High Redundancy Actuator

by

(Xinli Du)

A Doctoral Thesis

Submitted in partial fulfilment of the requirements

for the award of

PhD of Loughborough University

August. 2008

ABSTRACT

High Redundancy Actuation (HRA) is a novel type of fault tolerant actuator. By comprising a relatively large number of actuation elements, faults in the elements can be inherently accommodated without resulting in a failure of the complete actuation system. By removing the possibility of faults detection and reconfiguration, HRA can provide high reliability and availability. The idea is motivated by the composition of human musculature. Our musculature can sustain damage and still function, sometimes with reduced performance, and even complete loss of a muscle group can be accommodated through kinematics redundancy, e.g. the use of just one leg.

Electro-mechanical actuation is used as single element inside HRA. This thesis is started with modelling and simulation of individual actuation element and two basic structures to connect elements, in series and in parallel. A relatively simple HRA is then modelled which engages a two-by-two series-in-parallel configuration. Based on this HRA, position feedback controllers are designed using both classical and optimal algorithms under two control structures. All controllers are tested under both healthy and faults injected situations. Finally, a hardware demonstrator is set up based simulation studies. The demonstrator is controlled in real time using an xPC Target system. Experimental results show that the HRA can continuously work when one element fails, although performance degradation can be expected.

Key words: high redundancy, fault tolerance, electro-mechanical actuator, classical control, LQG control, model reduction

List of Symbols

C_m	motor damping constant	Nm/rad/s
C_s	screw damping constant	Nm/rad/s
e	motor electromotive force (e.m.f)	N
e_p	pulsating (or transformer) e.m.f.	N
e_r	motional e.m.f.	N
i_a	motor current	A
J_m	motor mechanical inertia	Kgm^2
K_e	motor voltage constant	V/rad/s
K_m	motor series stiffness constant	N/m
K_s	screw stiffness	N/m
K_t	motor torque constant	Nm/A
L_{arm}	motor machine inductance	H
M_l	load mass	Kg
M_m	motor mass	Kg
M_s	screw mass	Kg
n	screw pitch	m/rad
R_{arm}	motor resistance	Ω
t_a	motor mechanical inertial torque	Nm
t_d	motor damping torque	Nm
t_l	motor load torque	Nm
t_f	motor friction torque	Nm

v_a	applied motor voltage	V
X_L	end-of-actuator position	m
X_m	motor position	m
X_s	screw position	m
ω_r	motor rotating speed	rad/s
θ_m	motor rotation	rad

List of Acronyms

HRA	High Redundancy Actuator
FTC	Fault Tolerance Control
FDI	Fault Detection and Isolation
IFDI	instrument fault detection and isolation
RKF	Robust Kalman Filter
FMEA	Failure Mode and Effects Analysis
PS	Parallel-in-Series
SP	Series-in-Parallel
PID	Proportional Integral Derivative
CV	Classical Voltage-Driven
CC	Classical Current-Driven
SVF	State Variable Feedback
CACSD	Computer-assisted Control System Design
LQR	Linear Quadratic Regulator
LQG	Linear Quadratic Gaussian
KF	Kalman Filter
PVA	Parameter Variation
BRT	Balanced Realization Truncation
PR	Physical Reduction
OF	Optimal Full Order
OM	Optimal Mathematical Reduction
OP	Optimal Physical Reduction
SE	Steady-state Error
RT	Rise Time
ST	Settling Time
OS	Overshoot
PV	Peak Velocity

P.M.	Phase Margin
G.M.	Gain Margin
BW	Band Width
DAQ	Data Acquisition

List of Figures

List of Tables

Contents

1. INTRODUCTION

Across the engineering industry, there is a constant pressure on engineers to build safer systems with improved availability that are, ultimately, more cost effective. A carefully designed and rigorously maintained mechanical component can help to reduce the possibility of it suffering a fault. But it is impossible to reduce it to zero. When a fault occurs, a fault tolerant capability will be required by the system, which is capable of tolerating component malfunctions whilst still maintaining desirable and robust performance and stability properties [1]. Approaches have been applied in many industrial and aerospace systems, e.g. for jet engines, flight control, electrical drives for railway traction, automotive engine management systems, etc. Low levels of functional redundancy in sensors and actuators (e.g. triplex and quadruple) are typically used to provide the system with the capability of fault tolerance and, thereby, to ensure the safety and stability of the whole system. In this thesis these concepts are extended to provide a highly reliable actuator with high levels of redundancy which may well be timely from an industrial viewpoint.

1.1 The Concept of High Redundancy Actuator

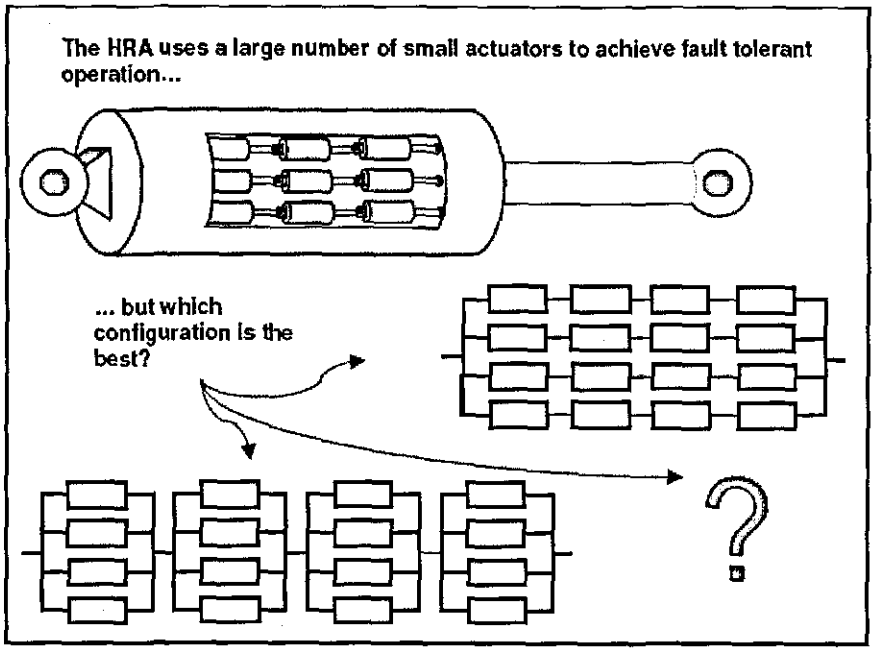


Fig. 1.1 High redundancy actuator comprising actuator elements in several possible configurations

The work in this thesis explores an idea in which actuators for high integrity and/or safety critical applications are provided by a system comprising a number of actuation elements, configured and controlled in such a way that faults in individual elements are inherently accommodated without resulting in a failure of the complete actuation system. A relatively high level of functional redundancy is used in this high redundancy approach, and it is motivated by the composition of human musculature. Human musculature can sustain damage and still function, sometimes with reduced performance, and even complete loss of a muscle group can be accommodated through kinematics redundancy, e.g. the use of just one leg. Fig 1.1 illustrates the concept schematically. As a high number of actuator elements are used in a redundant configuration, this approach is called a "High Redundancy Actuator" (HRA).

In the HRA, a large number of actuation elements work together as a single actuator. Any individual actuation element would not have enough power to satisfy the performance requirements. Depending on the exact configuration, travel and force of the actuator elements are added up, and together they exceed the requirements by a certain margin (determined by the designer). Element failures will lead to performance degradation depending on the fault and the configuration. However, the dimensioning is such that the remaining actuators can still satisfy the performance requirements, so that the system remains operational.

In some traditional approaches, fault tolerance is achieved at the control signal level so that such approaches are called fault tolerant control. The control algorithm engages fault detection and isolation units to monitor the controlled system, and when faults are found, the system and/or controller will be reconfigured using a backup channel(s) or different control algorithms to accommodate faults in existing channels. Some details about this technology will be introduced later in the literature review. However, it will be argued that such approaches present a potential problem: "Who will monitor the monitor?". In general such systems do not have any backup for the FDI and reconfiguration progress. The fault tolerant control can not accommodate faults in itself.

In the HRA approach, fault tolerance is achieved in another way: by engaging a more complex mechanical structure rather than a more complex control structure. In some traditional mechanical systems, a small number of parallel actuators are used instead of a single actuator to achieve the fault tolerance capability. The HRA comprising both series and parallel configurations will be more flexible to meet specific requirements, more effective for fault tolerance (through redundancy) with a graceful degradation. Most importantly, because it comprises a relevantly high

number of actuator elements, it will be able to ensure the system remains operational even if a certain number of actuation elements are blocked/faulty. The complexity in the mechanical structure will help to accommodate faults inherently but also could create problems with the controllability of the actuator. The challenge here is to achieve a balance between the two aspects to provide an acceptable performance with a reasonable cost. It might be argued that the system will suffer a higher probability of faults by complicating the mechanical structure. This is not the case. The HRA structure can reduce the possibility of failure of the whole actuator and increase the reliability of the actuator. It accommodates failures in individual elements as faults in the whole actuator until a certain level. From this point of view, the HRA can be called a failure tolerant system. By removing the possibility of failure in fault detection and isolation (FDI) or reconfiguration and accommodating failures in individual actuation elements, the HRA can avoid failing to finish its function and, in doing so, provides a higher reliability of the whole actuator system.

1.2 Methodology

Whilst the HRA concepts are largely technology independent, it is assumed throughout the thesis that the HRA being considered is an electro-mechanical actuator. This choice is one of convenience as components are readily available for this type of actuator. Ultimately HRAs could be developed from a variety of actuator technologies.

In order to verify the viability of the HRA approach as outlined above, the research described in this thesis spans three main areas: Concepts and Configurations, Control, and Demonstration. Each of these will discussed below.

The first area is aimed at understanding what different configurations of actuation elements can achieve and what features and advantages different configurations can provide. This will be achieved through building up and evaluating simulation models in MATLAB/Simulink. As mentioned in section 1.1, the HRA will comprise both series and parallel configurations, hence individual (element) actuator models will be built up that can be connected in series and parallel configurations. These are combined to produce two by two network structures which are evaluated in simulation studies.

The second area is aimed at verifying that a high redundancy actuator is controllable in both the fault free case and after several individual element failures. It is also desirable to quantify the performance levels and constraints imposed by this approach. Based on the models built, one of the two by two structures will be used for the controller design. Two design approaches, one classical and one modern control design, will be applied, with two different kinds of control structures (4 controllers in all).

Simulation results for the different designs will be generated under both healthy and fault injected situations.

Finally, to verify the simulation results, a lab-scale high redundancy actuator incorporating four actuation elements was built. The first real HRA uses a two by two structure with electromechanical actuators and is controlled with a MATLAB/xPC Real Time Target system. As in the simulation studies, several controllers were designed for this lab-scale HRA with testing under both healthy and fault injected situations.

1.3 The Contribution of the Thesis

This thesis introduces a new approach to safety critical actuation. The High Redundancy Actuator comprises a number of individual actuation elements that will work together to accommodate failures in a certain number of these elements. The literature survey verifies that this approach is very different from existing fault tolerant control system approaches in that it engages a more complicated mechanical structure rather than a more complex control structure.

The simulation studies discussed in this thesis provide initial verification of the controllability of a HRA under both fault-free and faulty conditions. They also permit judgements to be made about the best approaches to design of controllers for an HRA.

The final contribution and most significant of the thesis is to discuss the demonstrator. The world's first HRA (albeit with relatively low levels of actuation elements) allows verification of the simulation results (both open and closed loop) and proves that it is viable to control such an actuator without recourse to overly complicated controller designs or indeed controller reconfiguration.

A number of papers have arisen from the work described. Herein:

1. "Assessment of Strategies for Control of High Redundancy Actuators", 10th International Conference on New Actuators (Actuator 2006), Jun. 2006, Germany
2. "Modelling and Control of a Highly Redundant Actuator", International Control Conference (ICC 2006), Aug. 2006, U.K.
3. "LQG Control of a High Redundancy Actuator", IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Sep.

2007, Switzerland

4. "Validation of a Two by Two High Redundancy Actuator – Experimental Results", 11th International Conference on New Actuators (Actuator 2008), Jun. 2008, Germany
5. "Modelling and Control of a High Redundancy Actuator", MECHATRONICS - The Science of Intelligent Machines, (review)
6. "Experimental Results of a High Redundancy Actuator", Control Engineering Practice, (planned)

1.4 Structure of the Thesis

The thesis is structured in the following way. An introduction to the idea of a high redundancy actuator has been given above. The next chapter (Chapter 2) contains a literature survey where existing approaches/technologies for fault tolerant control will be presented. The main body of the thesis (Chapters 3 to 6) discuss the majority of the work mentioned in Section 1.2 (Methodology). That is, modelling, control design, simulation, and the demonstrator. The modelling study, Chapter 3, starts with model-building for an individual actuation element of the HRA, which is an electro-mechanical actuator, as well as some possible faults in the element. Then, series and parallel configurations will be modelled, and based on these models, two kinds of two-by-two grid structures including both series and parallel configurations will be given. The control design part will be located in Chapter 4, based on one of the two-by-two structures. A discussion of robustness will be firstly given, as well as an introduction of the control structures. Then it will introduce the design progress using classical approaches based on loop shaping method and modern optimal control approach (LQG optimal control). All the controllers will be tested and compared under both healthy and faults injected situations in Chapter 5, which contains the simulation part. The last chapter, Chapter 6, introduces the design and building up of a demonstration HRA control system based on the previous modelling and simulation study. The demonstration HRA engages a two-by-two structure based on electro-mechanical actuation elements, and the control system is set up based on a MATLAB/xPC Target toolbox. Both classical and optimal controllers will be designed for the hardware demonstration HRA, and results without and with faults injected will be compared. The thesis will finish with conclusion and some recommendations for further work in Chapter 7.

2. LITERATURE SURVEY

As introduced in the first chapter, the ability to tolerate faults (fault tolerance) is one of the most important features of the High Redundancy Actuator. Through the ability, it helps to increase the reliability and availability of some safety-critical systems, which are substantial activities in the area of fault tolerant research.

In this chapter, some background information about fault tolerant systems will be introduced, starting with an introduction to faults and the basic structure of fault tolerant systems. Some important technology will also be presented, including fault detection and isolation, control reconfiguration, and robust control. The design process for a fault tolerant system will be discussed.

2.1 Faults and failures

Generally, a fault is something that changes the behaviour of a system such that the system no longer satisfies its purpose [2]. It could be blocking (lock-up) of an actuator, disconnection from power supply, or a wrong control action given by a controller. From a system level, faults can happen to actuators, sensors, the plant itself, and even a controller as shown in Fig. 2.1.

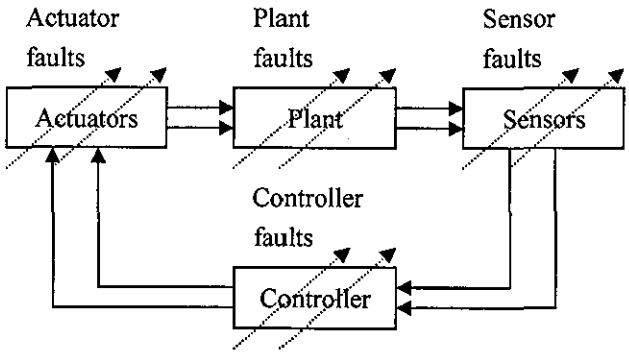


Fig. 2.1 Classification of faults in a feedback control system

Effects of faults will change the behaviour of whole system. If the system behaviour can be defined as a subset B on a space of all possible combinations of input (U) and output (Y) signals. Faults could affect the behaviour of a system from the faultless white set (B0) to a faulty grey set (Bf), as described in Fig. 2.2.

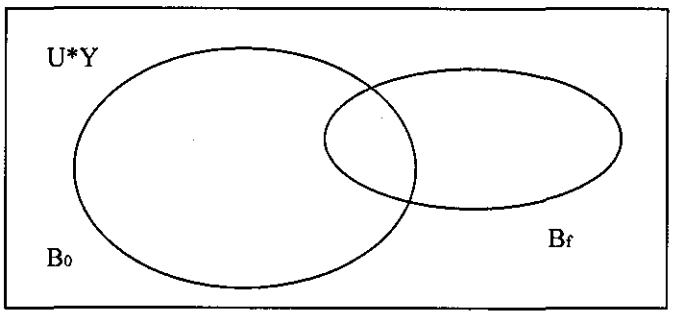


Fig. 2.2 Effect to system behaviour

As described above, a faulty system remains operational but the system's behaviour is changed under an undesired way. In addition, since the system's behaviour has been changed, the original controller could not behave like the designed way as well. In contrast to this, a failure describes the inability of a system to provide its desired function, which means the system has to be shut down to avoid damage to machinery and human. Although a faulty system can continue to operate in some situations, faults could develop into failures to cause damage. With these notions, the idea of fault-tolerant control can be stated as follows: Fault-Tolerant Control (FTC) has to prevent a fault from causing a failure at the system level [1].

2.2 Fault-Tolerant Control

Across the engineering industry, people are always looking for safe machinery. As described in the last section, fault-tolerant systems can avoid faults causing damage to machinery and human. Before talking about fault-tolerant control in detail, it is necessary to define several notions, including Safety, Reliability, and Availability.

Safety describes the absence of danger [2]. A safety system is engaged to protect the technological system from permanent damage; more importantly, to protect people from being injured. It will enable a shut-down before damage is caused.

Reliability is the probability that a system accomplishes its intended function for a specified period of time under normal conditions [2]. The reliability of the plant components is based on the design of the components themselves. Fault-tolerant control cannot improve the reliability of the components, but it can affect the reliability of the overall system.

Availability is the probability of a system being operational when needed [2]. Performance degradation is acceptable. Redundancy and reconfiguration can help to raise the availability of a system.

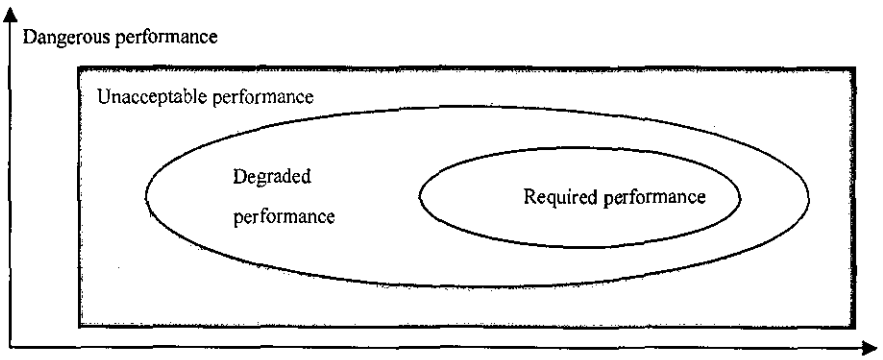


Fig. 2.3 Regions of system performance

Assuming that the system performance can be described in a two-dimensional plane, then regions of different system performance can be described in Fig. 2.3. A healthy system will be working in the region of required performance. When the system is affected by a fault, its performance will be degraded but the whole system is still available. A fault tolerant controller should be able to prevent further degradation of the performance towards the unacceptable or dangerous regions and it should move the system back into the region of required performance. Both unacceptable and dangerous performances should be avoided by a fault-tolerant controller. Especially the region of danger must be prevented by a fault-tolerant system.

A number of fault tolerant control theories are largely motivated by the control problems encountered in aircraft system design. Most application studies are based upon aerospace systems [1]. A typical application for a fault tolerant control system is the Airbus fly-by-wire system which is built to very stringent dependability requirements both in terms of safety and availability with the tolerance to both hardware and software failures [3]. A classical architecture of the Airbus fly-by-wire systems is illustrated in Fig 2.4. Redundancy is provided with five computers which are simultaneously active and each computer is divided into two physically separated channels. Failure detection is mainly achieved by comparing the difference between the two channels with a pre-determined threshold. The control computers must be robust which means the detection parameters (threshold, timing difference) must be sufficiently 'wide' to avoid unwanted disconnections and sufficiently 'tight' so that undetected failures are tolerated by the computer's environment. After double or triple failures, when it becomes impossible to compare the data of independent sources, the normal control laws are reconfigured into laws of the direct type where the control surface deflection is proportional to the stick input. The whole process includes all the aspects

of fault detection and isolation system, robust controller, reconfigurable control, and supervision, which make the whole system very complex and suffer a bigger possibility of fault in each part of the system.

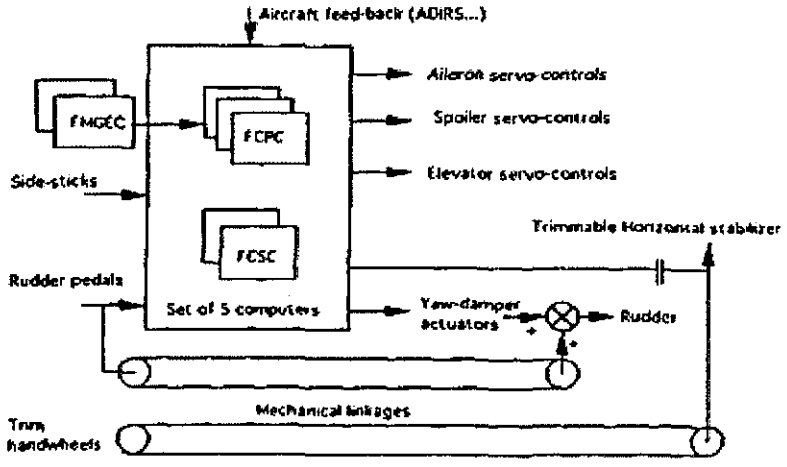


Fig. 2.4 Airbus fly-by-wire system [3]

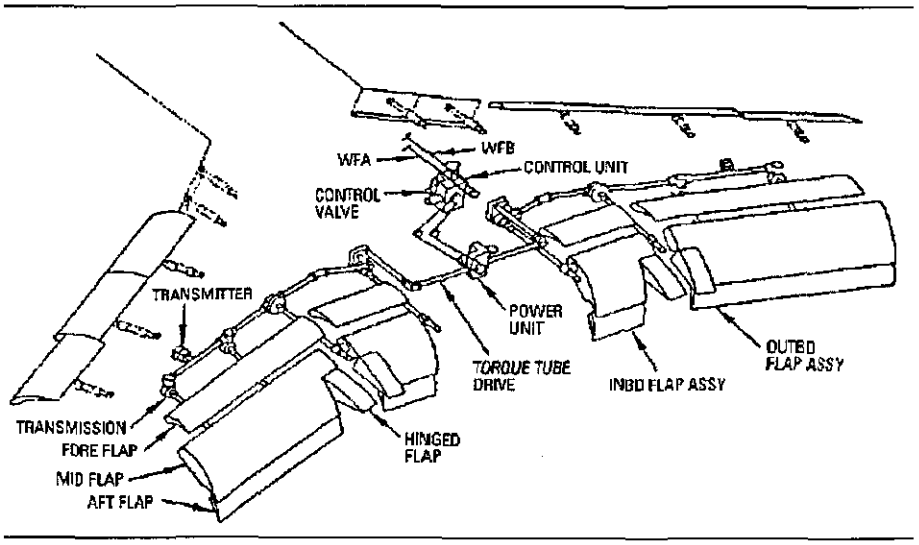


Fig. 2.5 Boeing 737 trailing edge flap drive system [4]

Redundancy can also be found in Boeing 737 trailing edge flap drive system which is explained in [4]. Four actuators are connected directly in parallel to each primary control surface, which can be found in Fig 2.5. Each actuator is synchronized and centrally powered. Shafting is generally designed to

withstand jam failures so that other devices are redundant safety features. Dual motors on the power drive units guarantee functional reliability on demand. But since all actuators are connected in parallel, one locked up actuator will cause a failure of the whole actuation system.

In these applications, fault-tolerant control systems generally employ redundancy in the plant and its automation system to make "intelligent" software that monitors the behaviour of components and function blocks [5]. The overall FTC strategy is to keep plant availability and accept reduced performance when critical faults occur.

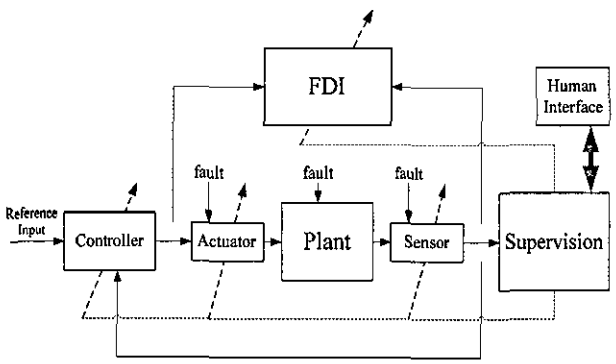


Fig. 2.6 Scheme of fault-tolerant control system with supervision subsystem [1]

A general FTC scheme is illustrated in Fig. 2.6. The traditional approach uses a relatively small number of "control channels" to provide functional redundancy, possibly supported by analytical redundancy but almost invariably involving fault detection and isolation and sometimes reconfiguration of the control action. Four main components are included in the scheme: the plant itself (including sensors and actuators), the fault detection and isolation (FDI) unit, the feedback (or feed-forward) controller, and the supervision system [1]. The solid line represents signal flow, and the dashed line represents adaptation (tuning, scheduling, reconfiguration or

restructure). The plant is considered to have potential faults in sensors and actuators (or other components). The FDI unit is responsible for providing the supervision system with information about the onset, location and severity of any faults. Based on the system inputs and outputs together with fault decision information from the FDI unit, the supervision system will reconfigure the sensor set and/or actuators to isolate the faults, and tune or adapt the controller to accommodate the fault effects. Such a process is called reconfiguration.

As an alternative to the structure shown in Fig. 2.6, fault tolerance also can be accomplished by means of well established control methods. Such methods, including robust control, are called passive fault tolerance [2], because the controller will not be changed under faulty conditions.

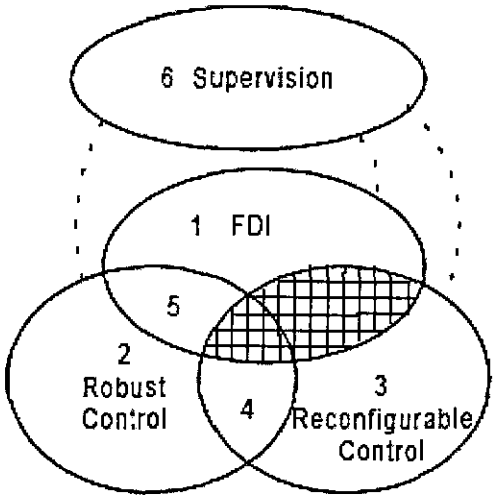


Fig. 2.7 The scattered areas of fault-tolerant control research

Here, attention is paid to the following fault-tolerant control aspects: FDI, Robust Control, Reconfigurable or Restructurable Control, and Supervision, which are shown in Fig 2.7. The FTC engages one or more aspects (described as grids in the Fig. 2.7) to ensure the system operate safely.

Technologies and examples will be introduced in the section 2.2.1 to 2.2.3.

2.2.1 Fault detection and isolation (FDI)

Based on the fault-tolerant structure as shown in Fig. 2.6, the first task concerns the detection and identification of faults. The diagnostic problem can be defined as: Find the fault for a given I/O pair [2].

The first required step of a diagnostic process is the fault detection. In this step, it needs to be decided whether or not a fault has occurred. Three requirements have to be satisfied for system faults to be detected: the system must be monitored, residuals must be available, suitable thresholds have to be evaluated for all the residuals [6].

The condition monitoring process not only involves continually acquiring system characteristic signals, but also evaluating the health of a system and its components through the analysis and interpretation of signals acquired from sensors and transducers [7].

For this purpose, plant monitoring systems combine sets of instruments or sensors, connected with computer systems, to provide information about system performance and working conditions, as well as environmental information. After being acquired, characteristic signals can be shown and analysed under both time and frequency domain for an on-line fault detection purpose. Investigating the fault patterns and their features associated with various failure modes also can help people to build up a knowledge base for detailed fault source diagnosis [7].

The signal processing techniques used in [7] is actually a continuous

comparison of the system under analysis with the knowledge base of failure modes. The residual used here is generated by "analytical redundancy". In another example about the Airbus fly-by-wire control system which has been introduced in [3], fault detection is achieved by comparing the difference between the two channels inside one computer with a pre-determined threshold, and the five computers in the fly-by-wire system provide "physical redundancy" for fault tolerant control.

From the examples, it can be found that redundancy is the only way to generate residuals which are based on the continuous comparison of the system under analysis with another system (or systems) characterized by the same behaviour and working in the same conditions. When this comparison is made among identical systems, it is know as "physical redundancy", on the contrary, when the comparison is made with whatever kind of modelling, it is know as "analytical redundancy" [8].

During the comparison, suitable thresholds have to be determined through experimental or simulation test. It has to be noted that threshold values mainly depend on measurement uncertainty and modelling errors, and strongly influence the fault detection. The higher the thresholds, the lower the fault sensitivity but also the lower the false alarm risk [6].

Residuals and thresholds may lead to the next step of the diagnostic process establishing: the fault location. The comparison between the active systems with the redundancy sub-system can help to locate the fault. Since this project is focused on the high redundancy actuator, some technologies used in the instrument fault detection and isolation (IFDI) systems is given in Fig. 2.8 as introduced in [6].

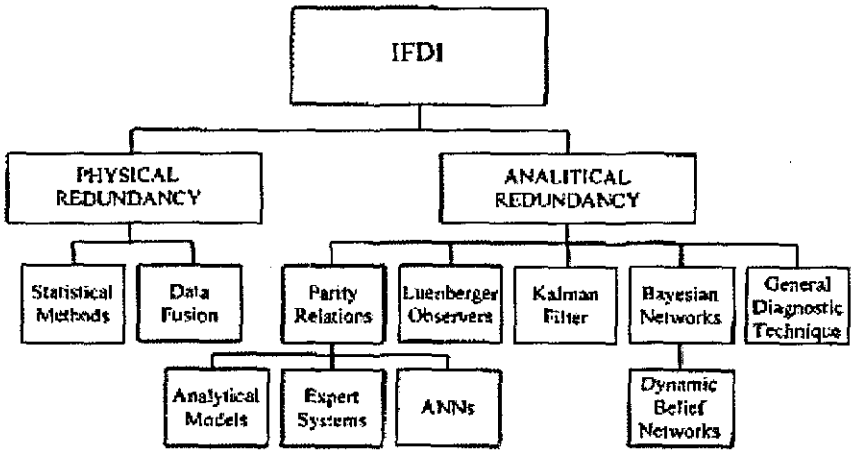


Fig. 2.8 Residual generation techniques [6]

All these techniques are based on either physical or analytical redundancy. Physical redundancy based solution requires double or triple the number of identical sensors for each measurement. This method reduces the effectively of the FDI progress and increases the cost as well. The model-based redundancy approach is often integrated with physical redundancy or even substituted, thus reducing costs and dimension. But the model based method can not simulate all faulty situations so that some faults could not be detected in some situations. An example can be found in [9] about the fault-tolerant flight control system design. The IFDI system is based on the Robust Kalman Filter (RKF). More details about other techniques are explained in paper [6].

The next step of fault diagnostic is identification and fault estimation to determine the kind of fault and its severity. To achieve this, information about the faults and about the way in which the faults affect the system is necessary [2]. Not all faults can be distinguished. Sometimes, it's only possible to locate a set of candidate faults, rather than a certain fault with a given measurement configuration, because the measured I/O pair may occur for all fault candidates. In this situation, no diagnostic method can remove

this ambiguity because the diagnostic result is caused by the system rather than the diagnoser. To solve this question, the monitoring system has to be changed by measuring different characteristics.

The information obtained from the diagnostic algorithm is going to be used to activate the controller redesign. Techniques for reconfigurable control will be introduced in next section.

2.2.2 Reconfigurable control

One way to continually satisfy the requirements on the closed-loop system after a fault changes the controlled system's dynamic behaviour is to reconfigure the control loop. Here, the reconfiguration means change in input-output between the controller and plant through changing the controller structure and parameters so that the original control objective is achieved (although performance may degrade) [5].

A control reconfiguration action will be taken following a fault detection signal. Examples can be found in [3], and [9]. In the Airbus fly-by-wire system, as soon as the active computer interrupts its operation relative to any function (control law or actuator control), one of the standby computers almost instantly changes to active mode with no or limited jerk on the control surfaces. Typically, duplex computers are designed so that they permanently transmit healthy signals which are interrupted as soon as the 'function' outputs are lost. After double or triple failures, when it becomes impossible to compare the data of independent sources, the normal control laws are reconfigured into laws of the direct type where the control surface deflection is proportional to the stick input. In another example of the sensor/actuator FDI and reconfigurable control for flight control system

design, the method based on the so called pseudo-inverse technique or control mixer concept is adopted for the reconfiguration approach [9].

Generally the controller design is based on the behaviour of the healthy system so that the closed-loop system performance can meet the requirements. But when the plant becomes faulty, it changes its behaviour, as well as the closed-loop system performance, so that its performance will no longer meet the requirements and could even become dangerous. In this case, the controller has to be re-designed in order to restrict the behaviour of the faulty system to meet the original requirements. This expresses the necessity and aim of controller reconfiguration from a system behaviour point of view.

The reconfiguration process has different levels. A simple or low level of controller re-design process can be called fault accommodation, which means to adapt the controller parameters to the dynamical properties of the faulty plant [2]. Only parameters in the controller will be changed in this principle. The controller configuration and the plant remains the same. A simple established way of fault accommodation is based on pre-computed gain parameters, each of which has been selected off-line for a specific fault. Then the reconfiguration process is simply to switch among these pre-designed controllers. However, the controller pre-design is based on the understanding of all possible faults and all resulting controllers have to be stored in the control software.

After severe faults have occurred, like sensor or actuator failures, which lead to serious structural change of the plant dynamics and even control loop break-down, it is not possible to adapt the controller by simply changing the parameters to meet the control requirements. A higher level principle of control reconfiguration by activating back-up systems and adjusting the whole control loop is necessary. The reconfiguration process includes

selection of a new control configuration, and more important, alternative actuator or/and sensors, so that the reconfigured controller can satisfy the performance specifications on the closed-loop system.

Techniques for control reconfiguration can be found in a large number of papers. Control law re-scheduling can be viewed as a system with feedback control where the feed back gains or structure are adjusted by feed forward compensation. Some methods include Feedback Linearization [10, 11], Model-following approaches [12, 13], and Pseudo-inverse modelling methods [14, 15]. Feedback linearization is an established technique in flight control. The estimated parameters are used to update the new parameters of the controller. Model-following is an alternative to feedback linearization. There are basically three strategies with basic fault tolerance properties: Explicit, Implicit, and Multiple model Kalman filtering. The goal is to emulate the performance characteristics of a desirable model with or without faults. The pseudo-inverse method principle is to modify the constant feedback gain so that the reconfigured system approximates the nominal system in some sense. This method uses no FDI mechanism and certain fault models are assumed.

The reconfiguration process has to be finished under real-time constraints and the time window should be as short as possible. This leads to a requirement on the reconfiguration process that it needs a robust FDI algorithm. It means the FDI algorithm can not be too sensitive which will produce fault alarm by mistake, but must be sensitive enough to make a correct alarm. The diagnostic residual in an FDI system has to be designed to be robust against modelling uncertainty while sensitive to faults. This is defined as a robust fault detection problem [16].

2.2.3 Robust control

The last two sections introduced the two main techniques which are used for the common FTC structure illustrated in Fig. 2.6. In the structure, FDI unit, together with Reconfigurable controller, work as a supervisor. The supervision level monitors the closed-loop system in a faultless situation, and re-designs the controller to make the control loop fault tolerant.

A closed-loop system also can have limited fault tolerance by means of a carefully chosen feedback design, taking care of effects of both faults and system uncertainties [1]. Making use of robust control techniques ensures that a closed-loop system remains insensitive to certain faults using constant controller parameters and without use of on-line fault information. The impaired system continues to operate with the same controller and system structure. The main objective is to recover the original system performance.

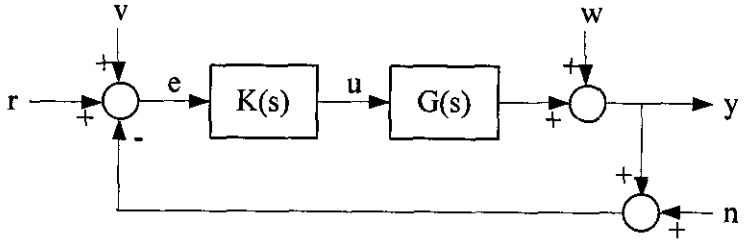


Fig. 2.9 A tracking system with input, output, and sensor noise

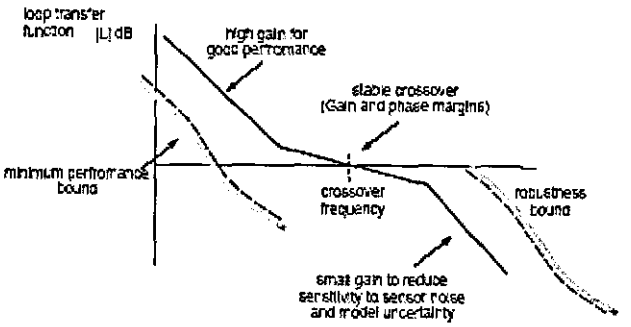


Fig. 2.10 Bode magnitude diagram for loop shaping

Fig. 2.9 illustrates a closed-loop tracking system with input, output, and sensor noise, marked by v, w, n respectively. $K(s)$ and $G(s)$ express the controller and the system respectively. r is the reference input, e is the control error, u is the control input, and y is the system output. A system is robust when it exhibits the desired performance despite the presence of substantial uncertainties of the system and the noises. A robust design on a bode magnitude diagram is showed in Fig. 2.10. A high gain at low frequency part ensures a good performance for tracking purpose. A suitable crossover frequency ensures a stable performance. A low gain at high frequency part can help to reduce the sensitivity to the sensor noise and model uncertainty. But when sensor noises and disturbances occur in the low frequency part, such a design could cause problems.

For any control system, the robustness against disturbances and modelling errors is a difficult but basic requirement, because it is impossible to get a perfect match between the mathematical model and the real process, and to describe disturbances introduced by sensors, actuators, and plant components precisely. If the effects of faults are similar to those of modelling errors and disturbances, the robustness ability can also be used to develop controllers to be insensitive to certain faults. Some faults have the effect of deviations on system dynamic parameters. These are effectively the multiplicative faults which affect the residual signal as a product of state/or control terms with parameter deviations. Other faults have an additive effect upon the system inputs and/or outputs and therefore affect the residual signals additively; these can be referred as additive fault signals. In the additive case, if the fault signals are not physically separable from the signals in nominal system signal flow, i.e. with significant difference in frequency band or signal direction etc., it is difficult, sometimes impossible, to use one robust controller to deal with both nominal and faulty conditions.

[1]

Some papers illustrate different technologies using robust control to provide the system with a fault tolerant capability. The technologies include quantitative feedback theory [17, 18], the frequency domain approach based on H_∞ [19], and robust design approaches to integrated control and fault estimation based upon the four parameter controller [20, 21].

2.3 Design for Fault tolerant control systems

Fault tolerant control should ideally be accompanied by a systematic and integrated approach to design. The strategy should commence with an understanding of the structure of the system, the reliability of different components, the type of redundancy available (or to be generated) and the types of controller function which are available and might be required. Each part will be discussed in this section.

2.3.1 Fault tolerant system requirement analysis

The first stage of the development of a fault tolerant system is the requirement analysis and system plan. This can include the following procedures: Possibility analysis for component faults; Failure Mode and Effects Analysis (FMEA); System reliability analysis; Reliability distribution [22].

2.3.2 Redundancy design

At this stage, the nature and location of all redundancies in the process must be determined, i.e. the type of redundancy and whether it is suitable for the particular situation; the level of redundancy, etc. As described in [1], two types of redundancy are used in fault tolerant systems: Physical redundancy, and Analytical redundancy.

Physical or hardware redundancy means that multiple independent hardware channels (e.g. triplex/quadruplex replication) with a majority vote selection

of healthy system channels are used [1]. A usual procedure is that a non-impaired identical alternative component is active to replace an impaired component when a fault occurs. The replacement is often based on the basis of known reliability - use up the best components first.

Analytical or functional redundancy is achieved by careful design or by arranging different subsystems to make the function of these subsystems overlap [1]. Analytical redundancy provides higher system independence than direct redundancy. As it is based upon functional or model information, it suffers more from system non-linearity and parameter or model structure uncertainty.

Redundancy is necessary to achieve fault tolerance of the systems. It is also necessary in the FDI systems as explained in section 2.2.1.

2.3.3 Fault accommodation design

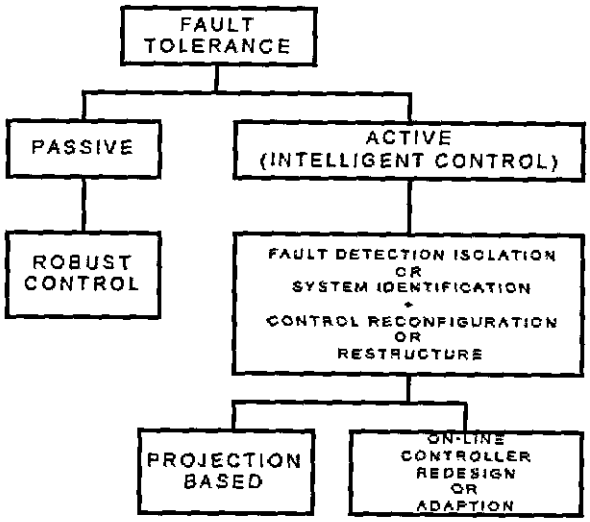


Fig. 2.11 Decomposition of fault tolerant control

Next the control and input/output requirements for each fault must be determined to provide guidelines for the development of a fault tolerant control design. The basic fault tolerant control methods are listed in Fig. 2.11. Basically, two kinds of methods are provided: the active approaches, and the passive approaches.

The passive approaches are using robust control technologies, and the active approaches are using FDI systems and the reconfiguration control technologies. Both kinds of approaches have been described from section 2.2.1 to 2.2.3. But as introduced in these sections, each technology has its own problem. The active method is useful, but the complexity of such a system could cause greater possibility of faults at the plant as well as the FTC system itself. The robust control method is simpler, but it can only be tolerant to limited types of faults and to design a robust system is difficult.

An example of fault tolerant control system design can be found in [22]. The overall structure of the proposed fault tolerant control system is depicted in Fig. 2.12, which includes modules of command management, reference models, fault detection and diagnosis scheme, reconfigurable mechanism, and model following reconfigurable control. Performance degradation is considered in the impaired systems so that two reference models are designed in the structure, because the saturation or worse use of the actuators will cause further damage to the whole system [22]. An actuator faults model also is considered in this paper using a control effectiveness factors γ_k^i. The objective of fault detection and diagnosis is to determine the extent of the loss in the control effectiveness by estimating γ_k^i on-line in real-time so that an on-line automatic reconfigurable controller can by synthesized. Three reconfigurable controllers $\{K_x, K_{x^m}, K_r\}$ are used to ensure that the closed-loop system follows the degraded reference model. The whole design

process is presented in [22].

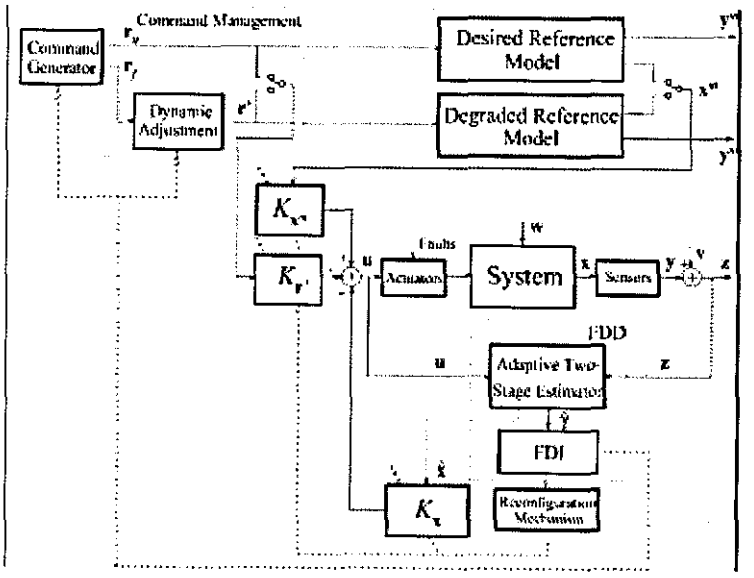


Fig. 2.12 Overall structure of proposed fault tolerant control system [22]

This example shows the complexity of a FTC system. Although the plant system can be monitored and controlled well to avoid a failure of the system, the monitor system itself does not have such a monitor system to ensure the effective and correct detection of faults. This disadvantage leads us to the high redundancy approach.

2.4 High redundancy approach

A literature survey summarising fault tolerant control systems has been given in this chapter. Technologies used in the fault tolerant control system design, including FDI, reconfigurable control, and robust control, and some examples most of which are applied in the aerospace industry are introduced.

In these techniques, the ability of fault tolerance is approached on the control level. A much more complex control algorithm is applied to monitor the nominal system. But the introduction of the monitor system also will introduce the possibility of faults in the monitor system itself. The approach can be argued with such a question: Who or what will monitor the monitors?

The HRA proposes an alternative route to fault tolerant actuation by engaging a relatively complex mechanical structure rather than a control structure. Through connecting the redundancy actuation elements and making them work together, the HRA can keep operating using the same controller without any reconfiguration. The FDI unit is no longer necessary for this fault tolerant system although it still can be included for monitoring purpose. The safety of the system will be improved through removing the possibility of faults occurring within the FDI reconfiguration process. On the other hand, the HRA also introduces new theoretical challenges, both because of the complexity and because of the high number of possible configurations. The challenges could be concluded as two problems: how to connect the redundancy actuation elements, and how to control such a complex configuration? These are the main problems that the author has been addressing

Some transportation systems, as shown in Fig. 2.13 and 2.14, has been

reported in [23, 24] to engage a redundancy actuator system. Actuators are connected in both series and parallel and move one by one to transport packages. By engaging more actuators, the system can move bigger or heavier parcels for a longer distance.

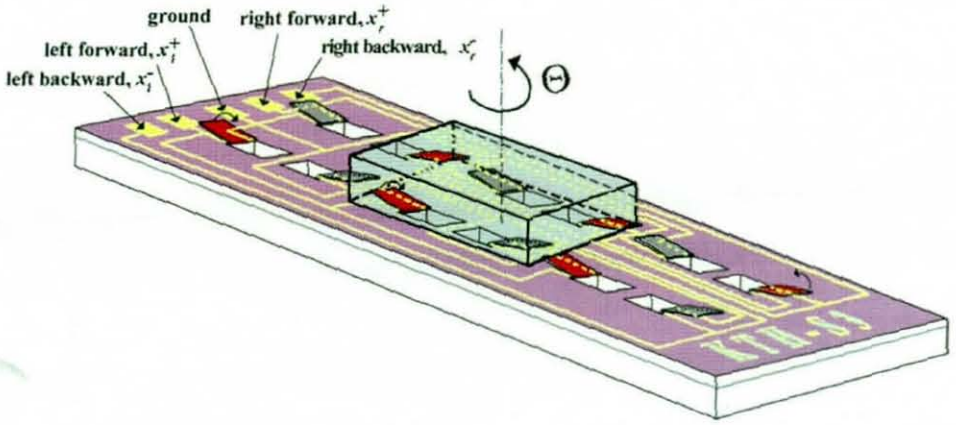


Fig. 2.13 A robust micro conveyer realized by arrayed polyimide joint actuators [23]

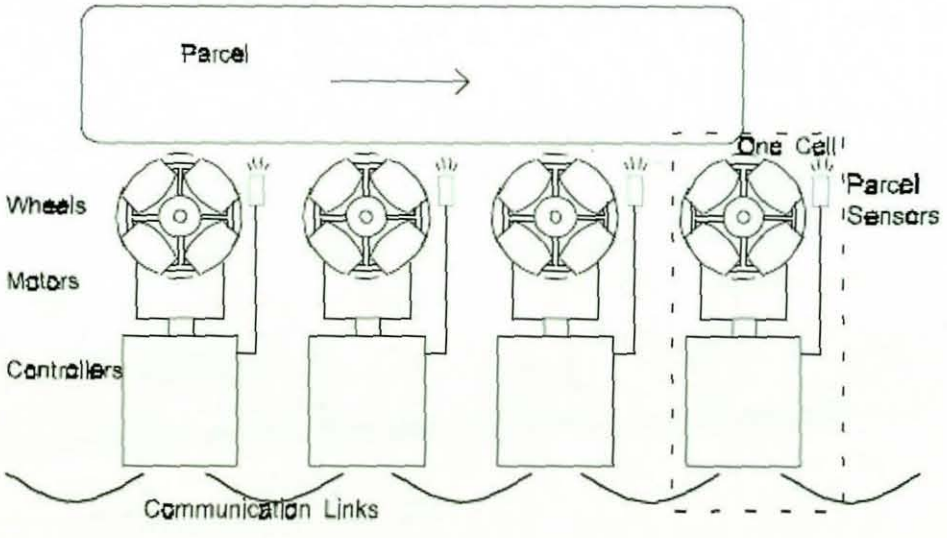


Fig. 2.14 Parcel Manipulation and Dynamics with a Virtual Vehicle [24]

Here the main purpose is not for fault tolerance and the redundancy actuators are not moving together so that the efficiency is still low, but these transport systems give basic ideas about how we can connect the HRA

actuation elements and make them work together to provide the fault tolerance ability.

After introducing some basic ideas of fault tolerant control, the next step is to modelling a relatively simple configuration of the HRA.

3. MODELLING STUDIES

As discussed previously, a HRA comprises a number of actuation elements to provide the fault tolerance capability. But before building up a real HRA, some questions have to be answered like: what kind of structure can be used to connect actuation elements; how the HRA is going to be controlled; and the most important thing, whether the HRA can accommodate elements' faults without reconfiguration? All these questions are needed to be studied in a computer associated environment based on a series of simulation models. The purpose of the modelling studies is to explore some key characteristics and behaviours of a HRA system based on mathematical expressions of physical laws. Here all models are built in the MATLAB/Simulink toolbox.

This chapter starts by introducing a single electro-mechanical actuation element. Then two simple configurations, series and parallel, will be modelled. Finally grid structures comprising both series and parallel configurations will be introduced. Related modelling work has been published in [25, 26, 27, 28].

3.1 Individual Actuation Element

An electro-mechanical actuation is chosen as the technology for this study because it is used in many industry applications, including manufacturing, process plant, railway vehicles and aerospace systems. The structure is relatively simple, and it is easy to be controlled. Fig 3.1 describes the basic components of an electro-mechanical actuator. A DC (direct current) motor, working as the electrical part, generates the torque and moves the mechanical part (screw), to provide a linear speed and output force. By changing the voltage applied to the motor, the whole actuator can be controlled to get the desired force/speed.

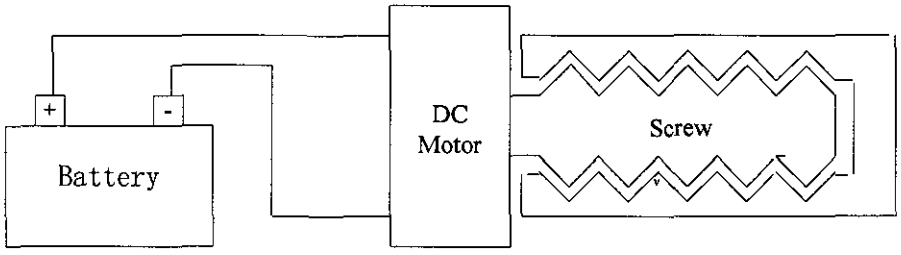


Fig. 3.1 An individual electro-mechanical actuator

3.1.1 Modelling of DC Motor

DC motor is one working mode of a direct current machine where electrical power is converted into mechanical power as output. Two systems are included, the electrical one and the mechanical one as described in Fig. 3.2.

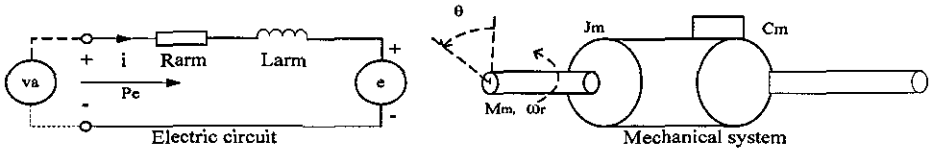


Fig. 3.2 Electric circuit and mechanical system

In the electric circuit, the electrical power applied with a voltage of v_a is opposed by a resistance of the conducting path (R_{arm}), the machine inductance due to non-useful magnetic flux (L_{arm}), and an e.m.f. (Electromotive Force) $e = e_r + e_p$, where e_r is the motional e.m.f. resulting from the linear or angular speed, and e_p is the pulsating (or transformer) e.m.f. appearing in the circuit if the flux linking it changes with time. A permanent magnet excitation is assumed here so that e_p does not exist. Then the voltage relation is expressed as: $v_a = R_{arm} i_a + L_{arm} \dot{i}_a + e_r$.

In the mechanical system, the quantities concerned with the mechanical nature of the machine include the inertial, damping and frictional properties of its structure. The mechanical system is assumed as a rotating system. An electromagnetic torque translated from previous electric circuit is applied to rotate the mechanical machine having a moment of inertia J_m at the rotating speed of ω_r. It is opposed by an inertial torque t_a accounting for kinetic energy storage resulting from a change of speed: $t_a = J_m \dot{\omega}_r$, a friction torque t_f and a damping torque t_d accounting for loss torques in friction, windage, and eddy currents, etc, which can be taken as proportional to speed with an appropriate 'damping' coefficient C_m: $t_f + t_d = C_m \omega_r$, and a load torque t_l. Then the torque balance equation can be expressed as: $t_e = J_m \dot{\omega}_r + C_m \omega_r + t_l$.

Inside a DC machine system, the electrical power $e_r i_a$ is converted to a mechanical power $t_e \omega_r$ (where t_e is the electrical torque) to drive the shaft,

i.e. $e_r i_a = t_e \omega_r$. In permanent magnet excitation, the field flux is constant so that e_r is directly proportional to the speed ω_r and can be expressed as $e_r = K_e \omega_r$, where K_e is called the voltage constant depending on the number of armature turns. And the electrical torque is proportional to the armature current which can be expressed as $t_e = K_t i_a$, where K_t is called the torque constant. The basic relation $e_r i_a = t_e \omega_r$ shows that $K_t = K_e$.

According to previous analyse, the behaviour of a DC motor can be expressed using the following equations:

$$\begin{cases} v_a = R_{arm} i_a + L_{arm} \dot{i}_a + K_e \omega_r \\ K_t i_a = J_m \dot{\omega}_r + C_m \omega_r + t_l \end{cases} \quad (3.1)$$

which can be combined as:

$$v_a = \frac{L_{arm} J_m}{K_t} \ddot{\omega}_r + \frac{R_{arm} J_m + L_{arm} C_m}{K_t} \dot{\omega}_r + \frac{R_{arm} c_m + K_e K_t}{K_t} \omega_r + \frac{R_{arm} t_l}{K_t}$$

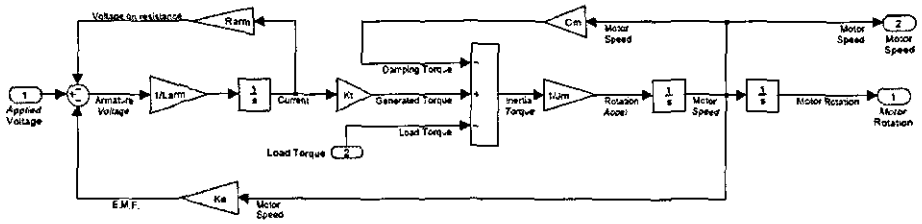


Fig. 3.3 DC motor linear model

Using these equations, a DC motor can be modelled in Simulink using rotation of the motor θ_m as the output, where $\omega_r = \dot{\theta}_m$, as shown in the Fig. 3.3

A practical DC servo motor will be applied to test this Simulink model. The motor we chosen is the M500 series motors from McLennan Servo Supplies Ltd [29], which provide the ideal solution for use in applications that require

output power ratings up to 100 Watts combined with fast response and accurate control. Details of the motor with technical specification and performance are listed in Table. 3.1.

Table. 3.1 Technical specification and performance of the DC servo motor

Technical Specification			
Maximum supply voltage	60Vdc	Mass	2kg
Motor voltage constant	0.121Nm/A	Motor torque constant	$0.121 V/rad/s$
Maximum peak torque	36Ncm	Mechanical time constant	8ms
Rotor inertia	$0.530 kgcm^2$	Terminal resistance	2.2Ω
Rotor inductance	6.4mH	Rotor construction	ironless
Commutation	copper	Bearings	Ball
Maximum axial force	22N	Maximum radial force	45N
Performance @ 36Vdc			
No load speed	2700rpm	Rated speed	2250rpm
Rated torque	0.4Nm	Peak torque	1.44Nm
No load current	0.3A	Rated current	3.5A

Considering a very small friction and damping torque loss which is about 0.01% at rated speed which is about $250 rad/s$, we get:

$$C_m = 0.01\% * 0.4 / 250 = 1.7 * 10^{-7} Nm/rad/s$$

Table. 3.2 Parameters of a real electro-mechanical actuator

Parameter	**Description**	**Value/Unit**	**Parameter**	**Description**	**Value/Unit**
R_{arm}	motor resistance	2.22 Ω	L_{arm}	inductance of the windings	6.4 mH
K_e	voltage constant	0.121 V/rad/s	K_t	torque constant	0.121 Nm/A
J_m	motor inertia	5.3e-5 Kgm^2	C_m	damping coefficient	1.7e-7 Nm/rad/s

Using the parameter values listed in Table. 3.2, experiments are tested using this Simulink model. A voltage of 36V is applied at 0.1 second and a step load of 0.4Nm is applied at 0.2 second. Speed, torque and current outputs of the model are shown in Fig. 3.4.

