

This item was submitted to Loughborough's Institutional Repository (<https://dspace.lboro.ac.uk/>) by the author and is made available under the following Creative Commons Licence conditions.

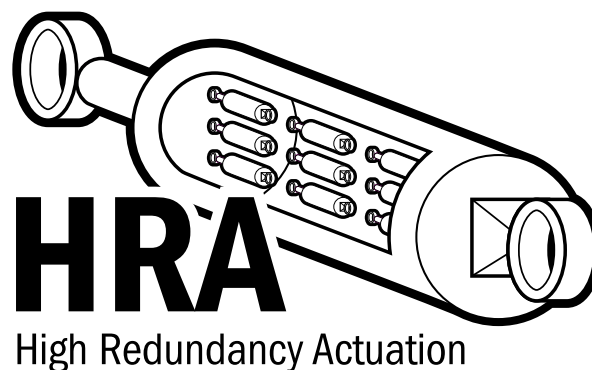


For the full text of this licence, please go to:
<http://creativecommons.org/licenses/by-nc-nd/2.5/>

Modelling, Control and Monitoring of High Redundancy Actuation

by

Jessica Davies



A doctoral thesis submitted in partial fulfillment of the requirements

for the award of Doctor of Philosophy of

Loughborough University

December, 2009

© *Jessica Davies 2009*

Abstract

The High Redundancy Actuator project investigates a novel approach to fault tolerant actuation, which uses a relatively high number of small actuation elements, assembled in series and parallel in order to form a single actuator which has intrinsic fault tolerance. Faults in elements will affect the maximum capability of the overall actuator, but through control techniques, the required performance can be maintained. This allows higher levels of reliability to be attained in exchange for less over-dimensioning in comparison to conventional redundancy techniques. In addition, the combination of both serial and parallel elements allows for the intrinsic accommodation of both lock-up (loss of travel) and loose (loss of force) faults.

Research to date has concentrated on high redundancy actuators based on electromechanical technology, of relatively low order (4 elements), which are controlled through passive fault tolerant control methods. The objective of this thesis is to expand upon this work. High redundancy actuator configurations of higher order (16-100 elements), formed from electromagnetic actuators are considered. An element model for a moving coil actuator is derived from first principles and verified experimentally. This element model is then used to form high-order, non-linear high redundancy actuator models for simulation, and reduced-order representations for control design purposes.

A simple, passive fault tolerant control design is then made for the high redundancy actuation configurations, the results of which are compared to a decentralised, multiple-model, gain-scheduling approach applied through a framework based upon multi-agent concepts. The results of this comparison indicate that limited fault tolerance can be achieved through simple passive control methods, however, performance degradation occurs, and requirements are not met under theoretically tolerable fault levels. The active fault tolerant control approach, which detects faults in elements and reconfigures the control within the remaining active elements, offers substantial performance improvements, meeting the requirements of the system under the vast majority of theoretically tolerable fault scenarios. However, these improvements are made at the cost of increased system complexity and a performance that relies upon the accuracy of the fault detection.

Fault detection and health monitoring of the high redundancy actuator is also explored. A simple rule-based approach to fault detection and diagnosis, for use within the decentralised active fault tolerant control method, is described and simulated. An interacting multiple model fault detection approach is also examined, which is more suitable for health monitoring within a centralised passive control scheme. Both of these methods provide the required level of fault information for their respective purposes. However, they achieve this through the introduction of complexity. The rule-based method increases system complexity, requiring high levels of instrumentation, and conversely the interacting multiple model approach involves complexity of design and computation.

Finally, the development of a software demonstrator is described. Experimental rigs at the current project phase are restricted to relatively low numbers of elements (4-16) for practical

reasons such as cost, space and technological limitations. Hence, a software demonstrator has been developed which provides a visual representation of high redundancy actuators with larger numbers of elements, and varied configuration for further demonstration of this concept. The software demonstrator is developed in Matlab/Simulink and includes features such as animated visualisation of the system and health monitoring, real-time fault injection, input and disturbance controls, and controller choice.

List of Figures

1.1	Conventional sensor and actuator redundancy.	1
1.2	High redundancy actuation.	2
1.3	High redundancy actuator configurations under fault.	4
2.1	Behavioural representation of nominal and faulty systems.	12
2.2	Process flow diagram for fault tolerant design(13).	13
2.3	Block diagram schematic of a redundant hydraulic actuator.	15
2.4	Diagrammatic representation of capabilities and graceful degradation in HRA.	16
2.5	Fault tolerant control strategies.	18
2.6	Typical active fault tolerant control scheme.	19
2.7	Multi-Agent system.	24
2.8	Multi-Agent applications.	25
2.9	Abstract agent architecture: Purely reactive agent.	27
2.10	Abstract agent architecture: Agent that maintains state.	27
2.11	Subsumption	29
2.12	Beliefs-Desires-Intentions architecture	30
2.13	Hybrid architectures	30
2.14	Hierarchical agency architecture	34
2.15	Autonomous agency architecture	34
2.16	Blackboard-based agency architecture	35
2.17	Facilitator agency architecture	36
2.18	Broker agency architecture	36
2.19	FIPA abstract architecture	37
3.1	Moving coil actuator.	43
3.2	Magnetic flux within the static system.	44
3.3	Electrical subsystem equivalent circuit.	47
3.4	Simplified electrical subsystem equivalent circuit.	47
3.5	Mechanical subsystem and its free-body diagram.	47
3.6	Simulink block diagram representation of the moving-coil actuator model.	49
3.7	Experiment setup.	51
3.8	Frequency response of free-moving coil: voltage-position.	52
3.9	Frequency response of free-moving coil: voltage-acceleration.	52

3.10	Frequency response of clamped and free-moving coil: voltage-current.	53
3.11	Model and experimental frequency response of clamped coil: voltage-current.	54
3.12	Model and experimental frequency response of free-moving coil: voltage-position.	56
3.13	Model and experimental frequency response of free-moving coil: voltage-acceleration.	56
3.14	Model and experimental frequency response of free-moving coil: voltage-current.	57
3.15	HRA model sizes for element models of varying order.	58
3.16	Bode diagrams of the 4th, 3rd and 2nd order models.	60
3.17	Additive error between the 4th, 3rd and 2nd order models.	61
3.18	Step response of the 4th, 3rd and 2nd order models.	61
3.19	Cut-bobbin electrical subsystem equivalent circuit.	62
3.20	Bode diagrams of the 4th, 3rd and 2nd order cut bobbin models.	63
3.21	Bode diagrams of the 4th, 3rd and 2nd order models.	64
3.22	Step response of the 4th, 3rd and 2nd order cut-bobbin models with unit voltage input.	64
3.23	Step response of travel limited element.	66
3.24	Two-by-two series-in-parallel assembly.	67
3.25	Free-body diagram of two-by-two series-in-parallel assembly.	67
3.26	Simulink subsystem model.	69
3.27	Element subsystem from Figure 3.26.	69
3.28	Mass subsystem from Figure 3.26.	69
3.29	Subsystem arrangement for two-by-two series-in-parallel element assembly.	70
3.30	Loose fault in parallel actuation elements	72
3.31	Loose fault in serial actuation elements	72
3.32	Lock-up fault in serial actuation elements	73
3.33	Lock-up fault in parallel actuation elements	73
3.34	Simulation of lock-up faults through passing the mass and impulse - Nominal system	75
3.35	Simulation of lock-up faults through passing the mass and impulse - Lock-up in element 2	76
5.1	Diagrammatic representation of passive and active fault tolerant control of HRA.	83
5.2	Passive fault tolerant control structure.	84
5.3	Series-in-Parallel and Parallel-in-Series HRA.	85
5.4	Frequency response of 4x4 PS HRA	87
5.5	Frequency response of 4x4 SP HRA	88
5.6	Frequency response of 10x10 PS HRA	88
5.7	Singular values for 4x4 PS HRA with 10% parameter uncertainty.	89
5.8	Singular values for 4x4 SP HRA with 10% parameter uncertainty.	90
5.9	Singular values for 10x10 PS HRA with 10% parameter uncertainty.	90
5.10	Singular values of additive error between nominal and parameter deviated 4x4 PS HRA.	91

5.11 Singular values of additive error between nominal and parameter deviated 4x4 SP HRA.	91
5.12 Singular values of additive error between nominal and parameter deviated 10x10 PS HRA.	92
5.13 Singular values for 4x4 PS HRA with loose faults	93
5.14 Singular values for 4x4 SP HRA with loose faults	93
5.15 Singular values for 10x10 PS HRA with loose faults	94
5.16 Singular values of additive error between nominal and 4x4 PS HRA with loose faults	94
5.17 Singular values of additive error between nominal and 4x4 SP HRA with loose faults	95
5.18 Singular values of additive error between nominal and 10x10 PS HRA with loose faults	95
5.19 Singular values for 4x4 PS HRA with lock-up faults	97
5.20 Singular values for 4x4 SP HRA with lock-up faults	98
5.21 Singular values for 10x10 PS HRA with lock-up faults	98
5.22 Singular values of additive error between nominal 4x4 PS HRA and 4x4 PS HRA with lock-up faults	99
5.23 Singular values of additive error between nominal 4x4 SP HRA and 4x4 SP HRA with lock-up faults	99
5.24 Singular values of additive error between nominal 10x10 PS HRA and 10x10 PS HRA with lock-up faults	100
5.25 Frequency response of 4x4 PS HRA with passive control	102
5.26 Step response of 4x4 PS HRA with passive control	103
5.27 Frequency response of 4x4 SP HRA with passive control	104
5.28 Step response of 4x4 SP HRA with passive control	105
5.29 Frequency response of 10x10 PS HRA with passive control	105
5.30 Step response of 10x10 PS HRA with passive control	106
5.31 Singular values for controlled 4x4 PS HRA with 10% parameter uncertainty .	107
5.32 Singular values for controlled 4x4 SP HRA with 10% parameter uncertainty .	107
5.33 Singular values for controlled 10x10 PS HRA with 10% parameter uncertainty	108
5.34 Step response of 4x4 PS HRA with passive control and parameter deviations .	108
5.35 Step response of 4x4 SP HRA with passive control and parameter deviations .	109
5.36 Step response of 10x10 PS HRA with passive control and parameter deviations	109
5.37 Singular values for controlled 4x4 PS HRA with loose faults	112
5.38 Singular values for controlled 4x4 SP HRA with loose faults	112
5.39 Singular values for controlled 10x10 PS HRA with loose faults	113
5.40 Step response of 4x4 PS HRA with passive control and loose faults	113
5.41 Step response of 4x4 SP HRA with passive control and loose faults	114
5.42 Step response of 10x10 PS HRA with passive control and loose faults	114
5.43 Response of SP 4×4 HRA with two loose faults in two branches to pulse train input.	116

5.44	Singular values for controlled 4x4 PS HRA with lock-up faults	117
5.45	Singular values for controlled 4x4 SP HRA with lock-up faults	117
5.46	Singular values for controlled 10x10 PS HRA with lock-up faults	118
5.47	Step response of 4x4 PS HRA with passive control and lock-up faults	118
5.48	Step response of 4x4 SP HRA with passive control and lock-up faults	119
5.49	Step response of 10x10 PS HRA with passive control and lock-up faults	119
5.50	Transient characteristics of passively controlled 4x4 PS HRA under fault	122
5.51	Transient characteristics of passively controlled 4x4 SP HRA under fault	122
5.52	Transient characteristics of passively controlled 10x10 PS HRA under fault	123
6.1	Rule-based fault detection in multi-agent control of HRA	126
6.2	Lock-up fault detection	127
6.3	Flow chart for lock-up fault detection	128
6.4	Loose fault detection	128
6.5	Flow chart for loose fault detection	129
6.6	Rule-based loose fault detection simulation	130
6.7	HRA input and output during loose fault profile.	131
6.8	Rule-based lock-up fault detection simulation	131
6.9	HRA input and output during lock-up fault profile.	132
6.10	IMM estimation	133
6.11	IMM FDI for 4×4 PS HRA	137
6.12	Mode probabilities produced by IMM loose FDI from measured load position output and known voltage input.	139
6.13	Estimated number of loose faults.	140
6.14	Estimated force capability.	140
6.15	Mode probabilities and estimated travel capability.	141
6.16	Measured load position output and known voltage input for HRA during lock- up faults.	142
6.17	Fault detection properties.	143
7.1	Agency structure for PS HRA.	146
7.2	Agency structure for SP HRA.	147
7.3	Agent architecture	149
7.4	Fault communication example	151
7.5	Control layer	152
7.6	Multiple-model and passive control design	153
7.7	Multiple-model and passive control performance	154
7.8	Singular values of nominal and loose MAC controlled 4×4 PS HRA.	155
7.9	Singular values of nominal and loose MAC controlled 4×4 SP HRA.	156
7.10	Singular values of nominal and loose MAC controlled 10×10 PS HRA.	156
7.11	Step response of nominal and loose MAC controlled 4×4 PS HRA.	157
7.12	Step response of nominal and loose MAC controlled 4×4 SP HRA.	157

7.13	Step response of nominal and loose MAC controlled 10×10 PS HRA.	158
7.14	Response of SP 4×4 HRA with two loose faults in two branches to pulse train input.	158
7.15	Response of SP 4×4 HRA with two loose faults in two branches to pulse train input with element de-activation.	160
7.16	Step response of nominal and loose MAC controlled 4×4 SP HRA with element de-activation.	161
7.17	Singular values of nominal and locked up MAC controlled 4×4 PS HRA. . .	162
7.18	Singular values of nominal and locked up MAC controlled 4×4 SP HRA. . .	162
7.19	Singular values of nominal and loose MAC controlled 10×10 PS HRA.	163
7.20	Step response of nominal and locked up MAC controlled 4×4 PS HRA. . . .	163
7.21	Step response of nominal and locked up MAC controlled 4×4 SP HRA. . . .	164
7.22	Step response of nominal and locked up MAC controlled 10×10 PS HRA. . .	164
7.23	Communication of location specific fault information.	166
7.24	Comparison of active and passive fault tolerant control performance with loose faults.	168
7.25	Comparison of active and passive fault tolerant control performance with lock-up faults.	169
7.26	Transient characteristics of MAC controlled 4×4 PS HRA under all fault combinations	170
7.27	Transient characteristics of MAC controlled 4×4 SP HRA under all fault combinations	171
7.28	Transient characteristics of MAC controlled 4×4 SP HRA under all fault combinations	171
7.29	Transient characteristics of MAC controlled 4×4 SP HRA under all fault combinations	172
7.30	Dynamic response of 4×4 PS MACHRA with 1 lock-up fault at $t=0$	174
7.31	Dynamic response of 4×4 PS MACHRA with multiple faults at $t=0$	175
7.32	Fault flags and control modes of agents in 4×4 PS HRA in response to multiple faults.	176
7.33	Response of 4×4 PS HRA with multiple faults and multiple or single control switches.	176
7.34	Response of 4×4 PS HRA with multiple faults and a reduced reconfiguration rate.	177
8.1	2×2 SP HRA electromechanical experiment.	179
8.2	Software Demonstrator navigation window.	182
8.3	Main simulation window for 10×10 PS system.	183
8.4	Input selection pop-up.	184
8.5	Control selection pop-up.	184
8.6	Input noise pop-up.	185
8.7	Fault injection controls.	186

