throughout this particular optimisation)
- K is the controller
- y_{ss} is a constant 4-vector with

$$y_{ss,1} = y_{ss,2} = y_{1,\,first\ vertex}(T)$$
$$y_{ss,3} = y_{ss,4} = y_{4,\,first\ vertex}(T)$$

(6-9)

which is the steady state response of the initial LQ controlled plant with independent axes (without inherent coupling). This represents a reasonable degree of steady state gains to which y(t) is normalised. Also, in an effort to discourage any unstable behaviour, the step response error is multiplied by t^3. In summary, the performance vector of this particular optimisation is the root mean square of a normalised step response error weighted by t^3. The performance index, P.I., however, has to be a scalar and a further definition

$$P.I. = p^T W p$$

(6-10)

where $W \in \Re^4$ is a 4-by-4 weighting matrix, provides such a number. Fine tuning of the performance requirements can be made by adjusting W.

As mentioned before, the termination criteria can be delicate. From the point of view of the algorithm, a natural termination criterion can be related to the machine precision (in MATLAB for windows 3.11, eps, the smallest number defined, has a default value of 2.22 x 10^{-16}) on which the optimisation is performed. From the point of view of the problem, however, the model is only an approximation of reality since the physical parameters are quoted to only 4 or 5 significant figures. This means that no useful information can be gained for the optimisation to go beyond this particular scale. If the function's terrain has a significant structure in this scale, the resulting controller may be too sensitive to parameter changes and the optimisation will have to be reformulated. Bearing these points in mind, a few trial runs were made with reference to the decoupling part of the benchmark problem and the optimisation is set to terminate when one of the following conditions becomes true;

- r_{tol}, the index of the "shape" as defined in equation (6-11) below, falls below 10^{-3} which indicates that the simplex is lying flat on a "valley" and no significant gain can be made through iteration
- number of iteration becomes too large (i.e. greater than 10000) and should be terminated gracefully for inspection
- limit of numerical accuracy has been reached

After 1766 evaluations of performance index, r_{tol} dropped below 10^{-3} and the process terminated with the controller modified to:

```
kc' = 1.0e+003 *
   5.42626437893150  -0.40942075930002
   1.55822886821556   2.27123292995936
   1.60388814781357   0.89115912492173
  -0.02329725596817   0.62449963851690
   0.24318489024252   2.39091595936341
   0.02226715095221  -0.07316764470323
  -1.74083101088200   0.06727257857515
  -0.01748039175206   0.65147926783649
   4.50231187346294  -3.24312568759037
```

The changes in the 18 controller parameters are shown in the bar charts in Figure 6-3.

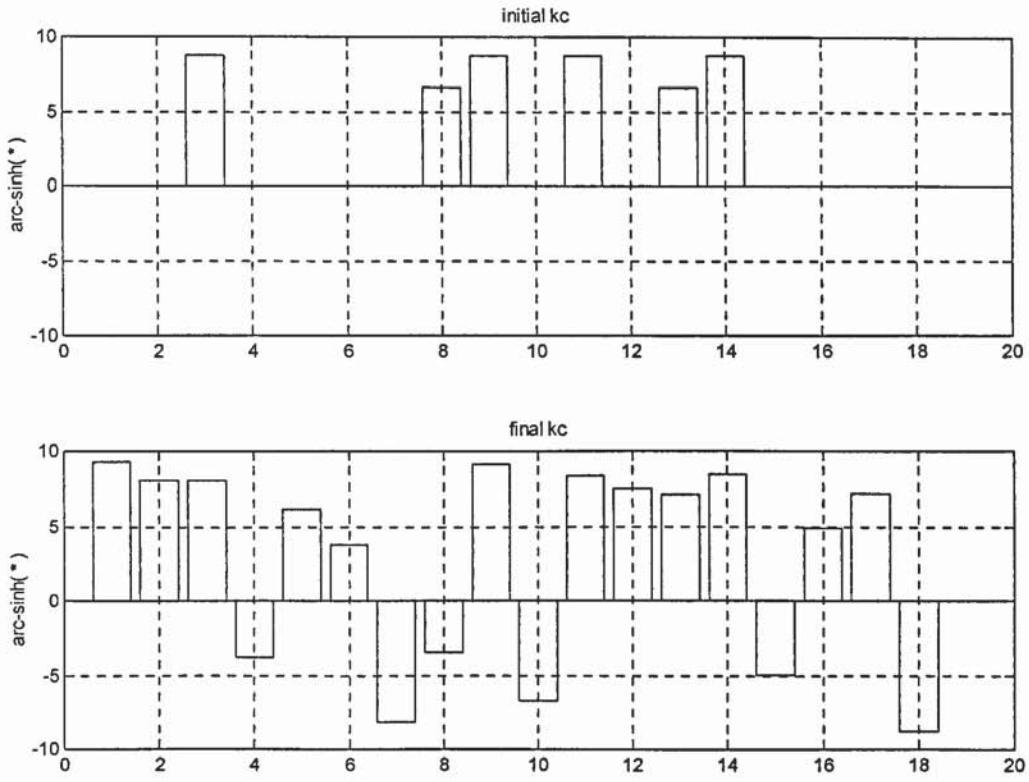

Figure 6-3 Arc-Sinh (compresses range, preserves sign) of Controller Coeffiencents

To monitor the progress of the optimisation process, the performance index (P.I.) and the shape of the simplex (represented by another index r_{tol} as defined below) are recorded for each iteration which give some indications as to how the simplex behaves over time.

$$r_{tol} = \frac{P.I.(\text{hightest point}) - P.I.(\text{lowest point})}{P.I.(\text{hightest point}) + P.I.(\text{lowest point})} \qquad (6\text{-}11)$$

In a way, r_{tol} shows the progress made by the simplex relative to the local scale of details. An inspection of Figure 6-4 and Figure 6-5 shows that the sharpest reductions of the logarithmic performance index occurs when the shape of the simplex is most "elongated" in the direction of optimisation, that is, when $\log(r_{tol})$ is at its highest peaks in the plot.

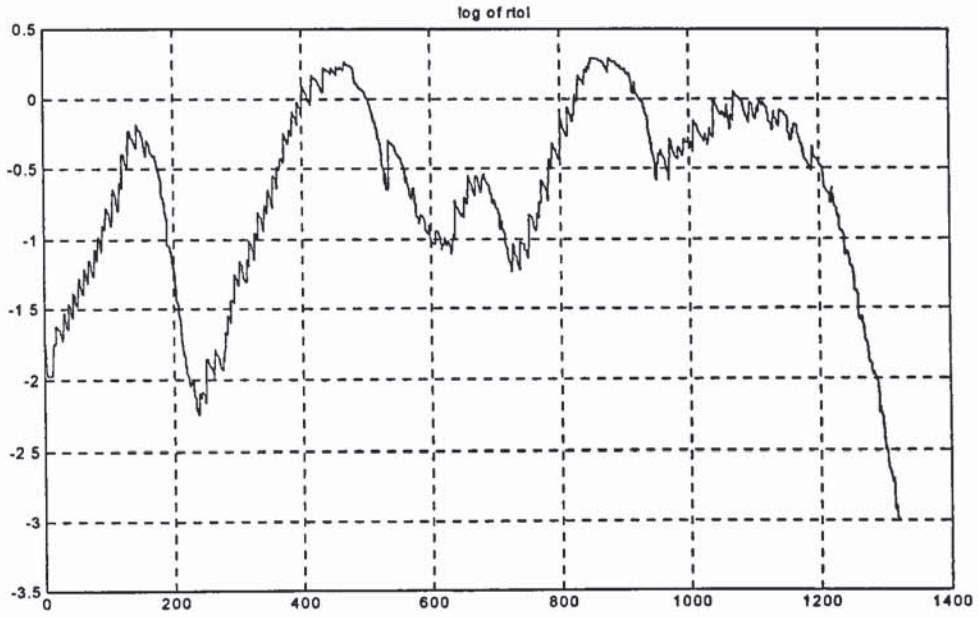

Figure 6-4 Tracking the Shape of the Simplex Over Each Iteration

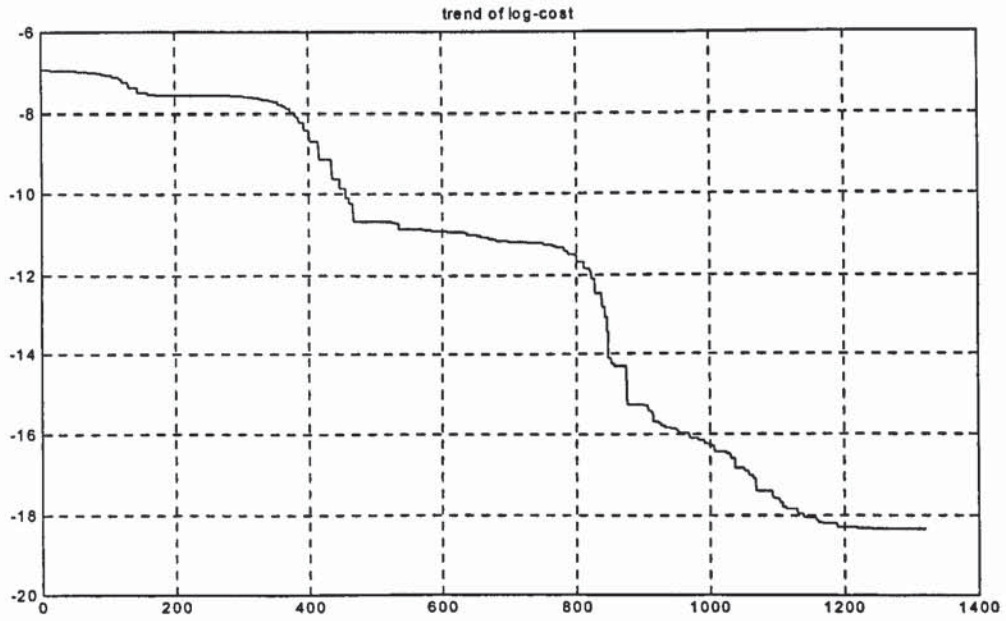

Figure 6-5 Log of Performance Index Over Each Iteration

In Figure 6-6 and Figure 6-7, the step responses of the plant under the LQ controller and the modified controller are shown respectively. Full steady state decoupling is achieved in under about 0.1 sec with the new controller.

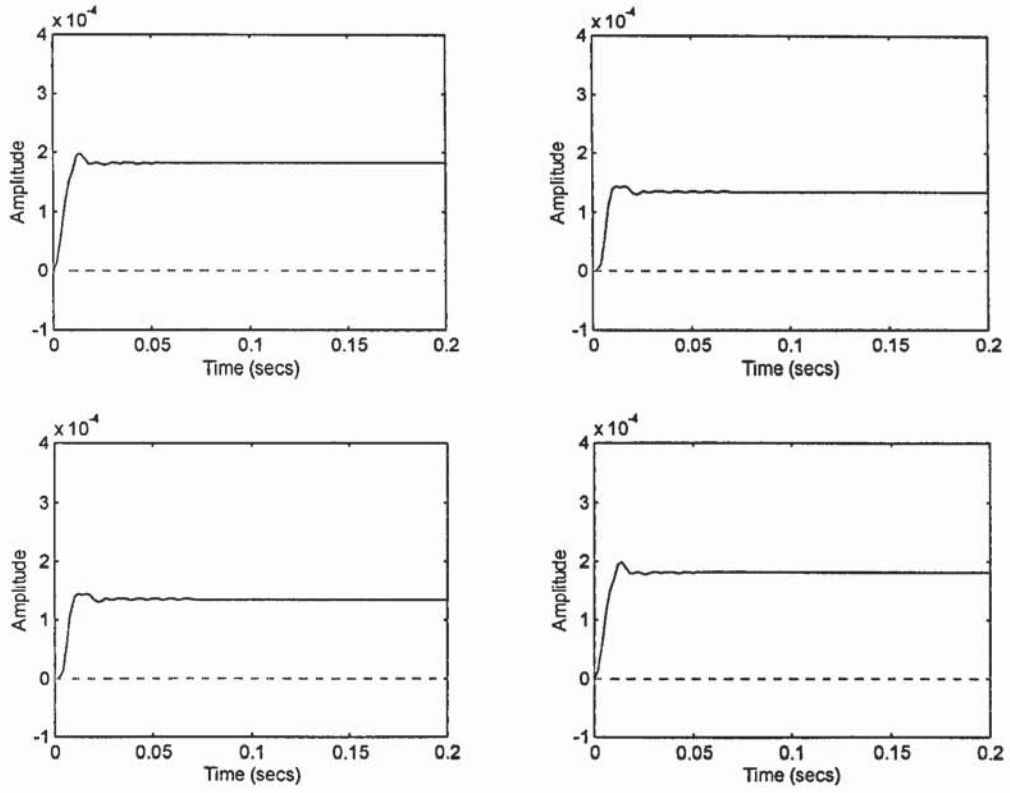

Figure 6-6 Step Response of the Coupled Plant (Axes A & C) Under LQ Control

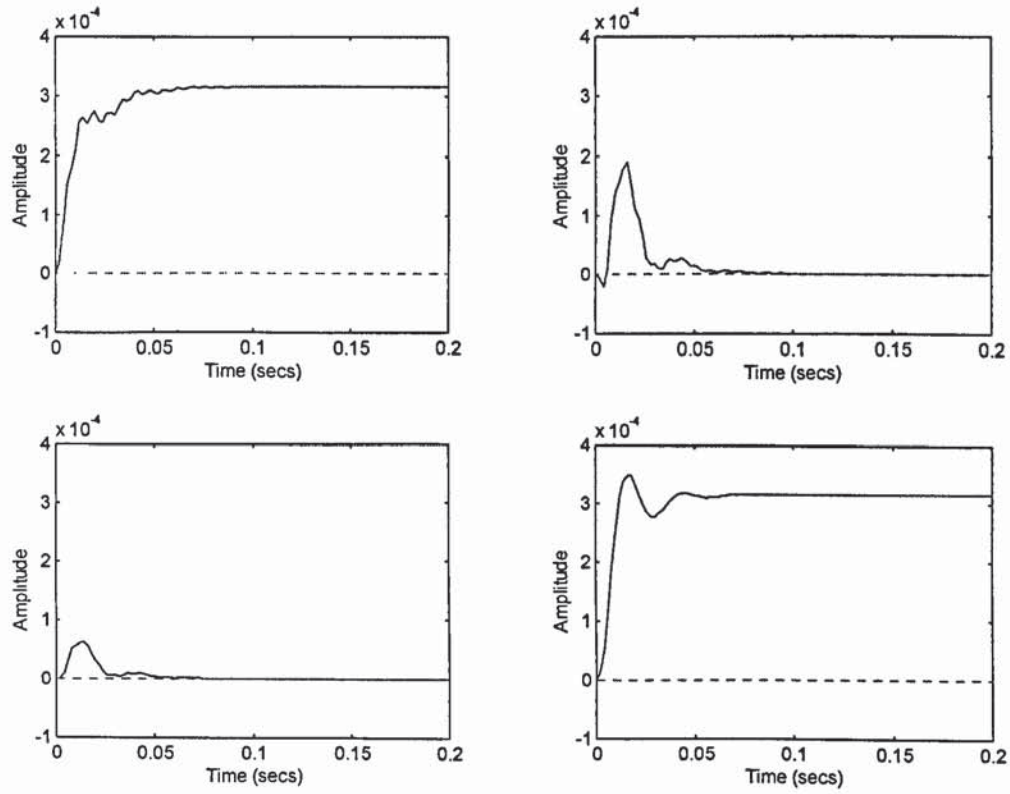

Figure 6-7 Step Response of the Coupled Plant with Controller Modified by Amoeba

Figure 6-8 shows how the step responses change over the optimisation process. Notice that apparent oscillations occur at the early stage of the optimisation. In fact the optimisation was terminated a few times manually by the author before it reached the termination conditions for the fear that the process is "running away". There was no obvious way to tell if the (apparently unstable) oscillations would settle eventually, since one is well aware that conflicting requirements were to be balanced by the optimisation of a scalar number. The details of this required balance can be very complicated when the number of axes involved become larger and hence the possible conflicting requirements becomes even larger. Great care has to be taken in translating the specification into performance indices.

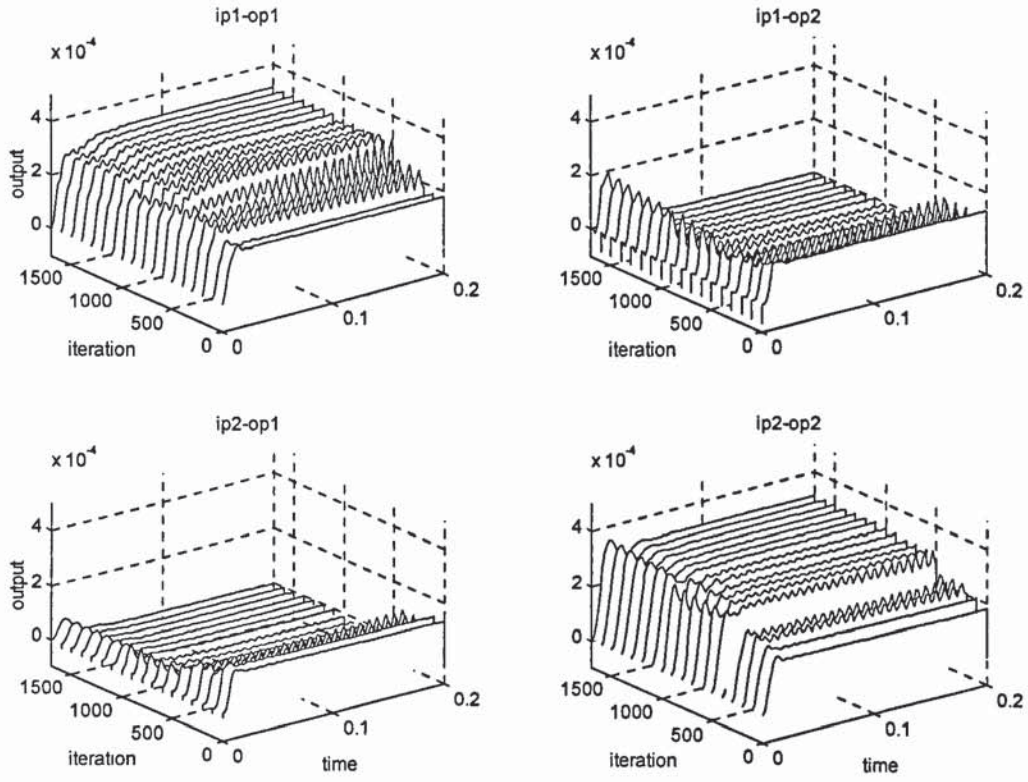

Figure 6-8 Trend of Step Responses During Optimisation

The BRG arrays for the closed loop system under LQ control and under decoupling control are shown in Figure 6-9 and Figure 6-10. Recall that the perfectly isolated subsystem has an identity matrix as its BRG, the BRG's of axis A and C each was reduced from about 7dB to 0dB for the low frequencies. This decoupling controller obtained by the simplex method is far superior to the controller obtained by the other decoupling methods investigated so far. Perhaps this is not surprising for the following reasons: Firstly, the decoupling controller is obtained by a "modification" of a LQ state feedback controller

and in general a closed loop controller performs better than an open loop compensator (c.f. chapter 5). More importantly is the fact that the step response error is taken directly as the performance index and hence the controller is optimised to perform well for this particular task. The implication is that this controller may not perform as well when the requirements are changed. For the same reason, however, one may argue that a separate optimisation should be performed with a different performance index, which is tailored to the changed requirements, in order to obtain another controller.

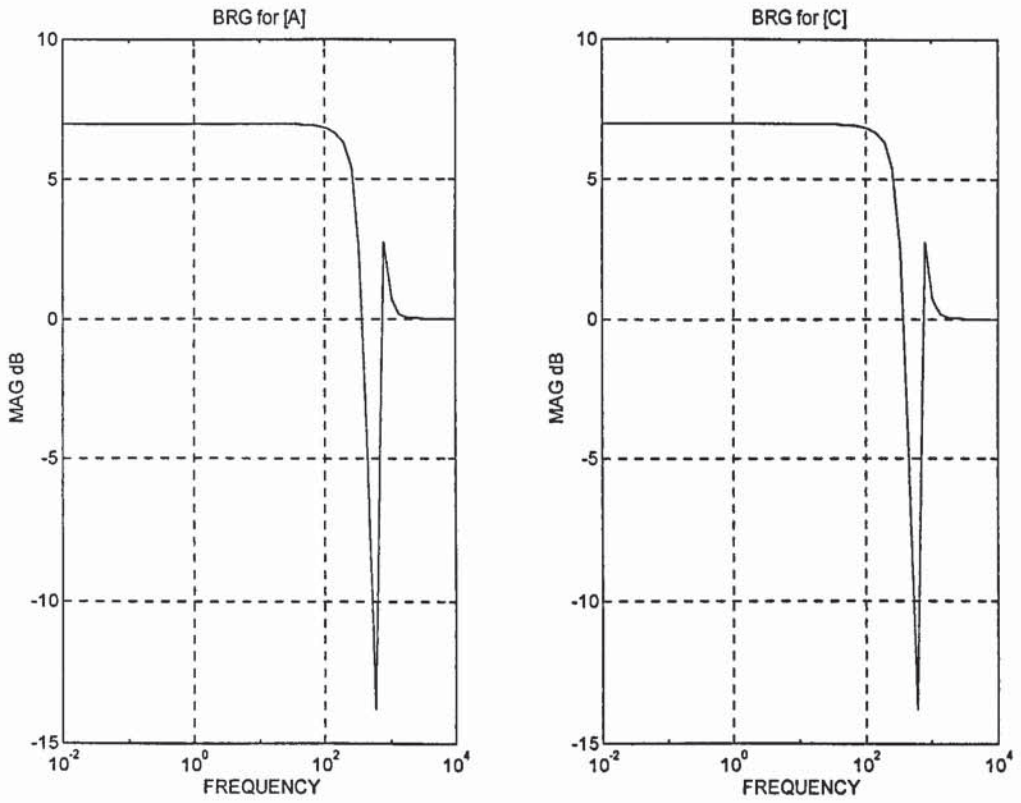

Figure 6-9 BRG Arrays for Individual Axis Under LQ Control

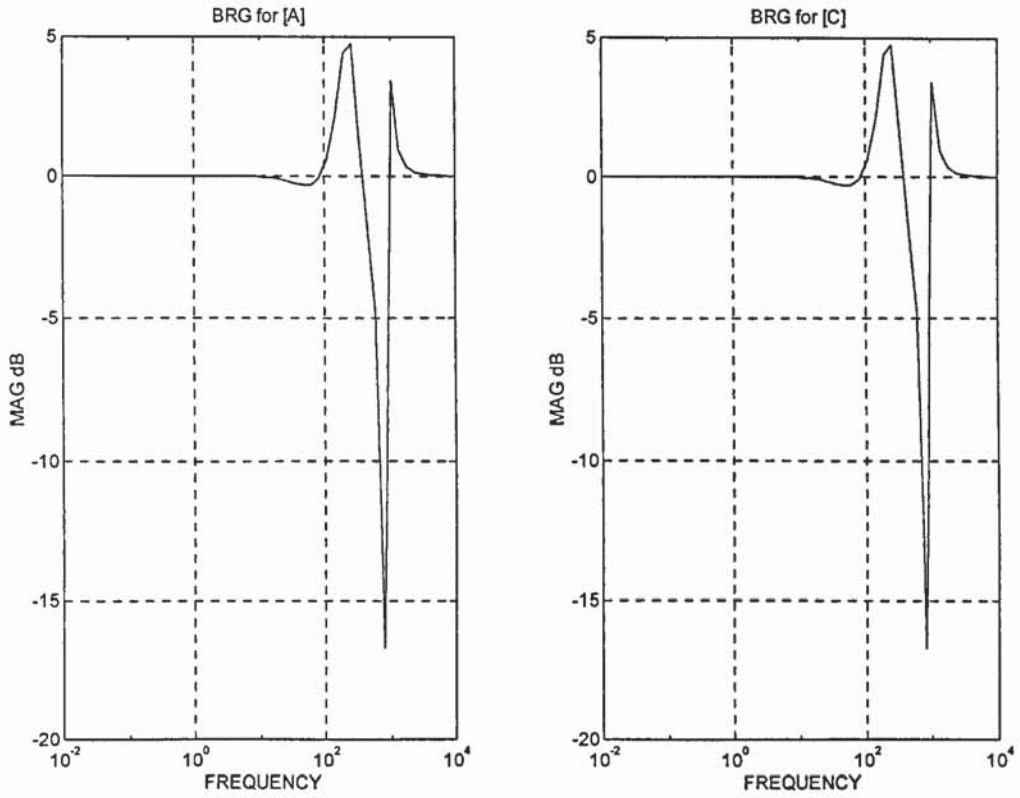

Figure 6-10 BRG Arrays for Individual Axis with Controller Obtained from Amoeba

6.3.4 Symmetric vs. Asymmetric Solution

An inspection of the controller gain and the step responses (Figure 6-7) of the control system obtained in the previous section shows that the feedback matrix K_c is not symmetrical although the plant model under control is. This asymmetry is a result of the inherent redundancy built into the optimisation process and is reflected by the existence of level hyper-planes in the optimisation landscape. Since the order of the states within the state vector is generated automatically by MATLAB and cannot be specified by the user, this implies some difficulties when the solutions generated has to be verified.

Further investigation is conducted by looking specifically at only the axes involved in the decoupling problem. This is achieved by repeating the simplex optimisation with the three axes model but setting the parameters related to the non participating axis to zeros. The sensitivity of the cost function towards the parameters are plotted in Figure 6-11 and Figure 6-12, which indicate the complexity of the landscape in the principal directions at the initial K_c obtained for the new model.

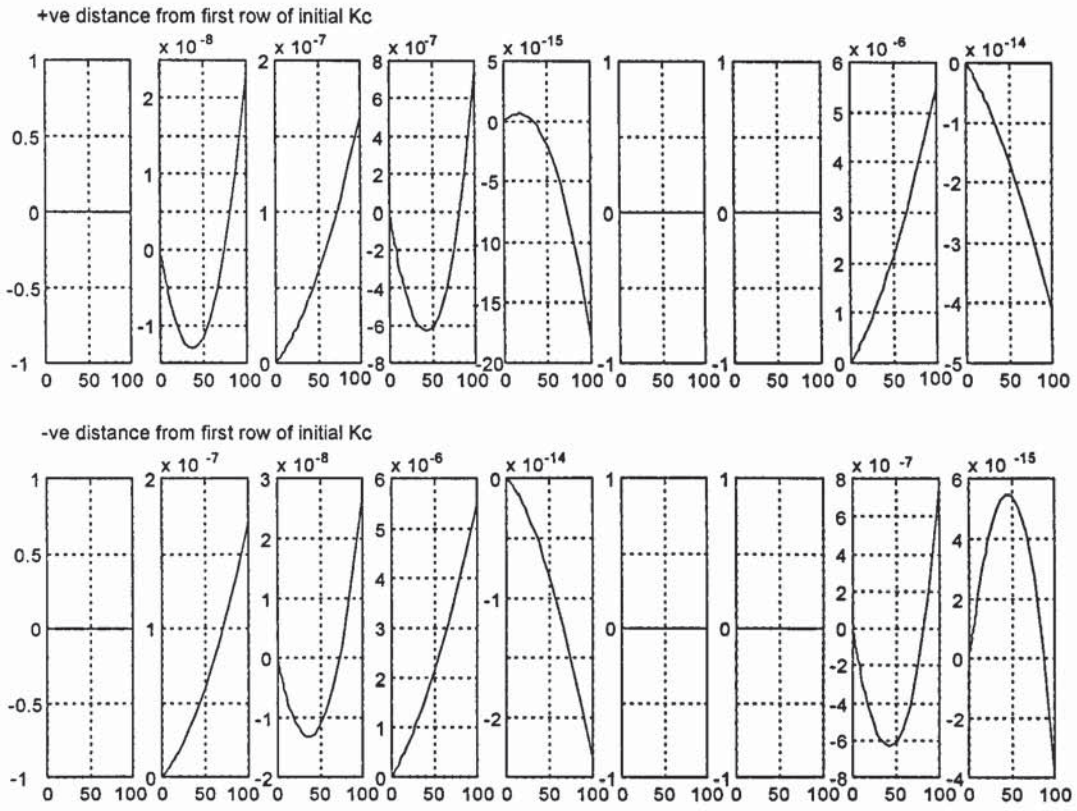

Figure 6-11 Sensitivity of Cost Function from Variations of the First Row of Initial Kc

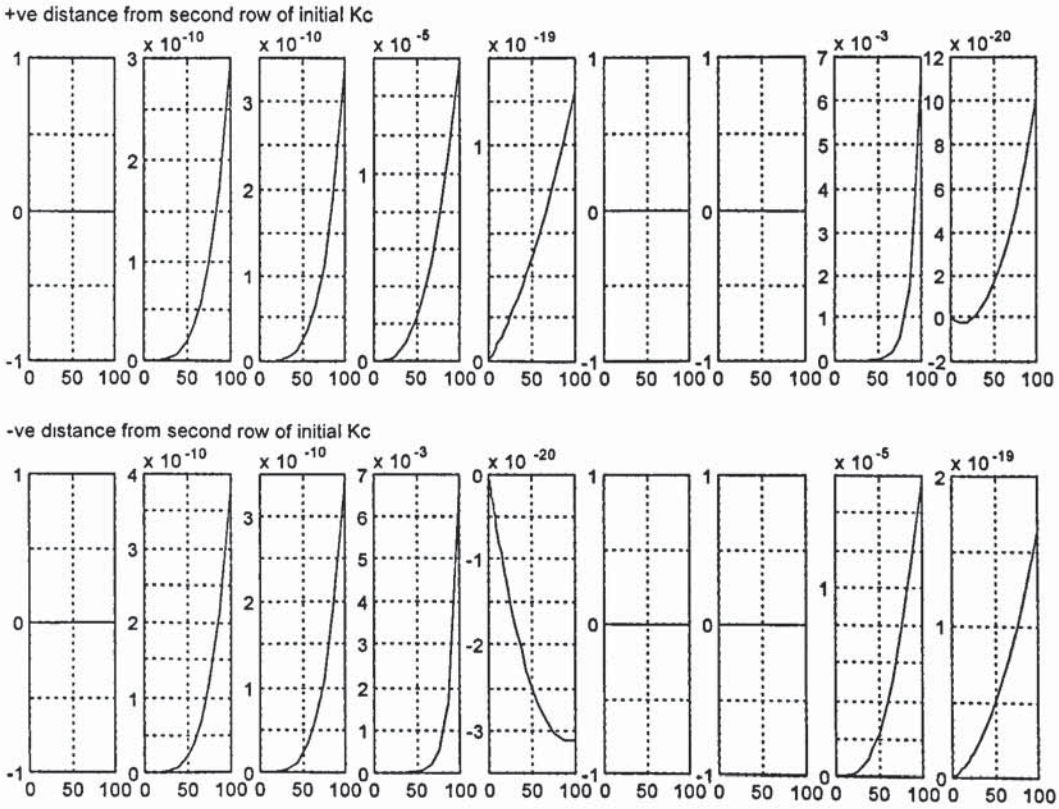

Figure 6-12 Sensitivity of Cost Function from Variations of the Second Row of Initial Kc

A number of variations of the optimisation strategy were applied and studied. These included changing the termination conditions and varying the starting point for hill descending. After some trial and error, a workable strategy towards a symmetrical solution is developed by virtue of the fact that any matrix can be represented by the sum of its symmetric and asymmetric parts. A symmetrical decoupling solution is obtained by successively re-initialising the amoeba routine in the following steps:

1. start downhill simplex as normal.
2. repeat *n* hill descending loops.
3. re-initialise the simplex to the symmetrical part of the current K_c and set the non-participating parameters to zeros.
4. iterate to step 2 unless terminating condition is reached.
5. take symmetrical part of the current K_c as final solution.

After 3745 performance cost evaluations, the optimisations terminated when the r_{tol} falls below 10^{-4} and the following controller gains are produced. The symmetry is readily observable in the bar chart in Figure 6-13. The effects of the re-initialisations can be seen more graphically in Figure 6-14 where the cost function is plotted and in Figure 6-15 where the changes in the shape of the simplex is traced. For completeness, the step responses of the symmetrically decoupled system is shown in Figure 6-16.

```
kc' = 1.0e+003 *
         0         0
  1.71597888569204  1.44632608072983
  1.44632608072983  1.71597888569204
  0.27197285558463  0.28554128990372
  0.59469082187451  4.24145039658015
         0         0
         0         0
  0.28554128990372  0.27197285558463
  4.24145039658015  0.59469082187451
```

This exercise concluded that a symmetric solution is obtainable. Nonetheless, an asymmetric solution is just as acceptable since it is also obtained by minimising the same cost function. The performance, be it not symmetrical in its response, is still considered meeting the criteria. In general, generic models under considerations have non trivial and complicated geometry and the skewness of performance introduced may not be easily spotted. However, a solution is ultimately judged acceptable only if the primary

performance specification such as absolute and relative tolerance in position errors are met but not how balanced the control efforts were. As a result, the need of a symmetrical solution is only secondary and will not be pursued further here.