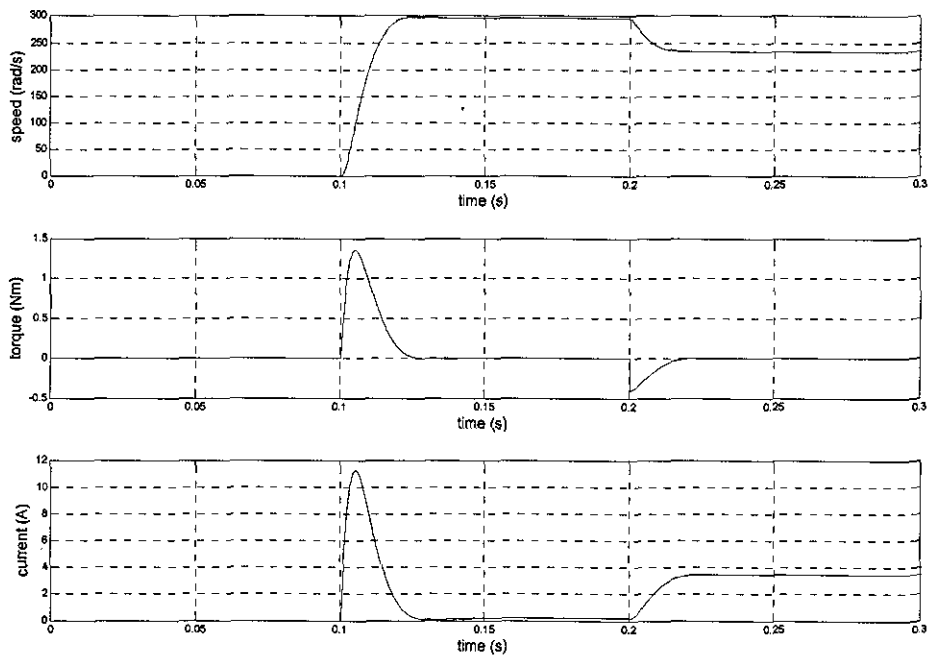


Fig. 3.4 Simulation result of DC motor

According to the simulation, the no-load motor speed is about 295rad/s (about 2800rpm) and the rated speed is about 235rad/s (about 2240rpm). The peak torque generated is about 1.4Nm and the rated torque rated is about 0.4Nm. The no-load current is at about 0.2A and the rated current is about 3.5A. These simulation results are very similar to the performances listed in Table 3.1 so that the model can represent the real motor well.

3.1.2 Modelling of mechanical structure

A structure of an electro-mechanical actuator has been given in Fig. 3.1. An equivalent model will be shown in Fig. 3.5 so that the mechanical structure can been discovered more clearly, where n represents the screw pitch; K_m represents the motor series stiffness; K_s and C_s represent the screw stiffness and damping respectively; θ_m expresses the angular rotation of the motor shaft; X_s and X_L express the motions of the screw and the end of the actuator respectively; M_s and M_l represent the masses of the screw and the load respectively.

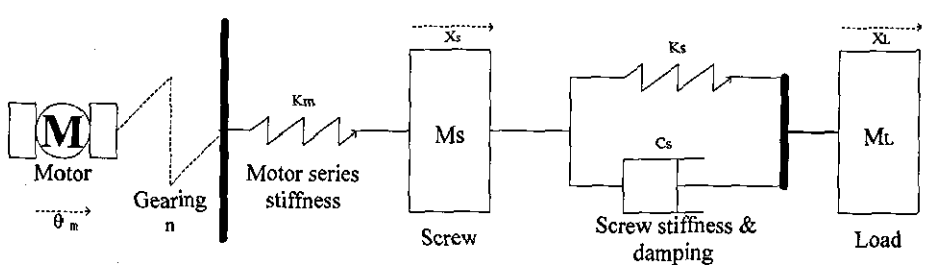


Fig. 3.5 Equivalent electro-mechanical actuator model

As shown in the model, the angular rotations generated by the DC motor are transformed by a gearing system to provide a linear motion to the screw. The output of the actuator is the force which is generated by compression of the screw as the motor is turned. A free body diagram of mechanical parts, including both the screw and the load, will be given in Fig. 3.6, using which mathematical expressions can be explored. Note the actuator is assumed to be acting horizontally so that gravity is neglected.

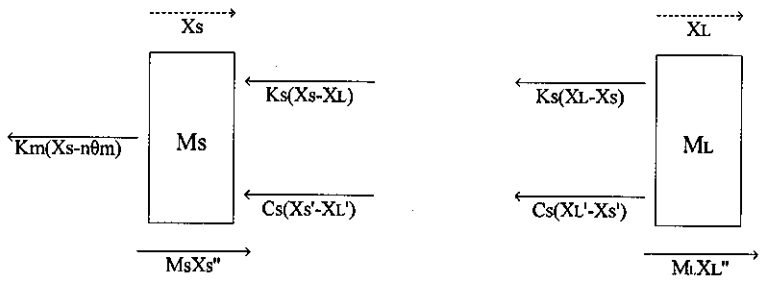


Fig. 3.6 Free body diagram of the screw and the load in individual actuator

The analyses are based on the Newton laws. Since $\sum F = 0$, according to the free body diagram, we can get the following formula:

$$\begin{cases} M_s X_s'' + K_m\left(X_s - n\theta_m\right) + K_s\left(X_s - X_L\right) + C_s\left(X_s' - X_L'\right) = 0 \\ M_L X_L'' + K_s\left(X_L - X_s\right) + C_s\left(X_L' - X_s'\right) = 0 \end{cases} \quad (3.2)$$

Using this equation, a linear model of the mechanical actuator is built in Simulink, which is given in Fig. 3.7. The actuator is driven by the motor introduced in Section 3.1.1, and feed a load torque back to the motor. Then, the actuator force, velocity, and position can be calculated when suitable

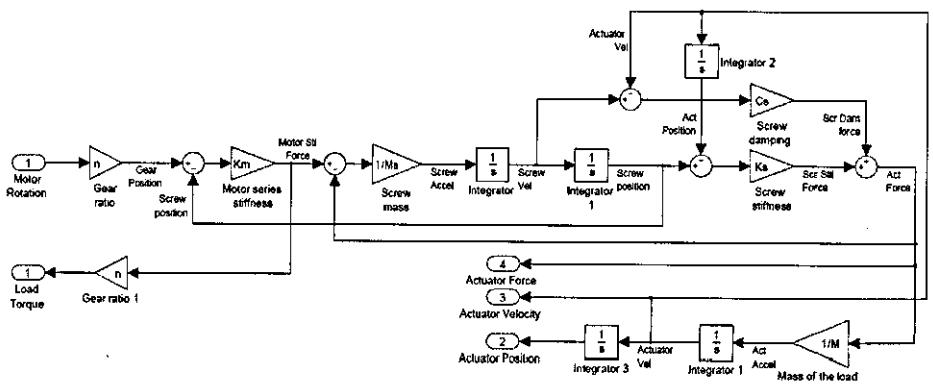


Fig. 3.7 Mechanical actuator linear model

3.1.3 Modelling of electro-mechanical actuator

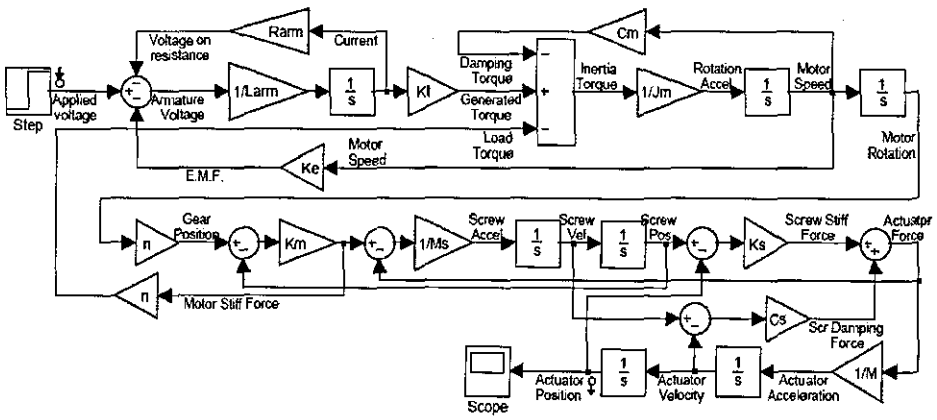


Fig. 3.8 Individual electro-mechanical linear model

Table. 3.3 Parameters in Simulink model

Parameter	Description	Value/Unit	Parameter	Description	Value/Unit
R_{arm}	motor resistance	2.2 Ω	L_{arm}	inductance of the windings	6.4 mH
K_e	voltage constant	0.121 V/rad/s	K_t	torque constant	0.121 Nm/A
J_m	motor inertia	5.3e-5 Kgm^2	C_m	damping coefficient	1.7e-7 Nm/rad/s
K_m	motor stiffness	1e7 N/m	n	screw pitch	3.82e-4 m/rad
K_s	screw stiffness	1.8e5 N/m	C_s	screw damping	1.2e3 N/m/s
M_s	screw mass	2 kg	M_L	load mass	20 kg

Both parts of an electro-mechanical have been expressed mathematically in equation 3.1 and 3.2. Both models also have been built up in Simulink shown in Fig. 3.3 and 3.7. The whole model of an electro-mechanical

actuator can be obtained by linking them together. The model is shown in Fig. 3.8. Parameters' value are listed in Table. 3.3, where the motor parameters are provided by McLennan [29] and other typical mechanical parameters are derived from Pratt [30]. Then experiments can be tested.

With a step voltage of 36V supplied at 0.1 second to the actuator with a load of 20kg. The simulation results of force, velocity, position outputs will be given in Fig. 3.9.

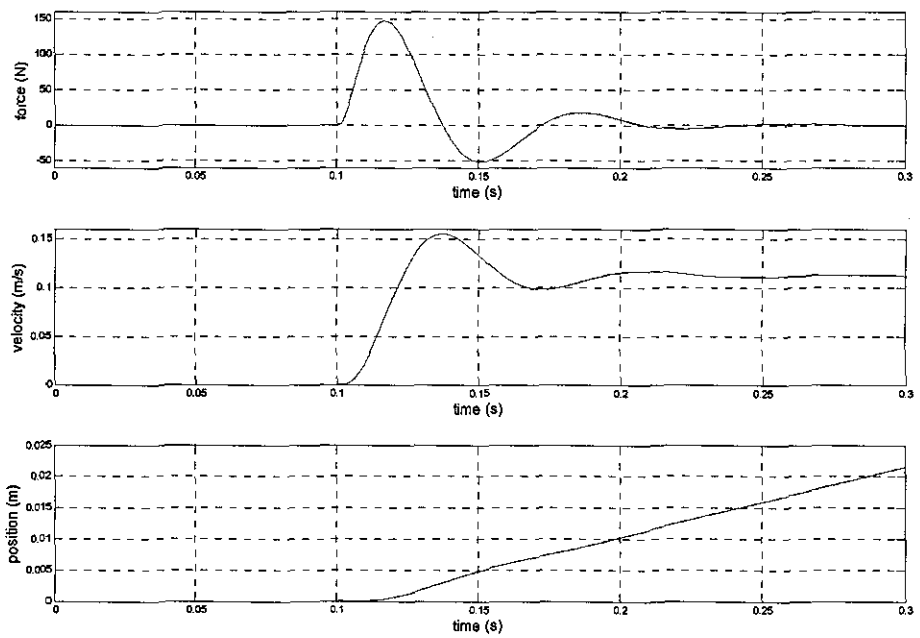


Fig. 3.9 Simulation result of the individual actuator

From the figure, it can be seen that the actuator reaches the steady state at about 0.25 second with a steady extension velocity at about 0.11m/s. The position gives a ramp output which can be expected. Although no performance document can be used to compare the simulation results, the results can be expected based on the physical understanding of the electro-mechanical actuator. Based on this model of individual actuator (or actuation element), further research will focused on modelling connections

of a number of elements.

Based on the mathematical expressions as shown in equation 3.1 and 3.2, a state-space model of the electro-mechanical actuator also can be obtained in the form: $\dot{x} = Ax + Bu$.

The state vector is chosen as $x = \begin{bmatrix} i_a & \dot{\theta}_m & \theta_m & \dot{X}_m & X_m & \dot{X}_L & X_L \end{bmatrix}^T$, whose meaning is described in Table. 3.4. The input u and output y are chosen as $u = v_a$, $y = X_L$, where v_a expresses the applied voltage and X_L expresses the actuator position.

Table. 3.4 States in state-space model

State	Description	Unit	State	Description	Unit
i_a	armature current	A	θ_m	angular rotation of the motor shaft	rad
X_m	screw position	m	X_L	end-of-actuator position	m

Then the state matrix A, input matrix B, and output matrix C can be expressed following, while D matrix is 0.

$$A = \begin{bmatrix} -\frac{R_{arm}}{L_{arm}} & -\frac{K_e}{L_{arm}} & 0 & 0 & 0 & 0 & 0 \\ \frac{K_t}{J_m} & -\frac{C_m}{J_m} & -\frac{n^2 K_m}{J_m} & 0 & \frac{nK_m}{J_m} & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{nK_m}{M_s} & -\frac{C_s}{M_s} & -\frac{K_s + K_m}{M_s} & \frac{C_s}{M_s} & \frac{K_s}{M_s} \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{C_s}{M_L} & \frac{K_s}{M_L} & -\frac{C_s}{M_L} & -\frac{K_s}{M_L} \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, B = \begin{bmatrix} \frac{1}{L_{arm}} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

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

3.1.4 Modelling of faults

As introduced, an electro-mechanical actuator comprises two parts: a DC motor and a mechanical actuator, so that faults also can be separated to electro and mechanical types. Although there are a lot possible faults could happen in both parts, we are only looking at some serious faults which will cause failure of an individual actuator. These faults can not be accommodated in a control level. Using HRA, we are trying to prove the possibility of using mechanical structure to accommodate these kinds of faults in actuation elements. Three kinds of faults are considered here: open circuit, short circuit, and lockup. The first two are electro faults and the final one is a mechanical fault.

Open circuit means a break in electrical circuit so that no current nor torque will be generated by the DC motor. This means a loss of force capability of an actuator. Without drive, the mechanical actuator can not be moved by itself, but on the other hand, it also will not generate any reverse force when it is moved by any outside force. The actuator will become very soft to be moved. In Simulink, this can be modelled by replacing the current inside the DC motor model with a zero constant.

Short circuit means outside voltage can not be applied to move the actuator, but the electrical circuit is not broken. In this situation, the motor can not generate any positive torque to drive the actuator either, but when being moved, the motor will generate reverse torque trying to stop the movement. This will make the actuator stiffer to be moved. In Simulink, this can be modelled by linking the actuator to a zero voltage permanently.

Lockup means that the actuator can not be moved any more no mater driven by its inside motor or any other outside force. Generally this is due to

mechanical interference within the mechanism. Under this situation, the actuator becomes very stiff and hardly to be moved and this means a loss of travelling capability of the actuator. In Simulink, it can be modelled by lock the integrator which will integrate the actuator velocity to actuator position.

3.2 Series and Parallel Connection

Series and parallel are the two basic ways to connect actuation elements. In the individual actuator model, as expressed in Fig. 3.8, force, speed and position all can be chosen as the output of the actuator. All of them are vectors, which give us a great number of choices to set up the network since the vectors can be added dimensionally. Some three-dimensional actuation configurations have interested some researchers [8], but we are interested in planar configurations and will place the actuators linearly since it is the most efficient way to connect them based on the principle of the addition of vectors. Intuitively it would seem that the biggest value can be attained by addition of the vectors in the same direction, so that the output direction won't change in the case of element failure. Same as the individual actuator modelling, both mathematical expression and Simulink model will be introduced.

3.2.1 Modelling of series configuration

Firstly, an equivalent model of a two in series configuration is given in Fig. 3.10. The basic components are the same as the individual actuator model. However, the motor's mass M_m is introduced because the motor of the second actuator appears as part of the load of the first one.

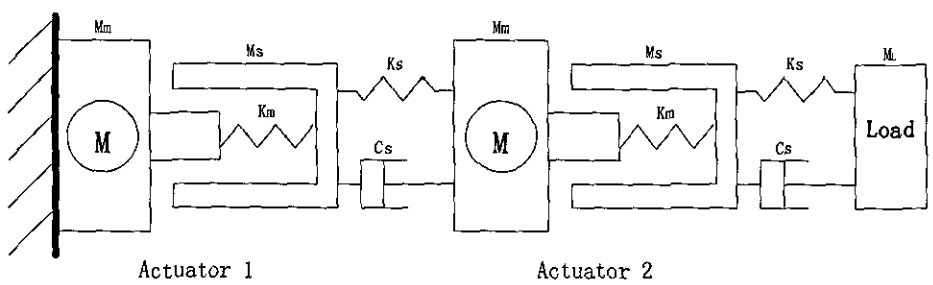


Fig. 3.10 Equivalent series configuration model

Excluding the mass of the motor of actuator 1 which is fixed, there are four moving parts in the model as shown in Fig. 3.11: the screw of actuator 1 (top left), the motor of actuator 2 (top right), the screw of actuator 2 (bottom left), and finally the load (bottom right). The identifiers are remained same as used in the individual actuator model, while '1' and '2' express the actuator 1 and 2 respectively and X_m expresses the movement of the motor part.

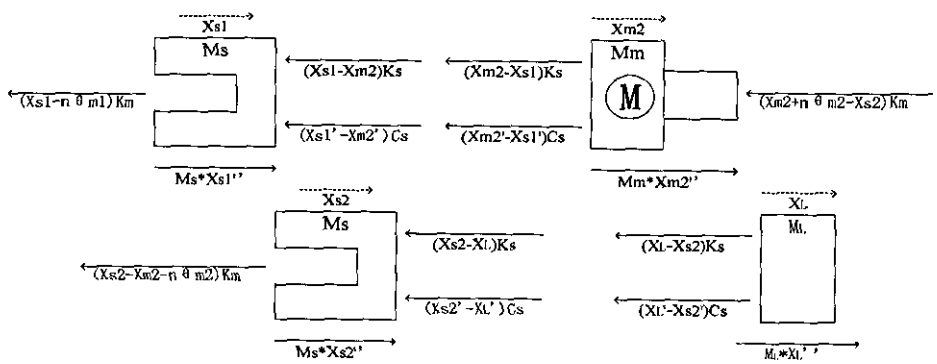


Fig. 3.11 Free body diagram of series configuration

Based on the Newton laws, we can derive the equations of movement for each part expressed in the figure above:

$$\begin{cases} M_s X''_{s1} + (X_{s1} - n\theta_{m1})K_m + (X_{s1} - X_{m2})K_s + (X'_{s1} - X'_{m2})c_s = 0 \\ M_m X'_{m2} + (X_{m2} - X_{s1})K_s + (X'_{m2} - X'_{s1})C_s + (X_{m2} + n\theta_{m2} - X_{s2})K_m = 0 \\ M_s X''_{s2} + (X_{s2} - X_{m2} - n\theta_{m2})K_m + (X_{s2} - X_L)K_s + (X'_{s2} - X'_L)C_s = 0 \\ M_L X''_L + (X_L - X_{s2})K_s + (X'_L - X'_{s2})C_s = 0 \end{cases} \quad (3.3)$$

A Simulink model of two actuators in series is expressed in Fig. 3.12.

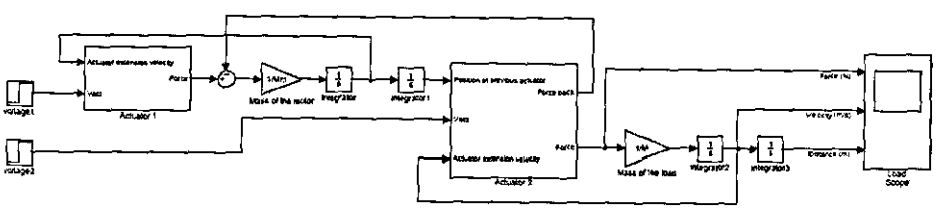


Fig. 3.12 Simulink model of a two in series configuration

There are two voltage blocks which means the two actuators can be controlled separately. Inside the configuration, the Actuator 1 is same as the individual actuator model shown in Fig. 3.8 with a load of the Actuator 2's motor. But the Actuator 2 model (as shown in Fig. 3.13)has been changed because it not only need information of Actuator 1's position but also send a force back to the Actuator 1 according to the equations 3.3.

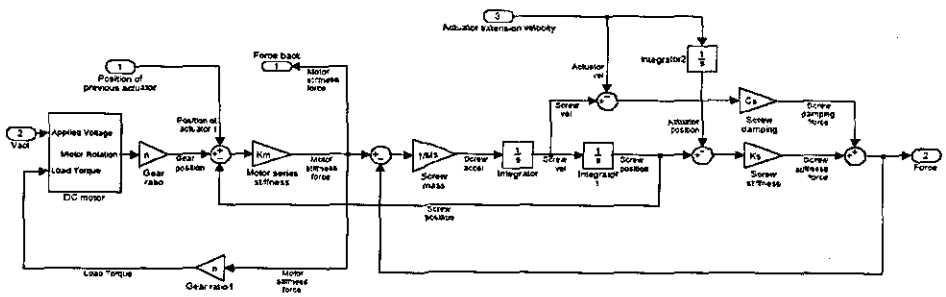


Fig. 3.13 Actuator 2 linear model

Only two elements are considered in this series configuration. It is also possible to connect more elements in one series configuration. When there are three or more elements are connected, the elements can be separated into three kinds. The first two have been shown in Fig. 3.12: a bottom actuator (as the Actuator 1), which is fixed and can not be moved, and a top actuator (as the Actuator 2), which is connected with the load as the end of the configuration. The bottom actuator is connected with the next actuator's motor as the load, and gives its position information to the next one. The top actuator gets the information about the previous actuator's position and feeds a force back. Together with the force generated by the top actuator, this in turn is used to get the previous element's position. The third kind of actuator, which is not shown in Fig. 3.12, is called intermediate actuators, which are connected between bottom and top actuators. This kind of elements combines features of both bottom and top ones: using the force back to generate its own position information and then pass it to the next element.

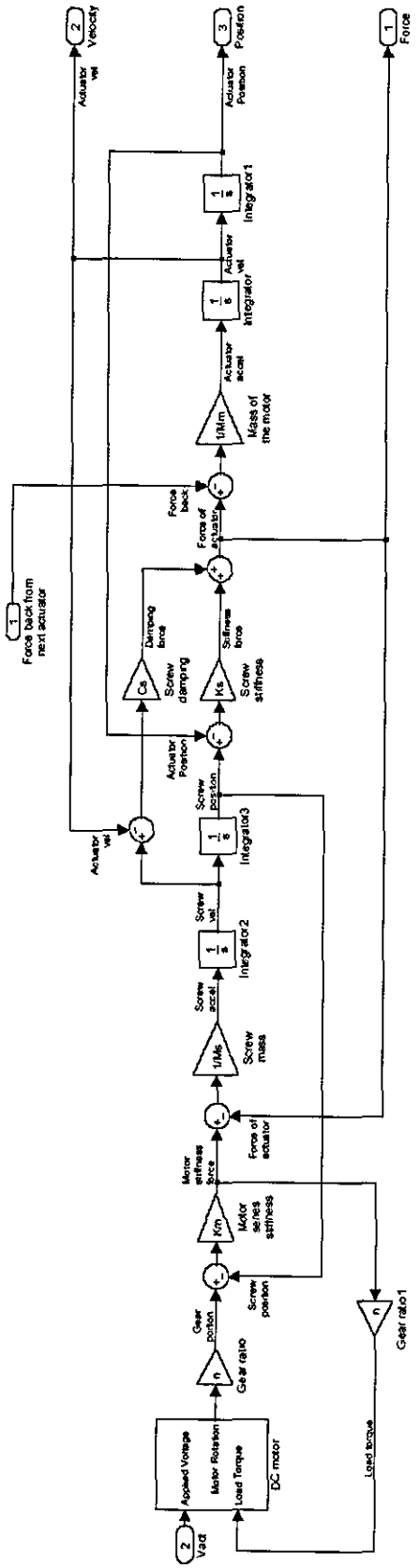


Fig. 3.14 Modularization model of the bottom actuator

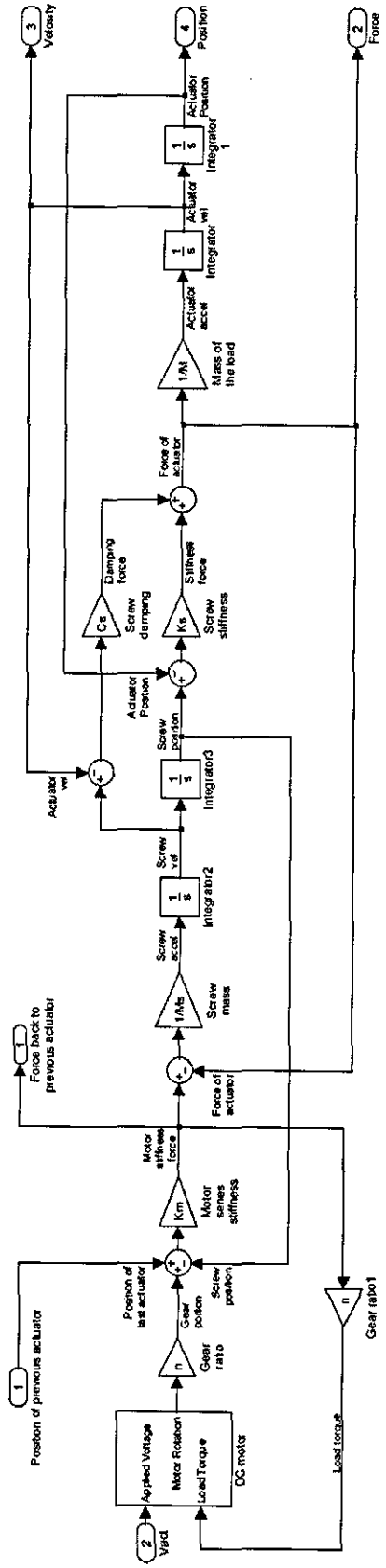


Fig. 3.15 Modularization model of the top actuator

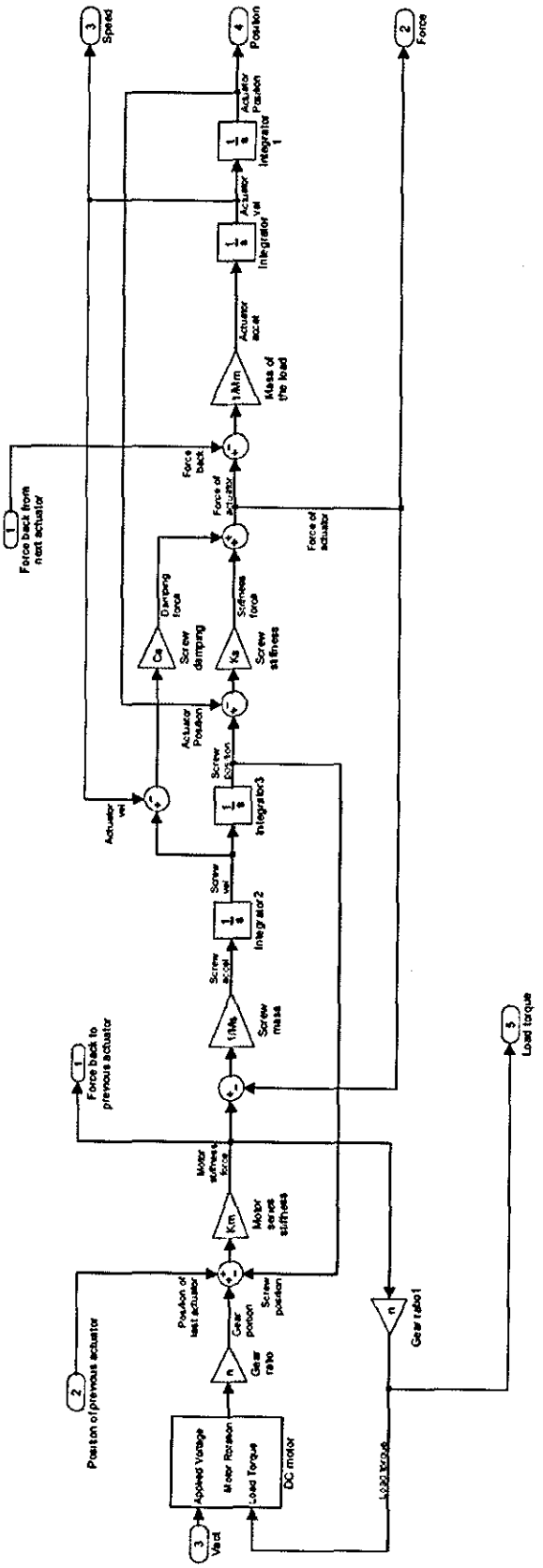


Fig. 3.16 Modularization model of the intermediate actuator

The modularized models of the three kinds of elements have been given in the figures from Fig. 3.14 to Fig. 3.16 respectively. Using the modularized models, it is straight forward to obtain models of series configurations by connecting them together. In each case, only one bottom and one top element can be included, and any number of intermediate elements (including zero) can be included between them. A model including three actuation elements is shown in Fig 3.17.

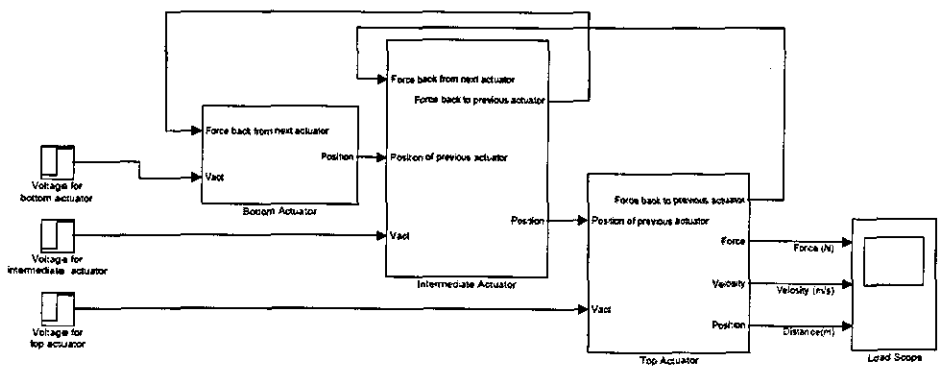


Fig. 3.17 Simulink model of a three in serial configuration

3.2.2 Modelling of parallel configuration

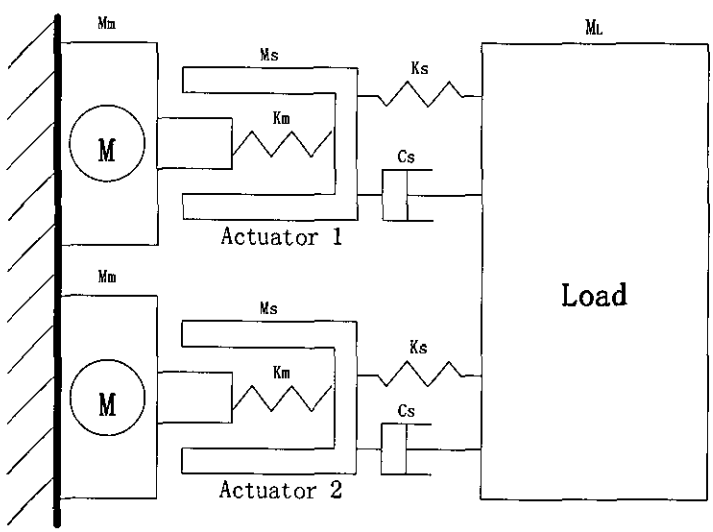


Fig. 3.18 Equivalent parallel configuration model

An equivalent model is firstly given in Fig. 3.18. Here the load is assumed to suffer no rotation in both healthy and faulty conditions (same as in grid structures). Three moving parts are considered in the free body diagram as shown in Fig. 3.19: the screw of actuator 1 (left top), the screw of actuator 2 (left bottom), and the load (right). Both motors are fixed so that they are not included in the free body diagram.

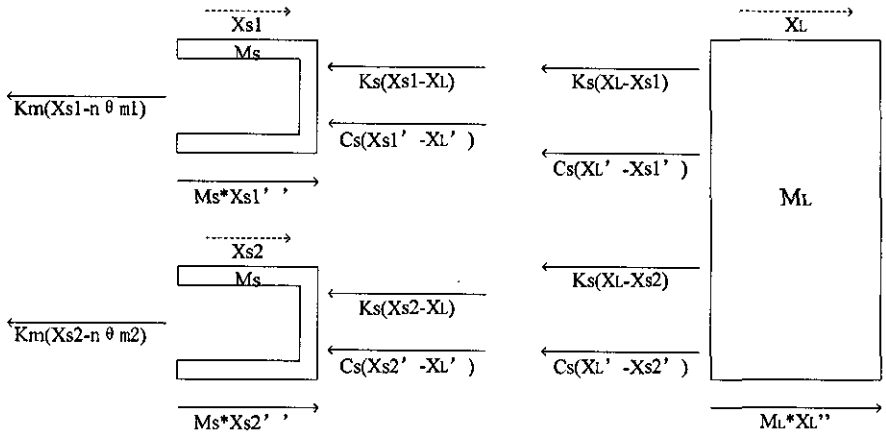


Fig. 3.19 Free body diagram of parallel configuration

Based on the Newton laws, following formulas can be got:

$$\begin{cases} M_s X_{s1}'' + (X_{s1} - \theta_{m1} n)K_m + (X_{s1} - X_L)K_s + (X_{s1}' - X_L')C_s = 0 \\ M_s X_{s2}'' + (X_{s2} - \theta_{m2} n)K_m + (X_{s2} - X_L)K_s + (X_{s2} - X_L)C_s = 0 \\ M_L X_L'' + (X_L - X_{s1})K_s + (X_L' - X_{s1}')C_s + (X_L - X_{s2})K_s + (X_L' - X_{s2}')C_s = 0 \end{cases} \quad (3.4)$$

A Simulink model of a two actuators in parallel is shown in Fig. 3.20. Both force outputs from actuators are added move the load. Since the ends of the two actuators are fixed together with the load, they share the same extension speed and distance, and the extension speed will be fed back to both actuators. Also more elements can be connected in one parallel configuration, but different from the series configuration, all of them are working equally so that they will be considered as the same kind of elements. When more elements are included in the parallel configuration, the force outputs of them will be added together to applied on the load and share the same speed and position which is same as the two in parallel configuration.

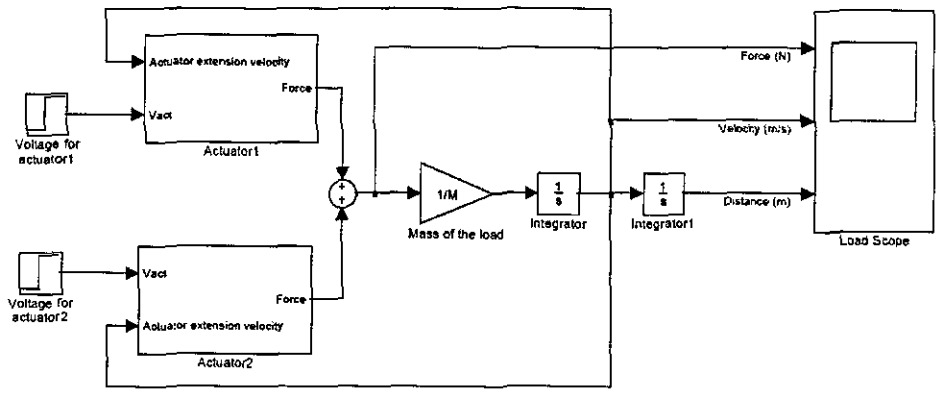


Fig. 3.20 Simulink model of a two in parallel configuration

3.2.3 Simulation results

First of all, the series and parallel configurations will be compared with the individual actuator. Only two elements are connected in both configurations and both actuators are applied with a same 36V step voltage at 0.1s. The simulation result of the series configuration is shown in Fig. 3.21 and the parallel one is shown in Fig. 3.22.

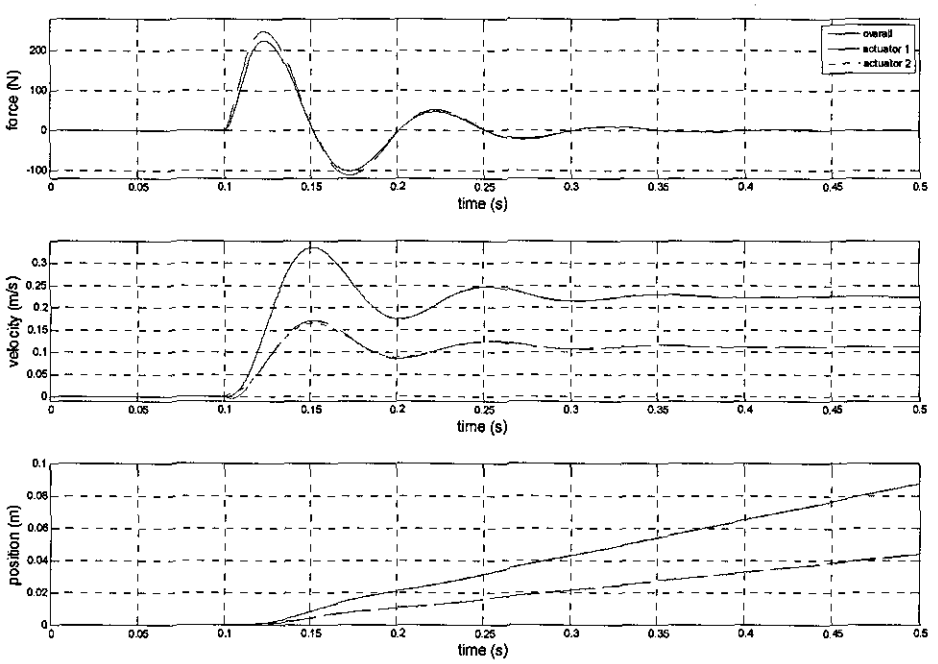


Fig. 3.21 Simulation result of the series configuration

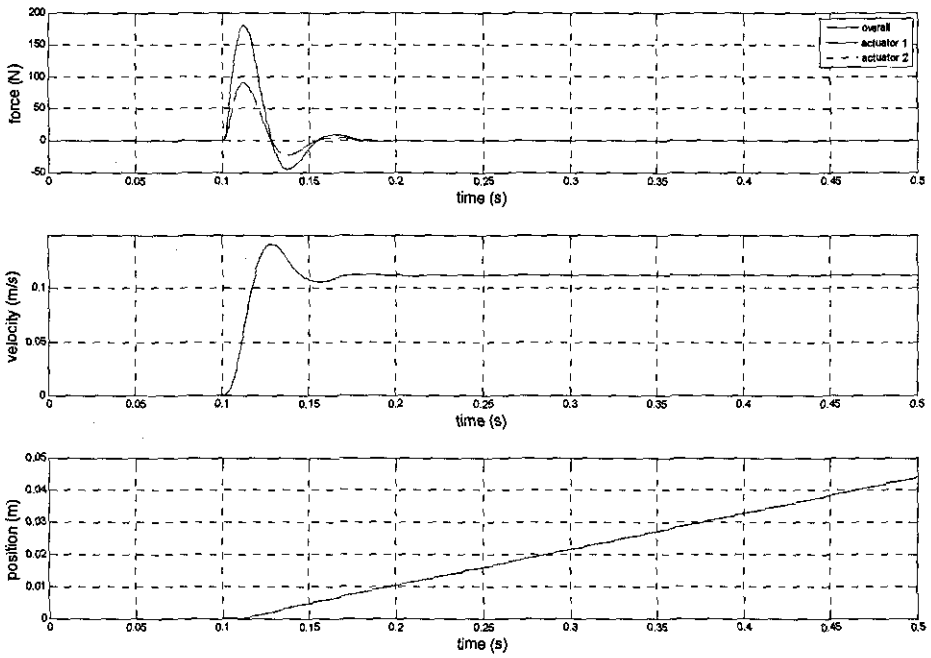


Fig. 3.22 Simulation result of the parallel configuration

The individual actuator's result has been shown in Fig. 3.9. The peak force output is about 150N, the steady velocity is about 0.11m/s at about 0.25s. Looking at the result of series configuration (Fig. 3.21), the force output of the overall configuration, actuator 1, and actuator 2 are almost equally. The velocity outputs of single elements are also almost equally and also equal to the individual result, but it takes longer time for the series configuration to reach the steady state at about 0.4s. The overall velocity is the sum of two elements' velocity. Same situation also can be found in position output. So it can be concluded that the two in series configuration can provide twice outputs at the velocity and the position, but the force output is remained same as the individual element. Looking at the result of parallel configuration (Fig. 3.22), the configuration and two elements share the same velocity and position outputs. The force output of the configuration is twice the sum of the elements' output, and this makes the load reach the steady state faster at about 0.2s. So it can be concluded that the two in parallel configuration provide twice the force output but share the same velocity and

position output with the single elements.

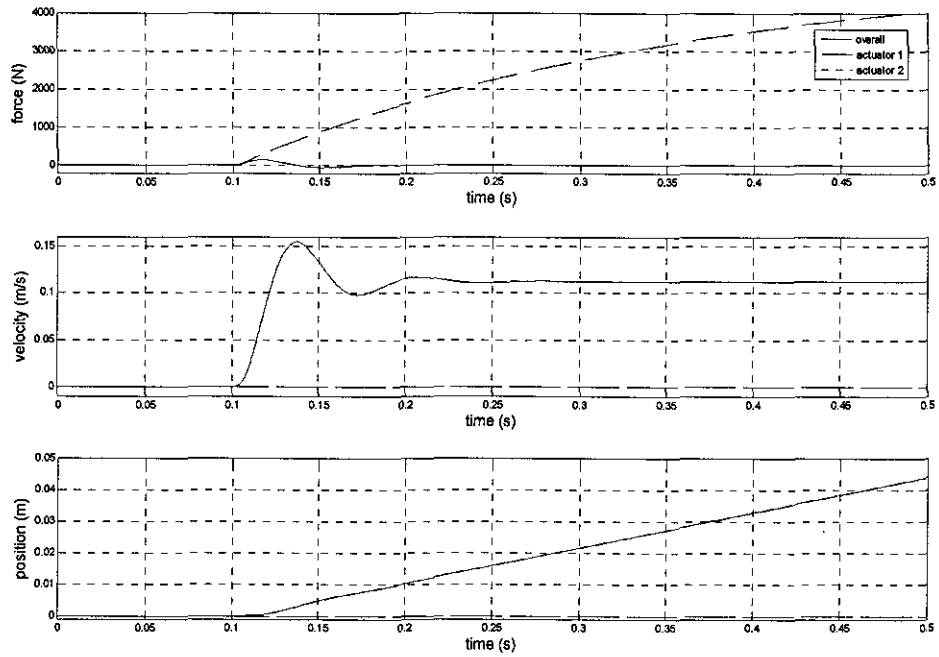


Fig. 3.23 Simulation result of the series configuration with one lockup

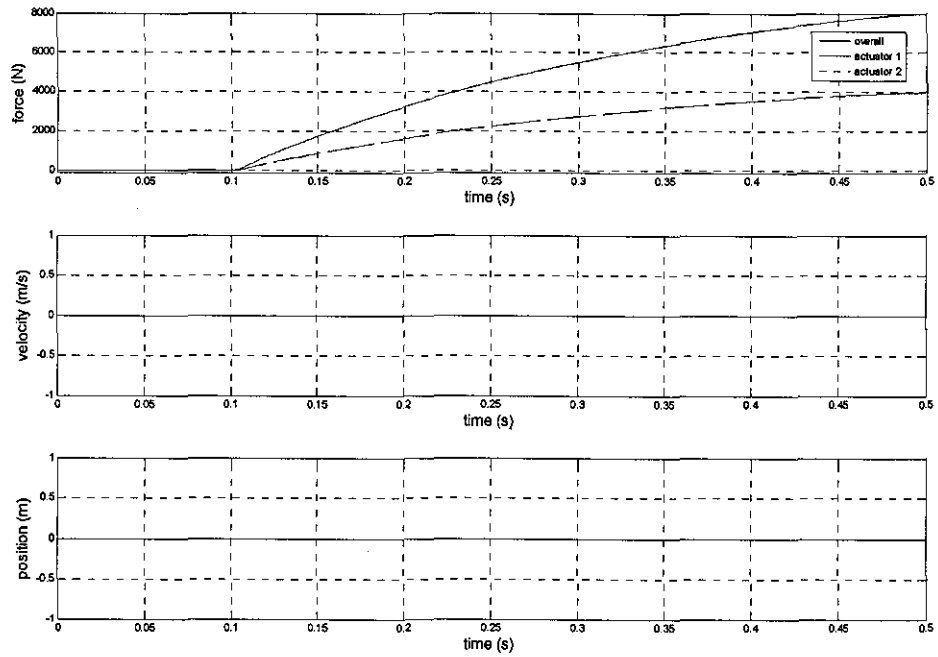


Fig. 3.24 Simulation result of the parallel configuration with one lockup

Then both structures are tested under faults injected situations. The Actuator

1 both configurations is considered faulty. Firstly, a travelling capability loss (lockup) situation is considered. The simulation results are shown in Fig. 3.23 (series configuration) and Fig. 3.24 (parallel configuration). When the actuator is locked up, it can not provide any output at either position or velocity, so that the force output is keep rising until it reaches the maximum output power of the motor inside the actuator. In the series configuration, although the Actuator 1 is locked up, the Actuator 2 still can provide a travelling capability, so that the overall performance is equal to the Actuator 2's performance which is also equal to a single actuator's performance. In the parallel configuration, both actuators are fixed together to share the same extension velocity and position, so that although only the Actuator 1 is locked up, the Actuator 2 neither can provide any output at either position or velocity. The whole parallel configuration is then locked up without any position output.

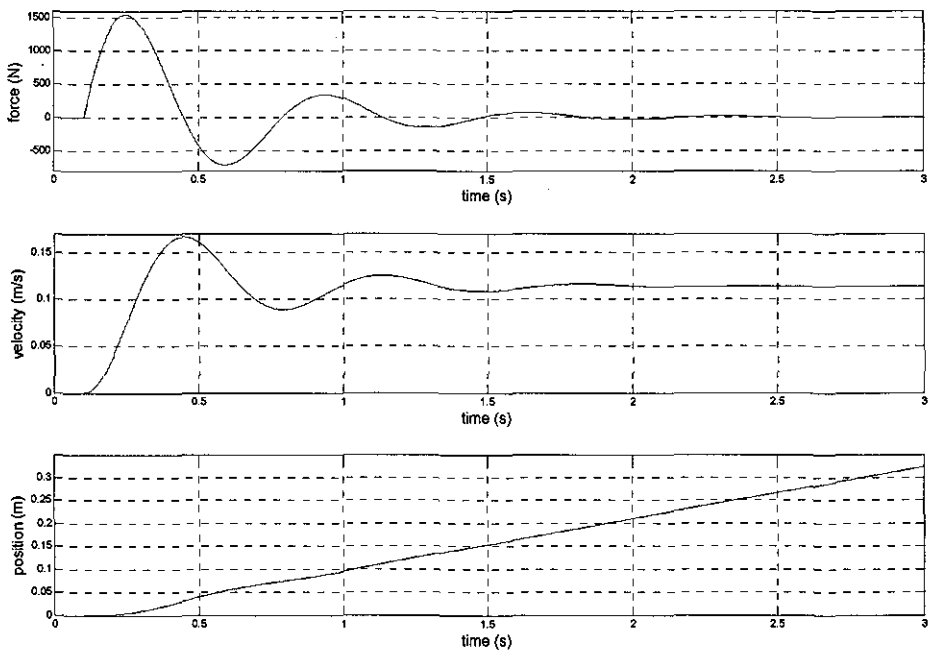


Fig. 3.25 Simulation result of the individual actuator with a load of 2000kg

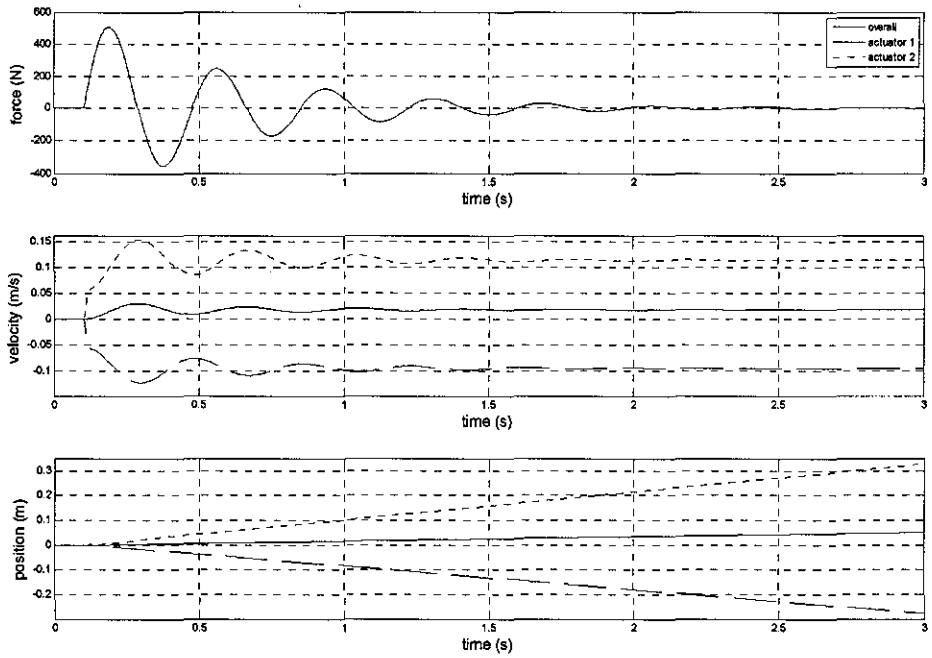


Fig. 3.26 Simulation result of a open circuit series configuration with a load of 2000kg

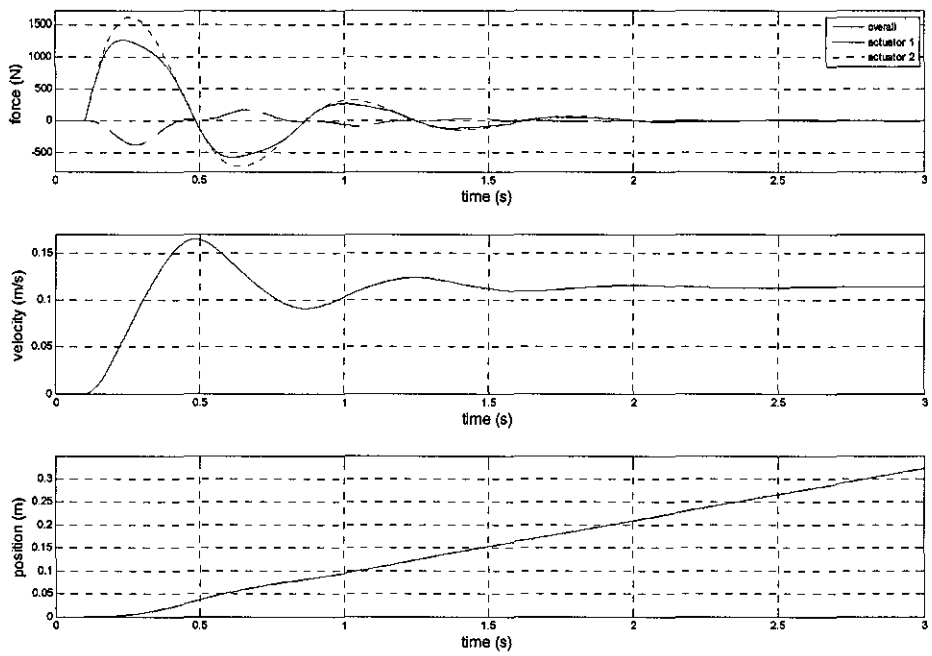


Fig. 3.27 Simulation result of a open circuit parallel configuration with a load of 2000kg

Now a force capability loss (open circuit) situation is considered. To have a

clearly look at the effect of the fault, the mass of the load is increased to 2000kg, which means the load is much harder to be moved. The performance of the individual actuator with a higher mass of load is firstly shown in Fig. 3.25. Result of the series configuration is shown in Fig. 3.26, and the parallel configuration's result is shown in Fig. 3.27. For the individual actuator, it is more difficult to push the high mass load than the low mass load. The peak force output is almost ten times bigger. The velocity output takes longer time to reach the steady state, but the steady state value is same (about 0.11m/s). In both configurations, only the Actuator 2 can generate force while the Actuator 1 can only be moved softly because of the open circuit fault. In the series configuration, the Actuator 2 can extend toward both sides, i.e. push the Actuator 1 back while push the load forward. When the load is chosen with a high mass, its inertia is higher than the Actuator 1's so that the Actuator 1 is pushed back with a higher velocity while the load is pushed forward with a lower velocity. The result can be found in Fig. 3.26. Although the Actuator 2 still can generate a velocity output (about 0.11m/s), the Actuator 1 is pushed back at the speed of 0.1m/s. The overall velocity is limited at a low value (about 0.01m/s) which means the load is hardly moved. In the parallel configuration (as seen in the Fig. 3.27), the situation is similar with the individual one. Both the Actuator 1 and the load are pushed forward by the Actuator 2. The Actuator 2 not only need to overcome the inertia of the load but also of the Actuator 1, so that a higher force is generated by the Actuator 2 and it takes a little bit longer to reach a steady state compared with the individual actuator's result.

By testing both configurations in healthy and faulty situation, it can be concluded that the series configuration can combine the velocity and position outputs together while the force output is remained same; the parallel configuration can combine the force output together while the velocity and position outputs are remained same. In the situation of faults,

the series configuration is more sensitive to elements' force loss situation while the parallel configuration is more sensitive to the elements' travelling loss situation. To overcome both faults, it is necessary to combine both configuration so that grid structures can be obtained.

3.3 Grid Structures

As discussed previously, the series configuration is more useful in a lockup situation (loss of travelling capability) while the parallel configuration is more useful in an open circuit situation (loss of force capability). A better structure should be able to accommodate both kinds of faults so that it need to combine both basic configurations. There are two ways to combine them. The first can be called series-in-parallel (SP in short), and the second can be called parallel-in-series (PS in short). A two-by-two structure is considered here, which is easier to analyze. Through a careful design, the final mechanical design can ensure no introduction of additional failure mechanisms (E.g. a rail will be engaged to support the weight of the load and ensure that no rotation of the load occurs, linkages will be used between the two moving blocks to avoid bump between two series elements). Such design can be seen at the hardware demonstrator which will be introduced in Chapter 6. Both mathematical expressions and Simulink models of both structures will be presented.

3.3.1 Modelling of parallel-in-series structure

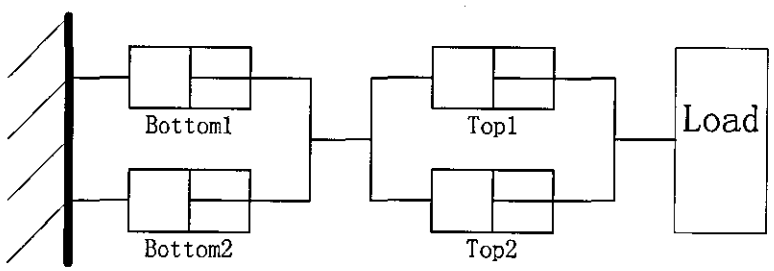


Fig. 3.28 A two-by-two parallel-in-series structure

A parallel-in-series structure is firstly introduced. It means connecting parallel configurations in series. In a two-by-two network, two groups of actuators, including two actuators connected in parallel each, are connected

in series. The structure is shown in the Fig. 3.28, and an equivalent model is shown in Fig. 3.29.