8.8	Typical simulation screen.	187
8.9	HRA animated display.	187
8.10	Flow chart describing operation of s-functions within the software demonstrator that provide real-time visualisations.	189
8.11	Health monitor window.	190
8.12	Simulation traces.	191
8.13	Agent state display window.	192
C.1	4x4 PS HRA	231
C.2	4x4 SP HRA	233
D.1	Inner-loop multi-agent control.	236

List of Tables

3.1	Analogous electrical and magnetic quantities.	44
3.2	SMAC LAL30 moving coil actuator specifications.	50
3.3	Parameter values.	55
4.1	Typical application requirements	78
4.2	Actuation element specifications	78
4.3	Application example requirements	79
4.4	Control requirements for example systems	79
4.5	HRA capabilities and fault tolerance levels	80
5.1	Loose fault injection for 10×10 PS HRA	85
5.2	Lock-up fault injection for example systems	86
5.3	Stability margins of nominal example systems	87
5.4	Stability margins and additive error of 4×4 PS HRA under nominal and fault conditions	92
5.5	Stability margins and additive error of example HRAs under nominal and loose fault conditions	96
5.6	Stability margins and infinity norm of additive error for example HRAs under nominal and lock-up fault conditions	100
5.7	Stability Margins and transient characteristics of globally controlled example HRAs	103
5.8	Stability margins and transient characteristics of example HRAs under nominal and fault conditions	110
5.9	Stability margins and transient characteristics of example HRAs under nominal and loose fault conditions	115
5.10	Stability margins and transient characteristics of example HRAs under nominal and lock-up fault conditions	120
6.1	Fault detection and health monitoring requirements and resources	125
6.2	Loose fault simulation profile (refer to Figure C.1 for element numbers)	130
6.3	Lock-up fault simulation profile	131
6.4	Loose fault simulation profile	139
6.5	Lock-up fault simulation profile	141

7.1	Stability margins and transient characteristics of example MAC controlled HRAs under nominal and loose fault conditions	159
7.2	Stability margins and transient characteristics of example MAC controlled HRA under nominal and lock-up fault conditions	165
7.3	Transient characteristics and no. of systems within the requirements under 3-4 lock up faults	167
8.1	Software Demonstrator Requirements	181
8.2	Software demonstrator requirement status	193
B.1	Frequency Sweep Data for Free-Moving Actuator Before Offsetting (03/08/07).	228
B.2	Frequency Sweep Data for Clamped Actuator Before Offsetting (06/08/07).	229
B.3	Measurement Offsets.	229
D.1	Fixed PI control gains for MAC of HRA examples	236
D.2	Look-up table of control parameters for MAC of PS 4×4 HRA.	237
D.3	Look-up table of control parameters for MAC of SP 4×4 HRA.	238
D.4	Look-up table of control parameters for MAC of PS 10×10 HRA.	239

Nomenclature

Abbreviations

ACL Agent Communication Language

AFTC Active Fault Tolerant Control

AMS Agent Management Service

BDI Belief Desire Intention

CCL Constraint Choice Language

DARPA Defense Advanced Research Projects Agency

DF Directory Facilitator

emf Electromotive Force

FDI Fault Detection and Isolation

FIPA Foundation for Intelligent Physical Agents

FIPA SL FIPA Semantic Language

FMECA Failure Modes, Effects and Criticality Analysis

FTA Fault Tree Analysis

FTC Fault Tolerant Control

GM Gain Margin

GUI Guided User Interface

HRA High Redundancy Actuation/Actuator

IMM Interacting Multiple-Model

I/O Input Output

IP Internet Protocol

KIF Knowledge Interchange Format

KQML	Knowledge Query and Manipulation Language
LTI	Linear Time-Invariant
MACHRA	Multi-Agent Controlled High Redundancy Actuator
MAC	Multi-Agent Control
MAS	Multi-Agent Systems
mmf	magnetomotive force
OS	Overshoot
PI	Proportional Integral
PRS	Procedural Reasoning System
PS	Parallel in Series
RAID	Redundant Array of Inexpensive Disks
RDF	Resource Definition Framework
RT	Rise Time
SP	Series in Parallel
ST	Settling Time
TCP	Transmission Control Protocol
PM	Phase Margin

Symbols

B	Magnetic flux density, T
d	Damping constant, Ns/m
d_{lim}	Damping coefficient at limit of travel, Ns/m
e	Position error, m
E	Electromotive force, V
$e_{threshold}$	Position error threshold for fault detection, m
\mathcal{F}	Magnetomotive force, At
F	Force, N
I	Current, A

I_c	Coil current, A
I_{LB}	Bobbin inductance current, A
I_{Lm}	Mutual inductance current, A
I_{R1}	Winding resistance current, A
k	Force constant, N/A (Vs/m)
l	Turn length, m
L_1	Coil inductance, H
$\Lambda_{j(t)}$	Likelihood of mode j being the active state at time t
L_B	Bobbin inductance, H
L_{lock}	Fault detection law that diagnoses a lock-up fault
L_{loose}	Fault detection law that diagnoses a loose fault
L_m	Mutual inductance, H
m	Mass, kg
N	Number of turns
P	Error covariance matrix, a measure of the estimated accuracy of the state estimate
Φ_1	Coil flux, Wb
Φ_2	Bobbin flux, Wb
Φ_3	Core losses, Wb
Φ_b	Coil-bobbin flux, Wb
Φ_M	Mutual flux, Wb
p_{ij}	Mode transition matrix, a $m \times m$ matrix where element ij denotes the probability of transition from mode i to mode j in the next time frame
Q	Plant input noise covariance
\Re	Reluctance, At/Wb
R	Measurement noise covariance
r	Spring constant, N/m
R_1	Winding resistance, Ω
R_2	Bobbin resistance, Ω

R_{23}	Combined parallel resistance of R_2 and R_3
R_3	Resistive core losses, Ω
ρ_{ij}	Mixed probability
r_{lim}	Spring coefficient at limit of travel, N/m
u	Input voltage, V
v	Measurement noise
w	plant noise, V
x	Position, m
\hat{x}	Estimated value of x , m
x_{lim}	Travel limit for actuation elements, m
$\dot{x}_{threshold}$	Velocity threshold for fault detection

Fault Tolerance Terminology

Active fault tolerant control - A control strategy that allows the system to remain operational in the presence of faults through reconfiguration of control in response to detected faults. This reconfiguration may be achieved through online synthesis or online selection.

Adaptive control - Online modification of controller parameters, in response to variations in system parameters, in order to achieve control objectives.

Analytical redundancy - The generation of residuals using two or more methods to determine a variable, where at least one method uses a mathematical process.

Availability - Probability that a system will be fully operational at any point in time.

Error - The deviation between a measure or analytically derived value of an output variable and its true specified or theoretical value.

Disturbance - An unknown and uncontrolled input acting on a system.

Fail-safe - The ability of a system to fail to a state that is considered safe in the context of the application.

Failure - A permanent interruption of a system's ability to perform a required function.

Fault - The deviation of at least one of the characteristic properties or parameters of the system from the standard conditions.

Fault detection - The determination of faults present in a system and the time of detection.

- Fault diagnosis** - Determination of the type, magnitude, location and time of detection of a fault. Involves fault identification and isolation.
- Fault identification** - Determination of the magnitude and time-variant behaviour of a fault. Follows fault isolation.
- Fault isolation** - Determination of the fault type, location and time following the detection of a fault.
- Fault tolerance** - The ability of a controlled system to achieve optimal, or acceptably degraded levels of performance, despite the occurrence of a fault.
- FDI** - Fault Detection and Isolation.
- Monitoring** - A continuous real-time task of determining the conditions of a physical system, by recording information, recognising and indicating anomalies in the behaviour.
- Online selection** - Reconfiguration of control in response to faults by selecting between a set of pre-designed control algorithms.
- Online synthesis** - Reconfiguration of control in response to faults through online adaptation of the control law.
- Partial actuator fault** - A sub-class of actuator faults where the force or travel capability of the actuator is reduced, but not completely lost.
- Passive fault tolerant control** - A control strategy that allows the system to remain operational in the presence of faults through robust control design i.e. a single controller is designed to accommodate all fault conditions.
- Physical redundancy** - The generation of residuals through the comparison of two or more sets of measured data determined through hardware replication.
- Reconfiguration** - Modifications made to the controller in response to faults.
- Reliability** - Ability of a system to perform a required function, under stated conditions during a given period of time.
- Residual** - A fault indicator based on the deviation either between two or more sets of measurements or a measurement and analytical computations.
- Robust control** - The control of unknown plants with unknown dynamics subject to unknown disturbances.
- Supervision** - Monitoring a physical system and taking appropriate actions to maintain operation in the case of faults.
- Symptom** - A deviation from nominal behaviour of an observable quantity, that is used for fault diagnosis.
- Uncertainties** - Unknown dynamics or effects acting on a system.

Contents

1	Introduction	1
1.1	Problem Statement	1
1.2	High Redundancy Actuation	2
1.3	Thesis Motivation	3
1.4	Thesis Objectives	6
1.5	Thesis Overview	7
1.5.1	Background & Literature Review	7
1.5.2	Modelling of High Redundancy Actuation	7
1.5.3	Application Example	7
1.5.4	Passive Fault Tolerant Control of High Redundancy Actuation	7
1.5.5	Fault Detection and Health Monitoring	8
1.5.6	Active Fault Tolerant Control of High Redundancy Actuation	8
1.5.7	Software Demonstrator	8
1.5.8	Conclusions	8
1.6	Contributions	8
1.6.1	Publications	9
2	Background & Literature Review	11
2.1	Terminology	11
2.2	Faults, Failures and Fault Tolerance	11
2.3	Fault Tolerant System Design	12
2.3.1	Fault Tolerant Actuation	14
2.3.2	Traditional Actuation Redundancy	15
2.3.3	High Redundancy Actuation	16
2.4	Fault Tolerant Control	17
2.4.1	Passive Fault Tolerant Control	18
2.4.2	Active Fault Tolerant Control	19
2.4.2.1	Fault Detection and Isolation Methods	19
2.4.2.2	Online Selection & Online Synthesis Control Algorithms	22
2.5	Multi-Agent Systems	23
2.5.1	The Agent Concept	23
2.5.2	Multi-Agent Systems	24
2.5.3	Typical Applications	25

2.5.4	Central Concepts in Multi-Agent Systems	26
2.5.4.1	Agent Architectures	26
2.5.4.2	Agency Structures	30
2.5.5	Multi-Agent Systems & Fault Tolerant Control	38
2.6	Conclusions	39
3	Modelling of High Redundancy Actuation	41
3.1	Modelling of a Closed-Bobbin Moving Coil Actuator	42
3.1.1	Operating Principles of a Moving Coil Actuator	42
3.1.2	Modelling Methodology	43
3.1.3	Electrical Subsystem Modelling	44
3.1.4	Mechanical Subsystem Modelling	47
3.1.5	Full Model	48
3.1.6	Parameter Identification	50
3.1.6.1	Experiment Setup	50
3.1.6.2	Frequency Response	51
3.1.6.3	Parameter Fitting	53
3.1.7	Model Reduction	57
3.1.7.1	Third Order Element Model	58
3.1.7.2	Second Order Element Model	59
3.1.7.3	Model Reduction Quality	59
3.2	Modelling of a Cut-Bobbin Moving Coil Actuator	62
3.2.1	Model Reduction	62
3.3	Non-Linearities	65
3.3.1	Travel Limits	65
3.3.2	Input Saturation	65
3.4	Modelling of HRA Configurations	66
3.5	Fault Modelling	71
3.5.1	Parameter Deviations	71
3.5.2	Mechanical Loose Faults	71
3.5.3	Mechanical Lock-up Faults	72
3.6	Conclusions	76
4	Application Example	77
4.1	Introduction	77
4.2	Typical Application Requirements	77
4.3	Example HRAs Requirements	78
4.4	Capability & Fault Tolerance	79
4.5	Conclusions	81

5	Passive Fault Tolerant Control of High Redundancy Actuation	82
5.1	Introduction	82
5.1.1	Fault Injection Methodology	84
5.1.1.1	Parameter Uncertainties	84
5.1.1.2	Loose Faults	84
5.1.1.3	Lock-up Faults	85
5.1.2	Evaluation Methodology	86
5.2	Open-Loop System Analysis	86
5.2.1	Nominal system	87
5.2.2	Parameter Deviations	87
5.2.3	Loose Faults	89
5.2.4	Lock-up Faults	97
5.2.5	System Analysis Summary	101
5.3	Control Design	101
5.3.1	4x4 PS HRA	101
5.3.2	4x4 SP HRA	102
5.3.3	10x10 PS HRA	104
5.3.4	Control Design Summary	106
5.4	Fault Simulation	106
5.4.1	Parameter Deviations	106
5.4.2	Loose Faults	111
5.4.3	Lock-up Faults	116
5.4.4	Fault Simulation Summary	121
5.5	Conclusions	123
6	Fault Detection & Health Monitoring	124
6.1	Introduction	124
6.2	Rule-Based Fault Detection for AFTC	125
6.2.1	Lock-up Fault Detection	126
6.2.2	Loose Fault Detection	127
6.2.3	Fault Simulations	127
6.2.3.1	Loose Faults	130
6.2.3.2	Lock-up Faults	132
6.2.3.3	Summary	132
6.3	Interacting Multiple-Model Fault Detection for Health Monitoring	132
6.3.1	IMM Estimation Algorithm	133
6.3.2	IMM Mode Allocation	135
6.3.3	Fault Simulation	138
6.3.3.1	Loose Faults	138
6.3.3.2	Lock-up Faults	140
6.3.3.3	Summary	142
6.4	Conclusions	142

7	Active Fault Tolerant Control of High Redundancy Actuation	144
7.1	Introduction	144
7.2	Rationale for Multi-Agent Control of HRA	144
7.3	Design of Multi-Agent Control of HRA	145
7.3.1	MACHRA Agency Structure	146
7.3.2	MACHRA Agent Architecture	148
7.3.2.1	Fault Detection Layer	148
7.3.2.2	Communication Layer	149
7.3.2.3	Control Layer	150
7.3.3	Control Design	152
7.3.3.1	Nominal Control	152
7.3.3.2	Fault Control	153
7.4	Fault Simulations	154
7.4.1	Static Fault State Simulations	154
7.4.1.1	Loose faults	155
7.4.1.2	Lock-up faults	161
7.4.1.3	Comparison with Passive Fault Tolerant Control Performance	167
7.4.1.4	Static Performance Summary	167
7.4.2	Dynamic Fault Injection Simulations	170
7.4.2.1	Single Fault Injection	173
7.4.2.2	Multiple Fault Injection	173
7.5	Conclusions	177
8	Software Demonstrator	179
8.1	Introduction	179
8.2	Requirements	180
8.2.1	Functional Requirements	180
8.2.2	Visualisation Requirements	180
8.2.3	Usability, Portability and Extendability Requirements	181
8.3	Implementation	181
8.3.1	Navigation Window	182
8.3.2	Simulation Window	182
8.3.2.1	Configuration	182
8.3.2.2	Simulation Visualisation	185
8.4	Conclusions	192
9	Conclusions & Future Work	195
9.1	Conclusions	195
9.2	Suggestions for Further Work	197
A	Piezo Technology Report	212
B	Modelling	217

C Example Models	231
D Multi-Agent Control	236
E Software Demonstrator	240
F Publications	250

Chapter 1

Introduction

1.1 Problem Statement

Over the past 40 years, automatic control systems have become widespread throughout the aerospace and automotive industries, manufacturing, and critical infrastructures, and the complexity of these systems is increasing. Control system theory is well established, and able to ensure a stable, pre-defined performance, given that the components within these systems are operating correctly. Unfortunately, all components, regardless of the reliability of their design or frequency of maintenance, are subject to faults. As automated systems are closely coupled, faults in individual components are likely to affect the operation of the system at large. In safety-critical systems, faults may result in damage to the system, its environment, or people within its vicinity. Faults within non-safety critical systems can have profound economic impacts, increasing down-time and life-cycle costs. Hence, the development of systems that are able to tolerate faults is of great significance for many applications, and consequently, fault tolerant system design has developed into a major area of research in the past 30 years.