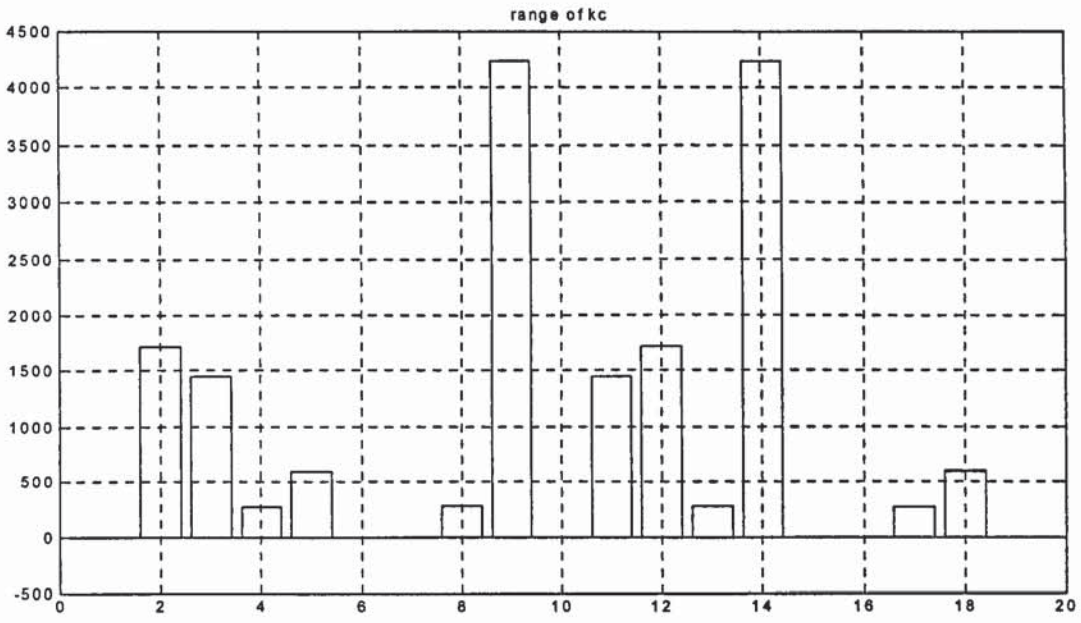

Figure 6-13 Range of Parameters

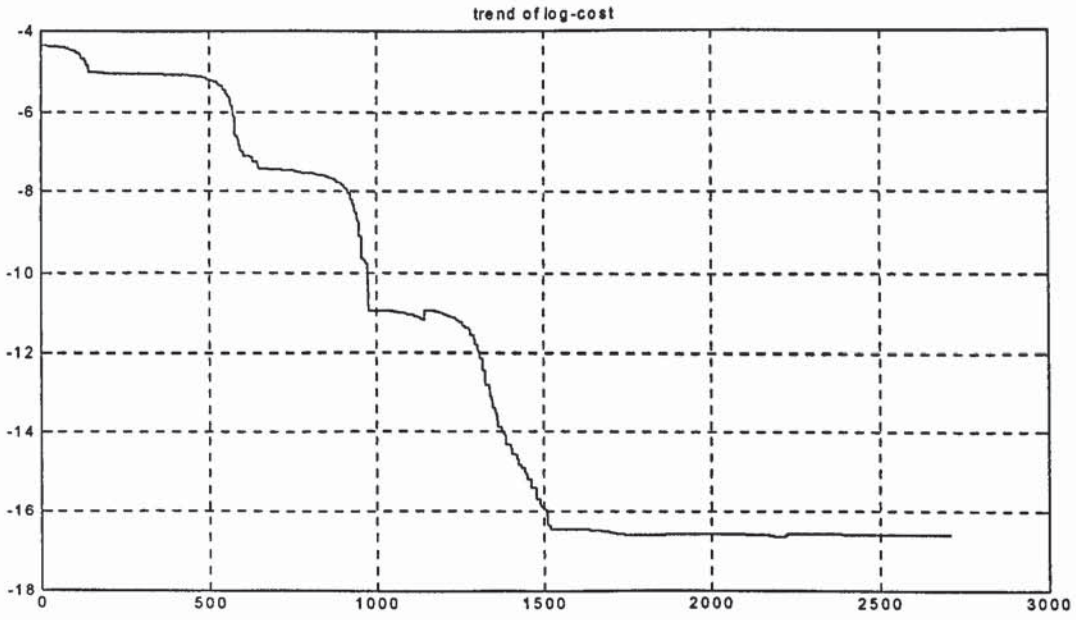

Figure 6-14 Value of Cost Function over the Optimisation

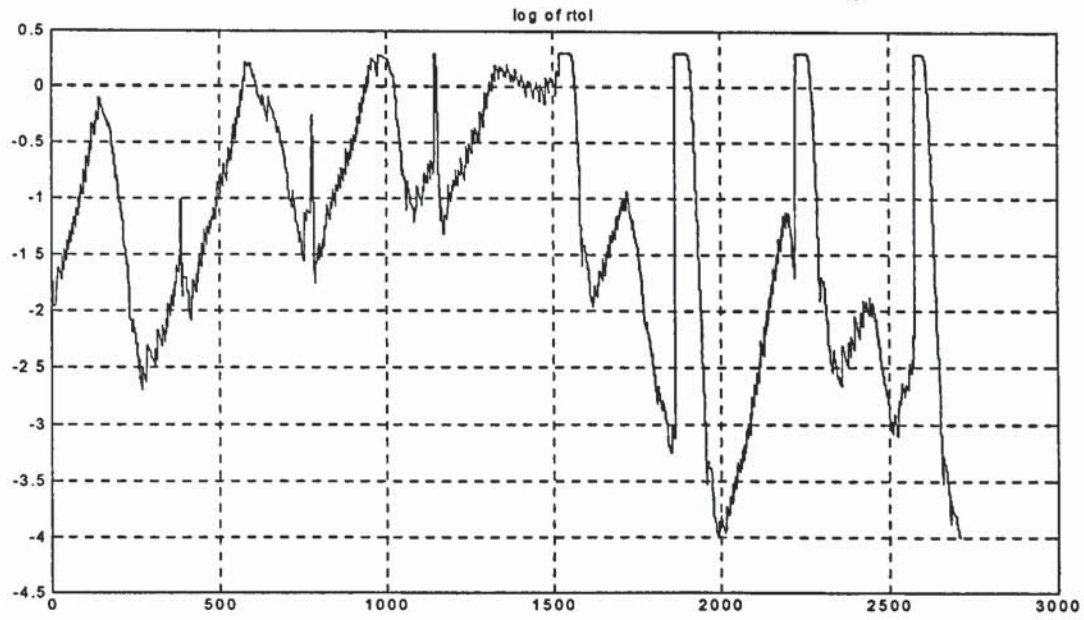

Figure 6-15 Shape of Simplex over Optimisation with Peaks due to Re-initialisation

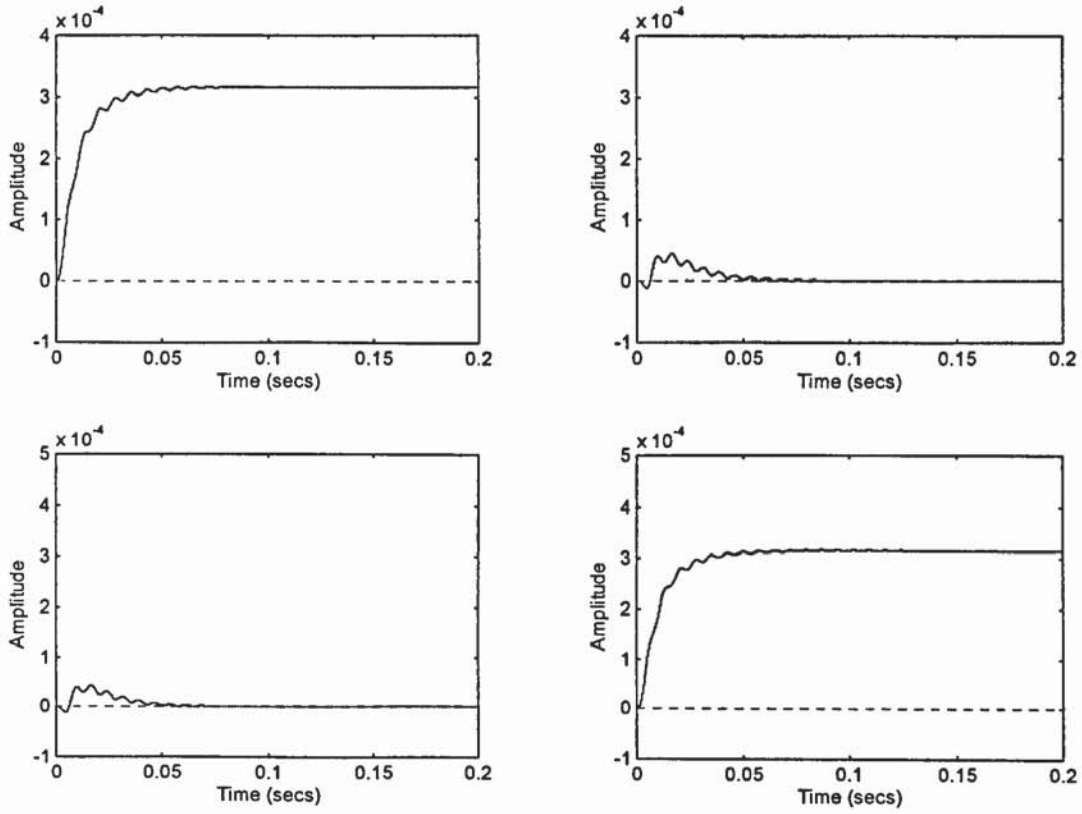

Figure 6-16 Step Responses of the Symmetric Solution

6.3.5 The Decoupling Controller for a 3 Axes Model

At first axis B seems to interfere with the optimisation of the decoupling of axes A and C, even when it should not be involved. The problem was then simplified by discarding the input and output of axis B and the result is presented in the previous section. However, it is important that the controller synthesis procedure can cope with a large number of axes without the resultant controlled response skewing towards one way or the other. For the sake of automation and simplicity, it is preferable that the designer does not have to consider the configuration of the overall interaction and tailor make the optimisation once these interactions were entered into the model. This aim, however, can only be achieved if the optimisation can be performed within a reasonable timeframe and still retain numerical accuracy.

An LQ state feedback controller is designed for the benchmark plant with all inherent coupling set to zero and the result is a 3-by-9 matrix. This will be refereed as the *non interactive* solution and will be taken as the first vertex of the initial simplex. The optimisation will now be performed over the benchmark model with full interactions (i.e. all inherent couplings set back to the nominal values). After repeated adjustment to the way the performance index was defined, it seems to work fine.

```
fstkc' = 1.0e+003 *
  0.00000000000000  2.23606797749979  0.00000000000000
  0.00000000000000  0.00000000000000  3.16227766016839
  3.16227766016837  0.00000000000000  0.00000000000000
  0.00000000000000  0.00000000000000  0.38194205291771
  0.00000000000000  0.00000000000000  3.10590831123196
  0.00000000000000  0.18050102691887  0.00000000000000
  0.00000000000000  0.90297855988297  0.00000000000000
  0.38194205291771  0.00000000000000  0.00000000000000
  3.10590831123195  0.00000000000000  0.00000000000000
```

The performance index is defined similarly as in equations (6-8) and (6-10), but with y(t) now a 9-vector, with entries $[y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9]^T$, correspond to the 3 sets of cross channel input/output responses, and y_{ss} the normalisation vector with entries

$$y_{ss,1} = y_{ss,2} = y_{ss,3} = y_{1,\textit{first vertex}}(T)$$
$$y_{ss,4} = y_{ss,5} = y_{ss,6} = y_{5,\textit{first vertex}}(T)$$
$$y_{ss,7} = y_{ss,8} = y_{ss,9} = y_{9,\textit{first vertex}}(T)$$

(6-12)

After 10,000 iterations, the optimisation terminated with the following modified controller as a result and the changes in the 27 controller parameters are shown in Figure 6-17. The changes in the performance are also recorded in Figure 6-18.

```
kc' = 1.0e+003 *
 -0.00007454520206  2.23657720054438  0.00020976423204
  1.27788190672139 -0.59042698514564  2.86100718947012
  1.88519712029785  0.59109326352858  0.30173467059020
 -0.13090344634061  0.44184784362896  0.41865616289673
  0.30701863995118 -1.63135438180615  3.33108750740679
  0.19495331151791  0.35725572276014 -0.14498338373972
 -1.39897812192994  1.33100507175732  1.82751652585730
 -0.13887549375109  0.41281748525538  0.47459587642718
  3.49983889658276 -0.91149798091246  0.26300707665430
```

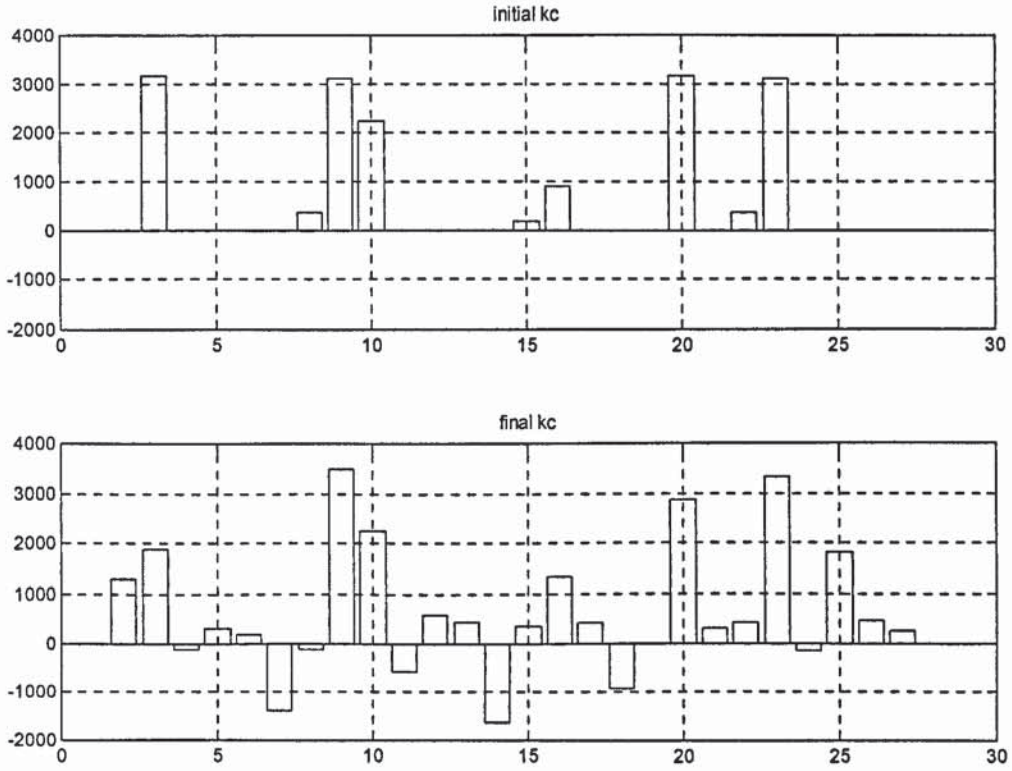

Figure 6-17 Change in Controller Parameters

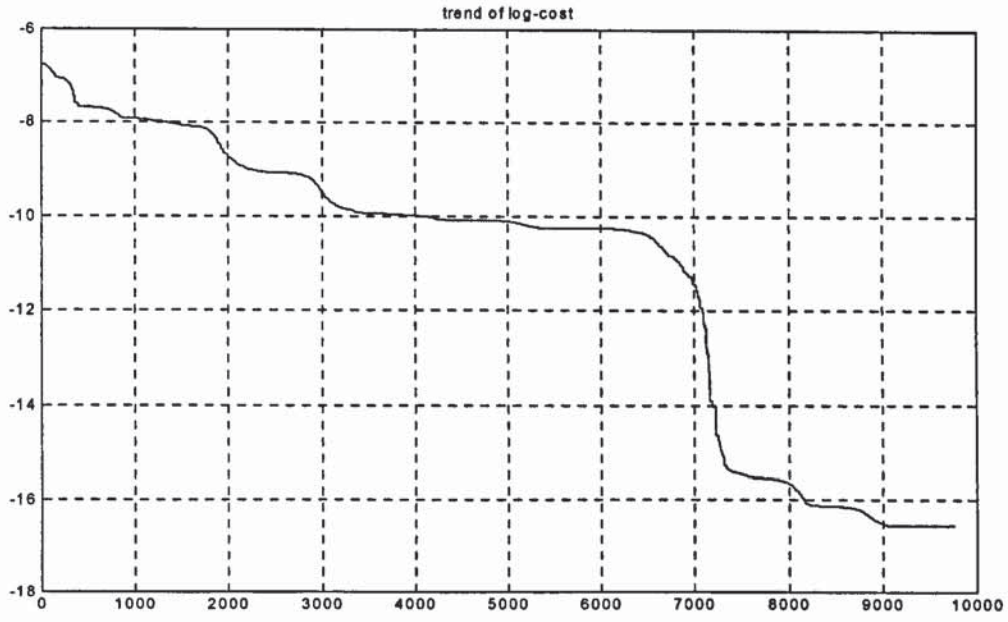

Figure 6-18 Log of Performance Index

In Figure 6-19 and Figure 6-20, the cross channel step responses of the benchmark plant under non interactive and decoupling control are shown. In each diagram, the first row is the responses of all the 3 axes when a step input is presented to axis A. The other 2 rows are defined similarly. It is clear that steady state decoupling of axes A and C is achieved in about 0.1 sec (similarly in the previous section) but some disturbance has spilled over to axis B, which is recovered in a similar time frame. An inspection of the BRG's of the individual axis before (Figure 6-21) and after (Figure 6-22) the optimisation shows the that the level of interactions between them has been greatly reduced. Furthermore, the BRG's in Figure 6-23 and Figure 6-24 show how the subsystem [A B] becomes relatively acceptable as its interaction with the rest of the system is vastly reduced. This of course is subjected to the required bandwidth of the system as the interaction in the higher frequencies, especially around the resonance frequencies, has increased. Perhaps as a trade off, axes A and C as a subsystem, acquired more interaction with the rest of the system after the optimisation.

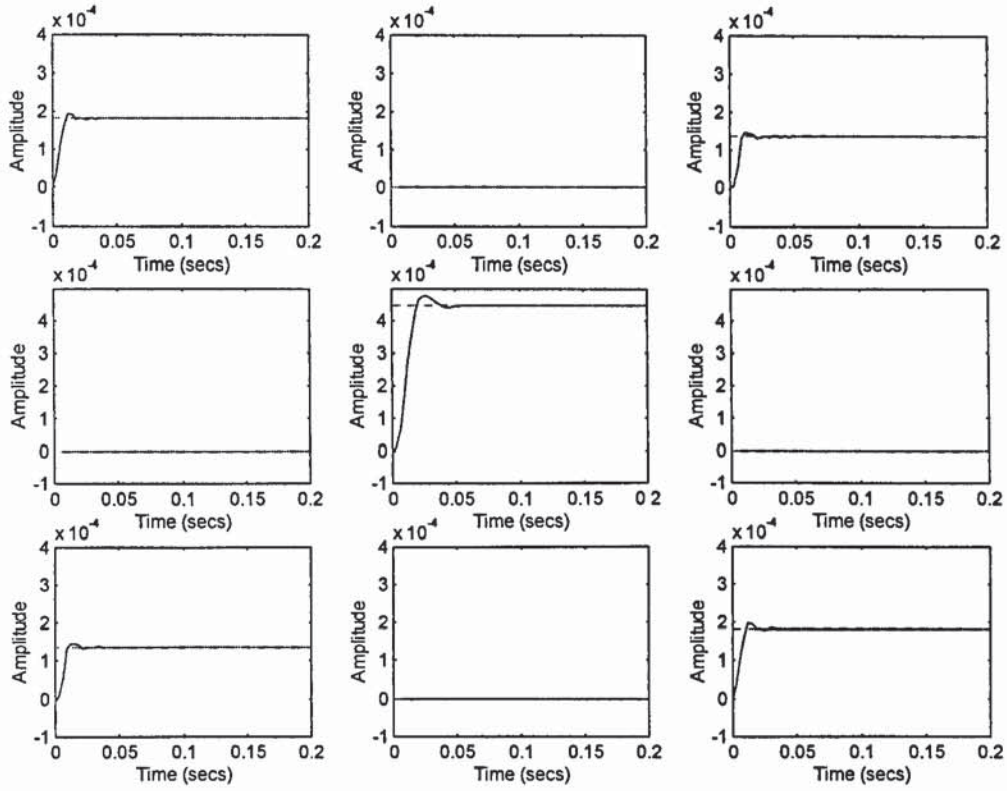

Figure 6-19 Step Responses of LQ Non Interactive Controlled Plant

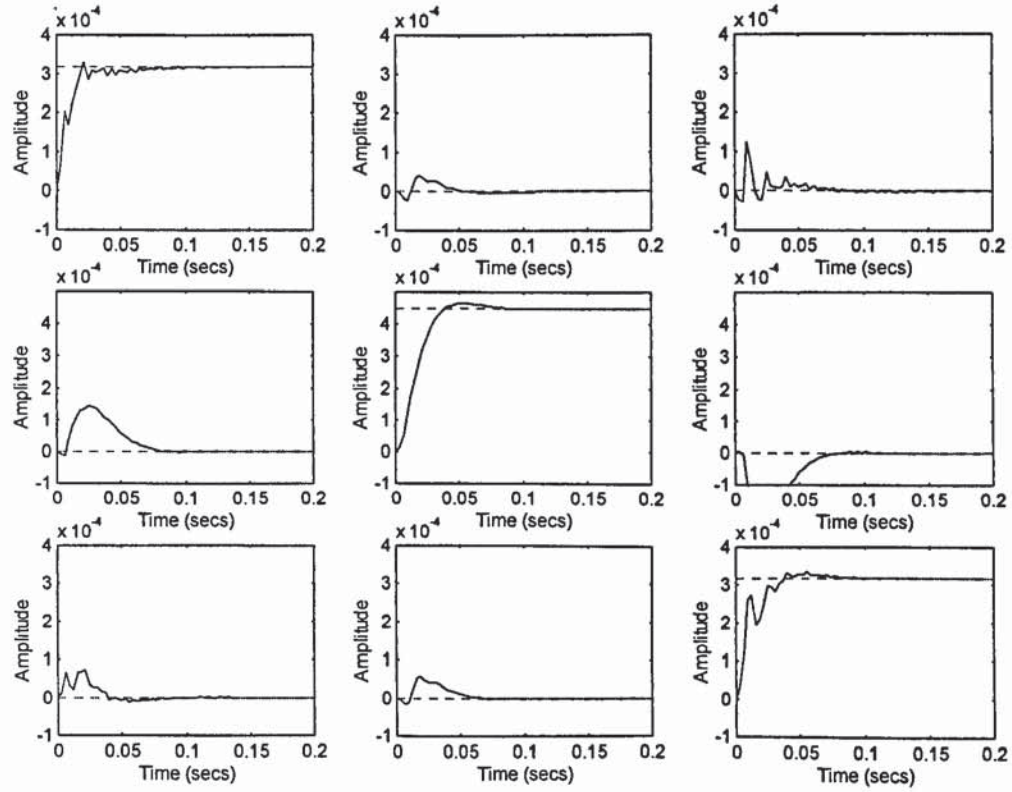

Figure 6-20 Step Responses of Decoupled Plant

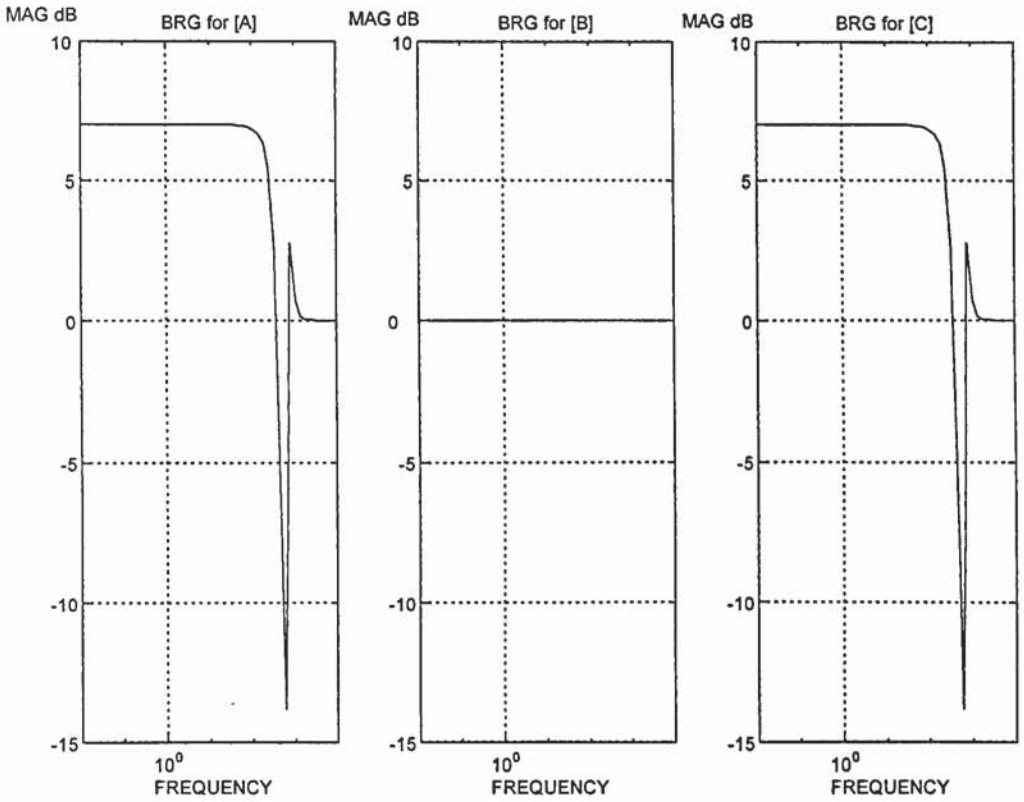

Figure 6-21 BRG of Individual Axis -- LQ Non Interactive Control

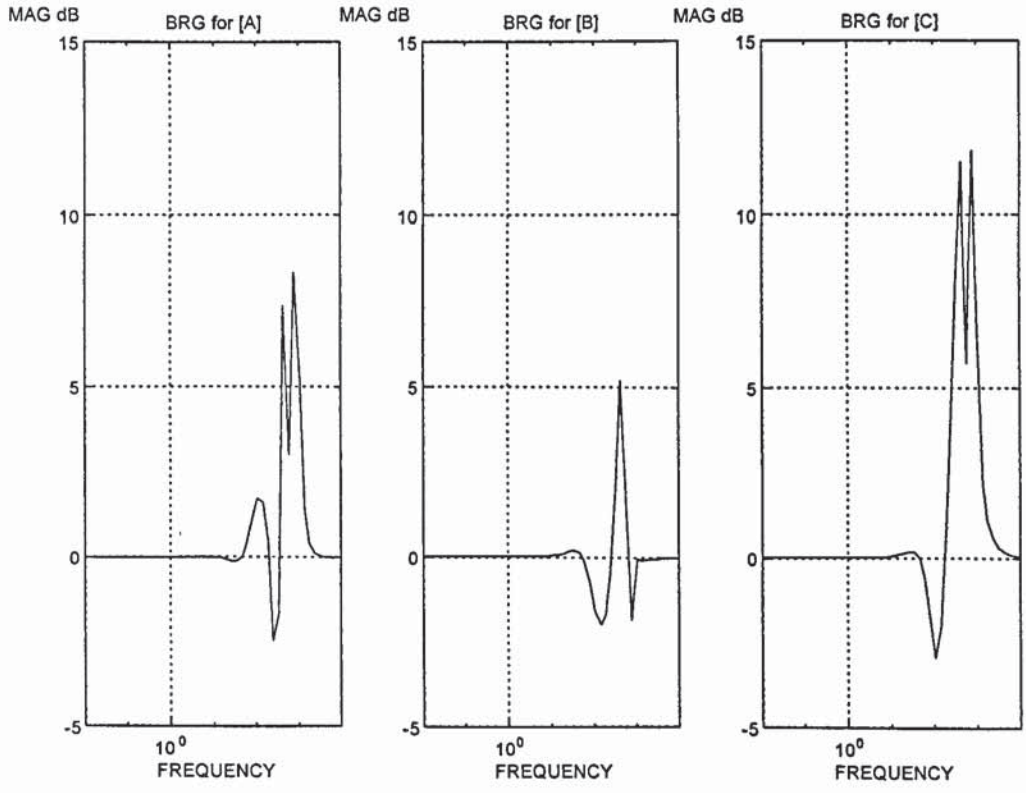

Figure 6-22 BRG of Individual Axis -- With Decoupling Controller

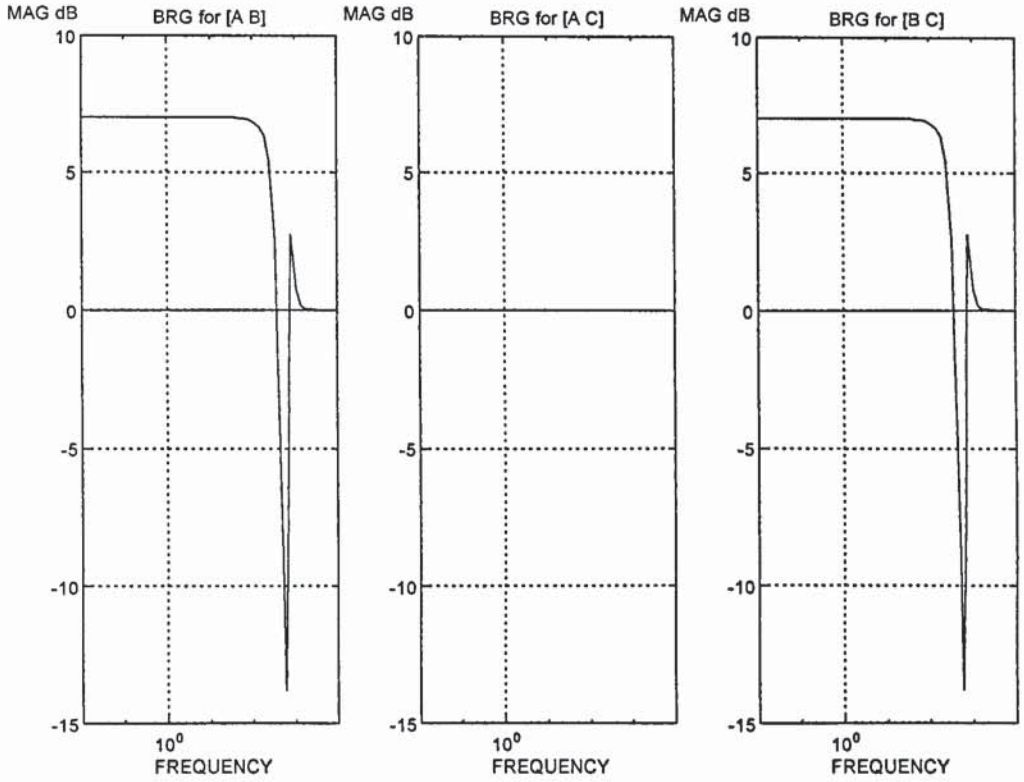

Figure 6-23 BRG Arrays for the 2-axes Subsystems -- LQ Non Interactive

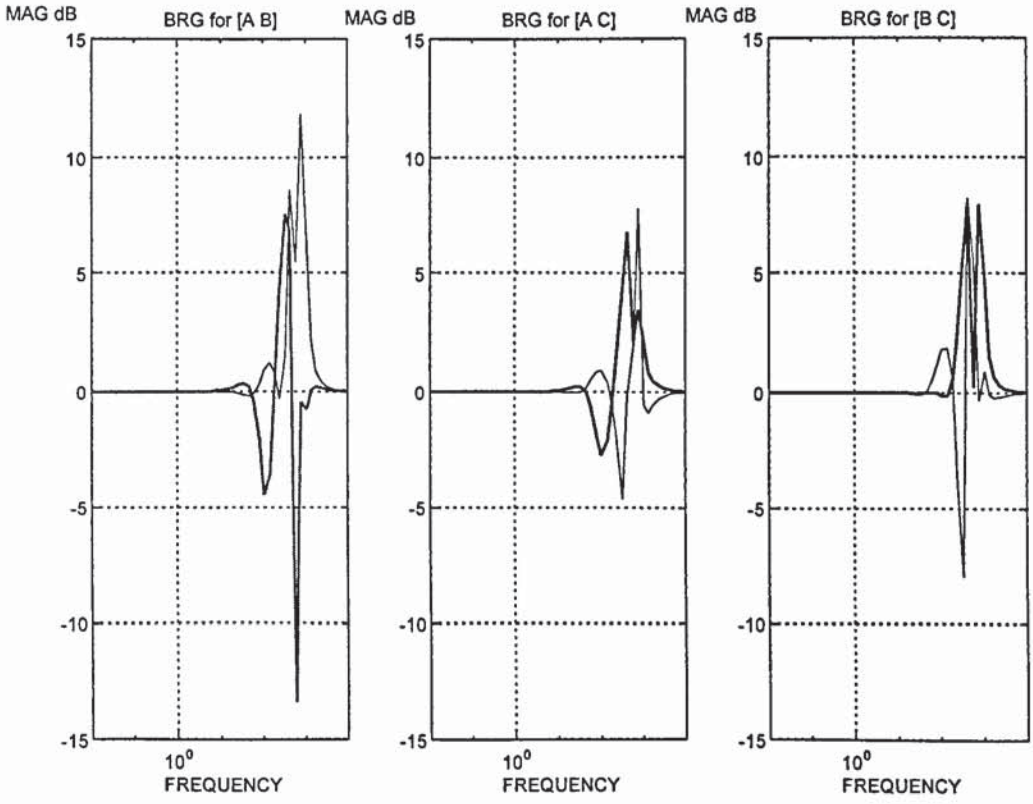

Figure 6-24 BRG Arrays for the 2 Axes Subsystems -- Decoupling Controller

6.4 Synchronisation by Amoeba

Having obtained a good decoupling solution, attention has turned to approaching the synchronisation problem with the same downhill simplex method. Attempts were made to incorporate the synchronisation requirements into the performance index. When there is no synchronisation, the reference used in evaluating the performance index will be set to unity in the direct channels and zero in all the cross channels. In the case of the benchmark problem where synchronisation is required between axis A and B, fictitious reference were used so that each input demand of A and B contribute evenly to the final positions of output A and B.

For example, the actual inputs reference to axis A and B are unit steps but the reference used in the performance index is set to steps of half a unit to both the direct and the cross channel so that total output of axis A will be made up of a half unit response to input A and a half unit response to input B. In this way, any disturbance coming into input A will be spread evenly to both axes A and B and hence the axes are synchronised. Potentially, the desired level of synchronisation can be adjusted by spreading the input with a different proportion, say, a unit response of output can be made up by a sum of 0.7 unit from input A and 0.3 unit from input B.

Figure 6-25 shows the step response of the non interacting plant under the non interacting LQ controller which is taken as the initial vertex of the simplex. Several attempts were made with the performance index constructed as described above so as to synchronise axes A and B but no acceptable solution was reached. In one attempt, the step response after 5000 iterations is shown in Figure 6-26. It can be seen that a limited degree of synchronisation is present but perhaps not to an acceptable level. Also the responses are too oscillatory and some undesirable effect has spilled over the axis C. The evolution of the step responses over the optimisation is shown in Figure 6-27. The solution essentially stay the same, that is remain oscillatory with the same level of interaction (synchronisation), even after 15000 iterations.

Unlike the synchronising controller obtained by the LQ method which can be tuned in terms of physical quantity (stiffness of virtual spring) and has good and well understood analytical properties, the simplex approach offers only limited control of the degree of synchronisation and no analytical background. Moreover, the LQ method produces superior synchronising controllers and the execution of the procedure takes at least 2 order

of magnitude less time than the simplex optimisation for the benchmark problem. If the only feasible way to obtain a controller that simultaneously decouples and synchronises a set of motion axes is to synchronise a decoupled plant in 2 stages, it would be preferable to synchronise by the LQ and decouple by the simplex method.