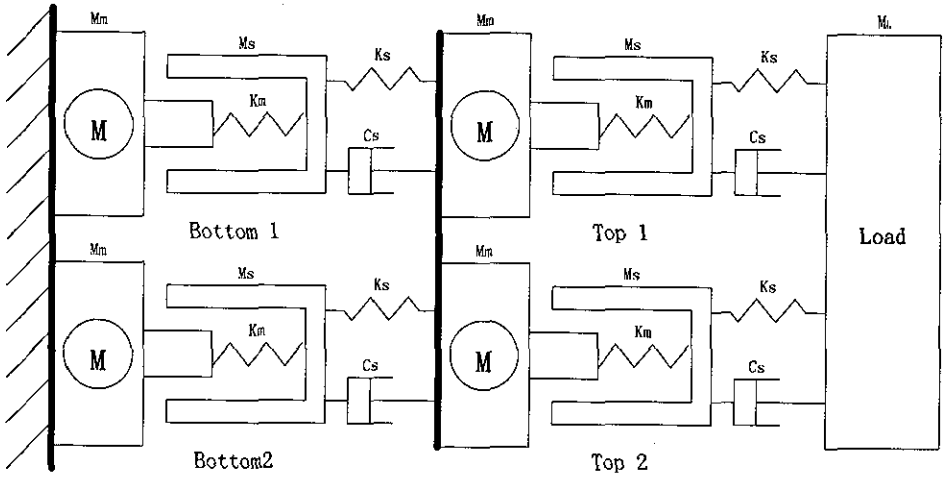


Fig. 3.29 Equivalent model of a two-by-two parallel-in-series structure

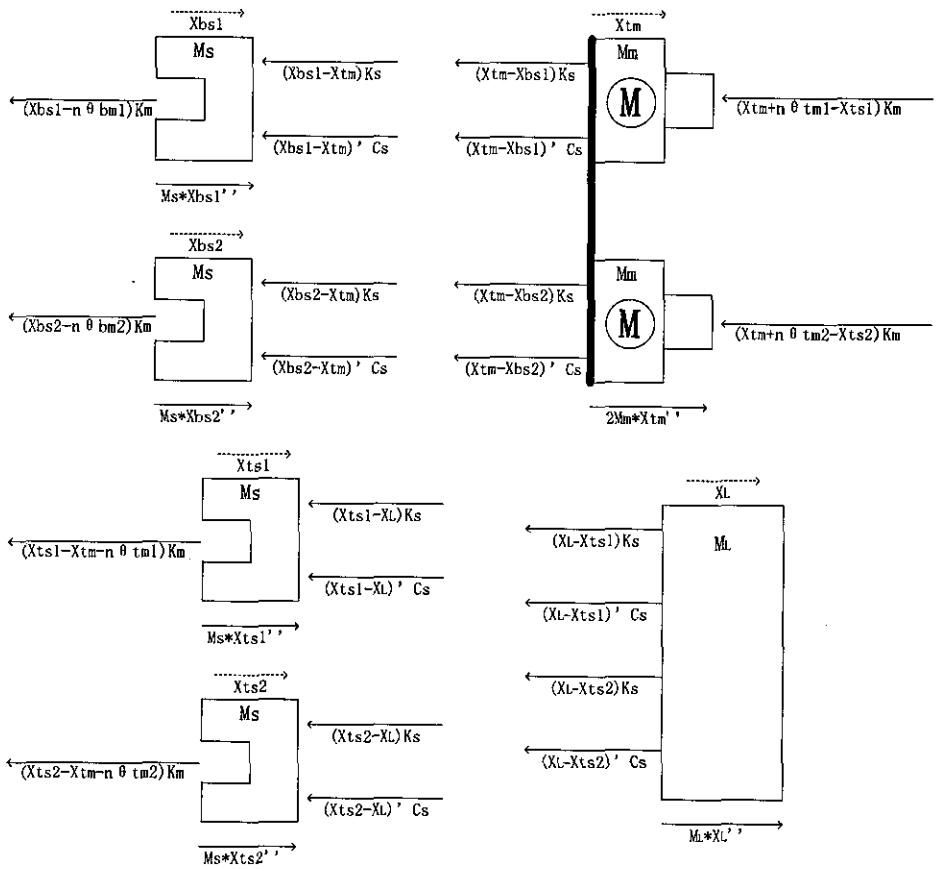


Fig. 3.30 Free body diagram of a two-by-two parallel-in-series structure

In this structure, the first two actuation elements are fixed together to work as bottom actuators and the other two are fixed together to work as top actuators in the series structures. Again a free body diagram (shown in Fig. 3.30) is used for analyse so that mathematical expressions can be obtained.

Here, the identifiers are remained same as used in the individual actuator model, while subscript 'b' and 't' express bottom and top respectively; subscript '1' and '2' express actuator 1 and 2 respectively. In the free body diagram, the moving parts in the model include: the screws of bottom actuators (top left), the two fixed motor of top actuators (top right), the screws of top actuators (bottom left), and the load (bottom right). Using the free body diagram, the mathematical expression of moving parts are shown in equations 3.5.

$$\begin{cases} M_s X''_{bs1} + (X_{bs1} - n\theta_{bm1})K_m + (X_{bs1} - X_{tm})K_s + (X'_{bs1} - X'_{tm})C_s = 0 \\ M_s X''_{bs2} + (X_{bs2} - n\theta_{bm2})K_m + (X_{bs2} - X_{tm})K_s + (X'_{bs2} - X'_{tm})C_s = 0 \\ 2m_m X''_{tm} + (X_{tm} - X_{bs1})K_s + (X'_{tm} - X'_{bs1})C_s + (X_{tm} - X_{bs2})K_s + (X'_{tm} - X'_{bs2})C_s \\ + (X_{tm} + n\theta_{tm1} - X_{ts1})K_m + (X_{tm} + n\theta_{tm2} - X_{ts2})K_m = 0 \\ M_s X''_{ts1} + (X_{ts1} - X_{tm} - n\theta_{tm1})K_m + (X_{ts1} - X_L)K_s + (X'_{ts1} - X'_L)C_s = 0 \\ M_s X''_{ts2} + (X_{ts2} - X_{tm} - n\theta_{tm2})K_m + (X_{ts2} - X_L)K_s + (X'_{ts2} - X'_L)C_s = 0 \\ M_L X''_L + (X_L - X_{ts1})K_s + (X'_L - X'_{ts1})C_s + (X_L - X_{ts2})K_s + (X'_L - X'_{ts2})C_s = 0 \end{cases} \quad (3.5)$$

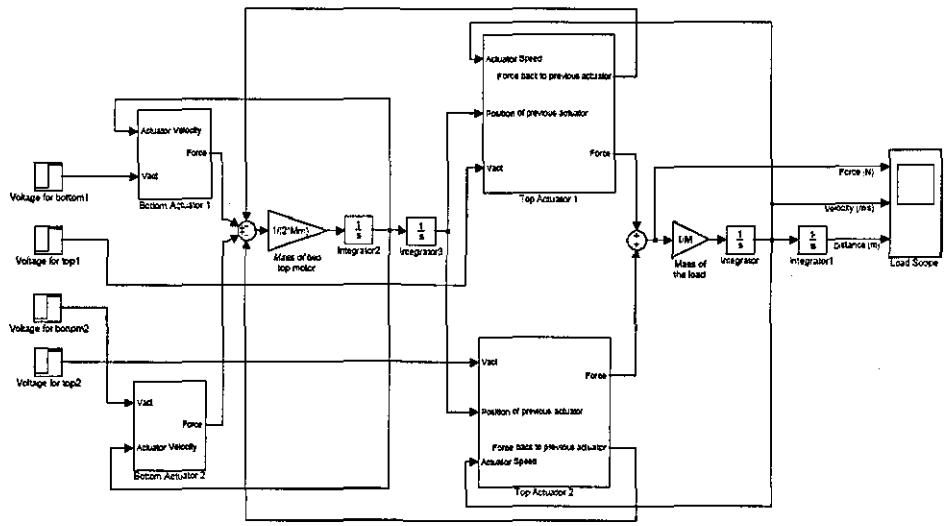


Fig. 3.31 Simulink model of a two-by-two parallel-in-series structure

Using the equations, a Simulink model of a two-by-two parallel-in-series structure can be shown in Fig. 3.31. The bottom and top actuator modules have introduced in the series configuration section.

3.3.2 Modelling of series-in-parallel structure

Another way to build up a grid structure is the series-in-parallel structure, which means placing series configurations in parallel. A two-by-two series-in-parallel structure is shown in Fig. 3.32.

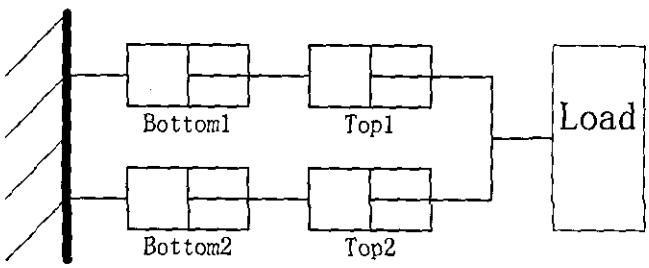


Fig. 3.32 A two-by-two series-in-parallel structure

Then an equivalent model of the structure is shown in Fig. 3.33, and a free body diagram is shown in Fig. 3.34.

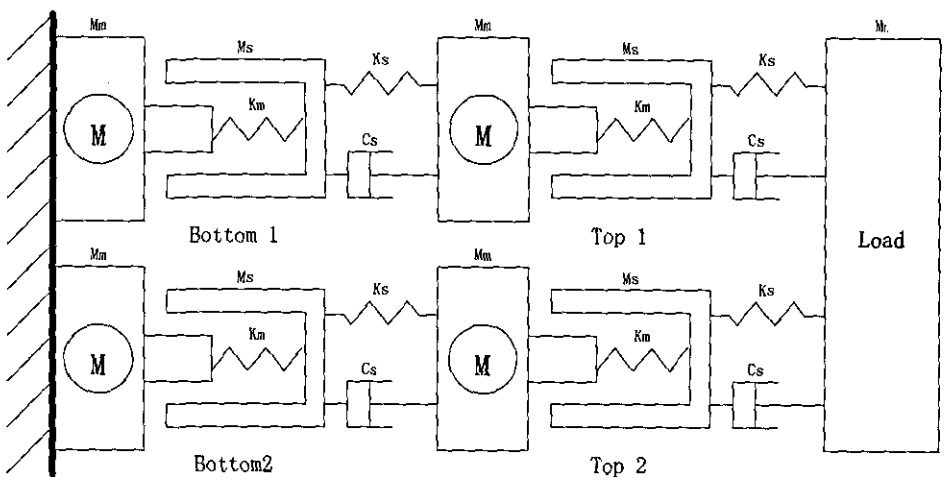


Fig. 3.33 Equivalent model of a two-by-two series-in-parallel structure

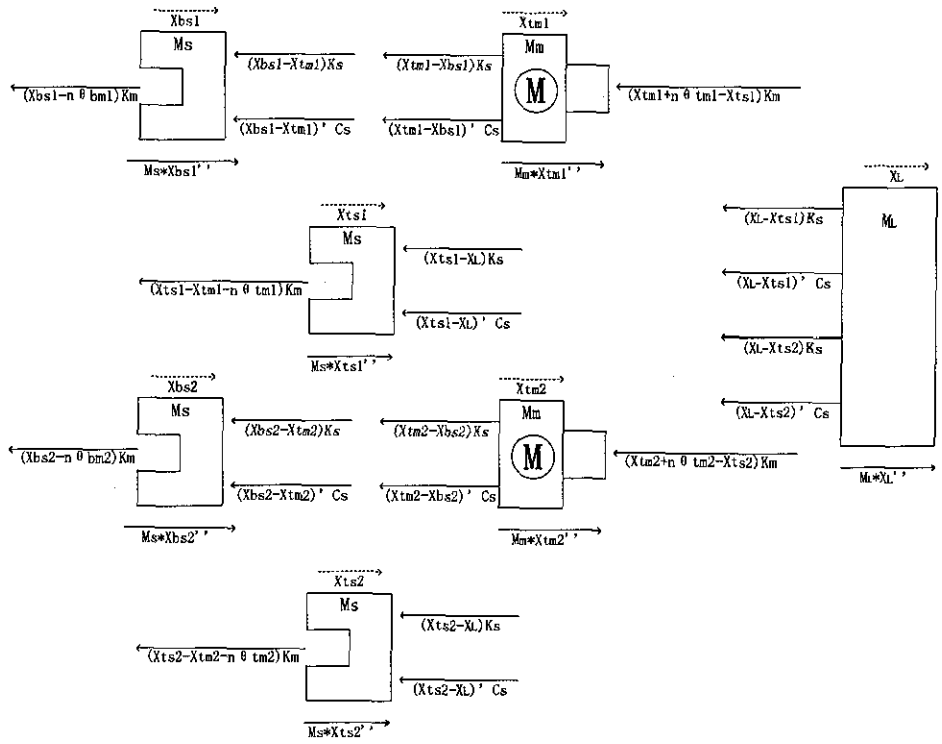


Fig. 3.34 Free body diagram of a two-by-two series-in-parallel structure

The moving parts in the free body diagram include the screws of bottom actuators, the motors of top actuators, the screws of top actuators, and the load. Then the mathematical expressions of these moving parts are given following:

$$\begin{cases} M_s X''_{bs1} + (X_{bs1} - n\theta_{bm1})K_m + (X_{bs1} - X_{tm1})K_s + (X'_{bs1} - X'_{tm1})C_s = 0 \\ M_m X''_{tm1} + (X_{tm1} - X_{bs1})K_s + (X'_{tm1} - X'_{bs1})C_s + (X_{tm1} + n\theta_{tm1} - X_{ts1})K_m = 0 \\ M_s X''_{ts1} + (X_{ts1} - X_{tm1} - n\theta_{tm1})K_m + (X_{ts1} - X_L)K_s + (X'_{ts1} - X'_L)C_s = 0 \\ M_s X''_{bs2} + (X_{bs2} - n\theta_{bm2})K_m + (X_{bs2} - X_{tm2})K_s + (X'_{bs2} - X'_{tm2})C_s = 0 \\ M_m X''_{tm2} + (X_{tm2} - X_{bs2})K_s + (X'_{tm2} - X'_{bs2})C_s + (X_{tm2} + n\theta_{tm2} - X_{ts2})K_m = 0 \\ M_s X''_{ts2} + (X_{ts2} - X_{tm2} - n\theta_{tm2})K_m + (X_{ts2} - X_L)K_s + (X'_{ts2} - X'_L)C_s = 0 \\ M_l X''_L + (X_L - X_{ts1})K_s + (X'_L - X'_{ts1})C_s + (X_L - X_{ts2})K_s + (X'_L - X'_{ts2})C_s = 0 \end{cases}$$

(3.6)

Based on these equations, the Simulink model can be shown in Fig. 3.35. Again, the modularized model of the bottom actuators are used in the series structures.

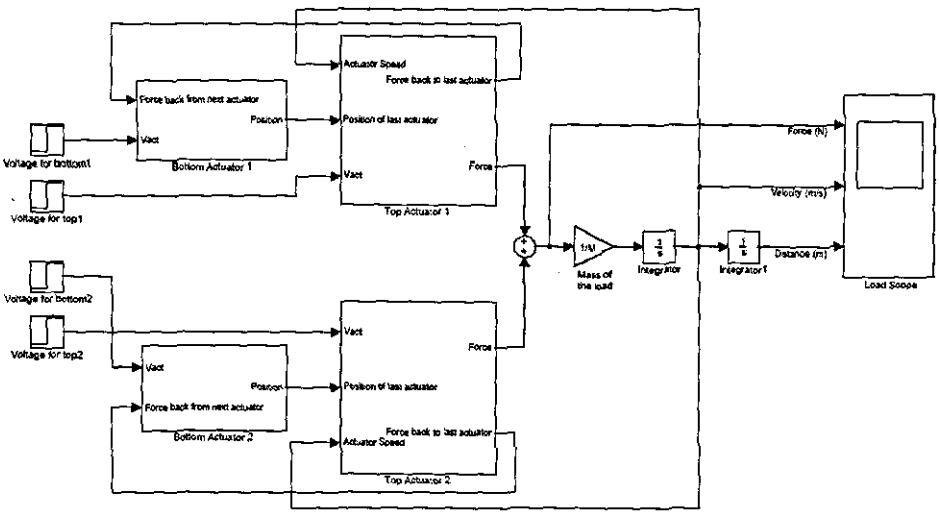


Fig. 3.35 Simulink model of a two-by-two series-in-parallel structure

3.3.3 Simulation results

Both structures will be tested under healthy and faulty situations. Healthy results are shown firstly.

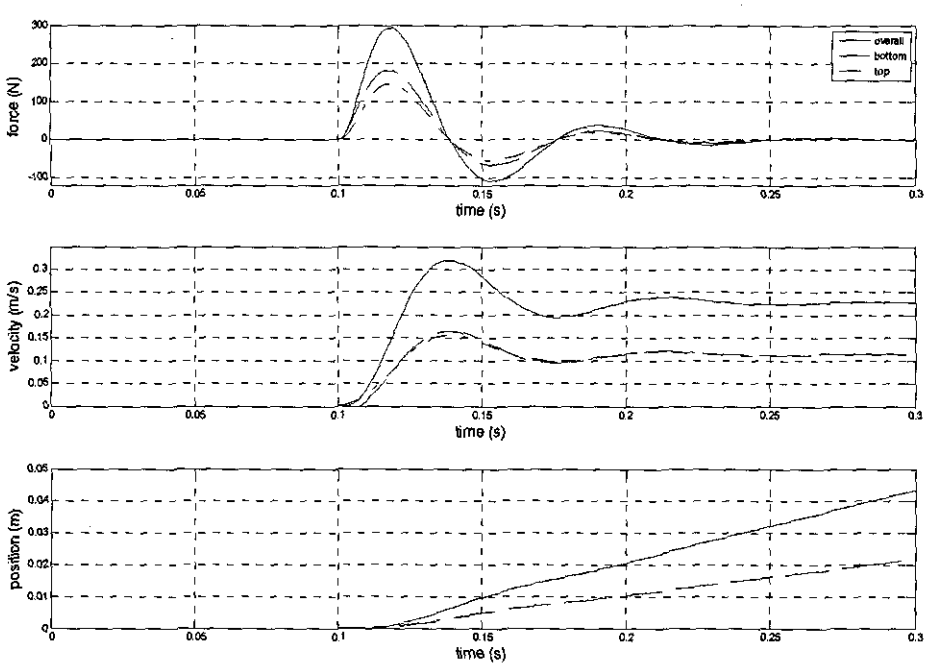


Fig. 3.36 Simulation result of the PS structure

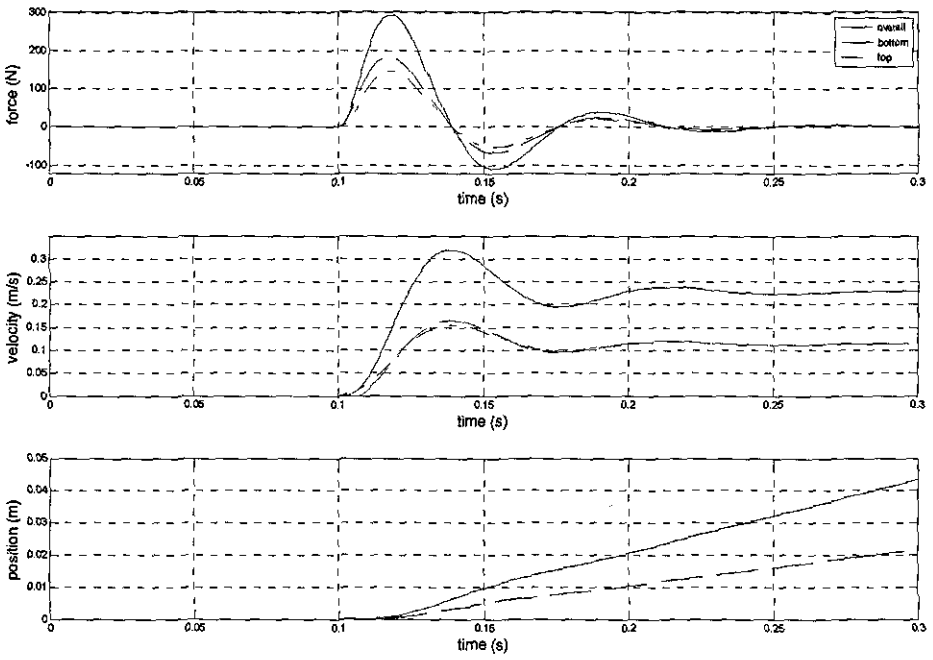


Fig. 3.37 Simulation result of the SP structure

Simulation result of the PS structure is shown in Fig. 3.36 and the SP structure is in Fig. 3.37. Both structures have very similar performance because they combine both series and parallel configurations so that they combine both features: double the force, velocity, and position outputs (in a two-by-two level). Both bottom and top elements perform very similar as well which is same in a series configuration.

Both structures are then tested under a lockup situation. Firstly, the Bottom 1 actuator is assumed locked up. Simulation result of the PS structure is shown in Fig. 3.38 and the SP structure is in Fig. 3.39 with each element's result shown in Fig. 3.40.

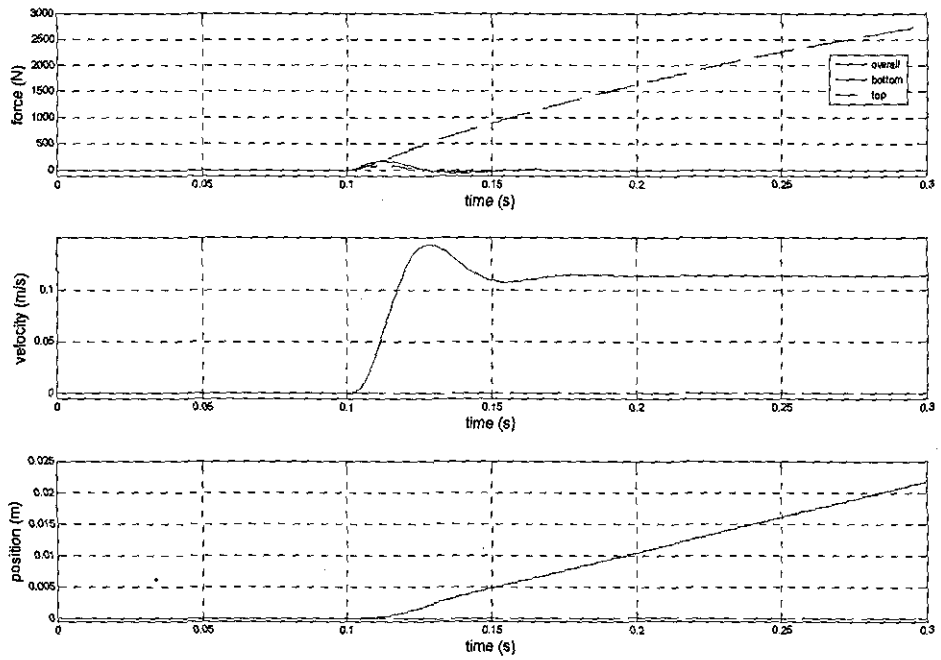


Fig. 3.38 Simulation result of the PS structure with Bottom 1 locked up

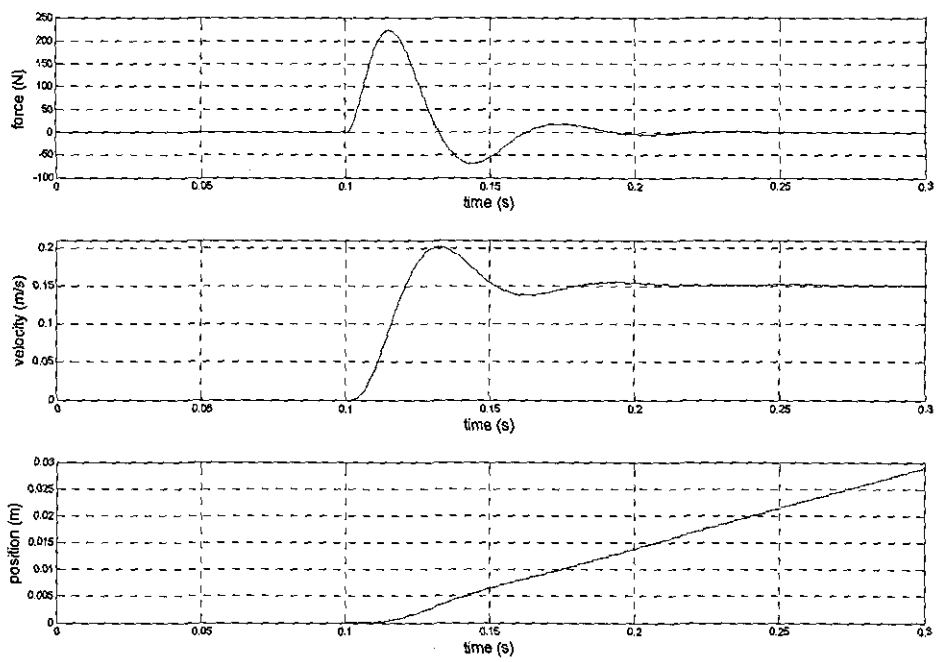


Fig. 3.39 Simulation result of the SP structure with Bottom 1 locked up

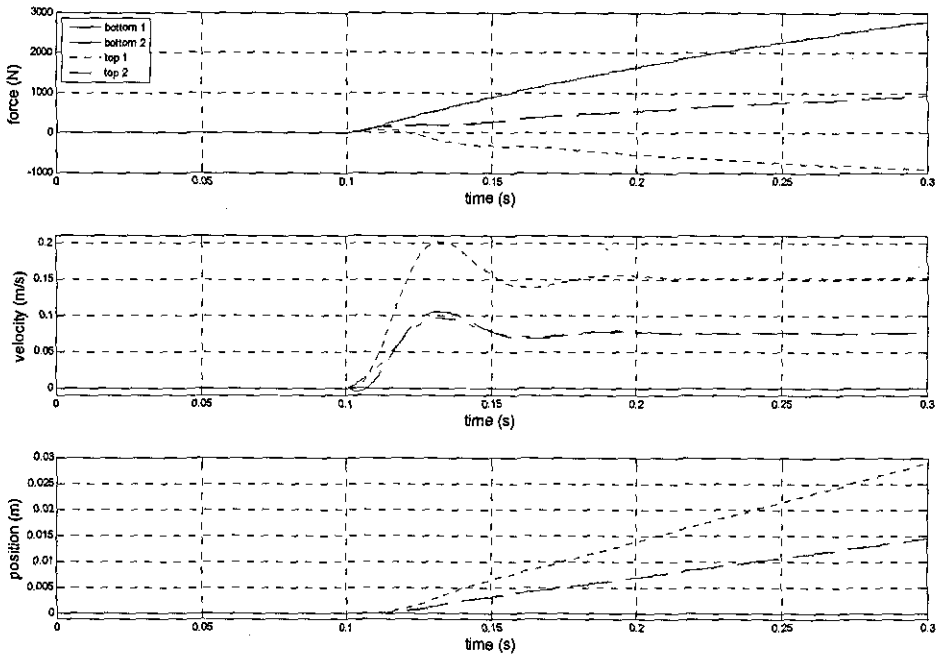


Fig. 3.40 Simulation result of elements in a SP structure with Bottom 1 locked up

When one element is locked up, it can be found that the SP structure can accommodate the fault better than the PS structure by providing a higher velocity output. In a PS structure, the two bottom elements are fixed together so that when one element is locked up, the other can not be moved either. Only the top parallel configuration can provide position and velocity output so that the overall performance loss about half of the outputs (0.11m/s at velocity, half of the healthy performance). In a SP structure, there is only one parallel configuration, so that when the Bottom 1 element is locked up, all other three elements still can move. The Top 1 element will be pushed by the other set of series configuration, so that the force output of the Top 1 occurs negative value. The overall velocity reaches about 0.15m/s which is about 3/4 of the healthy performance and is higher than the PS structure.

It is also possible to the situation with two elements locked up, but the performance will depend on the position of the faults. In a PS structure,

when one element locked up it will cause the failure of the other element in the same parallel configuration, so that if the two faulty elements happen in the same parallel configuration it will not cause a failure of the whole structure, but if happen in different parallel configurations it will make the whole structure locked up. In a SP structure, the situation is different. It can not accept that two faults happen in the same series configuration because this will make the series configuration locked up and make the whole structure locked up, but it can accommodate two faults separated in different series configurations. Comparing both structures, the PS structure can only accommodate two situations with two lockup faults (two in bottom parallel configuration and two in top), while the SP structure can accommodate four situations (one in each series configuration). It can be concluded that the PS structure is more sensitive to the lockup fault (travelling loss) than the SP structure. This is because the PS structure combines two parallel configurations while the SP structure only combines one, and the parallel configuration is very sensitive to the travelling loss.

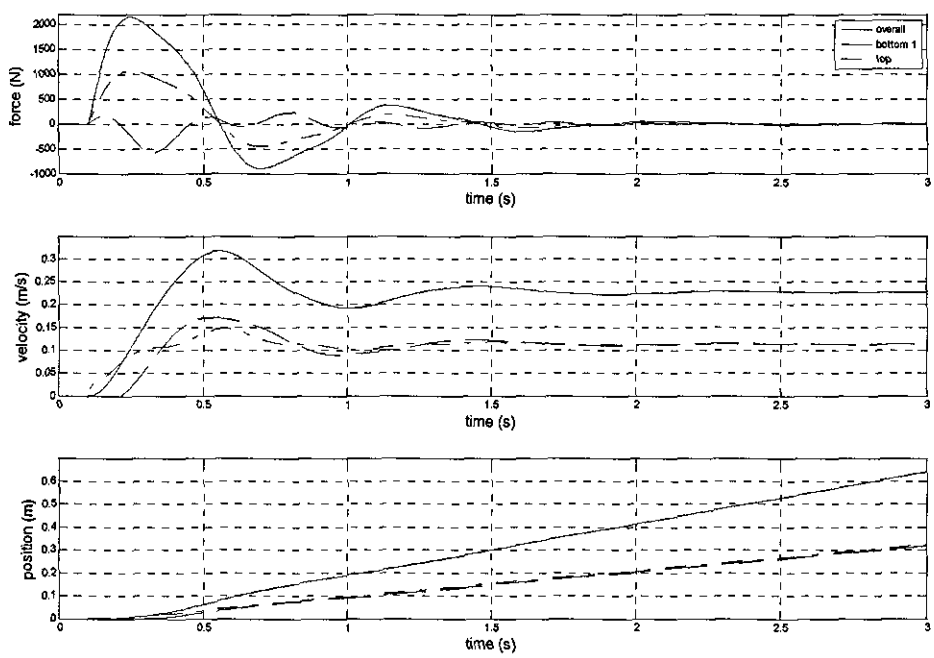


Fig. 3.41 Simulation result of the PS structure with Bottom 1 open circuit

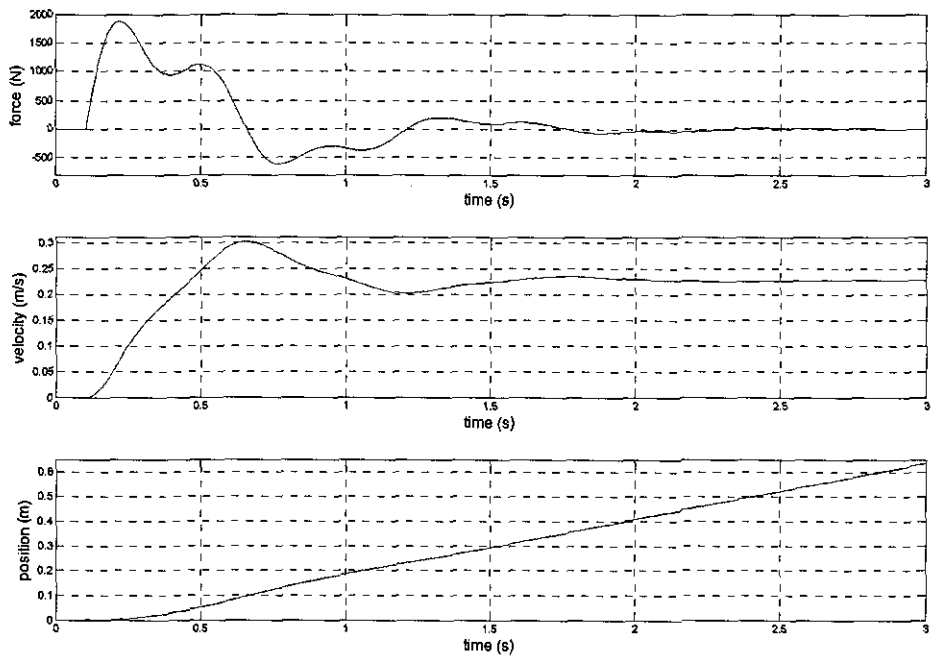


Fig. 3.42 Simulation result of the SP structure with Bottom 1 open circuit

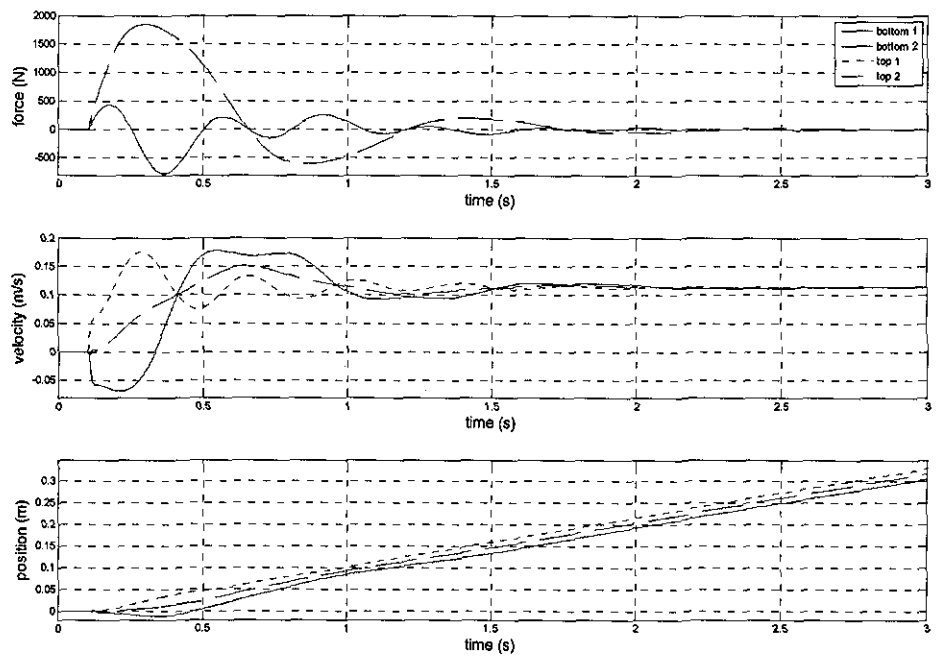


Fig. 3.43 Simulation result of elements in a SP structure with Bottom 1 open circuit

The open circuit fault is then considered. The Bottom 1 element is assumed as the faulty element again and the mass of the load is increased to 2000kg to make the effort look more clearly.

Simulation result of the PS structure is shown in Fig. 3.41 and the SP structure is in Fig. 3.42 with each element's result shown in Fig. 3.43. Comparing the overall performance, both structures are very similar with a full velocity output (same as the healthy performance). The open circuit fault is accommodated by the parallel configuration which means the Bottom 1 element is pushed by other element(s). From this point of view, the open circuit fault is more easily to be accommodated than the lockup fault.

When two open circuit faults happen, the result is depended on the position of the faults. For the PS structure, it will make a failure when both faults happen in the same parallel configuration, while the SP structure will fail when faults are separated into different series configuration. From this point of view, it can be concluded that the SP structure is more sensitive to the open circuit fault (force loss) than the PS structure. This is because the PS structure combines one series configuration while the SP structure combines two, and the series configuration is very sensitive to the force loss.

Considering that the open circuit (force loss) is more easily to be accommodated than the lockup (travelling loss), the two-by-two series-in-parallel structure is a better structure. Hence it will be used for the control design studies which will be introduced in Chapter 4.

3.4 Chapter Summary

In this chapter, a set of Simulink models has been built using electro-mechanical actuation technique. Using the single element, the series and the parallel configurations are modelled. The simulation studies show that the series configuration is more effective at position and velocity outputs while the parallel configuration is more effective at force output. The open-loop tests also show that the series configuration is more sensitive to force loss (open circuit) faults while the parallel configuration is more sensitive to travelling loss (lockup) faults. To comprise both advantages, a pair of two-by-two grid structures (parallel-in-series and series-in-parallel) is modelled. Both structures show similar performance at all three outputs but under open-loop faults injected situations, the PS structure is more sensitive to lockup fault while the SP structure is more sensitive to the open loop fault because they comprise different number of series and parallel configurations. Considering the open loop fault is more easily to be accommodated, the two-by-two SP structure will be used for the control design, which will be introduced in Chapter 4.

4. CONTROL STUDIES

Two kinds of two-by-two grid structures have been built up in Simulink and some open-loop tests have been done in both healthy and faults injected situations. Closed-loop control issues will be discussed in this chapter. As concluded at the end of Chapter 3, a two-by-two series-in-parallel structure is used for the control design. Position control is considered. By closing the loop, the HRA is controlled to track a step signal.

In this chapter, a robust analysis is firstly given with an introduction of two control structures. Then, both classical and optimal control algorithms will be applied for control design. A proportional controller is first designed as a benchmark, which means all controllers are designed to have similar performance. All controllers will be tested in a healthy situation in this chapter. The performances of each controller under faults injected situations will be presented later in Chapter 5.

4.1 Robust Analysis and Control Structures

Before starting the control design, it is necessary to do some analysis about the mechanical structure used for a HRA. The analysis will be based on the Simulink model and state-space model as well, so that finding out the state-space model is the first step to do.

4.1.1 State-space model of a two-by-two SP structure

The structure has been introduced in Section 3.3.2. Mathematical expression for the mechanical parts is given in equation 3.6. Same as the individual actuator, the state-space model combines both electrical and mechanical states. There are four elements included in this grid structure and here all of them are assumed identical, and driven by same voltage input. A twenty-six state state-space model then can be got in the form of $\dot{x} = Ax + Bu,\ \ y = Cx + Du$. The applied voltage is chosen as the input and the position of the overall structure is chosen as the output. The states are chosen same as individual ones for each element, whose meaning was shown in Table. 3.4, but here subscripts have to be used to identify different elements included in the structure, where 'b' means bottom; 't' means top; and '1' and '2' express actuator 1 and 2 respectively. The state vector is chosen as:

$$x = [\, i_{b1}\ \ \dot{\theta}_{bm1}\ \ \theta_{bm1}\ \ \dot{X}_{bs1}\ \ X_{bs1}\ \ \dot{X}_{tm1}\ \ X_{tm1}\ \ i_{t1}\ \ \dot{\theta}_{tm1}\ \ \theta_{tm1}\ \ \dot{X}_{ts1}\ \ X_{ts1}\ \ i_{b2}\ \ \dot{\theta}_{bm2}\ \ \theta_{bm2}\ \ \dot{X}_{bs2}\ \ X_{bs2}$$

$$\dot{X}_{tm2}\ \ X_{tm2}\ \ i_{t2}\ \ \dot{\theta}_{tm2}\ \ \theta_{tm2}\ \ \dot{X}_{ts2}\ \ X_{ts2}\ \ \dot{X}_{L}\ \ X_{L}\ ]^{T}.$$

In the SP structure, the state matrix A will comprise both the model of each individual actuator element and the coupling between different elements, thus it can be partitioned into a 4x4 matrix based on the physical meaning with elements b1, b2 (the bottom elements), t1, t2 (the top elements) in the

diagonal, and their coupling relations in the off-diagonal positions.

$$A=\begin{bmatrix} b1 & b1*t1 & b1*b2 & b1*t2 \\ t1*b1 & t1 & t1*b2 & t1*t2 \\ b2*b1 & b2*t1 & b2 & b2*t2 \\ t2*b1 & t2*t1 & t2*b2 & t2 \end{bmatrix}_{4*4} \quad \text{or,} \quad A=\begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{bmatrix}_{4*4}$$

The sub-matrices A_{ij} for all $i=1,2,3,4$ and $j=1,2,3,4$ are given as follows:

$$A_{11}=\begin{bmatrix} -\frac{R_{arm}}{L_{arm}} & -\frac{K_e}{L_{arm}} & 0 & 0 & 0 & 0 & 0 \\ \frac{K_t}{J_m} & -\frac{C_m}{J_m} & -\frac{n^2K_m}{J_m} & 0 & \frac{nK_m}{J_m} & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{nK_m}{M_s} & -\frac{C_s}{M_s} & -\frac{K_s+K_m}{M_s} & \frac{C_s}{M_s} & \frac{K_s}{M_s} \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{C_s}{M_L} & \frac{K_s}{M_L} & -\frac{C_s}{M_L} & -\frac{K_m+K_s}{M_L} \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}_{7*7},$$

$$A_{12}=\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{nK_m}{M_m} & 0 & \frac{K_m}{M_m} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{7*5}, \quad A_{13}=0_{7*7},\ A_{14}=0_{7*7},$$

$$A_{21}=\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{nK_m}{J_m} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{K_m}{M_s} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{5*7},$$

$$A_{22}=\begin{bmatrix} -\frac{R_{arm}}{L_{arm}} & -\frac{K_e}{L_{arm}} & 0 & 0 & 0 \\ \frac{K_t}{J_m} & -\frac{C_m}{J_m} & -\frac{n^2K_m}{J_m} & 0 & \frac{nK_m}{J_m} \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \frac{nK_m}{M_s} & -\frac{C_s}{M_s} & -\frac{K_m+K_s}{M_s} \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}_{5*5}, \quad A_{23}=0_{5*7},$$

$$A_{24}=\begin{bmatrix}0&0&0&0&0&0&0\\0&0&0&0&0&0&0\\0&0&0&0&0&0&0\\0&0&0&0&0&\frac{C_s}{M_s}&\frac{K_s}{M_s}\\0&0&0&0&0&0&0\end{bmatrix}_{5*7},\quad A_{31}=0_{7*7},\ A_{32}=0_{7*5},$$

$$A_{33}=\begin{bmatrix}-\frac{R_{arm}}{L_{arm}}&-\frac{K_e}{L_{arm}}&0&0&0&0&0\\\frac{K_t}{J_m}&-\frac{C_m}{J_m}&-\frac{n^2K_m}{J_m}&0&\frac{nK_m}{J_m}&0&0\\0&1&0&0&0&0&0\\0&0&\frac{nK_m}{M_s}&-\frac{C_s}{M_s}&-\frac{K_s+K_m}{M_s}&\frac{C_s}{M_s}&\frac{K_s}{M_s}\\0&0&0&1&0&0&0\\0&0&0&\frac{C_s}{M_L}&\frac{K_s}{M_L}&-\frac{C_s}{M_L}&-\frac{K_m+K_s}{M_L}\\0&0&0&0&0&1&0\end{bmatrix}_{7*7},$$

$$A_{34}=\begin{bmatrix}0&0&0&0&0&0&0\\0&0&0&0&0&0&0\\0&0&0&0&0&0&0\\0&0&0&0&0&0&0\\0&0&0&0&0&0&0\\0&0&-\frac{nK_m}{M_m}&0&-\frac{K_m}{M_m}&0&0\\0&0&0&0&0&0&0\end{bmatrix}_{7*7},\quad A_{41}=0_{7*7},$$

$$A_{42}=\begin{bmatrix}0&0&0&0&0\\0&0&0&0&0\\0&0&0&0&0\\0&0&0&0&0\\0&0&0&0&0\\0&0&0&\frac{C_s}{M_L}&\frac{K_s}{M_L}\\0&0&0&0&0\end{bmatrix}_{7*5},\quad A_{43}=\begin{bmatrix}0&0&0&0&0&0&0\\0&0&0&0&0&0&-\frac{nK_m}{J_m}\\0&0&0&0&0&0&0\\0&0&0&0&0&0&\frac{K_m}{M_s}\\0&0&0&0&0&0&0\\0&0&0&0&0&0&0\\0&0&0&0&0&0&0\end{bmatrix}_{7*7},$$

$$A_{44}=\begin{bmatrix}-\frac{R_{arm}}{L_{arm}}&-\frac{K_e}{L_{arm}}&0&0&0&0&0\\\frac{K_t}{J_m}&-\frac{C_m}{J_m}&-\frac{n^2K_m}{J_m}&0&\frac{nK_m}{J_m}&0&0\\0&1&0&0&0&0&0\\0&0&\frac{nK_m}{M_s}&-\frac{C_s}{M_s}&-\frac{K_s+K_m}{M_s}&\frac{C_s}{M_s}&\frac{K_s}{M_s}\\0&0&0&1&0&0&0\\0&0&0&\frac{C_s}{M_L}&\frac{K_s}{M_L}&-\frac{2C_s}{M_L}&-\frac{2K_s}{M_L}\\0&0&0&0&0&1&0\end{bmatrix}_{7*7}$$

and the corresponding B,C matrices are:

$$B=\left[\frac{1}{L_{arm}}\ 0\ 0\ 0\ 0\ 0\ 0\ \frac{1}{L_{arm}}\ 0\ 0\ 0\ 0\ \frac{1}{L_{arm}}\ 0\ 0\ 0\ 0\ 0\ 0\ \frac{1}{L_{arm}}\ 0\ 0\ 0\ 0\ 0\ 0\right]^T$$

$$C=[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1]$$

In the A matrix, sub-matrices $A_{14}, A_{23}, A_{32}, A_{41}$ describe relationships between uncoupling elements b1 and t2 as well as b2 and t1 respectively. Elements b1 and b2 are fixed to the same end, but they both can move independently, so that sub-matrices, A_{13} and A_{31}, describing relationships between elements b1 and b2, are also zero. Other sub-matrices are non-zero describing individual elements or coupling relationships between different elements. All elements are assumed identical and all parameters are remained the same as the individual actuator which are shown in Table. 3.3.

4.1.2 Open loop robust analysis

Before starting the control design process, it is useful to do some robust analyses based on the state-space model using Robust Control Toolbox of MATLAB. Robust Control Toolbox is a collection of functions and tools that help to analyse and design control systems with uncertain elements, which are at the heart of robust control. Model uncertainty arises when system gains or other parameters are not precisely known, or can vary over a given range. On the other hand, unmodelled dynamics also introduce uncertainty into existing models, which may affect not only values but also the shapes of models' frequency response in some specific frequency ranges.

In the simulated HRA system, multiple actuation elements are included which are assumed to be identical. But in the real world, it is impossible to have identical elements, and so the parameters of different elements could vary over a given range. This is the first uncertainty that needs to be

considered and analysed. On the other hand, a more important uncertainty to a HRA system is the changes in dynamics when faults are injected to some of the elements. From the view of fault tolerant control, the changes can be accommodated by reconfiguring the control structure. But from the view of robust control, a fixed robust controller is needed to accommodate changes or uncertainties of dynamics. In this section, both parameter and dynamic uncertainties will be analysis based on the two-by-two SP HRA state-space model.

4.1.2.1 Healthy situation

A healthy situation is considered firstly, which means a HRA with identical elements and without any faults injected. A Bode diagram based on the state-space model will be given in Fig. 4.1.

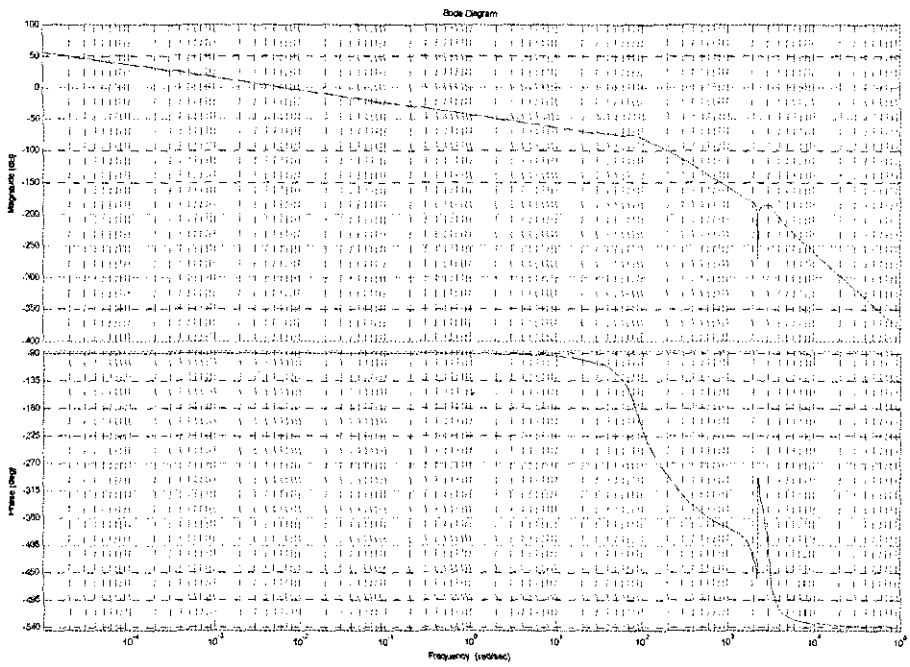


Fig. 4.1 Bode diagram of a HRA under healthy situation

The figure illustrates the frequency performance from 10^{-5} Hz to 10^{5} Hz. From this performance, a proportional controller is then going to be designed (in Section 4.2) to raise the low frequency gain. The close-loop bandwidth is then can reach about 1Hz. According to this figure, some uncertainties will be added into the model to test the difference from the healthy situation. If a big difference exists, a more robust controller will be needed.

4.1.2.2 Parameter uncertainties

As introduced, there are two kinds of uncertainties, parameter and dynamic. The parameter uncertainties describe some uncertain parameters in a given model structure. The reason for such an uncertainty includes error or approximate estimate during the measuring of parameters, parameter variety due to nonlinearities or changes in the operating conditions, and so on. Because of these reasons, it is almost impossible to get exactly accurate parameters in a simulation model. Sometimes, error in the parameters may make a big difference from the view of controller design so that it is important to do such an analysis before applying the designed controller based on a simulation model to the real application.

In the HRA model, there are four actuation elements included assumed to be identical. In a real world, the elements can not be identical because of the parameter uncertainty, although their structures are same. In this analysis, 20% (from -10% to 10%) uncertainty is added into each parameter related in the model. The frequency performance from 10^{-5} Hz to 10^{5} Hz including the parameter uncertainties is illustrated in the next figure.

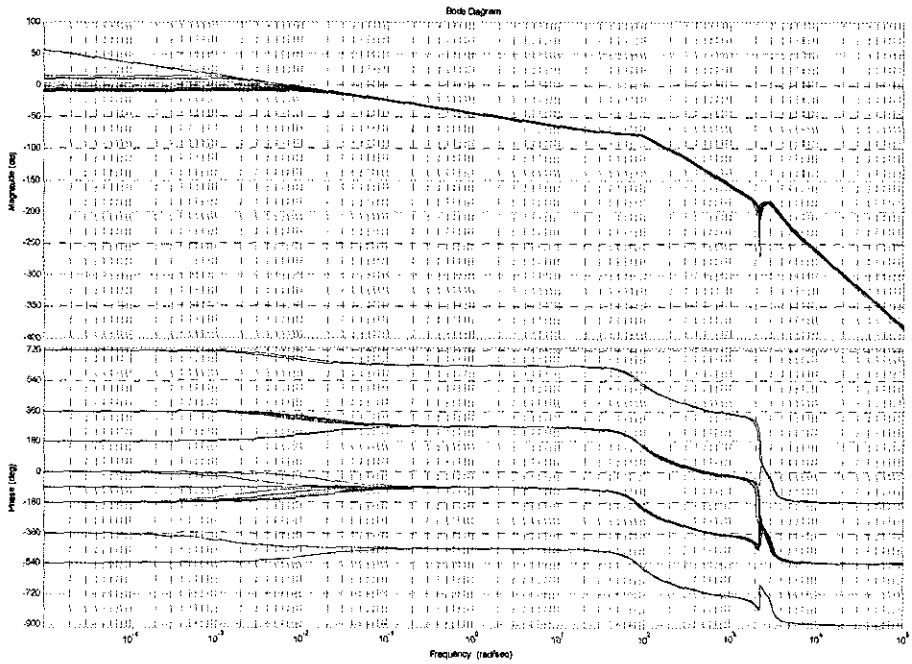


Fig. 4.2 Bode diagram of a HRA with 20% parameter uncertainty

In the figure, the red line presents the original system and all blue lines present systems with parameter uncertainties. The difference is bigger at frequency lower than 0.1rad/s, especially at very low frequency. But for a controller designed with a bandwidth at about 1Hz, the very low frequency part is not very important, so that it can be said that the parameter uncertainty is not very important for the control design.

4.1.2.3 Dynamic uncertainties

Comparing with the parameter uncertainties, dynamic uncertainties are much more complex to analyse because of changes of model structure. In a HRA, except normal unmodelled dynamics, dynamic changes with faults injected into one or more actuation elements are more important to be considered. Some possible faults of an electro-mechanical actuator also have been introduced in Section 3.1.4, including lockup, open circuit and short circuit. Firstly, situations with lockup faults injecting will be introduced.

As discussed at the end of Chapter 3, a grid structure can accommodate both force and traveling loss faults depending on the number and position of the faults, but performance degradations can also be expected. A SP structure is good at accommodating lockup faults but it also will fail when both elements in one series configuration locked up. A Bode diagram with different number of faults is shown in Fig. 4.3.

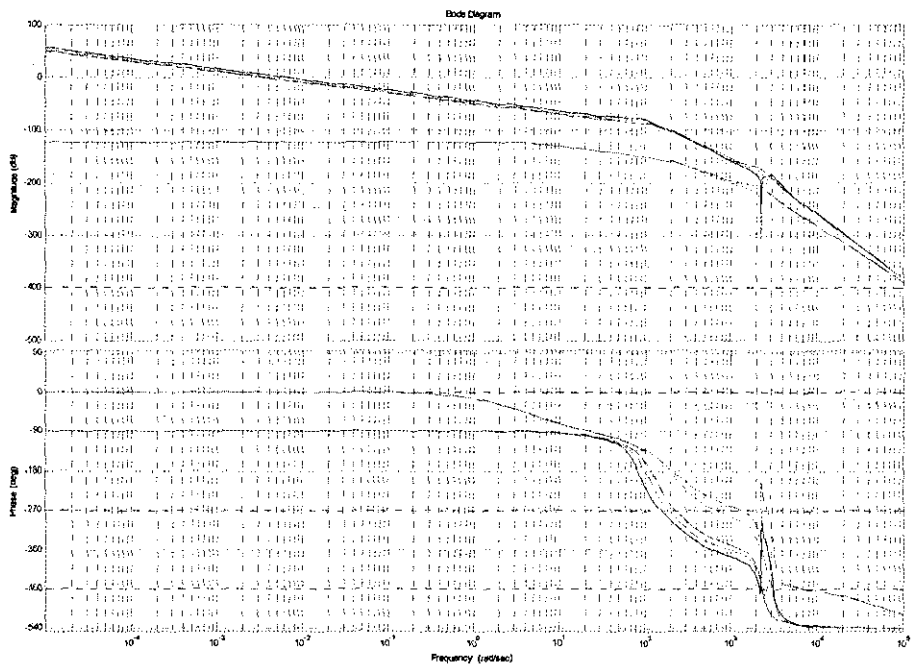


Fig. 4.3 Bode diagram of a HRA with lockup faults

In the figure, frequency performances of the systems with only one faulty element (B1), and two faulty elements in different series configurations (B1 and B2) are very similar to the healthy situation, which means the dynamics does not change very much. The other three faulty system's performances are very different, which means the dynamics have changed a lot. Situations with open circuit fault will be analysed following.

The open circuit fault has been shown to be more easily accommodated than lockup faults, especially when the mass of the load is small, which means

the inertia of the load is small. Here, the load mass is remained as 20kg. In this situation, the SP structure can continually work until all elements fails. A Bode diagram is shown in Fig. 4.4.

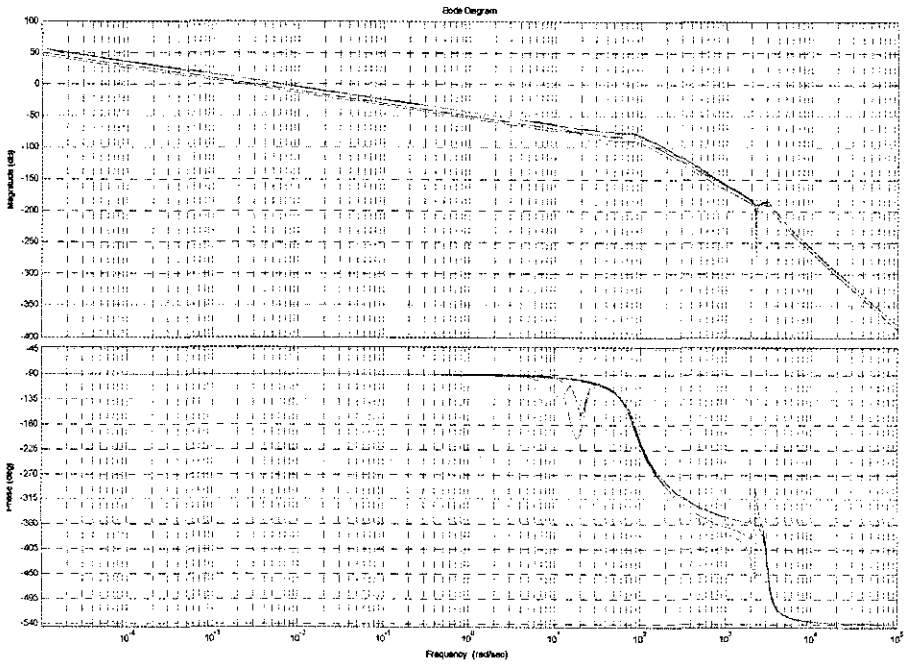


Fig. 4.4 Bode diagram of a HRA with Open circuit faults

In the figure, all performances are very similar except the system with all four elements fails which can not be shown on the figure. A short circuit fault is considered following.

As with open circuit, the HRA will not fail until all four elements are under short circuit. A Bode diagram is shown in Fig. 4.5.