Actuators and sensors are key to the operation of automated systems, as they provide the means of controlling and observing the system. Consequently, a common method of achieving the required reliability of these instruments has been to employ some redundancy (Figure 1.1). Typically, 3 or 4 sensors/actuators are used in parallel, and their output is combined through

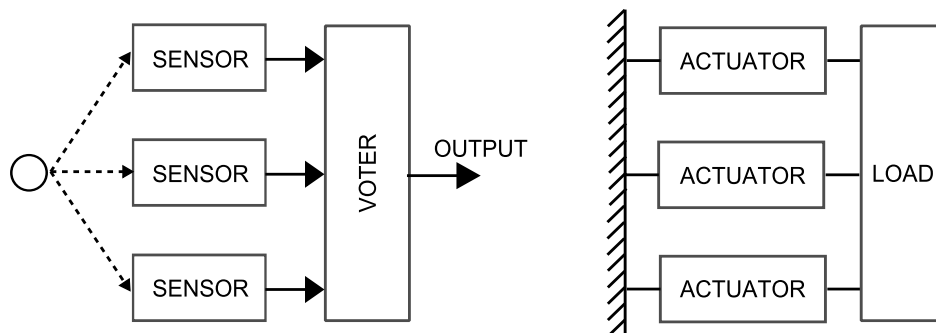


Figure 1.1: Conventional sensor and actuator redundancy.

voting (in the case of sensors) or mechanical consolidation (in the case of actuators). If faults occur within these instruments, then one alone could provide the necessary function.

However, this over-engineering incurs penalties as cost and weight are increased, and subsequently, the efficiency reduced. Parallel actuator redundancy is also problematic, as jamming (lock-up) faults in actuators effectively lock the whole actuation assembly in place, rendering the redundancy useless. Research has produced many methods of mitigating the need for redundant sensors through analytical replication. However, these strategies are not applicable to actuators, as actuation force is an unavoidable necessity to keep the system in control and bring it to the desired state (1). Hence, research in this area has mainly concentrated on compensating a small sub-class of actuator faults where the capability of the actuator is only reduced¹, as opposed to where the capability is completely lost as is the case in loose or lock-up faults.

1.2 High Redundancy Actuation

The high redundancy actuator concept is a novel approach to fault tolerant actuation. Figure 1.2 provides a representation of a high redundancy actuator, which comprises many small actuator elements arranged in both series and parallel. Each actuation element provides only a small contribution to the required force and travel of the actuator. As the capability of each element is small, the effect of faults within individual elements on the overall actuator is also small, and as such, faults in elements can be intrinsically accommodated. This concept is inspired by biological muscles, which are composed of many individual cells, each of which providing only a minute contribution to the force and the travel of the muscle. These properties allow the muscle, as a whole, to be highly resilient to substantial levels of cell damage.

High redundancy actuation alleviates the problems incurred by conventional redundancy schemes, as the extent of over-engineering is reduced. For example, a high redundancy actuator may employ 100 actuation elements, but the specified operation may only require 80.

¹These types of faults are termed 'partial' faults within the literature, as they only partially affect the actuator.

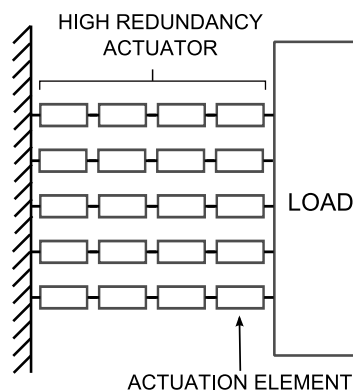


Figure 1.2: High redundancy actuation.

Thus, the high redundancy actuator in this case is only 20% over-engineered in comparison to a triplex or quadruplex scheme which are 200% and 300% over-engineered respectively.

Issues with lock-up faults are resolved as they are inherently tolerated by the inclusion of serial actuation. The manner in which the high redundancy actuator degrades is also of advantage. Instead of large changes of capability, like that witnessed in parallel redundancy systems where the system may change from fully operational to total failure within a short period, a high redundancy actuator will decrease in capability more gradually i.e. it will gracefully degrade.

1.3 Thesis Motivation

The high redundancy actuation concept poses a number of research questions:

- In what configuration should actuation elements be arranged, and how can the reliability of these configurations be quantified?
- How should high redundancy actuation be controlled in order to provide fault tolerance?
- How can health monitoring for the system be provided for maintenance purposes?
- Which actuation technologies are most suitable for forming high redundancy actuators?

Several of these questions have been addressed to various extents. Work on the high redundancy actuator project thus far has involved several investigators. The first phase of the project was conducted by Xinli Du, a PhD student, and investigated the feasibility of the high redundancy actuation concept. Based upon these studies, a second project phase has been funded through a EPSRC (Engineering and Physical Sciences Research Council) grant. This second phase has been concerned with progressing the control and reliability studies associated with the high redundancy actuator, and has been mainly conducted by the author and Thomas Steffen, a research associate. Hence, the work contained within this thesis forms a contribution to the project at large, and does not aim to address all the research questions posed by the concept. A discussion of current progress in addressing each research challenge follows.

High Redundancy Actuation Configuration and Reliability

There are many ways in which actuation elements can be arranged to form a high redundancy actuator. The way in which elements are configured affects the tolerance of the system to particular faults. There are two major fault modes that are common to many types of actuators:

1. lock-up faults, where the actuator jams in place, and,
2. loose faults, where the actuator loses the ability to exert a force between its end-points.

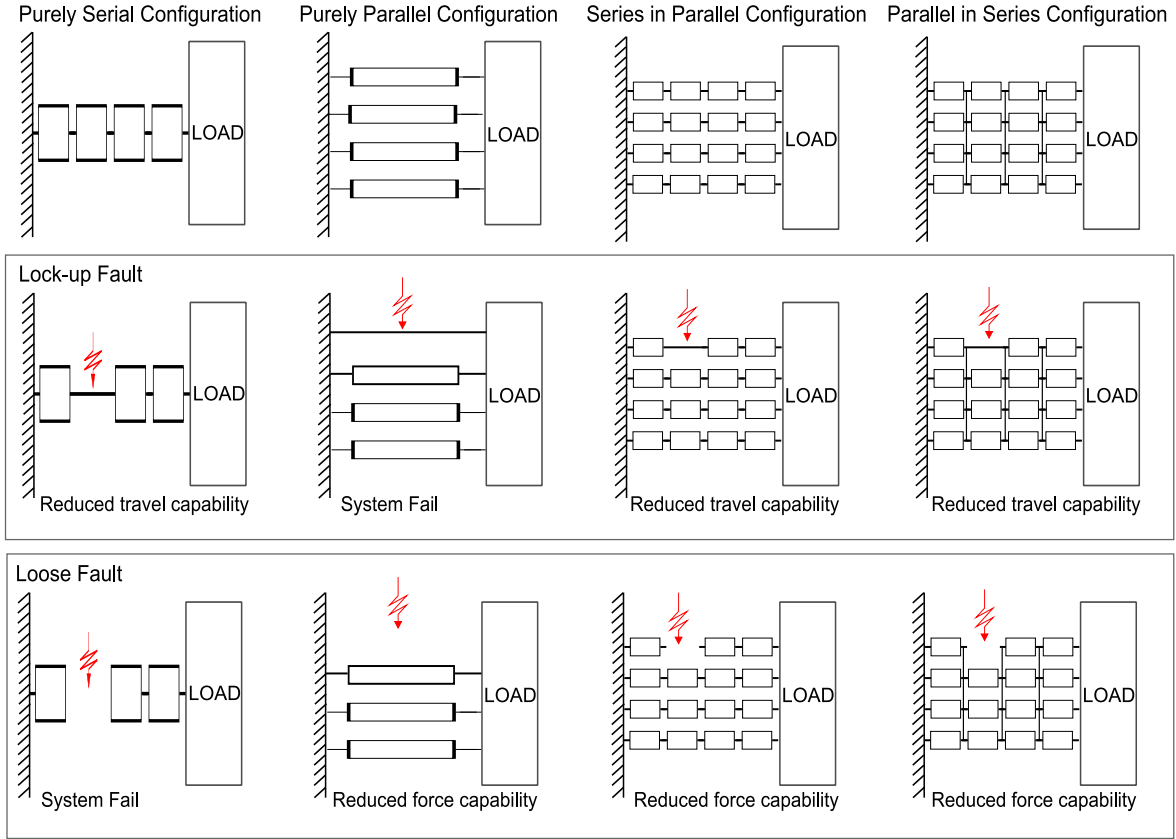


Figure 1.3: High redundancy actuator configurations under fault.

These two actuator fault types are the focus of the studies presented within this thesis.

Elements arranged in series provide tolerance to lock-up faults, as serial actuators are available to provide travel in the case of an element fault. However, purely serial elements are intolerant to loose faults, as the force capability of the whole system will be lost if one element becomes loose. Conversely, parallel elements offer tolerance to loose faults, but are susceptible to lock-up faults. This idea is summarised in Figure 1.3.

Hence, combinations of parallel and serial elements will provide tolerance to each fault type. Two major configuration types emerge when this problem is considered: parallel elements connected in series (parallel-in-series) and serial elements connected in parallel (series-in-parallel). These configurations are illustrated in Figure 1.3. Series-in-parallel configurations provide a greater tolerance to lock-up faults, and a degree of tolerance to loose faults. Parallel-in-series configurations provide a greater tolerance to loose faults, and a degree of tolerance to lock-up faults. The choice of configuration, and the number of elements therein, will be dictated by the likelihood of each fault mode in the actuator technology and the requirements of the application, hence, no definitive configuration can be made.

A method of analysing the reliability of these configurations has been addressed in (2). This work also proves that high redundancy actuation can provide higher levels of reliability in comparison to traditional parallel redundancy. These forms of analysis are not considered within this thesis.

Previous studies have concentrated on a 2×2 series in parallel high redundancy actuator (containing 4 elements) (3; 4; 5; 6; 7). This is a relatively low number of elements for a high redundancy actuator: it is envisaged that a high redundancy actuator suitable for industrial use may contain hundreds of elements. The number of elements will affect the operational characteristics of high redundancy actuators. Hence, this thesis extends the previous work and considers two levels of element redundancy and two configurations. Two 4×4 parallel-in-series and series-in-parallel systems (containing 16 elements) are analysed alongside a 10×10 parallel in series high redundancy actuator (containing 100 elements). These systems will provide an indication of the characteristics of high redundancy actuation at two levels of redundancy and in the two main configuration types.

Control of High Redundancy Actuators

Control research thus far has concentrated on passive fault tolerant control techniques where a single robust controller is designed which provides a suitable performance under all fault conditions. Extensive theoretical research and practical implementation of passive fault tolerant control has been conducted using a 2×2 electro-mechanical configuration (3; 4; 5; 6; 7). This work has proved the viability of the high redundancy actuation concept and shown that a passive approach can provide fault tolerance within this structure.

Passive fault tolerant control does not require faults in the system to be detected, or the control law to be altered. This simplicity and constancy make passive fault tolerant control an attractive solution for the high redundancy actuator, as its operation is easier to verify for high integrity applications. However, a passive fault tolerant controller must encompass a large system operation region, and as such, the resultant performance can be conservative. This problem is negated to some extent within the high redundancy actuator, as the change in system behaviour is reduced by having many low capability elements.

Active fault tolerant control, where the control is changed in response to faults, offers the possibility of improved performance of the system in nominal or fault conditions, and increased levels of fault tolerance. Active fault tolerant control of high redundancy actuation has not been explored previously. One aim of this thesis is to investigate an active fault tolerant control approach for the high redundancy actuator and assess any performance improvements that may be gained, whilst also considering the costs in terms of the associated complexity and uncertainty.

Health Monitoring of High Redundancy Actuation

From an operational perspective, some form of health monitoring of the high redundancy actuator is a necessity. It is envisaged that the high redundancy actuator will continue to operate within an acceptable performance region under element fault conditions, up until a point where the capability (be it travel or force) falls below that required by the application. At this point, or just before it, maintenance will be required to replace the high redundancy actuation unit. Hence, health monitoring is needed to provide an indication of the high redundancy actuator's capability for maintenance purposes. Additionally, this health information

could be used to make operational decisions.

Thorough research into providing health monitoring for the high redundancy actuator has not yet been completed. However, one possible method of providing health information is examined within this thesis.