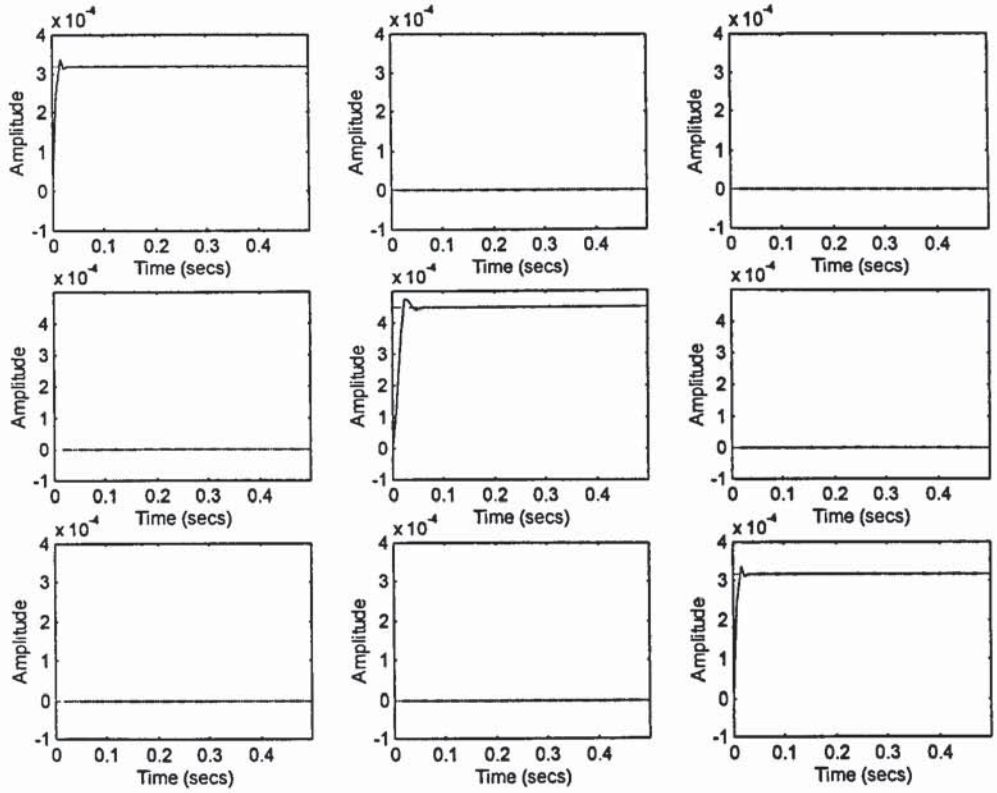

Figure 6-25 Non Interacting Plant With The Non Interacting Controller

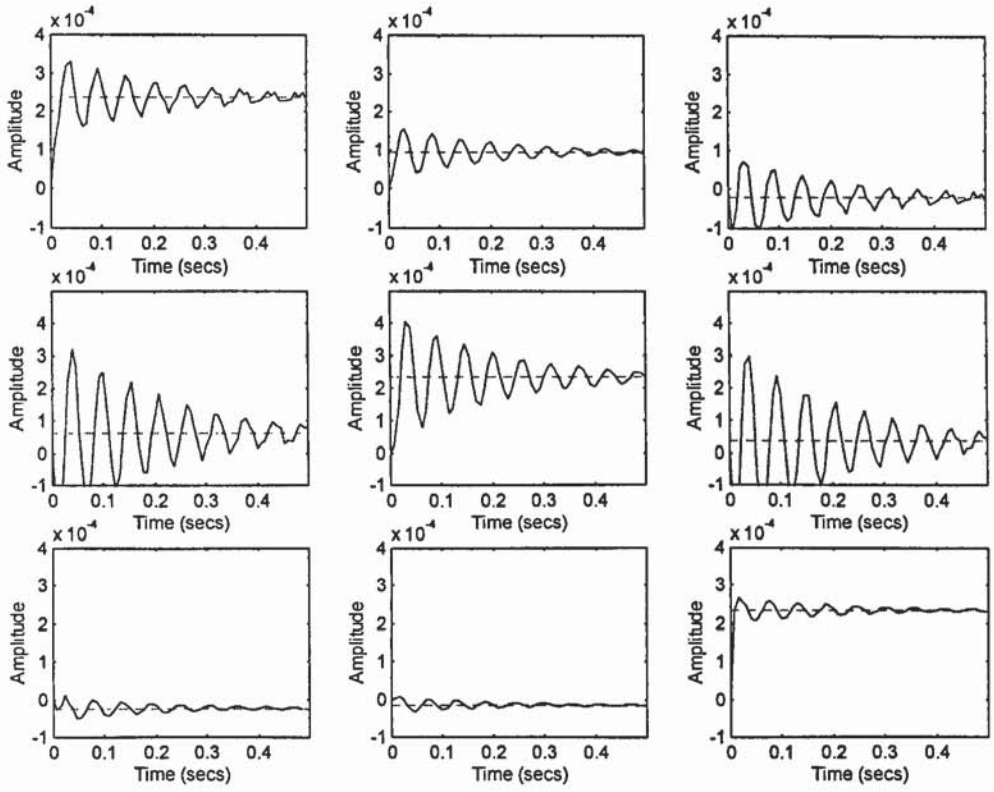

Figure 6-26 Non Interacting Plant With Controller After 5000 Iterations

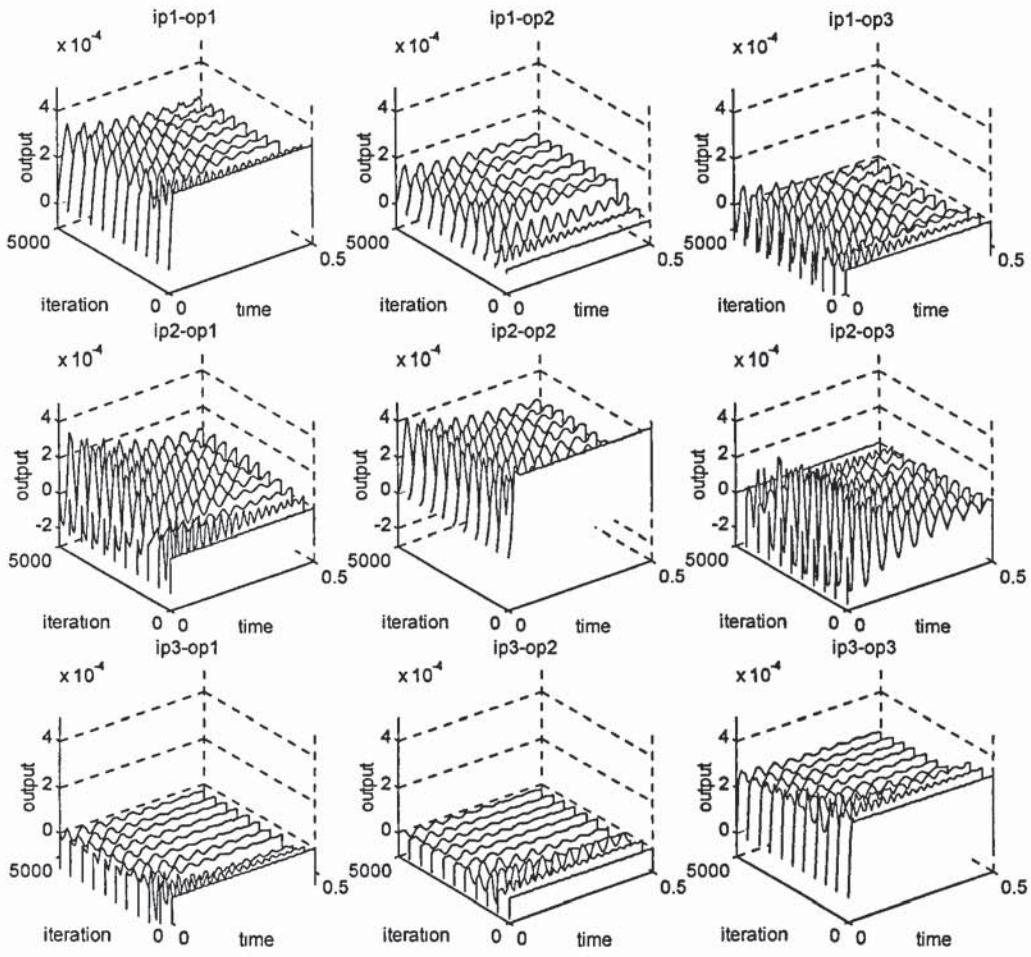

Figure 6-27 Trend of Step Responses over the Optimisation

6.5 Simultaneous Decoupling and Synchronisation By Amoeba

Although workable solutions have only been obtained for the decoupling problem and perhaps less than acceptable solution is obtained for the synchronisation problems, with its somewhat heuristic nature, it might still be worthwhile to try to achieve both requirements with the simplex optimisation all in one application of the amoeba routine. Figure 6-28 shows the evolution of the step responses over 1000 iterations for one of the attempts based on the benchmark problem. Figure 6-29 shows the actual step responses at the end of the 1000 iterations. It can be seen that there is a struggle between the need of synchronisation and the need of decoupling among the axes at each stage. For example, looking at the 3 output responses to input 1 (first row in Figure 6-28) , while output 1 and 2 become more synchronised at the end of the 1000 iterations, output 3 has also acquired a larger amplitude and becomes less decoupled. Comparing all 9 step responses over the optimisation reveals that there is a lack of balance in how the degree of synchronisation and decoupling are spread among the 3 axes as the optimisation progresses. Different

performance indices with different settling times, reference levels and weightings were used but no significant improvements were made. Attempts were also made to look for another local minimum by using other starting points or perturbing the solution and restart the routine. All yield either the same or similar solutions.

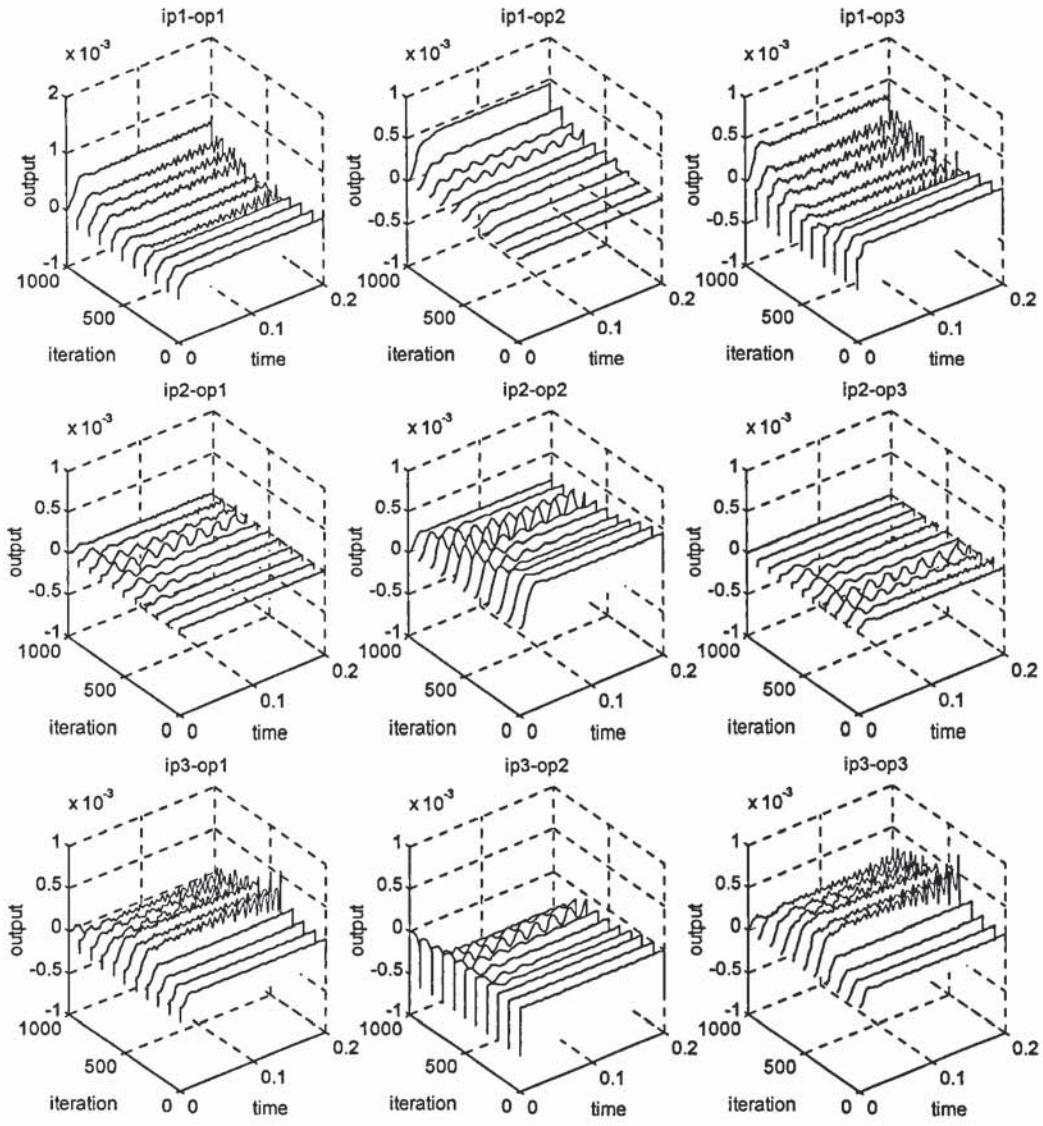

Figure 6-28 Trend of Step Response over Optimisation

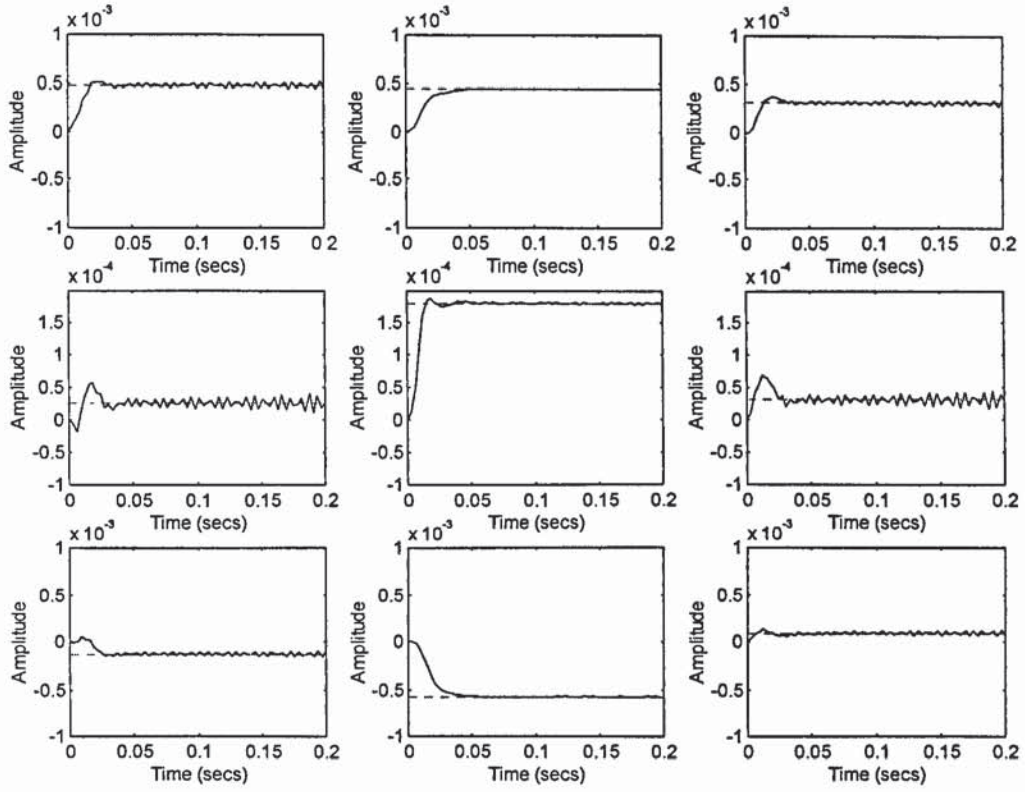

Figure 6-29 Step Responses after 1000 Iterations (simultaneously synchronisation and decoupling)

With reference to section 6.6 on the synchronisation of a decoupled plant, the Euclidean 27-vector norm of the non interacting controller K_c, the decoupling controller K_{de} and the synchronising-decoupling controller K_{total} are 6.74 $x10^3$, 7.36 $x10^3$ and 4.28 $x10^5$ respectively. These numbers suggest that the relative magnitude changes in controller parameters for decoupling is much less than that for synchronisation. This is consistent with the observation that the optimisation appears to be dominated by the effects of synchronisation rather than decoupling. In practice, however, this kind of information will not be available and an exhaustive trial of different performance indices and weightings may not be practical. Based on these evidence, it is decided that the simplex method is not a feasible way to achieve a simultaneous solution.

6.6 Synchronisation Of A Decoupled Plant

In section 6.3.5, it was shown that with a carefully defined performance index, it is possible to achieve a good degree of decoupling between 2 axes by a simplex optimisation. In addition, section 5.2.1 shown that it is possible to imposed a prescribed

degree of synchronisation between 2 uncoupled axes by LQ state feedback. A natural development will be the combination of the 2 methods. In Figure 6-30, a plant is decoupled by the state feedback K_{de} and a further state LQ controller K_{sy} is used to synchronise the appropriate axes. Unlike the open loop (dynamic) compensators, the 2 controllers do not add extra states to the system. In fact, the 2 feedback matrices can be combined to give

$$K_{total} = K_{decouple} + K_{sync} \tag{6-13}$$

which is depicted in Figure 6-31.

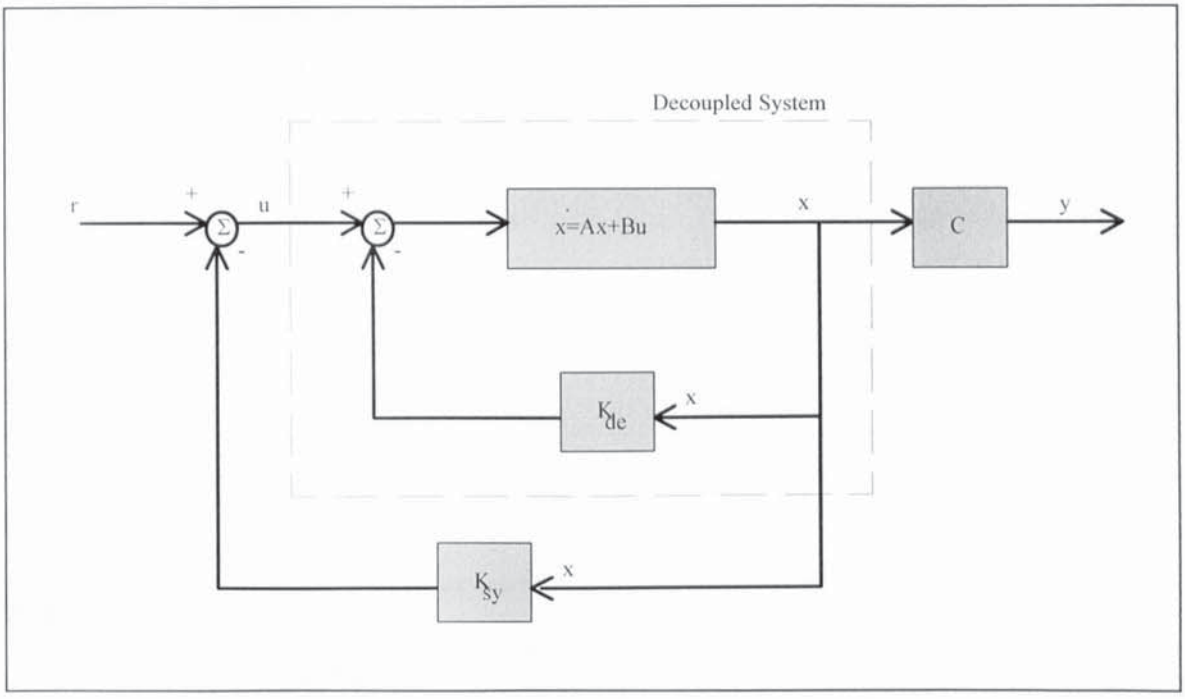

Figure 6-30 Synchronisation of a Decoupled Plant Using State Feedback

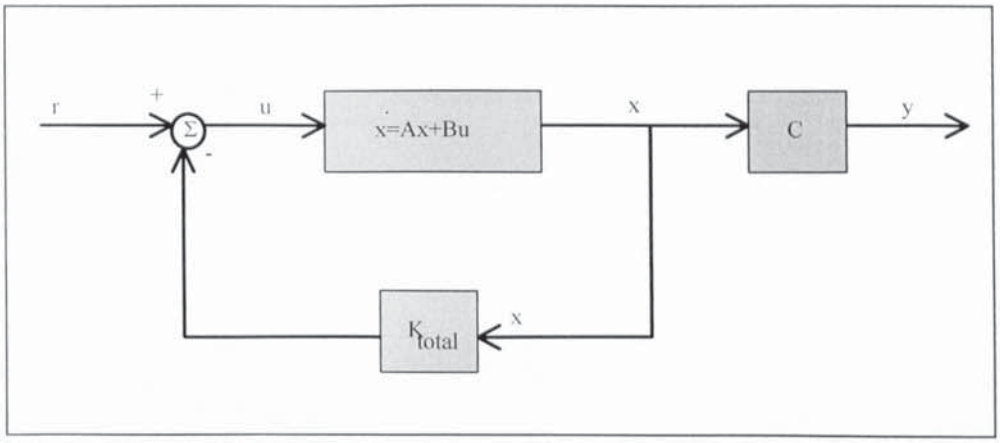

Figure 6-31 Overall State Feedback Control

To carry out this method, the decoupling solution in section 6.3.5 is taken as K_{de}. A new synchronising LQ controller K_{sy} is designed around the resultant decoupled system.

K'_{sy} = 1.0e+005 *

-3.07297619863303	0.72483776662479
0.36052263131698	-0.08696186662288
2.69380462452064	-0.60284325843323
0.00217848019772	-0.00045232633838
0.01626032049154	0.00431359936455
-0.00471669772345	0.00159296518511
-0.11840073013894	0.03315362528197
0.01992476112508	-0.00471669772345
0.58073406027563	-0.12503514387204

Again, the synchronising controller is designed for the whole plant (all the states) but only for 2 axes (A and B). A row of zeros is appended so that K_{sy} can be added to K_{de} to get the overall state feedback matrix K_{total}

K'_{total} = 1.0e+005 *

-3.07297694408505	0.74720353863023	0.00000209764232
0.37330145038419	-0.09286613647433	0.02861007189470
2.71265659572362	-0.59693232579795	0.00301734670590
0.00086944573431	0.00396615209791	0.00418656162897
0.01933050689105	-0.01199994445351	0.03331087507407
-0.00276716460827	0.00516552241271	-0.00144983383740
-0.13239051135824	0.04646367599955	0.01827516525857
0.01853600618757	-0.00058852287090	0.00474595876427
0.61573244924146	-0.13415012368117	0.00263007076654

With this overall controller, the step responses are plotted in Figure 6-32. Each row represents the set of responses of all the 3 axes when a step input is presented to the axis corresponding to the row number. Steady state is achieved within 0.1 sec and a good degree of synchronisation between axes A and B is evident. Axis C is also clearly decoupled from the other axes.

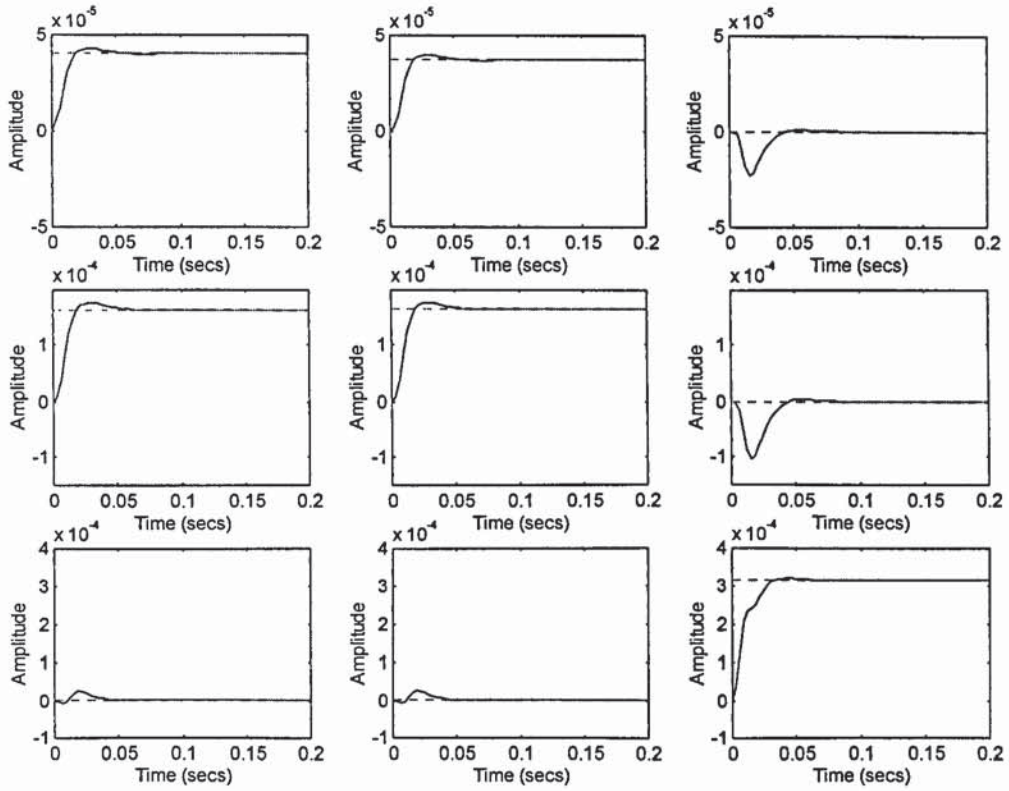

Figure 6-32 MIMO Plant with Decoupling(A,C) and Synchronisation(A,B)

An inspection of the BRG arrays of the individual axes and the relevant 2-axes subsystems in Figure 6-33 and Figure 6-34 confirm that as a closed loop system, axes A and B are closely "connected" whereas the interaction between axis C and the others are relatively minimal.

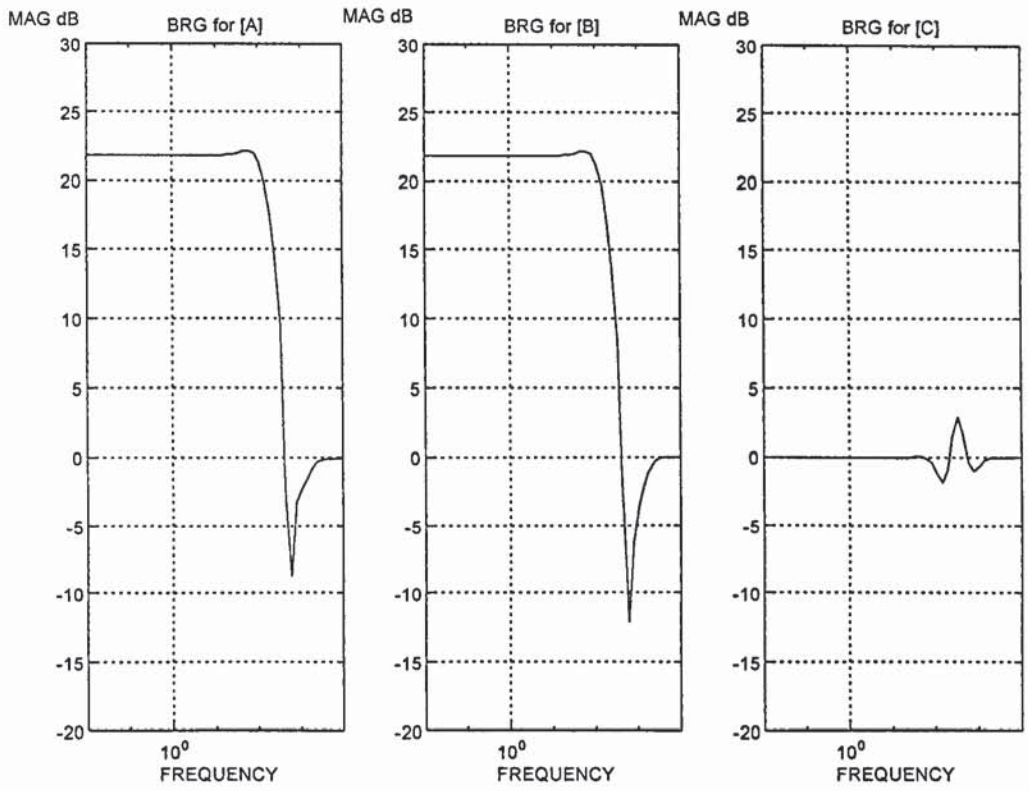

Figure 6-33 BRG Arrays for the Closed Loop System under Decoupling and Synchronisation

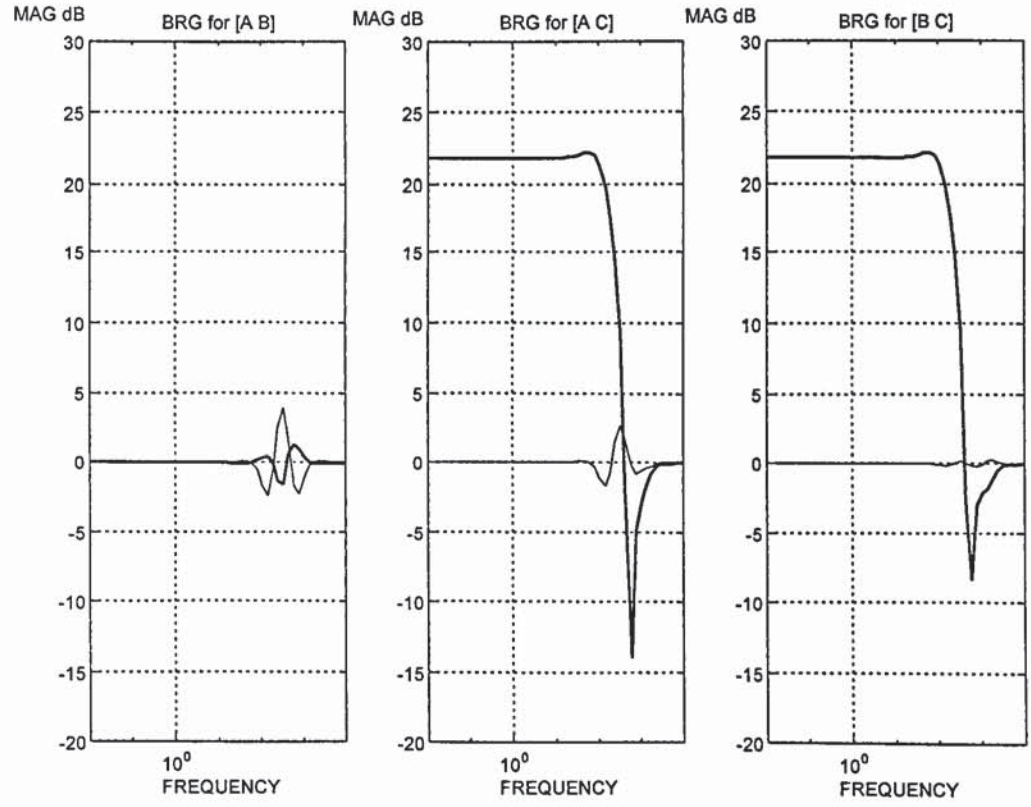

Figure 6-34 BRG Arrays for the Closed Loop System under Decoupling and Synchronisation

However, to determine the best possible topology for a decentralised control scheme, it is necessary to look at the BRG arrays of the overall dynamic compensator. Since K_{total} is only a state feedback matrix, a Kalman filter is designed around the uncontrolled plant so that state estimates are provided for feedback. The combined state estimator and the overall feedback matrix form a dynamic controller, which produces the BRG arrays in Figure 6-35 and Figure 6-36. From the BRG plots for the controller, if a subsystem is made up of axes A and B, the BRG ranges from 22 to 58 dB over the frequencies concerned, whereas the subsystem with axes A and C has BRG range from -7 to 25 dB. Therefore a conclusion can be drawn that as far as this particular set of requirements is concerned, the interactions involved in bringing about decoupling is larger in magnitude than that for bringing about synchronisation. The choice of a decentralised topology will be [A C] & [B].

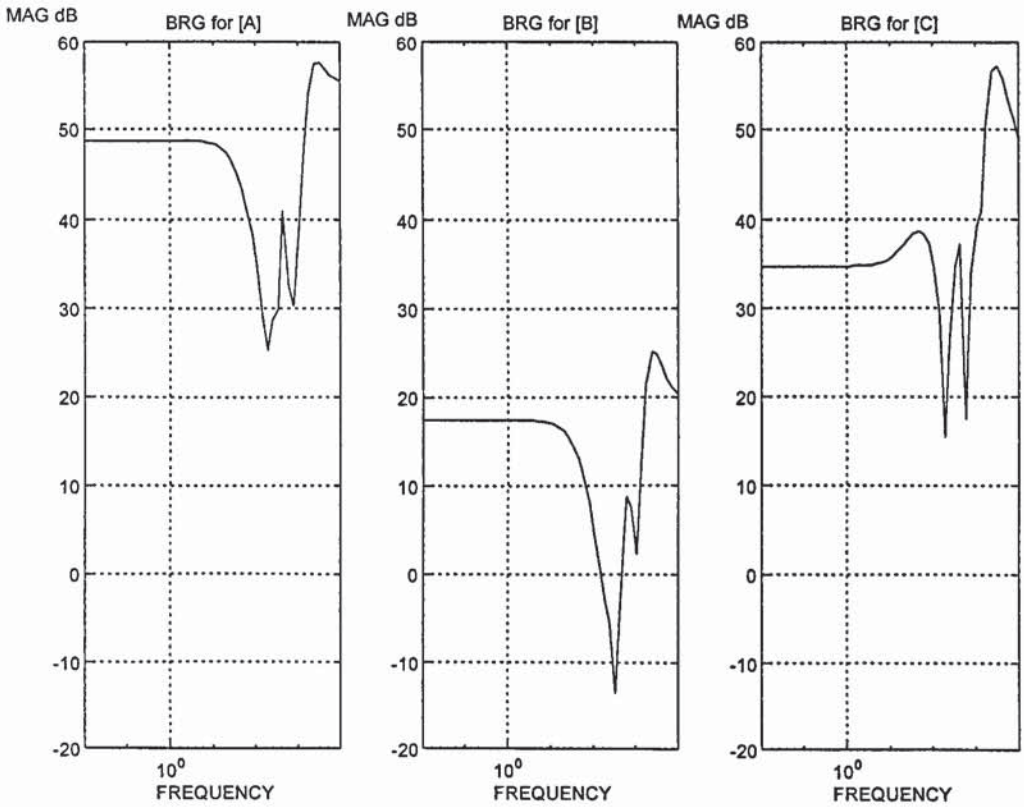

Figure 6-35 BRG Arrays for the Individual Channel of the Overall Dynamic Controller

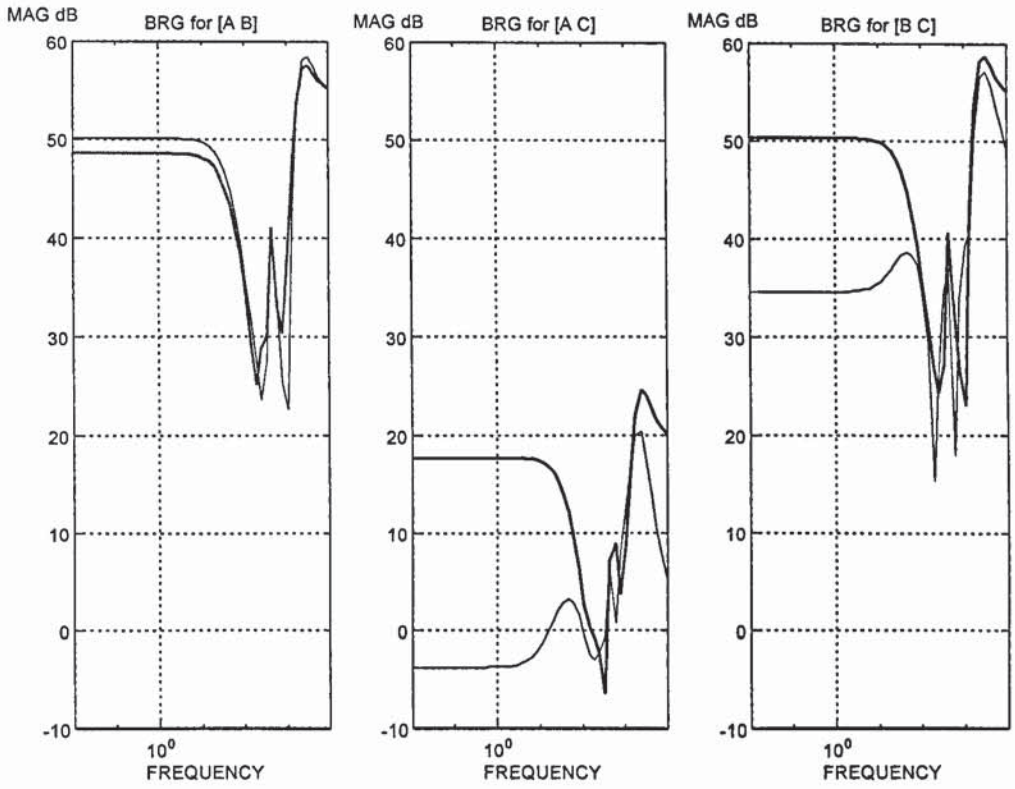

Figure 6-36 BRG Arrays for the Controller Subsystems

The benchmark model has 3 inputs and 3 outputs. The investigation so far has looked at only the natural pairing of the inputs and outputs, namely, axis A is the first input paired with the first output and so forth. It is possible and maybe even advantageous to have different pairings. To complete the picture, the (controller) BRG's of the other possible pairings are plotted for comparison. In Figure 6-37 , 'Bpq' denotes the BRG of the controller with an axis made up of input p paired with output q and in Figure 6-38, 'Bmnpq' denotes the BRG of the controller subsystem made up of input m and n paired with output p and q.