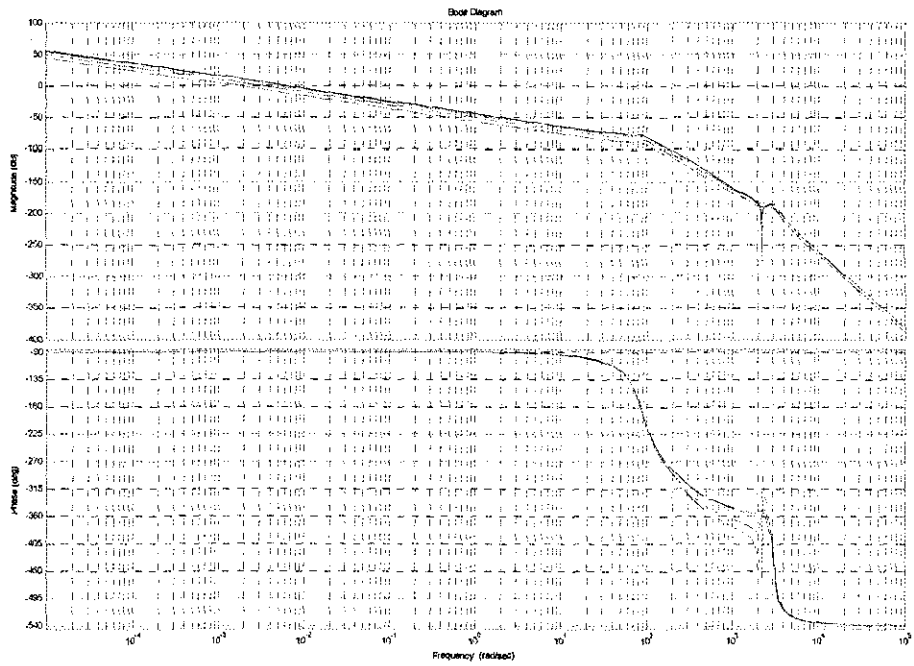


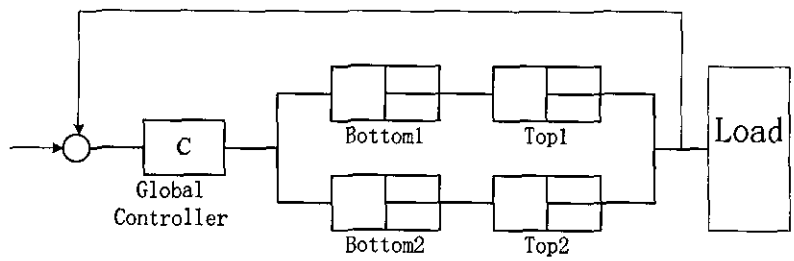
Fig. 4.5 Bode diagram of a HRA with short circuit faults

The result is similar the open circuit situation. Only the system with all elements failing can not be shown on the figure and the other systems perform similarly.

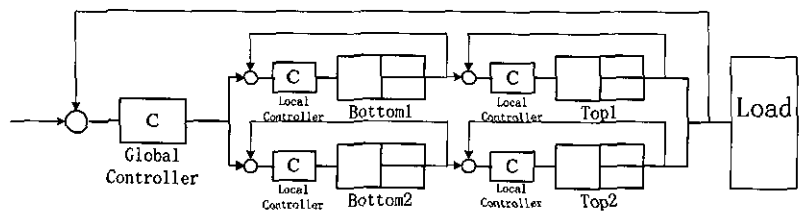
From the three frequency domain tests, it can be concluded that the frequency response of a two by two S-P HRA will not change very much until the HRA fails. This is because by using a more complex mechanical structure, the HRA itself provides some degree of robustness to faults from a point of view of robust control. But when there are too many elements fail, the HRA also will fail and no control action can avoid this. This feature provide a wide range of choices to design different kinds of controllers of a healthy HRA without considering too much about changes in dynamic when faults injecting unless the number of faults will lead to a failure of the HRA.

4.1.3 Control structures

After considering the mechanical structure, it also necessary to explore some features of control structures. Based on different level of feedback information, two kinds of controllers are defined here: a "Global Controller" is based on feedback information of the overall HRA; and a "Local Controller" is based on information fed back from each actuation element. Based upon these two kinds of controllers, two control structures can be got as shown in Fig. 4.6.



(a) voltage-driven



(b) current-driven

Fig. 4.6 Two kinds of control structures

In structure (a), there is only a global controller for the full actuator assembly. In structure (b), there is a global controller for the full actuator assembly with local controllers at each actuation elements.

For a position tracking purpose, a position feedback global controller is necessary in both structure, and current feedback is used for the local controllers. By adding a current inner loop (similar to force control), a better

control performance can be expected. Then the left structure can be called a voltage-driven structure because the global controller gives a voltage control signal to all elements, while the right one can be called current driven structure because the global controller gives a current control signal to all local controllers.

4.2 Classical Control

The robustness provided by the mechanical structure gives it a lot of possibilities to design different controllers for the HRA. Using the linear model built in Simulink and state-space model, the control design will be started with classical algorithms using loop shaping method. Proportional Integral Derivative (PID) controller, Lead and Lag compensations are some typical classical controllers used widely in industry. Here the classical control design will use these components, and started with the individual actuator.

4.2.1 Individual actuator

A Simulink model of an individual electro-mechanical actuator has been given in Section 3.1 as well as a state-space model including all parameter values listed in Tab. 3.2. The control design is firstly based on analysis of the state-space model in frequency domain, and then be verified on the Simulink model in time domain. A voltage-driven classical controller will be designed firstly.

4.2.1.1 Voltage-driven classical controller

In the frequency domain, the low frequency gain is needed to be compensated. A proportional controller with a gain of 2000V/m is then designed for the global position controller to raise the magnitude without overloading the DC motor. The controller is then applied to the Simulink model. A step signal with a set point at 0.03m is chosen as the reference signal. The performance in time domain is shown in Fig. 4.7.

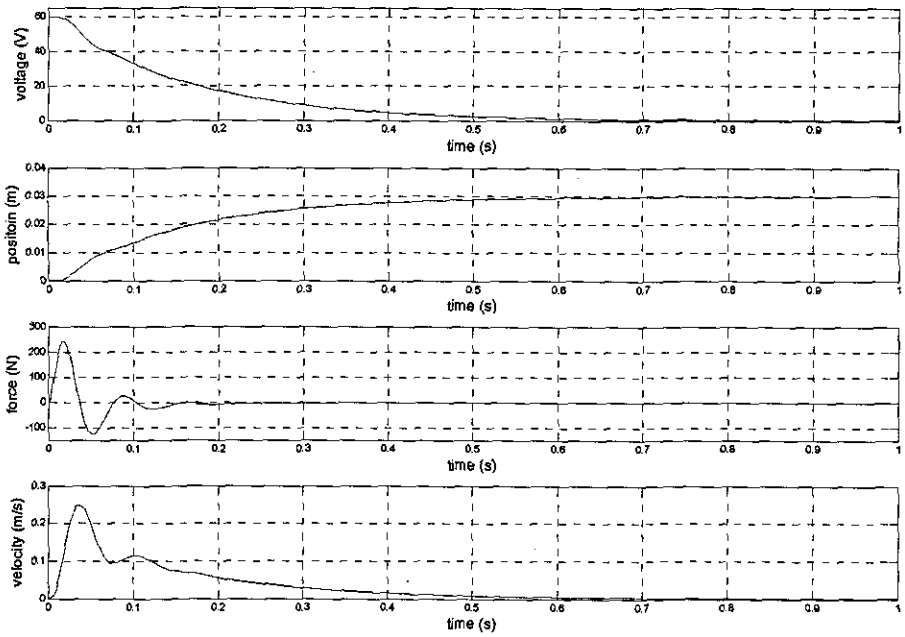


Fig. 4.7 Performances of an individual actuator with a voltage-driven classical controller

It can be found the actuator can reach the set point within about 1 second. After the gain increased, the phase margin (P.M.) is about 87deg and the gain margin (G.M.) is about 18.4dB so that no other compensator is needed. The closed-loop bandwidth is about 1Hz for the overall feedback system.

4.2.1.2 Current-driven classical controller

A current-driven control structure is considered. The voltage-driven controller will be used as a benchmark. The closed-loop bandwidth is also designed at about 1Hz. A current loop will be added. Generally an inner-loop controller will be much faster than the out-loop design. Here the bandwidth of the current loop is chosen at about 50Hz. Then an integral controller with a gain of 800 is used for the current control, with a closed-loop bandwidth of about 75Hz. A proportional controller with a lead compensator is designed for the position outer-loop. The proportional gain is set as 10, and the lead

compensator has a gain of 22, a zero at -1.14, and a pole at -25.31. Again, the controller is applied to the Simulink model with a set point at 0.03m. The time domain result is shown in Fig. 4.8.

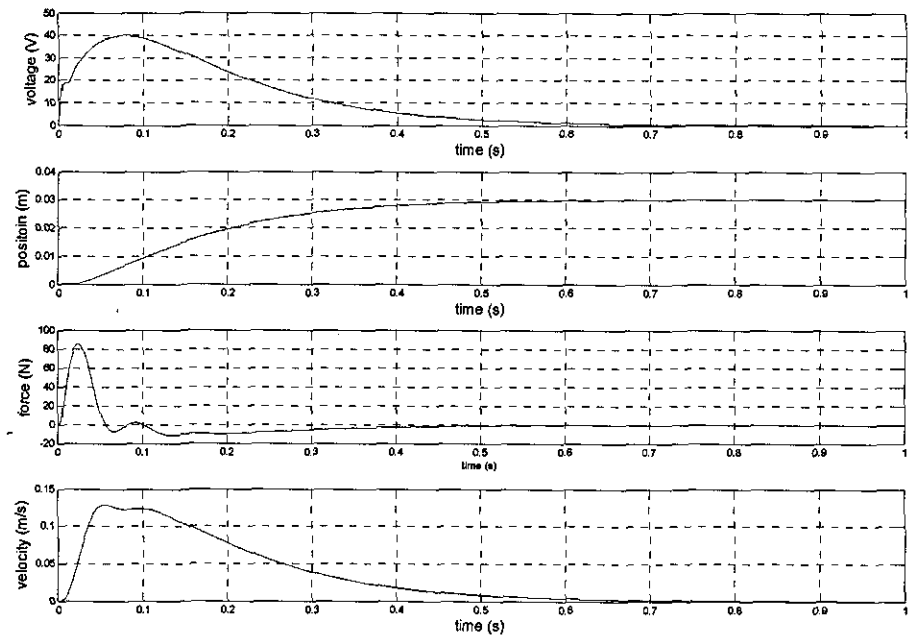


Fig. 4.8 Performances of an individual actuator with a current-driven classical controller

Similar performances can be found comparing with the voltage-driven controller, although the output voltage from the current-driven controller is limited at 40V, which means the system using current-driven structure can have faster responses. Another difference is that the control voltage reaches the peak value at about 0.1s so that the actuator has a soft start. After the compensation, the phase margin (P.M.) is about 76.7deg and the gain margin (G.M.) is about 26.1dB. The closed-loop bandwidth is about 1Hz for the overall feedback system.

4.2.2 Two-by-two series-in-parallel HRA

A similar process is applied to design classical controllers for the two-by-two series-in-parallel HRA whose Simulink model has been given in Section 3.3.2 and state-space model given in Section 4.1.1. The HRA is a combination of individual actuators so that it has very similar performance under frequency domain with the individual actuator. A voltage-driven controller is firstly designed.

4.2.2.1 Voltage-driven classical controller

Same as the individual actuator, a proportional controller is chosen as the global position controller. Because the SP structure can double the position output, the set point is chosen as 0.06m, and the proportional gain is reduced to 1000V/m without overloading the DC motors. The performance of the closed-loop system is shown in Fig. 4.9.

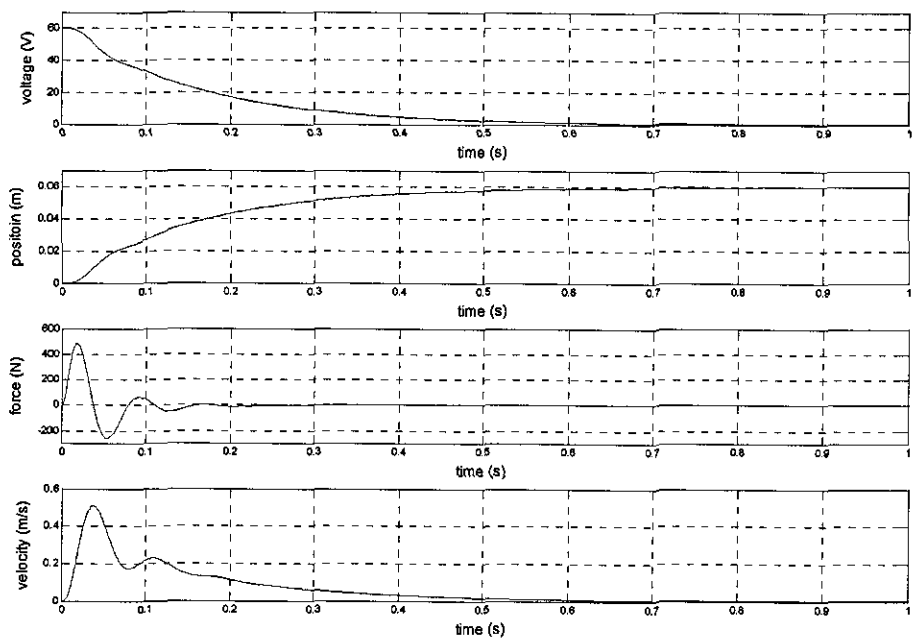


Fig. 4.9 Performances of an HRA with a voltage-driven classical controller

The performance is very similar with the individual actuator's although the set point has been increased. After the compensation, the phase margin (P.M.) is 83.7deg and the gain margin (G.M.) is 11.4dB so that no other compensator is needed. The closed-loop bandwidth is about 1Hz for the overall feedback system.

4.2.2.2 Current-driven classical controller

Same individual current loop controller is used as local controllers in the current-driven control structure. Using the voltage-driven controller as the benchmark, a proportional controller with a lead compensator is designed for the global position controller. The proportional gain is set as 4.7, and the lead compensator has a gain of 33, a zero at -1.09, and a pole at -36.31. A 0.06m step signal is used as a reference signal. The time domain performance is shown in Fig. 4.10.

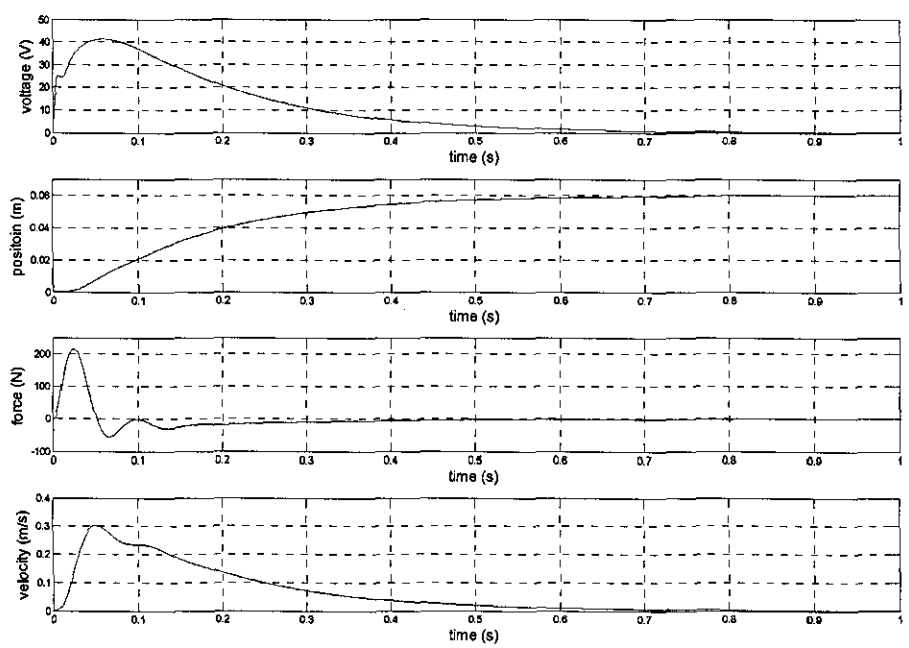


Fig. 4.10 Performances of an HRA with a current-driven classical controller

Although the current-driven structure will give separate control signal to each element in the HRA, all of them are very similar since the same elements are supposed to be used in the HRA so that only one voltage signal is shown in Fig. 4.10. The voltage input is limited at 40V so that the HRA can perform faster with a higher gain controller. With this design, the P.M. is 80.8deg, the G.M. is 23.5dB, and the closed-loop bandwidth is again about 1Hz.

4.2.3 Summary of the classical control design

Some classical controllers based on both voltage-driven and current-driven structures have been designed. Starting from classical loop shaping design can help to access with the performance of the HRA under both frequency and time domain. Using voltage-driven controller as a benchmark, both controllers work well under a healthy situation to reach the set point. More tests under faults injected situations will be given in Chapter 5. The optimal control design, in the next section will also use the classical voltage-driven controller as the benchmark.

4.3 Optimal Control

Optimal control design is different from the classical loop shaping method, because it seeks to get the best possible performance of the plant (generally computer-assisted). In order to achieve this remarkable aim, those aspects of the plant's behaviour are incorporated into a mathematical expression (cost function). The controller design process synthesizes a controller which will minimize the cost function.

Here, the optimal control design is based on State-space models of systems to find a state variable feedback (SVF) matrix to minimize the cost function. A basic state variable feedback approach, given a usual State-space representation of a system $\dot{x} = Ax + Bu, y = Cx + Du$, is to find out a regulator K making $A_C = A - BK$, where A_C is the closed-loop plant matrix. The eigenvalues of A_C are therefore the same as the closed-loop poles of the systems. Thus, the elements of K can be chosen to give A_C any required set of closed-loop eigenvalues (poles), so long as the original system is completely state controllable.

Optimal control methods provide another way of placing the closed-loop poles of a system, in order to achieve the desired behaviour. In this case, the designer does not know the closed-loop pole locations that satisfy the optimal performance. Instead, the poles are placed by the optimal control design procedure. Thus, optimal control methods are one more way of choosing the contents of the feedback matrix K in an SVF scheme.

The optimal control method used in the HRA application is called Linear Quadratic Regulator (LQR), which uses a simple quadratic cost function

expressed as $J = \int_0^t \left[x^T Qx + u^T Ru\right]dt$. The required closed-loop performance then can be got by tuning Q and R until the minimization of the resulting cost function leads to a suitable result on the plant. With modern computer-assisted control system design (CACSD) environments, it is quite easy to go through an iterative cycle of design. The MATLAB Control Systems Toolbox used in this application has a single command lqr for designing an optimal regulator given only the matrices Q and R, and the State-space model of the plant.

The LQR methods can also be extended to the LQG case, in which the presence of noise (which is assumed to be Gaussian-distributed) is taken to be corrupting the process. Such systems normally involve the use of a Kalman filter (KF) as a state estimator, combined with an optimal controller. The KF can extract state estimates from disturbed and noisy signals, but only based on accurate plant models and information about white noise disturbances. The MATLAB Control Systems Toolbox provides simple commands for designing the KF given the previous error covariance Q_n and the noise statistics R_n. The optimal control design is firstly applied on an individual actuator, and than extended to the two-by-two series-in-parallel HRA.

4.3.1 Individual actuator

As mentioned, LQG control is applied here and the purpose is again tracking a 0.03m step signal. Only the voltage-driven structure is considered for the optimal control which means there will be only one global controller based on the performance of the overall position. The current-driven structure is not considered because of the complexity of the model when current

inner-loop is added into a two-by-two or more complicate HRA structure. Since the LQR controller is a state variable feedback regulator, it is necessary to add another state, which is the integral of position error, to the previous seven states individual actuator model for a position tracking purpose. Also, a Kalman filter is combined to provide state estimate. Velocity and position outputs are chosen as the inputs to the KF. A new Simulink model is then created for the LQG control as shown in the next figure.

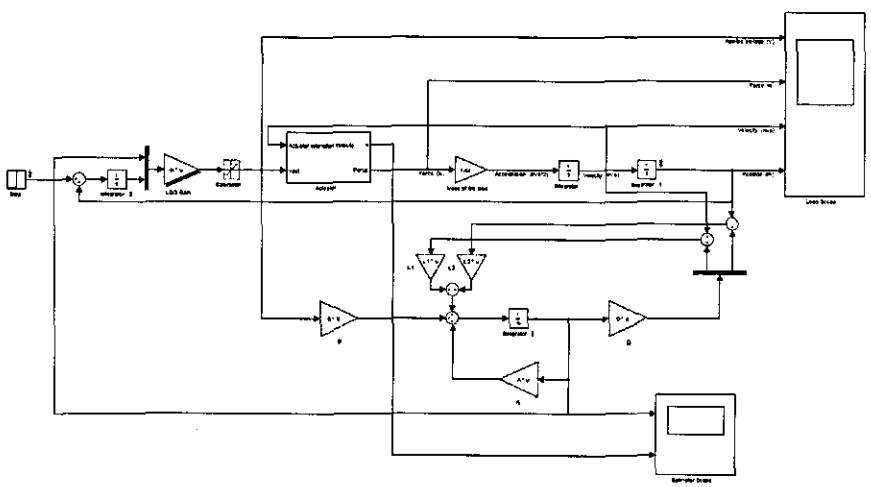


Fig. 4.11 Simulink model of an individual actuator using LQG control

In the model, the states estimated from the Kalman filter, plus the extra integration state, are fed back to the LQR regulator matrix to provide control signal to the individual actuator. The regulator K is calculated depending on the choice of weighting matrix Q and R in the cost function. There are eight states to be controlled so that state weighting matrix Q is chosen as an eight by eight diagonal matrix. To ensure a small steady state error, the eighth state which is the integral of position error must be very small, so that the corresponding parameter value in Q matrix has to be chosen as a very big value, which is 10^9. Comparing with this big value, all other seven parameters in Q can be chosen to have a relatively small value, which is 0.01

here. Since there is only one input in this system, input weighting R is chosen as a single value as 1. The Kalman filter is based on a State-space model of the individual actuator, so that state matrix, input matrix, and output matrix should be put in related position in the KF, i.e. block A, B and C. The regulator L also depends on the choice of previous error covariance Q_n and the noise statistics R_n. There is only one control input to the KF so that Q_n is estimated as 0.01 and two outputs are chosen as inputs to the KF, so that R_n is estimated as a two by two diagonal matrix with both values also set to 0.01. Using MATLAB commands, both regulators for LQR controller and Kalman filter can be calculated easily using these four tuning matrices, A simulated performances under time domain of the individual actuator are shown in Fig. 4.12.

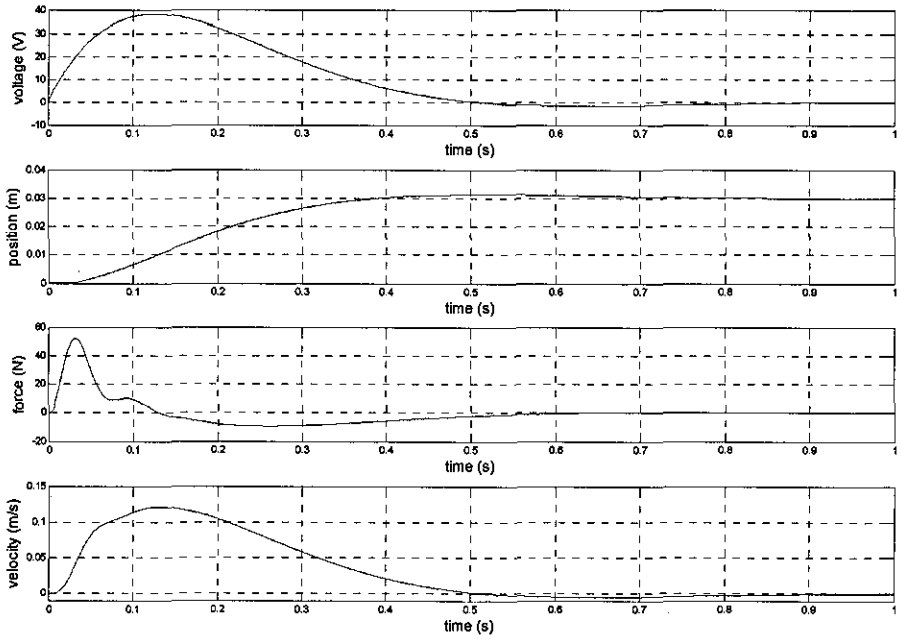


Fig. 4.12 Performances of an individual actuator using LQG control

A little overshoot can be found at the position output. The performance is very similar with the result using current-driven classical controller. The

voltage peak is less than 40V, and reaches the peak value at about 0.1s so that the actuator has a soft start. Under frequency domain, the phase margin (P.M.) is 64.9deg and the gain margin (G.M.) is 24.9dB. The closed-loop bandwidth is about 1Hz for the overall feedback system.

4.3.2 Two-by-two series-in-parallel HRA

The LQG controller is a state variable feedback algorithm so that a state-space model is necessary. A full order two-by-two series-in-parallel HRA State-space model comprises 26 states as introduced in Section 4.1. Only voltage-driven structure is applied to avoid making the control design too complex. Same design process as introduced for individual actuator will be applied. But when calculating control regulator, it will meet some computation problem for the HRA. This is due to the repeated dynamics inside the HRA model. When building up the state-space model of the two-by-two HRA, all four actuation elements are supposed to be identical, so that same dynamics are repeated four times inside the model. Although the four elements are in different places inside a HRA, each actuation element performs very similar when same control input is applied to all of them under a voltage-driven control structure. These repeated dynamics reduce the controllability and observability of the system which causes mathematical problems for the LQG control design.

To obviate this problem, three different approaches will be presented. The first approach is to add some small variations into the parameters so that each element's parameters are slightly different from the others inside the HRA. The other two methods employ model reduction. The central problem in model reduction is to find a low-order approximation given a high-order linear time-invariant stable model such that the infinity norm of the

difference is small. The advantage is that a simpler controller can be found by reducing the number of states. Here, the low-order approximations are found through two ways. One uses a mathematical method called balanced realization truncation and the other uses a physical reduction (based on the physical equations). All three controllers are designed to track a 0.06m step signal.

4.3.2.1 Parameter variation

The purpose of this method is to avoid repeated dynamics by making each actuation element slightly different from each other. All parameter values of an individual actuator have been listed in Table. 3.3. Ten percent variations of each parameter are introduced to each actuation element in different place, which means the value of each parameter is changed to a random value in the range between 90% and 110%. The parameter variation (PVA) in actuation element helps to avoid the repeated dynamics of each sub element which would otherwise make calculation of the controller and observer gains difficult. After using different parameters of each element, a new State-space model can be produced. Then both LQR control matrix and Kalman filter gains can be calculated using MATLAB commands.

An extra state which is the integral of position error is added for a position tracking purpose. Then a suitable regulator K can be calculated by tuning the weighting matrix Q and R. The corresponding parameter in Q matrix is chosen as big as 10^9, and all other parameters in Q is set as 0.01. The input weighting R is chosen as a single value as 1. Putting state, input and output matrixes into related position in the KF, choosing both previous error covariance Q_n and the noise statistics R_n as 0.01 for both input and outputs, the Kalman filter regulator L can be calculated in MATLAB.

Simulated performances under time domain of the HRA are shown in Fig. 4.13.

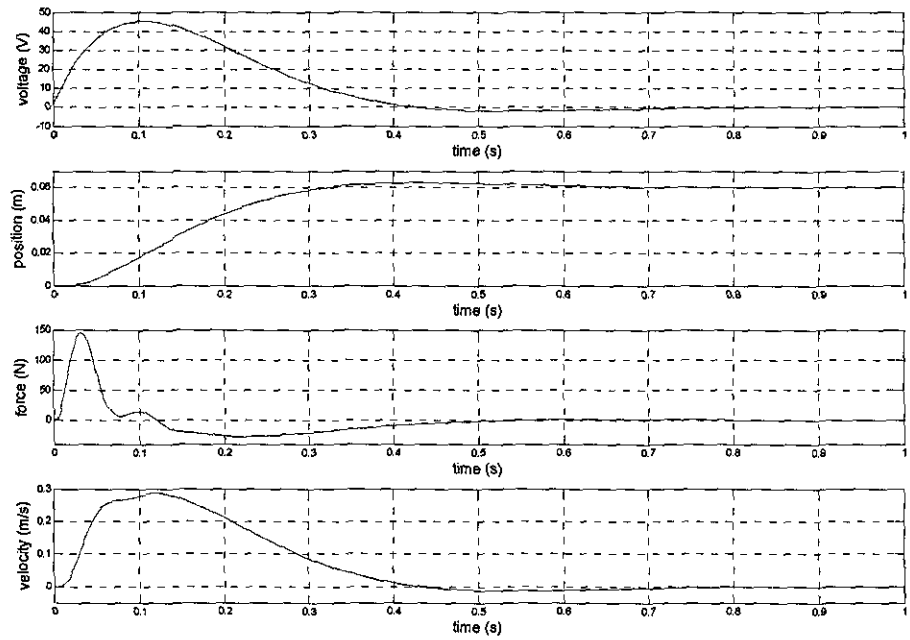


Fig. 4.13 Performances of a HRA using PVA LQG control

The position output can reach the set point with a little overshoot which is very similar as the performance of individual actuator using LQG controller. The control input still below 50V which means the controller can give a faster response when higher gain is applied. Under frequency domain, the phase margin (P.M.) is 64.2deg and the gain margin (G.M.) is 23.7dB. The closed-loop bandwidth is about 1.6Hz for the overall feedback system.

4.3.2.2 The balanced realization truncation

As introduced, the central problem in model reduction is to find a low-order approximation given a high-order linear time-invariant stable model such that the infinity norm of the difference is small. The advantage is that a simpler controller can be found by reducing the number of states. The

balanced realization truncations (BRT) based on the balanced realization of the model which evaluates the contributions to the response of each mode. The state coordinate basis is selected as a diagonal matrix in descending order. The magnitudes of the diagonal entries reflect the contributions. Only the most effective states, which affect the input-output mostly, are kept so that similar performance still can be achieved. Using the command balmr in MATLAB Control System Toolbox, it helps to find the balanced-truncation model Ghed(s) with order k from the original system G with order n, such that the infinity-norm of the error (Ghed(s) - G(s)) less than two times of sum of the (n-k) smaller Hankel singular values, i.e. $\|G-Ghed\|_\infty \le 2\sum_{k+1}^{n}\sigma_i$.

In this case, eight states are kept in the balanced truncation model. The error band is 0.055. Two figures showing both original and the reduced order systems' performances in time domain and singular values in frequency domain are given Fig. 4.14 and Fig. 4.15 respectively.

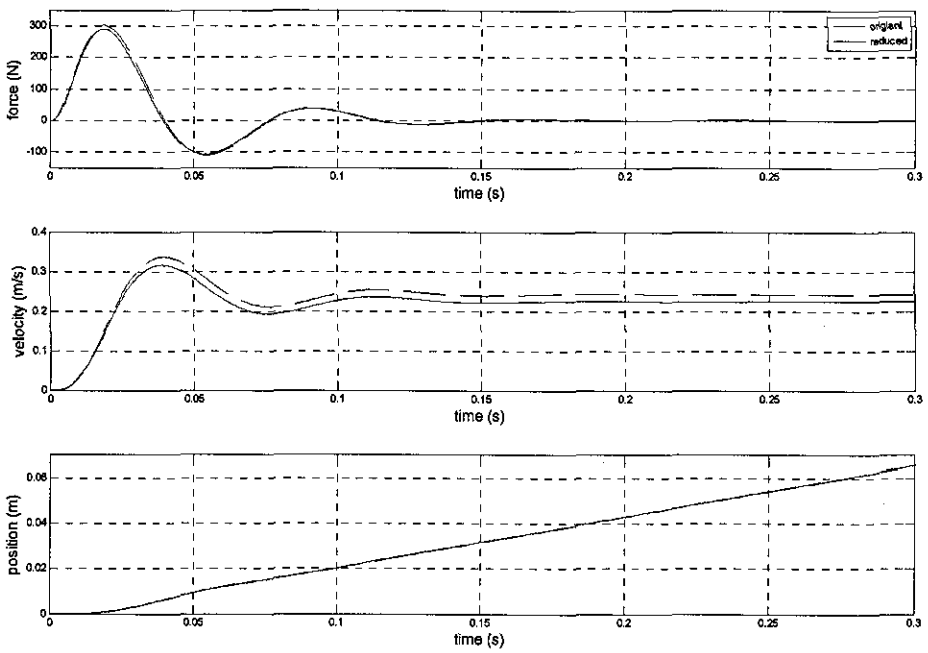


Fig. 4.14 Comparison under time domain between full order and BRT model

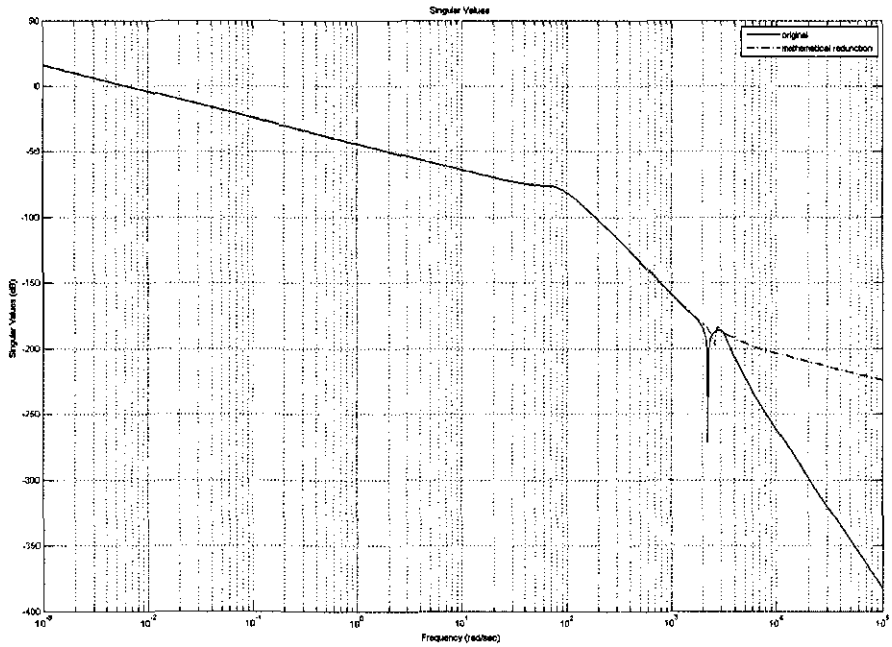


Fig. 4.15 Comparison under frequency domain between a full order and a BRT model

A typical 36V voltage is applied to both original full-order model and the reduced-order model. In the time domain, slight differences can be found in velocity output. In both force and position outputs, both models perform very similar. In the frequency domain, the differences between two models increase after 2000rad/s. For lower frequencies, the performances are very similar. Based on these results it can be said that the eight states model using balanced realization truncation is very close to the full-order system, and the controller can be designed based on this reduced-order model with an order of 8.

An extra state which is the integral of position error is added, and the corresponding parameter in states weighting matrix Q is chosen as 3^8 to ensure a small steady state error. All other eight parameters in Q matrix are set at 0.01. The input weighting R is chosen as a single value as 1. For the KF design, both previous error covariance Q_n and the noise statistics R_n

are set at 0.01 for both input and outputs. Simulated performance is shown in Fig. 4.16.

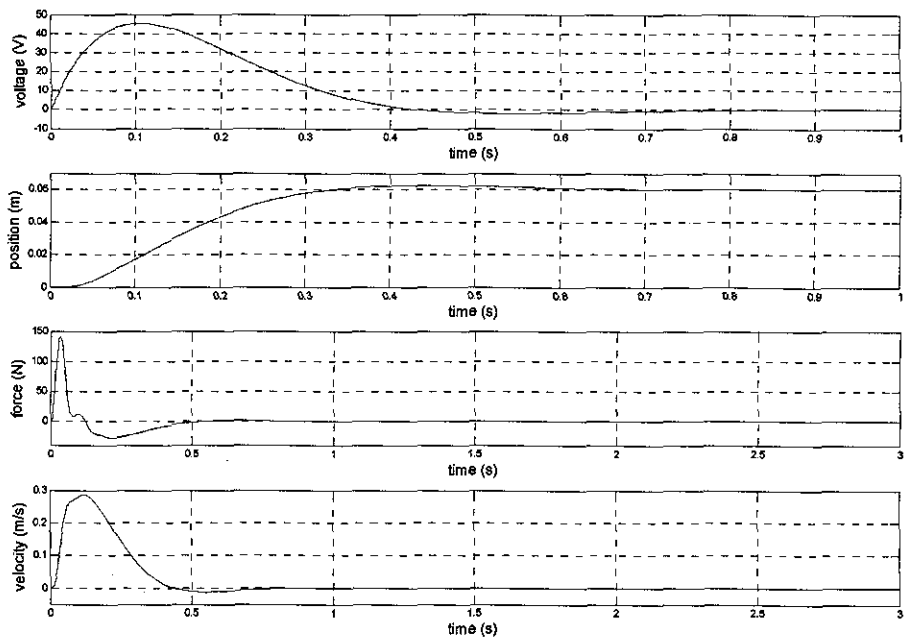


Fig. 4.16 Performances of a HRA using BRT LQG control

In the frequency domain, the phase margin (P.M.) is 64.7deg and the gain margin (G.M.) is 22.1dB. The closed-loop bandwidth is about 1.6Hz for the overall feedback system.

4.3.2.3 The physical reduction

As for the balanced realization, the purpose for physical reduction (PR) is to find a low-order model which has similar performance to the high-order model. The difference here is that the reduction is based on physical understanding rather than mathematical methods. In this approach, the HRA is assumed as a bigger individual actuator with four times power input, but double force, speed and position outputs. This approach gives a State-space model with only seven states, as the individual actuator model. Both the full

order model and the reduced one are compared under time and frequency domain and shown in Fig. 4.17 and Fig. 4.18 respectively.

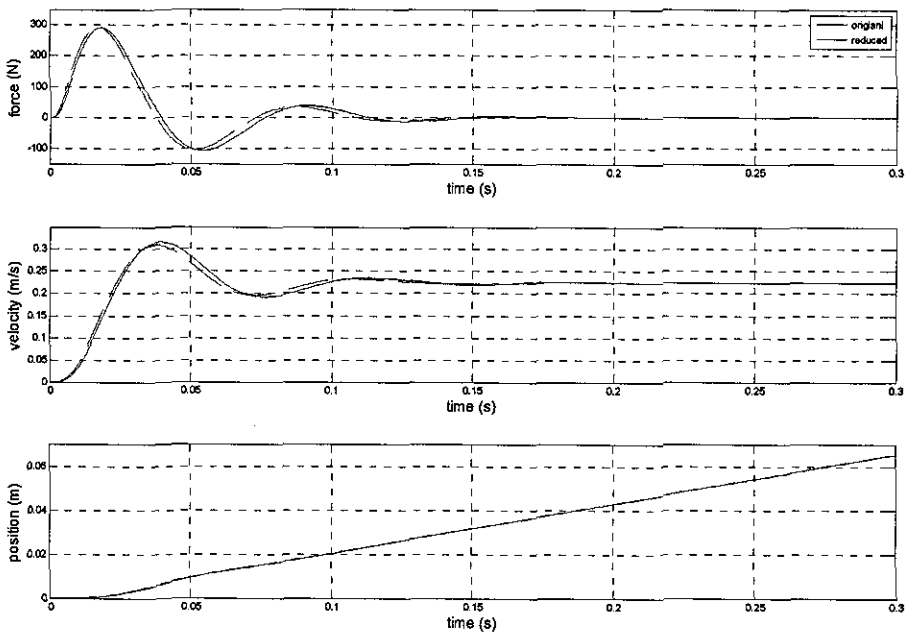


Fig. 4.17 Compare under time domain between a full order and a PR model

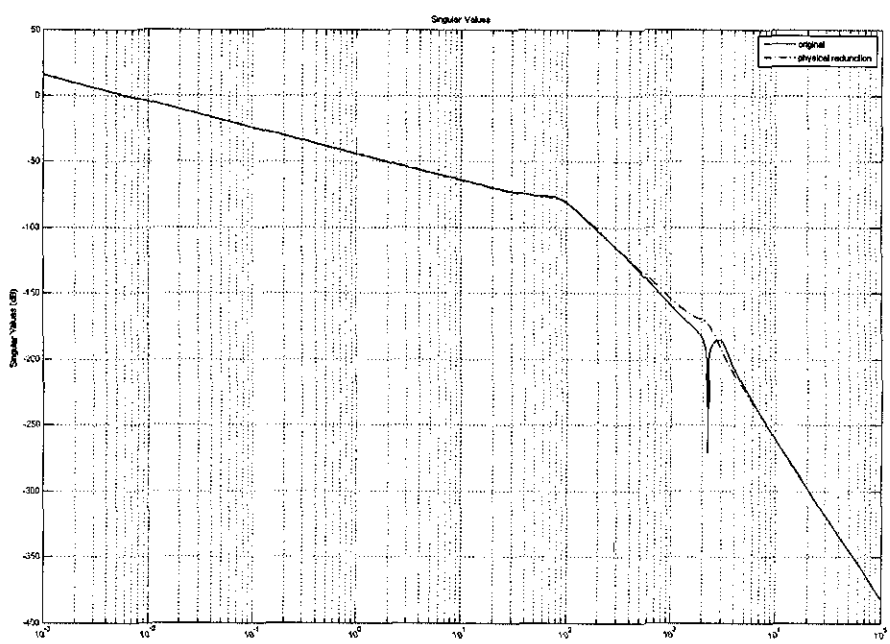


Fig. 4.18 Compare under frequency domain between a full order and a PR model

The simulation results show very similar performance between the full order and reduced order model in both frequency and time domain. Then, based on this 7 order State-space model, a LQG controller can be designed.

An extra state which is the integral of position error is added, with a corresponding parameter in states weighting matrix Q set as 10^{10}. All other seven parameters in Q are chosen as 0.01. Input weighting R is chosen as a single value as 1. Putting state, input and output matrixes into related position in the KF, choosing both previous error covariance Q_n and the noise statistics R_n as 0.01 for both input and outputs, the Kalman filter regulator L is also calculated in MATLAB. Simulated result is shown in Fig. 4.19.

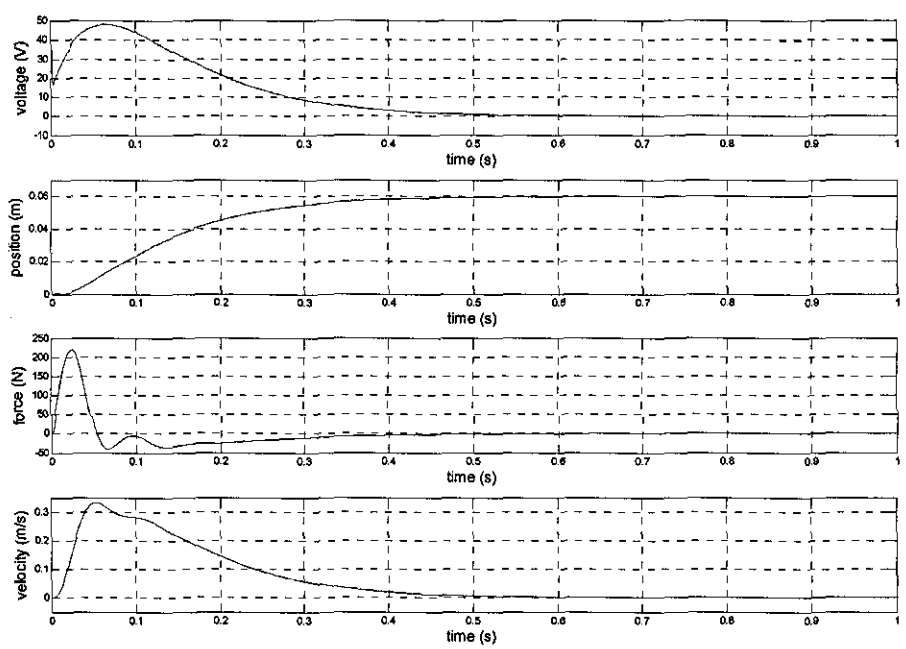


Fig. 4.19 Performances of a HRA using PR LQG control

The performance is very similar as the result using the other two kinds of LQG control although there is no overshoot in position output. The control

input still below 50V which means the controller can give a faster response when higher gain is applied. Under frequency domain, the phase margin (P.M.) is 76.1deg and the gain margin (G.M.) is 23dB. The closed-loop bandwidth is about 1.3Hz for the overall feedback system.

4.3.3 Summary of the optimal control design

Three kinds of methods are used to avoid mathematical problem caused by repeated dynamics. The simulation results show that they all have similar performances using the classical voltage-driven controller as a benchmark. The peak control voltage does not reach the maximum value (60V), which means a faster response can be got. The physical reduction method is a better model reduction method because it provides similar performance remaining smallest number of states (seven) and it is not based on the full order model which makes the modelling process much simpler. But different from the full order one, the order reduced models can not present performance of each actuation element inside the HRA because the states lose the physical meaning during the reduction process.

4.4 Chapter Summary

In this chapter, a discussion of robustness of the two-by-two series-in-parallel HRA structure based on a state-space model has been given. The results show that the frequency domain performance will not change significantly until the whole structure fails. This feature provides a wide range of choices of control design for the HRA. Five kinds of controllers are then designed including both classical and optimal (LQG) algorithms. Using the voltage-driven classical controller is used as a benchmark for other controllers' design, all controllers show similar performances under a healthy situation, but the other four controllers' peak control voltages do not reach the limitation of 30V so that they can perform faster. For the classical designs, two kinds control structures are applied. For the optimal designs, because the repeated dynamics in the state-space model causes mathematical problems for the computer to calculate the controller regulator, three approaches are applied to solve this problem including both full order and model reduction methods. The physical reduction method is a better model reduction method because it provides similar performance remaining smallest number of states (seven) and it is not based on the full order model which makes the modelling process much simpler. But different from the full order one, the order reduced models can not present performance of each actuation element inside the HRA because the states lose the physical meaning during the reduction process. A more detailed compare of all five controllers will be listed later in Chapter 5, including faults injected situations.

5. SIMULATION RESULTS

Five controllers using both classical and optimal algorithms have been designed based on the two-by-two series-in-parallel HRA model, and all controllers have been tested under a healthy situation. As introduced previously, the main advantage of the HRA is the fault tolerance capability without any reconfiguration. To demonstrate this capability, the HRA will be tested under faults injected situations with all five controllers. Three kinds of faults are considered, open circuit, short circuit, and lockup. The Bottom 1 element is supposed to be faulty. The HRA is supposed to continually finish the position tracking task, although performance degradation can be expected and acceptable.

In this chapter, performances of all five controllers under a healthy situation will firstly be compared as a reference. Then the three kinds of faults are considered separately with performance degradation level concluded at the end.

5.1 Healthy Situation

The results of all five controllers design in Chapter 4 can be seen in Fig. 5.1, which shows the response to a 0.06m step input.

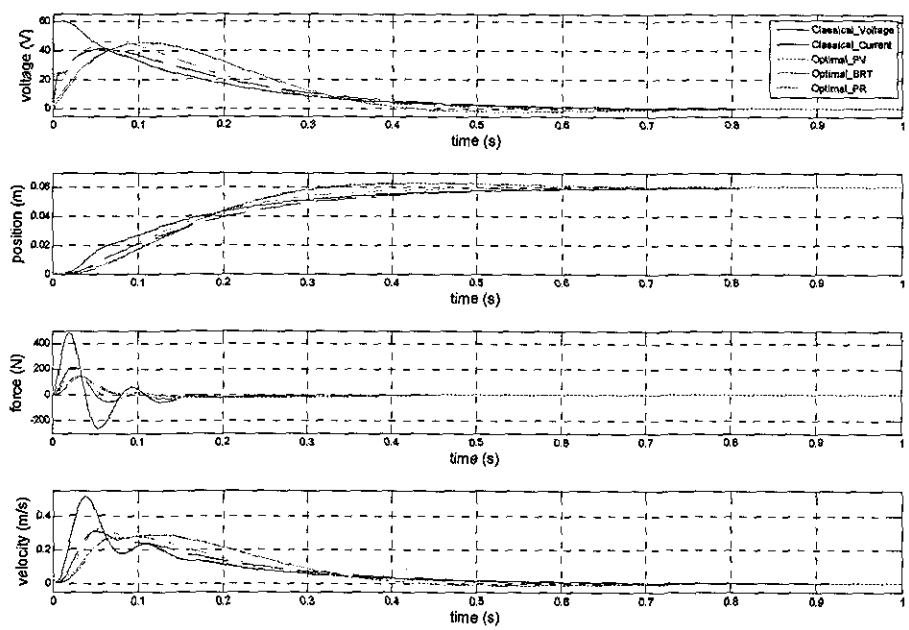


Fig. 5.1 Simulation results of all controllers under a healthy situation

Basically all controllers perform very similar at position output because other controllers are designed using the classical voltage-driven controller as a benchmark. In contrast to the classical voltage-driven controller, all other controllers can be designed to respond faster because the control input has not reach the maximum voltage the DC motor can accept. More detailed features at position output under time domain, including Steady-state Error (SE), Rise Time (RS), i.e. from 0% to 90%, Settling Time (ST), i.e. with a 5% tolerance, Overshoot (OS) and Peak Velocity (PV), and frequency domain, including open-loop Phase Margin (P.M.), Gain Margin (G.M.) and closed-loop Band Width (BW), are shown in the Table. 5.1.

Table. 5.1 Time and frequency features of all controllers with no fault

Time Domain	SE(m)	RT(s)	ST(s)	OS	PV(m/s)
Classical Voltage-Driven (CV)	0	0.36	0.46	0%	0.51
Classical Current-Driven (CC)	0	0.39	0.49	0%	0.30
Optimal Full-Order (OF)	0	0.26	0.28	4.7%	0.29
Optimal Mathematical-Reduction (OM)	0	0.26	0.29	4.1%	0.29
Optimal Physical-Reduction (OP)	0	0.31	0.38	0%	0.32

Frequency Domain	P.M.(deg)	G.M.(dB)	BW(Hz)
Classical Voltage-Driven	83.7	11.4	1.1
Classical Current-Driven	80.8	23.5	1.0
Optimal Full-Order	64.2	23.7	1.6
Optimal Mathematical-Reduction	64.7	22.1	1.6
Optimal Physical-Reduction	76.1	23	1.3

From the table, it can be found that at the optimal controllers responds a little faster than classical controllers although a little overshoot can be found for the OF and OM controllers. The OP controller responds more like a classical controller compared with the other two optimal controllers. In the frequency domain, same situation can be found. The OF and OM optimal controllers provide a higher closed-loop bandwidth at about 1.6 Hz, while the PR optimal controller is closer to both classical controllers, which provide a closed-loop bandwidth at about 1 Hz. Using both the figure and the table as a benchmark, performances of all controllers under different faulty situations will be shown in the following sections.

Here, the Bottom 1 actuation element in a two-by-two series-in-parallel

HRA is supposed to be suffering a fault and the other three elements will help to overcome the failure. If only one element fault is considered, the position of the faulty element is not important because all four elements are playing a similar status in a two-by-two structure. If more faults are considered, the positions of faults will cause different results, i.e. two lockup faults happen in series will cause a failure of the overall HRA while in parallel will not result a failure. Same step signal with a 0.06m set point will be applied as the reference. All five controllers will be tested under faulty situations without any reconfiguration in either control level or mechanical level. The HRA is expected to continue working under faulty situations although performance degradation will happen. The following sections present results for different fault conditions.

5.2 Open Circuit Fault

Open circuit causes force capability loss of an individual actuator. The open-loop test shows that the SP structure responses slower when one open circuit happens. In the closed-loop for position tracking, it can be expected that the HRA will continually track the reference signal using a longer time to reach the set point. Performances under time domain of all five controllers when the open circuit fault happens to the Bottom 1 element are shown in Fig. 5.2.

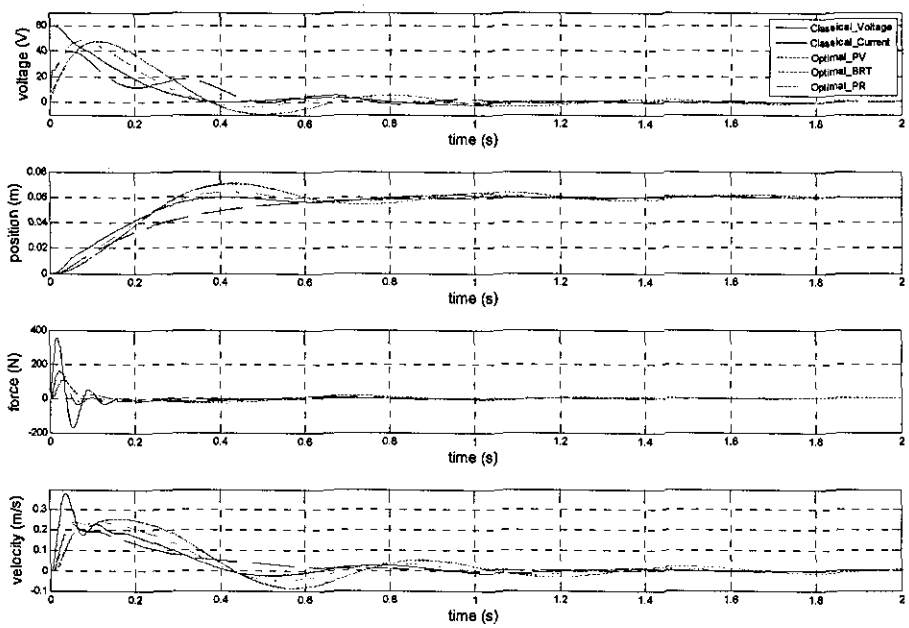


Fig. 5.2 Simulation results of all controllers with an open circuit fault

Comparing with the results under a healthy situation as shown in Fig. 5.1, an open circuit fault will make the overall performance show some fluctuations at position output, i.e. more overshoot and longer settling time. The same performances characteristics which were given for the healthy system are listed in Table. 5.2.

Table. 5.2 Time and frequency features of all controllers with an open circuit fault

Time Domain	SE(m)	RT(s)	ST(s)	OS	PV(m/s)
Classical Voltage-Driven	0	0.29	0.70	2.5%	0.38
Classical Current-Driven	0	0.52	0.67	0%	0.23
Optimal Full-Order	0	0.27	1.11	18%	0.26
Optimal Mathematical-Reduction	0	0.27	1.11	17%	0.25
Optimal Physical-Reduction	0	0.28	0.76	6.7%	0.24

Frequency Domain	P.M.(deg)	G.M.(dB)	BW(Hz)
Classical Voltage-Driven	78.5	20.2	1.7
Classical Current-Driven	83.1	26.1	0.7
Optimal Full-Order	53.8	26.7	1.8
Optimal Mathematical-Reduction	54.7	25.1	1.8
Optimal Physical-Reduction	67.4	25.9	1.8

The table shows that in all cases the HRA still can reach the set point with zero steady-state error, but when compared with the healthy situation, some performance degradation can be seen especially when using optimal controllers. Although the rise time does not change much, even faster than a

healthy situation under some controllers, the settling time is longer than under a healthy situation. The classical CV controller shows some overshoot while there is no overshoot under a healthy situation. When using optimal OF and optimal OM controllers, the settling time is more than four times longer and the overshoot is three times bigger than under a healthy situation. The peak velocity is also decreased by about one fourth compared with a healthy situation. The classical CC controller shows the smallest degradation. Although the rise time is increased, the settling time is the smallest compared with other controllers and only increased by about 37% compared with the same controller under a healthy situation. In optimal controllers, OP controller shows best performance with the fastest response and smallest overshoot.

The performances of the various controllers will change depending upon the detailed design, so that it is hard to say which kind of controller is most suitable to the HRA under a faulty situation. However all controllers show the ability of fault tolerance by achieving the required function because of the redundancies in mechanical structure. Performance degradation can be expected, and using the current designs of each controller, the classical current-driven controller shows the best performance when open circuit fault happens to one element.

5.3 Short Circuit Fault

Short circuit is the other electronic fault modelled here. Under this situation, control voltage cannot be applied to the DC motor so that the motor can not generate any positive torque to drive the mechanical actuator. But, different from an open circuit fault, the current inside the motor will not become zero when the mechanical part is moving because the electro loop is still closed, so that the motor will generate a negative torque as a consequence of being pushed by the rest of the actuator system. Under this situation, the actuator will have some similarity with a lockup fault. The simulation results will be shown in Fig. 5.3.

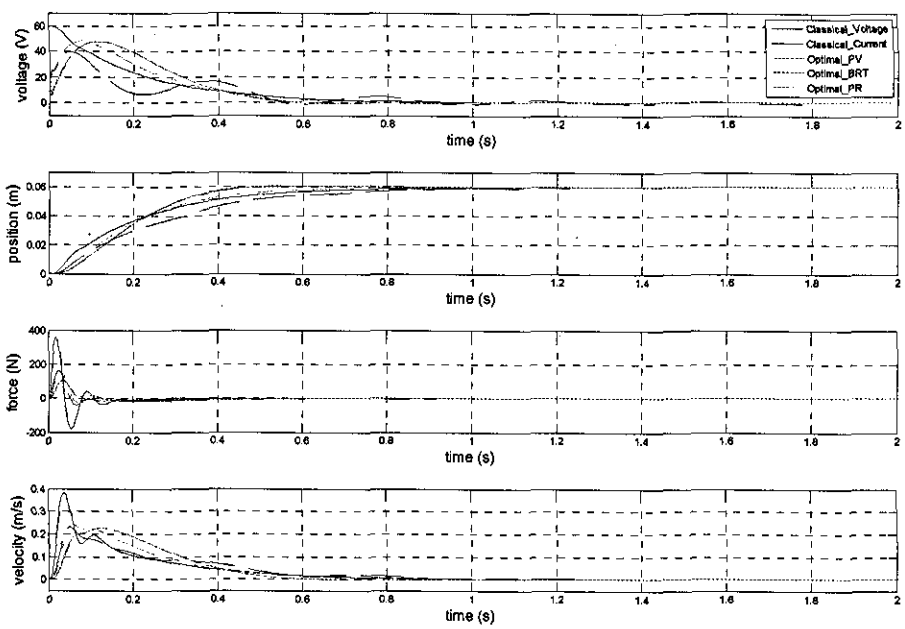


Fig. 5.3 Simulation results of all controllers with a short circuit fault

Basically, it can be found that the HRA still can follow the step signal. Different from open circuit fault, all controllers show quite smooth response at position output under a short circuit fault. The same performances characteristics are listed in Table. 5.3.