High Redundancy Actuator Technology

The identification of technologies that are suitable for use as actuation elements within high redundancy actuators has not been seriously addressed to date. This is because research is at a proof of concept stage. Technology used within concept demonstrators so far has aimed to illustrate the increased reliability and control properties afforded by the topology of a high redundancy actuator, rather than form a product suitable for industrial use². A 2×2 demonstrator was previously developed using electromechanical actuators and a 4×4 high redundancy actuator that uses electromagnetic actuators is in the final stages of development. Hence, the work within this thesis considers the use of electromagnetic actuation and is applicable to this latest demonstrator.

This 4×4 system is still some way short of the (element) levels of redundancy intended for use within a high redundancy actuator. Thus, another objective of the work within this thesis is to provide a software demonstrator that will visually illustrate the operation of high redundancy actuators with greater numbers of elements.

1.4 Thesis Objectives

Given the discussions of the previous section, the main objectives of this thesis may be summarised as follows:

1. to investigate the use of moving coil actuators as elements in the high redundancy actuation scheme, including the modelling of an element and assemblies, model verification and modelling of lock-up and loose actuation element faults.
2. to investigate active fault tolerant control strategies based upon multi-agent system concepts for use within the HRA, and compare its performance to a passive fault tolerant controlled system under lock-up and loose actuation element fault conditions.
3. to explore two fault detection and health monitoring methods for the high redundancy actuator.
4. to develop a software demonstrator which illustrates the operation of high redundancy actuation systems that comprise many elements.

²Progress on the next phase of the project concerning technology identification is in the initial stages, and a report on the suitability of piezoelectric actuators for use within a high redundancy actuator has been written by the author. The interested reader is directed towards Appendix A, where a copy of this report is included.

1.5 Thesis Overview

The objectives stated in 1.4 are achieved through the following methodology.

1.5.1 Background & Literature Review

Firstly, background information and a survey of relevant literature is provided, which forms a foundation for the work that follows. The design of fault tolerant systems is addressed and methods of fault tolerant control and fault diagnosis are discussed. An introduction to multi-agent systems is also given, as these concepts form the basis of the active fault tolerant control framework presented later in the thesis.

1.5.2 Modelling of High Redundancy Actuation

The main contributions of the thesis begin in Chapter 3 with the modelling of electro-magnetic high redundancy actuation, as producing a representation of the system upon which the control and fault detection methods can be designed is the first logical step. Modelling of a moving coil actuator based on first principles is presented, which is subsequently verified experimentally. Two types of actuator are modelled. Firstly, a moving coil actuator with a closed bobbin, as this was the first actuator used within the project. This arrangement is atypical, leading to an unconventional model. The second actuator has a cut bobbin, which is more archetypal and as such the model produced is fairly standard. Full-order non-linear and linear reduced-order versions of the model are presented for both actuator types, which are to be used for simulation and control design purposes respectively.

A methodology for assembling these element models into high redundancy actuation structures is then described, and a procedure for representing faults in high redundancy configurations given.

This chapter addresses the first objective as described in Section 1.4.

1.5.3 Application Example

In order to illustrate the various characteristics of high redundancy configurations, three example systems are defined for use in the control studies in Chapter 4. Two 4×4 systems are created, one in a series-in-parallel, and the other in a parallel-in-series configuration. This will allow a comparison between the two main configurations. The third system is a 10×10 parallel-in-series system, which illustrates the consequences of increasing the number of elements in the high redundancy actuator. These example systems are based upon real application requirements to add weight to the control studies that follow. Performance criteria are also defined for use within the control studies.

1.5.4 Passive Fault Tolerant Control of High Redundancy Actuation

Having established simulation and control design models, and defined a number of example systems, the effects of faults on these systems are analysed and subsequently, passive fault

tolerant control is designed using classical methods in Chapter 5. The performance of the system is then considered under nominal and fault conditions. This control performance acts as a benchmark, against which the active fault tolerant control results can be compared.

1.5.5 Fault Detection and Health Monitoring

Chapter 6 addresses objective 3 of section 1.4. Fault detection is required by the active fault tolerant control design, hence it is described within this chapter. A decentralised rule-based approach is proposed as this complements the localised nature of the active fault tolerant control strategy presented. Health monitoring using interacting multiple model fault detection is also described, which illustrates an approach to estimating the health state of a more centralised scheme for maintenance purposes. Both fault detection approaches are simulated and their results are compared and discussed.

1.5.6 Active Fault Tolerant Control of High Redundancy Actuation

The fault detection described in the previous chapter is utilised within the active fault tolerant control strategy described in Chapter 7. This control strategy is based upon multi-agent concepts. A motivation for using multi-agent system ideas in the design of active fault tolerant control for the high redundancy actuator is established. The design of the multi-agent control approach is then described. Simulation results are presented and the performance of this scheme is compared to the benchmark passive fault tolerant control performance. This chapter, in combination with Chapter 5, satisfies objective 2.

1.5.7 Software Demonstrator

The development and functionality of a software demonstrator that illustrates the operation of high redundancy actuation configurations containing large numbers of elements is described within Chapter 8. This work addresses the final aim of this thesis: objective 4.

1.5.8 Conclusions

Finally, conclusions are made in Chapter 9. Further extensions to the work are then suggested.

1.6 Contributions

As the high redundancy actuator is a new approach to fault tolerant actuation, there is much originality in the research that is associated with it. The main contributions made within this thesis are as follows:

Modelling Contributions As stated earlier, the closed-bobbin moving coil actuator described in Chapter 3 is a non-standard arrangement. This closed-bobbin actuator has been modelled previously in (8). However, the model presented there approximates the electrical characteristics with a third order transfer function fitted to experimental data. The model

derived here is done so from first principles, maintaining the physical relevance of the parameters. This approach has not been found within the literature. Hence, this model and its subsequent reductions and extensions form an original contribution to knowledge.

The fault modelling presented in Chapter 3 also forms some contribution to knowledge. The use of actuators in series is unusual and thus faults in serial actuators are not commonly covered in the literature, whereas this configuration is a particular feature of the HRA concept. The fault types included here in serial actuation have been previously covered in the work of (7). However, these faults were introduced into an electromechanical system, and thus consideration of these fault types within electromagnetic actuation forms a new contribution to the project.

Control Contributions The active fault tolerant control presented in Chapter 7 forms the main contribution of this thesis. Prior to these studies, only passive fault tolerant control techniques have been investigated for use with the high redundancy actuator.

In addition to the application of active fault tolerant control to the high redundancy actuator problem, the nature of the active fault tolerant control presented also offers a degree of novelty. In the studies made in Chapter 2, no previous work was found where control and management of redundant actuators for fault tolerance purposes was achieved through the application of multi-agent concepts, and as such Chapter 7 may be considered a novel approach to this problem.

The passive control described in Chapter 5 forms a smaller contribution to knowledge. Whilst classical passive control of electro-mechanical actuators has been investigated previously, simple passive control of electro-magnetic HRAs has not been widely addressed, particularly control of series-in-parallel structures, and as such, these control studies provide a contribution.

Fault Detection and Health Monitoring Contributions Fault detection of lock-up and loose faults in the high redundancy actuator has not been addressed previously, as much emphasis has been put on passive fault tolerant control research, where fault detection is not critical. Health monitoring for high redundancy actuators has also not been addressed previously within the project. Thus whilst the methods used to achieve fault detection and health monitoring in Chapter 6 are not original, their application to this problem is novel.

Software Demonstrator The final contribution of this thesis is the development of a software demonstrator to illustrate the operation of a larger range of high redundancy actuator configurations and sizes than is available through experimental demonstration.

1.6.1 Publications

A number of papers have been published (or are currently under review) in connection to the work outlined in this thesis (copies of the papers where the author is stated as first author are available within Appendix F):

1. J. Davies, T. Steffen, R. Dixon, R.M. Goodall, 'Multi-Agent Control of High Redundancy Actuation', Submitted to IMechE Journal of Systems and Control Engineering, 2009.
2. J. Davies, T. Steffen, R. Dixon, R.M. Goodall, 'Active versus Passive Fault Tolerant Control of High Redundancy Actuation', In Proceedings of European Control Conference, Budapest, 2009.
3. J. Davies, H. Tsunashima, T. Steffen, R. Dixon, R.M. Goodall, 'Fault Detection in High Redundancy Actuation using a Interacting Multiple Model Approach', In Proceedings of Safeprocess '09, Barcelona, 2009.
4. R. Dixon, T. Steffen, J. Davies, R.M. Goodall, 'HRA - Intrinsically fault tolerant actuation through high redundancy', In Proceedings of Safeprocess '09, Barcelona, 2009.
5. J. Davies, T. Steffen, R. Dixon, R.M. Goodall, 'Multi-Agent Control of a 10x10 High Redundancy Actuator', In Proceedings of IAR-ACD, Coventry, 2008.
6. J. Davies, T. Steffen, R. Dixon, R.M. Goodall, 'Multi-Agent Control of High Redundancy Actuation', In Proceedings of UKACC, Manchester, 2008.
7. J. Davies, T. Steffen, R. Dixon, R.M. Goodall, J. Pearson, A. Zolotas, 'Modelling of high redundancy actuation utilising multiple moving coil actuators', In Proceedings of IFAC World Congress, Seoul, 2008.
8. T. Steffen, R. Dixon, R.M. Goodall, J. Davies, 'Failure modes and probabilities of a high redundancy moving coil actuator', In Proceedings of IFAC World Congress, Seoul, 2008.
9. T. Steffen, J. Davies, R. Dixon, R.M. Goodall, 'Using a series of moving coils as a high redundancy actuator', In Proceedings of Aim, Zurich, 2007.

Chapter 2

Background & Literature Review

This literature review provides a general outline of design of high integrity systems, with special emphasis on redundancy, and a brief overview of approaches to fault tolerant control. A summary of multi-agent concepts is also made, as these are used in the development of an active fault tolerant control framework in Chapter 7.

2.1 Terminology

Unfortunately, much of the terminology used within the fault-tolerant control field is used with inconsistent meaning. This incoherence makes it difficult to understand the objectives of contributions and to compare the different approaches to fault tolerance. This problem has been acknowledged frequently within the fault tolerant community and is discussed in (9; 10; 11; 12). The terminology used throughout this literature review is consistent with the terminology used within (9) and the IFAC SAFEPROCESS terminology as found in (10). Definitions of the most important terms are provided in the glossary.

2.2 Faults, Failures and Fault Tolerance

A fault is a defect that occurs in the hardware or software of a system, which may be located in the controller, power supply, actuators or sensors of the system, or indeed in the plant itself. Faults often result in unexpected or undesirable behaviour changes of the system, and where faults result in the system being unable to complete an expected action, the system is said to have failed.

This concept is expressed in Figure 2.1. The nominal system behaviour, b_n lies within a region of acceptable system behaviour where the system is operational. Inevitably, a bound of uncertainty for the system surrounds this point, representing parameter uncertainties that may exist between actuator units, and system input uncertainties. Faults will change the system behaviour, and thus the position of the faulty system behaviours within this diagram will differ to that of the nominal. A range of fault system behaviours (represented by $\{b_f\}$) may lie within the acceptable region, and as such the system is considered tolerant to these particular faults. However, if a fault behaviour lies outside this region, then the system will

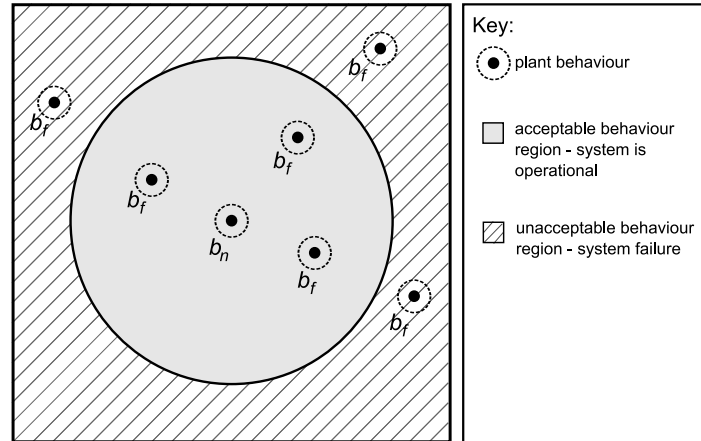


Figure 2.1: Behavioural representation of nominal and faulty systems.

be unable to complete expected actions and the fault has led to system failure.

The consequences of failures in engineering systems can include loss of revenue, damage to the plant and its environment, and in extremis, loss of life. A fault tolerant system is able to avoid failures and achieve adequate system performance in the presence of faults. Fault tolerance is important in safety-critical systems such as aeroplanes, trains and road vehicles and this importance is increasing as these industries move from mechanical to electronic solutions in X-by-wire strategies, and safety cannot be ensured through mechanical component integrity alone. However, fault tolerance can also be of importance for non safety-critical systems, such as machine tools and production robots, where increased reliability can increase operation times and reduce maintenance and life-cycle costs.

2.3 Fault Tolerant System Design

The most effective means of achieving fault tolerance is through systematic analysis and integrated design. An understanding of the system's structure, the reliability of its components and its current redundancies should be developed and analysed to determine vulnerable areas, and how fault tolerance could be provided. A representation of this process used by NASA in programmes such as Voyager, and Cassini as a preferred code of practice is given in Figure 2.2 (13). Whilst this flow chart is not exhaustive in the types of analyses that may be conducted, it captures the essence of the design approach.

The process usually starts with highly domain specific analysis of the system's components to identify possible fault modes. A Failure Modes, Effects and Criticality Analysis (FMECA) is then conducted, where the effects of each possible failure mode is evaluated at the local and intermediate system level, and the severity of their effect on overall system operation is assessed.

A Fault Tree Analysis (FTA) may also be completed, which complements the FMECA by starting with a top level failure effect and tracing this to potential faults that may induce that failure. FTA is aimed at analysing how multiple low-level faults can combine to cause system-level failures.