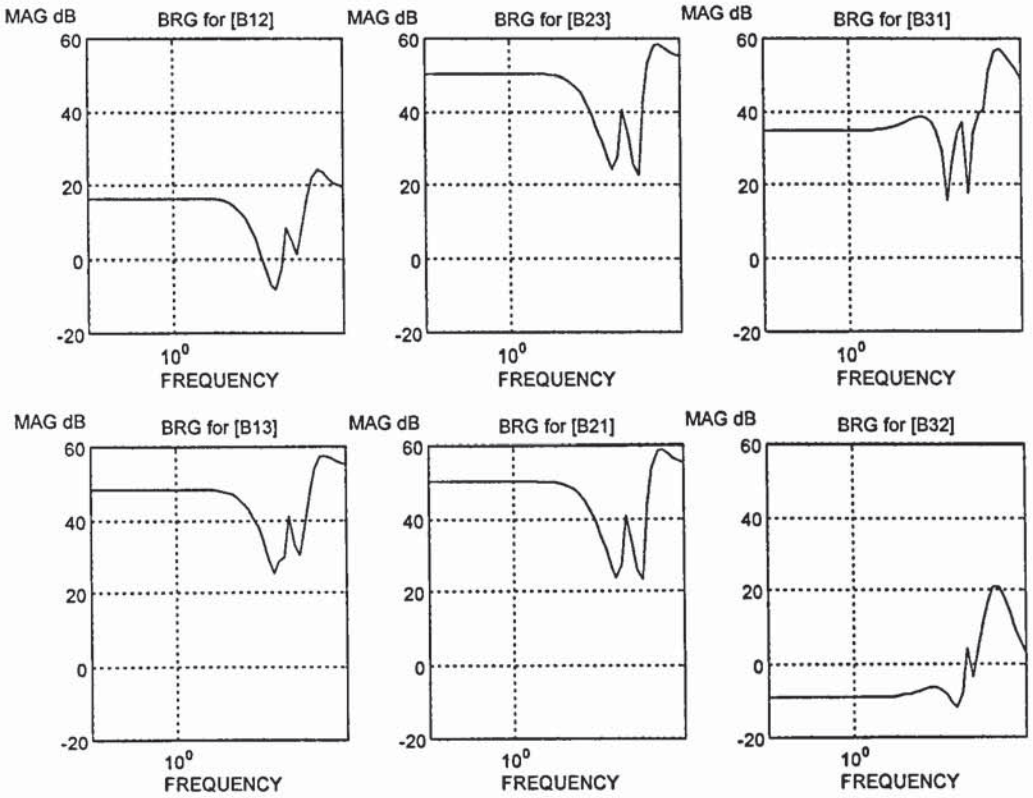

Figure 6-37 BRG's for other Controller Input-Output Pairings

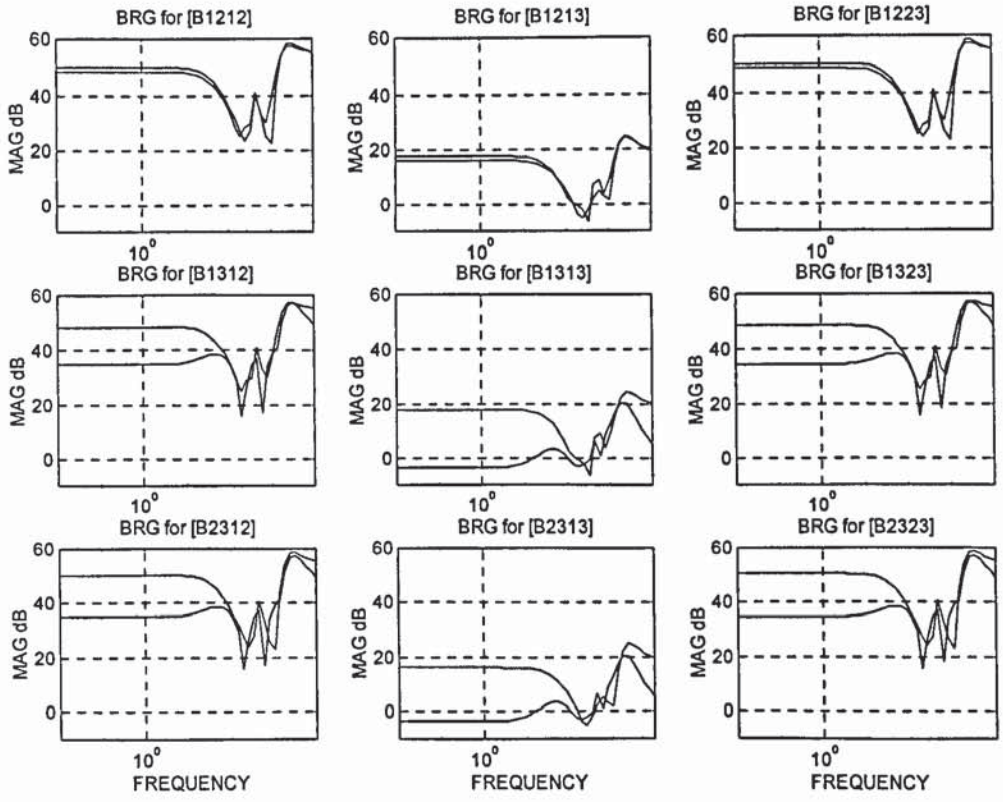

Figure 6-38 BRG's for other Input-Output Controller Subsystems Pairings

6.7 Trade Off Between Decoupling and Synchronisation

In section 6.4, the "preferred" decentralised topology for the benchmark model is determined for a particular set of performance requirements. As explained in section 5.1.3, the performance requirements are likely to be much harsher than a real life application. Nonetheless, the exaggerated requirements are adopted to illustrate the conflict involved in a typical design and accordingly the trade off needed to be made. Once a decentralised topology is determined, the designer has to decide if the compromise made is acceptable. The designer may want to modify the original specification to strike a better balance. In the case of the benchmark problem, the trade off between the decoupling and the synchronisation requirements can be illustrated by a series of BRG arrays.

In Figure 6-39, based on the decoupled system obtained in section 6.3.5, LQ controllers of different stiffness are designed and their corresponding system and controller BRG's are plotted. As expected, when synchronisation stiffness is close to zero, the system BRG's of all the axes are close to 0dB whereas the controller BRG's of axes A and C are significant as they are there to perform the decoupling action. As the synchronisation stiffness becomes larger, the system BRG's of axes A and B increase accordingly to reflect the synchronisation between them. The controller BRG's, however, display a more complicated picture as the controller is now trying to perform both the decoupling (axes A and C) and the synchronisation (axes A and B) action. Such a plot provides some insight into the trade off between the two requirements. For example, information on the range of frequencies in which the requirements remained valid can easier be read off this plot. It is also possible to determine, taken into account of variations over frequency, the point at which synchronisation and decoupling would require the "same" effort from the controller and thus design decisions can be made accordingly.

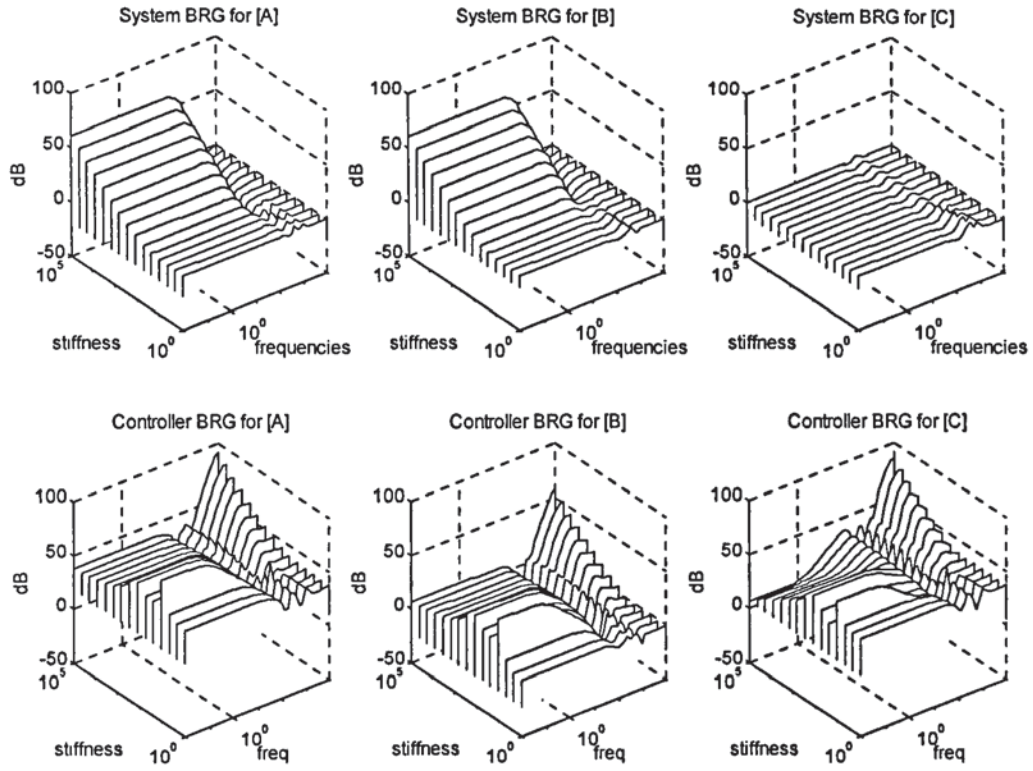

Figure 6-39 Effects of Synchronisation Stiffness on BRG's

7. Topology Selection & Decentralised Control

7.1 Introduction

In chapter 6, a simultaneously decoupling and synchronising controller is designed for the benchmark problem by a combination of downhill simplex optimisation and LQ synchronisation method. In the proposed design methodology, the performance of this full MIMO controller is viewed as a reasonable upper bound of the "achievable" performance with decentralised control schemes. On the other hand, the performance of the same plant under a fully decentralised control scheme with a set of SISO loops can be viewed as the lower bound of any feasible decentralised control scheme. Block relative gain arrays of the most probable decentralised topologies are obtained and compared. The "goodness" of these topologies is assessed by the performance of the respective decentralised controlled systems in closed loops, with emphasis on the degree of synchronisation and decoupling achieved. This part of the design methodology is tested, for the purpose of illustration, based on exaggerated and perhaps less than realistic performance specifications.

With the MIMO controller in place, a topology catering for the performance requirements was determined. The purpose of this chapter is to show that the chosen topology is indeed the most appropriate topology. So far the approach to the topology has been taken based on the system input output behaviour, since the main objective of the project is to produce a generic design methodology that is as automated as possible. During the background study stage of this project, some attempts were made to deploy rigorous control structure analysis with regard to the possible decentralised topology. Tools such as graph theory which focuses on the connectivity of the states were studied, but such analysis entails a high degree of user's input which is contrary to the main objective. Therefore this line of investigation was not pursued any further. Instead, attention is focused on system behaviour and in particular step responses of the relevant decentralised topologies.

7.1.1 Viability of a Topology - Block Relative Gain Array

Consider a transfer function matrix with a selected block diagonal structure. Interactions exist between the various blocks. In order to select the most suitable or the "optimal" structure, some form of indicative measures of the relative magnitudes of these interactions within the structure will be required. The Block Relative Gain (BRG) array is

found to have suitable properties for this purpose. A full definition of the BRG array is given in section 2.7.3. For a partitioned system with transfer function matrix

$$G = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix}$$

(7-1)

and

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = G \cdot \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

(7-2)

the (left) BRG is defined as follows (where entries in F refer to feedback paths),

$$\begin{aligned} BRG_L &= \left[\frac{\partial y_1}{\partial u_1}_{\substack{u_2 = 0 \\ F = 0}} \right] \cdot \left[\frac{\partial y_1}{\partial u_1}_{\substack{y_2 = 0 \\ F_1 = 0 \\ F_2 = I}} \right]^{-1} \\ &= G_{11} \cdot [G^{-1}]_{11} \\ &= G_{11} \cdot \left[G_{11} - G_{12} G_{22}^{-1} G_{21} \right]^{-1} \end{aligned}$$

(7-3)

Essentially the BRG matrix can be understood as the ratio of the multivariable frequency response from the selected set of inputs to the selected set of outputs with all other loops open, to the corresponding frequency response when all the other outputs are controlled perfectly by feedback loops. In other words, whenever the selected set of inputs and outputs of a plant has a BRG equal to the identity matrix, its response are not affected by whether or not the rest of the inputs and outputs are under perfect closed loop control. In this context, the selected set of inputs and outputs is a suitable partition for decentralisation. This includes the case when neither G_{12} nor G_{21} is zero but the product $G_{12}G^{-1}{}_{22}G_{21}$ in equation (7-3) equals the null matrix. That is, when there are interactions between the partitions but their responses can only be modified by closing local and not remote loops. Bearing this in mind, BRG can be treated as a measure of the suitability of a partition for decentralisation and in this restricted sense, a measure of interaction between each block and the rest of the overall system.

7.2 Multi-Loop SISO Control of the Benchmark Model

7.2.1 Multi-Loop PID Control of the Benchmark Model

For the purpose of comparison, 3 parallel PID controllers for the benchmark model as shown in Figure 7-1, are designed by trial and error. No "optimal" PID settings were offered since that depends on the specific requirements such as robustness, tolerance to steady state errors and overshoot, to name but a few qualities of the controllers. In any case, the settings would not affect the cross axis interactions, which is the focus of the exercise. Their step responses are shown as in Figure 7-2. This represents a set of "minimal" controlled performance of the plant. As expected, there is no synchronisation between axes A and B nor decoupling between axes A and C. For the class of machinery under consideration (as described in chapter 1), the motors are often required to follow very similar or identical motion profiles. In such cases, the step responses produced in the study provide information as to how the axes will react to cross axes disturbance.

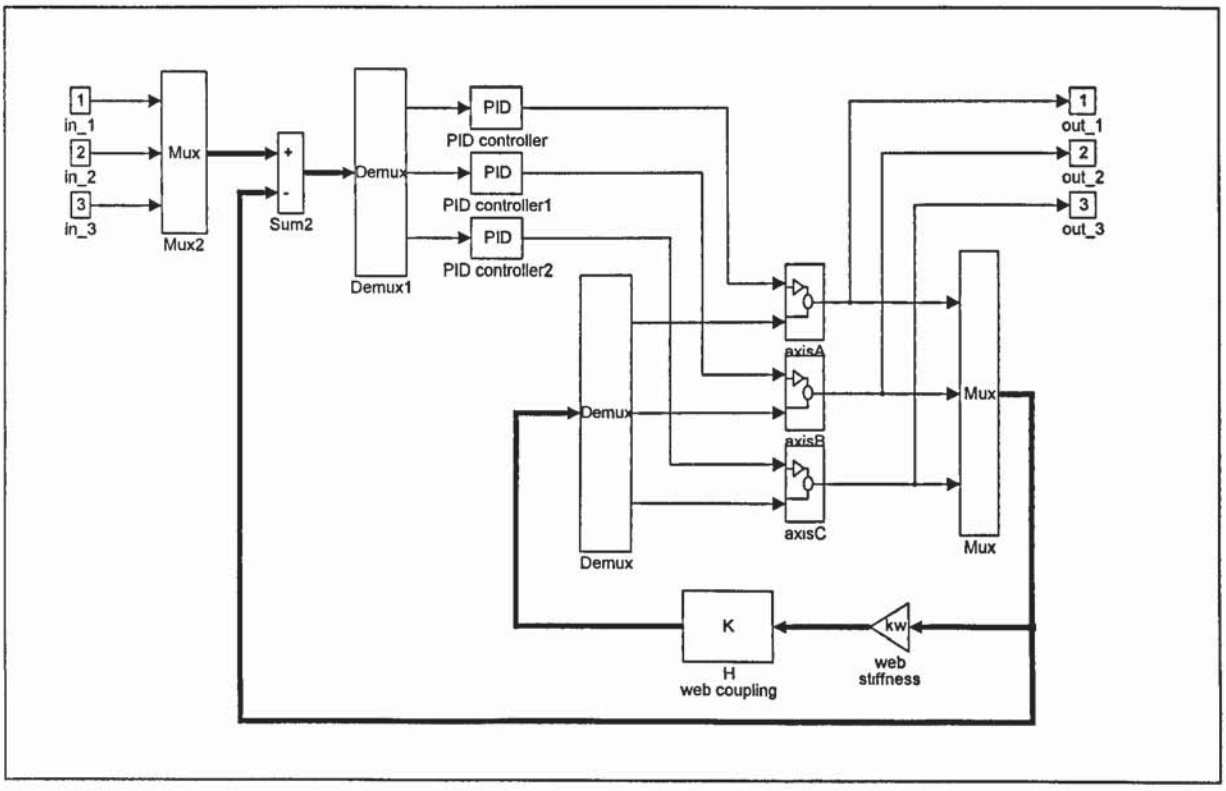

Figure 7-1 Benchmark Model under Multi Loop PID Control

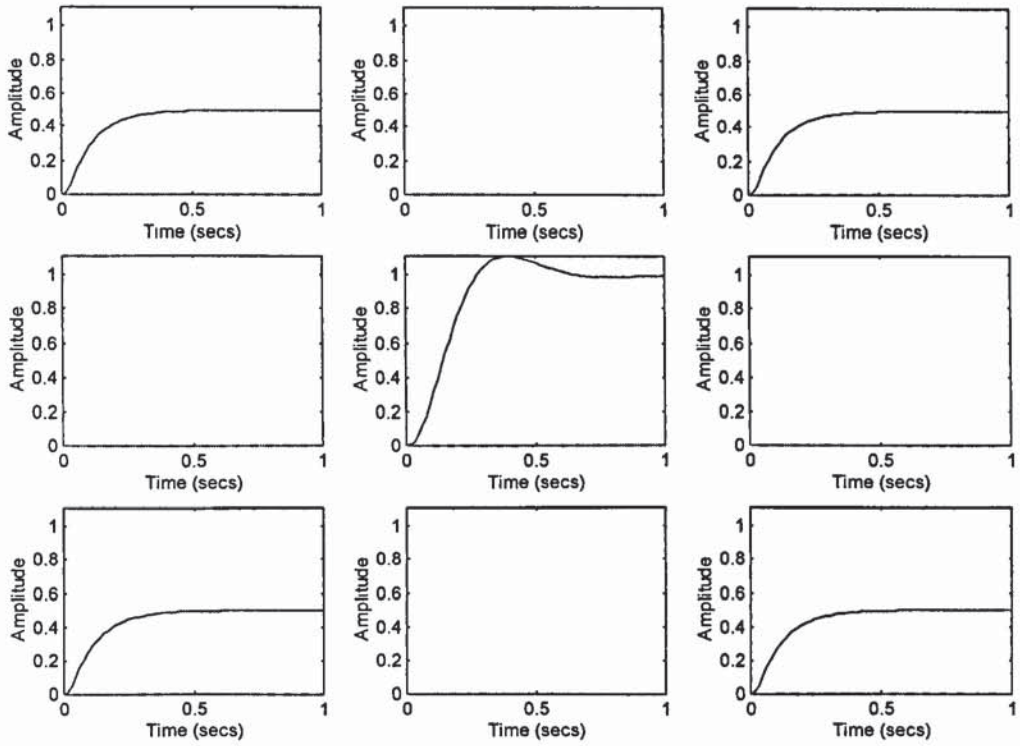

Figure 7-2 Step Responses of Plant under Multi Loop PID Control

7.3 Full MIMO Control of the Benchmark Model

A simultaneously decoupling and synchronising state feedback MIMO controller with the following feedback gain

K'_f = 1.0e+005 *

-3.07297694408505	0.74720353863023	0.00000209764232
0.37330145038419	-0.09286613647433	0.02861007189470
2.71265659572362	-0.59693232579795	0.00301734670590
0.00086944573431	0.00396615209791	0.00418656162897
0.01933050689105	-0.01199994445351	0.03331087507407
-0.00276716460827	0.00516552241271	-0.00144983383740
-0.13239051135824	0.04646367599955	0.01827516525857
0.01853600618757	-0.00058852287090	0.00474595876427
0.61573244924146	-0.13415012368117	0.00263007076654

is obtained for the benchmark model by a combination of downhill simplex optimisation and LQ synchronisation method in chapter 6. Together with an appropriate Kalman state estimator, a dynamic output feedback controller is put around the benchmark model as depicted in Figure 7-3.

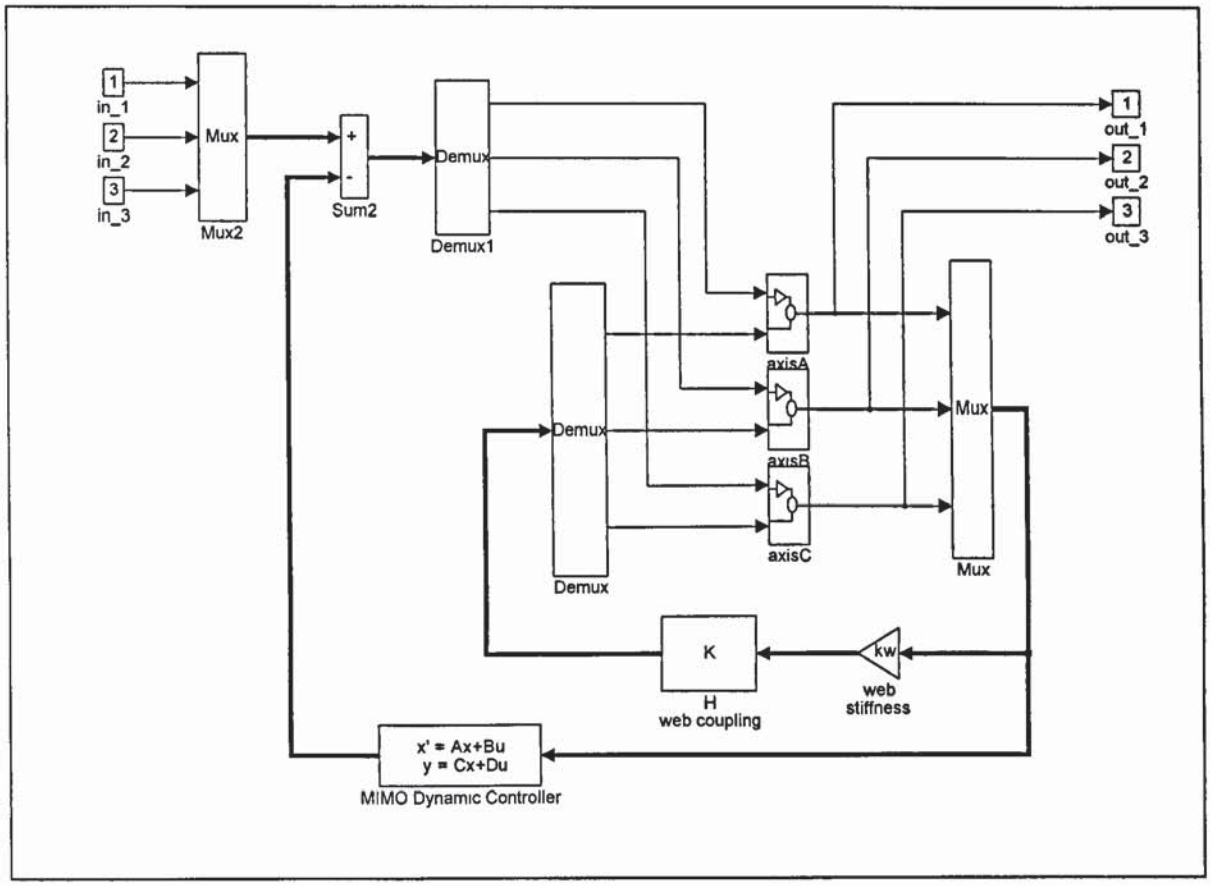

Figure 7-3 Full MIMO Control of the Benchmark Model

Pre-multiplying the input with the static gain

```
Kr = 1.0e+004 *
  1.30233201436959  0.30379203876424  0.31294455682139
```

the step responses of this closed loop control system were shown in chapter 6 and are reproduced in Figure 7-4. These responses represent, in the present context, a reasonable upper bound of the "achievable" performance with decentralised control schemes. The overall dynamic controller clearly synchronises axes A and B and decouples axes A and C in the closed loop system (Figure 7-4 and Figure 7-5). The BRG arrays for the alternative controller subsystems in Figure 7-6 suggests that when presented with the alternative decentralisation partitions [A B] and [A C] , the partition [A C] is much more viable for decentralisation since it will encounter much less disturbances from the rest of the system. Therefore the overall performance of the [A C] & [B] decentralisation scheme should be superior to that of the [A B] & [C] scheme.

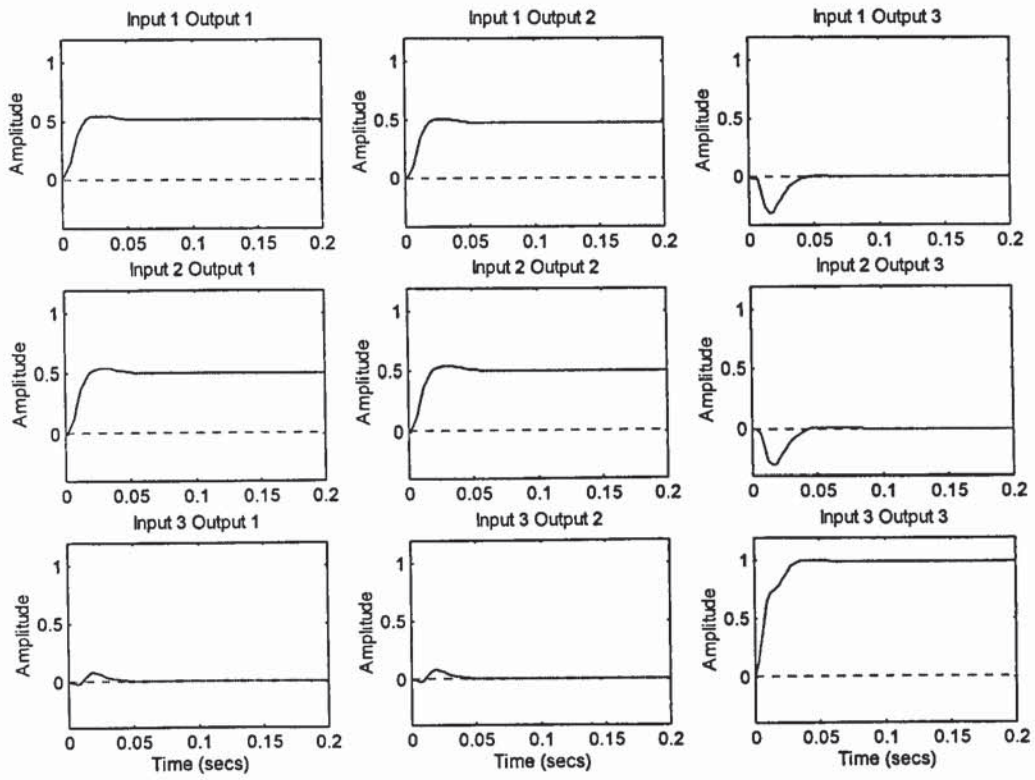

Figure 7-4 MIMO Plant with Decoupling(A,C) and Synchronisation(A,B)

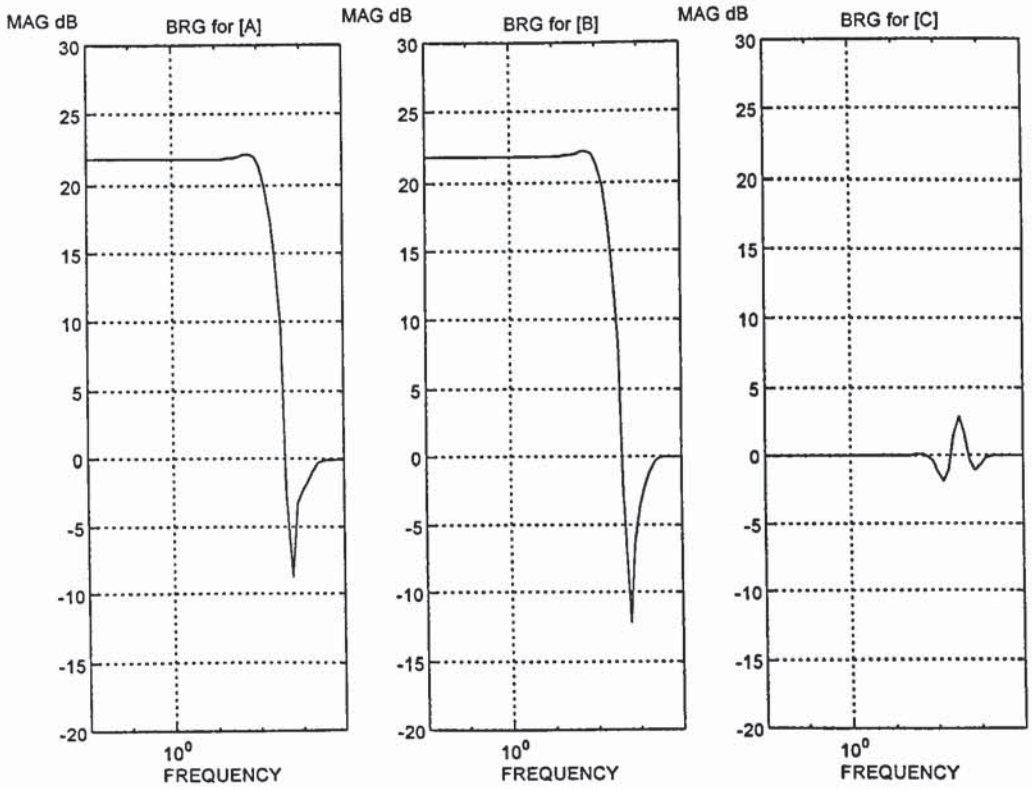

Figure 7-5 BRG Arrays for the Closed Loop System under Decoupling and Synchronisation

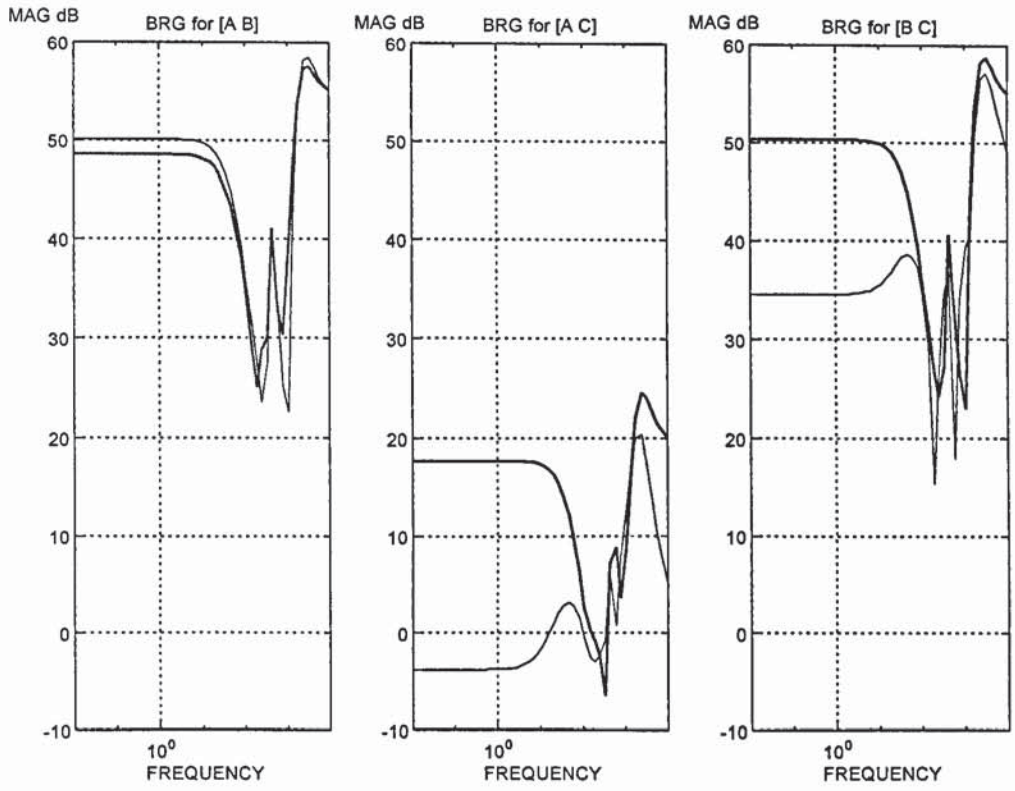

Figure 7-6 BRG Arrays for the Controller Subsystems

7.4 Decentralised Control of the Benchmark Model

For the 3-by-3 benchmark model, a block or partially decentralised scheme (Figure 7-7) consists of a 2-by-2 and a SISO dynamic controller in parallel is employed. There are several plausible ways to assign the controller coefficients and they are described in the following subsections.

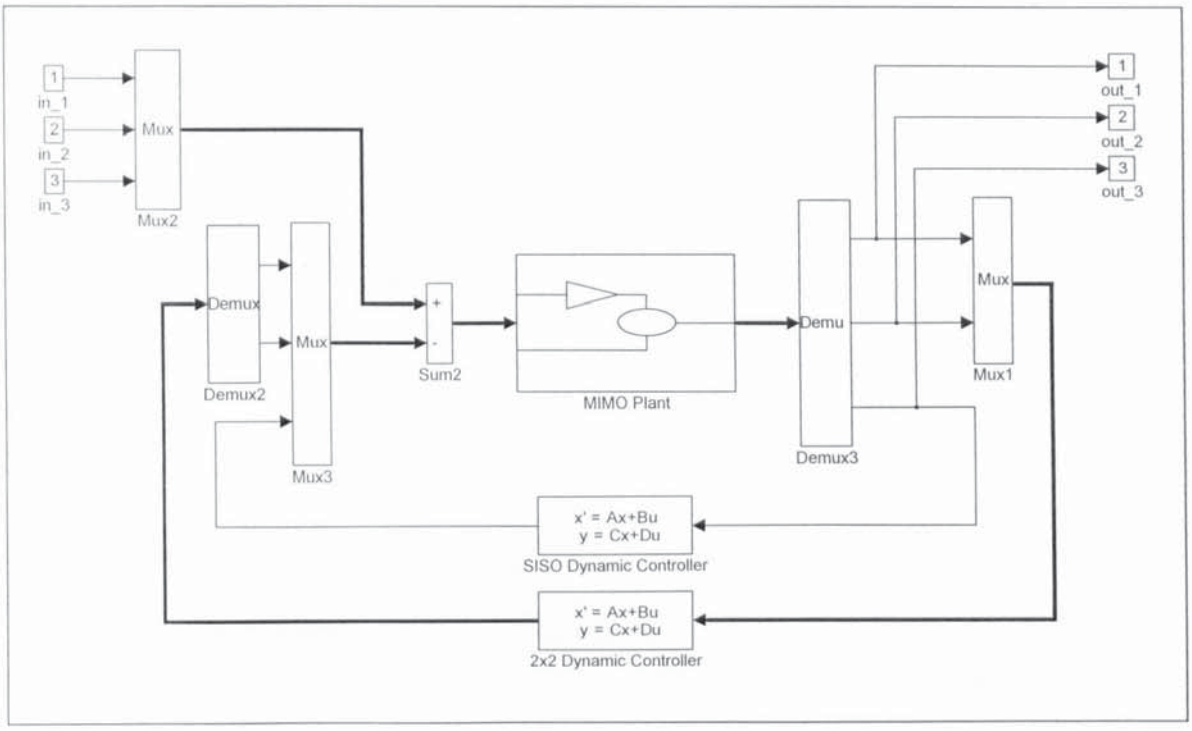

Figure 7-7 Decentralised Control with one SISO and one 2X2 Dynamic Controller

7.4.1 Masking the Full MIMO Controller

With the full MIMO controller at hand, representing the theoretical best solution, it is natural to try to extract the required controllers from it. The first attempt is simply to mask (discard) the off-block-diagonal transfer functions of the full MIMO dynamic controller as indicated in Figure 7-8. For both the [A B],[C] and [A C],[B] topologies, however, the benchmark model under the controllers obtained by masking are found to be unstable. This is not surprising since, in both cases, the resultant controllers take no account of the strong interaction (inherent or imposed) existing between the axes.