Table. 5.3 Time and frequency features of all controllers with a short circuit fault

Time Domain	SE(m)	RT(s)	ST(s)	OS	PV(m/s)
Classical Voltage-Driven	0	0.48	0.62	0%	0.38
Classical Current-Driven	0	0.60	0.77	0%	0.23
Optimal Full-Order	0	0.35	0.40	1.2%	0.23
Optimal Mathematical-Reduction	0	0.36	0.41	0.8%	0.23
Optimal Physical-Reduction	0	0.43	0.54	0%	0.24

Frequency Domain	P.M.(deg)	G.M.(dB)	BW(Hz)
Classical Voltage-Driven	87.7	20.1	0.78
Classical Current-Driven	84.3	25.8	0.62
Optimal Full-Order	69.6	26.2	1.19
Optimal Mathematical-Reduction	70.1	24.6	1.17
Optimal Physical-Reduction	79.5	25.5	0.91

Compared with Table. 5.1, it can be seen that both rise time and settling time under faulty situation are longer than under healthy situation. Peak velocity is smaller, but overshoot is also smaller. Here both optimal OF and OM controllers perform in a very similar manner, faster than other controllers, but the degradation in settling time for these two controllers reaches about

45% while the smallest degradation happens using classical CV controller which is about 35%. Optimal OP controller is a little bit slower than the other two optimal controllers, but it is faster than classical controllers also with a degradation of 42%. The classical CC controller shows the worst performance with slowest response and biggest degradation which is about 57%. Under frequency domain, changes also can be found. Both phase margin and gain margin rises while the bandwidth is smaller. As a conclusion, optimal controller shows a faster response than classical controller. But when considering the performance degradation, the classical voltage-driven controller is most suitable to the open circuit fault. Again, the results could be changed with different design.

5.4 Lockup Fault

Lock-up is a mechanical fault which makes the actuator lose the capability of traveling. In an HRA, if all elements in series are locked up, then the whole HRA will fail. From this point of view, the lock-up fault is more difficult to be accommodated and more degradation can be expected compared with the electrical faults. The simulation performance of all controllers is given in Fig. 5.4.

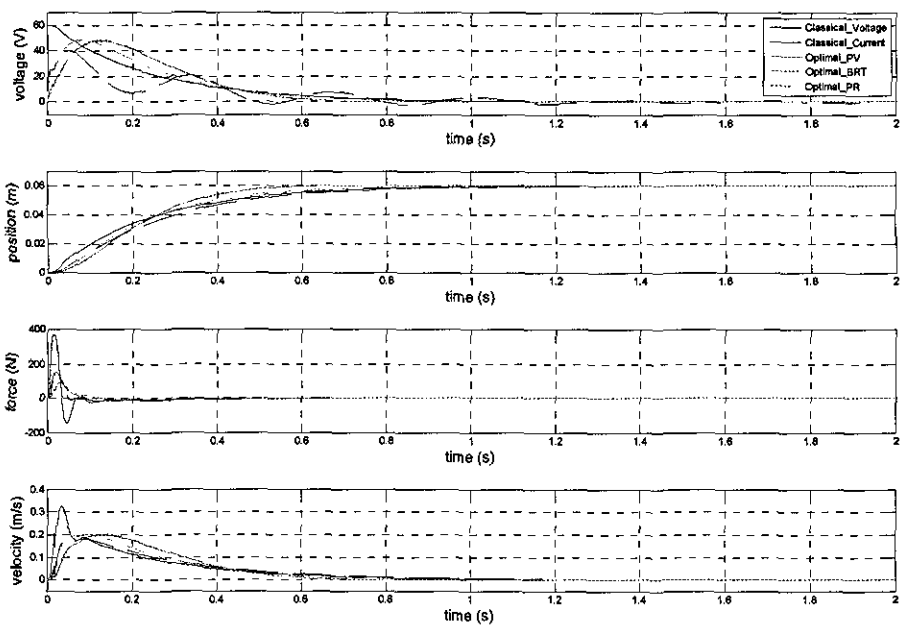


Fig. 5.4 Simulation results of all controllers with a lock-up fault

As introduced in Section 5.3, the short circuit fault is similar to the lock-up fault so that the performances look similar but bigger degradation is expected. Again, it can be found that the HRA can complete the function when fault is injected. The same performances characteristics are listed in Table. 5.4.

Table. 5.4 Time and frequency features of all controllers with a lock-up fault

Time Domain	SE(m)	RT(s)	ST(s)	OS	PV(m/s)
Classical Voltage-Driven	0	0.54	0.70	0%	0.33
Classical Current-Driven	0	0.60	0.76	0%	0.19
Optimal Full-Order	0	0.41	0.48	0.3%	0.20
Optimal Mathematical-Reduction	0	0.41	0.47	0%	0.20
Optimal Physical-Reduction	0	0.49	0.62	0%	0.20

Frequency Domain	P.M.(deg)	G.M.(dB)	BW(Hz)
Classical Voltage-Driven	88.0	24.1	0.7
Classical Current-Driven	83.8	30.0	0.6
Optimal Full-Order	72.2	29.1	1.0
Optimal Mathematical-Reduction	72.2	26.8	1.0
Optimal Physical-Reduction	80.7	29.3	0.8

Compared with the features listed in Table. 5.3 under a short circuit fault, same result can be concluded with a slower response and bigger degradation. The degradation in settling time of three optimal controllers reach about 70% while degradation of classical CV controller is about 50% and of classical CC controller is about 55%. The classical current-driven controller

still gives the slowest response, but the degradation is no longer the biggest. Again, the optimal controllers respond faster but degrade more. The classical voltage-driven controller shows a slower response but degrades less. Under frequency domain, same changes also can be found. Both phase margin and gain margin rises while the bandwidth is smaller.

5.5 Simulation Conclusion

Table. 5.5 Rise time of controllers in healthy and faulty situations

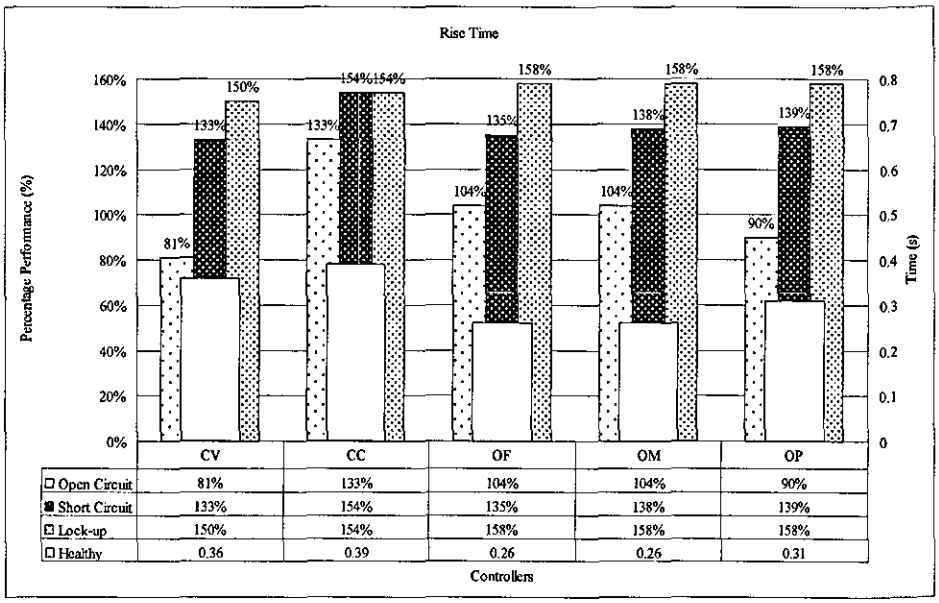


Table. 5.6 Settling time of controllers in healthy and faulty situations

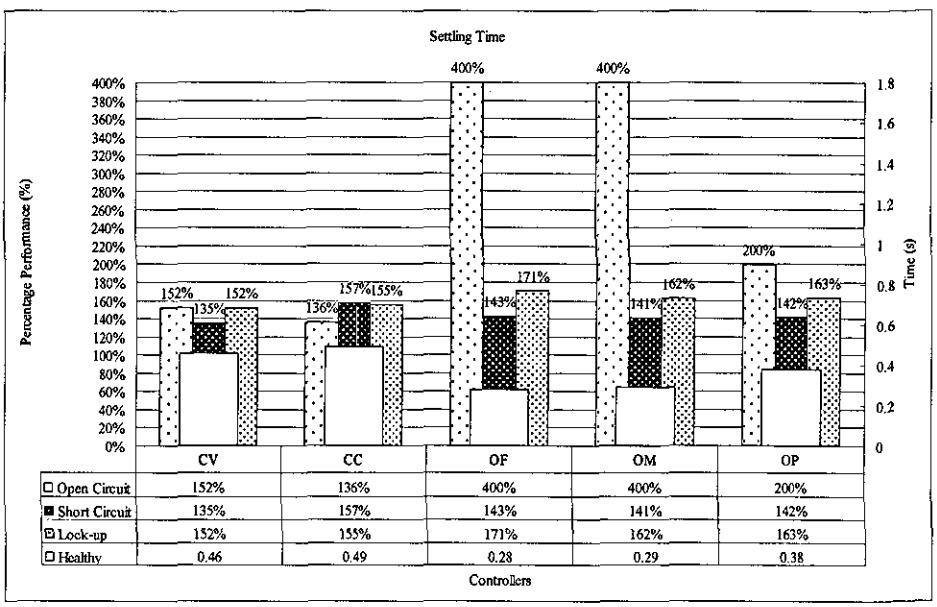
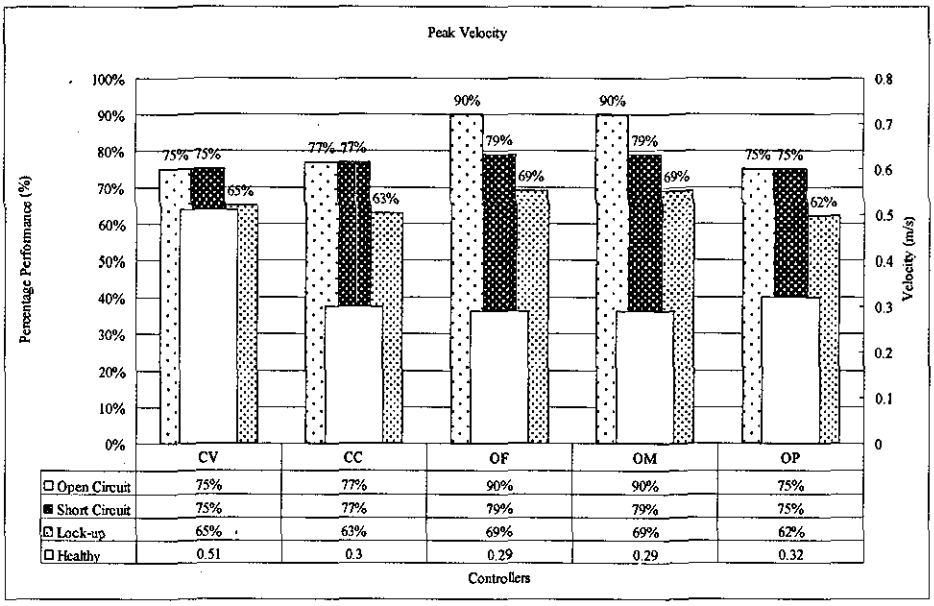


Table. 5.7 Peak velocity of controllers in healthy and faulty situations



In previous sections, all five controllers designed in Chapter 4 are tested under three kinds of fault situations. The HRA can achieve the function to reach the set point no matter which kind of fault is injected, although different level of performance degradation can be found. The degradation level in Rise Time, Settling Time and Peak Velocity are listed in Table. 5.5 to 5.7 respectively.

For the open circuit fault, the classical current-driven controller shows the smallest degradation of settling time, while optimal controllers show very big degradation. For the short circuit and the lock-up fault, the classical voltage-driven controller shows the smallest degradation of settling time. The HRA shows more degradation when a lockup fault happens, especially for the peak velocity output.

The result may be changed when using different design of controllers, but these test results demonstrate a fault tolerance ability of an HRA under some kinds of faults, and some controllers are more sensitive to some kinds of

faults with more performance degradation. More kinds of controllers could be designed and tested under more kinds and more number of faults, but these tests have not been done because of the limit of time. Based on these simulation results, the next step is to build up a real hardware demonstrator controlled in real time. Same kinds of controllers will be designed and tested under both healthy and faults injected situations. The design progress and experimental results will be shown in Chapter 6.

6. EXPERIMENTAL STUDIES

The work discussed so far has all been based on simulation in a software environment (MATLAB/Simulink). In order to demonstrate fully the viability of the HRA approach it is necessary to control a real hardware HRA. This chapter describes an experimental system that was developed by the author in order to do this. The system comprises a two-by-two series-in-parallel HRA driven and controlled in real-time by a PC running MATLAB/xPC Target operating system. The three main parts are: a control system based on xPC Target, the power driver and sensor system, and the four identical electro-mechanical actuation elements. Both classical and optimal controllers will be designed for this experimental HRA and all controllers will be tested under both healthy and faults injected situations.

This chapter includes four main sections: hardware demonstration, model validation, individual actuator experimental results and HRA experimental results.

6.1 Hardware Demonstration

In this section: first the real-time control (xPC) system will be introduced; next a test set-up comprising a single actuator element with its associated power drivers and sensors is discussed; and finally the two-by-two HRA experimental set-up is described.

6.1.1 The Real-Time Control System (xPC Target)

The real-time control system is based on MATLAB's xPC Target system [31]. To run this, two PCs are required: one working as a host PC and the other working as a target PC. The standard Windows XP operating system and MATLAB/Simulink software (including the xPC Target toolbox) are installed on the host PC, whilst the target PC boots from a floppy disk (created on the host PC) and loads MATLAB's own real-time operating system.

Table. 6.1 The PCI cards

Device	Analog I/O	Features
NI PCI-6704	16 Analog Outputs	Sending control signals
NI PCI-6024E	16 Analog Inputs	Monitoring current measurements
NI PCI-6602	8 Counters	Monitoring encoder signals

Besides the software, data acquisition (DAQ) interface devices are also needed for taking measurements from and sending control signals to the analogue world outside the PCs. The three I/O boards used here were from National Instruments and are listed in Table. 6.1. The three cards are put inside the Target PC through PCI slots on the motherboard, and each card is connected to the experiment by means of screw terminal break-out blocks.

A Target PC is a real-time processor on which compiled Matlab/Simulink C-code can be executed in real-time. The outputs measured from the experiment (via ADC and encoder) are used by the control algorithm(s) and the control actions (inputs) are applied to the experiments via the DAC card. The control algorithms (and associated code) are written and compiled on the Host PC, from where they can be downloaded to the Target PC. The Host PC also provides a means for monitoring and interacting with the real-time algorithms running on the Target. Communication between the Target and Host PC takes place over a TCP/IP network connection (so network cards were also installed in both PCs).

6.1.2 The Single Actuator Experiment

The purpose of the single electro-mechanical actuator experiment is to allow validation of the modelling and control approach for a simpler single actuator before moving on to a HRA comprising several elements.

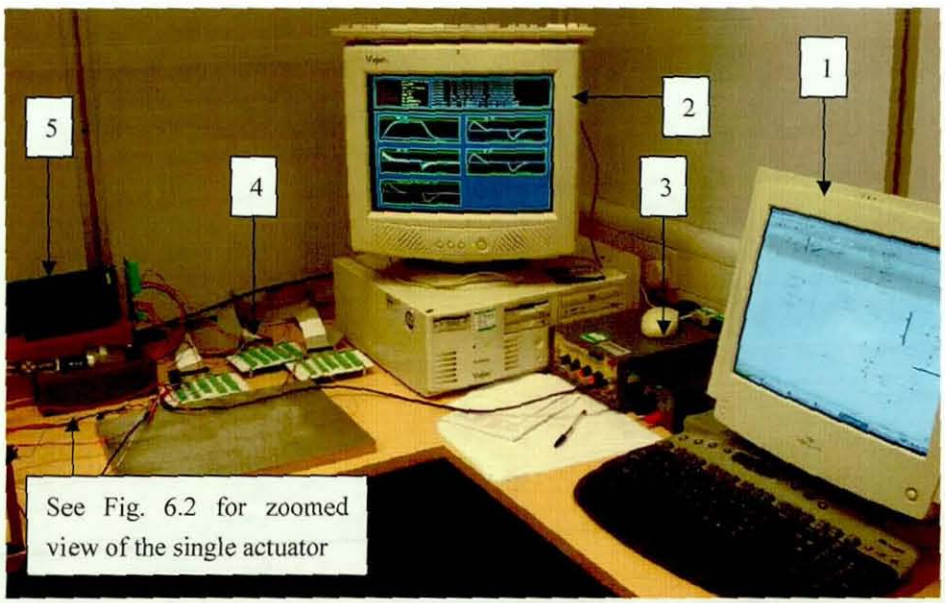


Fig. 6.1 xPC Target system and single electro-mechanical actuator

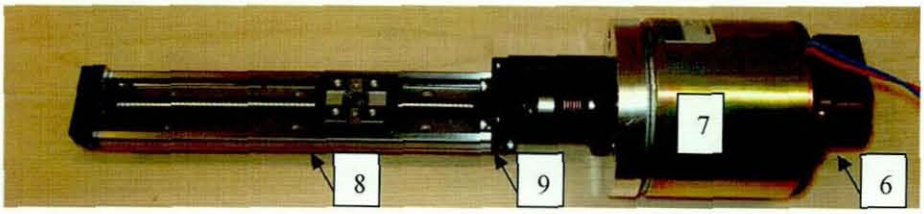


Fig. 6.2 Zoomed view of the single electro-mechanical actuator

Fig. 6.1 shows a photograph of the single actuator experimental setup. Fig. 6.2 shows a zoomed image of the actuator and motor. The key components of the experiment are labelled and will be summarised below.

1. The Host PC

This was described in Section 6.1. It allows coding of the controller and monitoring of the real-time process for offline analysis.

2. The Target PC

This was described in Section 6.1. It runs the real-time control algorithms and sends data back to the Host PC.

3. Signal Generator

This is to generate the command voltage signal or reference signal for position control.

4. Break Out Blocks

This is for connection of the experiment to the xPC Target control system.

5. Amplifier

A linear DC servo amplifier is used to drive the DC servo motor. It provides a continuous output power rating up to 60W and an amplification gain of 2.84V/V. The amplifier can also provide current measurement from of the DC motor (at 2.4V/A). Note: the amplifier has a facility for closed-loop

control using armature voltage sensing or tacho feedback, but these features will not be used here. The amplifier also needs a 24Vdc supply (a basic supply was sourced from RS and mounted in a case for safety reasons).

6. Encoder

The position information is provided by an encoder attached behind the DC motor. The encoder provides rotational shaft position for the motor. This information can be transferred to position information of the electro-mechanical actuator by taking into account the lead of ballscrew. Full technical details can be found in Appendix 1.

7. Motor

The motor is a permanent magnet DC motor rated at 24V and with a top speed of 2300rpm. Full technical details can be found in Appendix 1.

8. Ballscrew

The ballscrew and carriage assembly is a linear motion guide actuator (KR type) from THK Ltd. The ballscrew has a lead of 2mm and a 150mm range of movement. Full technical details can be found in Appendix 1.

9. Connection Adapter

An electromechanical actuator can be obtained by connecting the DC motor and the ballscrew through a connection adapter. The adapter used here is also designed by THK Ltd which is shown in Fig. 6.3

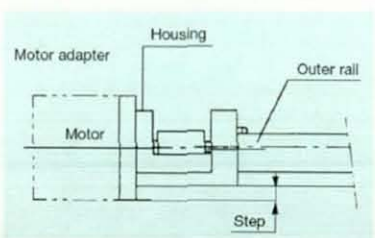


Fig. 6.3 Connection between motor and actuator

The 9 components have been introduced to set up the overall single actuator control system. The overall system can be separated into three parts as introduced in the beginning of this chapter. The control system comprises components 1 to 4; the drive and sensor system comprises components 5, 6; and the actuation system comprises components 7 to 9. The overview of an individual actuator control system is shown in Fig. 6.4.

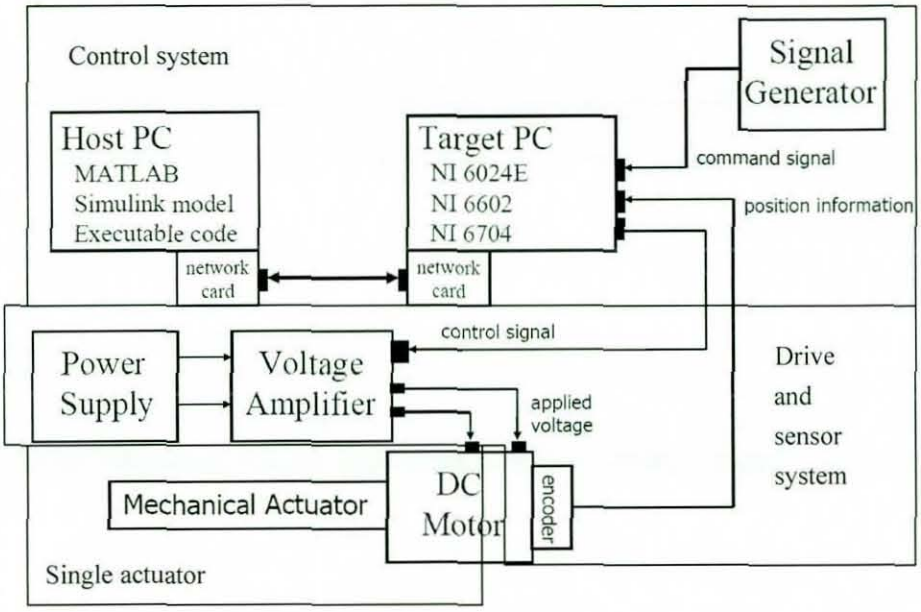


Fig. 6.4 Overview of the individual electro-mechanical actuator control system

6.1.3 The Two-by-Two Series-in-Parallel HRA Experiment

Using four of the electro-mechanical actuation elements Fig. 6.2, a two-by-two series-in-parallel HRA can be built up. This mechanical structure has been introduced previously in Chapter 2. A side view and a top view of the hardware HRA are shown in Fig. 6.5 and Fig. 6.6 separately.

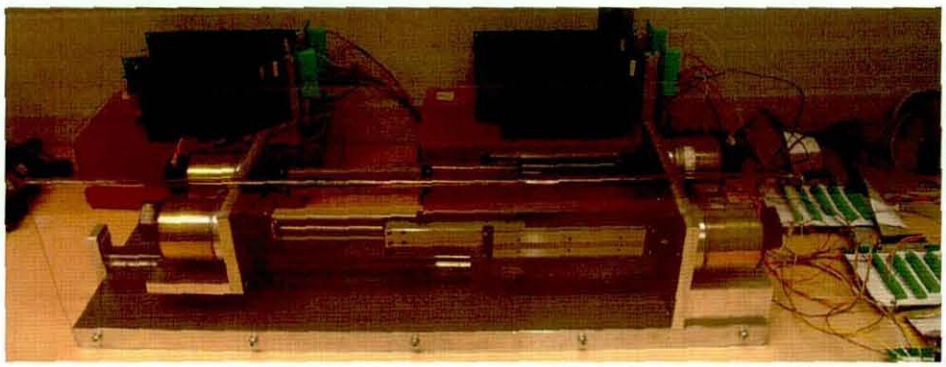


Fig. 6.5 Side view of the experimental HRA with full extension

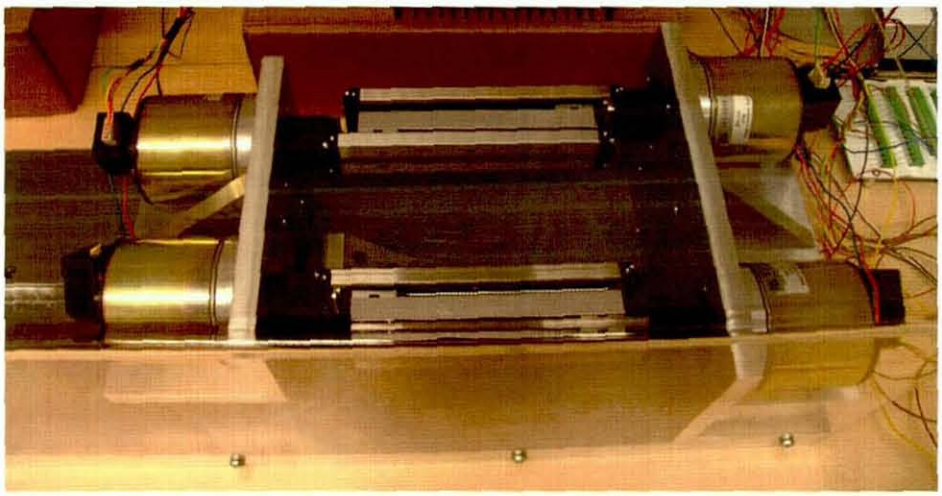


Fig. 6.6 Top view of the experimental HRA with no extension

The mechanical design drawings can be found in Appendix 2. In this design, the motors of both bottom and top actuation elements are mounted on the bottom and top bases separately. There are also two linkages between bottom and top elements to link the moving blocks to provide a separation. The top base is mounted on a rail to support the weight of the top elements and avoid shaking when moving, especially when faults are injected.

The same control, drive and sensor systems as introduced in Section 6.1.2 are used to control the HRA in real time, but the difference here is that it needs four amplifiers to drive all four actuation elements separately and four encoders to provide position information of all elements. Both the amplifier

cards and encoders can be found in Fig. 6.5. By adding the bottom and top elements' travelling distance, the position information of the HRA can be obtained.

6.2 Model Validation

Using the hardware demonstrator, it is possible to validate the controllability of a HRA. Several control strategies will be designed in a software environment based on a MATLAB/Simulink model, so that it is necessary to validate the simulation model firstly by comparing the outputs of the model with the real hardware HRA. All Simulink models' structures have been introduced in Chapter 3 and listed in Appendix 3.

In this section, after being given parameter values, linear simulation models of the DC motor and the linear individual electro-mechanical actuator will be tested firstly. Then a non-linear individual actuator model will be introduced. Finally, a non-linear two-by-two series-in-parallel HRA model will be given and compared with the hardware HRA.

6.2.1 Parameter identification

All the simulation models have been introduced in Chapter 3, but because the modelling study came before the hardware setup, the parameter values used in the previous study are not coming from the real hardware demonstrator. In this section, the model structures developed will remain but the parameters will use some real values coming from datasheets. Some parameters which are not listed, including motor's and screw's damping and stiffness coefficients, screw and load mass, are remained same as in modelling study coming from [30]. All parameter values are listed in Table. 6.2. Using these parameters and the structure given in Chapter 3, the output from the simulation models will be compared with the real actuator. The DC motor model will be tested firstly.

Table. 6.2 Parameters of a real electro-mechanical actuator

Parameter	Description	Value/Unit	Parameter	Description	Value/Unit
R_{arm}	motor resistance	7.8 Ω	L_{arm}	inductance of the windings	5 mH
K_e	voltage constant	0.0996 V/rad/s	K_t	torque constant	0.09 Nm/A
J_m	motor inertia	$2.14e-5$ Kgm^2	C_m	damping coefficient	1e-7 Nm/rad/s
K_m	motor stiffness	1e7 N/m	n	screw pitch	3.183e-4 m/rad
K_s	screw stiffness	1.8e5 N/m	C_s	screw damping	1.2e3 N/m/s
M_s	screw mass	0.03 kg	M_L	load mass	0.04 kg

6.2.2 Testing of a linear DC motor model

By applying the parameter values listed in Table. 6.2 to the linear DC motor model introduced in Chapter 3, some simulation can be tested with different load to the motor. The performance of the real motor used in the hardware demonstrator is listed in Appendix 1. Several tests are made using the linear model. A 24V DC voltage is applied to the model at time, t = 1s. The simulations are carried out under no-load conditions firstly. Then a 'rated torque' which is 12Ncm is applied to the simulation motor at time t = 5s. Finally a 'peak torque' which is 27Ncm is applied at time t = 10s. The simulation result of the linear model is shown in Fig. 6.7, compared with a datasheet performance listed in Table. 6.3.

Table. 6.3 Performance of a real motor under 25V DC

Performance @ 24V DC			
No load speed	2300rpm	Rated speed	1600rpm
Rated torque	12Ncm	Peak torque	27Ncm

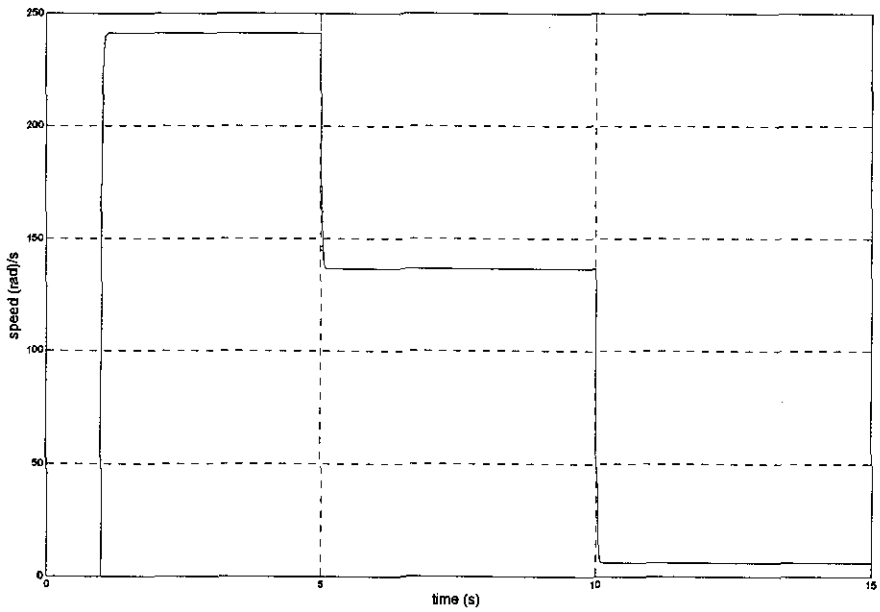


Fig. 6.7 Simulation result of a linear DC motor model

From the simulation result, it can be seen that the 'no load speed' is about 240rad/s which is about 2300rpm, the 'rated speed' is 140rad/s which is about 1400rpm, and finally it still gives about 5rad/s speed output when a 'peak torque' is loaded. Comparing with real performance of the DC motor given by datasheet, there is a little difference between the simulation result and the listed performance. That could be because of the errors of the parameters' values or some unmodeled dynamics of the real motor including some nonlinear parts. But since the difference between the model and the real performance is not very big, the simulation model can be used to represent the real DC motor.

6.2.3 Testing of a linear individual actuator model

Different from the DC motor, the performance of the real electro-mechanical actuator is not given from the datasheets because the actuator is built up by

connecting a DC motor and a mechanical actuator ordered separately rather than ordered as a whole product. The real actuator is driven by the xPC Target system as introduced in Section 6.1. An xPC Target model is then built (listed in Appendix 4) including both the linear simulation model and analog I/O blocks, i.e. the PCI cards. A 5V square voltage signal is applied to both simulation model and real electro-mechanical actuator through the amplifier. The position and velocity performances are shown in Fig. 6.8

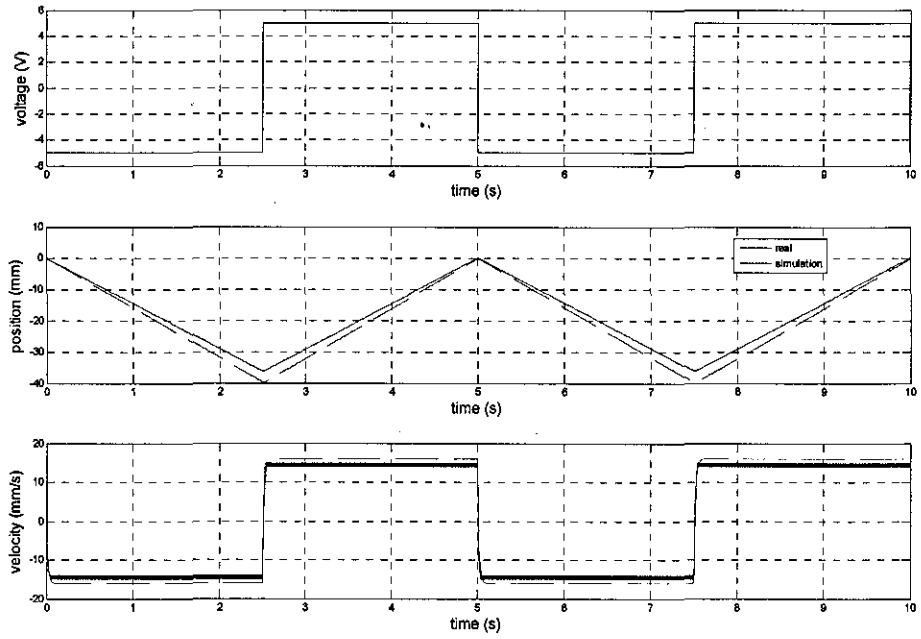


Fig. 6.8 Position and velocity outputs of the linear simulation model and the experimental individual actuator

Both the simulated and real results have been shown. From the figure, it can be seen that the simulated actuator moves a little faster than the real one. The reason for the difference could be errors of the parameters' values and some non-modelled dynamics. A non-linear effect which is not modelled is the friction in the mechanical actuator part, because there will be some friction between the moving block and the outer rail. This non-linear friction part is then will be considered to create a more accuracy non-linear simulation model of the individual electro-mechanical actuator.

6.2.4 Testing of a non-linear individual actuator model

To model the friction, an extra torque load can be added to the DC motor. The torque value is chosen same as the starting torque of the mechanical actuator given by Technical specification shown in Appendix 1.The direction of the friction is opposite to the moving direction of the actuator so that the sign of the friction torque also need to be changed. The non-linear individual actuator model is shown in Fig. 6.9

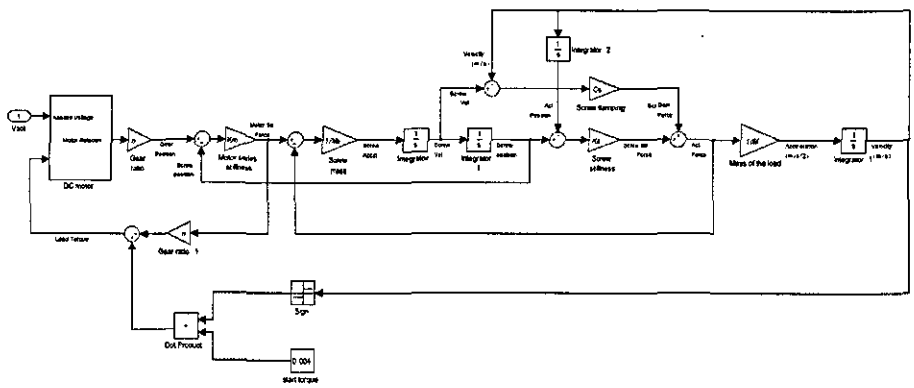


Fig. 6.9 Individual electro-mechanical non-linear model

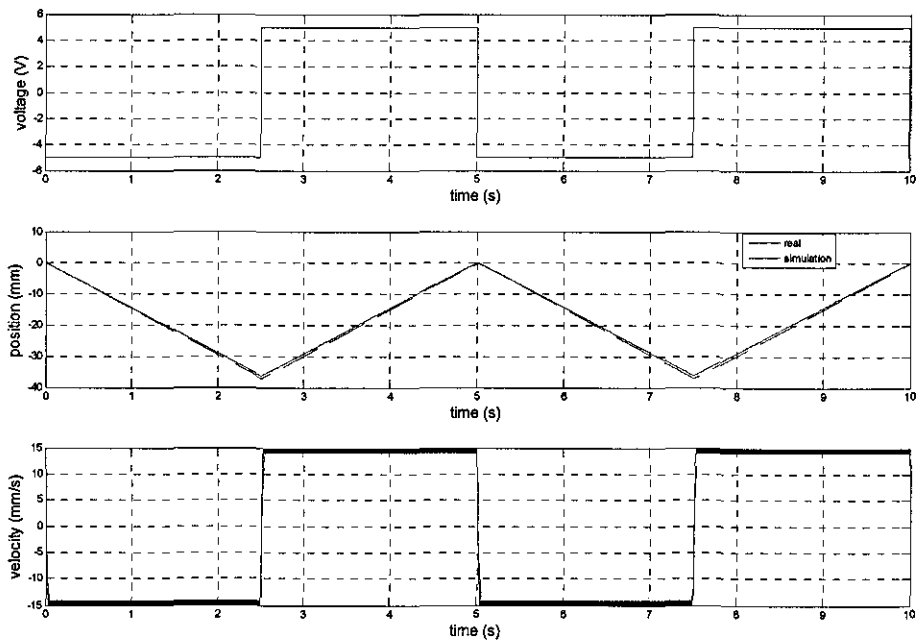


Fig. 6.10 Position and velocity performances of the non-linear model and the real individual actuator

The non-linear model is compared with the hardware actuator in real time with a 5V square voltage signal using the same xPC Target testing model. Both position and velocity performances have been shown in Fig. 6.10. The simulated actuator performs very similar with the real actuator when friction is considered in the model. This non-linear model therefore represents the real individual electro-mechanical actuator very well.

6.2.5 Testing of a non-linear HRA model

A HRA model is built up by connecting the appropriate individual actuator models. A two-by-two series-in-parallel HRA model has been introduced in Chapter 3 and listed in Appendix 3. By adding the friction part into each element's model, a non-linear HRA model can be obtained. Also an xPC Target model (listed in Appendix 4) is created to control the hardware HRA and compare the performance with a simulated HRA in real time. The three PCI cards are still used to transfer signal between the PC and real world, but four different channels are used for four individual elements. The overall position of the HRA can be obtained by adding the position outputs of all four elements and dividing by two, which can help to reduce the measurement errors.

The actuation elements can be controlled separately by applying separate voltage to each element, although some elements will be pulled or pushed in some situations because they are linked together. Two experiments are tested. Firstly, a control signal using same voltage is applied to each actuation elements which is 3V. The overall position and velocity are shown firstly in

Fig. 6.11, and the bottom and top actuation elements' information in Fig. 6.12 and Fig. 6.13 respectively.

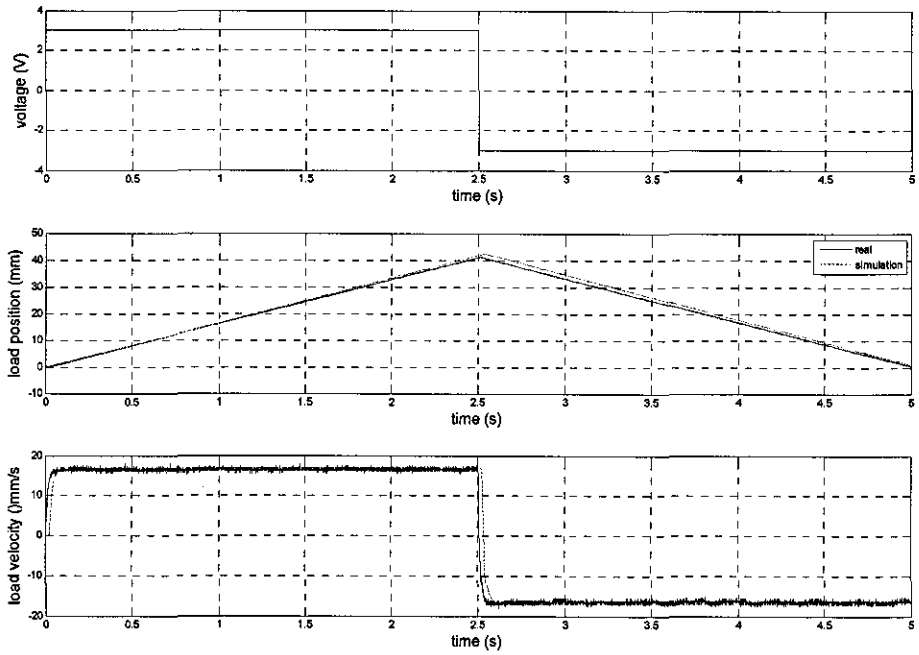


Fig. 6.11 Performances of a two-by-two series-in-parallel HRA with same voltage

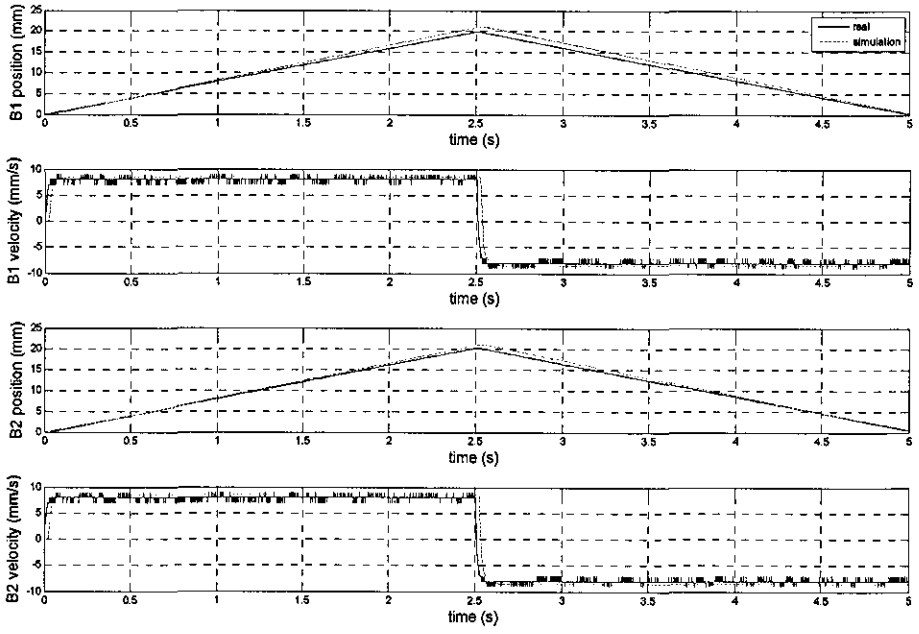


Fig. 6.12 Performances of bottom elements with same voltage

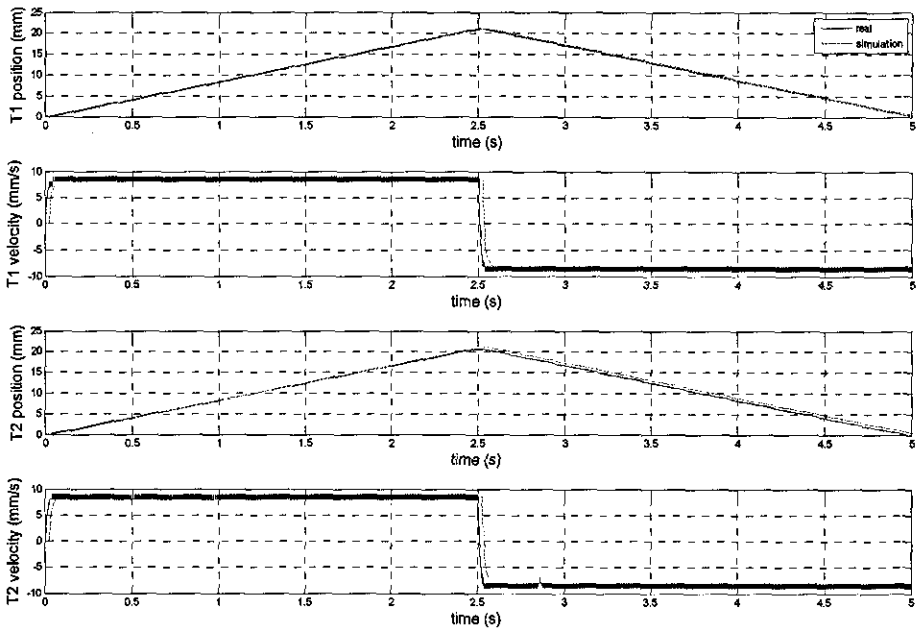


Fig. 6.13 Performances of top elements with same voltage

When same voltage is applied to the elements, they perform very similar no matter where they are located inside the HRA. Also the simulation results match the real actuator's performance very well.

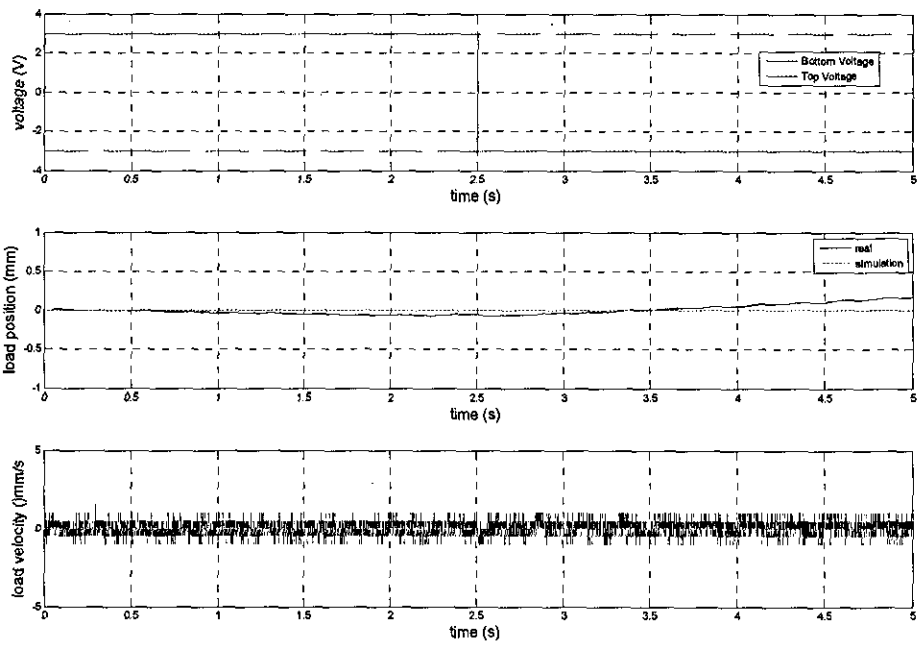


Fig. 6.14 Performances of a two-by-two series-in-parallel HRA with inverse voltage

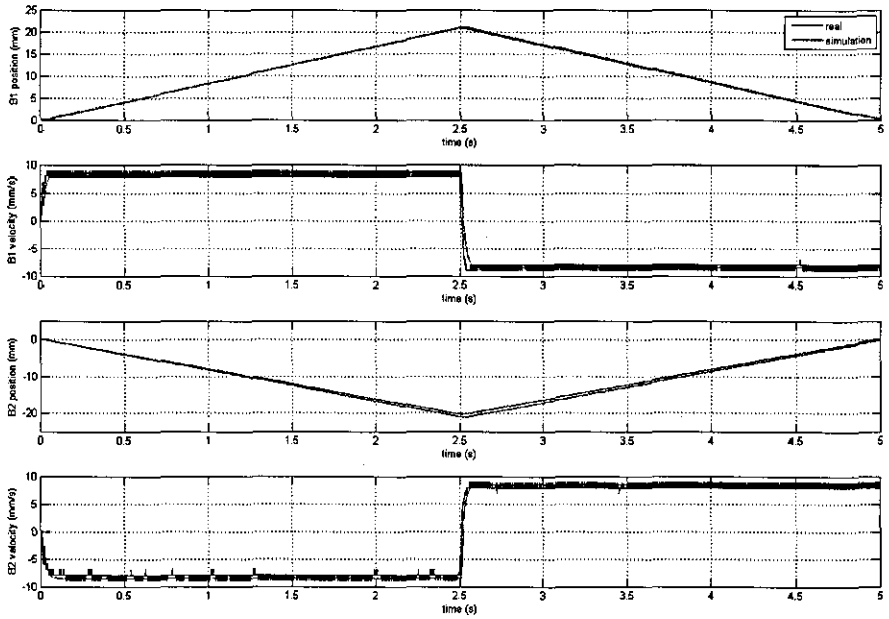


Fig. 6.15 Performances of bottom elements with inverse voltage

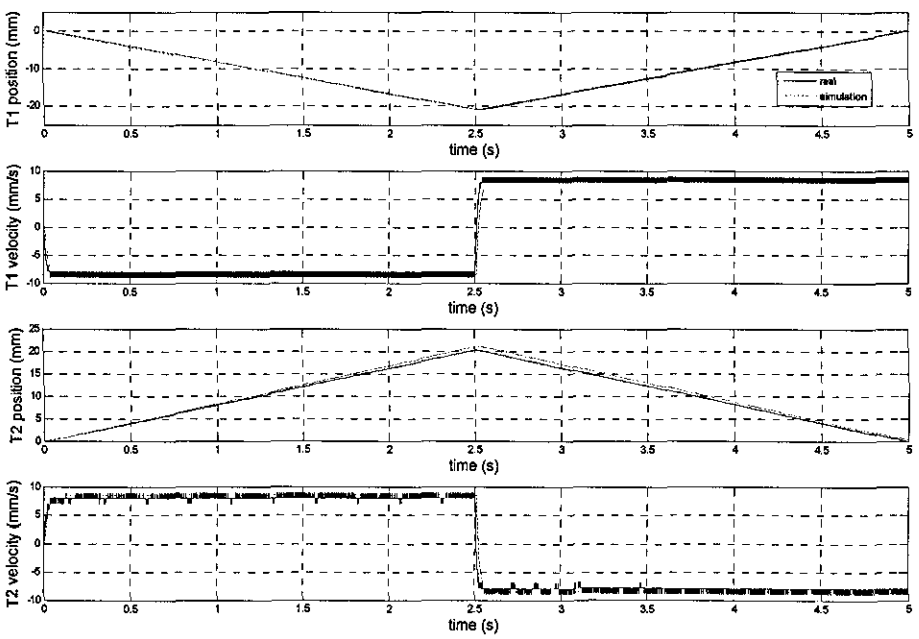


Fig. 6.16 Performances of top elements with inverse voltage

In the second test, the elements have different applied voltages. The bottom voltage shown in Fig. 6.14 is applied to Bottom 1 and Top 2 actuation

elements, while the top voltage is applied to Bottom 2 and Top 1 elements. Performances of the HRA and each element are shown in Fig. 6.14 to Fig. 6.16 separately. Because inverse voltages are applied to the two sets of elements, the overall output is zero. The result can be expected and predicted from the simulation model. This proves that the actuation elements inside the HRA can be controlled separately, and the overall performance is the sum of the outputs of the two elements in series. From both tests, it can be said the non-linear simulation model represents the experimental HRA hardware well.

6.3 Experimental Results - Individual Actuator

As shown in Section 6.2, the non-linear model can give excellent prediction of the real hardware's behaviour and the linear version, likewise, showed that the main dynamic models were captured with only minor discrepancies due to friction. These two models can now be used in control design. The linear model will be used to design the controller and the non-linear one to test performance prior to application to the real actuator.

In this section, a series of controllers will be designed and applied on the experimental individual actuator so that the control design can be extended to the hardware HRA in the next section. The classical loop shaping method will be introduced firstly and then the optimal control method will be applied. Both controllers are designed for position control and the design progress has been introduced in Chapter 4.

6.3.1 Voltage-driven classical controller

A 30mm square signal is applied as a reference, and the purpose of the system is to track the reference signal. A proportional controller is applied with a gain of 500V/m. Using this controller, the open-loop phase margin is 88.2deg and the gain margin is 54.6dB. The closed-loop bandwidth is about 0.25Hz. The controller is then applied to the non-linear individual actuator Simulink model. The purpose is to make sure the controller will work when applied to the real actuator and will not cause any damage to the hardware.

After being verified, the controller is then applied in the real world through an xPC Target model. The experimental results comparing with the non-linear model are shown in Fig. 6.17.

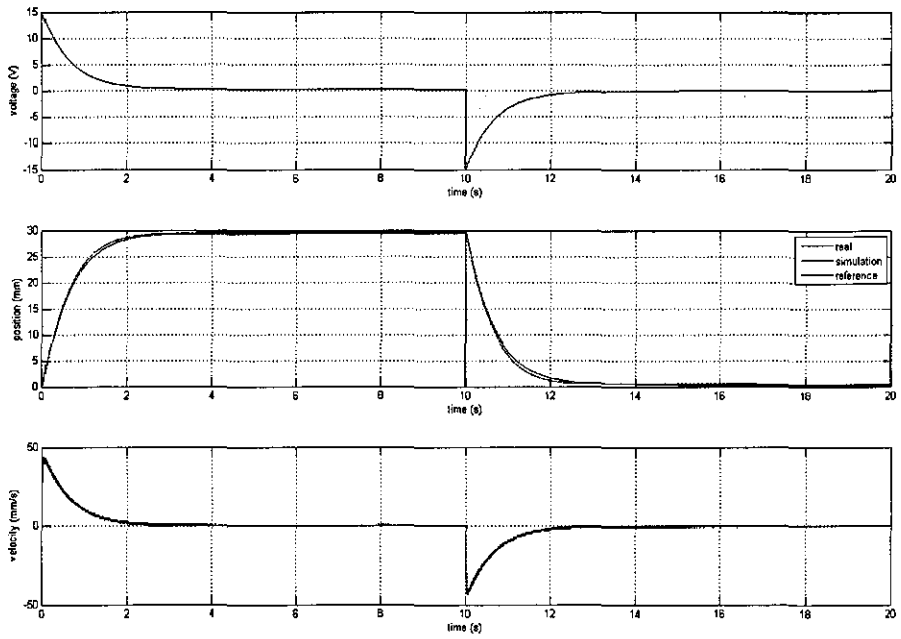


Fig. 6.17 Performance of an individual actuator with a voltage-driven classical controller

Firstly, it can be seen that the simulated results are very close to the experimental ones which proves again the validity of the non-linear model. The position information shows that the real actuators can track the reference signal well although there exists a little steady state error. The reason is the friction between the moving block and out rail. When the moving block is near the set-point, the control voltage will become smaller than the value which is need to overcome the friction so that the actuator will not reach the set point exactly. The method to overcome the friction is to increase the control voltage. The simplest way is to increase the controller's gain, but this will lead to changes in dynamics of the whole system in both time and frequency domain. The other way is to introduce a Coulomb and Viscous Friction block added at the output of the controller. The block

models a discontinuity at zero and a linear gain otherwise. The offset corresponds to the Coulomb friction: the gain corresponds to the viscous friction. The block is implemented as $y = sign(u) * (Gain * abs(u) + Offset)$, where y is the output, u is the input, and Gain and Offset are block parameters. Here, the gain is chosen to be unity. By adding an offset, the controller output is increased automatically so that the compensated control signal can drive the actuator to overcome the friction. The offset needs to be chosen carefully because it will lead to oscillations of the actuator if the offset is too big, but otherwise a small value can not make the control voltage overcome the friction. The theoretical value is the minimal voltage for the actuator to start moving. Here the offset is chosen as 0.1V and the steady state error is reduced from 0.5mm to 0.3mm. Although the error is not avoided completely, the effect can be seen.

6.3.2 Current-driven classical controller

A current feedback inner loop is then added into the voltage-driven controller. A proportional controller with a gain of 10V/A is used for the current loop. The closed loop bandwidth of the current loop is about 550Hz which is too high to be applied on the hardware. Then a Butterworth low pass analog filter is applied to the current close loop to reduce the bandwidth to about 200Hz. For the position outer-loop, a proportional controller with a gain of 50A/m is applied. The open-loop phase margin is 86.1deg, and gain margin is 72.2 dB. The closed-loop bandwidth is about 0.25Hz. An xPC Target model (listed in Appendix in 4) is used to apply this controller to the hardware actuator. A 30mm square signal is applied as reference of the actuator. The experimental result compared with non-linear simulation model is shown in Fig. 6.18.