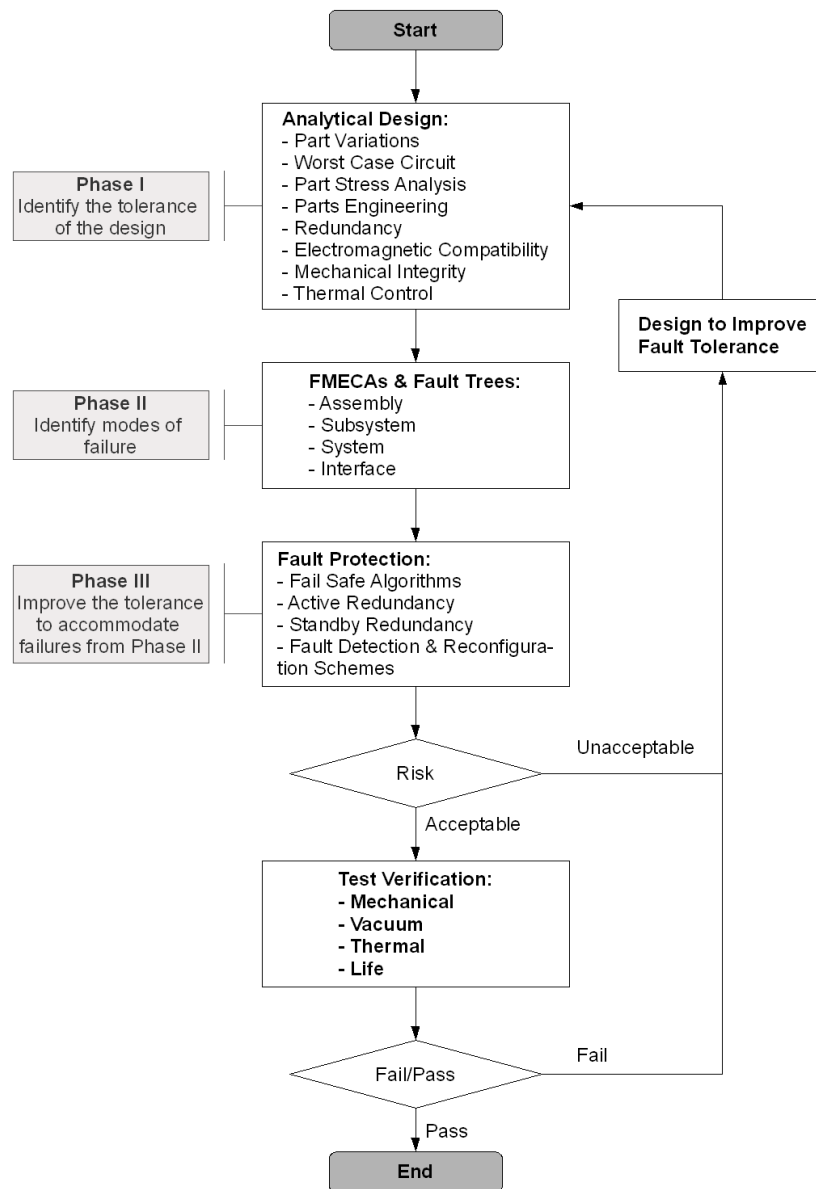


Figure 2.2: Process flow diagram for fault tolerant design(13).

The information from the FMECA and FTA can then be used to identify critical faults and decide whether to take measures that reduce the probability of failure (i.e. through strengthening components), prevent the propagation of failures (i.e. include fail-safe measures), or compensate for the effects of failures (i.e. through control reconfiguration) (14). This process is iterative, and ceases when either the cost of the next design iteration (in terms of added weight, volume, time or money constraints) is unacceptable or when the risk is reduced below a stated goal.

2.3.1 Fault Tolerant Actuation

Whilst faults can occur in many areas of the system, faults within the actuation components of a system are the impetus of this work. Actuation fault tolerance is central to the integrity of safety-critical systems as they are the effectors of control within a system.

As previously noted, there are several ways in which the reliability of the system can be increased including:

- Increase the reliability of individual components,
- Introduce redundancy,
- Add fail-safe mechanisms,
- Incorporate fault tolerant control strategies.

There are occasions where some of these options are not feasible, particularly for actuators. Development and testing may prove that critical components cannot provide the required reliability, and fail-safe mechanisms are not a good solution for systems that cannot afford to cease operating (such as primary control surfaces in naturally unstable aircraft). Control reconfiguration as a sole strategy for accommodating actuator faults can also be unsuitable. If the fault results in actuator failure, the control will not be able to influence the system. Hence, redundancy may be necessary to ensure that the system meets reliability guidelines, or in some cases it can prove more cost effective to duplicate than to take another approach.

Redundancy is where an element in a system is replicated by analytical or physical means. Physical redundancy is where multiple hardware channels are available in the system, and analytical redundancy is where devices are duplicated by means of suitable mathematical models (15). Analytical redundancy is only suitable for sensors, however, and is not applicable to actuators. This is attributable to the fundamental differences between actuators and sensors. Sensors deal with information, and the signals they produce may be processed or replicated analytically to provide fault tolerance. Actuators deal with energy conversion, and as a result actuator redundancy is essential if fault tolerance is to be achieved in the presence of actuator failure. Actuation force will always be required to keep the system in control and bring it to the desired state (1). No approach can avoid this fundamental requirement.

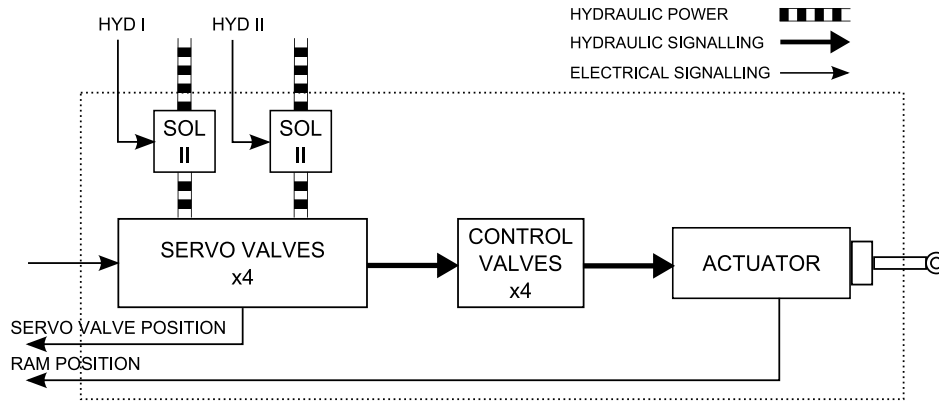


Figure 2.3: Block diagram schematic of a redundant hydraulic actuator.

2.3.2 Traditional Actuation Redundancy

The usual way of providing fault tolerant actuation is through parallel replication (Figure 1.1). Actuators that are capable of providing the required control action individually are arranged in triplex or quadruplex with some form of consolidation to sum their output. Hence, if faults occur within one or more of these actuators, a single remaining healthy actuator would be able to meet the control requirement, avoiding an actuation system failure.

An example of this parallel redundancy is illustrated in Figure 2.3, which shows a multiple redundancy hydraulic actuation system used within the Panavia Tornado's fast jet taileron and rudder control, details of which can be found in (16) alongside other aircraft redundant actuation examples. Only one lane of redundancy is shown within this diagram, and in effect there are four identical parallel channels. Each channel has four servo valves which are charged by two independent hydraulic feeds. These servo valves control the position of the first-stage valves which are mechanically summed before feeding into four control valves that modulate the position of the actuator ram. Failures are detected by comparing the outputs of the actuators in a voting scheme, which determines if one of the channels is faulty and subsequently removes this lane of actuation. At least three actuators are required to make the voting detection scheme effective, and as such this quadruplex actuation system can remain full operational even with two individual actuator failures.

Whilst this redundancy provides fault tolerance, this over-engineering incurs penalties as cost and weight are increased and subsequently efficiency is reduced. Also the use of actuation technology can be restricted using this direct parallel redundancy approach. A move towards more electric aircraft has been witnessed within recent years to improve weight, fuel consumption, installation and maintenance costs (17; 18; 19). As a result, electro-mechanical actuation has been used for control of secondary flight surfaces in civil aircraft and helicopter flight control systems. However, a major concern regarding the use of electro-mechanical actuation exists regarding actuator jamming. In hydraulic systems, each channel could be disengaged on detection of a fault, removing its influence from the output. However, if electro-mechanical actuators jam, they will effectively fix the whole parallel redundancy in place. This issue has prevented their use in the control of primary surfaces (16). Research is

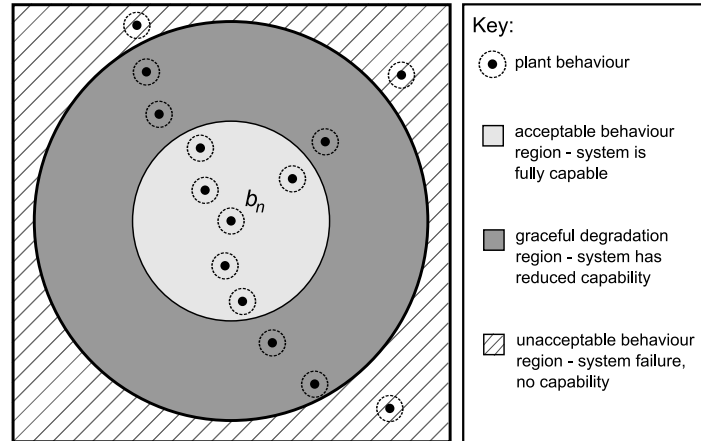


Figure 2.4: Diagrammatic representation of capabilities and graceful degradation in HRA.

currently underway to reduce the risks of jamming faults within electro-mechanical actuators (19).

These issues are well-documented in the aircraft industry, but they are by no means specific to this application. If new forms of actuation are to be used in parallel redundant configurations to provide fault tolerance for any high integrity system, the risks associated with actuator jamming and associated costs of over-dimensioning will still exist.

2.3.3 High Redundancy Actuation

The High Redundancy Actuation (HRA) concept is a novel topological solution for fault tolerant actuation. It is inspired by musculature, where the tissue is composed of many individual cells, each of which provides a minute contribution to the overall contraction of the muscle. These characteristics allow the muscle, as a whole, to be highly resilient to individual cell damage.

This principle of co-operation in large numbers of low capability elements can be used in fault tolerant actuation to provide intrinsic fault tolerance. HRA uses a high number of small actuator elements, assembled in parallel and series, to form one high redundancy actuator (see Figure 1.2).

When lock-up (jam) faults and loose (loss of force) faults occur in the actuation elements of this structure, a reduction in overall system capability, in terms of travel and force respectively, will result. However, the HRA has a capability that is in excess of the requirements of the wider system, and as such, a designed fault level can be intrinsically tolerated. As the quantity of faults in elements exceeds the pre-designed redundancy, the capability of the HRA will fall below that required. The capability will not immediately reduce to zero, however, the performance of the HRA will gracefully degrade. This idea of capability and graceful degradation are summarised in Figure 2.4.

The HRA has similarities to partitioning of processing within disk drives i.e. Redundant Array of Inexpensive Disks (RAID) schemes (20). System reliability and performance can be improved by using multiple disk drives, where each drive is not capable of providing the

required processing power, however, the entire array is in excess of the required capability, providing a margin of redundancy that can be used for system back-up. However, the consolidation of processing power is a more difficult task to achieve than consolidation of actuation outputs, which can be summed mechanically.

Potential advantages of the HRA concept include:

- Increased reliability
- Reduced over-dimensioning and weight.
- Intrinsic accommodation of both lock-up and loose fault modes.
- Graceful degradation.

The HRA solution deals with the issues of actuator lock-up (that was a concern for implementation within primary critical systems) and has the potential to decrease cost and weight, which may increase its suitability for further less safety-critical applications. However, these benefits are gained at a different cost: the HRA is a complex system. Arranging actuators in series introduces many more moving masses increasing the order of the system significantly. In addition, faults in actuators will change the behaviour of the HRA. For these reasons, control of HRA has been perceived as an issue in the past. Other new research challenges include:

- Quantification of reliability in multi-actuator structures.
- Health monitoring.
- Technology choice and manufacture.

Many of these challenges have been addressed to some extent (3; 4; 5; 6; 7; 21; 22; 23), but the research has particularly concentrated on the development of passive fault tolerant control approaches.

2.4 Fault Tolerant Control

The term Fault Tolerant Control (FTC) is used to describe a control strategy that is designed to retain stability and provide the required performance in the presence of system faults. A number of survey papers exist (24; 25; 26; 27; 11; 28; 29; 30), which provided an overview of FTC techniques at the time of their publication and several books provide an introduction to the area (31; 32; 33). Figure 2.5 illustrates how FTC strategies may be sub-divided.

All methods of fault-tolerant control can be described as either passive or active. Passive methods employ a single control law for all fault conditions, whereas active control methods change the control law in response to faults. This may be through online selection, where the controller is selected from a set of pre-designed control laws in response to faults, or through online synthesis, where the controller is synthesised in real-time.

Further discussions of these approaches to fault tolerant control, and their respective advantages and disadvantages, are made in the following subsections.

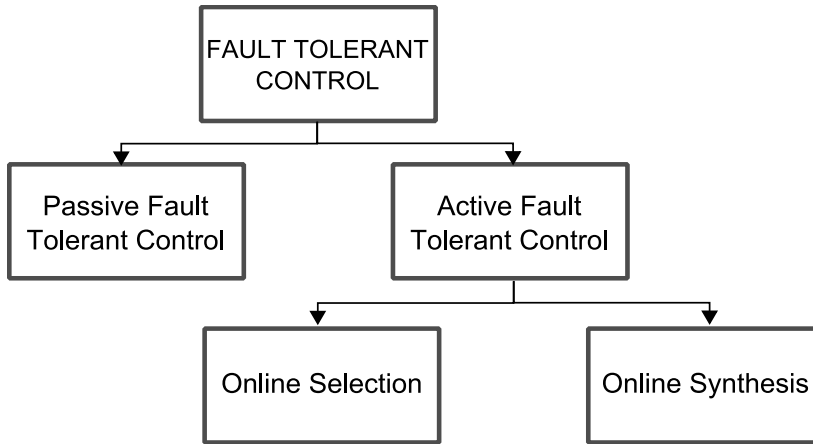


Figure 2.5: Fault tolerant control strategies.

2.4.1 Passive Fault Tolerant Control

Passive FTC is where a single robust control law is designed to provide adequate stability and performance under both nominal and fault conditions. This approach is attractive in its simplicity. There is no requirement for fault detection or control reconfiguration, making its stability more easily verifiable for high-integrity applications. Examples of passive fault tolerant control design can be found in (34; 35).

A well-designed, robust feedback controller will reduce the plant's output sensitivity to measurement errors and disturbance inputs (28), and in this sense the system is error-tolerant. If the system's behaviour under known fault conditions is also considered during the design of the robust controller, then some fault-tolerance may also be achieved. Generally, the design will only be able to accommodate a small number of faults, possibly only one (27). Nonetheless, this may be suitable for restricted cases, perhaps where a fault has a small effect on the system or if the effects of faults are similar to the effects of disturbances on the system (27).

This is applicable to the HRA concept. Within a truly high redundancy actuator, the number of elements is very large (in the order of 100-1000), and as such each element's influence on behaviour is small. Hence, the effects of faults on the overall system will also be small, and should be accommodated by robust control design.

When robust control is combined with redundant actuators, then this approach is referred to as reliable control (36). Hence, the HRA, when combined with passive FTC, may be termed reliable control. Examples of reliable control methods can be found in (37; 38; 39; 40; 41; 42). Within these examples, redundant actuators are arranged in parallel (Figure 1.1) and as such, the effects of faults are mediated by force averaging. A variety of robust control techniques are combined with redundancy e.g. Eigenvalue assignment (41), Linear quadratic regulator (42; 39) and \mathcal{H}_∞ control (40). There are a variety of books detailing robust control methods used in these examples (43).