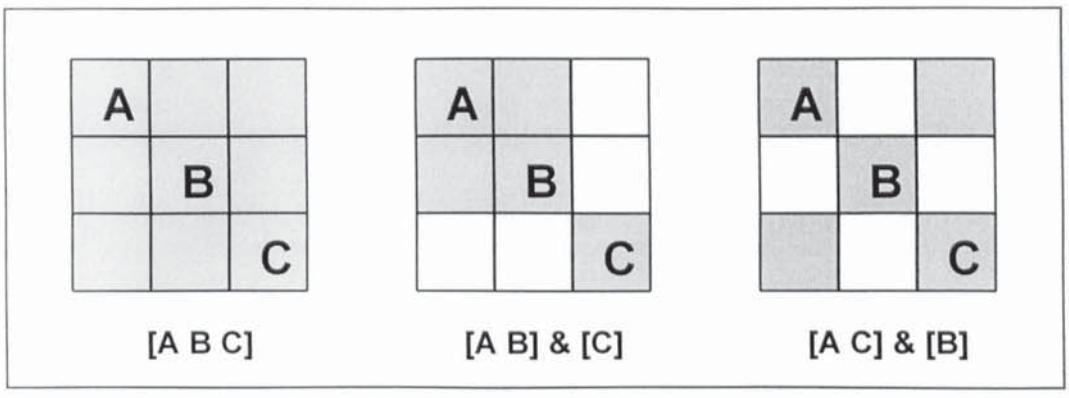

Figure 7-8 Masking Block Diagonal Transfer Functions

7.4.2 Extracting Decentralised Controllers as Subsystems of the Full Controller

The internal structure of the full MIMO controller obtained for the benchmark problem is depicted in Figure 7-9 in which K_c and K_f denote the state feedback and the state estimator gains whereas *A*, *B*, *C* and *D* are the system matrices of the plant model.

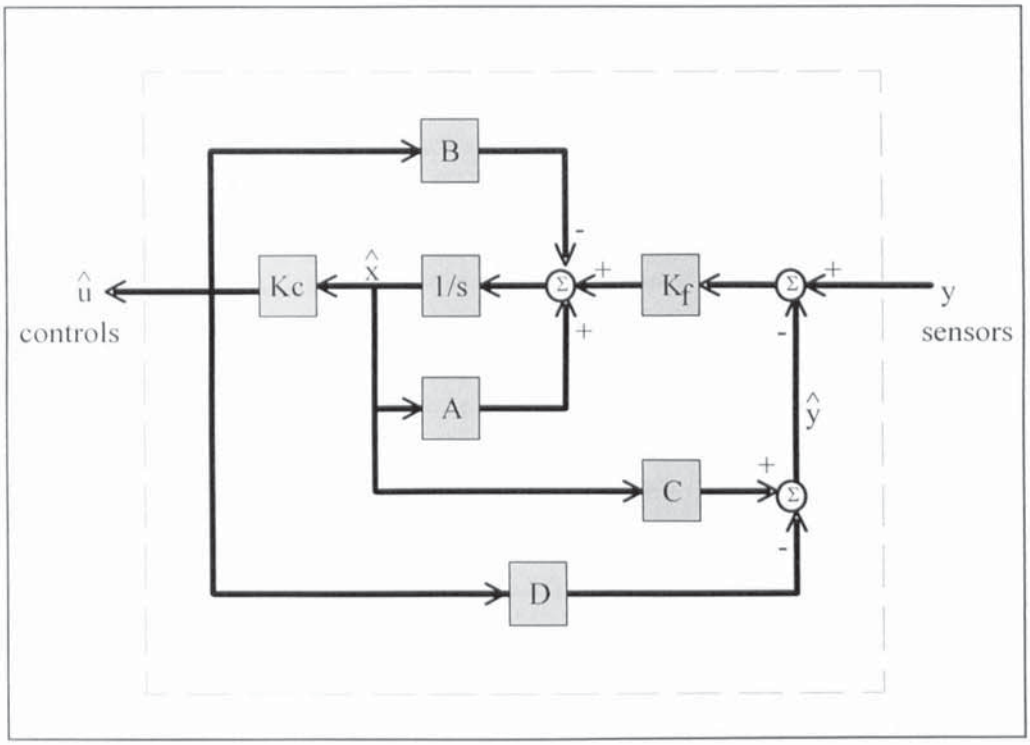

Figure 7-9 Internal Structure of a Dynamic Controller

As the plant has no direct feedforward, *D*=0. The overall controller can be summarised by

$$\dot{\hat{x}} = [A - BK_c - K_f C] \cdot \hat{x} + K_f y$$
$$\hat{u} = K_c \hat{x}$$

(7-4)

In general, the state equation of any MIMO dynamic controller (without feedforward), with a state vector ϕ, which acts on the output y of the plant and produces a control signal u is

$$\dot{\phi} = \alpha\phi + \beta y$$
$$u = \gamma\phi$$

(7-5)

Therefore, in the case of the benchmark problem, the full MIMO controller is defined by

$$\alpha = A - BK_c - K_f C$$
$$\beta = K_f$$
$$\gamma = K_c$$

(7-6)

This can be partitioned as follows

$$\dot{\phi} = \alpha\phi + \begin{bmatrix}\beta_1 & \beta_2\end{bmatrix} \cdot \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$
$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix} \cdot \phi$$

(7-7)

Two decentralised controllers, one SISO and one 2X2, can be extracted by selecting the appropriate input and output vectors $[y_1\ y_2]'$ and $[u_1\ u_2]'$, to suit either the [A B] & [C] or the [A C] & [B] topology. For example, for the [A B] & [C] topology, by setting $y_1=[y_A\ y_B]'$, $y_2=[y_C]$, $u_1=[u_A\ u_B]'$, $u_2=[u_C]$ and partition the β and γ matrices accordingly, a 2X2 controller

$$\dot{\phi}_1 = \alpha\phi_1 + \beta_1 y_1$$
$$u_1 = \gamma_1 \phi_1$$

(7-8)

and a SISO controller

$$\dot{\phi}_2 = \alpha\phi_2 + \beta_2 y_2$$
$$u_2 = \gamma_2 \phi_2$$

(7-9)

can be obtained. Notice that both controllers in equations (7-8) and (7-9) retain the full system matrix α and the full state vector ϕ_1/ϕ_2 and thus effectively contain the full plant dynamics. Although the vectors ϕ_1 and ϕ_2 represent the same set of state variables in both of the controllers, each follows a different trajectory since they are implemented independently. This arrangement reduces the demand on communication between different axis. Similar decentralised controllers can be obtained for the [A C] & [B] topology by the

appropriate partition of the input and output vectors. The step responses of the corresponding closed loop decentralised control systems for the [A B] & [C] and [A C] & [B] topologies are shown in Figure 7-10 and Figure 7-11 respectively.

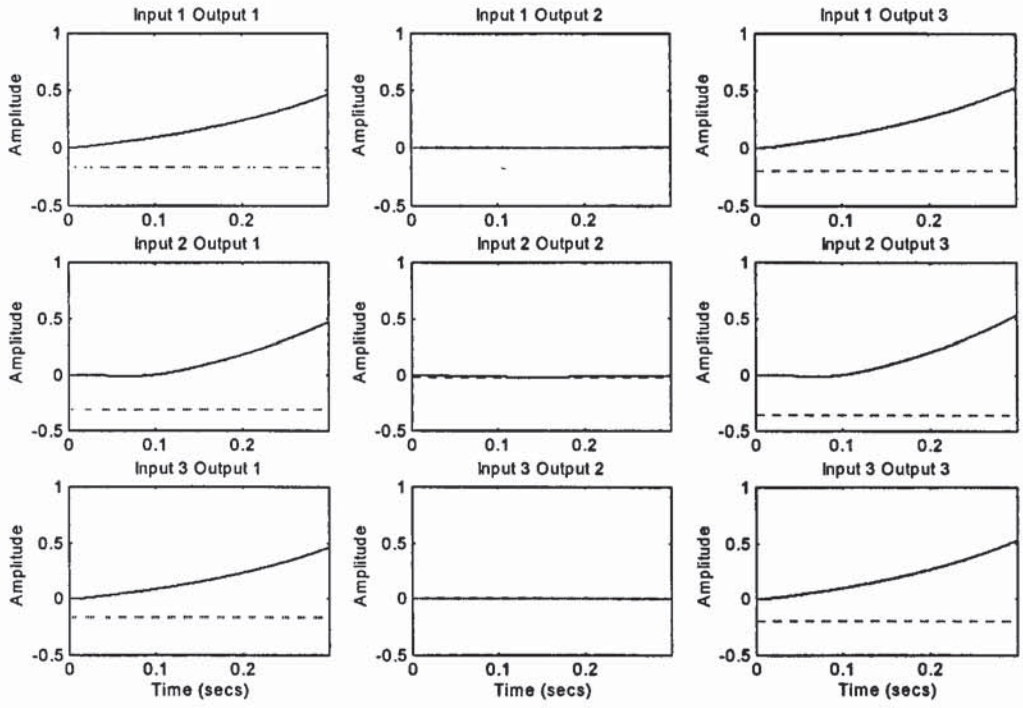

Figure 7-10 Closed Loop Step Responses with Extracted Controllers for [A B] & [C]

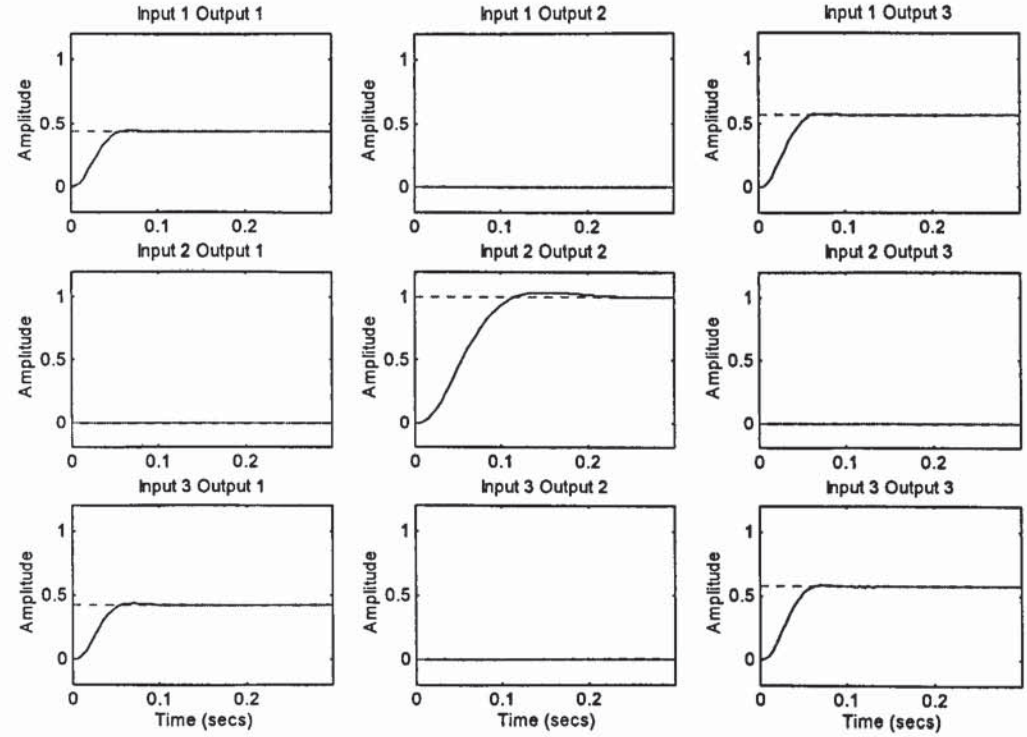

Figure 7-11 Closed Loop Step Responses with Extracted Controllers for [A C] & [B]

7.4.3 Decentralised Controllers as Reduced Order Subsystems of the Full Controller

Similar the development in the previous section, but instead of equation (7-7), the full MIMO controller can be partitioned as follows

$$\begin{bmatrix}\dot{\psi}_1\\ \dot{\psi}_2\end{bmatrix}=\begin{bmatrix}\alpha_{11} & \alpha_{12}\\ \alpha_{21} & \alpha_{22}\end{bmatrix}\cdot\begin{bmatrix}\psi_1\\ \psi_2\end{bmatrix}+\begin{bmatrix}\beta_{11} & \beta_{12}\\ \beta_{21} & \beta_{22}\end{bmatrix}\cdot\begin{bmatrix}y_1\\ y_2\end{bmatrix}$$
$$\begin{bmatrix}u_1\\ u_2\end{bmatrix}=\begin{bmatrix}\gamma_{11} & \gamma_{12}\\ \gamma_{21} & \gamma_{22}\end{bmatrix}\cdot\begin{bmatrix}\psi_1\\ \psi_2\end{bmatrix}$$

(7-10)

Comparison with equations (7-7) shows that in addition to the input and output vectors, the state vector is also partitioned. A decentralised controller of the form

$$\dot{\psi}_1=\alpha_{11}\psi_1+\beta_{11}y_1$$
$$u_1=\gamma_{11}\psi_1$$

(7-11)

can be extracted by assigning y_1, u_1 and ψ_1 to the appropriate input, output and states variables and setting y_2, u_2 and ψ_2 to zero. For example, by assigning $y_1=[y_A\ y_B]'$, $y_2=[y_C]$, $u_1=[u_A\ u_B]'$, $u_2=[u_C]$, together with an appropriate ψ_1, a 2X2 decentralised controller for the sub-system [A B] of the benchmark model can be obtained.

Notice that the partition of the state vector ψ can be arbitrary while the partition for the input vector, *u,* and the output vector, *y*, are restricted by the choice of topology. Consequently, the state equation (7-5) of the full MIMO controller can also be represented with a different (arbitrary) partition consistent with the state partition $\psi=[\psi'_1\ \psi'_2]'$.

$$\begin{bmatrix}\dot{\psi}'_1\\ \dot{\psi}'_2\end{bmatrix}=\begin{bmatrix}\alpha'_{11} & \alpha'_{12}\\ \alpha'_{21} & \alpha'_{22}\end{bmatrix}\cdot\begin{bmatrix}\psi'_1\\ \psi'_2\end{bmatrix}+\begin{bmatrix}\beta'_{11} & \beta'_{12}\\ \beta'_{21} & \beta'_{22}\end{bmatrix}\cdot\begin{bmatrix}y_1\\ y_2\end{bmatrix}$$
$$\begin{bmatrix}u_1\\ u_2\end{bmatrix}=\begin{bmatrix}\gamma'_{11} & \gamma'_{12}\\ \gamma'_{21} & \gamma'_{22}\end{bmatrix}\cdot\begin{bmatrix}\psi'_1\\ \psi'_2\end{bmatrix}$$

(7-12)

Another decentralised controller of the form

$$\dot{\psi}'_1 = \alpha'_{11}\psi'_1 + \beta'_{11}y_1$$
$$u_1 = \gamma'_{11}\psi'_1$$

(7-13)

which is different form (7-11), can be extracted. By assigning $y_1=[y_C]$, $y_2=[y_A\ y_B]'$, $u_1=[u_C]$, $u_2=[u_A\ u_B]'$, together with an appropriate ψ'_1, a SISO decentralised controller can be obtained for the sub-system [C] of the benchmark model. In general, a different state partition, which implies a different partition in the α, β, γ matrices, can be employed for each of the decentralised controller required. Thus different and independent "partial" dynamics of the full plant model is retained for each of the decentralised controllers. This flexibility maximises the number of state variables to be used which remains practical to be implemented.

This extraction of decentralised controllers represents a more significant reduction in computation need than the previous section. However, since state space realisations are not unique, some criterion has to be imposed to select the appropriate partition. For the benchmark problem, the approach taken is to select the realisation of the full MIMO controller which has equal diagonals of controllability and observability gramians and thus "balances" its input and outputs. Furthermore, the 9 variables in the balanced state vector ψ is partitioned in such a way that only the 6 fastest states variables would be retained in the decentralised controllers. More specifically, the state vector is partitioned so that ψ_1=[6 fastest state variables] in equation (7-11) and ψ'_1=[6 fastest state variables] in equation (7-13). In this way, a pair of 2X2 and SISO decentralised controllers is obtained for each of the [A B] & [C] and [A C] & [B] topologies. The step responses of the corresponding closed loop decentralised control system are shown in Figure 7-12 and Figure 7-13 respectively.

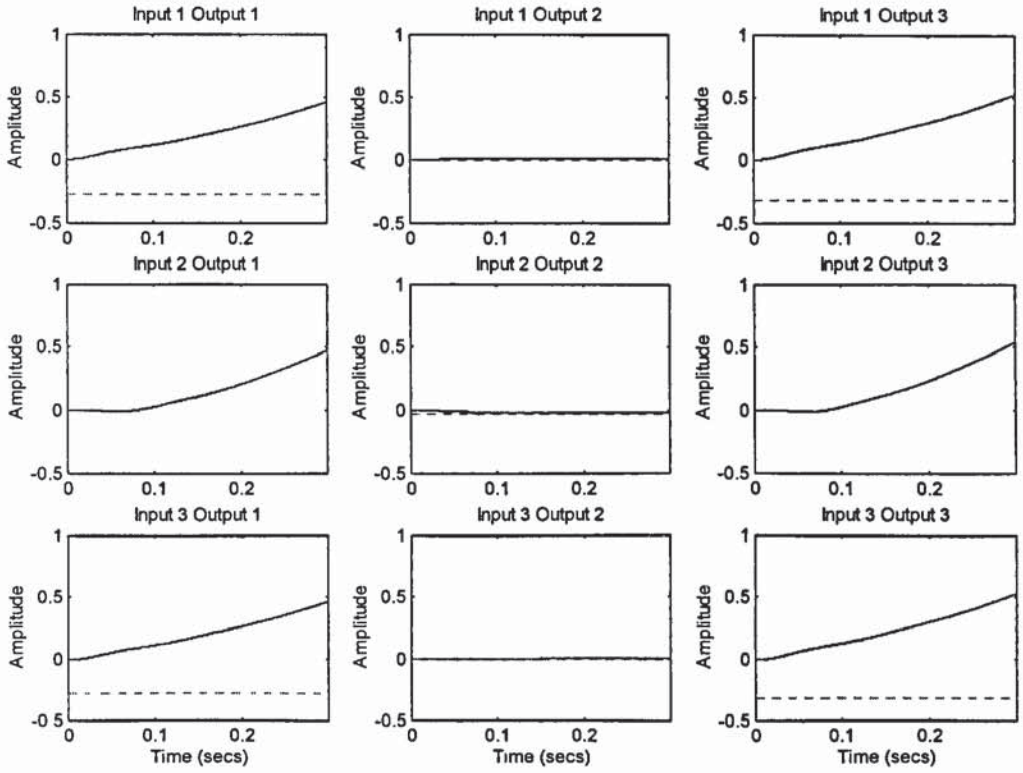

Figure 7-12 Closed Loop Step Responses with Extracted Controllers for [A B] & [C]

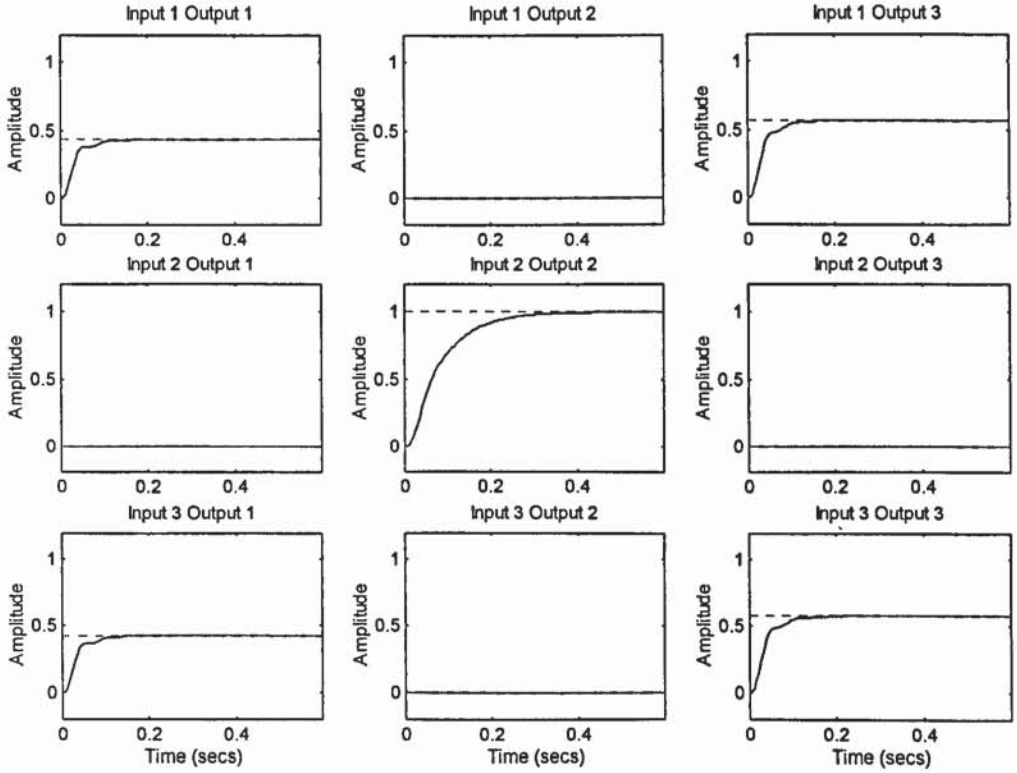

Figure 7-13 Closed Loop Step Responses with Extracted Controllers for [A C] & [B]

7.5 Decentralising Controllers by Masking Control and Estimator Gains

With reference to the internal structure of the full MIMO controller in Figure 7-9, equation (7-4) can be partitioned in the following way.

$$\dot{\hat{x}} = \left[A - \begin{bmatrix} B_1 & B_2 \end{bmatrix} \cdot \begin{bmatrix} K_{c1} \\ K_{c2} \end{bmatrix} - \begin{bmatrix} K_{f1} & K_{f2} \end{bmatrix} \cdot \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} \right] \cdot \hat{x} + \begin{bmatrix} K_{f1} & K_{f2} \end{bmatrix} \cdot \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

$$\begin{bmatrix} \hat{u}_1 \\ \hat{u}_2 \end{bmatrix} = \begin{bmatrix} K_{c1} \\ K_{c2} \end{bmatrix} \cdot \hat{x}$$

(7-14)

Similar to section 7.4.2, a pair of SISO and 2X2 decentralised controllers can be constructed by the appropriate assignment of input and output variables together with the corresponding partition of the feedback and estimator gains K_c, K_f and the B, C matrices of the plant model.

$$\dot{\hat{x}}_1 = [A - B_1 K_{c1} - K_{f1} C_1] \cdot \hat{x}_1 + K_{f1} y_1$$
$$\hat{u}_1 = K_{c1} \hat{x}_1$$

(7-15)

$$\dot{\hat{x}}_2 = [A - B_2 K_{c2} - K_{f2} C_2] \cdot \hat{x}_2 + K_{f2} y_2$$
$$\hat{u}_2 = K_{c2} \hat{x}_2$$

(7-16)

The two controllers have the same set of state variables but each follows a different trajectory since they are implemented independently. Comparing to the equations (7-8) and (7-9), the system dynamics matrix [A - BK_c - K_fC] of the decentralised controllers in (7-15) and (7-16) reflects only the gains of the relevant input and output channels, which provides a closer approximation to the relevant part of the plant that the individual decentralised controller is attempting to control. The full state vector is used in the following examples but a reduced order state vector can be employed along the lines developed in section 7.4.3.

7.5.1 [A B] & [C] Decentralised Control

By assigning $y_1=[y_A\ y_B]'$, $y_2=[y_C]$, $u_1=[u_A\ u_B]'$, $u_2=[u_C]$ and partition the B, C, K_c and K_f matrices accordingly, a pair of decentralised controllers can be obtained for the benchmark model in the [A B] & [C] configuration by utilising equations (7-15) and (7-16). The closed loop step responses of the decentralised control system are shown in Figure 7-14 and the corresponding BRG arrays for the closed loop system are shown in Figure 7-15. As expected, no decoupling is offered by this control scheme. However, a just detectable level of synchronisation is present between axes A and B. Because of the inherent (web) coupling between axes A and C, this synchronisation has also spread to axis C.

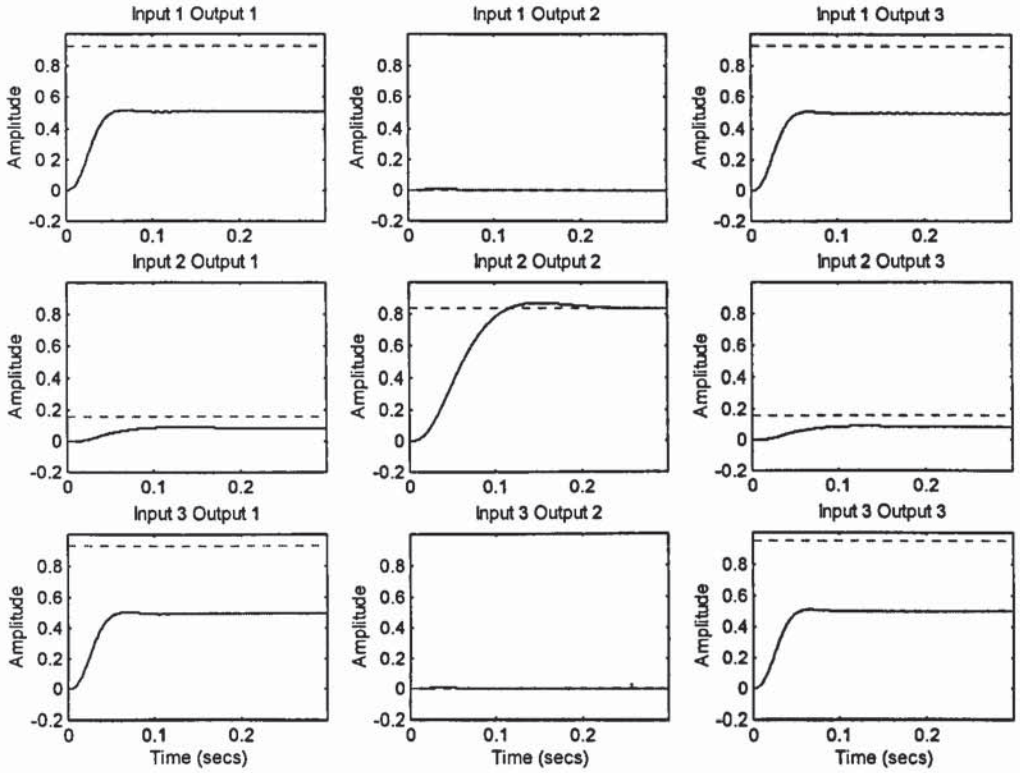

Figure 7-14 Closed Loop Step Response under Controllers with Masked Gains for [A B] [C]

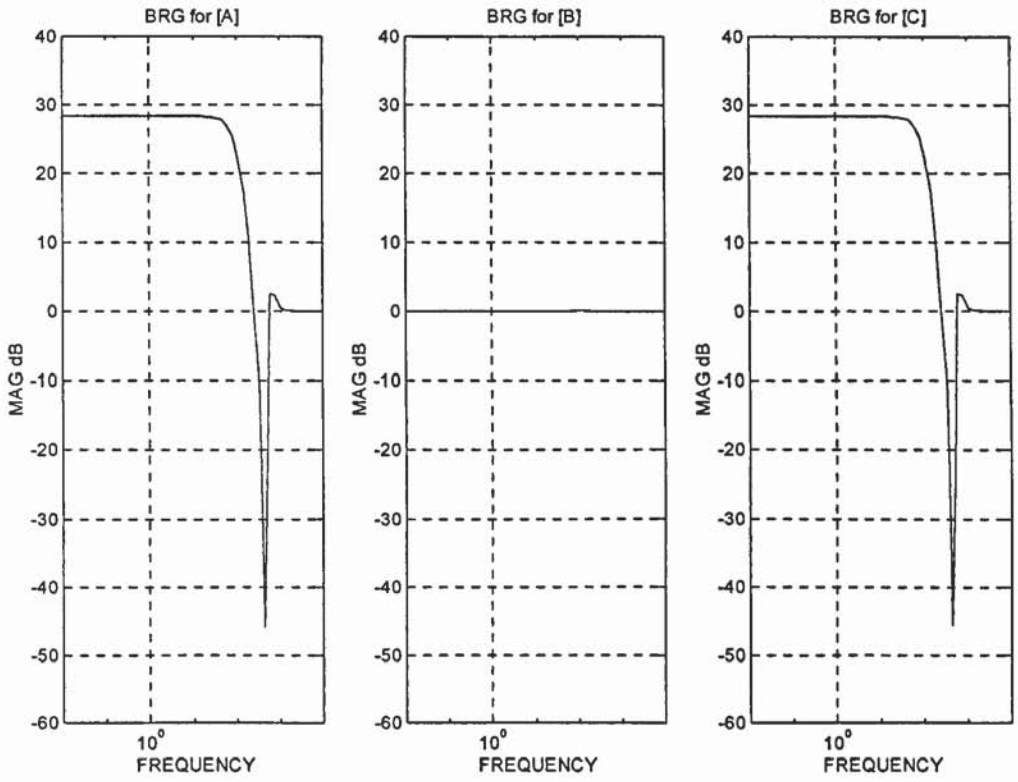

Figure 7-15 BRG Arrays for Masked Decentralised Controlled System

7.5.2 [A C] & [B] Decentralised Control

Similar to section 7.5.1, a 2-by-2 MIMO controller and a SISO controller can be constructed for the [A C] & [B] topology by assigning $y_1=[y_A\ y_C]'$, $y_2=[y_B]$, $u_1=[u_A\ u_C]'$, $u_2=[u_B]$ in equation (7-14) and partition the B, C, K_c and K_f matrices accordingly. The closed loop step responses of this combination are shown in Figure 7-16 together with the corresponding BRG arrays in Figure 7-17. As expected, the closed loop system exhibited no synchronisation but it is disappointing that no decoupling effects are recognisable either.

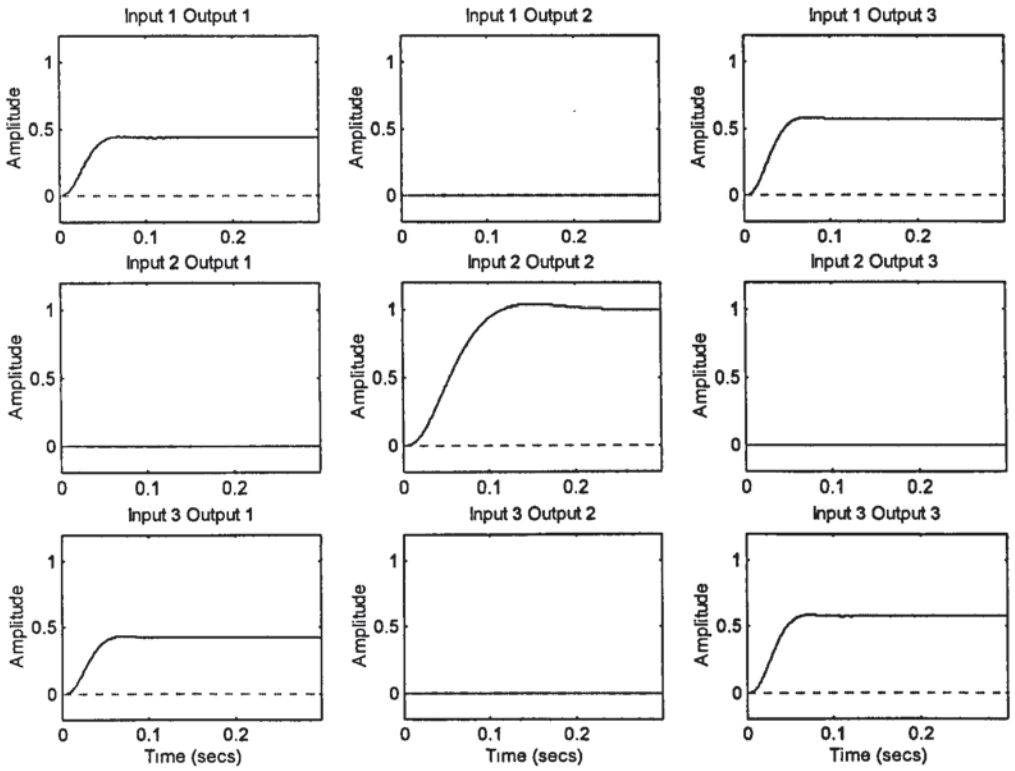

Figure 7-16 Closed Loop Step Response under Controllers with Masked Gains for [A C] [B]

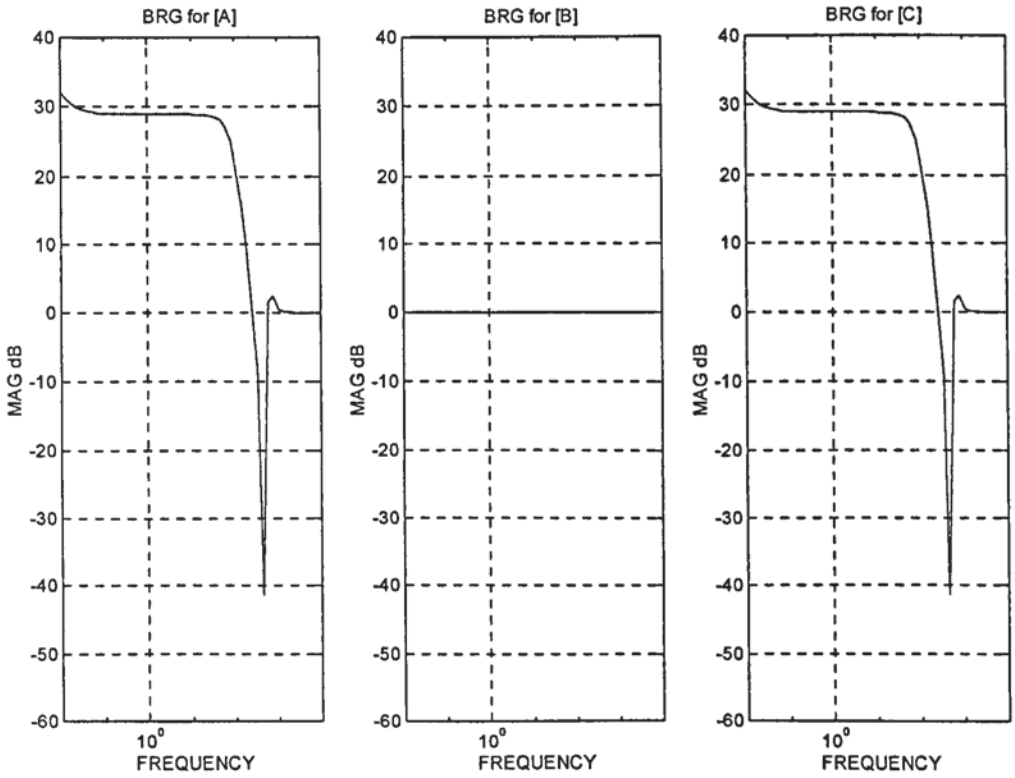

Figure 7-17 BRG Arrays for Masked Decentralised Control System

In this and the pervious section, variations on the theme of trying to extract a set of decentralised controllers from the full MIMO controller are described. The results generated are particular to the harsh requirements of the benchmark problem and it is not appropriate to generalise. The overall idea, however, is to extract as much as possible the system information available from the full MIMO controlled system. This is achieved by a combination of input/output/state vector and gains partitions together with an appropriate choice of reduced order plant model, bearing in mind the cost and limitation of actual implementation. In the general case, it is worthwhile to go through a similar exercise to see if a set of satisfactory decentralised controllers can be obtained. This is also an area where further investigation should be carried out.