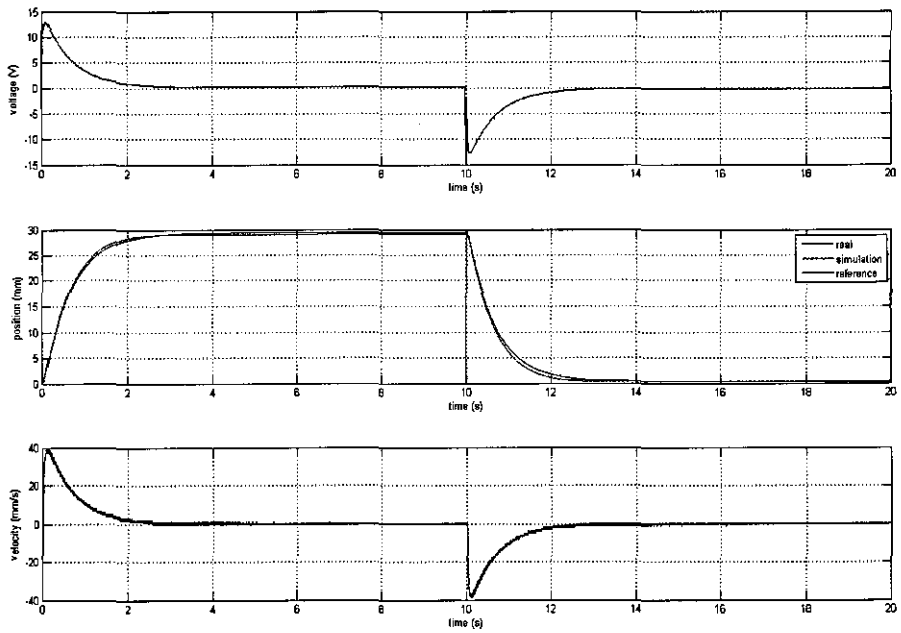


Fig. 6.18 Performance of an individual actuator with a current-driven classical controller

Both non-linear model and real actuator can follow the reference signal well, although there is still a little steady state error because of the friction. A Coulomb & Viscous Friction block to avoid the steady state error is added before the current loop. The gain is chosen as unit, and the offset value is chosen as 0.04. The steady state error has almost been avoided

6.3.3 Voltage-driven optimal controller

The control design is then extended to an optimal control algorithm. A state-space model-based LQG controller is designed as introduced in Chapter 4. An xPC Target model is created for the real-time control, including a seven order Kalman Filter design using velocity and position measurements, plus an LQR design with an extra integral of position error state. A discrete time method is used here with a sampling time of 0.001s to avoid the CPU overload. A 30mm step signal is applied. The experimental

results of both real and non-linear simulation model are shown in Fig. 6.19.

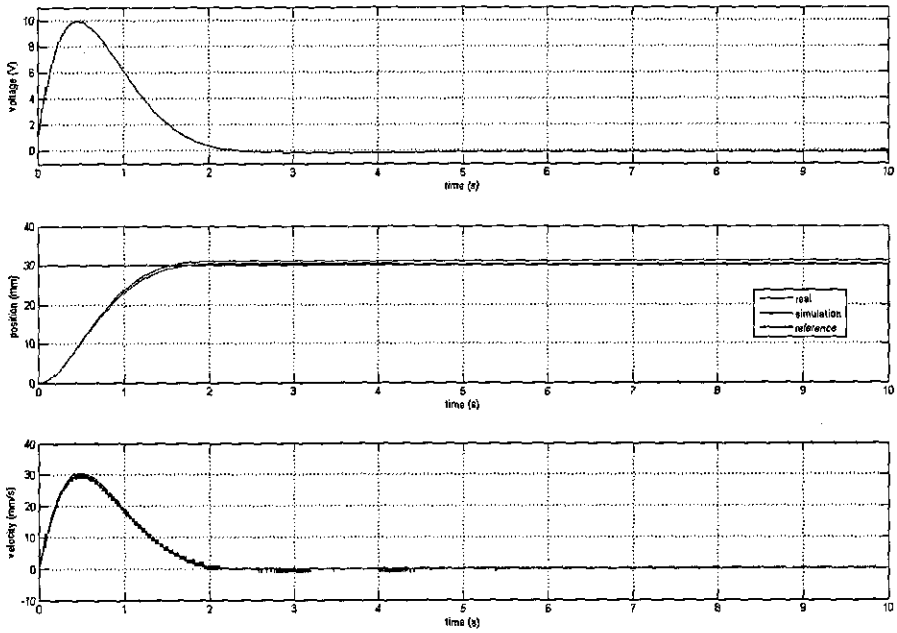


Fig. 6.19 Performance of an individual actuator with a discrete-time LQG controller

In time domain, both non-linear model and real actuator can track the reference well with a very small steady state error. In the frequency domain, the open-loop phase margin is 68deg, and the gain margin is 48.1dB. The closed-loop bandwidth is about 0.26Hz.

To use the continuous model, a decreased order state-space model has to be used due to processing limitations. Using the mathematical model reduction method called balanced truncation as introduced in Chapter 4, a state-space model with three states is obtained from the full states model. Performances of both non-linear model and real actuator to track a 30mm step signal are shown in Fig. 6.20.

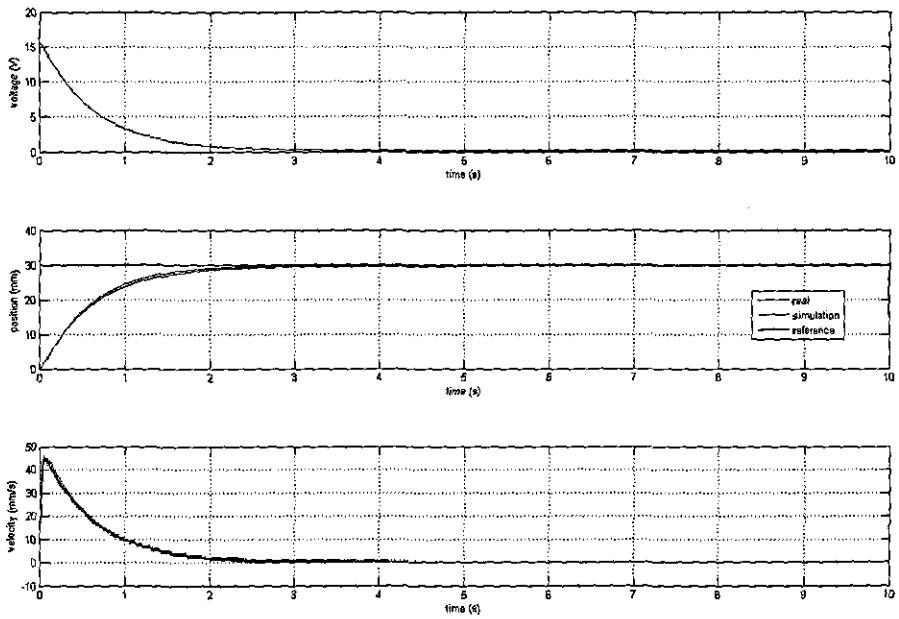


Fig. 6.20 Performance of an individual actuator with a continuous-time LQG controller

It can be found that both actuators track the reference well with a very small steady state error. The open-loop phase margin is 87deg, and the gain margin is 55.5dB. The closed-loop bandwidth is 0.37Hz.

6.3.4 Comparison of individual actuator controllers

The four controllers described above are now compared under both time domain (in Fig. 6.21) and frequency domain (in Table. 6.4)

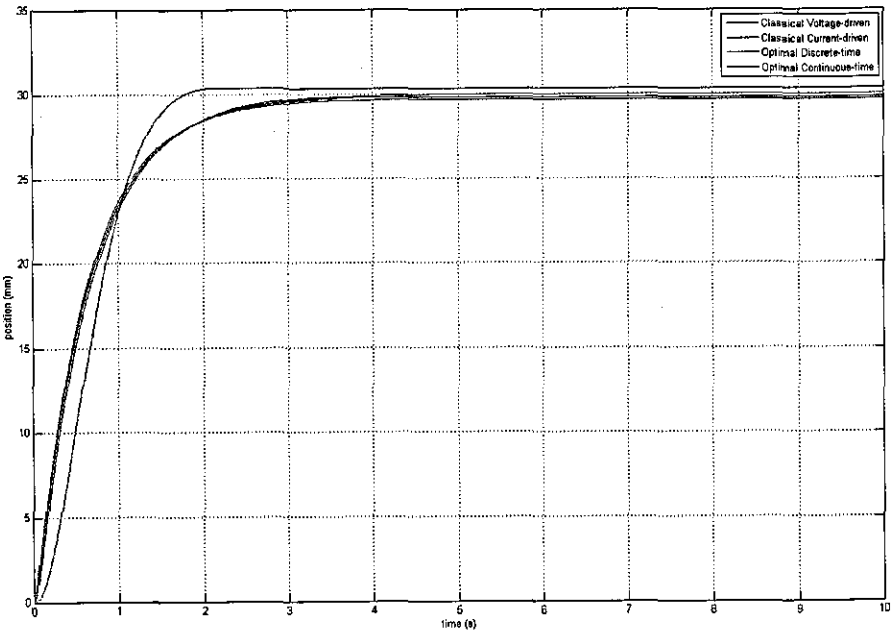


Fig. 6.21 Position output of an individual actuator with four kinds of controllers

Table. 6.4 Frequency features of an individual actuator with four kinds of controllers

Controllers	PM(deg)	GM(dB)	BW(Hz)
Classical Voltage-driven	88.2	54.6	0.25
Classical Current-driven	86.1	72.2	0.25
Optimal Discrete-time	68	48.1	0.26
Optimal Continuous-time	87	55.5	0.37

Fig. 6.21 summarizes the step response of the real individual actuator to a 30mm step signal for the four controllers. Table. 6.4 summarizes three frequency domain features including open-loop Phase Margin (PM), Gain Margin (GM), and closed-loop Bandwidth (BW). These four controllers were designed to perform very similar. Then similar control design progress will be applied to the two-by-two series-in-parallel HRA.

6.4 Experimental Results - HRA

Having validated the model and demonstrated control of an individual actuator, the next step is to extend the control design to the two-by-two series-in-parallel HRA. Both classical and optimal controllers will be designed and tested with healthy and fault injected situations in real time to verify the fault tolerant capability of the HRA. Using the same design progress as the individual actuator, the controller is firstly designed using linear simulation model, then verified on the non-linear model and finally used to control the hardware HRA in real time using the xPC Target system.

Position feedback control is used here. Classical controllers including both voltage-driven and current-driven structures are designed firstly. Optimal controllers including full order and order reduction designs are then applied. A low speed design is applied firstly to avoid damage to the hardware, but only a typical classical voltage-driven control result is shown here. More detailed experimental results will be introduced using a fast control design.

6.4.1 Slow design controller

A proportional voltage-driven classical controller is designed with a gain of 300V/m for the position control to track a 60mm step signal. Each element inside the HRA is controlled separately using different channels of the DAQ cards although the control signal is the same. A xPC Target model is given in Appendix 4. Using this design, the frequency domain performance is listed following: Gain Margin: 41.1dB, Phase Margin: 87.8deg, Closed-loop Bandwidth: 0.31Hz. The time domain results of the HRA are given in Fig. 6.22 and results of each element are shown in Fig. 6.23.

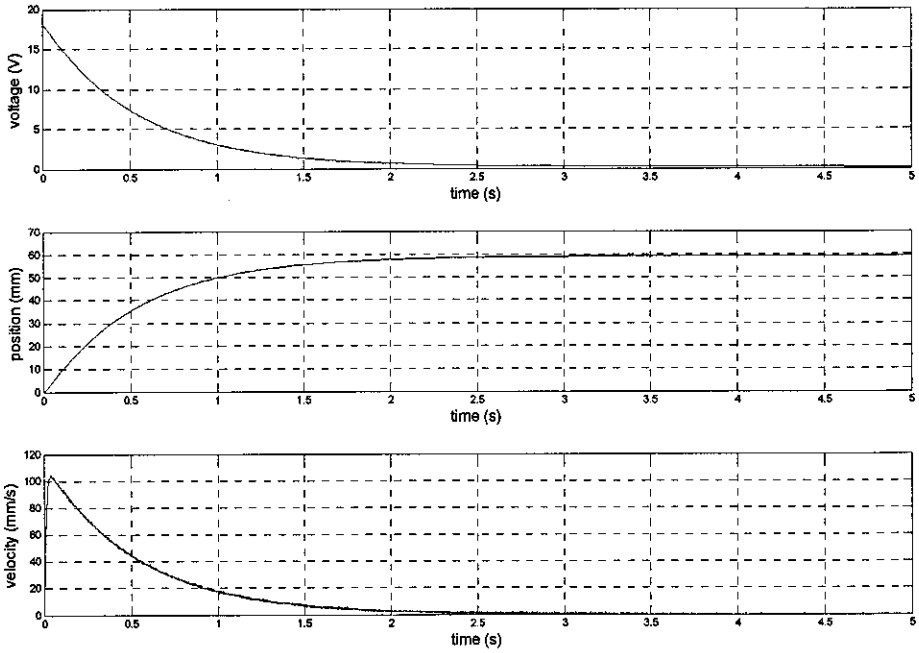
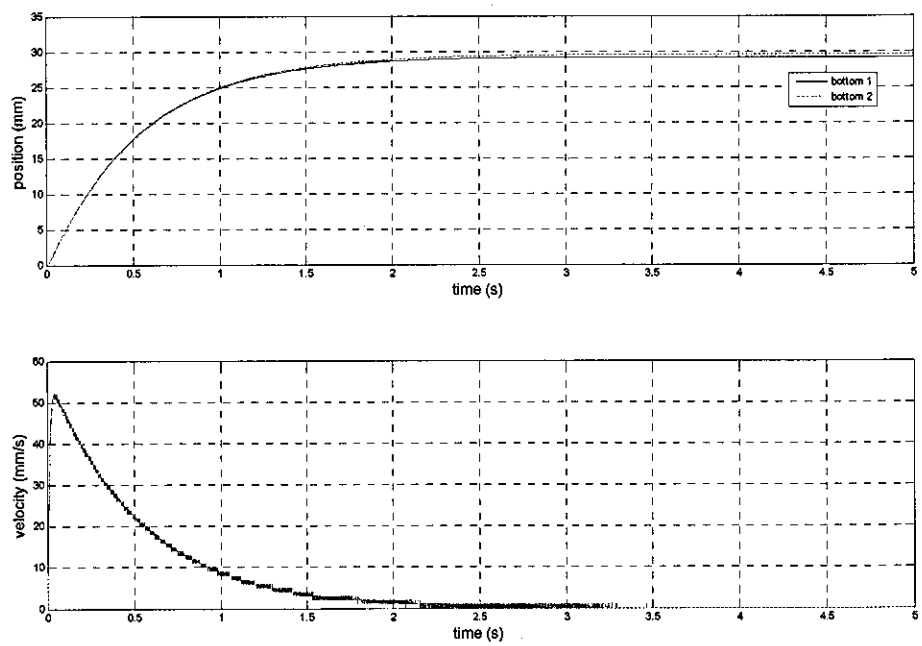
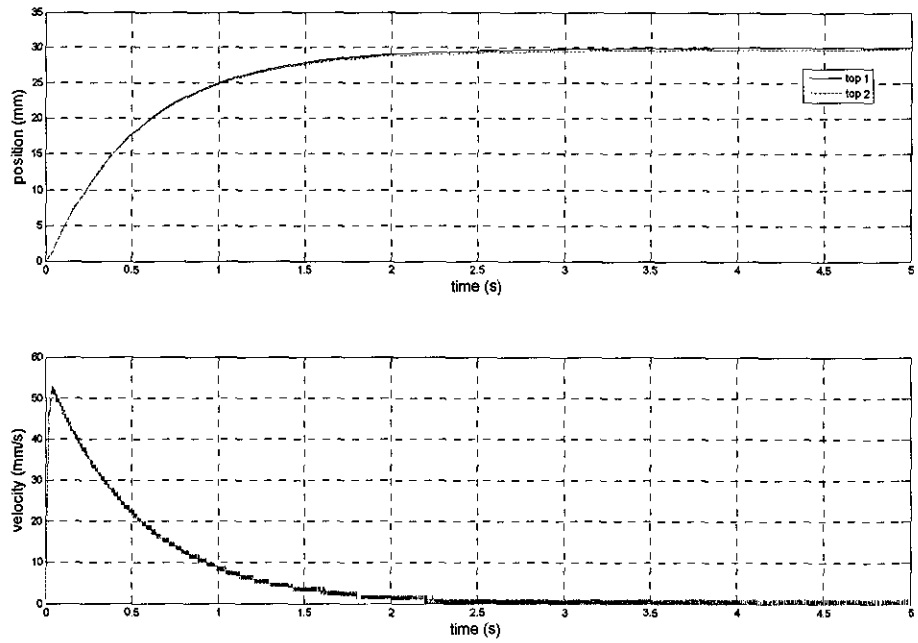


Fig. 6.22 Performance of the HRA with a voltage-driven classical controller



Bottom Elements



Top Elements

Fig. 6.23 Performance of individual elements in the HRA with a voltage-driven classical controller

Looking at the HRA results, overall position can reach the set point with negligible steady-state error due to the non-linear friction. Looking at details about the four elements' results, they have similar performance which is about half of the overall performance. Based on this slow control design's result, a series of controllers using a faster design to track a 30mm step signal will be presented.

6.4.2 Voltage-driven classical controller

A PI controller will be applied for classical control designs. The proportional gain is chosen as 1000V/m and the integral gain is chosen as 5000V/m. There is also a limitation band from -30V to 30V, which is the maximum applied voltage of the DC motor inside the actuation element, to avoid overload the actuation elements. Results of both the overall HRA and each

element will be shown in Fig. 6.24 and Fig. 6.25 respectively.

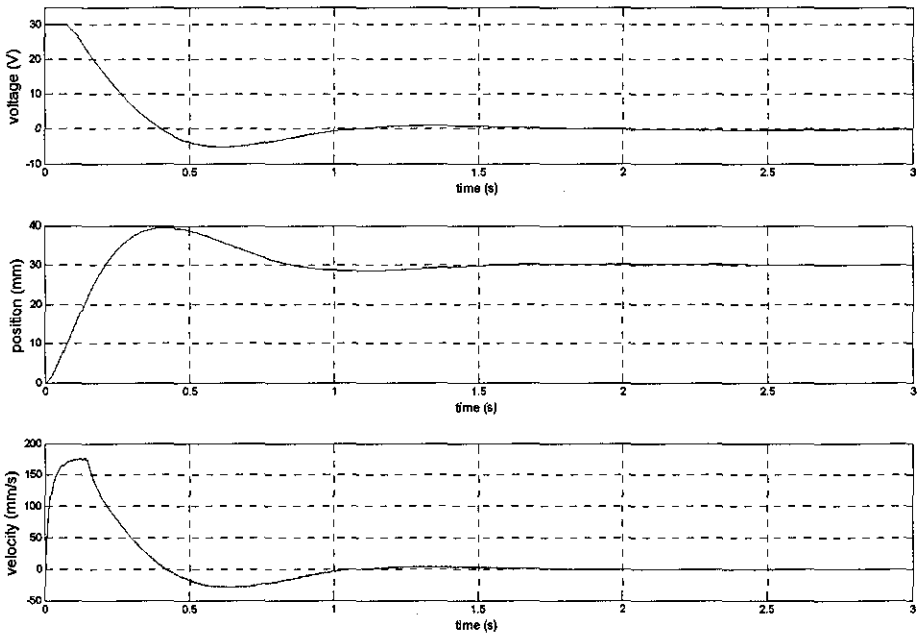
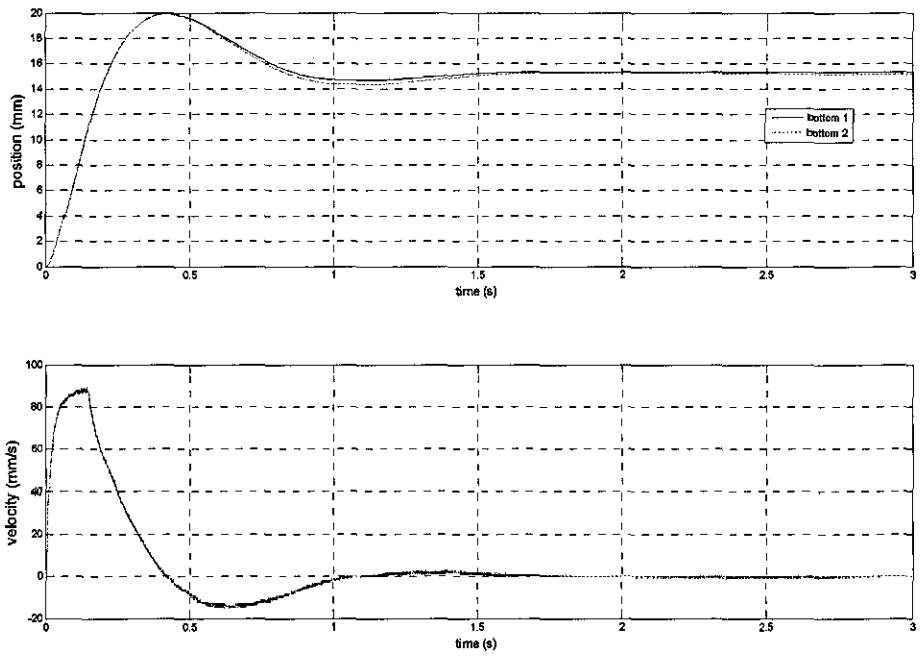
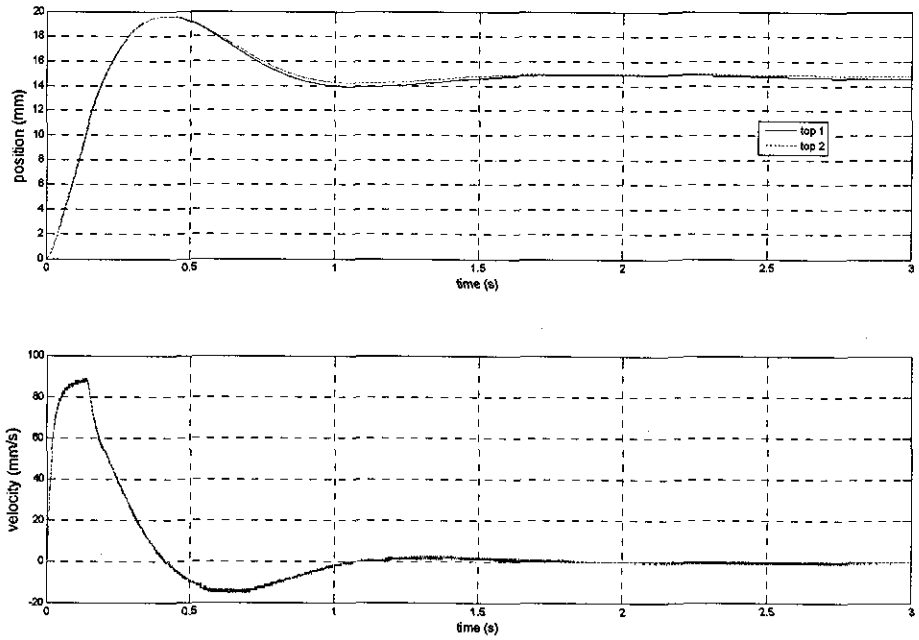


Fig. 6.24 Performance of an HRA with a voltage-driven classical controller



Bottom Elements



Top Elements

Fig. 6.25 Performance of individual elements in HRA with a voltage-driven classical controller

With the high gain of both proportional and integral actions, it can be seen that steady state error can be avoided but a 30% overshoot appears. At the control signal, it is limited at 30V at the beginning. For actuation elements, all four elements perform very similar. An overshoot also can be found for each element.

6.4.3 Current-driven classical controller

Same current loop controller used for individual actuator (Section 6.3) is also applied as local controllers in this control structure. An PI controller with a proportional gain of 3000A/m and the integral gain of 1000A/m for the PI controller is used for the out-loop position feedback global controller. Again the control signal is limited at 30V to avoid overload. Results will be shown in Fig. 6.26 and Fig. 6.27.

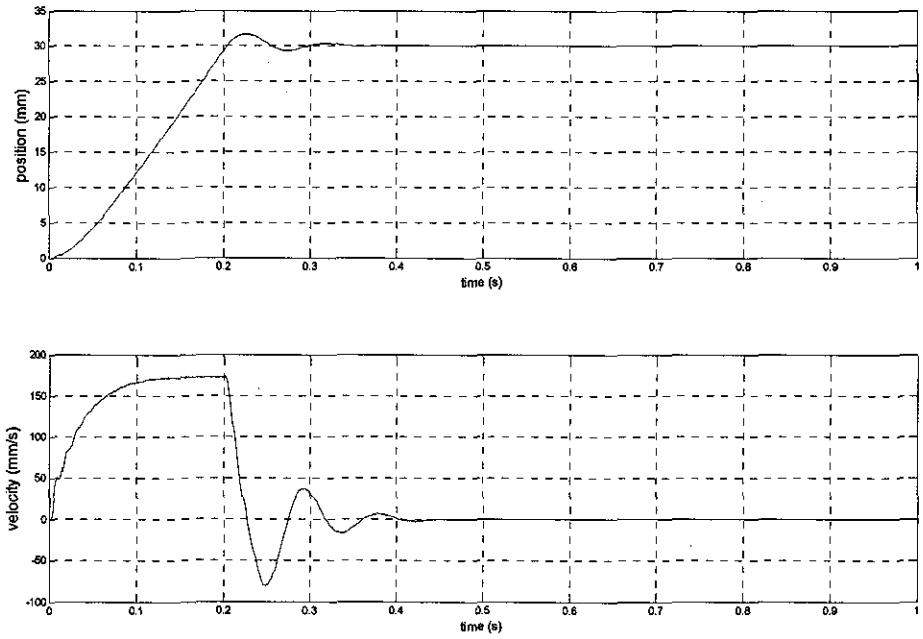
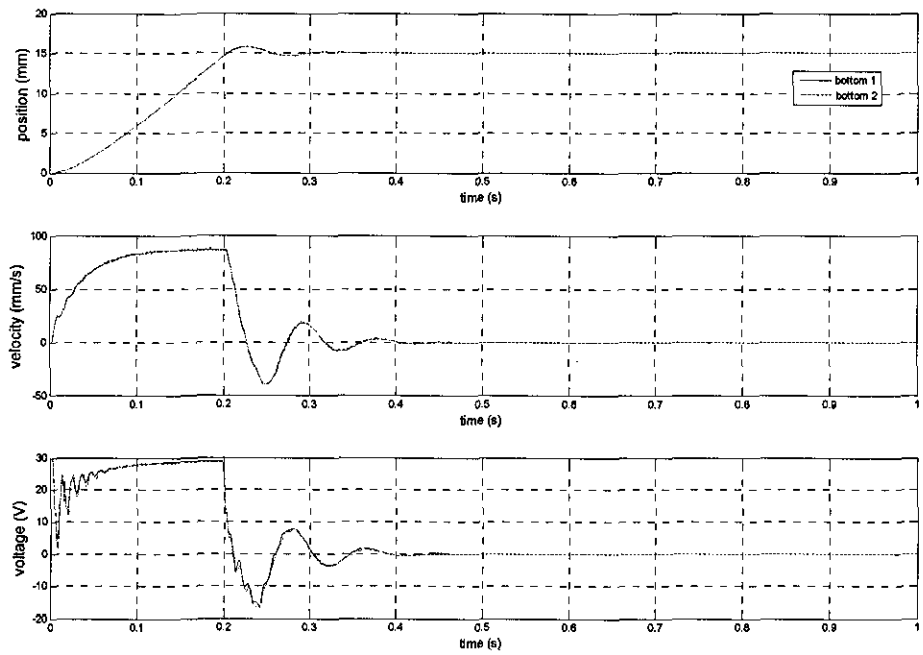
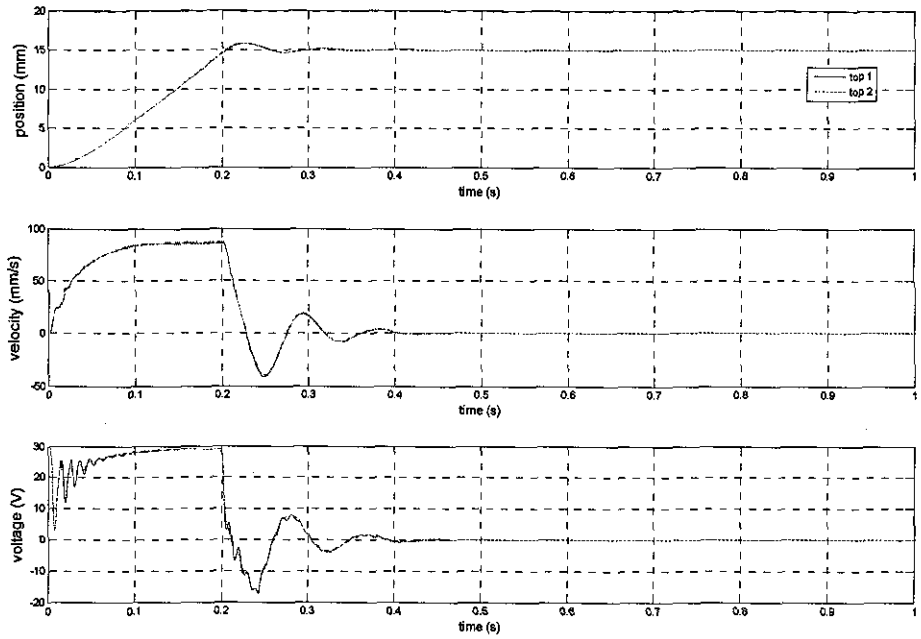


Fig. 6.26 Performance of an HRA with a current-driven classical controller



Bottom Elements



Top Elements

Fig. 6.27 Performance of individual elements in HRA with a current-driven classical controller

Fig. 6.26 shows the overall performance and Fig. 6.27 shows the individual elements'. Comparing with the voltage-driven control structure, the current-driven control design provides a faster response with less overshoot. This design has fully used the actuator because the control signal has reached the maximum voltage of the individual actuation element. The optimal control design also will use the full ability of all actuation elements inside the HRA.

6.4.4 Full order LQG controller

As introduced in Chapter 4, the repeated dynamics will cause mathematic problem when using MATLAB to calculate the control regulator. The two method used are to add parameter variations and to reduce the model's order. The full order design is introduced in this section by adding five percent variations of each parameter to avoid repeated dynamics in the state-space

model. Same as the individual actuator design, a discrete time design is applied.

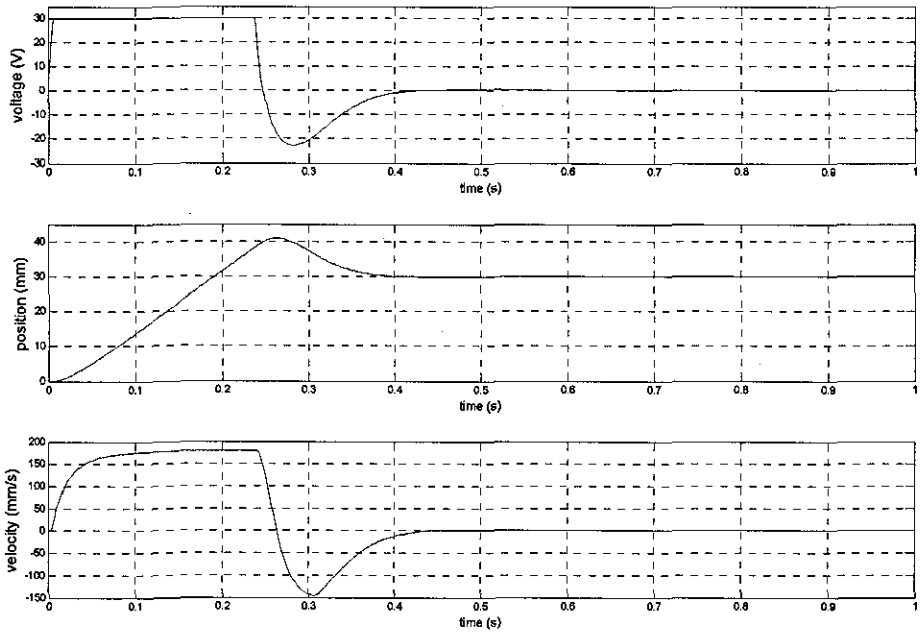
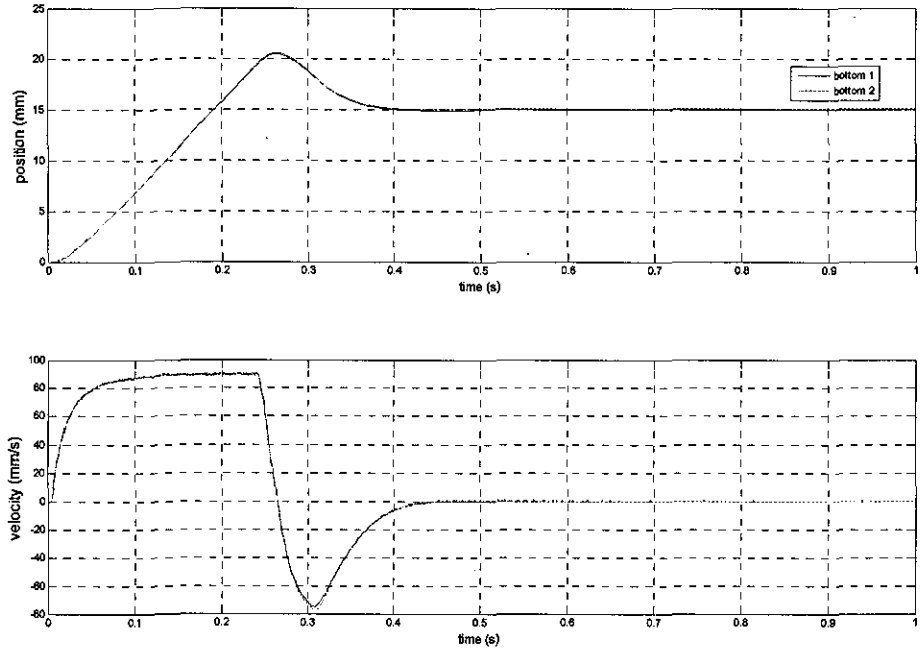
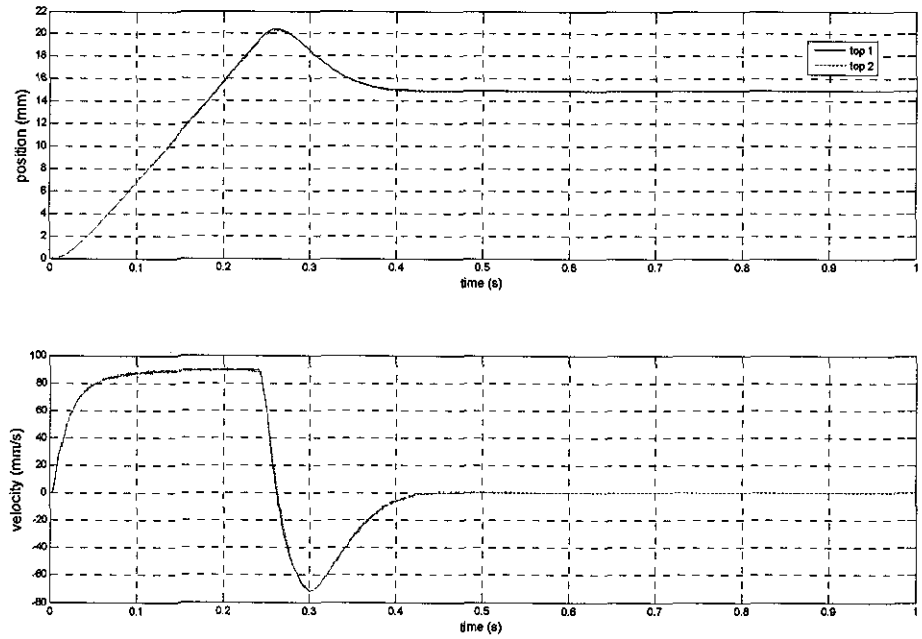


Fig. 6.28 Performance of an HRA with a full order LQG controller



Bottom Elements



Top Elements

Fig. 6.29 Performance of individual elements in HRA with a full order LQG controller

Results of both the overall HRA and each element have been shown in Fig. 6.28 and Fig. 6.29 respectively. Steady state error is reduced to zero due to the high gain in the controller. A fast response can be seen at the position output with a 30% overshoot. The control input has been limited at 30V to avoid over drive of actuation elements.

6.4.5 Mathematical reduction LQG controller

Model reduction methods are also applied to avoid repeated dynamics in state-space model. The mathematical method called balanced realization truncation is firstly used. Eight states are remained. Results of both the overall HRA and each element have been shown in Fig. 6.30 and Fig. 6.31 respectively.

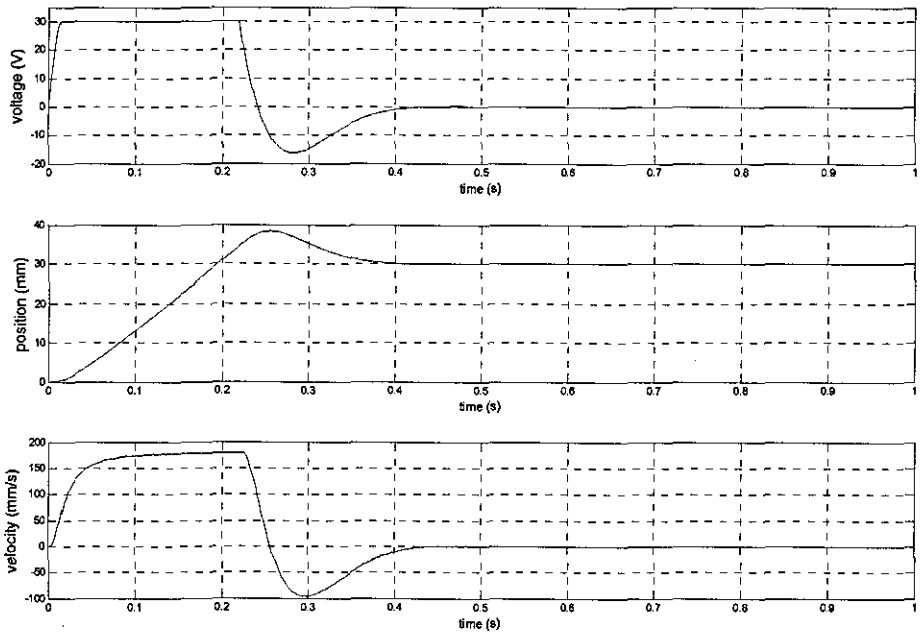
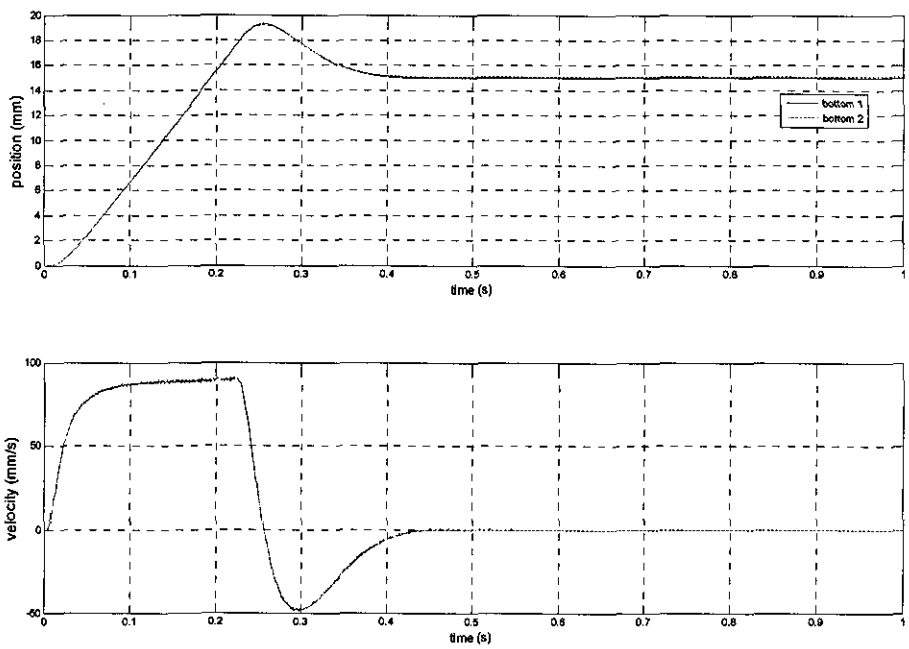
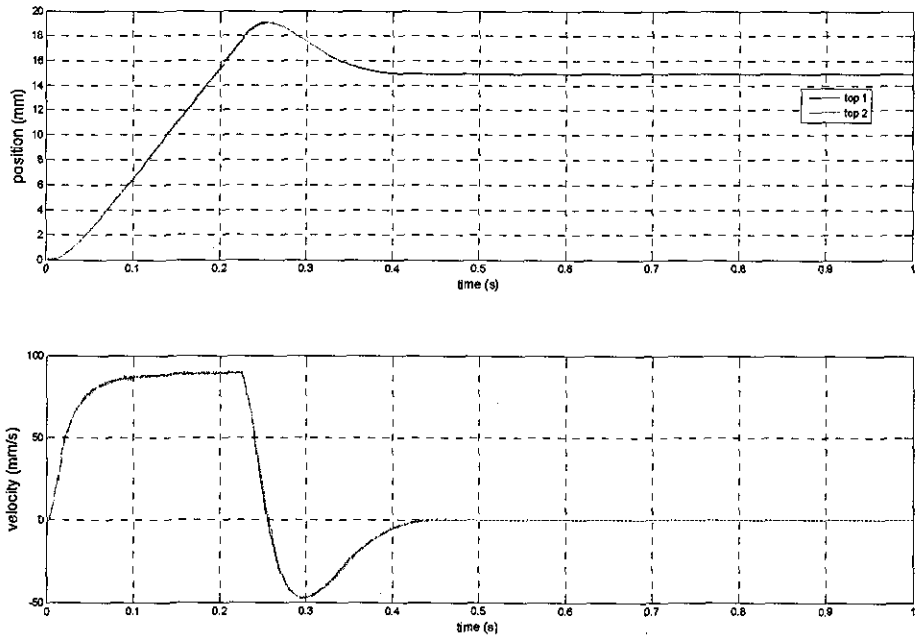


Fig. 6.30 Performance of an HRA with a mathematical reduction LQG controller



Bottom Elements



Top Elements

Fig. 6.31 Performance of individual elements in HRA with a mathematical reduction LQG controller

The performance is very similar with full-order design's performance with a little smaller overshoot. Bottom and top elements also show very similar performances because same control input is applied.

6.4.6 Physical reduction LQG controller

A physical reduction method is then applied by assuming the HRA as a bigger individual actuator. The seven states individual state-space model with a four times bigger input matrix and a half magnitude output matrix is used. Results of both the overall HRA and each element have been shown in Fig. 6.32 and Fig. 6.33 respectively.

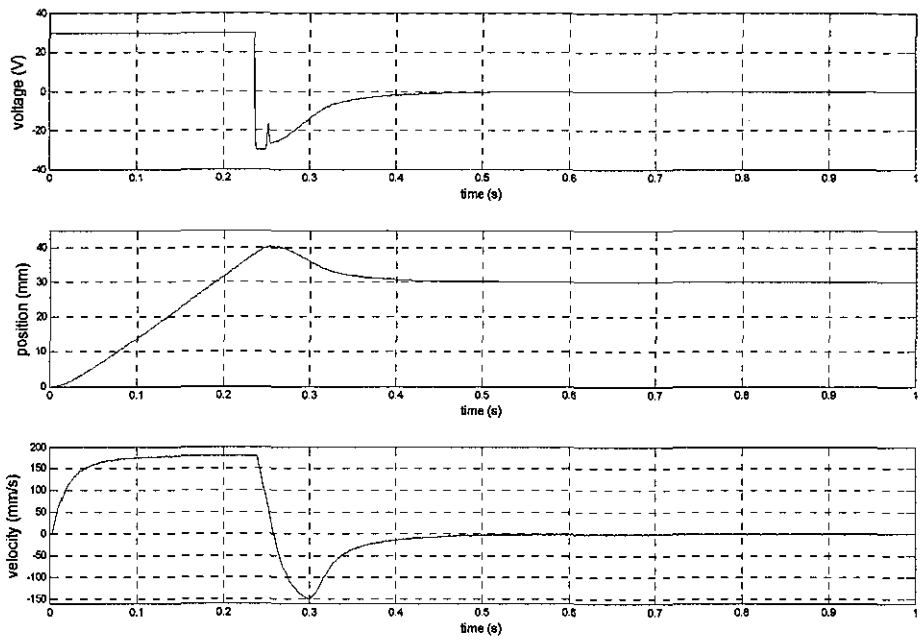
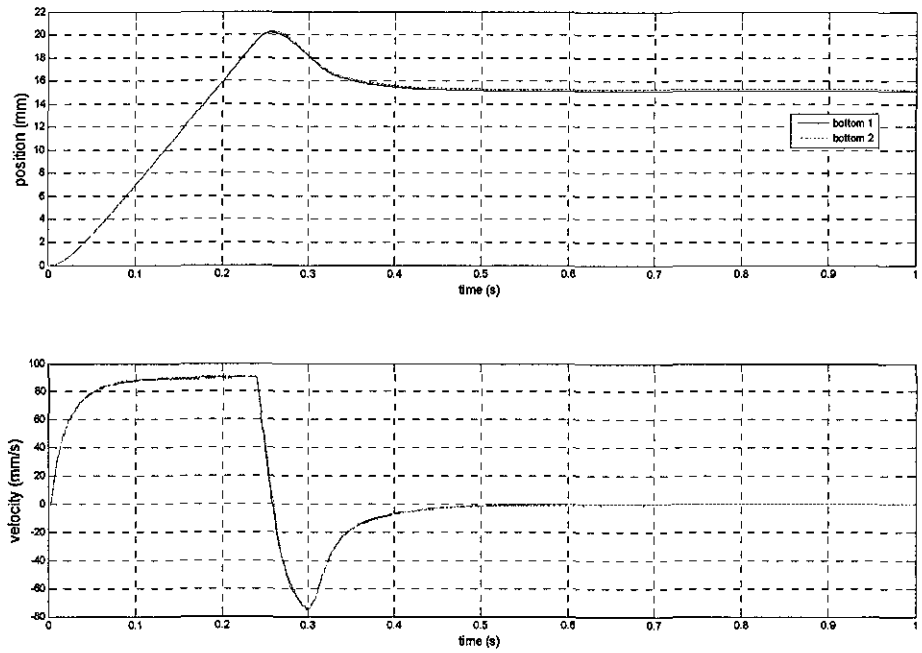
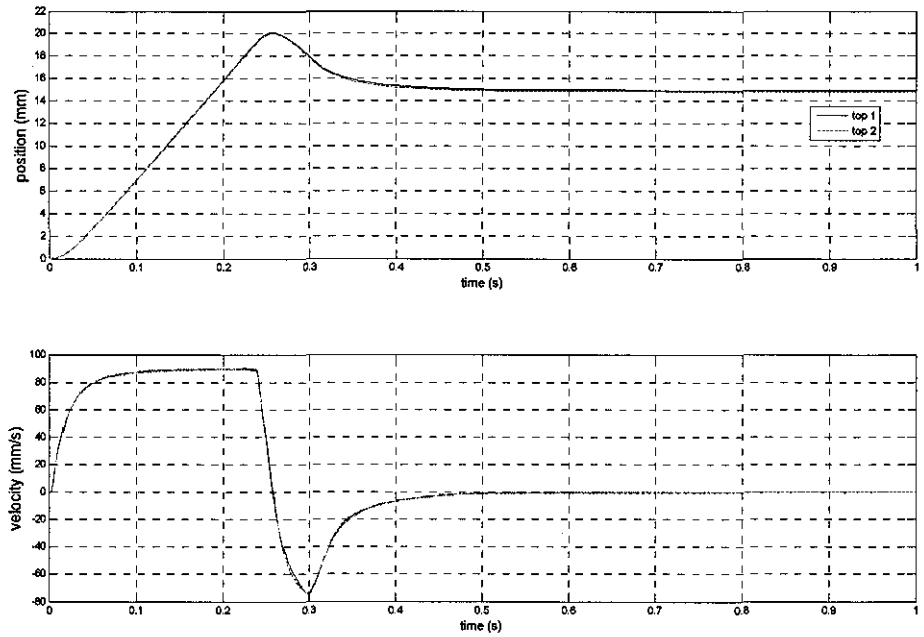


Fig. 6.32 Performance of an HRA with a physical reduction LQG controller



Bottom Elements



Top Elements

Fig. 6.33 Performance of individual elements in HRA with a physical reduction LQG controller

The performance is very similar to the other two optimal control designs. The controller controls the HRA to reach the setpoint with zero steady state error and a 30% overshoot at position output.

6.5 Experimental Results - Faults Injected

Five kinds of controllers using both classical and optimal algorithms have been introduced and applied to the two by two series-in-parallel HRA. All these algorithms have been shown to work well in the healthy situation (without any faults injected). But the main advantage of the HRA is the fault tolerance ability without changing controllers. To demonstrate this, all these controllers are tested under fault conditions (without changing any controller parameters).

It is assumed that there is only one faulty actuation element: Bottom 1 element. Three kinds of faults are considered here: open circuit, short circuit, and lock up. These are readily simulated on the experiment. Open circuit is simulated by disconnecting the DC motor from its amplifier physically. Short circuit is simulated by disconnecting the DC motor electrically, i.e. setting applied voltage to be zero rather than physically. Lockup is simulated by lock the linkage with the mechanical actuator using two screws through the two holes made on the linkage.

Performances of the HRA and each element under all the three faults will be shown and compared with the healthy situation. In all cases designs are as discussed earlier (based on fault free models).

6.5.1 Voltage-driven classical controller

Firstly, the classical voltage-driven controller is tested using the same design shown in Section 6.4.2. Performances of the HRA and actuation elements will be shown in Fig. 6.34 and Fig. 6.35 respectively.

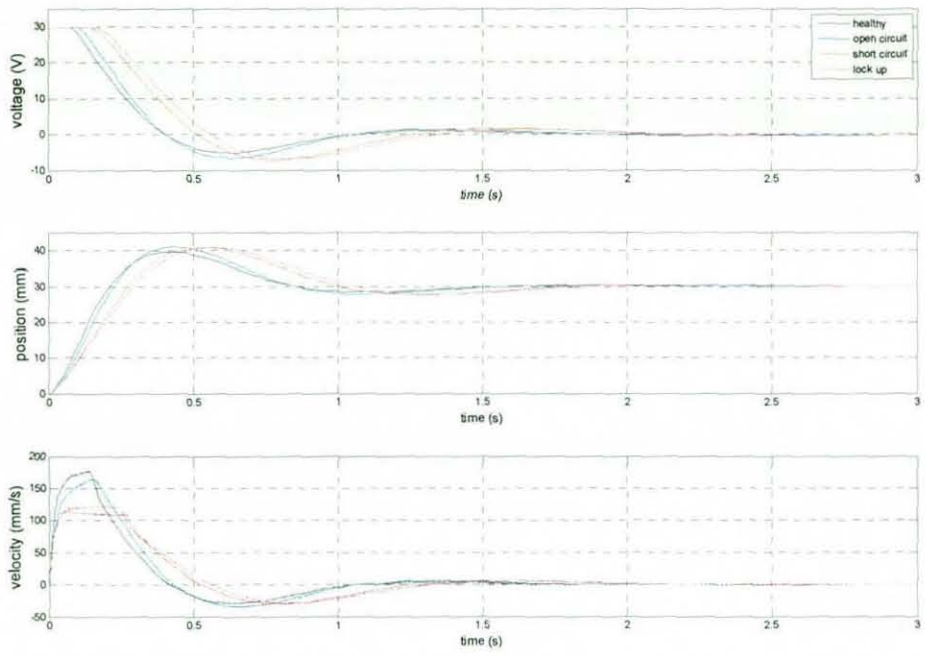
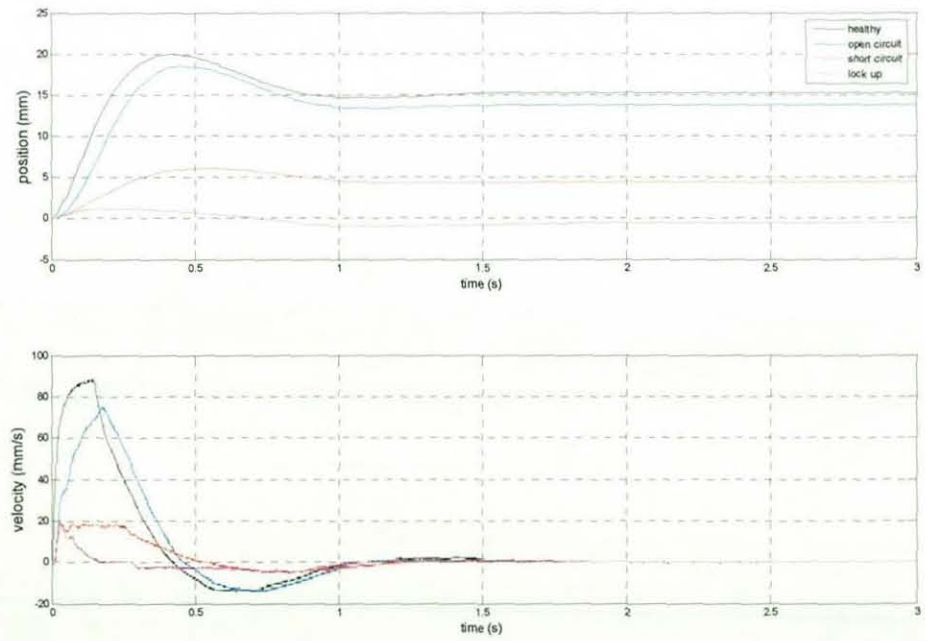
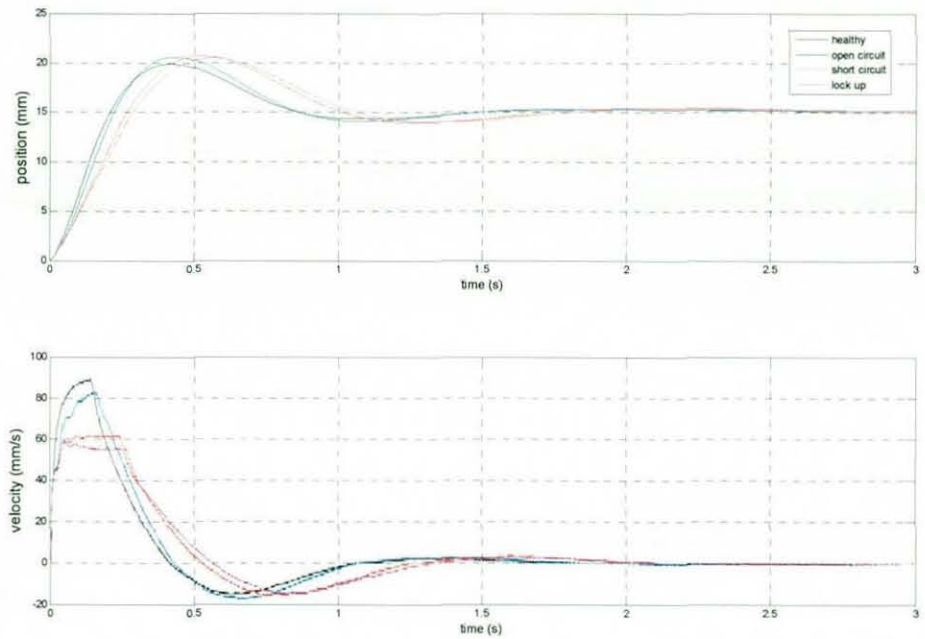


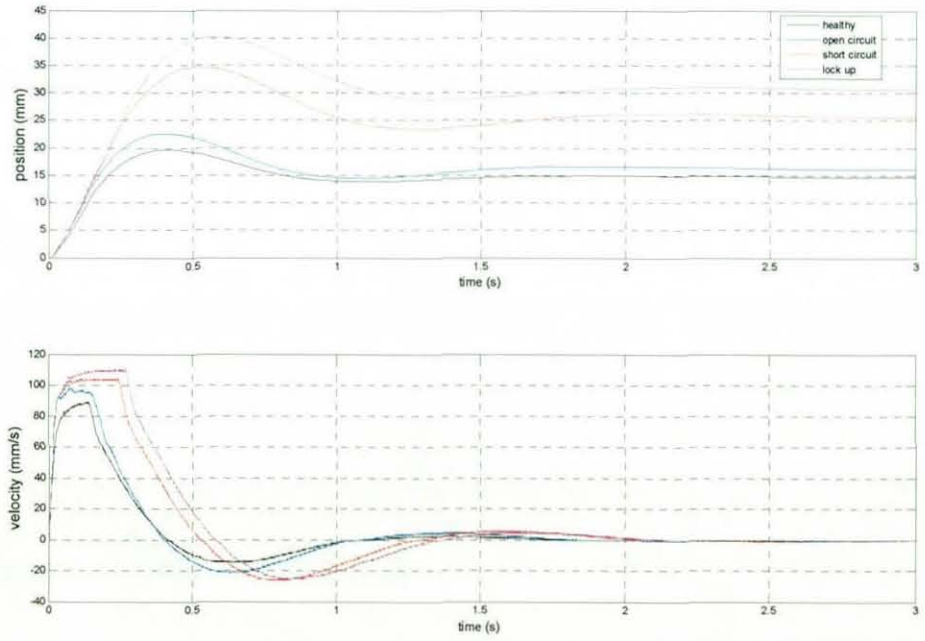
Fig. 6.34 Performance of an HRA with a classical voltage-driven controller when faults injected



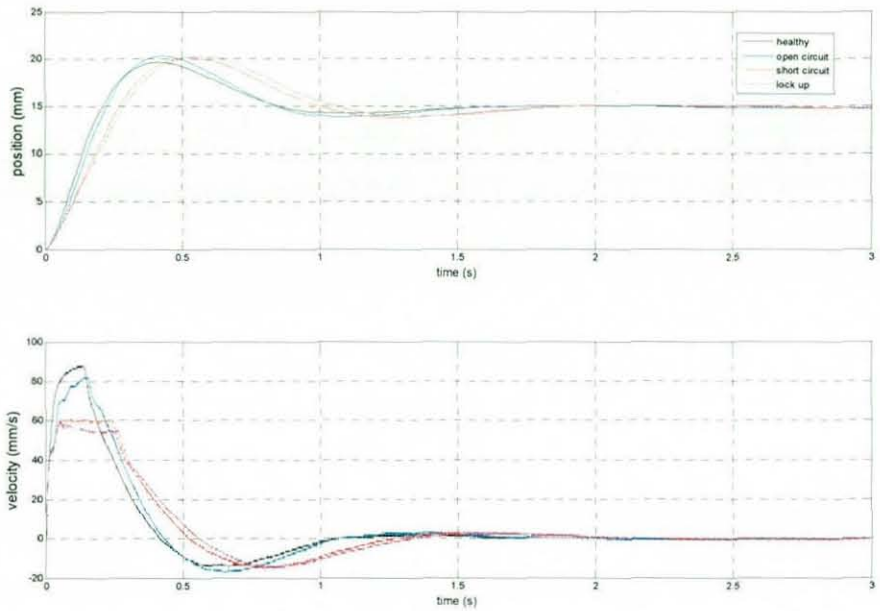
Bottom 1



Bottom 2



Top 1



Top 2

Fig. 6.35 Performance of elements with a classical voltage-driven controller when faults injected

Using the healthy performance as a reference, it can be seen that the position output still can reach the 30mm set point but degradations also can be found when the faults are injected. The lockup fault makes the HRA response slowest, while the open circuit fault is the fastest response in the faults injected situations. The short circuit fault's response is similar to the lockup situation but a little faster.