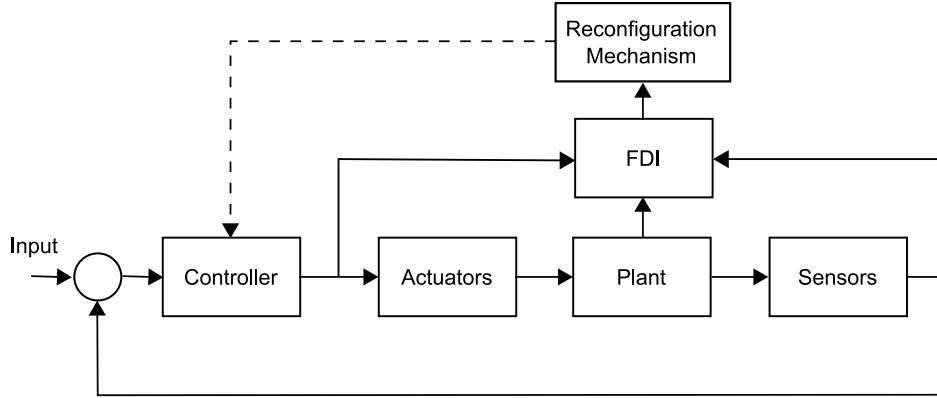


Figure 2.6: Typical active fault tolerant control scheme.

2.4.2 Active Fault Tolerant Control

In contrast to passive approaches, active fault tolerant strategies take action to accommodate faults. Figure 2.6 gives a general schematic for active FTC (32). Active FTC compensates for the effects of faults either by selecting a pre-computed control law or by synthesising a new control law in real-time, and as such, active FTC methods can be classified as online selection (sometimes termed ‘projection-based control’) or online synthesis (Figure 2.5). Both approaches usually require some method of Fault Detection and Isolation (FDI) to identify faults within the system and inform the reconfiguration of control.

2.4.2.1 Fault Detection and Isolation Methods

Significant research has been conducted into FDI methods, surveys of which may be found in (44; 15; 45; 46; 47; 48; 10; 49; 50; 51; 52).

In general, most FDI procedures consist of the following three steps (36):

1. Residual generation - Residuals are produced by comparing the system signals to a redundant signal.
2. Residual evaluation - The residuals are then compared to some predefined thresholds to produce symptoms.
3. Decision-making - Based on the symptoms, a decision is made as to which elements are faulty (i.e., isolation).

As mentioned in the first step, in order to identify a fault, the system signals must be compared to another other signal. This comparison will be achieved through redundancy, which may be physical or analytical.

Residual Generation through Physical Redundancy Physical redundancy involves the replication of hardware, such as sensors or actuators. This was the first method of FDI, and is still widely in use, which is probably owing to its ease of implementation and high certainty of fault detection in comparison to analytical methods. To detect faults in sensors

and actuators, multiple independent hardware channels are used and their outputs compared to produce residuals, although the technique is mainly aimed at sensor faults (1). These residuals may be evaluated via comparison to a set of a priori residual values, which indicate the fault type. The fault must then be isolated and the control strategy may be reconfigured.

Physical redundancy is a simple method of residual generation and benefits from a high certainty of fault detection. However, its disadvantages include (28):

- the added cost and weight of the redundancy,
- increased maintenance requirements,
- its inability to detect plant faults.

Residual Generation through Analytical Redundancy Analytical redundancy is a quantitative approach that negates the need for redundant components by constructing redundant signals from algebraic or temporal relationships between the states, and the inputs and outputs of the system. There are three main methods by which analytical redundancy can be achieved: parity relations; observers; or parameter estimation.

Parity Relations The parity space approach, pioneered by Chow and Willsky in the early 1980s (53), uses direct redundancy to produce residuals. A sensor directly measures a quantity and this quantity is also derived indirectly through the outputs of other available instruments, or both signals may be derived indirectly. The relationship between these two quantities is established under non-faulty conditions to form a nominal model. Actual measurements are then continuously checked against these relationships to produce residual functions. This allows faults in both the system and its actuators and sensors to be detected. Parity relations will only be effective as a means of FDI when:

- accurate mathematical models of actuators and the system under control are available.
- a slightly redundant instrument set is available, allowing sufficient overlapping to produce parity relations.
- all the quantities in the relationships are knowable and/or measurable with a low level of uncertainty.
- computation can be demanding in real-time application.

Observer-based methods The second approach to analytical FDI uses Luenberger observers (in deterministic settings), or Kalman filters (in stochastic settings), to estimate system outputs from measurement subsets in order to compare them to actual outputs and produce residuals. The simplest observer-based scheme uses the most reliable sensor output to reconstruct a whole measurement set for comparison to the actual measurements (45).

More complicated schemes may involve a number of observers producing estimates based on each sensor output, examples of which can be found in (54). This allows multiple faults

to be detected simultaneously. This method would require a considerable amount of computational effort, especially if the number of sensors was large.

Other methods use parallel banks of filters based on different system fault states to produce multiple residuals (55), which can be analysed to determine which fault state is in effect within the system. This method again requires considerable processing power if the number of fault states is large.

In addition to computational demands, a deep analytical understanding of the system under control is required to implement observer methods. However, advantages exist as the procedure of observer design is systematic; non-linear systems can be treated and a very sensitive reaction to faults can be achieved.

Parameter Estimation Parameter estimation methods are based upon the concept that faults in systems can manifest as changes to the parameter values of the system. Accurate parametric models of the system are required in order to identify parameter changes. Surveys of various methods using estimation principles such as ordinary and orthogonal least squares methods can be found in (48; 56). Sliding-mode (57) and extended Kalman filter methods (58) have also been suggested where the parameters are estimated in addition to the states. As with observer-based methods, parameter estimation can be computationally demanding if the system is complex.

Other, non-quantitative methods of residual generation are available such as Bayesian networks or Artificial Neural Networks (ANN) (59; 60; 61; 62; 63; 64), where models of the system are not required, and historical data are used to train the system. However, care must be taken to ensure the network receives sufficient training, and that the data sets used are representative of system behaviour if the result is to be accurate and useful. The required training periods and data requirements can be limiting factors.

Overall, analytical redundancy offers the advantages of reduced cost and weight when compared to physical methods. However, analytical methods often require accurate modelling of the system and its uncertainties, a deep understanding or statistical knowledge of the system, and extended design and tuning periods.

Residual Evaluation and Fault Isolation Once residuals have been produced, whether via physical or analytical means, the residuals must be evaluated to determine whether a fault has occurred and if so, the fault location must be determined.

Suitable residual thresholds may be determined through experimental or simulation tests. The number of thresholds exceeded and the values of the residuals characterise the fault symptoms and indicate the fault location. These thresholds will mainly depend upon the level of uncertainty in the measurement and the modelling errors, i.e. if uncertainty levels and modelling errors are high, higher thresholds will have to be implemented to ensure that the FDI does not falsely report a fault. However, if the thresholds are raised then the possibility of missing a fault is increased. The balance between the risk of false-alarm and missed-alarm is difficult to achieve (27). A number of techniques have been proposed to tackle this problem. (65; 66; 67; 68) suggest methods that decouple the uncertainties from the fault signals, thus

allowing thresholds to be lowered, making the system more sensitive to faults. Additionally, (69; 70; 71) propose more robust FDI methods that improve fault detection in the presence of modelling uncertainties. However, these methods do not work in the presence of both model uncertainties and disturbances (72). Methods proposed in (72; 73) claim to produce robust FDI schemes in the presence of both.

Other solutions to this problem may be found in adaptive thresholds, where each threshold becomes a function of the measurable quantities (74; 68). If a residual exceeds its threshold, then a symptom can be derived.

Learning-based methods such as neural networks may also be employed in diagnosing faults from quantitatively produced residuals, resulting in a hybrid fault detection scheme (75; 76), care must be taken to train the network appropriately.

Statistical testing of residuals is another option for fault diagnosis and isolation, where the likelihood of the fault state may be produced (55; 44; 77). This approach is applied and further discussed in Chapter 6.

2.4.2.2 Online Selection & Online Synthesis Control Algorithms

Having detected a fault and isolated its location and nature, an active FTC system then alters the control so that the stability and performance of the system can be maintained at an ideal or acceptable degraded level.

Control law re-scheduling, multiple-model and interacting multiple-model approaches are all examples of pre-computed active FTC methods. In each case, a number of control laws are designed offline to meet requirements under certain fault situations and these laws are enforced according to the fault state of the system as detected by the FDI algorithm. Examples of these approaches can be found in (78; 79; 80; 81; 82).

There are numerous online-estimation based methods for control reconfiguration, including pseudo-inverse methods (83; 84), eigenstructure assignment (85; 86; 87), model-following (88) and sliding mode control (89; 90).

There are advantages and disadvantages to both online selection and synthesis approaches. Pre-computed control laws are only useful for anticipated faults, whereas an online scheme could adapt to an unexpected fault. However, there is a risk of instability associated with active fault tolerant control methods. The FDI unit may diagnose a fault incorrectly, leading to a mis-reconfiguration of the control system where the stability of the closed loop system is not guaranteed. Pre-computed strategies may mitigate this risk by only including controllers that will maintain the closed-loop stability in any fault case (91). However, controller design in this case may prove to be conservative.

The two approaches have differing computational requirements. The memory requirement for storing a priori designs must be considered in the pre-computed case, whereas computational power is more of a critical factor for online-synthesis. Generally, memory requirement is less of an issue than processing capability.

The transition period is another consideration. The time taken between fault detection and control reconfiguration is vital, as in this time actuators could saturate or further damage

could be done to the system. Therefore, the quicker the new control law can be achieved the better, and online synthesis will be slower than pre-loaded methods.

This transition time is one of the main problems with active FTC. Another problem is the unwanted transients induced during the reconfiguration process. These transients may be harmful to the system itself, its environment or its human operators. This issue has not been addressed to a great extent, but one paper that treats this issue is (92).

2.5 Multi-Agent Systems

The active fault tolerant control approach described within Chapter 7 is based upon multi-agent concepts, and as such it is useful to provide an overview of multi-agent systems in order to establish a foundation for the work that follows.

The concept of an agent was first given by Minsky (93). In his book, 'The Society of Mind', he introduced the term agents to describe the workings of the mind. Each agent is only capable of a simple process, but these agents are numerous and diversely capable, and it is through the interaction of these agents that true intelligence can be achieved. The principles of Multi-Agent Systems (MAS) were further developed in the disciplines of distributed artificial intelligence and object-oriented programming 30 years ago, since then it has emerged as a discipline in its own right. Today, MAS concepts have become not only an important subject of research, but of industrial and commercial application in a diverse range of fields (94).

This section aims to introduce issues that are central to MASs for use within the active fault tolerant control approach of Chapter 7.

2.5.1 The Agent Concept

There is still some controversy within the agent community as to the exact definition of an agent and what qualities an agent must possess. There is a general consensus that autonomy is essential within an agent, however, the attribution of other qualities is still under debate. However, Jennings, Sycara and Wooldridge (95) provide a definition of an agent, which emphasises the fundamental characteristics of the agent philosophy:

‘An agent is a physical or virtual entity situated in an environment that is capable of flexible autonomous action in order to meet its design objectives’ (95)

This definition encompasses three key features:

- Situation - the term, used in this context, means that the agent exists within an environment which it receives sensory information from and has the capability to act upon.
- Flexibility - the agent is described as flexible as it not only responds to its environment, but also exhibits opportunistic, goal-directed behaviour. Hence the agent is not only reactive, but pro-active. The term flexible also encompasses the agents social ability. An agent should be capable of interaction with other agents in order to achieve their objectives, and aid the activities of others.

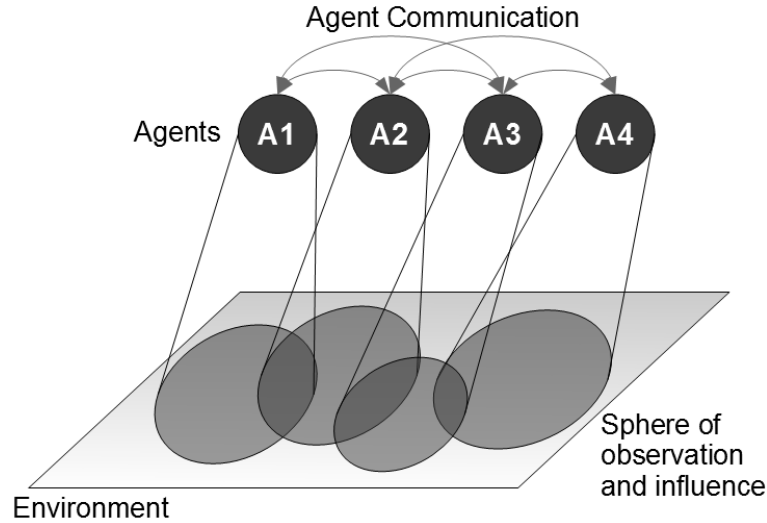


Figure 2.7: Multi-Agent system.

- **Autonomy** - the agent should act without the direct intervention of humans or other agents and have control over its own behaviours.

Certain aspects of this definition may be more important than others according to the application, but it is the presence of all these features that separates the agent concept from related fields such as object-orientated, distributed control and expert systems. The agent's situation, flexibility and autonomy resembles the concept of closed-loop control, as a closed-loop controller also senses and acts, and is designed to satisfy some objective. However, there are important differences within the agent concept. The most obvious difference is the social interaction and negotiation between agents. Also, the agent philosophy is strongly associated with localisation, as an agent often has a partial representation of the overall problem. This point is emphasised within (96).

Other features that are routinely attributed to agents include learning and mobility. However, it is thought that these aspects should not be included in a strict definition of an agent as their applicability is often questionable. In some applications learning may be of paramount importance, for others, it is very undesirable. For example, the self modification of an agent's behaviour would be unthinkable in terms of safety for systems such as Georgeff's Procedural Reasoning System (PRS) (97) which provides air traffic control. Also, the ability for an agent to move from one computer to another and be capable of execution on various platforms may be desirable in applications such as e-commerce, but unnecessary and unfeasible in most applications.

2.5.2 Multi-Agent Systems

A multi-agent system, as its name suggests, is a collection of agents that collaborate to achieve an objective. As each agent has only a partial knowledge and influence on the system, the agents must work together to solve problems that are beyond their individual capabilities. This point is illustrated in Figure 2.7.