7.6 Redesign Decentralised Controllers

Apart from extracting decentralised controllers from the full MIMO controller obtained in chapter 6, another approach will be to redesign the 2-by-2 and the SISO controllers anew for the adopted topology.

7.6.1 Redesign Decentralised Controllers for [A B] [C]

In this section, a synchronising controller for [A B] and a separate SISO LQG controller for [C] are designed for the benchmark model. In particular, the synchronising controller is designed using the method described in section 5.2.1. The design is made with a virtual stiffness of ξ= 10 over the full benchmark model but omitting the input and output of axis C in the process. The SISO LQG controller, however, can only be designed for an independent axis C (i.e. no web coupling) since a full plant model with only input and output of axis C is uncontrollable. Together, the state feedback matrices K_{AB} and K_C are generated and given below.

```
kc_AB' = 1.0e+002 *
 -2.92707129619237   1.19718571116038
  2.56156447876061  -0.53916404955548
  0.38594862208390  -0.60720697256163
  0.83392095434160  -0.36376756039455
  3.19818360198545  -1.31743945621575
 -0.36857453262009   0.23644337621047
 -1.55053708370977   0.81618949339067
  0.90613841424453  -0.36857453262008
  3.47730664346555  -1.37861709582648

kc_C' =
  3.16227766016840
  5.02466233706131
 11.28456640842161
```

The step responses of the benchmark model under this combination is shown in Figure 7-18. A comparison can be made with Figure 5-15, which shows the step responses of the benchmark model under LQG over all 3 axes with synchronisation on [A B]. Again no decoupling is offered as expected. However, a better degree of synchronisation is present between axes A and B than the controller obtained by simple masking. The degree of synchronisation between axis A and B is reflected in the BRG for the [A C] subsystem in Figure 7-19 which shows that the subsystem is not effectively isolated.

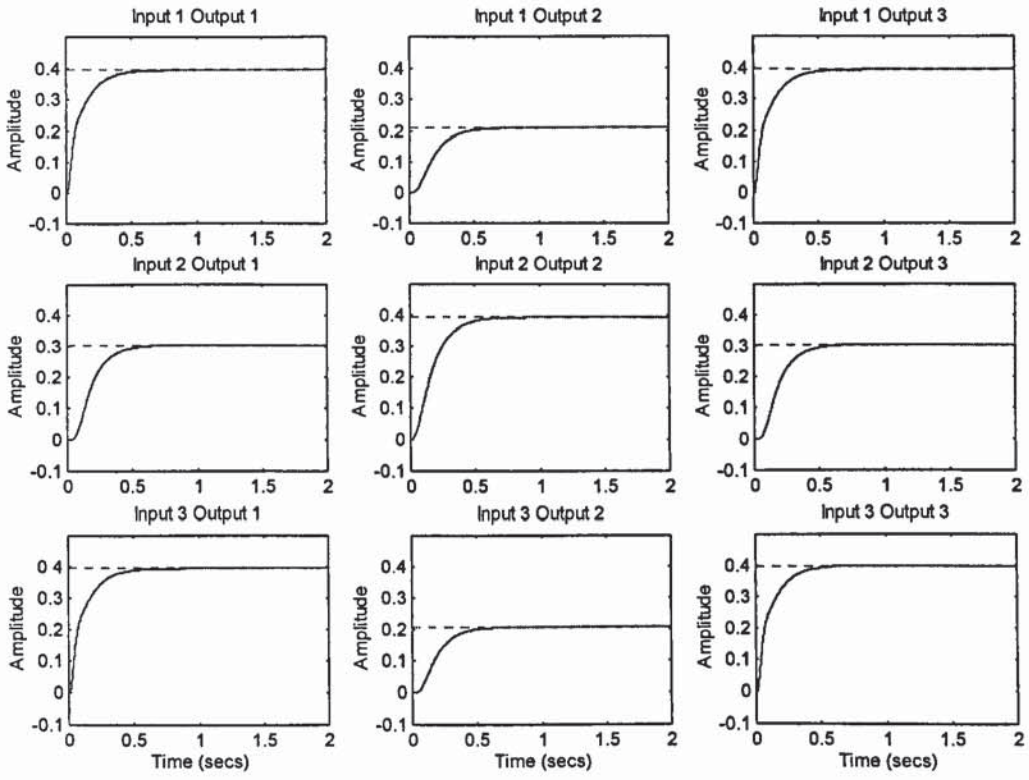

Figure 7-18 Step Responses : LQG Synchronisation on [A B] and LQG on [C]

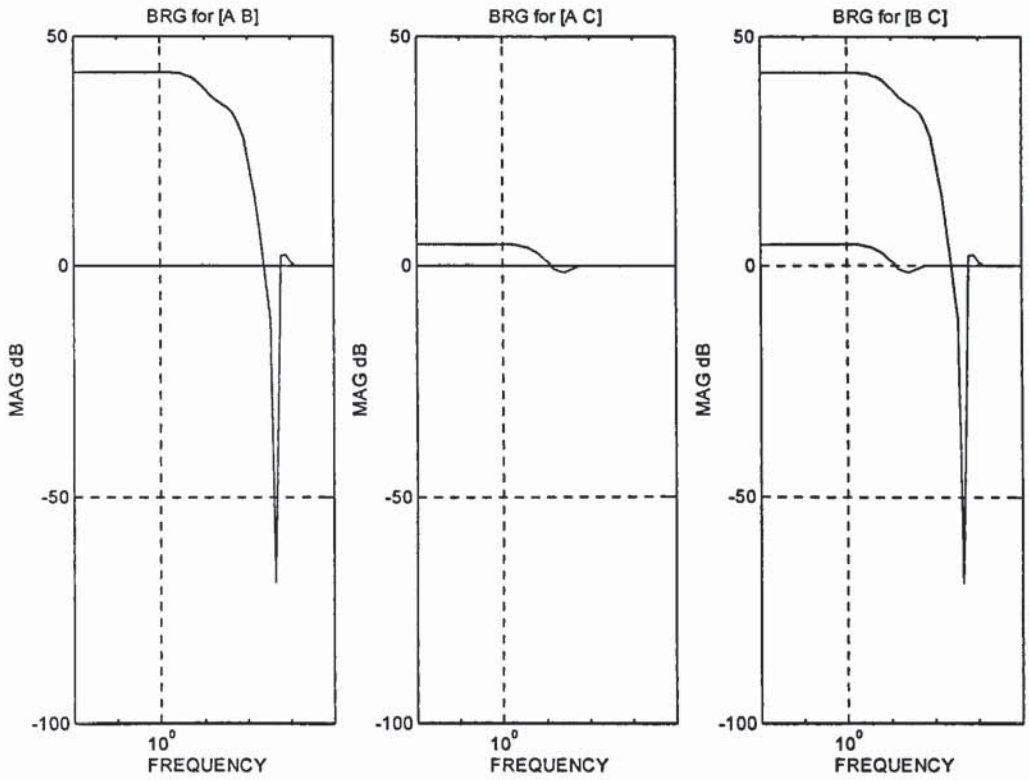

Figure 7-19 BRG Arrays for Redesigned Decentralised Controlled System

7.6.2 Redesign Decentralised Controllers for [A C] [B]

Since there is no interaction between axis B and the other two axes in the benchmark model, the subsystem [A C] can be treated as an isolated system and by running the amoeba optimisation, an appropriate state feedback decoupling controller can be designed for it. In fact, the described controller has been obtained previously in sections 6.3.3 and 6.3.4. Similarly, a separate LQG SISO controller can be designed for the independent axes B. With this combination of controllers, the state feedback step responses are shown in Figure 7-20.

Synchronisation is not present between axes A and B as expected. Decoupling between axes A and C is accomplished in a short time compared to the rise time of the heavier axis B. In Figure 7-21, a Kalman filter is used to provide the state estimate for the decoupling feedback. It can be seen that most of the decoupling effects have been reversed by the state estimator. This can be understood by considering the duality of the state feedback and state estimation problem in the LQG framework. Basically the Kalman filter optimises the errors in the state estimate which depends on the system dynamics, in this case the inherent coupling. It therefore suffers from the same kinds of difficulties (in reverse) as

trying to decouple with a LQ controller (details in chapter 5). Perhaps further efforts should be devoted to the investigation of decentralised observer/estimator.

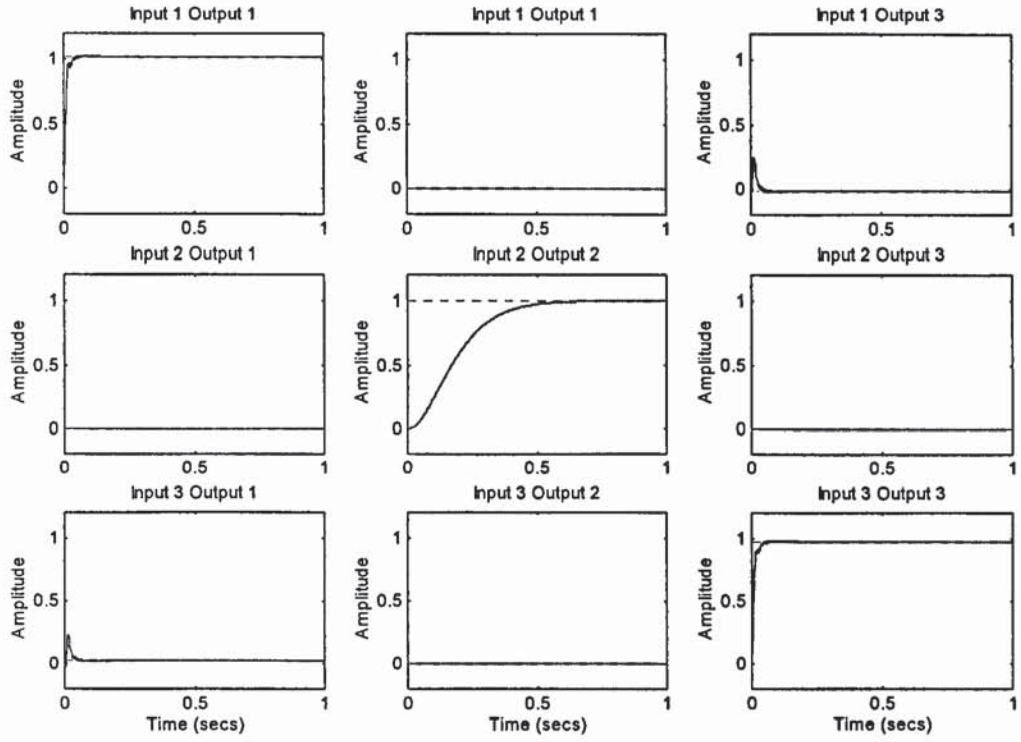

Figure 7-20 LQ Step Responses - Redesign Controller for [A C] & [B]

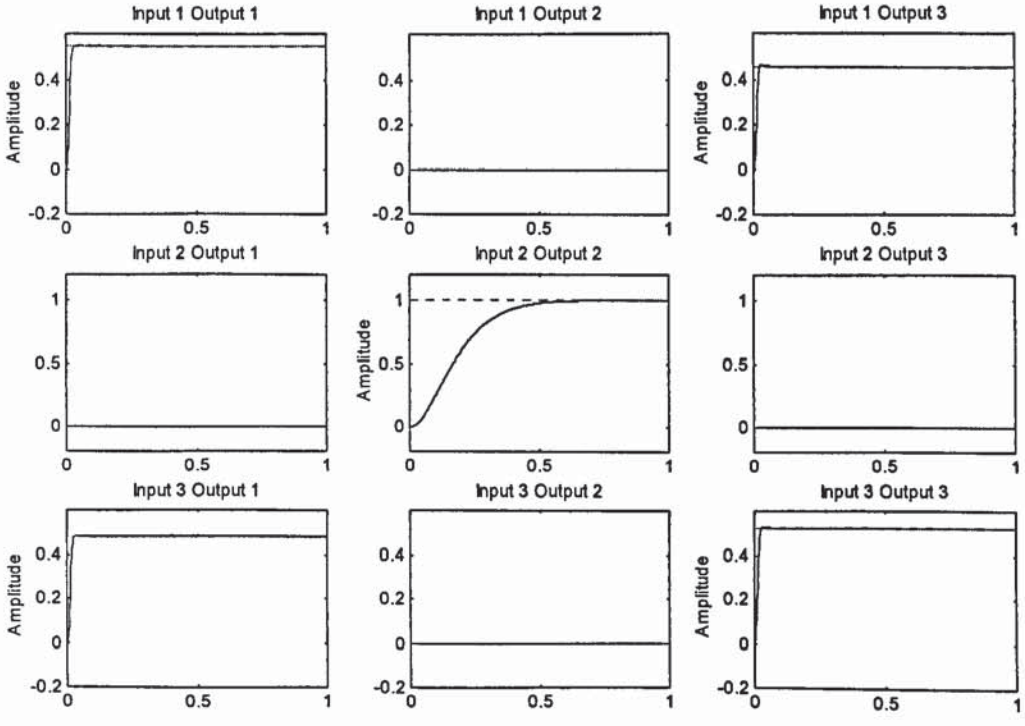

Figure 7-21 Step Responses - Redesign Output Feedback Controller for [A C] & [B]

7.7 Conclusion

The benchmark problem is set up with exaggerated demands on a machine which perhaps are not reasonable in real life applications. The deficiency in the performance of the decentralised control system reflects this fact. In more realistic applications, the machine will be more complex but the conflict of requirements are likely to be less pronounced since much of the potential conflicts will be excluded by the mechanical design in the first place. The merit of the methodology is in handling and balancing the remaining conflicts in an implicit manner.

7.7.1 State Feedback vs. Output Feedback

In the case of the benchmark model, axis A and C are coupled with a material web. The mechanism involves dictates that some states become unobservable in some of the possible decentralised configurations. This is part of the reason for the [A C] & [B] topology being superior to the [A B] & [C] topology. It may, at first sight, to be just what the conventional approach would prescribe; first decouple all axes (full decentralisation) and then synchronise to specification, which totally ignores the required levels and therefore the relative importance of decoupling and synchronisation. Furthermore, in the general case, when the largest possible subset of decentralisedly observable states can not be accommodated by the allowable size of decentralised observer/controller pair, the convention approach provides no answer. Some sacrifice in performance has to be made in order to obtain a workable solution. One possibility is the use of extra sensors other than the standard outputs which may improve the state estimation process and thus provide better decentralisation.

Subjected to the design conditions for each set of decentralised controllers, the [A C],[B] topology has outperformed the [A B],[C] topology.

7.7.2 The General Case

In the general case, following the logic of the design methodology, decentralised controllers are to be obtained after a preferable topology has been determined. This can be achieved either by extraction from the full MIMO controller or by re-designing smaller MIMO controllers for the appropriate subsets of input output pairs. The re-design of MIMO controllers, be it considerably smaller in size, is still a significant problem. Nevertheless, within the context of the design methodology, an approach has been

developed to synchronise, decouple (or synchronise and decouple simultaneously) a number of axes. This approach can be re-applied to the design of smaller MIMO sub-controllers. This is indeed the case in the benchmark problem where LQG synchronisation is applied to the [A B] sub-system in section 7.6.1 and the simplex method is used to generate a decoupling solution for sub-system [A C] in section 7.6.2. However, due to the limited number of axes in the benchmark problem, perhaps not enough information can be generated to properly assess the relative merit of the various approaches to the design of MIMO sub-controllers.

7.7.3 Revisiting the Design Methodology

Chapter 6 provides a full MIMO controller by first fully decoupled the plant and then a synchronising controller is added. This would be the theoretical best if it could be implemented. However, considerations such as the computation and communication cost for state estimates and high dimensional controllers together with the constraint of present day hardware for real-time applications limit the size of plausible MIMO controllers. The methodology suggests the most viable partial decentralised control scheme by comparing BRG arrays of the alternative configurations subject to the definition of BRG.

8. Discussion

8.1 Design Methodology

The aim of this thesis is to develop a design methodology for the servo system of a class of machinery. This machinery is typically used in high volume manufacturing which requires large numbers of co-ordinated sensors and actuators (usually rotating shafts) running at high speed. The current trend of replacing mechanical connecting or synchronising parts by integrated electrical, electronics and computational parts, or the application of what can be termed the mechatronic principle, in the design of machinery offers a number of advantages. First of all, it brings along a reduction in the number of moving parts. Instead of a complex construction of inter-linked mechanical moving parts such as cranks, gear boxes, etc. (Figure 1-8), a machine now resembles more to a collection of electrically powered actuators controlled and co-ordinated by dedicated computers. This results in simpler and cheaper mechanical design, which implies faster prototyping and hence lower development cost. It also means a shorter down time and hence lower maintenance cost. Also better overall control of the machine can be achieved since the task of complex operations is transferred to computers. Rigid mechanical linkages between the actuators are now replaced by independently driven servos that are controlled and co-ordinated by computers. More intelligent behaviour and a higher degree of fault tolerance can be expected and these all contribute to the machines' reliability. Also important is that better flexibility in configuration and operation can be obtained. In today's market, values are added to the products by increasingly higher degrees of customisation, which means productions are often of small batch sizes and process modifications are required frequently. Overall, the mechatronic approach has a very good potential for better machine performance.

8.1.1 Process Oriented Design

At present, the replacement of mechanical parts by "mechatronic" parts has already taken place to a large extent. Machines with a large number of shafts that were connected by mechanical linkages and driven by a single motor are now gradually replaced by multi-axis system with axes that are driven and controlled independently. These machines have been available on the market for a number of years. However, a formal design

methodology is still lacking for the design and development of the servo systems for this new class of machinery. Much of the design process relies on the engineer to draw on his experience and often the designs are over conservative. It was identified in chapter 4 that various kinds of interaction are often present between the shafts. Currently, there are two ways to avoid the potential problems they can cause. One way is to have the strongly interacted shafts linked together mechanically and driven by the same servo axis. This approach stops short of applying the mechatronic principle fully and may put undue restriction on the design. Another approach is simply to ensure that each shaft has a very high local stiffness, which could overcome any undesired interactions and as a result produces a very conservative design. In essence, the servos are over-designed at a local level to ensure the ease of global integration at a later stage. However, if the interactions between the shafts are taken into account in the early conceptual design, a more integrated system and hence better performance is possible.

8.1.2 Generic Machine Design

As described in earlier, the new generation of machine in its most basic and general form can be viewed as a collection of electro-mechanical devices (sensors and actuators) controlled and co-ordinated by dedicated computers. Apart from the flexibility in operation offered by the electronics, there is also more configurability in other dimensions. For example, the machine can now be designed as a collection of detachable modules each performing different functions. The overall functionality of the machine can be extended relatively easily by attaching additional modules, which can be easily integrated by a change of control software. Each module can also be commissioned and tested individually. In fact, a similar strategy has long been used in software design. In a way, the development of a generic machine design can be viewed as a natural extension of this concept.

Both the processes oriented and the generic machine design approach goes through the following stages:

- manufacturing process design
- servo selection
- control system design

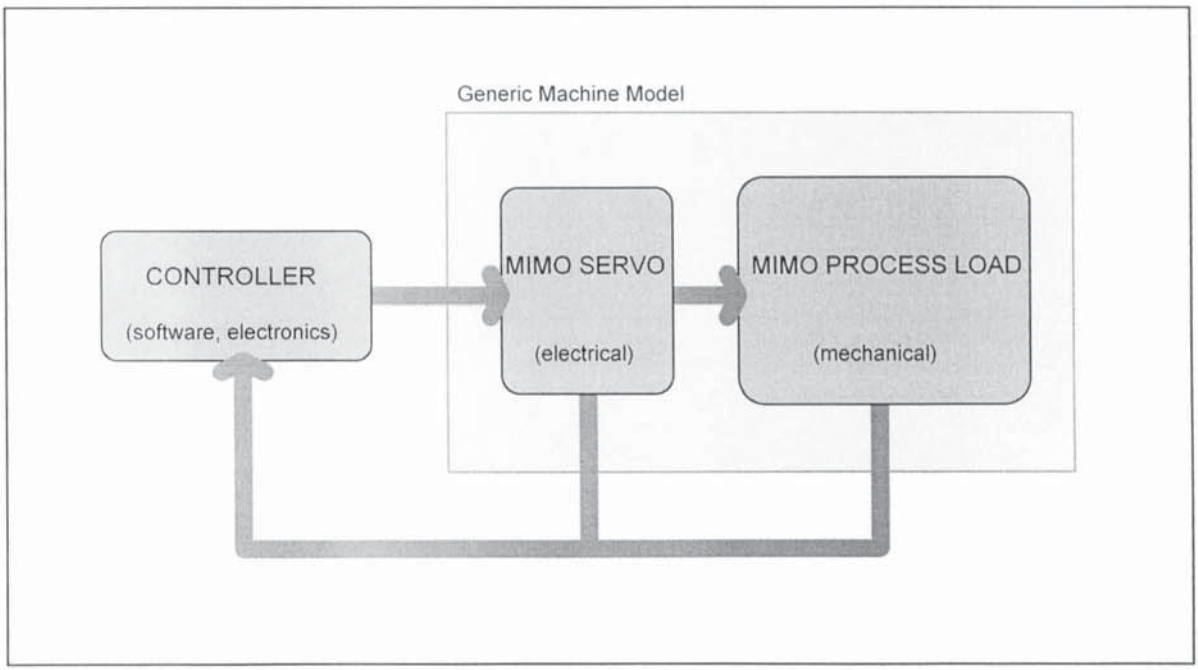

Figure 8-1 Schematic of the Generic Machine Model

In the process oriented approach, critical sequencing is achieved by mechanical means and driven by a single axis. However, in the generic machine design, the critical sequencing is done by software co-ordination of several axes. In practice, the design is likely to be a mixture of both approaches depending on the various physical restrictions, for example, the bandwidth and accuracy achievable by the available servo technology and the flexibility required. It is therefore clear that the design methodology provides more options in the design of manufacturing process. As an example, certain event sequences that had to be done by mechanical means can now be achieved simply by specifying motion profiles in the control software.

In the methodology, the concept of a generic machine is incorporated into the construction of conventional multi-axes machines. A whole class of machinery with similar functionality can be constructed out of the same core conceptual machine with reusable modules. Since the design specification is often dictated by the manufacturing process that the machine has to carry out, the actual design and the process requirements can be treated as the "add-ons" or modifications to this core machine. In practice, the design process is often limited by resources and time constraints. The success of any new implementation depends as much on the quality of the method as the commitment and determination of the implementor. The design methodology is in essence a set of consistent and integrated concepts and approaches and to its aid is a number of procedures and software tools. The

best way to present it is to look at a working example. To this end, this thesis is aimed to show the development and the validity of the methodology by illustrative examples.

8.2 Generic Machine Model

In this thesis, a design methodology for the class of machinery concerned is proposed which takes advantage of a full mechatronic approach. A MIMO paradigm is introduced at the outset in order to provide a framework to deal with the interactions between actuators. The first step towards the methodology is the construction of a generic machine model, which is described in detail in chapter 4.

8.2.1 Model Complexity and Organisation

The first issue which must be addressed is the complexity of the model since a large variety of devices and loadings may be involved in the construction across a wide range of machinery. The concept of genericity provides a natural framework that organises the model in a modular structure. Distinct classes of physical parts are represented, as far as possible, by their corresponding models, which are encapsulated in component modules with uniform inputs and outputs. Similar conceptual components are collected together and organised into a layered structure. In the top layer, the overall model is simply a servo module driving the MIMO load module. Going down the layers reveals progressively more details. The servo module consists of a number of axes which are actually motors together with their drives whereas the MIMO load is a collection of process actuators. In other words, the servo module is really a collection of electro-mechanical devices, which is precisely the core of the machine and the load is really a MIMO collective description of the process actuators, which are to be driven by the servos. In short, the modular structure ensures the portability and reusability of the model and thus facilitates the trial of new ideas and shortens the design and development cycle. Moreover, it also opens up the possibility of the development of a so-called icon based program or design environment, which is much welcomed by industries for its user friendliness.

The generic machine model is typified by having a large number of states. Large scale system such as electric power distribution network, which may involve hundreds of states, have long been an active area of research. The models of these systems are treated with great simplifications. However, for the type of machines under consideration, since the time constants, the sampling cycles and accuracies involved are different by many orders

of magnitude, the same degree of simplification will not be acceptable. Some form of formal model order reduction procedure will be needed to obtain a model that is minimal (no redundant states) and acceptable for the bandwidth concerned. For example, it is conceivable that the fast dynamics of the interactions may never show up simply because the bandwidths of the axes are relatively too low to react. As to how far the reduction should go, the idea is to retain as much information of the dynamics as possible and yet the number of states should be manageable so that the problem of controller design still remains reasonably straight forward. In this thesis, only little consideration has been given to the area of model order reduction and more investigations will be needed in the future. The 3-axis benchmark model considered in chapter 5 has a total of 9 states. The number of states was kept to the minimum to avoid the need of model reduction and so manageable and yet adequate to be used to develop and illustrate the design methodology.

8.2.2 Model Evolution and Convergence

Another foreseeable problem is the physical absence of the component parts during the early stages of design. This is particularly true in the practice of concurrent engineering where the design, development and prototyping as well as the setting up of production of a product are carried out in a parallel rather than a linear fashion, so as to reduce the concept to market time. Models of individual components are often available in the form of similar and reusable module from previous design. If not, simple linear models can be constructed as a first approximation. A reasonable approach therefore is to base the initial design on a generic model comprising some reusable and some approximated modules. As the development of the machine progresses and physical parts became available, modules can be replaced by better models constructed with appropriate measurements from these physical parts. Thus, these designs will be iterated throughout the design cycle with a number of experimental verifications at various stages and the overall model will converge to reality accordingly. This process clearly requires the input of the design engineer, especially at the initial stages. However, the organisation of the generic model is there to retain and collect as much as possible the expertise involved, in the form of libraries of reusable modules, so that it may be passed on to future designs.

8.2.3 Cross Axes Interactions

One particular area of the model, which may be new to the current designs, is the modelling of cross axes interactions. There are two categories of cross axes interactions,

namely, imposed and inherent couplings. Imposed couplings are those interactions which are required to synchronise different axes in order to guarantee event sequences. This is achieved by software control and hence the interaction is *virtual* and is not modelled in the generic machine. What needed to be modelled, however, is inherent couplings and a detailed description is given in section 4.5. Four types of inherent couplings were identified:

1. structural
2. web
3. DC
4. hybrid

In general, they represent the physical interactions exist between the axes and are the components which bring a MIMO dimension to the model. Hence harnessing further performance and flexibility through the use of MIMO controllers becomes possible. Note that these inherent interactions may or may not be desirable and therefore may need to be enhanced or compensated accordingly (see section 5.1.1). Although each type of inherent couplings comes in many different forms and variations, it should be possible to put them into similar format as those described in section 4.5. As for structural and web couplings, it should be possible to put them in the same feedback structure as long as they are modelled with first order liner dynamics, as in the examples given in section 4.5.

8.2.4 Determination of Physical Parameters

The determination of most of the physical parameters for the generic model is a matter of standard practice but perhaps some explanation is required for the case of inherent couplings. The determination of parameters for the interactions depends on the details of how these interactions are modelled, which in turns may depend on what parameters can be measured or inferred from the actual interactions. In the case of structural coupling, the approach to modelling the vibration behaviour of mechanical structures is well established. An example based on applying the Ritz method to a flexible platform is given in section 4.5.2.3. In the case of the other types of interactions, standard electrical and mechanical analysis should be suffice to produce the necessary model and provides an approximation for the overall generic machine model in the first instance. The corresponding physical parameters can be determined by performing simple

measurements. At a later stage, when more physical parts are available, it is possible to calibrate these parameters by some established procedures. For example, with appropriate equipment, MIMO transfer function arrays measured from the available parts can be used to tune the parameters and thus match the model to the machine (see appendix on transfer function matching for servo axis).

8.3 Centralised MIMO Controller Design

Apart from the potentially more flexible process design, the other major difference between the process oriented and the generic machine design approach is the construction of the controller. Instead of a number of SISO loads, the object that has to be controlled now is a full MIMO representation of the machine. It is of course still possible to design a series of SISO controllers but one should take advantage of the possibility of a MIMO controller. As discussed in section 3.7, a full centralised MIMO controller is to be designed for the model in simulation but eventually a block topology is to be selected and a set of decentralised controllers determined. To further the discussion, a simple benchmark model is selected so that investigation can be made into which MIMO controller design method is best suited for the performance requirements, specifically, the simultaneous decoupling and synchronisation of various combinations of motion axes. To simplify the problem, however, synchronisation and decoupling were initially considered separately.

As mentioned in chapter 3, the main focus of this thesis is the determination of a suitable topology for the decentralised control of a specific class of machinery and it is not the aim of this thesis to improve on a particular MIMO controller design method. Some methods will always converge to a solution by their very nature, even if it does not satisfy all the performance requirements, whereas some methods will generate no stable solution at all. It is also important to note that performance specifications are often drawn simply from desirable machine behaviour and it is not always possible to achieve them physically. It is for these reasons that the design methodology does not include a stage to prove the existence of a solution, which itself also depends on the method employed to obtain it. Furthermore, such a task is considered too involved, given the vast number of possible variations and details. Instead, by observing the behaviour of the solutions through trial runs, some insights were gained into whether the specifications were reasonable and how far the designer is from an acceptable solution.

8.3.1 Linear Quadratic Optimisation

Bearing in mind that one of the objectives of the methodology is automation and that it should assist engineers with limited knowledge of control theory to carry out the servo system design, a method such as linear quadratic optimisation seems to be appropriate at the first instant. Once a state-space model is available, the rest of the design involves only the tuning of weighting matrices, using appropriate heuristics. The study, as described in section 5.2, shows that a MIMO controller with good synchronisation property can be obtained by assigning fictitious spring stiffness between motion axes that have to be synchronised and recasting the problem into a standard linear quadratic (LQ) optimisation format.

However, the deployment of an LQ regulator has failed to provide adequate decoupling. Whenever cross axes interactions are presented in a MIMO state space model, each state is influenced by one or more inputs, directly or indirectly through chains of other states. The synchronisation requirement, through the re-combination of states, can be cast into an optimisation which forces the output states to behave similarly. On the other hand, the decoupling requirement demands an independence of certain states which can only be achieved by compensating for the undesirable influence on the states concerned. This compensation of states, however, is not in general consistent with the optimisation which, in this case, is the minimisation of a cost function.

Although it is possible to simply assign a very high weight to the output of the axes to be decoupled and hope that the feedback controller will force the coupling effects to manifest themselves only beyond the bandwidth concerned, attempts of this nature have been proved to be futile. Attention is then turned to the use of a LQ tracking controller which is a combination of a LQ regulator and a compensator in the forward path. It is shown in section 5.2.2 that such a controller will have a time and profile dependent control trajectory, which after some investigation, is found to be too demanding to be implemented on the hardware available.

8.3.2 Decoupling Controllers

The idea of designing a forward path decoupling controller explicitly for the compensation of undesired interactions was at first resisted because such a controller will introduce extra states to the overall system that has already got a problem of having too many states. Nevertheless, the synchronisation problem has effectively been solved by the LQ method

and it is conceivable that the LQ method can be applied to a decoupled system to produce a solution that is acceptable, although this would not be favoured, since it introduces another stage to the controller design and complicates the methodology further.

The method of decoupling by pseudo diagonalisation involves the construction of a compensator in the forward path which optimises the column dominance of the closed loop system. Decoupling by linear state feedback, as described by [Falb & Wolovich 67], on the other hand, involves both a feedforward and a feedback controller. Both methods were investigated and are detailed in section 5.3. In either case, the compensator achieves decoupling only for the higher frequencies and the system remains coupled in the lower frequencies. As a quick exercise, LQ feedback controllers were designed over these partially decoupled systems but the results were less than satisfactory. Another possible approach is the parameter optimisation method of Edmunds' mentioned in section 5.3.4. Although it demonstrates reasonable success with the benchmark problem, the result depends on the pre-requisite that the designer is able to select an appropriate controller form. Therefore, despite its potential, it was decided not to pursue any further.

8.3.3 Iterative Approaches

Standard controller design methods have so far failed to deliver the required controller, namely, a controller that simultaneously synchronises and decouples pre-defined sets of motion axes. The difficulty arises from the fact that each standard method has its unique approach to a problem which is tied very closely to how the problem is formulated. The requirements of synchronisation and decoupling are not only potentially conflicting objectives, but the natural approaches to their solutions are also very different. Within the LQ framework, for example, synchronisation can be formulated as the minimisation of certain "energy" associated with the relative position error between the axes concerned. However, no such "energy" can be associated with decoupling. It is for this reason that attention has been turned to non standard methods which perhaps are more heuristic in nature and may produce a one-hit procedure, and yet requires only minimal interaction with the designer. In chapter 6, the possibility of obtaining the required controller by numerical methods is investigated. Initially, emphasis is put on finding at least one working solution for the decoupling problem.

Two search strategies based on the movement of the performance vector relative to the specification boundary in the performance space were devised and investigated. Based on

the controller obtained for a plant with independent axes, the first strategy approaches the solution iteratively by imposing the coupling constraints in incremental steps, while keeping the tracking property at all time. The position of the performance vector is monitored and kept at the acceptable side of the specification boundary. The iterative process turn out to be oscillatory and never quite converging to within the specification boundary. The second strategy is a modification of the first one and it takes into consideration of the valid range of linearity of the performance as a function of the controller and the plant, when the performance vector is updated. After some tests, neither of these approaches were found to be favourable.

8.3.4 Numerical Optimisation

Attention was turned to the downhill simplex method or the amoeba algorithm, which is a multi dimensional optimisation of a performance index in N variables. This method has the advantage that only function evaluation is needed and derivatives are not required. More importantly, optimisation of this kind will guarantee at least a local minimum, even if it may not be an acceptable solution.