Looking at the individual elements, because faults are injected into the Bottom 1 element, it can not provide any output. Both Bottom 1 and Top 1 are pushed by the other set of series connection. In the short circuit and lockup situation, the Bottom 1 element is difficult to be pushed so that very limited output can be provided (almost zero in the lockup situation). In the open circuit situation, the only resistance is the friction so that it is pushed more than the other two fault situations but still less output can be provided than a healthy situation. The Top 1 is then pushed to provide more output

than in a healthy situation to make the overall output of Top 1 and Bottom 1 be equal to the other series configuration. Because the Bottom 1 element's performance is limited, the overall performance of HRA is degraded, but because of other elements, the HRA still can finish the function. Then the faults are accommodated by the HRA without changing the controller or the mechanical structure.

6.5.2 Current-driven classical controller

The current-driven classical controller will be tested following using the same design introduced in Section 6.4.3. Performances of the HRA and actuation elements are shown in Fig. 6.36 and Fig. 6.37 respectively.

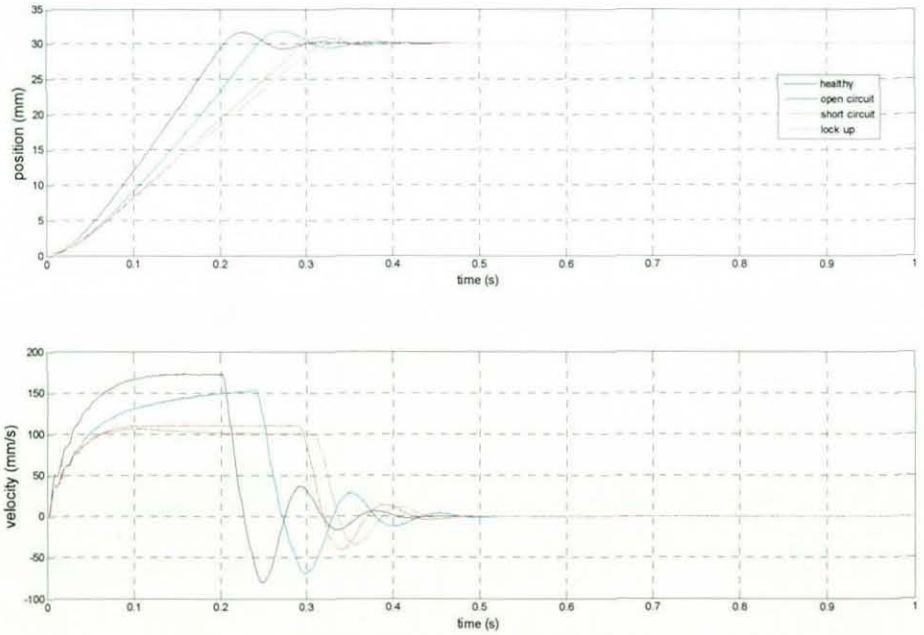
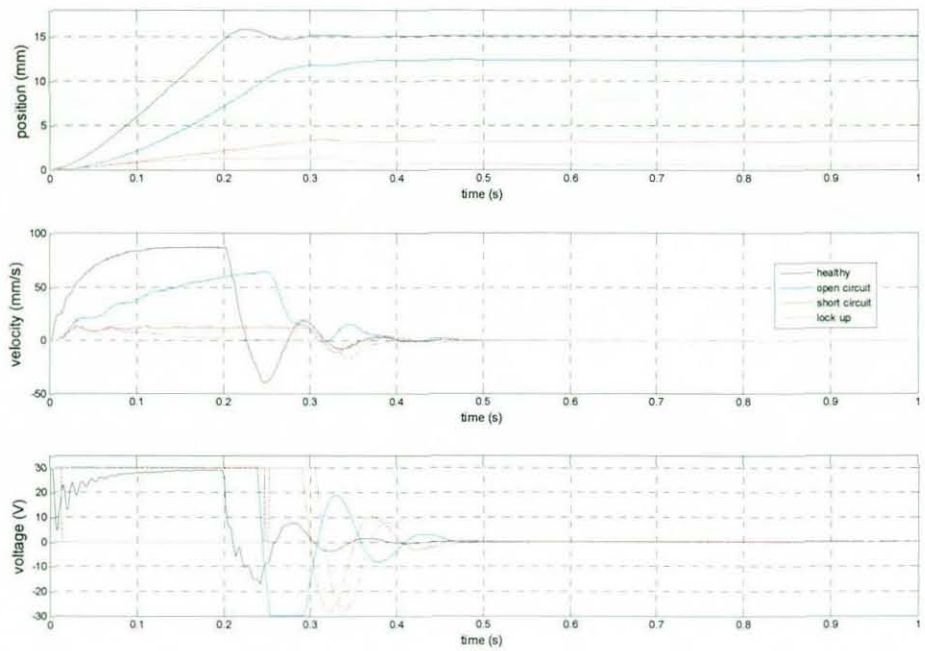
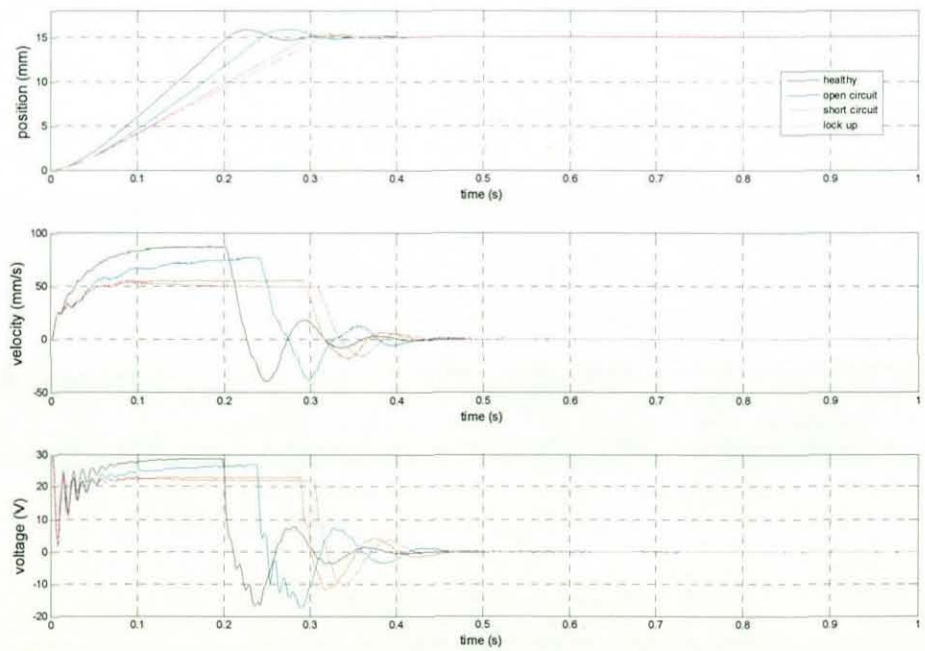


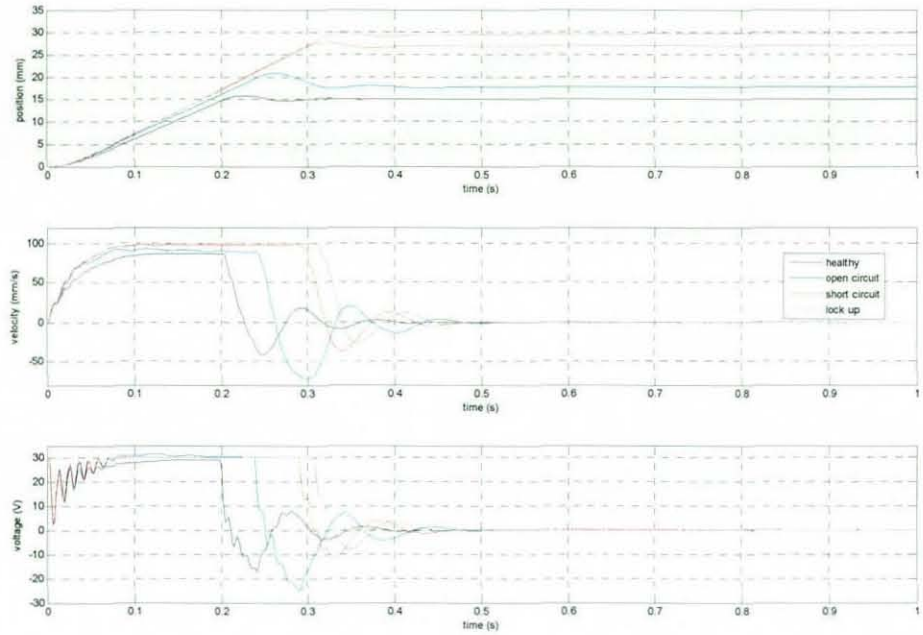
Fig. 6.36 Performance of an HRA with a classical current-driven controller when faults injected



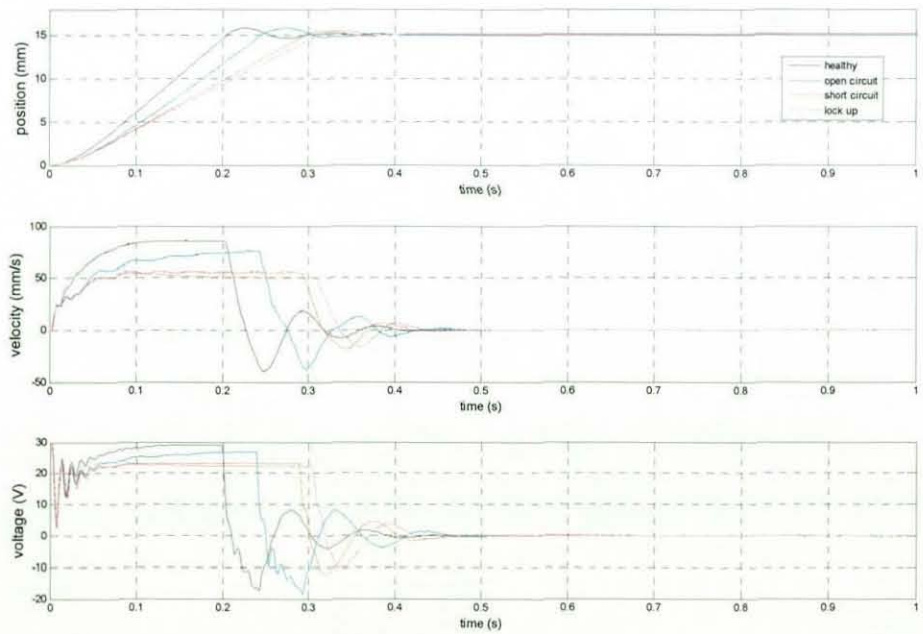
Bottom 1



Bottom 2



Top 1



Top 2

Fig. 6.37 Performance of elements with a classical current-driven controller when faults injected

Very similar tends in the results can be found comparing with the voltage-driven controller. The overall position still can reach the setpoint when faults injected but the response is slower. The Bottom 1 element provides limited position output and Top 1 provides more than a healthy situation because both of them are pushed by the other set of elements.

6.5.3 Full order LQG controller

The full order LQG controller as introduced in Section 6.4.4 will be tested. Performances of the HRA and actuation elements are shown in Fig. 6.38 and Fig. 6.39 respectively.

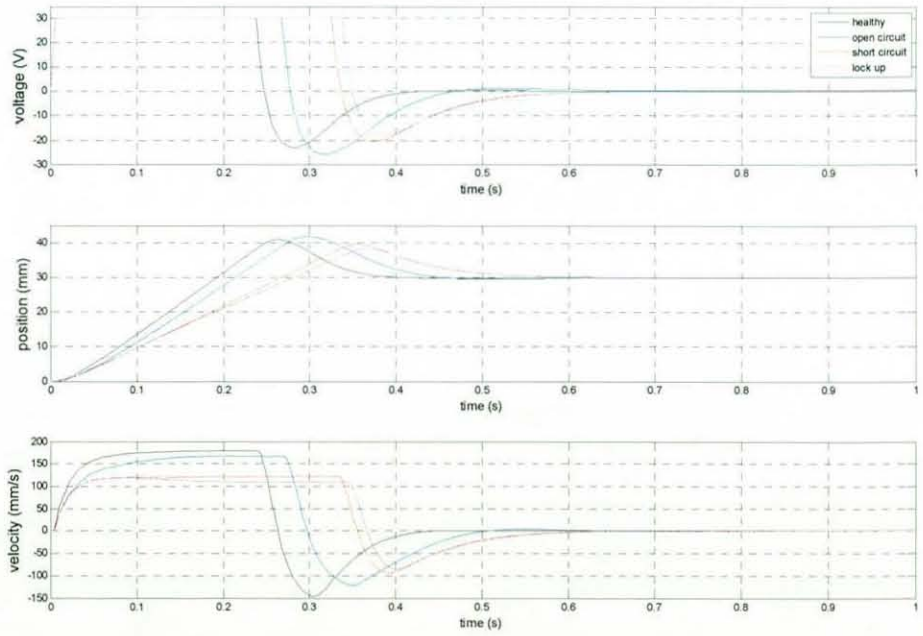
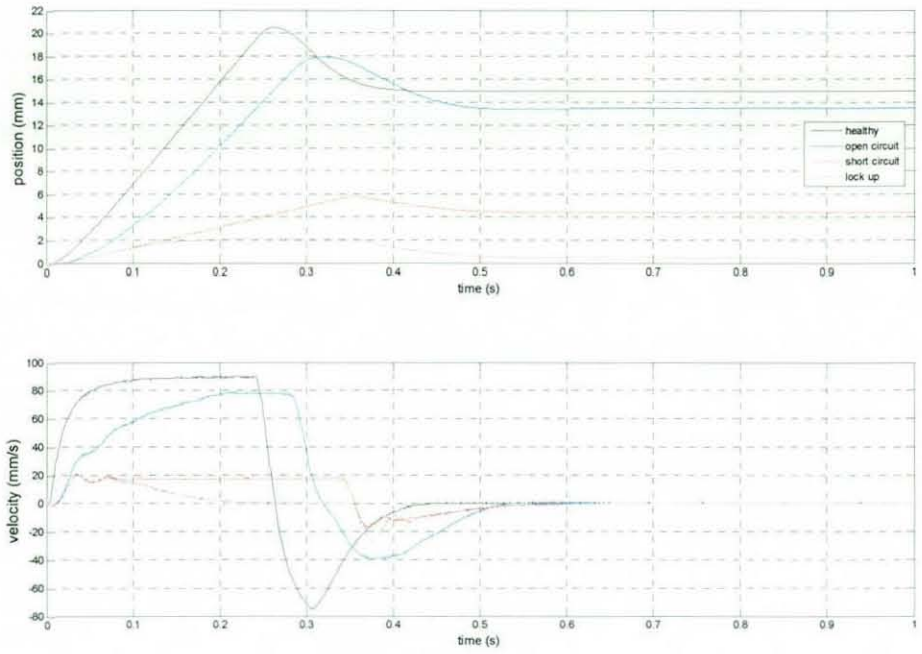
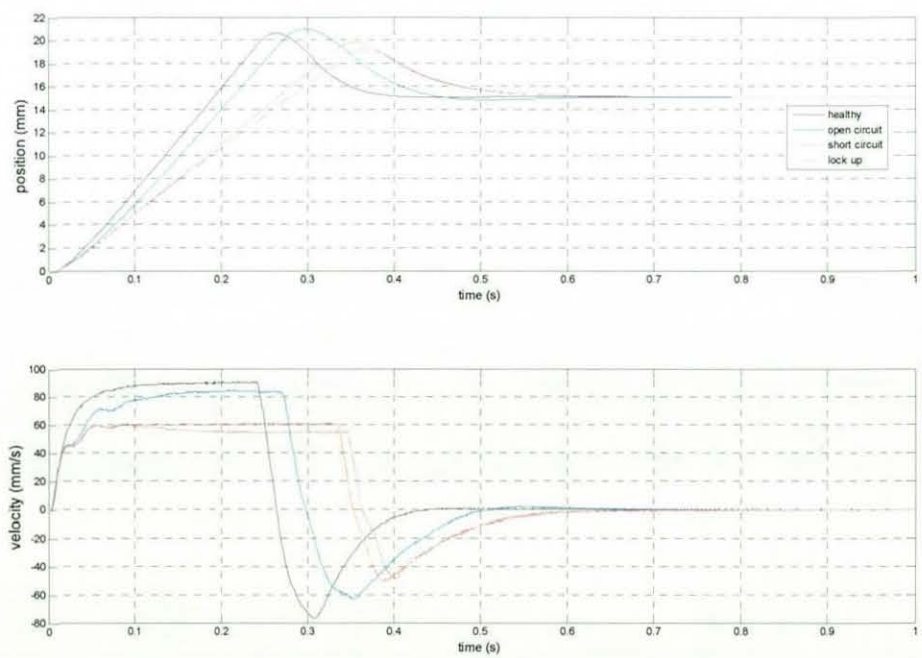


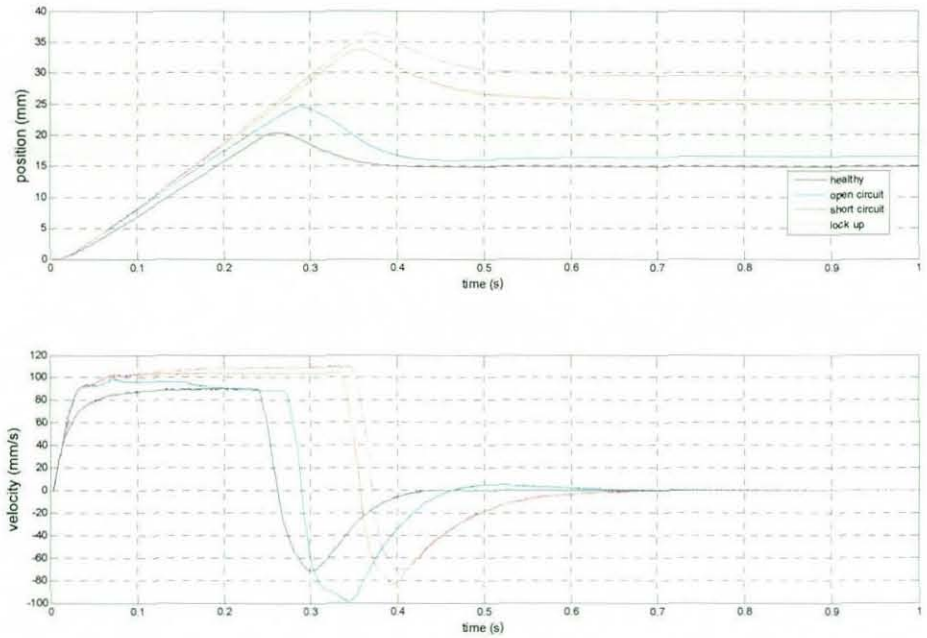
Fig. 6.38 Performance of an HRA with a full order LQG controller when faults injected



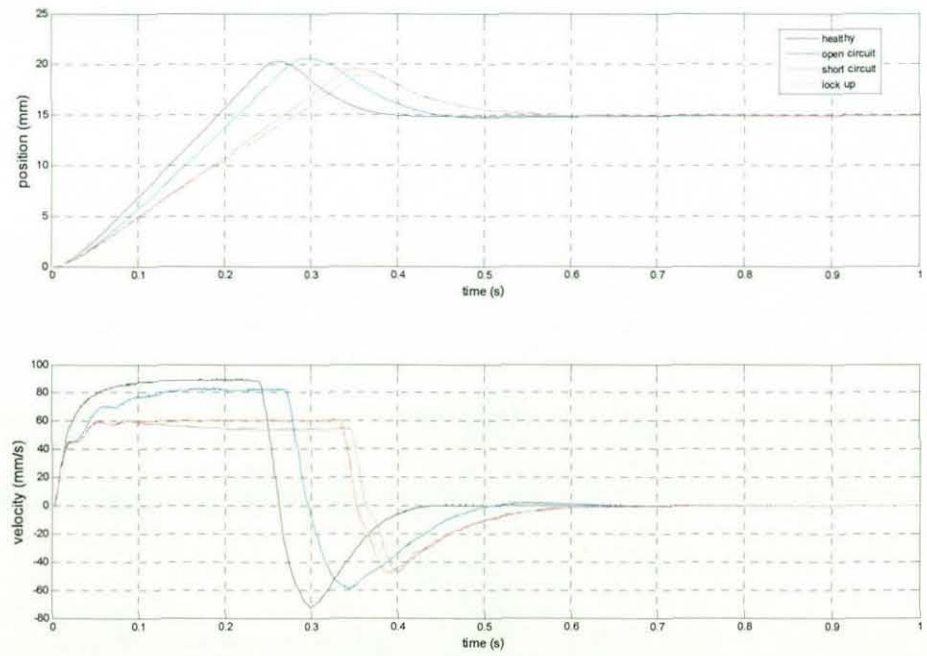
Bottom 1



Bottom 2



Top 1



Top 2

Fig. 6.39 Performance of elements with a full order LQG controller when faults injected

The HRA still can reach the setpoint when faults injected into the Bottom 1 element, although a slower performance can be expected. Inside the HRA, Bottom 1 element will provide limit position output especially with short circuit and lockup faults, which will be compensated by the Top 1 element. The other set of elements, i.e. Bottom 2 and Top 2, still share the similar performances.

6.5.4 Mathematical reduction LQG controller

The model reduction methods are then tested. Firstly the mathematical method as introduced in Section 6.4.5 is applied. Performances of the HRA and actuation elements are shown in Fig. 6.40 and Fig. 6.41respectively.

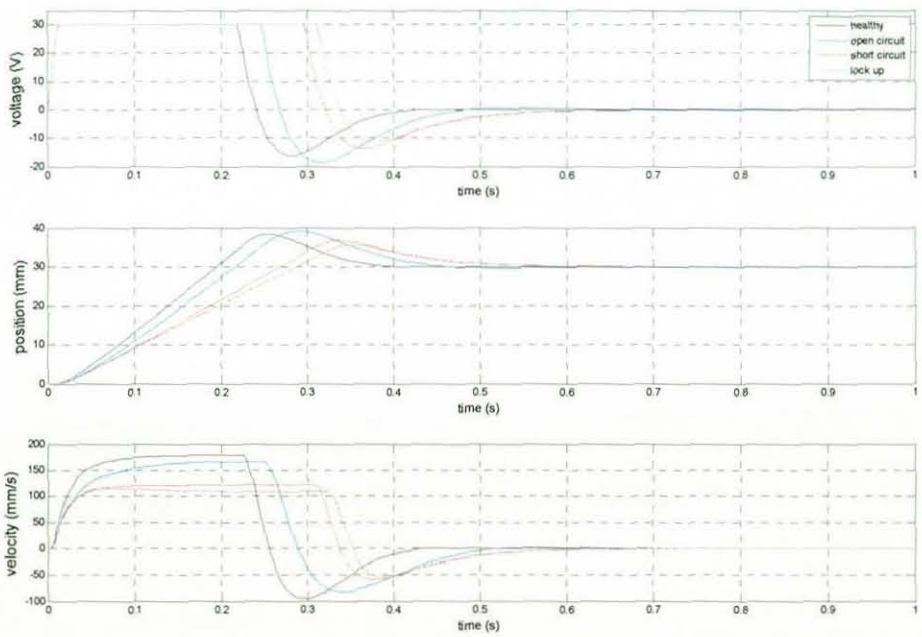
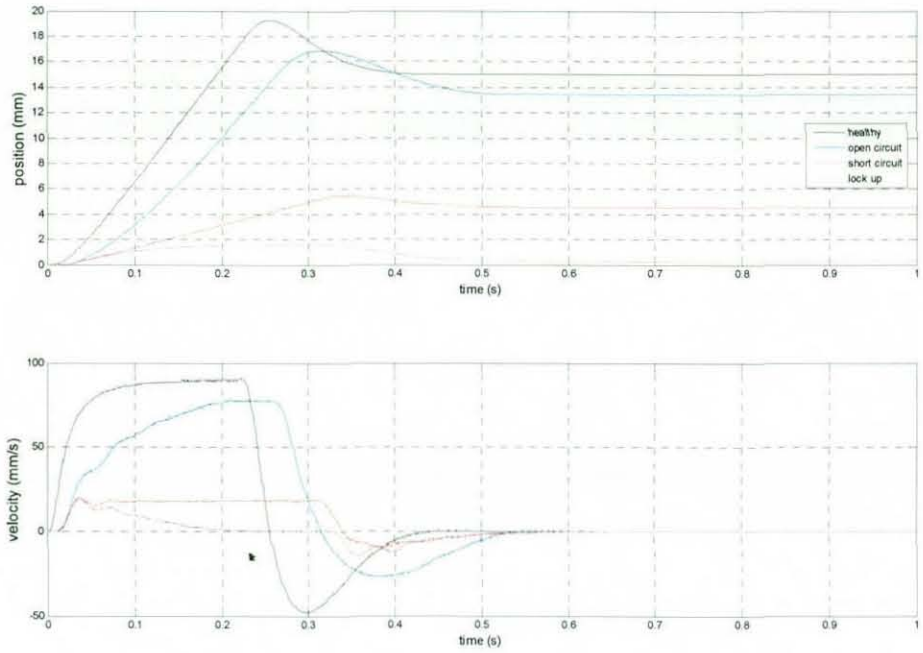
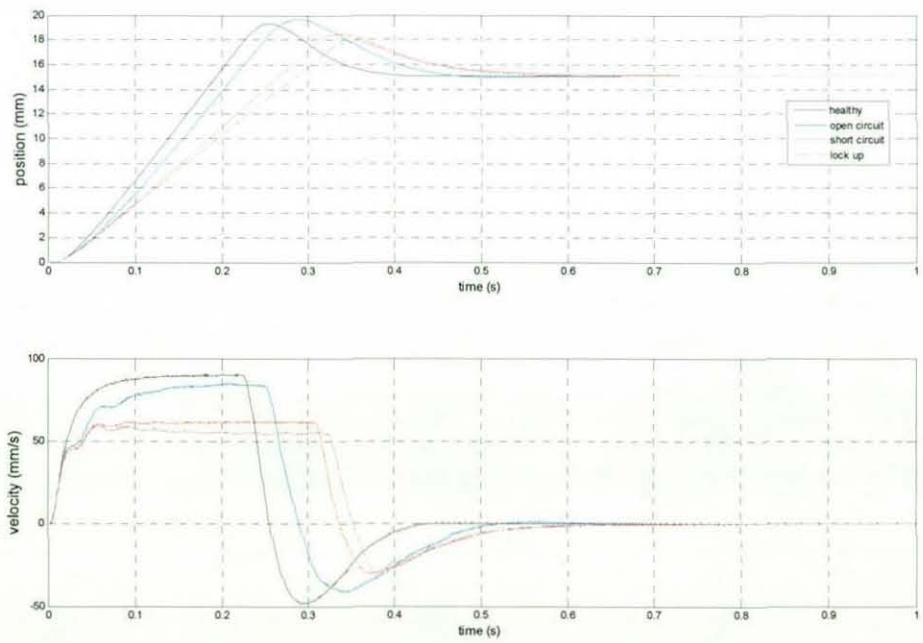


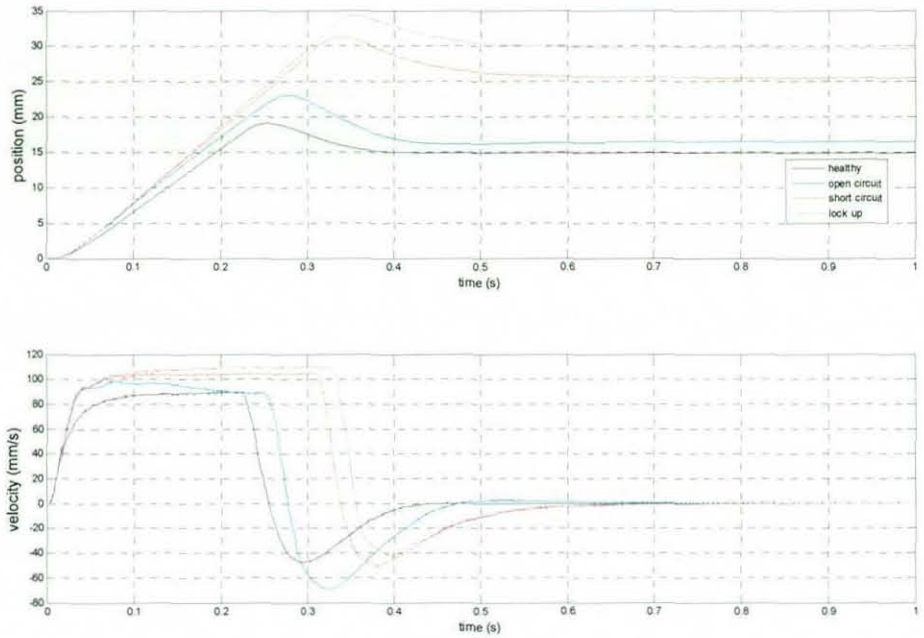
Fig. 6.40 Performance of an HRA with a mathematical reduction LQG controller when faults injected



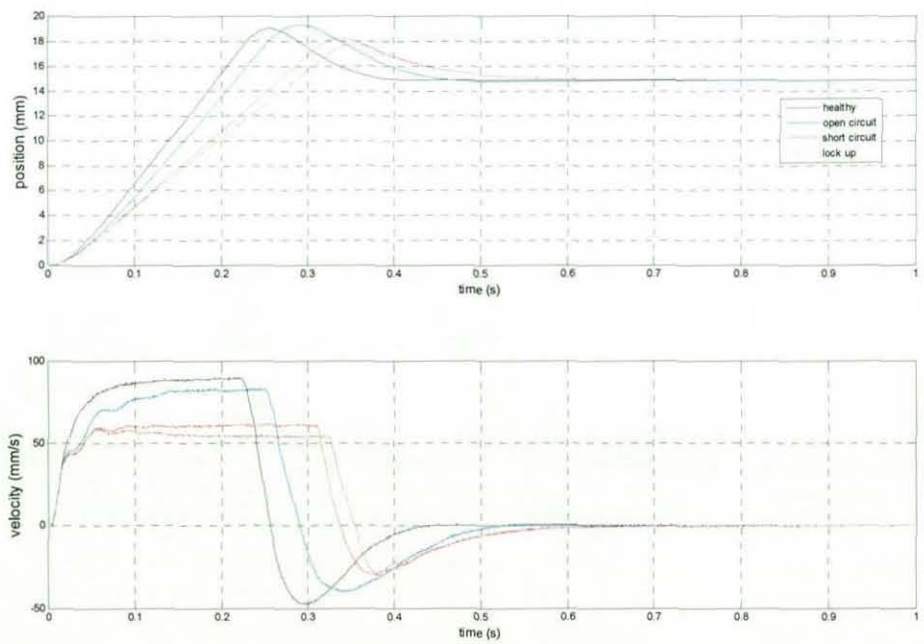
Bottom 1



Bottom 2



Top 1



Top 2

Fig. 6.41 Performance of elements with a mathematical reduction LQG controller when faults injected

6.5.5 Physical reduction LQG controller

The physical reduction method as introduced in Section 6.4.6 is then applied.

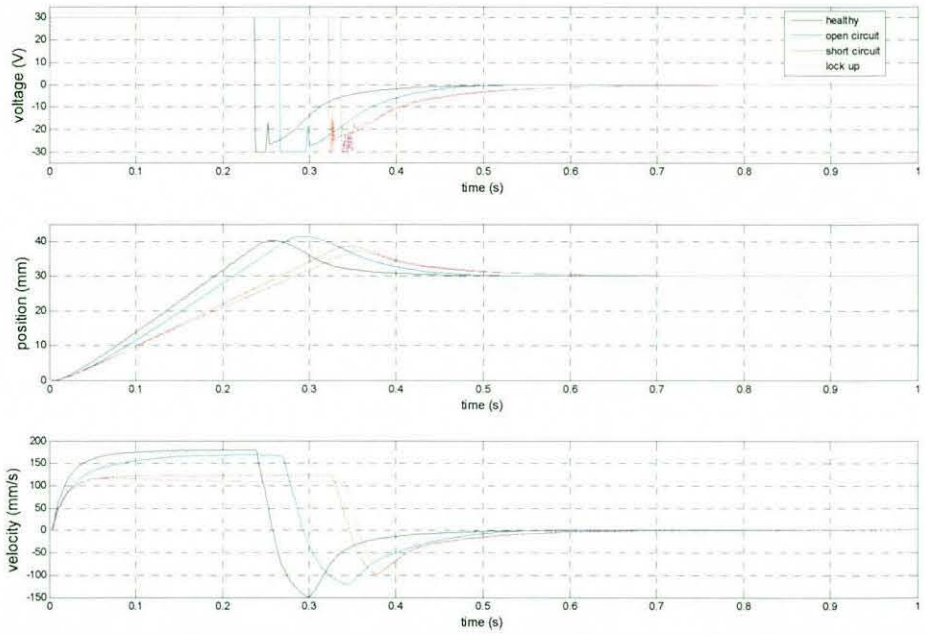
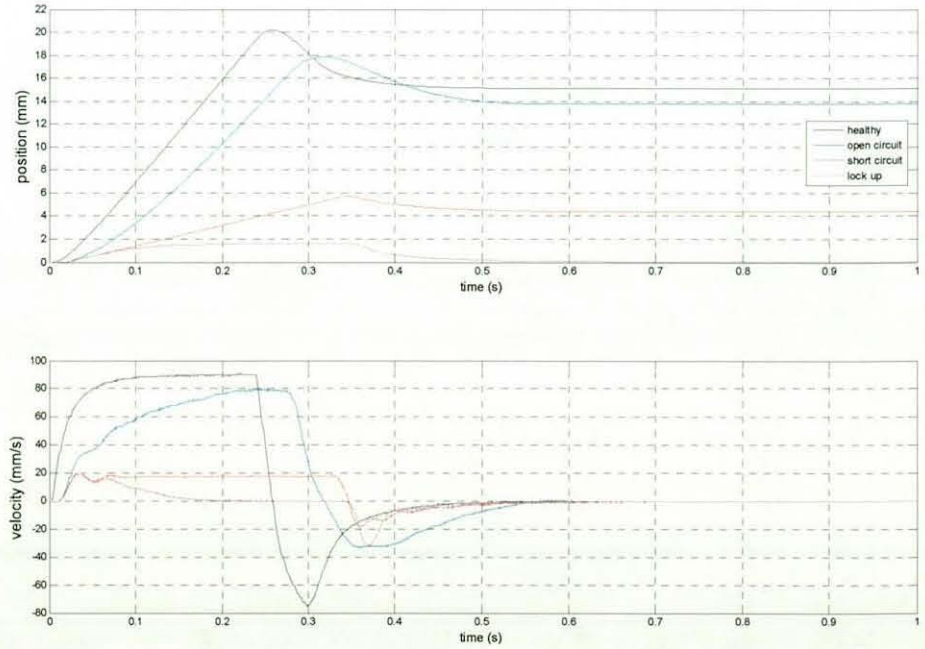
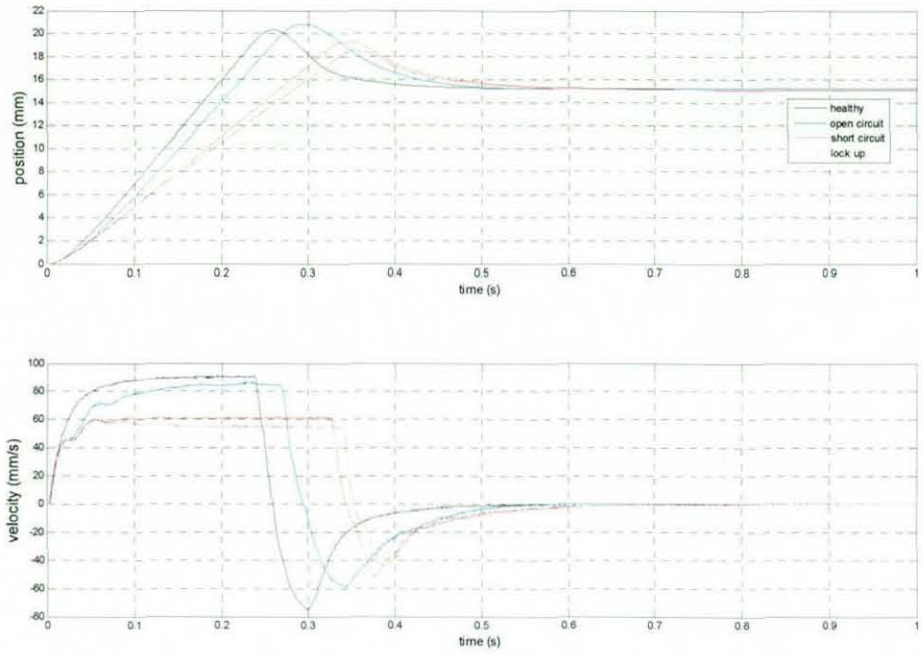


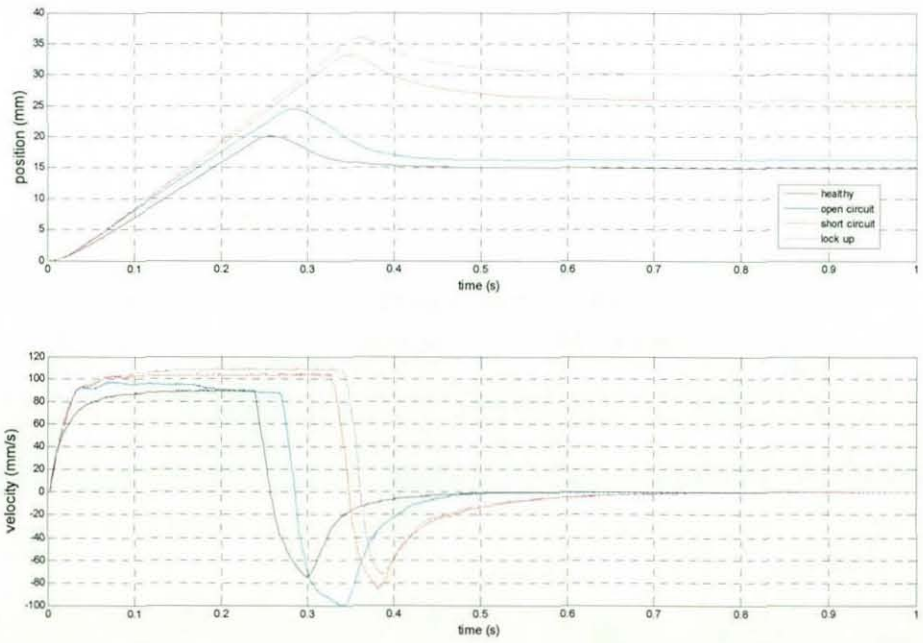
Fig. 6.42 Performance of an HRA with a physical reduction LQG controller when faults injected



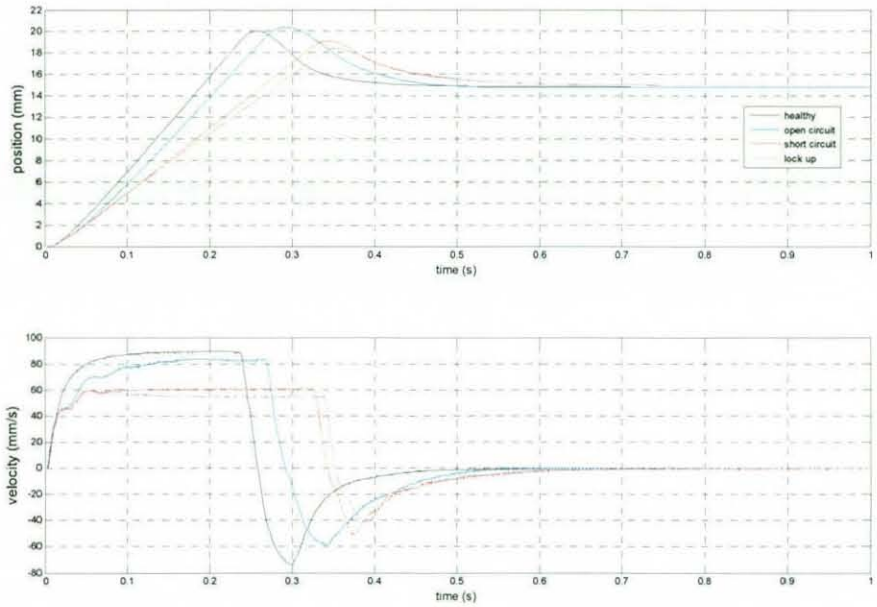
Bottom 1



Bottom 2



Top 1



Top 2

Fig. 6.43 Performance of elements with a physical reduction LQG controller when faults injected

Performances of the HRA and actuation elements are shown in Fig. 6.42 and Fig. 6.43 respectively. Comparing the reduced order methods and the full order design, very similar performance can be seen. The HRA can reach the setpoint with a slower response when faults are injected, which means the failure in Bottom 1 element can be accommodated by the HRA without reconfiguration.

To have a clearer look at how the performances are degraded when faults happen using different kinds of controllers, the performances will be listed and shown graphically in the next section.

6.6 Experimental Results - Brief Summary

All experimental results of the HRA using five kinds of controllers have been presented in Section 6.4 (healthy situation) and Section 6.5 (faults injected situations, i.e. open circuit, short circuit, lock up). From these results, it can be found that all controllers can control the HRA well to reach the setpoint under both healthy and faults injected situations although different levels of degradation can be found depending on different controllers and different faults. In this section, detailed features of the responses in different situations with different controllers are listed in order to have a clear comparison among different controllers. The features listed include Steady-state Error (SE), Rise time (RT), i.e. from 0 to 90%, Settling time (ST), i.e. 5% tolerance, and Overshoot (OT). The results using fast design are listed firstly followed by the slow design's results.

6.6.1 Fast design

All results of the fast design have been shown in previous sections. Here, Table. 6.5 lists some detailed features in time domain of both position and velocity outputs using all five controllers under both healthy and three fault injected situations. Table. 6.6 gives rise time of the position output separately. The value under a healthy situation using all five controllers are listed (CV for Classical Voltage-driven, CC for Classical Current-driven, OF for Optimal Full-order, OM for Optimal Mathematical-reduction, OP for Optimal Physical-reduction). Other values with faults injected are listed in percentage compared with the value under healthy situation so that the performance degradation can be seen clearly. Settling time of the position output is listed in Table. 6.7 and peak velocity is listed in Table. 6.8 respectively.

Table. 6.5 Performance of fast controllers in healthy and faulty situations

Situations		SE(mm)	RT(s)	ST(s)	OS	PV(mm/s)
Healthy	Classical Voltage-Driven	0	0.18	0.77	32%	178
	Classical Current-Driven	0	0.18	0.23	6%	174
	LQG Full-Order	0	0.18	0.35	36%	180
	LQG Mathematical-Reduction	0	0.18	0.35	28%	180
	LQG Physical-Reduction	0	0.18	0.35	28%	180
Open circuit	Classical Voltage-Driven	0	0.20	1.26	36%	165
	Classical Current-Driven	0	0.22	0.28	6%	153
	LQG Full-Order	0	0.19	0.41	39%	168
	LQG Mathematical-Reduction	0	0.20	0.41	31%	167
	LQG Physical-Reduction	0	0.20	0.41	31%	167
Short circuit	Classical Voltage-Driven	0	0.24	1.51	36%	122
	Classical Current-Driven	0	0.27	0.28	3%	111
	LQG Full-Order	0	0.24	0.49	32%	122
	LQG Mathematical-Reduction	0	0.24	0.46	23%	123
	LQG Physical-Reduction	0	0.24	0.46	23%	123
Lockup	Classical Voltage-Driven	0	0.26	1.55	36%	115
	Classical Current-Driven	0	0.28	0.30	3%	108
	LQG Full-Order	0	0.25	0.49	28%	120
	LQG Mathematical-Reduction	0	0.26	0.46	19%	116
	LQG Physical-Reduction	0	0.26	0.46	19%	116

Table. 6.6 Rise time of fast controllers in healthy and faulty situations

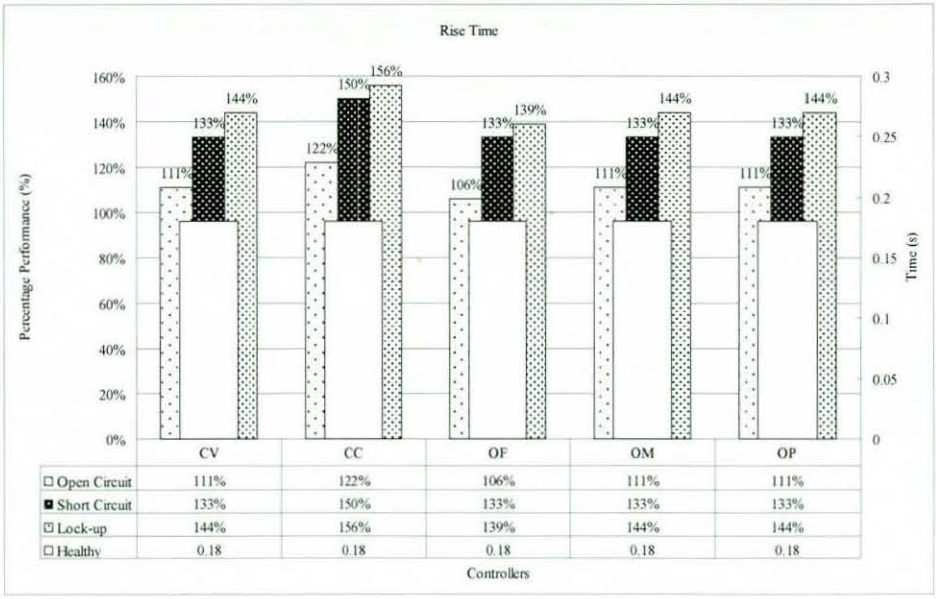


Table. 6.7 Settling time of fast controllers in healthy and faulty situations

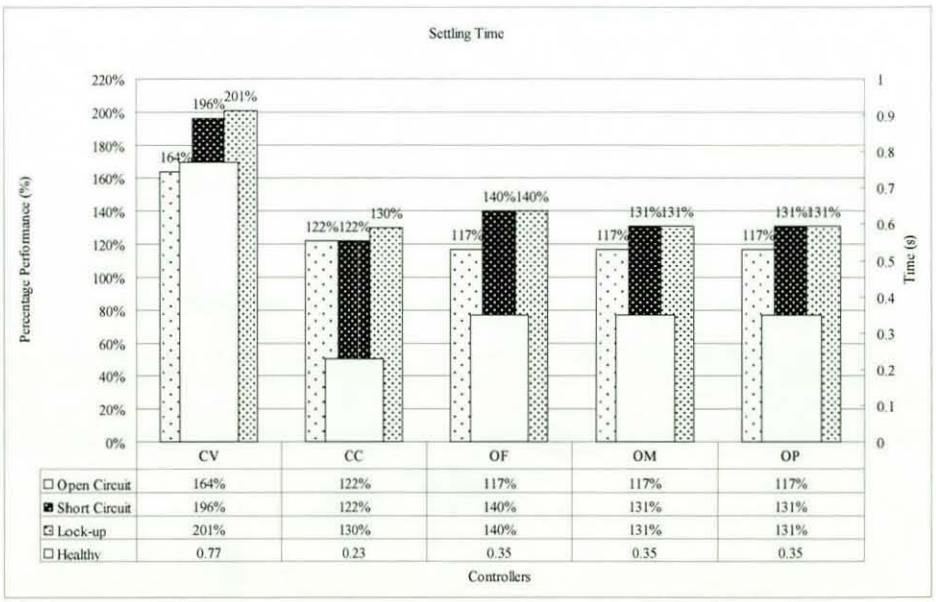
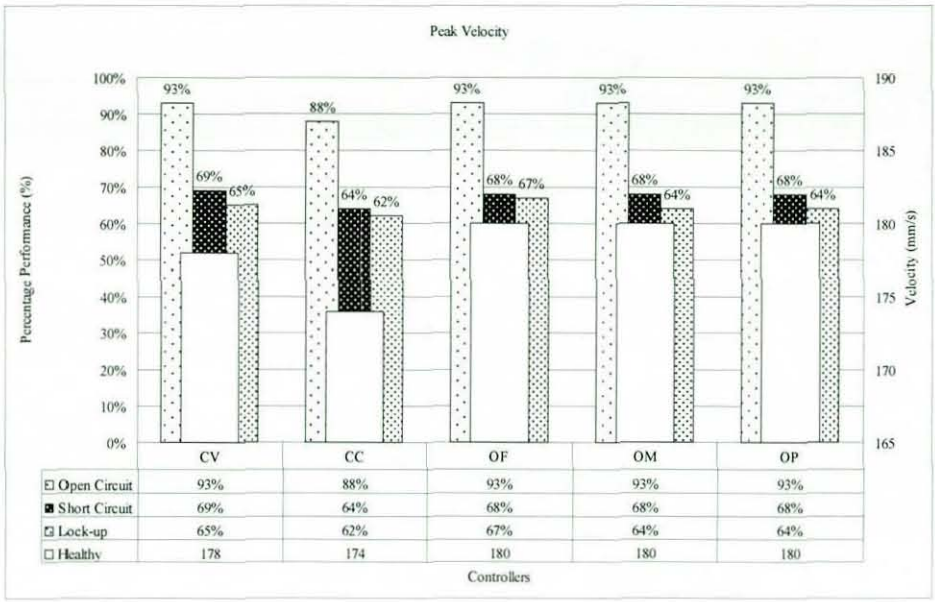


Table. 6.8 Peak velocity of fast controllers in healthy and faulty situations



Due to the high gain design, the HRA can reach the setpoint using all five controllers under both healthy and faults injected situations. From the four tables, it can be seen that the Classical Voltage-driven controller performs slowest while the Classical Current-driven controller is the fastest. The CV controller also degrades mostly when faults are injected. The other four controllers have similar degradation when faults are injected. All controllers degrade mostly when the lockup fault happens and degrade least when the open circuit fault happens.

6.6.2 Slow design

A slow design of controllers is introduced in Section 6.4.1 but only one result using classical voltage-driven controller is shown. The other four kinds of controllers have been designed using the CV controller as a reference (making the outputs similar). All five controllers using slow design are listed in details in this section.

Table. 6.9 Performance of slow controllers in healthy and faulty situations

Situations		SE(mm)	RT(s)	ST(s)	OS	PV(mm/s)
Healthy	Classical Voltage-Driven	0.65	1.31	1.77	0%	104.50
	Classical Current-Driven	1.33	1.43	2.12	0%	90.00
	LQG Full-Order	0.39	0.96	1.07	1.97%	78.50
	LQG Mathematical-Reduction	0.25	0.83	0.95	0.55%	94.50
	LQG Physical-Reduction	1.09	1.11	1.44	0%	82.50
Open circuit	Classical Voltage-Driven	0.75	1.42	2.02	0%	89.00
	Classical Current-Driven	1.37	1.76	2.67	0%	700
	LQG Full-Order	0.28	1.02	1.15	0.54%	700
	LQG Mathematical-Reduction	0.17	0.87	1.00	0%	91.00
	LQG Physical-Reduction	1.31	1.20	1.68	0%	79.50
Short circuit	Classical Voltage-Driven	0.87	1.85	2.54	0%	750
	Classical Current-Driven	1.61	2.19	3.15	0%	64.00
	LQG Full-Order	1.40	1.38	1.68	0%	61.00
	LQG Mathematical-Reduction	1.52	1.22	1.56	0%	72.50
	LQG Physical-Reduction	2.08	1.73	2.58	0%	63.50
Lock up	Classical Voltage-Driven	0.97	2.05	2.78	0%	75.00
	Classical Current-Driven	1.67	2.27	3.30	0%	64.00
	LQG Full-Order	1.91	1.59	2.07	0%	55.00
	LQG Mathematical-Reduction	1.85	1.42	1.91	0%	650
	LQG Physical-Reduction	2.42	1.96	3.00	0%	61.00

Table. 6.10 Rise time of slow controllers in healthy and faulty situations

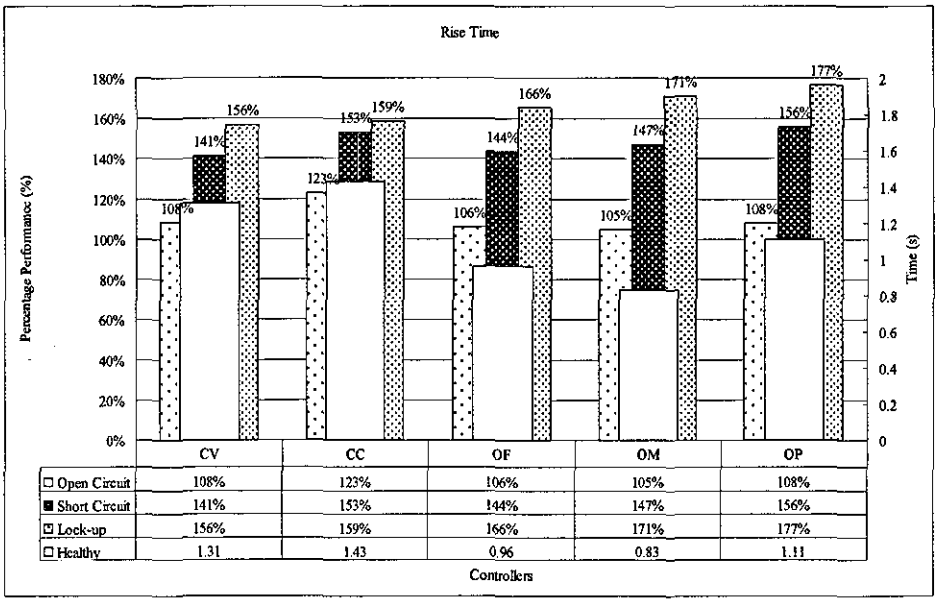


Table. 6.11 Settling time of slow controllers in healthy and faulty situations

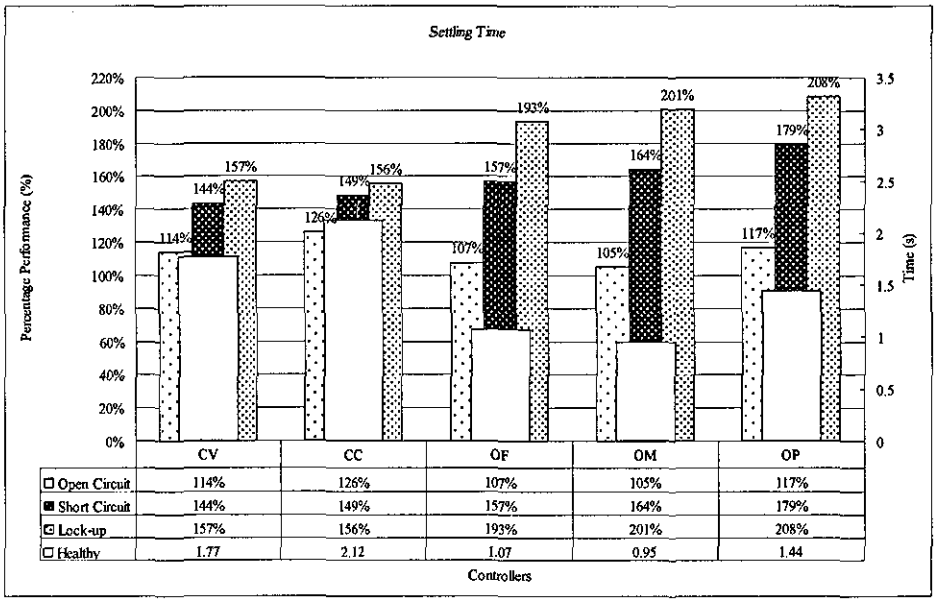
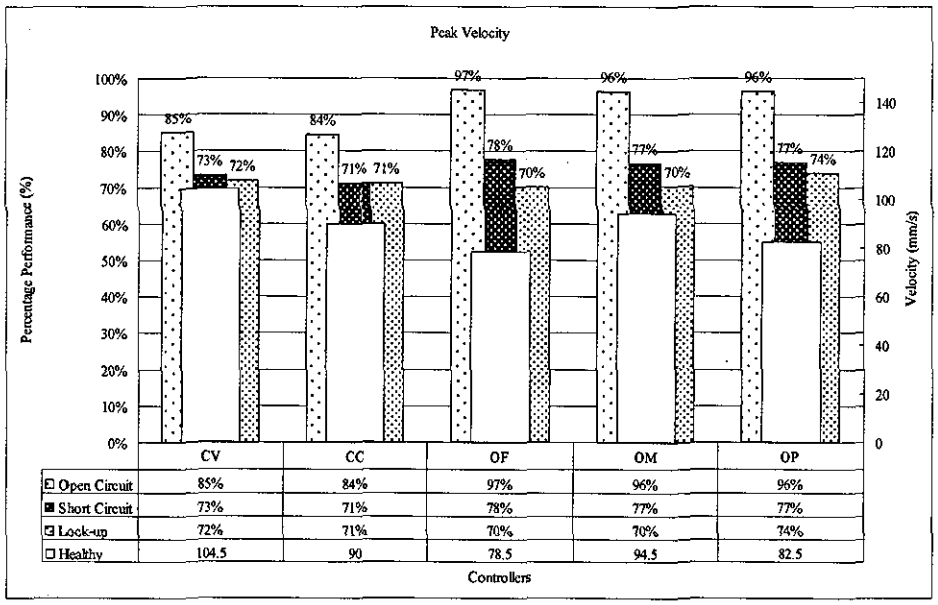


Table. 6.12 Peak velocity of slow controllers in healthy and faulty situations



Same as the fast design, Table. 6.9 lists all the time domain features firstly and Table. 6.10 to Table. 6.12 show the rise time, settling time, and peak velocity separately. Using this slow design, the HRA can reach the setpoint under both healthy and fault injected situations. But due to the small gain of all controllers, the controller can not overcome the friction so that there is slight steady state error. All controllers are designed to have similar performances so that the listed features do not have bigger difference among different controllers. Under healthy, the optimal controllers perform a little better than classical controllers with faster response. But when faults are injected, the performance degrades more using optimal controllers than using classical controllers. In the fast design, it can be found that the performance does not degrade as much as in the slow design except the CV controller. This is due to the high gain control design. When the HRA response slower with faults injected, a high gain controller can produce higher control voltage so that the HRA can accommodate the faults more smoothly. But in the fast CV control design, a high gain integrator is included. This integration causes the overshoot and makes the HRA reach

the steady state slower, so that the degradation of the HRA is much higher than other four controllers.