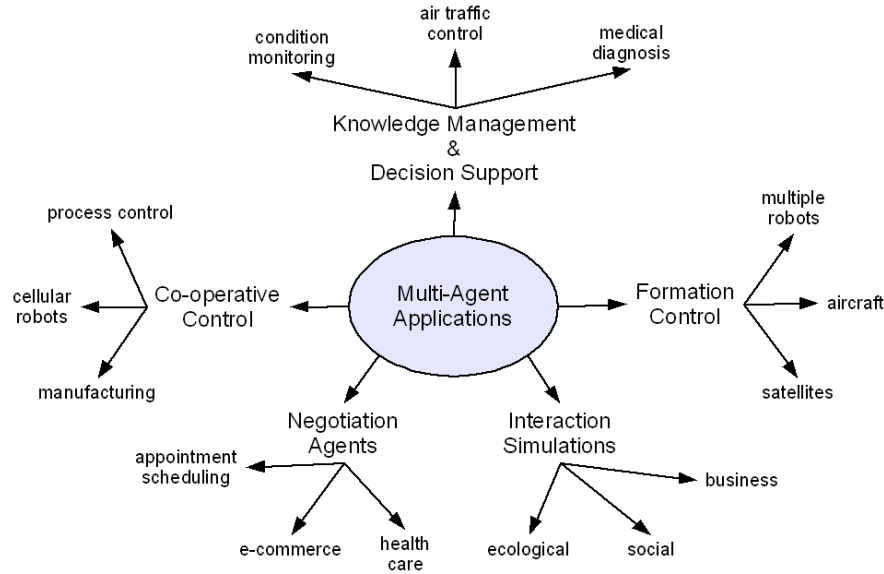


Figure 2.8: Multi-Agent applications.

Jennings, Sycara and Wooldridge go further in their definition of a MAS by specifying that, in addition to their localised nature, MASs have no global system control; their data is decentralised and they operate asynchronously (95).

2.5.3 Typical Applications

Multi-agent concepts are being researched or applied in a diverse range of applications, spanning industrial, commercial and public sectors. Generally however, the application of MASs falls into one or more of the categories shown in Figure 2.8. These categories are discussed below.

Formation Control MASs are suited to applications where a good degree of modularity exists. This is certainly true of formation control applications, where the spatial arrangement of multiple-robots (98; 99), aircraft, satellites or other autonomous vehicles needs to be managed, as each entity can be considered a module with its own state variables, and indeed it is employed within these fields.

Knowledge Management & Decision Support Today's world has become knowledge rich. There is an abundance of information available in databases and via the Internet. The sheer volume of data can be prohibitive, as it becomes increasingly difficult and time consuming to find the required information. Knowledge management is required to bypass the tedious and often inefficient task of search directing and find information in a more extensive and systematic manner. Agents can perform this management by filtering and gathering information on the behalf of a user whilst also adapting to the user's behaviours and preferences, learning what is useful and what is not.

Decision support takes knowledge management one step further. There are many large distributed systems such as energy distribution networks, environmental emergency services

and air-traffic control that require real-time decisions to be made based on an overabundance of information. Agents can be deployed in these cases to manage and filter this data and present summarised relevant information in a timely manner whilst also making lower-level autonomous decisions or giving warning of the development of undesired events. Examples of agent based knowledge management and decision support applications can be found in (100; 101; 102; 103).

Interaction Simulation Some large-scale modelling scenarios contain elements that interact in a co-operative, negotiative, or competitive manner such as business, social and ecological environment models. These elements can be effectively simulated through the use of agents, which allows social interaction between entities that have individual beliefs, pragmatical abilities and agendas.

Negotiation Agents The social and negotiative abilities of agent technology exploited in interaction modelling, can also be used to orchestrate and negotiate agreements with other agents, systems or humans on the behalf of a user. Agents, given their own agenda, can negotiate terms much more quickly and efficiently than human counterparts. Applications of this nature would include e-commerce transactions (104), meeting scheduling and health care applications.

Co-operative Control Perhaps the most relevant application with respect to the HRA, co-operative control encompasses those applications that have distributed modules that need to work together to achieve a common goal. Examples of this include process control (105), manufacturing (106), multiple degrees-of-freedom robots (107), and cellular robots (108).

2.5.4 Central Concepts in Multi-Agent Systems

2.5.4.1 Agent Architectures

Having defined what an agent is and its general characteristics in Section 2.5, the inner workings of an agent may now be considered. This can be termed as the architecture of an agent. Agent architectures describe the data structures within an agent, the possible operations performed on these data structures and the control of flow of data between them.

A formalisation of agent architecture can be made by taking a state-based perspective. The formalisations made here are based on those made by (94). The environment in which the agent is situated can be characterised by a set of environment states:

$$S = \{s_1, s_2, \dots\} \quad (2.1)$$

At any given time it is assumed that the environment is in one of these states. The agent senses the environment in some manner, through a sensory set and forms some perception based on this sensory information. These perceptions are represented as a set:

$$P = \{p_1, p_2, \dots\} \quad (2.2)$$

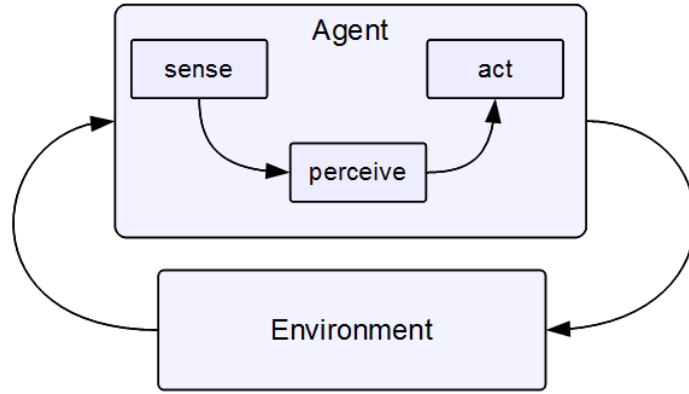


Figure 2.9: Abstract agent architecture: Purely reactive agent.

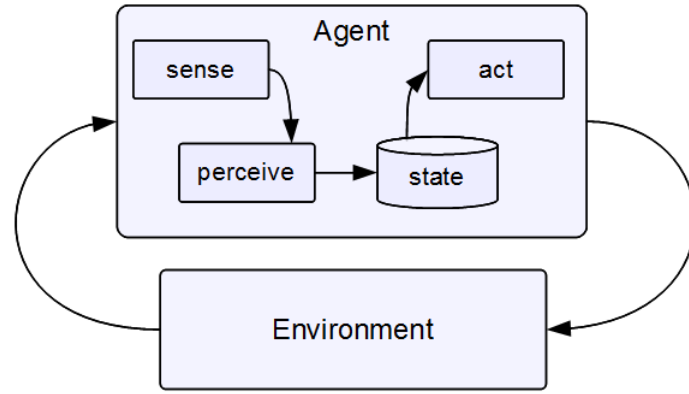


Figure 2.10: Abstract agent architecture: Agent that maintains state.

It is worth noting that P does not necessarily equal S . Due to the localisation of sensory ability intrinsic in the multi-agent definition, there will be states of the environment which a particular agent will not be able to differentiate between.

Finally, the actions which the agent is capable of applying to the environment is similarly represented by:

$$A = \{a_1, a_2, \dots\} \quad (2.3)$$

Based on these formalisations, there are two basic agent architectures: purely reactive agents and agents that maintain state.

Figure 2.9 gives an abstracted representation of a purely reactive agent architecture. Purely reactive agents perform actions based entirely on the perceived current state i.e. no consideration of previous environmental states is taken. This behaviour is represented by the functions:

$$\begin{aligned} \text{sense} &: S \rightarrow P \\ \text{act} &: P \rightarrow A \end{aligned} \quad (2.4)$$

Agents with state however, have an architecture akin to that depicted in Figure 2.10.

The agent now has an internal data structure, which is used to record history of the agent's perceptions. This gives the agent an internal state:

$$I = \{i_1, i_2, \dots\} \quad (2.5)$$

The agent's actions are now not only based on the current perception, but are also affected by the internal state of the agent. The sensed environment is still mapped to P as before. However, the action is now based on I , which is the mapping of the current perceived state and internal state.

$$\begin{aligned} \textit{sense} &: S \rightarrow P \\ \textit{next} &: I \times P \rightarrow I \\ \textit{act} &: I \rightarrow A \end{aligned} \quad (2.6)$$

These formalisations describe agent architectures in the abstract. However, they do not describe how actions are decided upon or how perceptions and internal states are formed. These aspects will be discussed by considering the main agent architecture classifications in the following sections.

Logic-Based Agents Logic-based agents were developed from the artificial intelligence planning field, that concerns itself with creating intelligence that decides what actions to take. Most logic-based agents are based on the symbolic representations made by Newell & Simon (109). Generally, an agent of this type will create a symbolic model of its environment, represented by some first-order predicate logic, based on its perceptions. The agent will also have a list of possible actions, symbolically formed, that specify the circumstances under which the action can be taken and the effects that action would have on the environment. A planning algorithm takes the model, the action set and a state representation of its goal and plans the required actions which would result in the attainment of that goal state.

Hence, the agent formulates a plan of action from first principles and will produce an optimum solution to the problem. However, the time taken to formulate symbolic models of the environment and logically deduce what actions are required restricts this schemes use within a real-time environment, as the environment state may change during the lengthy planning period.

Behaviour-Based Agents Behaviour-based architectures do not reason about their environment, avoiding the environment modelling and action planning present in logic-based agents. Instead, these architectures combine simple behaviours to produce complex intelligent overall behaviour. This is commonly termed as emergence.

The most widely known behaviour based architecture is subsumption (110), a representation of which is shown in Figure 2.11. The architecture consists of a number of finite-state machines, which contain task accomplishing behaviours. These behaviours are arranged in layers, in order of their abstraction with the most reactive, basal behaviour on the bottom

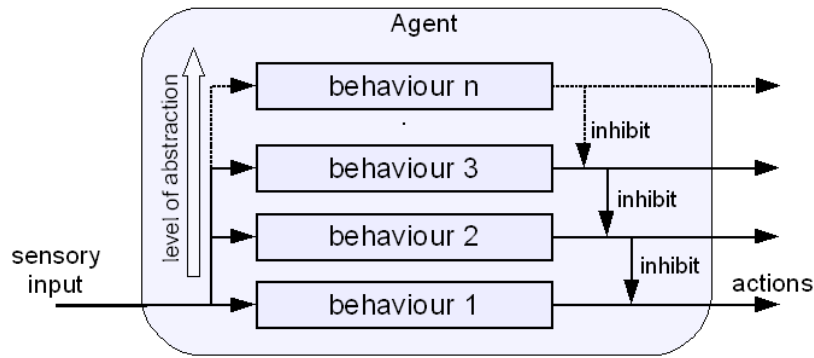


Figure 2.11: Subsumption

layer. Each layer receives the sensory input, which is still formally represented as a perception. However, this perception is not a symbolic representation of the environment but more akin to raw sensory data. The perceptions map to an action within each layer in the same manner as described in 2.4. The higher layers inhibit those below it, and as such a behaviour's action is executed if it is not inhibited.

The computational simplicity of this scheme results in reaction times more than sufficient for real-time implementation. The resultant behaviour is robust and timely, but not optimal as it was in logic-based agents (111). In addition, goal-directed behaviour is difficult to achieve through subsumptive architectures, as the state is not retained.

Practical Reasoning Agents Deliberative agents, in contrast to reactive agents, reason about what goals they want to achieve and how they will achieve them. Logic-based agents are the extreme embodiment of deliberative architectures. However, the most prominent deliberative architecture is the Belief-Desire-Intention model (BDI) which is a practical reasoning agent (112). Practical reasoning architectures are based upon theories regarding the reasoning capabilities of humans. These theories often attribute attitudes to the human thought processes such as beliefs, desire and intentions and regard the interaction of these attitudes as the root of rationality and planning.

Figure 2.12 gives a representation of the BDI architecture. The beliefs of the agent are formed from its current and previous perceptions. Desires represent the options, based on its current beliefs, that are available to the agent. The desires are then filtered to form the agent's intention i.e. its chosen course of action. This intention will be executed until the environment changes and its beliefs are altered.

This architecture allows the agent to achieve goal-directed behaviour, whilst also reacting to its environment to a greater degree than that afforded by logic-based agents, as no complete symbolic environment model is formed. However, the delay between perception changes and intention formation can still be inhibiting for real-time use in highly dynamic environments.

Hybrid agents Hybrid agents attempt to combine both deliberative and reactive architectures in order to produce a reactive agent capable of achieving long-term goals. This is frequently realised through layering of the architecture. This may be vertically or horizon-

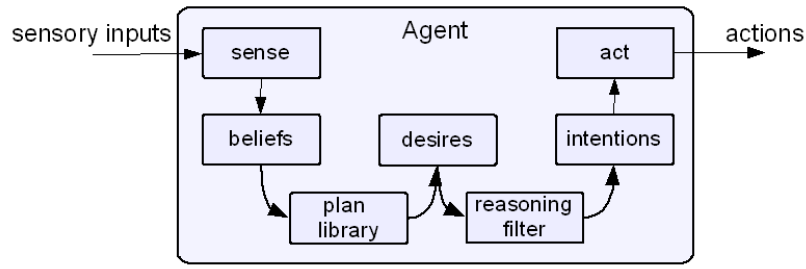


Figure 2.12: Beliefs-Desires-Intentions architecture

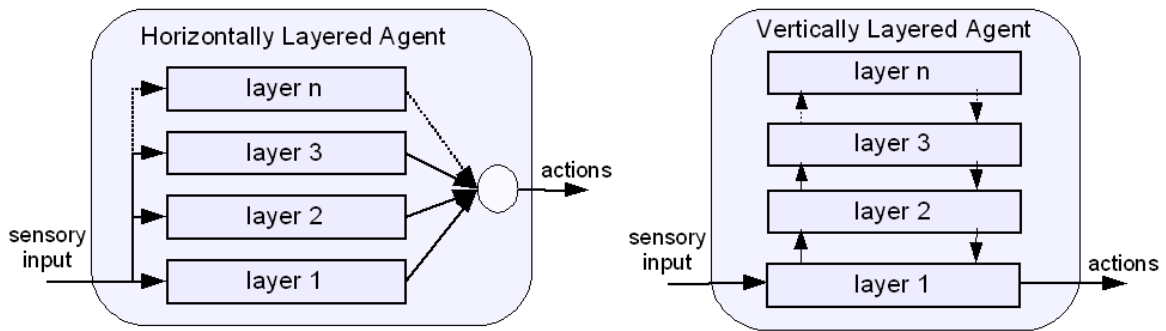


Figure 2.13: Hybrid architectures

tally as shown in Figure 2.13. The reactive layer is the lowest layer, which in itself, may be a number of reactive layers arranged as in subsumption. Higher layers deal with the formation of more complex beliefs from the sensory data, plan formation and communication with other agents. This architecture provides the first obvious structure for implementing co-operative multi-agent systems.