Again, taking the decoupling benchmark problem as an example, a polyhedron with $N+1$ vertices which is otherwise known as a simplex, "feels" its way downhill by performing one of a number of operations, as described in section 6.3.1. After some trial runs, a suitable performance index is constructed as in equation (6-8) to (6-10). It is immediately clear, from the BRG arrays as shown in Figure 6-9 and 6-10, that this heuristic method produces a far superior decoupling controller for the benchmark plant than the design methods investigated so far. Attempts for a synchronising controller or a simultaneously synchronising and decoupling controller with the simplex method, however, did not produce a convincing solution.

Finally, an acceptable solution was obtained by first decoupling the target plant with the downhill simplex method and then a synchronising LQ controller is designed around the decoupled plant. Since both procedures are based on the same state feedback structure, they can be superimposed to form one overall controller without the addition of extra states. The trade-off between these the conflicting requirements of synchronisation and decoupling is illustrated in Figure 6-39 which shows the BRG arrays of the same decoupled plant under LQ synchronisation with different synchronisation stiffness. With the simultaneous synchronising and decoupling controller in place, BRG arrays of the

closed loop system can be plotted and the overall level of interaction can be assessed. Also the BRG arrays of the MIMO controller can also be plotted from which a suitable topology for decentralised control can be determined, subjected to the specified level of synchronisation (section 6.7).

8.4 Topology Selection - A Unified Approach

As discussed in chapter 3, the central idea of this thesis is to find a MIMO controller that represents the best compromise between the conflicting requirements. Once this is achieved, the required level of imposed interactions will be captured by the off diagonal terms of the MIMO controller and BRG arrays of the controller can be plotted for the likely alternatives. The selection of the decentralisation topology can therefore be made quantitatively by comparing the magnitude of the imposed interaction off the block diagonal, that is the interaction between the subsystems, of the alternative topologies through the BRG arrays. If all the external inputs are known precisely, the interactions between the subsystems can be modelled with prescribed properties and can be treated as noise or disturbances for the decentralised controllers.

In the benchmark problem, the BRG arrays in Figure 6-33 and 6-34 show that a good degree of synchronisation between axes A and B and decoupling between axes A and C is achieved by the closed loop system. It is hypothetically suggested that 2-input 2-output MIMO controller is the limit of implementation and therefore some form of decentralisation is required. Figure 6-35 and 6-36 show that for this particular MIMO controller, the [A C] subsystem is more isolated than the [A B] or the [B C] subsystems, that is the BRG for [A C] is closer to an identity matrix than the BRG's for the other subsystems. This implies, in this particular case, that the efforts or interaction involved to decouple axes A and C is "greater" than that required to synchronise axes A and B. The preferred topology is therefore [A C] and [B] and a 2-by-2 MIMO and a SISO controller can be designed accordingly for the subsystems. Note that the new controllers are taking into consideration only the relevant pairs of input and output of the overall generic model, which, incidentally, remains a complete representation of the plant and produces the full simulated behaviour.

Note also that, for completeness, there is the alternative of implementing two 2-by-2 MIMO controllers in the overlapping configuration of [B A] and [A C]. Reference on the subject is given in section 2.3.1.4. and it is also discussed further in section 8.8.2.

8.5 Decentralised Control

Once a block diagonal topology is selected, a suitable set of decentralised controllers has to be determined. Since by this stage of the methodology, a full centralised MIMO controller is available, a natural step to take is to somehow extract the appropriate decentralised controllers from it. In chapter 7, a number of extractions are investigated. Their difference mainly lie on how much information is passed on from the full to the decentralised controllers. Namely, if full or partial details of the plant model, the state vector or the gains are used to construct the decentralised controllers.

Another approach is to redesign MIMO controllers for each of the block of the selected topology. Again, a number of variations are possible depending on how much of the complete plant dynamics is retained in the design.

Chapter 7 demonstrates that the topology suggested by the methodology, that is block diagonal structure corresponds to the most isolated subsystems in terms of BRG arrays, indeed produces the best decentralised scheme, subjecting to the conditions of the benchmark problem. With its relatively simple and exaggerated requirements and therefore predictable decentralised control structure, the benchmark problem illustrates the working of the generic machine design methodology. The full advantages of the proposed approach is clear when more complicated requirements are to be met for a more complex machine.

8.6 MIMO Test Rig and Experiments

The content of this thesis has been largely devoted to working through the benchmark problem as described in section 5.1, which illustrate the main ideas of the design methodology. Theoretical ground work and computer simulations were developed to demonstrate how a suitable set of decentralised controllers can be designed for the best control topology. However, the design methodology will remain an academic exercise unless it can be proven by hardware implementation. To this end, a custom built MIMO test rig was commissioned in the laboratory. This test rig together with the experimentation were developed in parallel with the simulation work. The servo hardware used were essentially identical to those described in section 1.5.2. In fact, the benchmark problem and the test rig have been constructed to reflect each other. For the clarity of presentation, the description of the test rig and the experiments were collected and put in the appendix.

8.7 Reference to Current Literature

In chapter 2, a survey and summary of the relevant issues in current literature is given. It represents a vast amount of background information yet only limited proportion of which proved to be applicable to the development of the design methodology. Some particular issues are highlighted below.

8.7.1 General Approach

In all of the available treatments of large scale systems, and more specifically, decentralised control, inevitably, the approach is to first solve the so-called pairing problem [Jamshidi 83, Sandell 76, Maciejowski 89 & Siljak 91, 96]. This is the (off-line) search of a suitable configuration based on identifying a partition with the "smallest" interactions between sub-systems in the open plant. As a result, a complex system is decomposed into a set of interconnected systems, to which decentralised control can be applied. However, the particular requirement of the design methodology demands a simultaneous treatment of both the inherent and the imposed interactions, which means that decentralisation has to be considered in a closed loop setting. The pairing problem, as it is usually understood, has to be viewed in a different light and a different approach, as presented in this thesis, is developed.

8.7.2 Decoupling and Synchronisation

Efforts in tackling the general problem of decoupling MIMO system are well represented in the literature. Approaches include open loop compensation (e.g. pseudo diagonalisation), closed loop feedbacks [Falb & Wolovich 67, Gilbert 69], sub-space analysis [Wonham & Morse 70] and optimisation [Edmunds 79, Xia et. al. 93], to name but a few. The problem of (block) diagonalisation is often discussed separately for its generality and relevant material is again abundant in the literature [Feingold & Varga 62, Ohta, Siljak & Matsumoto 86]. Ultimately, the required solution depends on the inherent structure of the plant and the design methodology is specifically characterised by this.

Synchronisation, on the other hand, is an imposed demand external to the plant. Whereas the decoupling of two motion axes require them not be influenced by each other, the synchronisation of two motion axes require them to follow each other, both based on the information exchanged between them. Their conflicting nature is clear. It appears that the synchronisation problem is not usually treated in general terms in the literature and often

discussions are focused on specific applications [Danbury & Jenkinson 94, Hu et. al. 90]. In the context of decentralised control, it is never discussed, probably for the assumption that synchronisation can be applied once a system is decoupled.

This thesis, however, follows a different sequence of steps. A controller that simultaneously synchronise and decouple different subsets of inputs and outputs has to be determined on the outset. After studying a number of analytical methods (refer to chapter 5), none has proved to be successful in accommodating both decoupling and synchronisation, although most design methods seems to be good at either the former or the latter. The results of synchronisation by LQG is particularly outstanding. Albeit this study is by no means conclusive, optimisation methods (refer to chapter 6) were developed instead to speed up the development of the methodology.

8.7.3 Decentralised Control Systems

There are a number of approaches on the formal treatment of decentralised (linear) systems currently being developed by various researchers. Some representative works include geometric theory [Wonham 79, Commault & Dion 92] which focuses on sub-space analysis, algebraic theory [Gundes & Desoer 90] which focuses on the factorisation of transfer functions and graph-theoretic approach [Reinschke 88, Siljak 91] which focuses on the connectivity between states. In [Trave, Titli & Tarras 89], it was shown that much of the afore mentioned approaches were essentially different constructions of the same concepts. These concepts provide important tools to characterise decentralised systems, namely, stability, structural robustness, etc. More importantly, the existing approaches concentrate on the decentralisation of open loop plants, whereas in the context of the present design methodology, consideration has to be given in a closed loop setting. Specific theoretical development, which caters for the decentralisation of closed loop control systems, has to be made before these approaches are adaptable to the design methodology.

8.8 Future Work

This thesis attempted to lay a foundation for a generic machine design methodology which by definition has a very wide scope and covers a large number of issues. As much as the author may like, it is impossible to deal with them exhaustively. Based on the work to

date, and referring back to the survey in chapter 2, the following matters are suggested to be investigated further.

Refinement of the Generic Machine Model

- Introduce non-linear elements
- Introduce standard process mechanism
- Substitute (measured) transfer functions as component models
- Consider other interacting loads (e.g. perpendicular load in machine tools)

Structural Analysis

Develop and utilise formal analytical tools to characterise structural properties of model. Attention should be given to the concepts of structural stability, controllability, observability and fixed modes among others. Graph theory has a clear advantage since visual aids are always desirable.

Decentralised Control

In the context of the design methodology, a centralised MIMO controller representing the theoretical 'best' is always available before decentralised controllers are implemented. The answer to how to extract the best decentralised controllers from the centralised controller may have a wider application than the present context and deserve more attention. Another interesting issue is the use of overlapping decentralised control and this is discussed further in section 8.8.2.

Decentralised State Estimators

The question of decentralised state estimator can be treated, to a certain extent, as the dual problem of decentralised control. Output feedback can only be guaranteed if satisfactory state estimates are available.

Block Decoupling Methods

In the benchmark problem, decoupling between two axes is achieved by the downhill simplex optimisation. In general, however, decoupling requirements between blocks are the more likely specification. Although block decoupling requirements can always be translated into compounded axis decoupling

requirements, it would be preferable if this can be achieved in some unified fashion. The following approaches may serve this need.

- *Scalar-field approach:* The parameters of the controller are optimised so that the actual trajectories of the motion axes will follow the path of minimal energy over a virtual potential field that varies along the space-time co-ordinate of the desired trajectories. These trajectories should catered for the simultaneous requirement of synchronisation and (block) decoupling. In [Danbury & Jenkinson 94], a description of the construction of a suitable potential field is given.
- *Block Gershgorin methods:* With the use of the generalised Gershgorin theorem (section 2.3.2.1), the plant model can be block diagonalised and decoupled accordingly.
- *Near-Decoupling by LQ Optimisation :* Refer to section 8.8.1.

Time Simulation

In the study of the benchmark problem, attention has been put on the steady state responses of the machine model. For a practical solution, however, consideration should also been given to the start up, shut down and other transient behaviour.

Experimental Verification

With reference to the section on experimentation in the appendix, it is regrettable that no workable MIMO controller has been implemented within the duration of this project. Nonetheless, it is still a less than ambitious goal, provided one can invest in a suitable control hardware. Once a general purpose MIMO control algorithm is available, it should be relatively easy to put the various topologies and decentralised controllers to trial.

8.8.1 Near-Decoupling by LQ Optimisation

By specifying synchronisation in terms of stiffness of hypothetical springs, it is found that synchronisation requirement can be accommodated by the standard LQ framework and produce satisfactory results. However, the standard LQ method fails to provide an adequate level of decoupling and hence compensation for undesired interactions. Recently, it has come to the attention of the author that [Xia, Rao, Sun & Ying 93] has suggested a

near decoupling optimal controller design by optimisation on cross channel influence. Referring to the plant's linear time-invariant state space model as the process model .

$$\dot{x}_p = A_p x_p + B_p u_p$$
$$y_p = C_p x_p$$

(8-1)

Assuming the desired decoupled transfer function matrix of the closed loop system to be

$$T(s) = diag\{T_{11}(s), T_{22}(s), \ldots, T_{pp}(s)$$

(8-2)

which is diagonal and non-singular and can be transform into a reference state space model

$$\dot{x}_{mi} = A_{mi} x_{mi} + B_{mi} v_{mi}$$
$$y_{mi} = C_{mi} x_{mi} \qquad \forall \mathrm{i} = 1,2,\ldots,\mathrm{p}$$

(8-3)

The process model can be made to follow the reference model by a near decoupling controller which minimises the following cost function

$$J = \sum_{i=1}^{p} J_i$$

(8-4)

$$J_i = \int \left\{ \left(y_{pi} - y_{mi}\right)^2 q_{ii} + \sum_{j \neq i}^{p} \left(y_{pj} - v_{mj}\right)^2 q_{ij} + u_p^T R_i u_p) \right\} dt$$

(8-5)

with the 3 terms in the bracket from left to right dealing with

1. dynamic tracking property
2. decoupling performance
3. amplitude of control signals

and the q's as weighting function for the relative importance of tracking and decoupling. The control signal involves the states of both the process and the reference models, and also the inputs and the disturbances. In this way, both the tracking and the decoupling properties of the control can be ensured. This optimisation, however, will require the system to similarly transformed by some suitable matrices so that the above cost function can be recast into the standard LQ format. This appears to be a comprehensive and consistent approach to obtain a solution. It may therefore be worthwhile to develop this further to cover both the issue of synchronisation and block decentralised structure.

8.8.2 Overlapping MIMO Controllers

Another interesting avenue worth investigating is the use of overlapping decentralised controllers. For the benchmark problem, it is possible to implement two 2-by-2 MIMO controllers in the overlapping configuration of [B A] and [A C] as depicted Figure 8-2. The survey in section 2.3.1.4 came across the method of expanding the state vector by some suitable transformation, which in this case effectively duplicate the axis A to an axis A′. Controllers can then be designed for the subsystems [B A] and [A′ C]. Inverse transformations are then applied to collapse the expanded back to the original state vector.

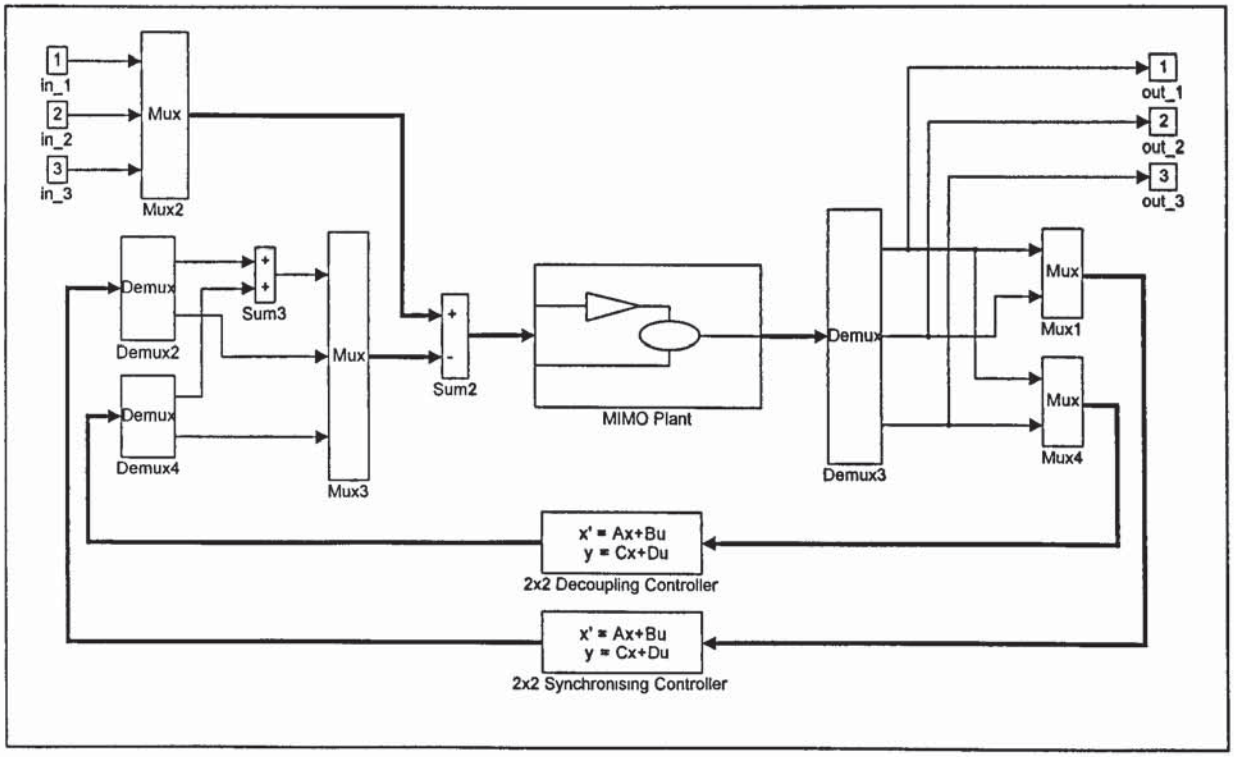

Figure 8-2 Overlapping Decentralised MIMO Controllers

8.9 Comment

Increasingly, in the design of manufacturing machinery, mechanical parts have been replaced by electrical and electronic parts. This can be seen as an extension of mechatronics which has now been a norm in product designs. A modular approach to design, in software, electronics as well as electro-mechanical parts is highly desirable since it improves flexibility and reusability. Not only does the design adapt better to changing production demands, it also allows shorter development cycle. This is a distinct advantage as the concurrent engineers are trying to reduce the product cycle.

With all these issues in sight, this thesis attempts to lay a foundation for an integrated methodology for the servo system design of a class of machinery. The methodology provides a formal approach to the motion control problem which has been handled in an ad hoc manner by the industry. Moreover, improvements in the quality of motion control by means of decentralised MIMO control system is another advantage.

As in many engineering design problems, the essence of this methodology is to provide a framework in which a balanced solution can be obtained for the conflicting objectives. In terms of the performance specification, the requirement of simultaneous synchronisation and decoupling of different groups of axes is potentially conflicting. In terms of the design cycle, it is a question of balancing computational complexity against analytical control methods. It is also a question of balancing the knowledge needed to be stored in the methodology against the user's experience.

The idea of implementing decentralised controllers based on a topology derived from a centralised controller based on two important principles. Firstly, closed loop design based on closed loop properties [Anderson 93] should produce a better solution. Secondly, in a closed loop system, the duality between the plant and the controller shift the required decentralisation from the plant to the controller so that it is possible to reconcile the inherent couplings and the imposed (non-) interactions.

The primary concern of the machinery under considerations are usually position errors (relative and absolute) which are prescribed in time domain. It is for this reason that the performance indices used in the investigations were built up mainly from step responses in time domain. Considerable effort has been put into the tailoring of suitable performance indices since they have to be constructed in such a way that the required qualities of the controlled system can be reflected quantitatively. Hence, although a workable performance

index is given in section 6.3.3, it may not be universally adaptable since it was tailored to the benchmark specifications. The standard LQ optimisation, which can be proved by dynamic programming in time, guarantees the solution to be globally optimal. However, numerically methods can not guarantee but local optimality and hence performance has to be evaluated in that light.

With ever advancing technology, performance of machinery are improving constantly and consistently in terms of speed, accuracy, flexibility, integrity, fault tolerance as well as cost and safety. The generic machine approach shifts the emphasis of the design from hardware towards software since better performance is achieved by replacing complex mechanisms with data structures and algorithms.

9. Appendix - Test Rig and Experiments

9.1 Single Servo Axis

The first group of experiments were aimed at familiarising with the control system. The description of both the hardware and software available is the same as that given in section 1.4. Specifically, the hardware consists of

- BRU200 motor and drive pair
- Themis 440 motion controller card
- VME rack with host computer running OS9 operating system

and were arranged in the same way as depicted in Figure 1-9. Once the separate components had been assembled and the software installed properly, a series of simple tasks were set to test and explore the various properties and capabilities of the servo system. A number of issues were investigated

- PID control of an independent single axis
- Use of different motion profiles
- Use of different mechanical couplings between the motor and a simple inertia load
- Use of dynamometer to emulate different load characteristics
- Implementation of data logging facility within the control software

The BRU200 can be set in Torque Mode or Velocity Mode. Using the servo system in Torque Mode leaves the inner current regulator and position loops functioning but disables the velocity control loop. The TSVME440 has the ability to independently implement control algorithms as well as to receive higher level commands from an external processor, via the VME bus, which can be used to change the control algorithms in an on-line situation.

9.2 Two Servo Axes

Two identical servo axes were set up side by side each coupled to a dynamometer of different inertia. There were no physical interaction between these two axes initially. A number of experiments were carried out to explore possible interactions between the axes by means of software. These includes:

- Implementation of a master-follower strategy for synchronisation - the actual position of the master is taken as the demand position of the follower
- Implementation of simple error cross feeding for synchronisation - fraction of relative error between the axes were added to the motion profiles in real time
- Implementation of a general 4 terms MIMO control routine (see section 9.5)
- Emulation of inherent coupling with two dynamometers (see section 9.2.1)

9.2.1 Emulation of Inherent Couplings

One of the most important aspect of the design methodology is the treatment of inherent coupling in a MIMO load. Since there were two dynamometers (of different sizes) available in the laboratory, one way to proceed is to couple them electronically so that they can emulate a MIMO load with the possibility of adjusting the level of interaction at an instant. The hardware were set up as depicted in Figure 9-1. In the simplest case, DC signals were fed to the dynamometers. Added to them were the cross channel velocity feedback which were provided by the tachometers built in to the dynamometers. All the signals were scaled, calibrated and summed with the use of simple electronics. This arrangement provides an opportunity to manipulate the signals in such a way that different kinds of coupling can be emulated if needed. The measurements resulted however, continuously suffer from the temperature sensitivity of the op-amps.

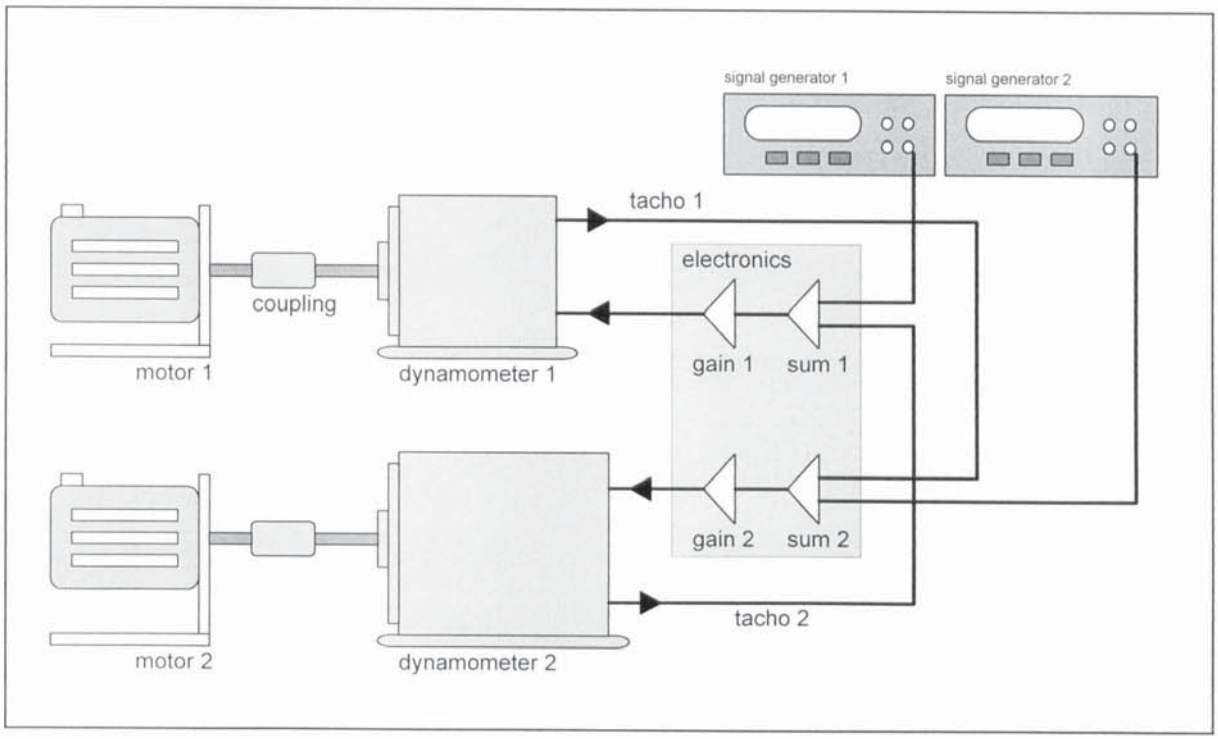

Figure 9-1 Emulation of Inherent Coupling

9.3 Construction of the MIMO Test Rig

To further the investigation, a MIMO mechanical load as part of a test rig were designed and commissioned. A schematic of this test rig is shown in Figure 9-2. The main features of this test rig include the followings:

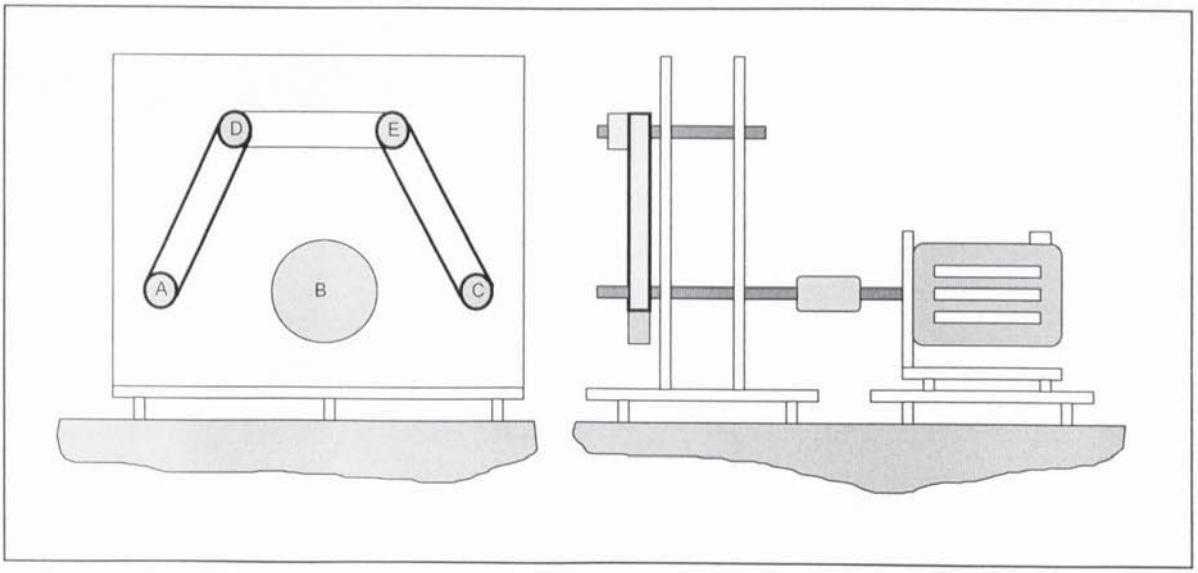

Figure 9-2 MIMO Test Rig

- Five identical steel shafts each supported by a pair of bearings are mounted through two backplates which are themselves mounted rigidly on a platform. The platform is supported by four rubber blocks so that it is mechanically isolated from the bench. (see Figure 9-3).
- Up to three toothed pulleys of various sizes can be attached to the end of each of the shafts easily. Steel wire reinforced toothed belts can be used to provide stiff linkage between any two shafts.
- Rubber bands of various stiffness can be put on any two shafts to represent web couplings.
- The pulleys by themselves are heavy enough to be treated as significant inertia loads.
- After some geometrical considerations, the five shafts are arranged in a way that a maximum number of linkages can be configured with a simple change of pulleys and belts and a minimal adjustment of tension rollers.
- A second platform is mounted on four rubber blocks so that it is mechanically isolated from the bench. Three identical motors are mounted on this flexible platform. Each motor together with its bracket is elevated from the platform by four pointed supports so that the platform retains its uniform stiffness and is free to vibrate. This provides a mechanism for structual coupling. (see Figure 9-4).
- Specailly designed aluminium couplings with high torsional stiffness, axial flexibility and low inertia are used to join the motor shatfs to the load shafts.
- Transparent perpex screens are used to cover moving parts for safety

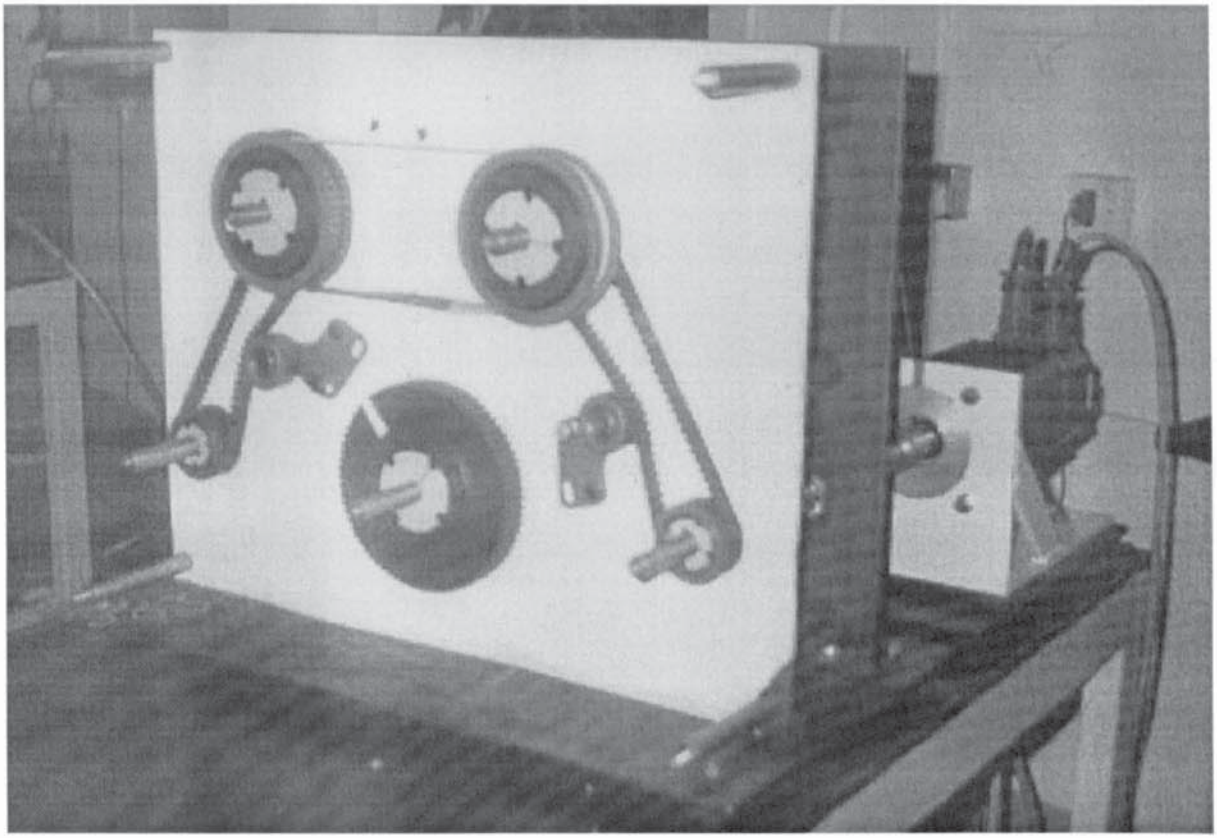

Figure 9-3 5 Pulleys on a Backplate - Rubber Band around the Top Two Pulleys

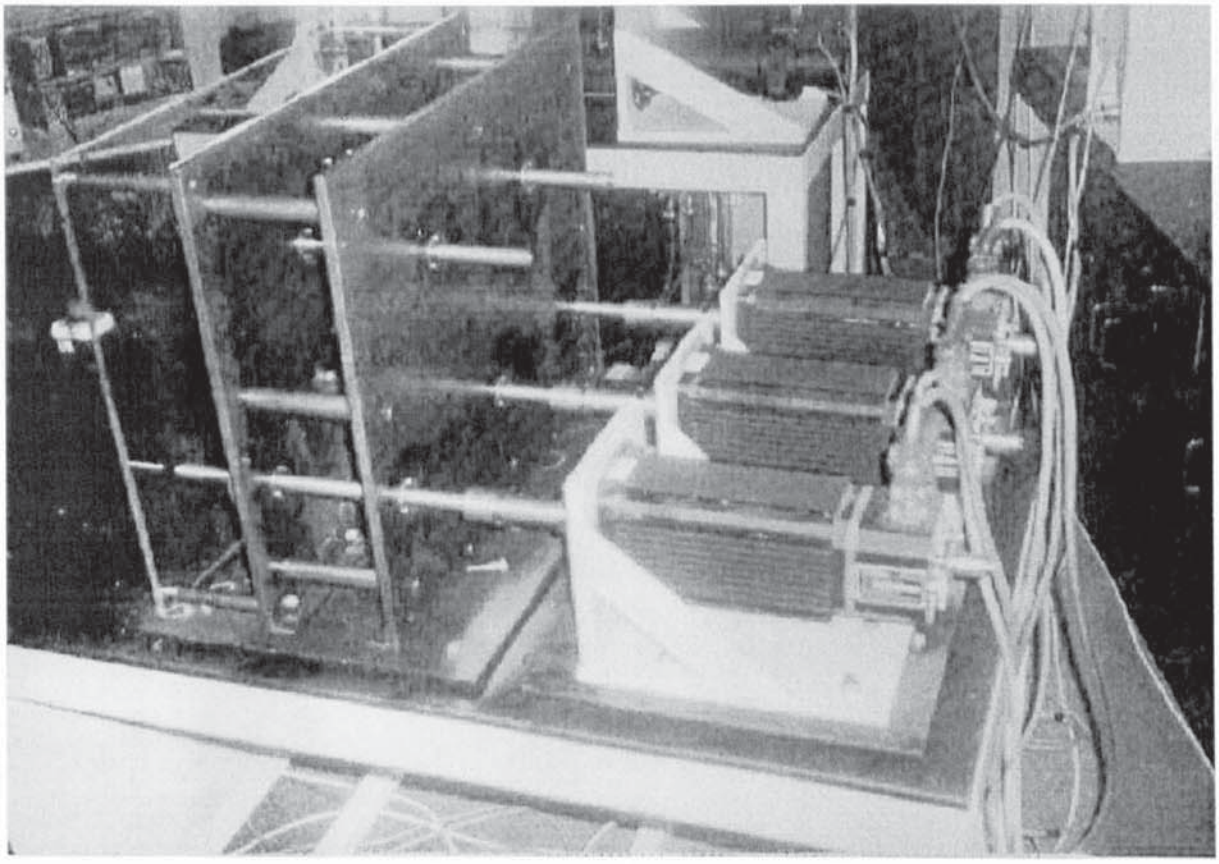

Figure 9-4 3 Motors on a Flexible Platform on the Right Hand Side

9.3.1 The MIMO Test Rig and the Benchmark Problem

The test rig and the benchmark problem are constructed to reflected each other. Referring the Figure 9-2, the three lower shafts are labelled A, B & C and are driven directly by the three motors. The shafts A & D and C & E are coupled by stiff toothed belts and so they are considered rigidly linked. Shafts D & E are linked by a rubber band which is elastic and thus represents a web coupling between axes A and C. Axis B stands independently but its has a heavier inertia load.

The benchmark problem can be summarised as the following. Axes A and C are required to be decoupled whereas simultaneously axes A and B are required to be synchronised. The decoupling represents a treatment of inherent couplings and the synchronisation represents a treatment of imposed interactions. Further, assuming that it is only feasible to implement MIMO controllers of size up to 2 inputs 2 outputs, the application of blockwise decentralised control can be illustrated.