From these results, it can be concluded that the HRA can be controlled using both classical and optimal controllers. The HRA has the ability of fault tolerance without any reconfiguration when faults are injected into one actuation element benefit from the mechanical structure. When the number of redundancy increases, this ability of also will increase to be tolerant more number of elements' faults. It also can be concluded that the HRA will perform better using a high gain design with faster response and less degradation, but the high gain integration also could make the HRA more difficult to reach the steady state.

7. CONCLUSION AND FUTURE WORK SUGGESTIONS

The High Redundancy Actuator is a novel type actuation system for safety critical applications. By comprising a number of individual actuation elements, the HRA is configured and controlled in such a way that faults in individual elements are inherently accommodated without resulting in a failure of the complete actuation system. The capability of fault tolerance provided by the HRA ensures the safety and stability of the whole system, which may well be timely from an industrial viewpoint.

The research works about the HRA introduced in this thesis include modelling, control design, simulation and hardware demonstrator. The content and contribution of each chapter will first be summarised in this chapter, and then some suggestions for future studies are given.

7.1 Conclusion

The thesis started with an introduction about the HRA approach. In the HRA, a large number of actuation elements work together as a single actuator. Any individual actuation element would not have enough power to satisfy the performance requirements. Travel, velocity and force of the actuator elements are added up depending on the exact configuration, and together they exceed the requirements by a certain margin (determined by the designer). Element failures will lead to performance degradation depending on the fault and the configuration. However, the dimensioning is such that the remaining actuators can still satisfy the performance requirements, so that the system remains operational. This approach is very different from existing fault tolerant control system approaches in that it engages a more complicated mechanical structure rather than a more complex control structure. Neither FDI nor reconfiguration is necessary in this approach. By removing the possibility of failure in FDI or reconfiguration and accommodating failures in individual actuation elements, the HRA can avoid failing to finish its function and, in doing so, provides a higher reliability of the whole actuator system. The research described in this thesis spans three main areas: Concepts and Configurations, Control, and Demonstration. Each of these is summarised.

The second chapter introduces a literature survey, which verifies that the HRA approach is very different from existing fault tolerant control system approaches. The traditional Fault Tolerant Control system uses a relatively small number of "control channels" to provide functional redundancy, possibly supported by analytical redundancy but almost invariably involving fault detection and isolation and sometimes reconfiguration of the control action. Four main components are included in the scheme: the plant itself (including sensors and actuators), the fault detection and isolation (FDI) unit,

the feedback (or feed-forward) controller, and the supervision system. Techniques about the FDI, reconfiguration control and robust control are introduced in Section 2.2. Combining these techniques, a design progress of fault tolerant system is introduced in Section 2.3. The HRA proposes an alternative route to fault tolerant actuation by engaging a relatively complex mechanical structure so that FDI or reconfiguration is not necessary. But the approach also introduces new research challenges as concluded in Section 2.4: how to connect the redundancy actuation elements, and how to control such a complex configuration. Both questions will be answered in the thesis using both simulation studies and hardware demonstrator.

A modelling study is introduced in Chapter 3 to explore some features of some mechanical structures so that an optimizing structure can be used for the control study and hardware demonstration. Using electro-mechanical actuation technology, an individual actuator is firstly modelled in Simulink as well as three kinds of actuator faults: open circuit, short circuit and lockup. All three kinds of faults will cause a failure of the individual actuator, which are needed to be accommodated in a HRA. Two basic configurations, series and parallel configurations, are then modelled and tested under healthy and faults injected situations. By adding up the output of each individual element, the series configuration can provide a higher output for both position and velocity, while the parallel configuration can perform better for force output. The faults injected studies illustrate that the series configuration is sensitive to an open circuit fault (force capability loss) and the parallel configuration is sensitive to a lockup fault (travel capability loss), but the open circuit fault is easier to be accommodated especially when the load's mass (inertia) is not very big. Then, two HRA structures, parallel-in-series and series-in-parallel, can be got and modelled at the two-by-two level by comprising both parallel and series configurations. Both structures can double the position, velocity and force outputs compared with an individual actuator. When there is only

one faulty element (open circuit or lockup), both structures will not fail although performance degradation can be expected. When there are two faulty elements, the performance will depend on the positions of the faults. Because the PS structure comprises two parallel and one series configurations, it will be more sensitive to the lockup faults. The SP structure will be more sensitive to the open circuit fault because it comprises two series and one parallel configurations. Considering that the open circuit (force loss) is more easily accommodated than the lockup (travelling loss), the two-by-two series-in-parallel structure is a better structure. It is also will be used for the control design studies which are introduced in Chapter 4.

In Chapter 4, an open loop robust analysis in frequency domain is firstly given based on the state-space model of the two-by-two SP structure. The analysis shows that the structure's frequency domain features will not change much when faults are injected until the whole structure fails. This demonstrates that, in a HRA, faults can be accommodated by the mechanical structure so that it gives a wide range of possible controllers. Two control structures are then introduced: the first contains only one controller (global controller) based on the feedback information of the overall HRA, which is called a voltage-driven structure; the second uses local controllers based on the feedback information of each element. Since current control is used for the local controllers, the second control structure is called current-driven structure. Position feedback control using both classical and optimal (LQG) algorithms is applied. When applying LQG control algorithms, repeated dynamics in the state-space model causes mathematical problems for MATLAB to calculate the control regulator. Three approaches are applied to avoid this problem: parameter variation, the balanced realization truncation (mathematical reduction) and the physical reduction. Together with the two classical approaches (voltage-driven and current-driven), all five controllers work well and similarly under a healthy situation using the voltage-driven

classical controller as a benchmark.

The five controllers are compared under both time and frequency domains in Chapter 5. A healthy situation is firstly listed. As introduced in Chapter 4, all controllers are designed using the voltage-driven classical controller as a benchmark so that no big difference will be found. When the three kinds of faults introduced in Chapter 3 are injected (assuming one faulty element), the controllers are unchanged (i.e. no reconfiguration). All five controllers show performance degradation but still can finish the tracking function. Among the three kinds of faults, the lockup fault causes the worst performance degradation to all five controllers, while the open circuit fault causes least degradation. For the five controllers, it is difficult to find the best one by estimating the performance degradation under faults injected situations. Different controller is sensitive to different fault. For example, the optimal controllers are very sensitive to the open circuit fault because they take much longer time to reach the steady state when open circuit happens. Also the result may be changed when using different design of controllers, but these simulation test results demonstrate a fault tolerance ability of an HRA under different kinds of faults.

Based on the simulation studies, a lab-scale hardware demonstration HRA is built up engaging a two-by-two series-in-parallel mechanical structure. It is controlled in real time by an xPC Target system using both classical and optimal controllers. The world's first HRA (albeit with relatively low levels of actuation elements) allows verification of the simulation results (both open and closed loop) and proves that it is viable to control such an actuator without recourse to overly complicated controller designs or indeed controller reconfiguration. Two sets of controllers are designed, the low gain set and the high gain set. The low gain set controllers are firstly designed to avoid any permanent damage to the hardware. Then a set of controllers using

higher gain and full ability of the actuation elements are designed. When using the high gain set of controllers, the voltage-driven classical controller is no longer the benchmark and the other four controllers will perform faster. The experimental results show that the hardware HRA is easily to be controlled and can provide the fault tolerant ability without any reconfiguration, although performance degradation will happen. For the controller design, it can be concluded that the HRA will perform better using a high gain design with faster response and less degradation. In the fast design, the classical voltage-driven controller, which is the simplest control algorithm, performs worst (with slowest response and biggest degradation). The classical current-driven controller performs similarly with optimal controllers although the current-driven controller requires current sensors while the optimal controllers are model based which make the design progress complicated. In the optimal controllers, the physical reduction method could be concluded as the best approach because it can avoid the full order modal which could be very complicated when the number of elements inside the HRA goes higher while providing a similar performance with the full order and mathematical reduction algorithms. Based on all these results, it can be concluded that the fast design of the classical current-driven controller and the optimal physical reduction controllers are better choices for the HRA.

7.2 Suggestion for Further Work

The work introduced in this thesis has provided an important start for the High Redundancy Actuator research. Relatively simple mechanical structures as well as control structures are considered. The choices of faults are limited as well. Some future studies could include: using different types of actuation elements, e.g. electro-magnetic actuation; studies about some more kinds of faults; extending the mechanical structure to comprise a relatively large number of elements, e.g. four-by-four, and finally a hundred-by-hundred structure can be considered; robust control and force feedback control studies. A health monitoring system is also required. The FDI is not necessary in the HRA, but it is still can be used to monitor the system so that people can look after the actuation system carefully. Another choice could be a performance degradation monitoring system so that the HRA can be replaced before the performance become unacceptable. Some industry application studies also are necessary. Although some mechanical structures and controllability of the HRA have been studied, the research is not application specific. Some industry application studies, e.g. flight control system, railway active suspension system, will be very useful to explore whether the HRA can improve the performances in a real world. A software demonstrator could be developed to show the performance with high levels of redundancy, and also for the application studies. Work on some of these aspects is continuing via an EPSRC funded research project.

Reference

1. R. Patton, Fault Tolerant Control: The 1997 Situation, SAFE-PROCESS'97, IFAC Symposium on fault detection, supervision and safety, Kingston Upon Hull, UK, 1997, pp. 1033-1054
2. M. Blanke, M. Kinnaert, J. Lunze, M. and Staroswiecki, Diagnosis and Fault-Tolerant Control, Springer-Verlag, Berlin Heidelberg, 2006
3. D Briere, C Favre, and P Traverse, A family of Fault Tolerant Systems: Electrical flight controls from Airbus A320/A330/A340 to future military transport aircraft, Microprocessors and Microsystems, Volume 19, Number 2, March 1995, pp. 75-82
4. Terry Ford, Actuation systems development, Aircraft Engineering and Aerospace Technology, Volume 70, Number 4, 1998, pp. 265-270
5. Mogens Blankel, Marcel Staroswiecki, and N. Eva Wu, Concepts and Methods in Fault-tolerant Control, Proceedings of the American Control Conference, Arlington, Volume 4, 2001, pp. 2606-2620
6. Giovanni Betta, and Antonio Pietrosanto, Instrument Fault

Detection and Isolation: State of the Art and New Research Trends, IEEE Transactions on Instrumentation and Measurement, Volume 49, No. 1, 2000, pp. 100-107

7. Youmin Zhang, and Jin Jiang, Condition Monitoring and Fault Detection of a Compressor Using Signal Processing Techniques, Proceedings of the American Control Conference, Volume 6, 2001, pp. 4460-4465
8. R. Isermann, Process fault detection based on modeling and estimation methods – A survey, Automatica, Volume 20, No. 4, 1984, pp. 387-404
9. Ch. M. Hajiyev, and F. Caliskan, Integrated sensor/actuator FDI and reconfigurable control for fault-tolerant flight control system design, The Aeronautical Journal, Volume 105, 2001, pp. 525-533
10. Ochi Yoshimasa, Application of Feedback Linearization Method in a Digital Restructurable Flight Control System, Journal of guidance, control, and dynamics, Volume 16, No. 1, 1993, pp. 111-117
11. Ochi Yoshimasa, and Kanai K., Design of restructurable flight control systems using feedback linearisation, Journal of Guidance, Control, and Dynamics, Volume 14, No.5, 1991, pp. 903-911

12. Morse W. D., and Ossman K. A., Reconfigurable control system design via perfect model following, International journal of control, Volume 56, No. 4, 1992, pp. 783-798
13. Ki-Seok Kim, Keum-Jin Lee, and Youdan Kim, Reconfigurable Flight Control System Design Using Direct Adaptive Method, Journal of Guidance, Control, and Dynamics, Volume 26, No. 4, 2003, pp.543-550
14. Z. Gao, and P. J. Antsaklis, Pseudo-inverse method for reconfigurable control with guaranteed stability, Proc. 11th IFAC World Congress, 1990
15. Stephen H. Lane and Robert F. Stengel, Flight Control Design Using Non-linear Inverse Dynamics, Automatica. Vol. 24, No. 4, 1988, pp. 471-483
16. R. J. Patton, J. Chen, and G. P. Liu, Robust fault detection of dynamic systems via genetic algorithms, Proc Instn Mech Engrs, Volume 211, Part 1, 1997, pp. 357-364
17. M. Karpenko, N. Sepehri, Robust Position Control of an Electro-hydraulic Actuator With a Faulty Actuator Piston Seal, Journal of Dynamic Systems, Measurement, and Control, Volume 125, 2003, pp. 413-423
18. Shufan Wu, Grimble M. J., and Wei Wei, QFT-based robust/fault-tolerant flight control design for a remote

pilotless vehicle, IEEE Transactions on Control Systems Technology, Volume 8, No.6, 2000, pp. 1010-1016

19. Williams S., and Hyde R. A., A comparison of characteristic locus and H_∞ design methods for VSTOL flight control system design, Proc of ACC '90, 1990, pp. 2508-2513
20. N. E. Wu, An integrated approach to controls and diagnostics using the four parameter controller, IEEE Control Systems Magazine, Volume 11, No. 6, 1991, pp. 22-29
21. Murad G. A., Postlethwaite I., and Gu D. W., A robust design approach to integrated controls and diagnostics, 13th IFAC World Congress, 1996, pp. 199-204
22. Youmin Zhang, and Jin Jiang, Fault tolerant control system design with explicit consideration of performance degradation, IEEE Transactions on Aerospace and Electronic Systems, Volume 39, No. 3, 2003, pp. 838-848
23. Thorbjorn Ebefors, Johan Ulfstedt Mattsson, Edvard K¨alvesten and Goran Stemme, A robust micro conveyer realized by arrayed polyimide joint actuators, Micro Electro Mechanical Systems, 1999, pp. 576–581
24. Jonathan E. Luntz, William Messner, and Howie Choset,

Parcel Manipulation and Dynamics with a Distributed Actuator Array: The Virtual Vehicle, IEEE International Conference on Robotics and Automation Proceedings, 1997, Volume 2, pp. 1541-1546

25. Xinli Du, Roger Dixon, Roger M. Goodall, and Argyrios C. Zolotas, Assessment of strategies for control of high redundancy actuators. In Proceedings of the International Conference on New Actuators (ACTUATOR 2006), 2006
26. Xinli Du, Roger Dixon, Roger M. Goodall, and Argyrios C. Zolotas, Modelling And Control Of A Highly Redundant Actuator, In Proceedings of the International Conference on Control (CONTROL 2006), 2006
27. Xinli Du, Roger Dixon, Roger M. Goodall, and Argyrios C. Zolotas, LQG control for a high redundancy actuator, In Proceedings of the International Conference on Advanced Intelligent Mechatronics (ASME 2007), 2007
28. Xinli Du, Roger Dixon, Roger M. Goodall, and Thomas Steffen, Validation of a Two by Two High Redundancy Actuator – Experimental Results, In Proceedings of the International Conference on New Actuators (ACTUATOR 2008), 2008
29. Xinli Du, Roger Dixon, Roger M. Goodall, and Argyrios

C. Zolotas, Modelling And Control Of A High Redundancy Actuator, The Science of Intelligent Machines, A Journal of IFAC, the International Federation of Automatic Control

30. Mclennan Servo Supplies Ltd.
http://www.mclennan.co.uk/

31. Ian Pratt (1996). Active suspension applied to railway train, PhD thesis in Loughborough University, pp. 59-66

32. xPC Target System
http://www.mathworks.com/products/xpctarget/

Appendix 1

Hardware Technical Specification

Technical specification and performance of the DC servo motor

Technical Specification			
Maximum supply voltage	40Vdc	Maximum continuous torque	14Ncm
Motor voltage constant	10.3 V.1000rpm	Motor torque constant	9Ncm/A
Maximum peak torque	36Ncm	Mechanical time constant	20ms
Rotor inertia	0.214kgcm2	Terminal resistance	7.8Ohms
Rotor inductance	5.0mH	Rotor construction	ironless
Commutation	carbon	Bearings	ball
Maximum axial force	15N	Maximum radial force	100N
Mass	1.03Kg		
Performance @ 24Vdc			
No load speed	2300rpm	Rated speed	1600rpm
Rated torque	12Ncm	Peak torque	27Ncm

Technical specification of the mechanical actuator

Positioning accuracy reproducibility	0.004mm	Positioning	0.04mm
Running parallelism	0.01mm	Backlash	0.01mm
Starting torque	0.4Ncm	Maximum Running Velocity	160mm/s
Lead of ball screw	2mm	Overall length	204mm
Mass	0.28Kg		

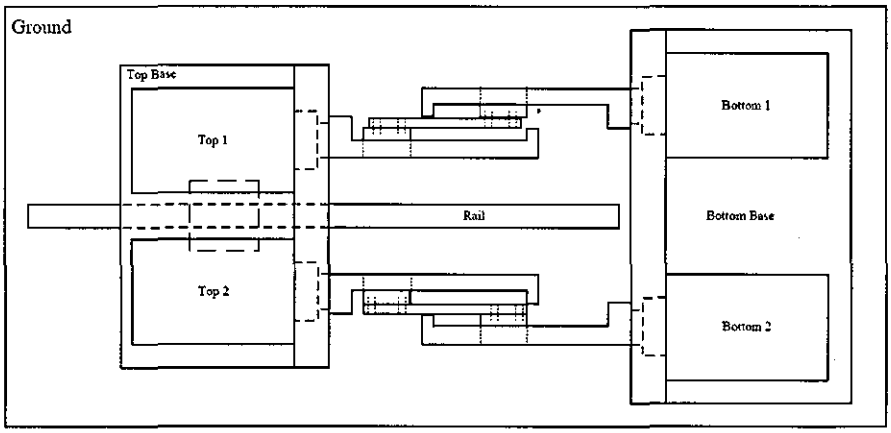
Technical specification of the encoder

Line count options	500	Supply voltage	5Vdc
Output signal	Dual track	Signal format	TTL
Max. output frequency	100kHz	Max. acceleration	250krad/sec2

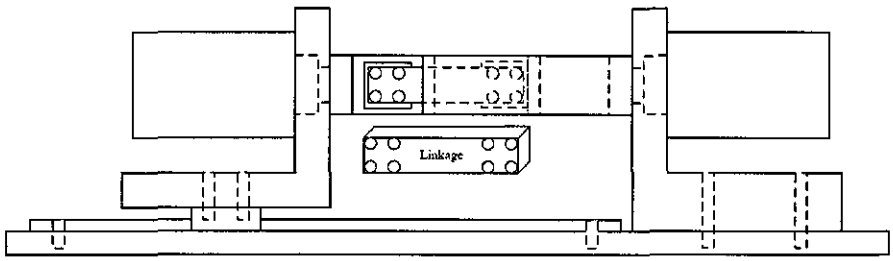
Appendix 2

Mechanical Design of the Hardware HRA

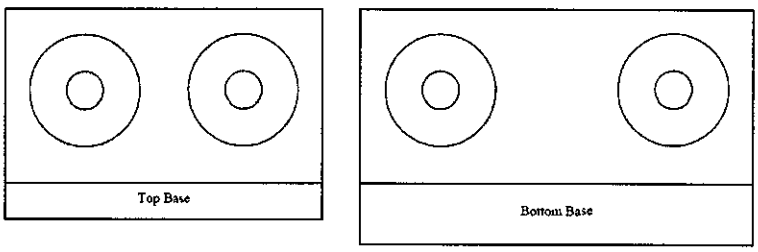
Plan view of the hardware 2x2 HRA



Side view of the hardware 2x2 HRA



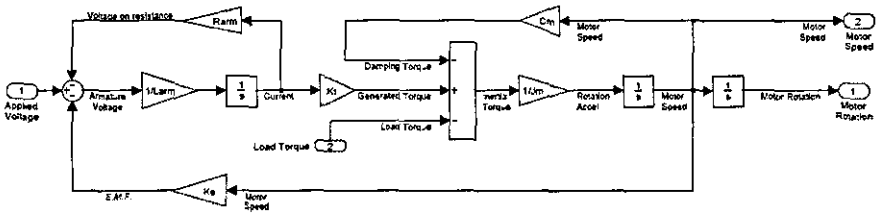
End view of bottom and top bases



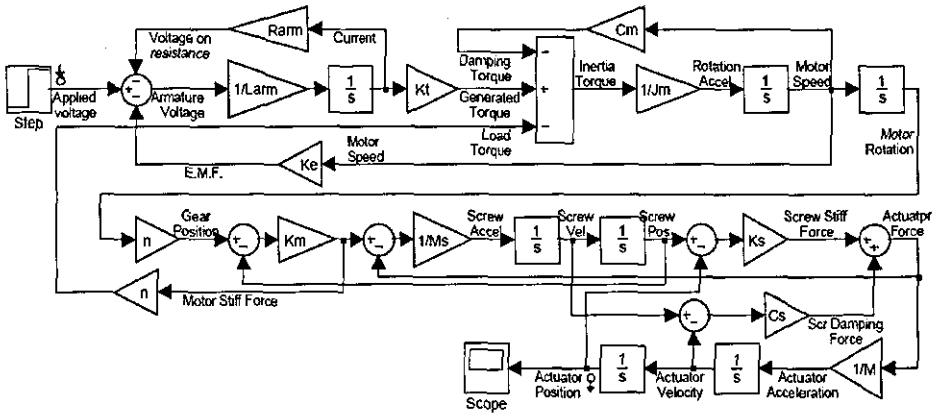
Appendix 3

Simulink Model Library

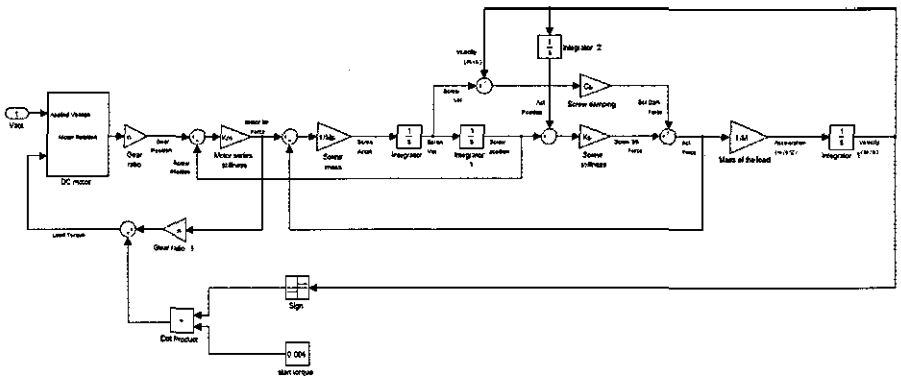
The DC motor linear model



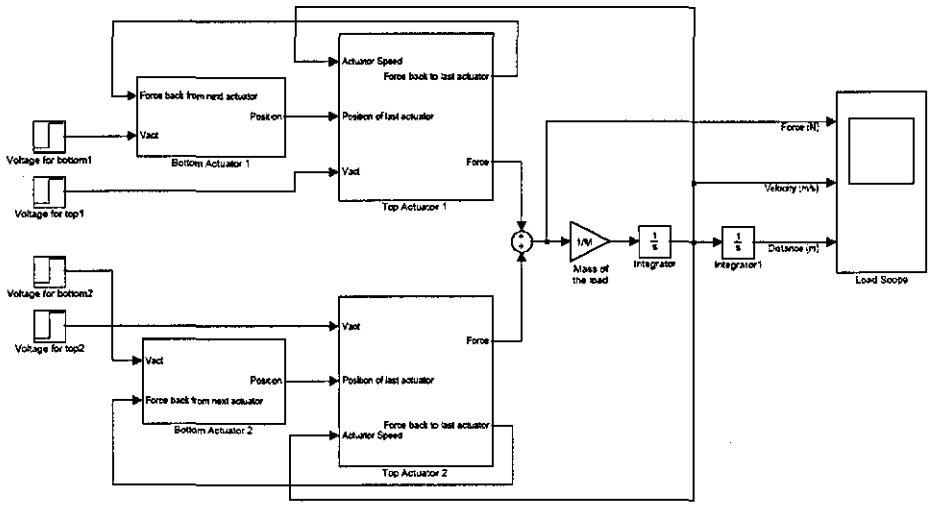
The linear individual electro-mechanical actuator model



The non-linear individual electro-mechanical actuator model



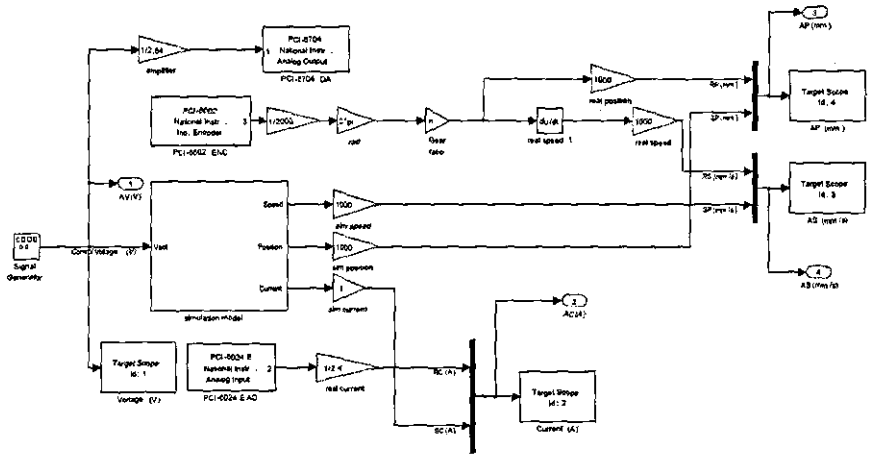
The two-by-two series-in-parallel HRA model



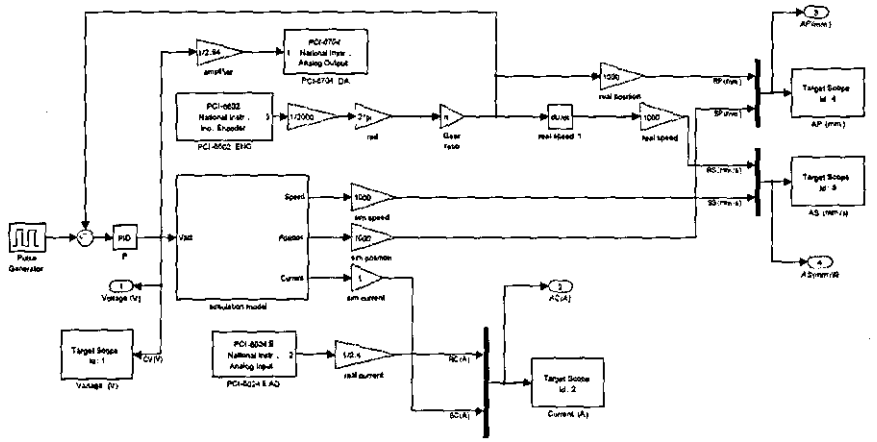
Appendix 4

xPC Target Model Library

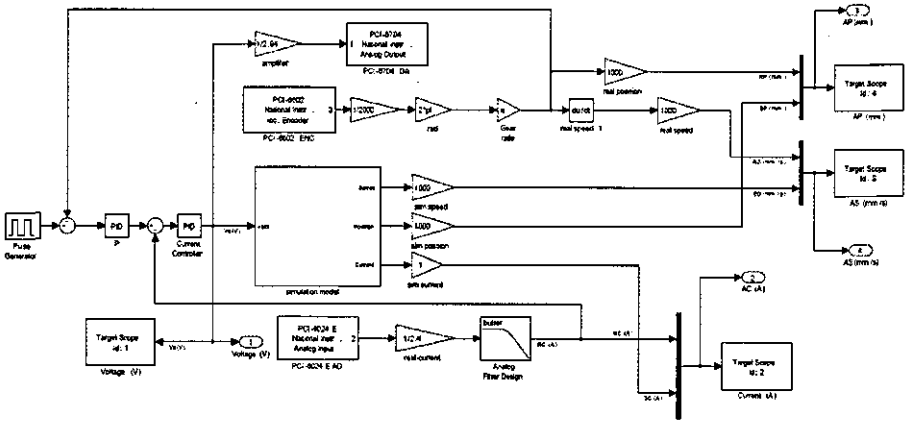
Test of individual actuator



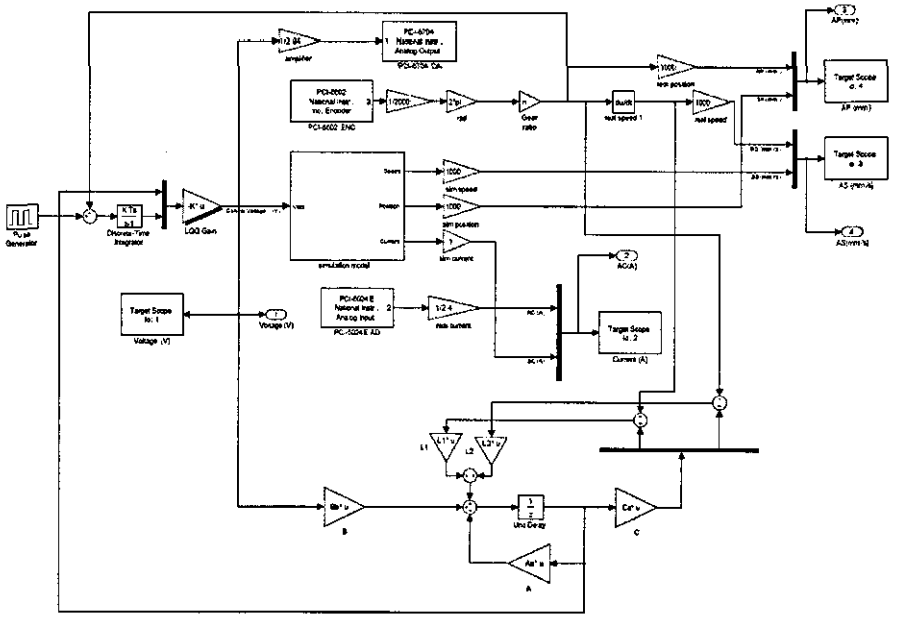
Classical voltage-driven control of individual actuator



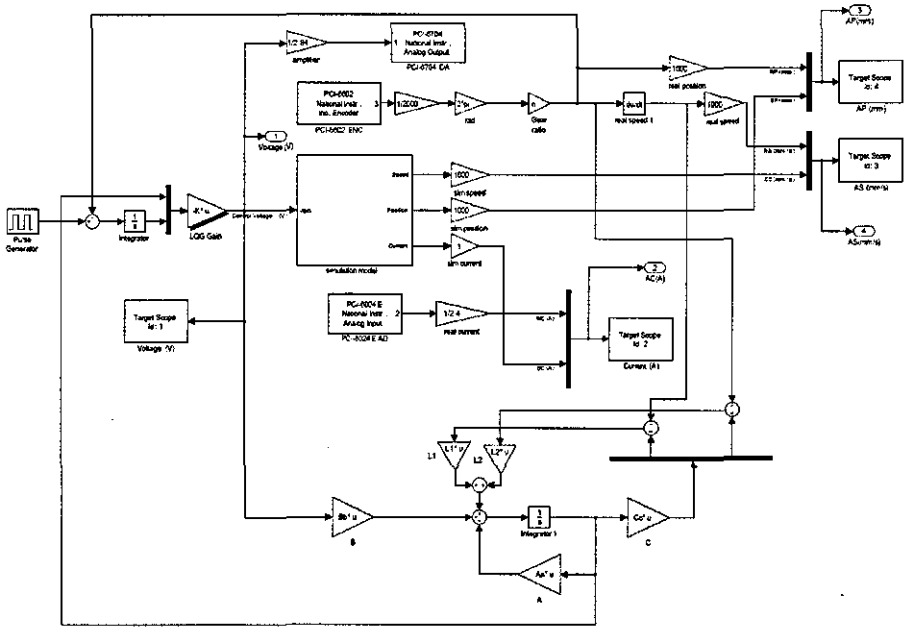
Classical current-driven control of individual actuator



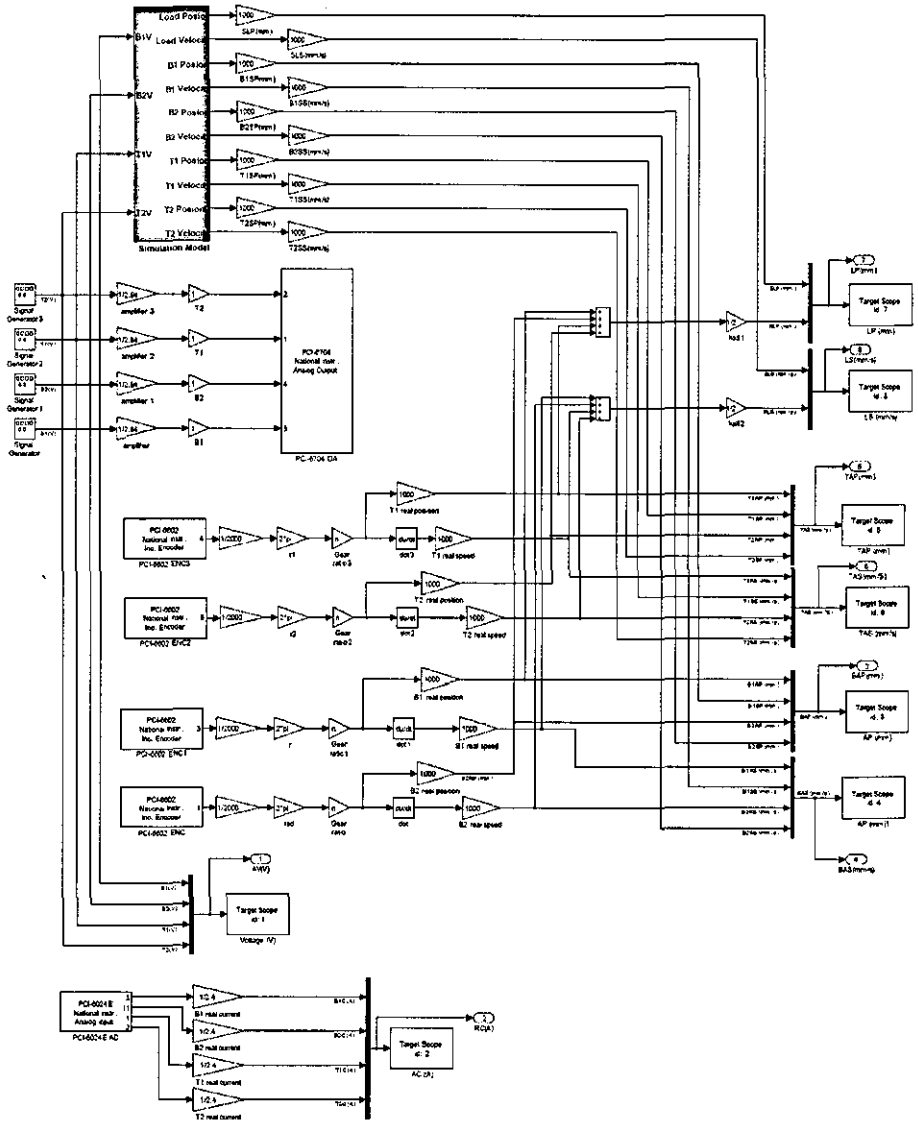
Discrete-time LQG control of individual actuator



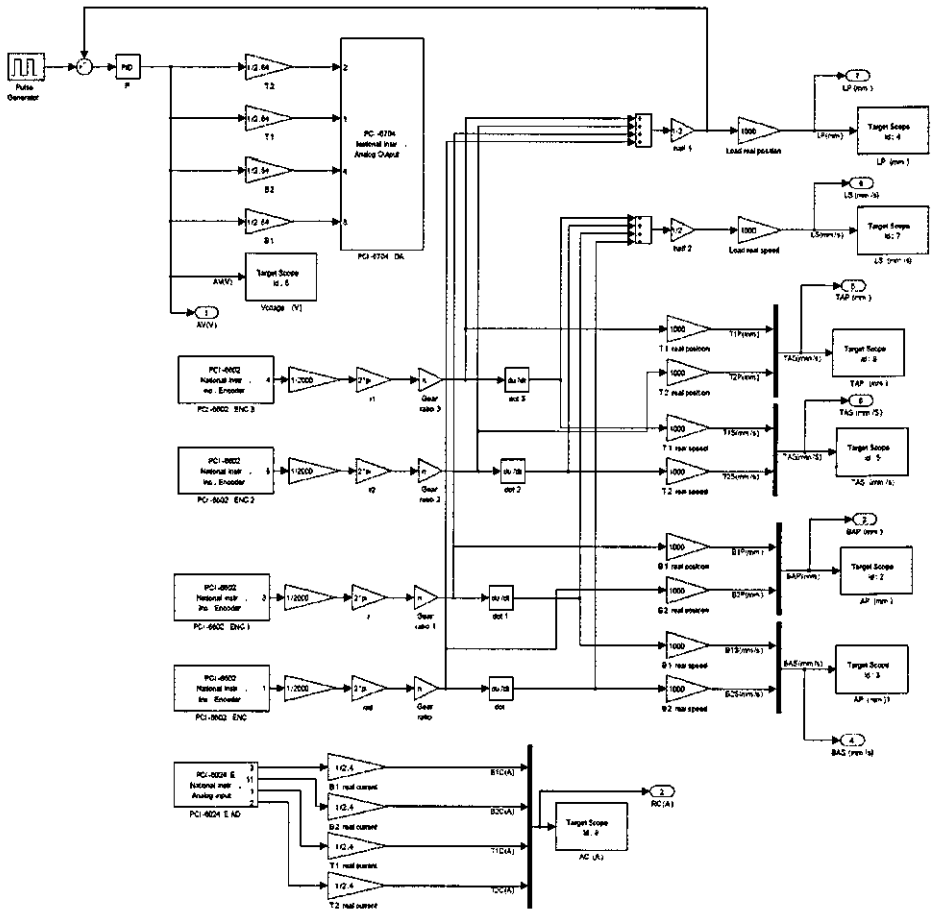
Continuous-time LQG control of individual actuator



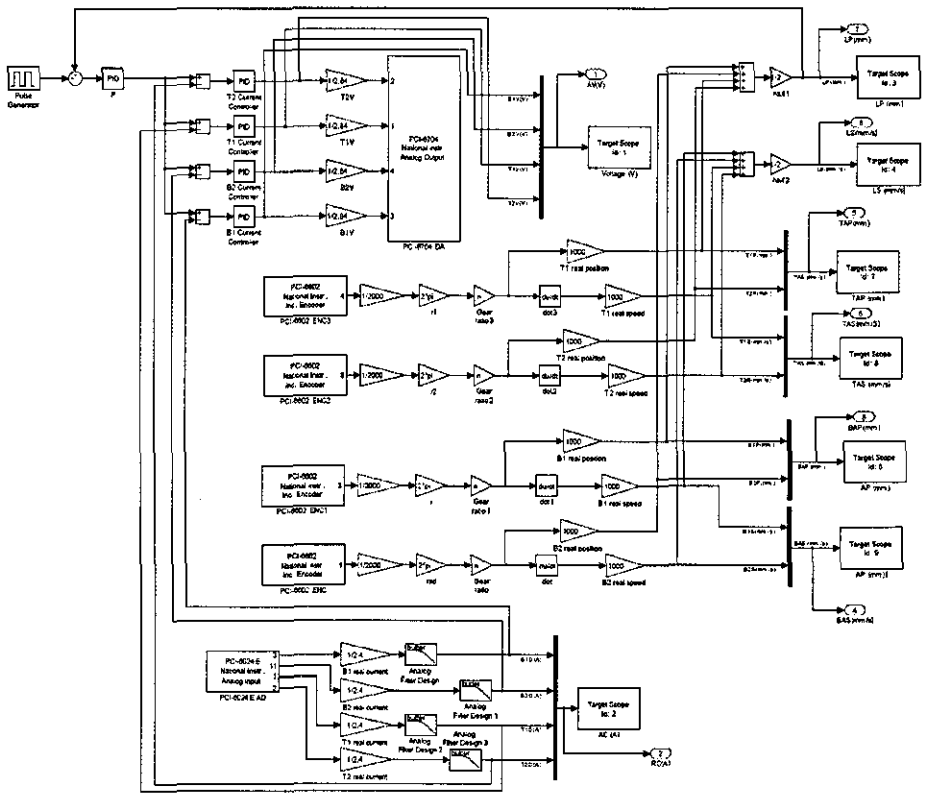
Test of HRA



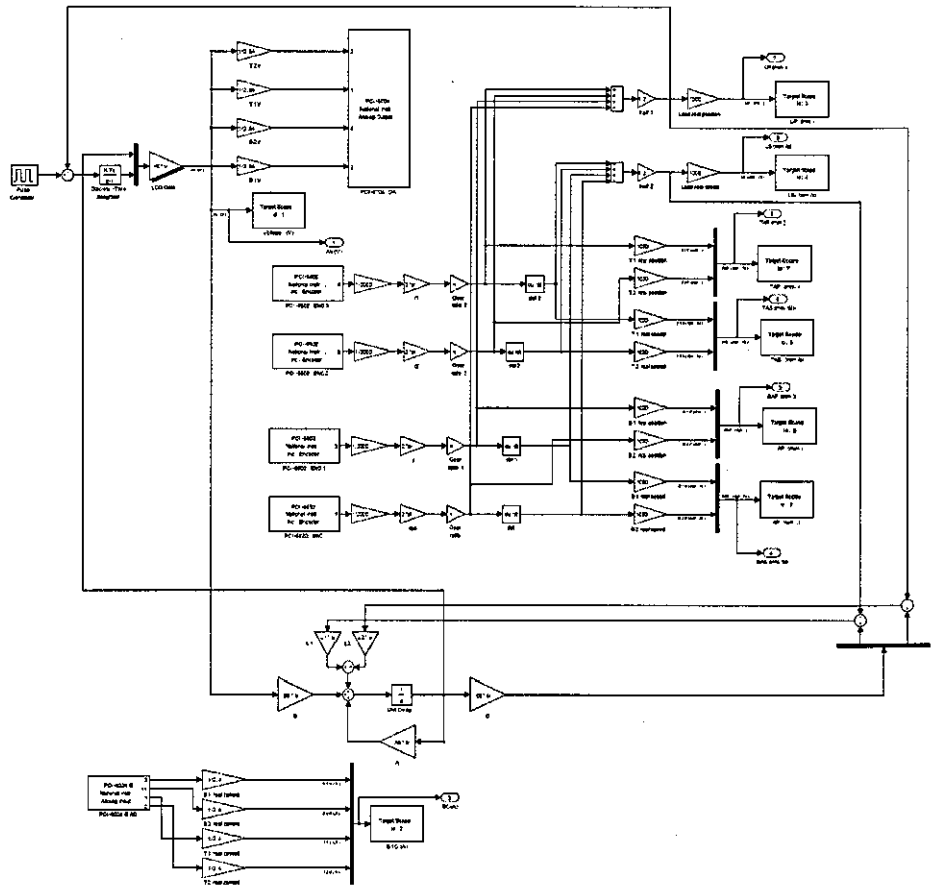
Classical voltage-driven control of HRA



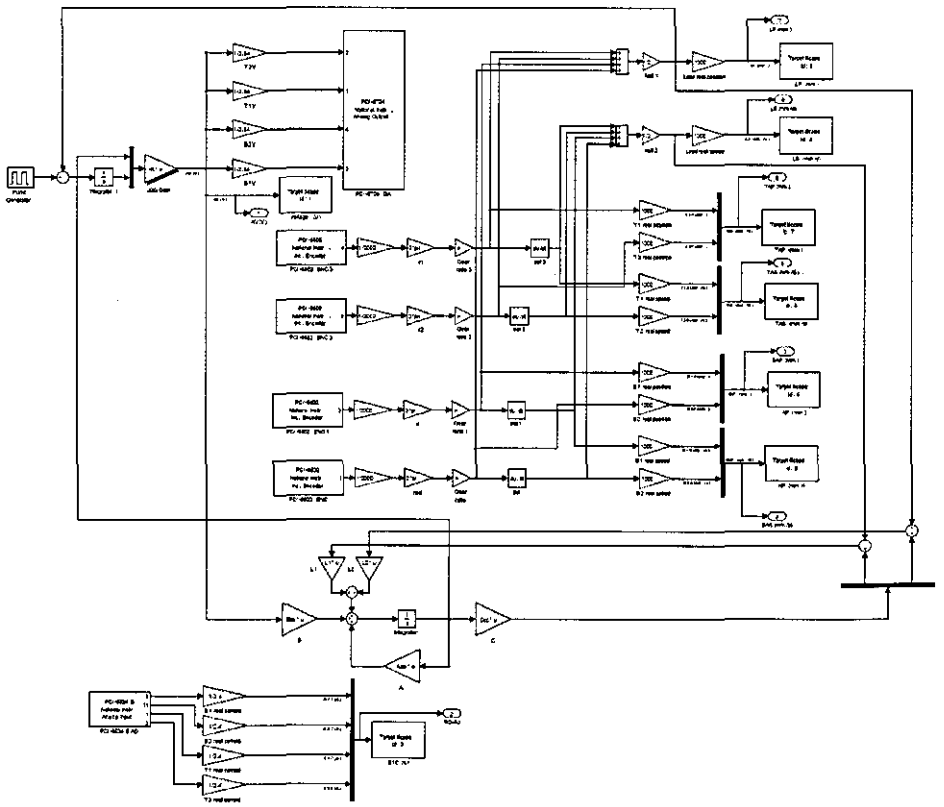
Classical current-driven control of HRA



Full order LQG control of HRA



Model reduction LQG control of HRA



Appendix 5

xPC Target Control Strategies

Individual Actuator State-space Model

```
%Individual actuator state-space model (1I1O)
%x = [i thita(dot) thita xs(dot) xs xl(dot) xl]'
%y = [i xl]'
%u = [Va]
%A
A = [-Rarm/Larm -Ke/Larm 0 0 0 0;
     Kt/Jm -Cm/Jm -n^2*Km/Jm 0 n*Km/Jm 0 0;
     0 1 0 0 0 0;
     0 0 Km*n/Ms -Cs/Ms (-Ks-Km)/Ms Cs/Ms Ks/Ms;
     0 0 0 1 0 0;
     0 0 0 Cs/M Ks/M -Cs/M -Ks/M;
     0 0 0 0 0 1 0];
%B
B = [1/Larm; 0; 0; 0; 0; 0; 0];
%C
C = [0 0 0 0 0 1 0; 0 0 0 0 0 0 1];
%D
D = [0; 0];
%continuous-time model
sys = ss(A, B, C, D);
%discrete-time model
sysd = c2d(sys, Ts);
[Aa, Bb, Cc, Dd] = ssdata(sysd);
```

Discrete-time LQG Control for Individual Actuator

```
%State-space model for tracking system
%x1=[x e(int)]' where e = r - xl
%A1
A1 = A;
A1(1:8, 8) = 0;
A1(8, 1:7) = 0;
A1(8, 7) = -1;
%B1
B1 = B;
B1(8, 1) = 0;
%C1
C1 = C;
C1(:, 8) = 0;
%D1
D1 = D;
%sys1
sys1 = ss(A1, B1, C1, D1);
sys1d = c2d (sys1, Ts);
[Aa1, Bb1, Cc1, Dd1] = ssdata(sys1d);

%LQR regulator
Q = 1e-1*eye(8);
Q(8,8) = 1e7;
R = 1e-1;
K = dlqr(Aa1, Bb1, Q, R, 0);

%Kalman Filter regulator
Qn = 1;
```

```
Rn = eye(2);
[kest,Ld,Pd,Md] = kalman(sysd,Qn,Rn,0);
L1 = Ld(:, 1);
L2 = Ld(:, 2);
```

Continuous-time LQG Control for Individual Actuator

```
%State-space model for tracking system
%x1=[x e(int)]' where e = r - xl
%A1
A1 = Aa;
A1(1:4, 4) = 0;
A1(4, 1:3) = 0;
A1(4, 3) = -1;
%B1
B1 = Bb;
B1(4, 1) = 0;
%C1
C1 = Cc;
C1(:, 4) = 0;
%D1
D1 = Dd;
%sys1
sys1 = ss(A1, B1, C1, D1);

%LQR regulator
Q = 1e-1*eye(4);
Q(4,4) = 1e13;
R = 1e-1;
K = lqr(sys1, Q, R);

%Kalman Filter regulator
Qn = 1;
Rn = eye(2);
[kest,L,Pp] = kalman(sysb,Qn,Rn,0);
```

```
L1 = L(:, 1);
L2 = L(:, 2);
```

State-space Model for Two-by-Two Series-in-Parallel HRA

```
%State-space model (1I2O)
%x = [ib1 thitab1(dot) thitab1 xbs1(dot) xbs1 xtm1(dot) xtm1 it1 thitat1(dot)
thitat1 xtm1(dot) xtm1 ib2 thitab2(dot) thitab2 xbs2(dot) xbs2 xtm1(dot)
xtm2 it2 thitat2(dot) thitat2 xtm2(dot) xtm2 xl(dot) xl]'
%y = [xl' xl]
%u = [Va]
%A
A = [-Rarm/Larm -Ke/Larm 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
      Kt/Jm -Cm/Jm -n*n*Km/Jm 0 n*Km/Jm 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0;
      0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
      0 0 Km*n/Ms -Cs/Ms (-Ks-Km)/Ms Cs/Ms Ks/Ms 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0;
      0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
      0 0 0 Cs/Mm Ks/Mm -Cs/Mm (-Ks-Km)/Mm 0 0 -n*Km/Mm 0
Km/Mm 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
      0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
      0 0 0 0 0 0 0 -Rarm/Larm -Ke/Larm 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
      0 0 0 0 0 0 -n*Km/Jm Kt/Jm -Cm/Jm -n*n*Km/Jm 0 n*Km/Jm 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0;
      0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
      0 0 0 0 0 0 Km/Ms 0 0 Km*n/Ms -Cs/Ms (-Ks-Km)/Ms 0 0 0 0 0 0 0 0
0 0 0 0 Cs/Ms Ks/Ms;
      0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
      0 0 0 0 0 0 0 0 0 0 0 0 -Rarm/Larm -Ke/Larm 0 0 0 0 0 0 0 0 0 0 0 0;
      0 0 0 0 0 0 0 0 0 0 0 0 Kt/Jm -Cm/Jm -n*n*Km/Jm 0 n*Km/Jm 0 0 0
0 0 0 0 0 0;
      0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
```

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 Km*n/Ms -Cs/Ms (-Ks-Km)/Ms Cs/Ms
Ks/Ms 0 0 0 0 0 0;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 Cs/Mm Ks/Mm -Cs/Mm (-Ks-Km)/Mm 0
0 -n*Km/Mm 0 Km/Mm 0 0;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -Rarm/Larm -Ke/Larm 0 0 0 0 0;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -n*Km/Jm Kt/Jm -Cm/Jm
-n*n*Km/Jm 0 n*Km/Jm 0 0;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 Km/Ms 0 0 Km*n/Ms -Cs/Ms
(-Ks-Km)/Ms Cs/Ms Ks/Ms;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0;
0 0 0 0 0 0 0 0 0 0 Cs/M Ks/M 0 0 0 0 0 0 0 0 0 0 Cs/M Ks/M
-2*Cs/M -2*Ks/M;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0];
%B
B = [1/Larm; 0; 0; 0; 0; 0; 0; 1/Larm; 0; 0; 0; 0; 1/Larm; 0; 0; 0; 0; 0; 0;
1/Larm; 0; 0; 0; 0; 0; 0];
%C
C = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0;
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1];
%Dd
D = [0; 0];
%continuous-time model
sys = ss(A, B, C, D);
%discrete-time model
sysd = c2d(sysm, Ts);
[Aa, Bb, Cc, Dd] = ssdata(sysd);
```

Full Order LQG Control for Two-by-Two Series-in-Parallel HRA

```
%State-space model after minim realization
sysm = minreal(sys);
[Am, Bm,Cm, Dm] = ssdata (sysm);

%State-space model for tracking system
%x1=[x e(int)]' where e = r - xl
%A1
A1 = Am;
A1(1:25, 25) = 0;
A1(25, 1:24) = -Cm(2, :);
A1(25, 25) = 0;
%B1
B1 = Bm;
B1(25, 1) = 0;
%C1
C1 = Cm;
C1(:, 25) = 0;
%D1
D1 = Dm;
%sys1
sys1 = ss(A1, B1, C1, D1);

%LQR regulator
Q=1e-2*eye(25);
Q(25,25)=1e7;
R=1;
K=lqrd(A1, B1, Q, R, Ts);
```

```
%Kalman Filter regulator
Qn=1;
Rn=0.01*eye(2);
[kest,L,P] = kalman(sysd,Qn,Rn,0);
L1 = L(:, 1);
L2 = L(:, 2);
```

Mathematical Model Reduction LQG Control for Two-by-Two Series-in-Parallel HRA

```
%State-space model after balanced realization truncation
sysm = minreal(sys);
sysb = balancmr(sysm,8);
[Aaa, Bbb, Ccc, Ddd] = ssdata(sysb);

%State-space model for tracking system
%x1=[x e(int)]' where e = r - xl
%A1
A1 = Aaa;
A1(1:9, 9) = 0;
A1(9, 1:8) = -Ccc(2, :);
A1(9, 9) = 0;
%B1
B1 = Bbb;
B1(9, 1) = 0;
%C1
C1 = Ccc;
C1(:, 9) = 0;
%D1
D1 = Ddd;
%sys1
sys1 = ss(A1, B1, C1, D1);

%LQR regulator
Q=1e-2*eye(9);
Q(9,9)= 1e7;
R=1;
K=lqr(sys1, Q, R);

%Kalman Filter regulator
Qn=1;
Rn=0.01*eye(2);
[kest,L,P] = kalman(sysb,Qn,Rn,0);
L1 = L(:, 1);
L2 = L(:, 2);
```

Physical Model Reduction LQG Control for Two-by-Two Series-in-Parallel HRA

```
%Physical reduction state-space model (1I2O)
%x = [i thita(dot) thita xs(dot) xs xl(dot) xl]'
%y = [i xl]'
%u = [Va]
%A
A = [-Rarm/Larm -Ke/Larm 0 0 0 0;
     Kt/Jm -Cm/Jm -n^2*Km/Jm 0 n*Km/Jm 0 0;
     0 1 0 0 0 0;
     0 0 Km*n/Ms -Cs/Ms (-Ks-Km)/Ms Cs/Ms Ks/Ms;
     0 0 0 1 0 0 0;
     0 0 0 Cs/M Ks/M -Cs/M -Ks/M;
     0 0 0 0 0 1 0];
%B
B = 4*[1/Larm; 0; 0; 0; 0; 0; 0];
%C
C = [0 0 0 0 0 0.5 0;
     0 0 0 0 0 0 0.5];
%D
D = [0; 0];
%sys
sys = ss(A, B, C, D);

%State-space model for tracking system
%x1=[x e(int)]' where e = r - xl
%A1
A1 = A;
A1(1:8, 8) = 0;
```

```
A1(8, 1:7) = 0;
A1(8, 7) = -1;
%B1
B1 = B;
B1(8, 1) = 0;
%C1
C1 = C;
C1(:, 8) = 0;
%D1
D1 = D;
%sys1
sys1 = ss(A1, B1, C1, D1);

%LQR regulator
Q = 1e-3*eye(8);
Q(8,8) = 1e7;
R = 1;
K = lqr(sys1, Q, R);

%Kalman Filter
Qn = 1;
Rn = 0.01*eye(2);
[kest,L,Pp] = kalman(sys,Qn,Rn,0);
L1 = L(:, 1);
L2 = L(:, 2);
```