9.4 System Identification

This section summarises the attempt of taking transfer function measurements on the completed MIMO test rig. As part of the benchmarking exercise, frequency responses of the MIMO rig were measured using a Fast Fourier Transform (FFT) analyser. The results can be incorporated into the generic machine model by replacing the corresponding components with look-up tables. Alternatively, transfer functions in the form of polynomials quotients

$$G(s) = \frac{b(s)}{a(s)} = \frac{b_1 s^{n-1} + b_2 s^{n-2} + \ldots + b_{n-1} s + b_n}{a_0 s^n + a_1 s^{n-1} + \ldots + a_{n-1} s + a_n} \qquad (9\text{-}1)$$

can be fitted to the data and thus provides an analytical plant description. These in turn can be compared with the initial generic machine model for refinement. In some cases, it is also more convenient to deal with these mathematical models in the synthesis of centralised MIMO controllers.

The measurement of frequency responses of the MIMO system involves the injection of a suitable excitation signal at an appropriate operating point. A DC signal of 3V was fed directly into all three BRU200 drives, as depicted in Figure 9-5, so that all shafts were

turning at the same constant speed and thus keeping the system at a suitable operating point. In doing so, non-linear effects can be minimised and backlash at origin can be avoided. The frequency response were measured using a set of data aquirsition (DP) cards installed in a PC. The DP card has 7 input channels and one output channel. A fast sweep sine signal of 0.3V (max.) generated by the output channel was fed into each of the drives in turns and all three outputs of the drives were logged simultaneously. The DP has on board FFT function and frequency responses can be obtained directly.

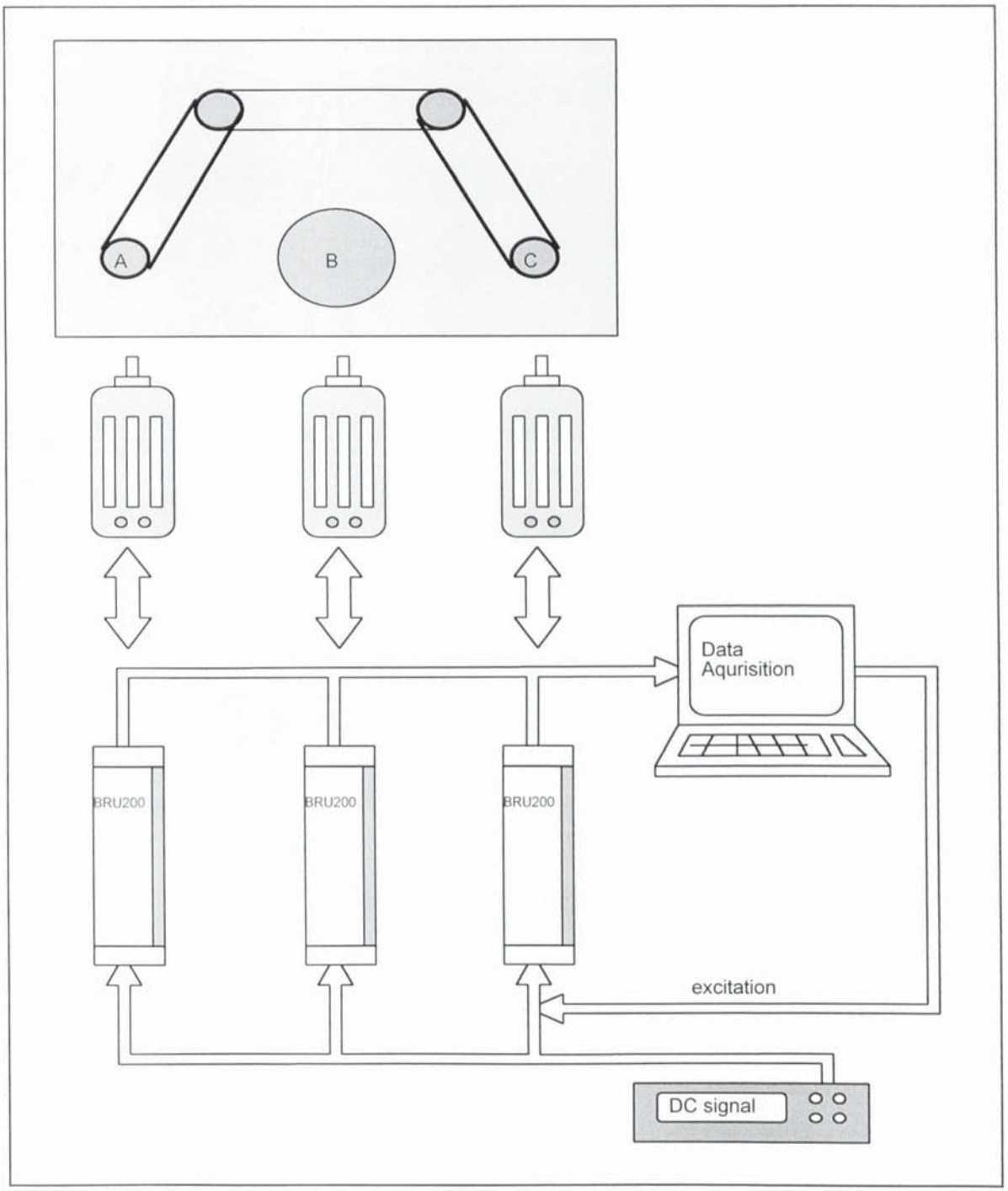

Figure 9-5 Frequency Responses of MIMO Test Rig

9.5 Implementation of the MIMO Control Algorithm

9.5.1 The MIMO Control Algorithm

Controllers implemented in the forward path has the advantage of dealing only with the error signal and thus avoid the usually much higher numerical values of demand and output signals. With suitable discretisation and order reduction, a 4x4 MIMO controller obtained off line can be put into the following recursive form

$$\begin{aligned} u_n(t) = \; & a_{n1}e_1(t) + b_{n1}e_1(t-1) + c_{n1}e_1(t-2) + d_{n1}u_1(t-1) \\ & + a_{n2}e_2(t) + b_{n2}e_2(t-1) + c_{n2}e_2(t-2) + d_{n2}u_2(t-1) \\ & + a_{n3}e_3(t) + b_{n3}e_3(t-1) + c_{n3}e_3(t-2) + d_{n3}u_3(t-1) \\ & + a_{n4}e_4(t) + b_{n4}e_4(t-1) + c_{n4}e_4(t-2) + d_{n4}u_4(t-1) \end{aligned} \qquad (9\text{-}2)$$

where n = {1, 2, 3, 4}, the e's and the u's are the errors and the control signals respectively. There are 64 coefficients in total. By suitably setting some of the coefficients to zero, one call of the algorithm can also be used to calculate the control signals for four 1x1 controllers or two 2x2 MIMO controllers. It may however be more efficient if specific routines were written specifically for the smaller MIMO controllers.

As the number of coefficients involved is large, it becomes very time consuming to fine tune any controller if it is necessary to recompile and download the control program each time a parameter is changed. It is therefore desirable to be able to change these coefficients interactively while the control routine is running (on-line).

In general, each channel will follow a different motion profile in a cyclic fashion. It may be necessary to specify separate profiles for system start-up and shut-down transients, diagnostic as well as maintenance purposes to ensure smooth running of the plant. Reference, output together with the control signals are all logged for off line diagnosis and reports. This however involve heavy data traffic and are limited by the length of the time slice as well as the duration of the motion profile cycle.

9.5.2 Implementation of the MIMO Controller on Themis 440

The generic code generator provided by industry takes the system definition files and link with application specific run time routines to generate the control program which run on

the host computer. These definition files specify the addresses of control and data buses for the sensors and actuators as well as all the system specific data, such as sampling period. Also generated is the target code for each of the Themis controller card.

Once the program is started, it wait for a sequence of interactive commands which initialise the servo hardware, download the target code together with the specified motion profiles to the corresponding Themis card. On issuing the start command, the Themis card is set into motion. It services each of the four channels in sequence during the duration of each time slice. The target code is proprietary and each channel has its own data structure which are defined in a way that it is not possible to parse data between the channels on board, despise considerable effort has been made in writing additional and modifying existing code in assembler. It is however possible to parse data first to the host computer. In trying to implement a MIMO controller as a run time routine, it is found that the set of hardware can not cope with the data traffic.

9.5.3 Implementation of the MIMO Controller on PC/C31

After some considerable difficulties in trying to implement a MIMO control algorithm on the Themis card, it is decided that an alternative solution must be found. By chance, a Laoughbourgh Sound & Images PC/C31card was made available. Unlike the Themis controller card, a general purpose DSP board has no architectural restriction between the I/O channels. This card also offer the advantage that code can be written in a higher level language, C. The card is based around the Texas TMS320C31 32 bit floating point DSP and is required to be hosted by a PC. Some of its features are summarised in Table 9-1. In addition, 4 analogue inputs and 4 analogue outputs are provided by 2 daughter modules on which the DAC's deliver +/- 3V output signals. These can be connected directly to the drives input ports which take +/- 10V full scale.

Table 9-1 Summary of the PC/C31 Board

DSP	TMS320C31 - 32 bit floating point
Memory Bank 0	32K zero wait state SRAM
Memory Bank 1	128K zero or one wait state SRAM or two wait states EPROM
Dual-Port RAM	Two wait states DPRAM allows fast information exchange - 2K x 32 on DSP side and 8K x 8 on PC side
Data Bus	32 bit
Address Bus	24 bit
Clock	33.3 M Hz
DSPLINK2	32 bit memory-mapped parallel expansion

9.5.3.1 Software

The control program is written in C in a development environment on the PC. The software is cross complied on the PC and then downloaded onto the DSP card. A skeleton program which initialises the card and partly establishes the communication between the PC and the DSP board is made available to the author.

Further work has been carried out to set up the I/O channels. Simultaneously, routines were developed so that data can be logged for later diagnosis. The idea of gathering the data from the DSP and sent to the PC for each time slice was found to be unrealistic because writing data to the PC is too slow. The next simplest idea is to log and hold as much data on the DSP card as the size of the memory allowed and upload the data to the PC at the end of the program. Based on a sampling time of 1ms, the general purpose MIMO control algorithm was coded. In order to prove the software, attention was focused on closing one loop first.

9.5.3.2 Decoder Board

The main drawback of this DSP card is the absence of digital inputs. In order to close the feedback loops, additional electronics are required. A decoder board is built for this purpose. This decoder board made use of the HCTL-2020 quadrature decoder/counter IC.

It takes the encoder signals from the motor (through the drive), filters, decodes and outputs a 16 bits position count. This number is then mapped directly onto the DSP via the DSPLINK2 expansion interface. However there are no external control line available to interrupt the slaved decoder board. Instead the position counts have to be buffered and wait until a watchdog circuit trigger an update after the data are fetched by the DSP. Another difficulty is to deal with the roll-over of the counters. The noisy signals complicate the matter further.

9.5.3.3 Proposed Experiments

Once all the hardware are in place and a general purpose MIMO control algorithm together with the supporting software are available, a flexible real time platform will be available to test the methodology. A number of experimental objectives based on the benchmark problem were purposed which reflect the development made in the simulation work.

- synchronisation of axes A and B
- decoupling control of axes A and C
- full 3x3 MIMO control of the test rig
- decentralised control of the test rig under the [A B] & [C] topology
- decentralised control of the test rig under the [A C] & [B] topology
- control of the test rig with 3 SISO loops

These objectives were set so that a study can be carried out to see how a blockwise decentralised MIMO control system performs compare with other alternatives. Particularly important is the opportunity to find out what the limitations of the methodology are in practice.

10. Appendix - Numerical Work

10.1 Numerical Specification for the Benchmark Problem

Given a state description of a linear Multi-Input Multi-Output plant

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$$
$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}$$

(10.1)

where x is the n x 1 state vector, u is the m-input vector and y is the m-output vector.

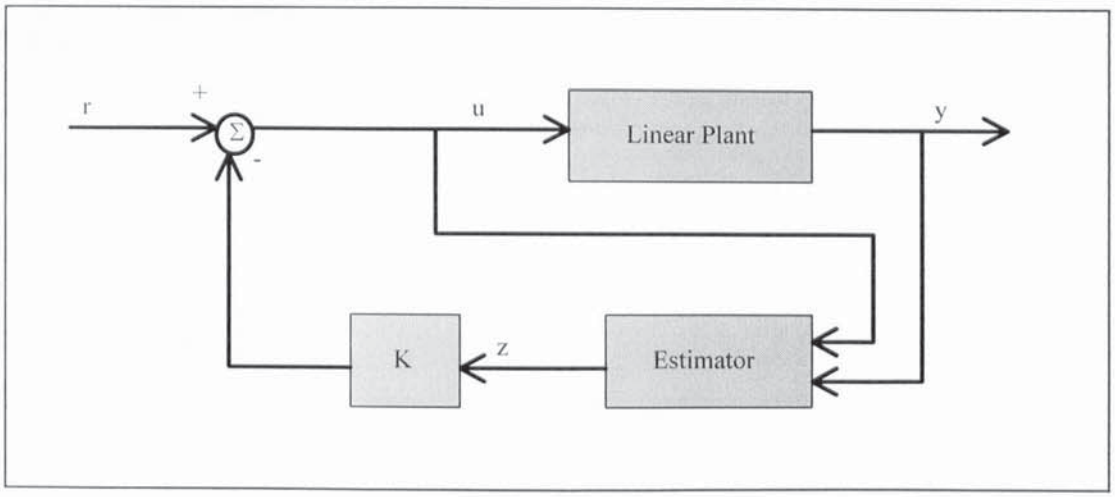

Figure 10-1 LQG Control Feedback Structure

The problem is to find a state estimator with gain $\mathbf{K}_e$ for the estimated state z

$$\dot{\mathbf{z}} = (\mathbf{A} + \mathbf{K}_e\mathbf{C})\mathbf{z} + \mathbf{B}\mathbf{u} - \mathbf{K}_e\mathbf{y}$$

(10.2)

and a state feedback control law with gain K

$$\mathbf{u} = \mathbf{r} - \mathbf{K}\mathbf{z}$$

(10.3)

such that the following constraints are satisfied:

Input Motion Profiles

For each of the m inputs r (where r_i, i=1...m) following a reference input profile,

$$|\dot{r}_i| < \beta_i$$
$$|\ddot{r}_i| < \alpha_i$$

(10.4)

Control Efforts

For each of the m control signals u (where u_i, i=1...m),

$$|u_i| < u_{\max i}$$

(10.5)

and the following is minimised (note that it is minimised over all channels);

$$\mathbf{J} = \int |\mathbf{u}(\mathbf{t})|^2 \, dt$$

(10.6)

Output Constraints (Performance Specification)

For the output

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}$$

(10.7)

Tracking

Each output follows the reference input to within a band of tolerance

$$|y_i(t) - r(t)_i| < \eta_i(t)$$

(10.8)

where η is the tracking tolerance and is time dependent in general.

Synchronisation

In general, synchronisation is specified by prescribing bounds ε to some linear combination L of the outputs y such that:

$$\mathbf{L}\mathbf{y}(t) < \varepsilon(t) \tag{10.9}$$

One example case is for i≠j,

$$\left|y_i(t) - y_j(t)\right| < \varepsilon_k(t) \tag{10.10}$$

Decoupling

Assuming there is no synchronisation requirement. For a m x m input matrix R specified by

$$\mathbf{R}_{ij}(t) = \begin{cases} 0, t < T \\ 1, t \geq T \end{cases}, \forall i = j$$
$$\mathbf{R}_{ij}(t) = 0, \forall i \neq j \tag{10.11}$$

the output matrix Y is bounded by |Y|<Δ where

$$\Delta_{ij} = \begin{cases} \infty, \forall i = j \\ \delta_{ij}, \forall i \neq j \end{cases} \tag{10.12}$$

and in general

$$\delta_i << 1 \tag{10.13}$$

When both synchronisation and decoupling are required, priority is given to synchronisation and in general the set of indices i,j when i ≠ j is reduced to a disconnected subset of (*1...m, 1...m*).

10.1.1 Example

Consider a 3 axes system consists on 3 identical motors driving 3 identical inertia loads. The first axis is running independently while the second and the third are interacting via a material web. The *state space model* is given as :

```
A= 1.0e+004 *
         0         0         0         0         0    0.2083
         0         0         0         0    0.2083         0
         0         0         0    0.2083         0         0
         0         0         0   -3.3552         0         0
    0.1077   -0.1077         0         0   -3.3552         0
   -0.1077    0.1077         0         0         0   -3.3552

B=       0         0         0
         0         0         0
         0         0         0
    1.2375         0         0
         0    1.2375         0
         0         0    1.2375

C=       0    0    1    0    0    0
         0    1    0    0    0    0
         1    0    0    0    0    0

D=       0    0    0
         0    0    0
         0    0    0
```

The *input profiles* are given as:

$$r_1 = r_2 = r_3 = r$$

and

$$\alpha = 20000 rads^{-2}$$
$$\beta = 100 rads^{-1}$$

And the *control effort* is unbounded; i.e. $u_{max} = \infty$.

The *tracking tolerance* is $\eta_1 = \eta_2 = \eta_3 = \eta = 0.005\pi$ rad.

Synchronisation is required for only the first 2 axes and is specified by:

$$|y_2 - y_1| < \varepsilon = 0.005\pi$$

and the level of *decoupling* between the last 2 axes are specified by the bound Δ on the responses to the step input matrix R:

$$\Delta = \begin{bmatrix} \infty & \infty & \delta \\ \infty & \infty & \delta \\ \delta & \delta & \infty \end{bmatrix}$$
$$\delta = 0.005$$

(10.14)

The problem is to search for a valid set of K and K_e such that the above criteria are satisfied.

10.2 The Amoeba Algorithm

Listing 10-1 ame.m

```
clc;echo on;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%%    Search for centralised controller:
%%    by amoeba (for sync)
%%
%%    begin: 8 May 96
%%
%%    By A.Y.S.Chan
%%
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% environment
echo on;clear;
format long E;format compact;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% init
mdat;
global w; w=logspace(-2,4,50);
global gr; gr='tm3';
global qg; qg=1;              %%%%%%%%%%%%%%%%%%%%%%%%%% adjust decouple
%global t; tf=0.04;nnn=50; t=0.00:tf/nnn:tf;
global t; tf=0.1;nnn=100; t=0.00:tf/nnn:tf;
[zzt,zt]=size(t);
global z9;z9=zeros(1,9);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 1st solution for the non interacting plant
kc=firstkc(gr,0);
%fstkc=kc;
[zz,z]=size(kc);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% size of Steady State
global yss;
yss=ysscal(gr,0,fullkw,0,z9,z9,kc,t,w,0);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% form initial specification vector
per=pqk2(yss,gr,0,fullkw,0,z9,z9,kc,t,w,0);
%per=pqk2(yss,gr,0,fullkw,0,z9,z9,kc,t,w,1);
kc=firstkc(gr,1);             %%%%%%%%%%%%%%%%%%%%%%%%%% adjust sync
fstkc=kc;
spec=per;
```

```
intspec=per;

%%%%%%%%%%%%%%%%%%%% form final specification vector
%tor=0.00000001;
tor=0.000001;tor=tor*tor*zt;
fnlspec=spec*1.5;fnlspec(3)=tor;fnlspec(7)=tor;

%%%%%%%%%%%%%%%%%%%% initial simplex
kctmp=kc;
%kctmp(1,:)=kc(1,:)+0.5*kc(3,:);
%kctmp(3,:)=0.5*kc(1,:)+kc(3,:);
%kctmp(3,:)=kctmp(1,:);

kcvv=[kctmp(1,:) kctmp(2,:) kctmp(3,:)];
%kcvv=3000*full(spones(kcvv));

for ii=1:27
%       ii
        kcvvtmp=kcvv;
%kcvvtmp(ii)=kcvv(ii)+abs(sqrt(kcvv(ii)))+1000;
        kcvvtmp(ii)=kcvv(ii)-10;
        kcvvm(ii,:)=kcvvtmp;
        kki=[kcvvtmp(1:9);kcvvtmp(10:18);kcvvtmp(19:27)];
        yyy(ii)=pcost(yss,gr,qg,fullkw,0,z9,z9,kki,t,w,0);
end;
kcvvm(28,:)=kcvv;
kki=kc;
yyy(28)=pcost(yss,gr,qg,fullkw,0,z9,z9,kki,t,w,0);
yyy

%%%%%%%%%%%%%%%%%%%% initialise loop
%ftol=sqrt(eps);
ftol=10^(-2.0);
lmax=27*27*10;
alpha=1;
beta=0.5;
gamma=2;
ndim=27;
psum=sum(kcvvm);
lcount=0;plcount=0;
t0=clock;

%%%%%%%%%%%%%%%%%%%% main loop
while 1>0,          %%%%%%%%%%%%%%%%%%%%%%%%
 %lcount
[Y,I]=sort(yyy);
ihi=I(28);
inhi=I(27);
ilo=I(1);
rtol=2*abs(yyy(ihi)-yyy(ilo))/(abs(yyy(ihi))+abs(yyy(ilo)));
rtolog=[rtolog log10(rtol)];
ppcc=[ppcc log10(min(yyy))];
 if lcount>=plcount
        lcount
        %time_elasped=etime(clock,t0)
        plcount=plcount+100;
        figure(1);
        bar(kcvv,'y');hold on;
        bar(kcvvm(ihi,:),'r');
        bar(kcvvm(ilo,:),'g');hold off;title('range of kc');grid;
        figure(2);plot(ppcc);grid;title('trend of log-cost');
        figure(3);plot(rtolog);grid;title('log of rtol');
```

```
        kc=[kcvvm(ilo,1:9);kcvvm(ilo,10:18);kcvvm(ilo,19:27)];
        per=pqk2(yss,gr,1,fullkw,0,z9,z9,kc,t,w,1);
        drawnow;
end;

if rtol < ftol
        disp('below tolerence bound');
        save now;
        figure(1);
        bar(kcvv,'y');hold on;
        bar(kcvvm(ihi,:),'r');
        bar(kcvvm(ilo,:),'g');hold off;title('range of kc');grid;
        figure(2);plot(ppcc);grid;title('trend of log-cost');
        figure(3);plot(rtolog);grid;title('log of rtol');
        kc=[kcvvm(ilo,1:9);kcvvm(ilo,10:18);kcvvm(ilo,19:27)];
        per=pqk2(yss,gr,1,fullkw,0,z9,z9,kc,t,w,1);
        drawnow;
        %diary log.txt;
        clock;
        %kcvvm(ihi,:)
        break;
end;
if lcount >=30000
        disp('too many iterations');
        save now;
        figure(1);
        bar(kcvv,'y');hold on;
        bar(kcvvm(ihi,:),'r');
        bar(kcvvm(ilo,:),'g');hold off;title('range of kc');grid;
        figure(2);plot(ppcc);grid;title('trend of log-cost');
        figure(3);plot(rtolog);grid;title('log of rtol');
        kc=[kcvvm(ilo,1:9);kcvvm(ilo,10:18);kcvvm(ilo,19:27)];
        per=pqk2(yss,gr,1,fullkw,0,z9,z9,kc,t,w,1);
        drawnow;
        clock;
        %kcvvm(ihi,:)
        break;
end;

[ytry,lcount,kcvvm,yyy,psum]=ametry(kcvvm,yyy,psum,ndim,ihi,lcount,-alpha);
%disp('reflect');
if ytry<=yyy(ilo)
        [ytry,lcount,kcvvm,yyy,psum]=ametry(kcvvm,yyy,psum,ndim,ihi,lcount,gamma);
        %disp('extend');
elseif ytry>=yyy(inhi)
        ysave=yyy(ihi);
        [ytry,lcount,kcvvm,yyy,psum]=ametry(kcvvm,yyy,psum,ndim,ihi,lcount,beta);
        %disp('contract');
        if ytry >= ysave
                disp('problem of ytry greater than yhi');
                for ii=1:ndim+1
                        if ii~=ilo
                                psum=(kcvvm(ii,:)+kcvvm(ilo,:))/2;
                                kcvvm(ii,:)=psum;
                                kkpsum=[psum(1:9);psum(10:18);psum(19:27)];
                                yyy(ii)=pcost(yss,gr,qg,fullkw,0,z9,z9,kkpsum,t,w,0);
                        end;
                end;
                lcount=lcount+ndim;
                psum=sum(kcvvm);
        end;
end;
```

```
end;      %%%%%%%%%%%%%%%%%%%%% end of main loop

disp(' ');time_elasped=etime(clock,t0)
 %kc=[kcvvm(ilo,1:9);kcvvm(ilo,10:18);kcvvm(ilo,19:27)];
per=pqk2(yss,gr,1,fullkw,0,z9,z9,kc,t,w,3);

%%%%%%%%%%%%%%%%%%%%% end of script
```

11. Reference

[Anderson 93] : Anderson, B.D.O. Controller Design: Moving from Theory to Practice. *IEEE Control Systems Magazine*, **August** 1993, 16-25.

[Anderson & Moore 71] : Anderson, B.D.O. & Moore, J.B. Linear Optimal Control. Prentice-Hall.

[Arkun 87] : Arkun, Y. Dynamic Block Relative Gain and Its Connection with the Performance and Stability of Decentralized Control Structures. *International Journal of Control* **46**, 1187-1193, (1987).

[Arkun 88] : Arkun, Y. Relative Sensitivity: A Dynamic Closed-Loop Interaction Measure and Design Tool. *AIChE Journal* **34**, 672-675.

[Bennett 94] : Bennett, S. Real-time Computer Control - An Introduction. Prentice-Hall.

[Bennett & Baras 80] : "Block Diagonal Dominance and Design of Decentralized Compensators", *Large Scale Systems: Theory and Applications*, 93-101.

[Boyd, Barratt & Norman 90] : Linear Controller Design: Limits of Performace Via Convex Optimization. *Proceedings of the IEEE*, **78**, 529-74.

[Bradley, Dawson, Burd & Loader 93] : Bradley, D.A., Dawson, D., Burd, N.C. & Loader, A.J. Mechatronics. Chapman & Hall.

[Bristol 66] : Bristol, E.H. On a New Measure of Interaction for Multivariable Process Control. *IEEE Transactions on Automatic Control* **AC-11**, 133-134, (1966).

[Bristol 78] : "Recent Results on Interaction in Multivariable Process Control", 71st, AlChE. Conference, Miami.

[Brogan 91] : Brogan, W.L. Modern Control Theory. Prentice-Hall.

[Bryson 96] : Bryson, A.E. Optimal Control -- 1950 to 1985. *IEEE Control Systems* **16-3**, 26-33.

[de Carli & Caccia 95] : A Comparison of Some Control Strategies for Motion Control. *Mechatronics* **5-1**, 61-71.

[Chen, Mahmoud & Singh 84] : Chen,Y., Mahmoud, M.S. & Singh, M.G. An iterative block-diagonalization procedure for decentralized optimal control. *Internation Journal of Systems Science* **15**, 563-73.

[Commault & Dion 92] : Commault, C. & Dion, JM. A Comprehensive Introduction to the Geometric Theory of Linear Multivariable Systems. *IEEE Transactions on Education* **35**, 92-97.

[Danbury & Jenkinson 94] : Synchronised Servomechanisms - the Scalar-Field Approach. *IEE Proceedings* **141-D**, 261-273.

[Davison 76a] : Davison, E.J. The Robust Decentralized Control of a General Servomechanism Problem *IEEE Transactions on Automatic Control* **AC-21**, 14-24.

[Davison 76b] : Davison, E.J. The Robust Control of a Servomechanism Problem for Linear Time-Invariant Multivariable Systems *IEEE Transactions on Automatic Control* **AC-21**, 25-34.

[Davison 76c] : Davison, E.J. Multivariable Tuning Regilators: The Feedforward and Robust Control of a General Servomechanism Problem *IEEE Transactions on Automatic Control* **AC-21**, 35-47.

[Doyle & Stein 81] : Doyle, J.C. & Stein, G. Multivariable Feedback Design: Concepts for a Classical/Modern Synthesis *IEEE Transactions on Automatic Control* **AC-26**, 4-16.

[Edmunds 79] : Edmunds, J.M. Control System Design and Analysis using Closed-loop Nyquist and Bode Arrays. *International Journal of Control* **30**, 773-802, (1979).

[Electro-Craft 87]: BRU-200 Servo Drives Instruction Manual. Robbins & Myers/Electro-Craft.

[Falb & Wolovich 67] : Falb, P.L. & Wolovich, W.A. Decoupling in the Design and Synthesis of Multivariable Control Systems. *IEEE Transactions on Automatic Control* **AC-12**, 651-59.

[Feingold & Varga 62] : Feingold, D.G. & Varga, R.S. Block Diagonally Dominant Matrices and Generalizations of the Gershgorin Circle Theorem. *Pacific Journal of Mathematics* **12**, 1241-1250.

[Fiedler & Ptak 62] : Generalized Norms of Matrices and the Locations of the Spectrum. *Czech. Math. Journal* **87**, 558-570.

[Friswell, Garvey, Penny & Chan 96] : Friswell, M.I., Garvey, S.D., Penny, J.E.T. & Chan, A. "Selection of Control Topology for Decentralised Multi-Sensor Multi-Actuator Active Control." International Modal Analysis Conference., pp 278-284.

[Gilbert 69] : Gilbert, E.G. The Decoupling of Multivariable Systems by State Feedback. *SIAM Journal of Control* **7**, 50.

[Golub & Ortega 92]: Scientific Computing and Differential Equations. Academic Press.

[Grosdidier, Morari & Holt 85] : Closed-Loop Properties from Steady State Gain Information. *Industrial Engineering Chemistry Fundamental* **24**, 221-235.

[Gundes & Desoer 90] : Algebraic Theory of Linear Feedback Systems with Full and Decentralized Compensators : Springer-Verlag.

[Hanselmann 87] : Implementation of Digital Controllers - A Survey. *Automatica* **23**, 7-32.

[Hu, Chiu & Tomizuka 90]: "On Motion Synchronization of Two Axes Systems." ASME Symposium on Monitoring and Control for Manufacturing Processes.,:pp. 267-288.

[Jamshidi 83] : Large-Scale Systems : Modelling and Control : North-Holland.

[Johnson & Grimble 87] : Recent Trends in Linear Optimal Quadratic Multivariable Control System Design. *IEE Proceedings* **134-D**, 53-71.

[Kamyama & Furuta 76] : *IEEE Transactions on Automatic Control* **AC-20**, 413.

[Kwakernaak 93] : Robust Control and H^{∞} -Optimization - Tutorial Paper. *Automatica* **29**, 255-273.

[Lee & Park 92] : Lee, M. and Park, S. Relative Sensitivity Matrices for Closed-Loop Interaction Analysis and Control Structure Synthesis. *Journal of Chemical Engineering of Japan* **25**, 187-195, (1992).

[Lieslehto, Tanttu & Koivo 93] : An Expert System for Multivariable Controller Design. *Automatica* **29**, 953-968.

[Linnemann & Wang 93] : Block Decoupling with Stability by Unity Output Feedback - Solution and Performance Limitations. *Automatica* **29**, 735-744.

[MacFarlane, Gruebel & Ackermann 89] : Future Design Environment for Control Engineering. *Automatica* **25**, 165-176.

[Maciejowski 89] : Multivariable Feedback Design : Addison-Wesley.

[Manousiouthakis & Nikolaou 89] : Analysis of Decentralized Control Structures for Nonlinear Systems. *AIChE Journal* **35**, 549-558.

[Manousiouthakis, Savage & Arkun 86] : Manousiouthakis, V., Savage, R. & Arkun, Y. Synthesis of Decentralized Process Control Structures Using the Concept of Block Relative Gain. *AIChE Journal* **32**, 991-1003, (1986).

[Morgan 64] : *IEEE Transactions on Automatic Control* **AC-9**, 405.

[Nader 79] : "The design of Distributed Computer Control Systems", Proceedings of the IFAC Workshop on Distributed Computer Control Systems, : pp. 157-179.

[Ng 89] : Interactive Multi-Objective Programming as a Framework for Computer-Aided Control System Design:: Springer-Verlag.

[Ng 93] : Perspectives on Search-Based Computer-Aided Control System Design. *IEEE Control System Magazine*, **April** 1993, 65-72.

[Ohta, Siljak & Matsumoto 86] : Ohta, Y., Siljak, D.D. & Matsumoto, T. Decentralized Control Using Quasi-Block Diagonal Dominance of Transfer Function Matrices. *IEEE Transactions on Automatic Control* **AC-31**, 420-430.

[Ozguner 89]: "Problems in Implementing Distributed Control." Proceedings of the Amercian Control Conference.,:pp. 274-279.

[Patel & Misra 91] : Numerical Computation of Decentralized Fixed Modes. *Automatica* **27**, 375-382.

[Pilly 91] : Application Characteristics of Permanent Magnet Synchronous and Burshless dc Motors for Servo Drives. *IEEE Transactions on Industrial Applications* **27**, 405-407.

[Press, Flannery, Teukolsky & Vetterling 92]: Numercial Recipes in C - The Art of Scientific Computing. Cambridge University Press.

[Reeves & Arkun 89] : Interaction Measures for Nonsquare Decentralized Control Structures. *AIChE Journal* **35**, 603-613.

[Reinschke 88] : Multivariable Control A Graph-theoretic Approach : Springer-Verlag.

[Sandell 76] : "Decomposition vs. Decentralisation in Large Scale System Theory", Proceedings of IEEE Conference on Decision and Control, Cleanwater FL : IEEE, pp.1043-1046.

[Sandell, Varaiya, Athans & Safonov 78] : Survey of Decentralized Control Methods for Large Scale Systems. *IEEE Transactions on Automatic Control* **AC-23**, 108-128.

[Shiller, Chang & Wong 96] : The Practical Implementation of Time-Optimal Control for Robotic Manipulators. *Robotics & Computer-Integrated Manufacturing* **12-1**, 29-39.

[Siljak 91] : Siljak, D.D. Decentralized Control of Complex Systems : Academic Press.

[Siljak 96] : Siljak, D.D. Decentralized Control and Computations: Status and Prospects. *Annual Reviews in Control* **20**, 131-141.

[Syrmos, Abdallah, Dorato & Grigoriadis 97] : Syrmos, V.L., Abdallah, C.T., Dorato, P. & Grigoriadis, K. Static Output Feedback - A Survey. *Automatica* **33**, 125-137.

[Tokuz & Rees Jones 91] : Tokuz, L.C. & Rees Jones, J. Programmable Modulation of Motion Using Hybrid Machines. *IMechE* **C414/071**, 85-91.

[Trave, Titli & Tarras 89] : Large Scale Systems : Decentralization, Structural Constraints and Fixed Modes : Springer-Verlag.

[Vaz & Davison 89] : The Structured Robust Decentralized Servomechanism Problem for Interconnected Systems. *Automatica* **25**, 267-272.

[Willems 81] : Almost Invariant Subspaces : An Approach to High Gain Feedback Design - Part I: Almost Controlled Invariant Subspaces. *IEEE Transactions on Automatic Control* **AC-26**, 235-252.

[Wonham 79] : Wonham, W.M. Linear Multivariable Control : a Geometric Approach., (2nd edition) : Springer-Verlag.

[Wonham & Morse 70] : Wonham, W.M. & Morse, A.S. Decoupling and Pole Assignment in Linear Multivariable Systems : A Geometric Approach. *SIAM Journal of Control* **8**, 1-18.

[Xia, Rao, Sun &Ying 93] : New Technique for Decoupling Control. I*nternational Journal of System Scinece* **24**, 289-300.

[Xu & Mansour 88] : Decentralized Robust Output Regulation Using H^{∞}-Optimization Techniques. *Control-Theory and Advanced Technology* **6**, 187-214.

[Xu & Mansour 90] : "An Application of H^{∞} Theory in Decentralized Control", Proceedings of the 27th Conference on Decision and Control, Austin, Texas, 1335-1340.

[Young & Reid 93] : Lateral and Longitudinal Dynamic Behavior and Control of Moving Webs. *ASME Journal of Dynamic Systems, Measurement, and Control* **115**, 309-317.