An example of a hybrid architecture is proposed by Franklin (112). Hybrid architectures potentially provides all the attributes necessary to classify as a MAS: autonomy, reactivity, pro-activity and social flexibility. However, the co-ordination of the layer's actions is an issue that can be difficult to solve. The co-ordination of the agents must also be considered, but these issues are more associated with the structure of the agency, which is discussed within the next section.

2.5.4.2 Agency Structures

Agency structures define the configuration of multi-agent systems and the interactions between the individual agents. Hence, this section will introduce the central ideas regarding the architecture of agencies, communication between agents and the co-ordination of their actions.

Communication Communication between agents is key, as it allows agents to share information and co-ordinate their actions to achieve individual or global goals that are beyond their individual capabilities.

Forms of Communication Communication in MASs can exist at a number of levels, ranging in sophistication. A MAS at its most primitive can contain no direct communication at all (113; 114). Agents, instead infer and speculate on other agent's plans and states based on observation of the agent's behaviour and its sensory knowledge of the environment. Communication of this sort is present in the field of game theory. However, communication of this sort can be very prohibitive to agent reasoning. One only has to examine a game of poker or bridge to realise that speculating on an opponent's or partner's hand and intentions based on their behaviour and the cards played, that reasoning and planning can become complex even in a situation where the rules and objectives are clearly defined. The situation also becomes highly inflated when large numbers of agents are involved.

Another indirect form of communication is through a shared data structure, where agents write their knowledge and plans and can access that of others. This method is called blackboarding and will be considered further later in this section.

Direct communication at its most basic level involves the swapping of primitive signals, with finite values and fixed interpretations in order to communicate basic intentions. A scheme with this level of communication was implemented by (112).

Message based communication is the next level of communication possible. Agents utilising this approach pass messages regarding their state, observations or plans depending on the purpose of their communication. The complexity of the messages depends on the characteristics of the agents involved. An early example of message based communication between agents is the Actors system (115).

Finally, the most sophisticated form of communication available to MASs is through high-level languages that are based on natural language formations, speech act-theory and conversational theory.

The communication made can be synchronous or asynchronous and be addressed to a single recipient or have multiple destinations. These features are determined by the agent and agency architecture, which ultimately determines the structure of interaction between the agents. In general, modes of communication may be:

- Point-to-point - In point-to-point communications the message is addressed to a specific agent. This type of communication is common in autonomous agency architectures (Section 2.5.4.2) that frequently contain deliberative agents.
- Broadcast - Messages can be broadcast to all rather than addressed to any particular recipient. This is useful in an environment where the presence of agents is uncertain and is utilised in schemes such as contract-net.
- Multi-recipient - As its name suggests, multi-cast communications are those made to a particular subset of agents. This may be a group of agents whose address is known to the sender and are known to be an interested party in the message content, or may be a group of agents determined by a description of the the service required as in broker architectures.

Communication Protocols and Languages Communication protocols are necessary to support the interaction of agents. There are two levels of protocol needed: one to support the communication linkages of the agent and another that supports the communication of information.

Probably the most well-known of the former is contract net (116). This protocol provides a method of dynamically assigning manager and contractor roles to agents, which specifies the subsequent form of their communications, in order to achieve tasks in a co-ordinated manner.

Agent communication languages (ACL) are the protocols that support the exchange of information. Well-known ACLs include:

- Knowledge Query and Manipulation Language (KQML) - KQML was developed in the early 1990s through the U.S. government's DARPA knowledge sharing program whose goal was to develop a high level language based on speech act theory to aid agent co-operation. It has been frequently commented however, that KQML lacked the precision, consistency and definity required for wide-spread, inter-operable use. Thus, in recent years, the use of KQML has been superseded by FIPA-ACL.
- FIPA-ACL - FIPA-ACL (Foundation for Intelligent Physical Agents) is also based on speech act theory and has similarities with KQML. However it was directly created using a rigorous semantic formula, which removed the ambiguity present in KQML. FIPA is a standardisation body which began its operations in 1995. It specifies 22 performatives that define the message content's type, and the expected subsequent flow of communication. Readers who wish to know more are directed to the FIPA website (117).

Content Languages and Ontologies The actual content of the message could be in any language, although FIPA has proposed standards for four different content languages: FIPA-Semantic Language (FIPA-SL); Knowledge Interchange Format (KIF); Resource Definition Framework (RDF) and Constraint Choice Language (CCL). The choice of language is important, as it defines how the ontology is shaped. The ontology is the specification of terms for objects, concepts and relationships necessary if the agents are to understand each other.

Co-ordination It is not enough to communicate within a MAS. Co-ordination of the individual agent's actions is necessary if tasks are to be achieved efficiently whilst meeting global constraints.

Various methods of attaining co-ordination exist, and often they are closely related to the architecture of the agency. The main approaches to co-ordination in general are (94):

- Direct supervision - Co-ordination can be enforced on a MAS by creating an agent that has more perspective on the workings of the system. This agent can then direct the sharing of information and assert some level of control over the actions of the agents. This creates a hierarchy, which is undesirable particularly in large systems where the bottle necking of information will occur.

- **Mediation** - Intermediary agents may be utilised to co-ordinate agents actions by controlling the flow of information between agents and matching an agent's needs and resources to other agents.
- **Reactive co-ordination** - Co-ordinated group behaviour can emerge by agents reacting independently with basic behaviours to its stimuli. This is evident in flocking, where flocks of birds turn with perceived synchronism due to the individuals sensing its neighbours movement and reacting to it.
- **Standardised co-ordination** - The standardisation of procedures in response to defined situations brings co-ordination into a multi-agent system in the same way as cultural norms and social laws bring order into human societies, without the added complexities of true free-will. This standardisation can be built into the system, or enforced by some form of supervision.
- **Mutual adjustment** - Finally, co-ordination can be achieved through information and plan sharing between agents. In this scenario, no agent has control over others or the flow of information. The mutual adjustment might be realised through negotiative communications between the agents or through agent modelling, where each agent plans around the plans of others. This is known as distributed multi-agent planning (118).

Agency Architecture The manner in which an agency is organised ultimately determines how effectively the individual agents communicate and co-operate. There are many ways in which agents may be organised. In general, however, architectures tend to fit into one of three categories as defined by (111): hierarchical, federated and autonomous architectures. These categories will be discussed in the following subsections.

Hierarchical Architectures An hierarchical agency architecture is where at least one agent, or group of agents have authority over another (Figure 2.14). Traditional control structures and supervisor systems use hierarchical control, where certain elements of the system will be superior to others. As discussed earlier, this centralisation of decision-making has issues associated with reliability and flexibility that MAS concepts attempt to avoid. Indeed, the idea of hierarchy within an agency architecture is not strictly in-keeping with the central ideas of multi-agent systems. However, it can be difficult to leave the hierarchical mindset, particularly when the application is arranged in an hierarchical fashion as it often is in manufacturing. Several hierarchical multi-agent designs have been proposed in this application area (119; 120; 121).

Autonomous Agency Architectures Autonomous agency architectures are the antithesis of hierarchy based architectures. No agent within the architecture is controlled or managed by another. Hence, these architectures are the apotheosis of multi-agent concepts. Figure 2.15 illustrates a typical autonomous agency architecture.

All agents make independent decisions regarding their actions and with whom they communicate. They communicate directly with any other agent in the agency at will. Each agent

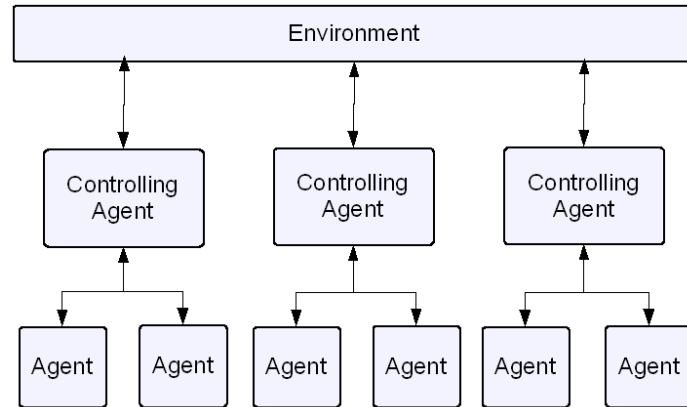


Figure 2.14: Hierarchical agency architecture

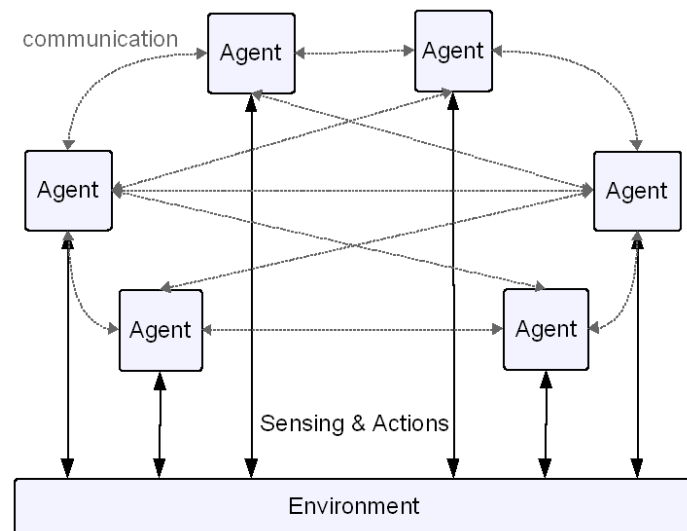


Figure 2.15: Autonomous agency architecture

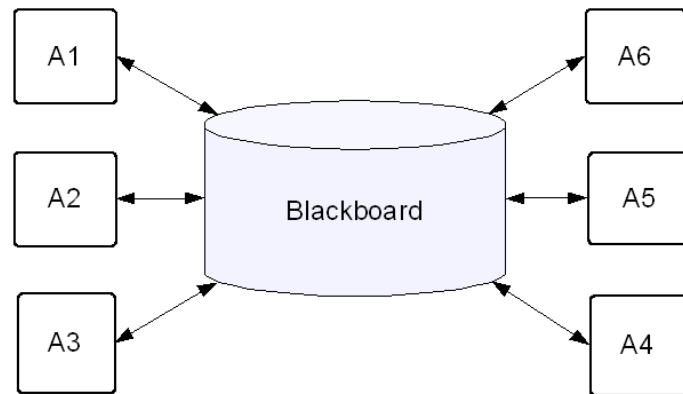


Figure 2.16: Blackboard-based agency architecture

will build up knowledge of the other agents i.e. there is no shared data structure that contains information about the agents or knowledge they acquire.

Communication and co-ordination of the agents actions in autonomous schemes is much more difficult than in hierarchical agencies. However, the interactions between the agents are much more flexible.

Federated Architectures The communication and co-ordination difficulties present in autonomous architectures can be aided by compromising a little on the flexibility of the system. Federated architectures allow agents to make independent decisions. However, it involves some mechanism for the sharing of data. This is often realised through an agent that controls the interaction between the resource agents. Several schemes have been proposed that implement this federated approach, these include:

- Blackboards
- Facilitators
- Brokers
- Contract-net

Each of these schemes will be discussed briefly here.

Blackboards The blackboard was an early solution proposed by (122) to agent co-ordination problems. Agents within a blackboard system are considered to be specialists in their knowledge or capabilities. The agents use a shared data structure on which the information they possess is written. It is analogous to a problem being solved by a group of experts around a blackboard. The problem and initial information is presented on the board and the expert that is capable of beginning to solve it does so on the board. Other experts step in when their particular expertise is applicable until the problem is solved. However, no communication takes place between the experts.

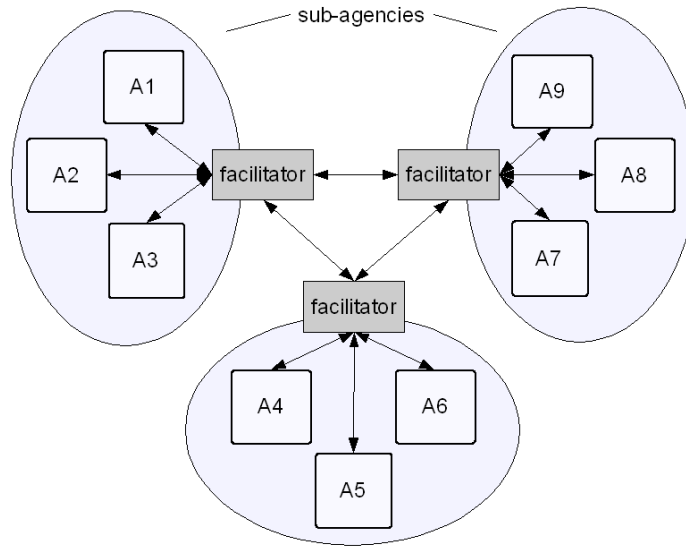


Figure 2.17: Facilitator agency architecture

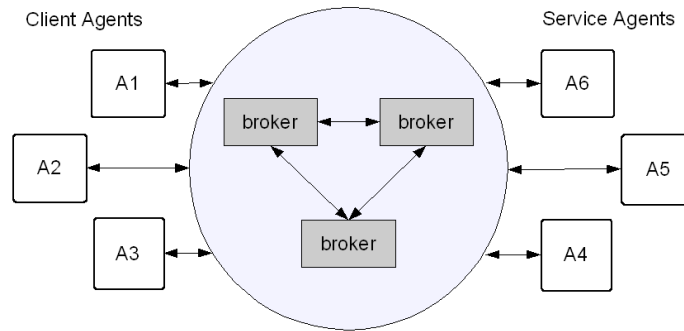


Figure 2.18: Broker agency architecture

Facilitators Figure 2.17 shows a typical agency architecture that uses facilitator agents. The facilitators route messages amongst agents and agencies and co-ordinate their action. This provides a reliable message routing system and allows agents to send and receive messages in their preferred format. The inter-agent communication that was present within the autonomous agency architecture is lost within this scheme, removing some of their independence regarding whom they communicate with.

Brokers A broker scheme works much like brokers in the real world operate. Figure 2.18 gives a general representation of a broker scheme. Within a MAS that utilises broker agents, there will be a group of agents that are able to provide a range of services and a group of client agents that require some service. The broker will receive requests from client agents and find suitable service agents to execute the request. Brokers may then continue to mediate the interaction between the service and client agents, or allow communication to be directly linked between the two.

The FIPA abstract architecture (123) is essentially a broker type architecture designed to facilitate inter-operability between agent based systems developed by different companies

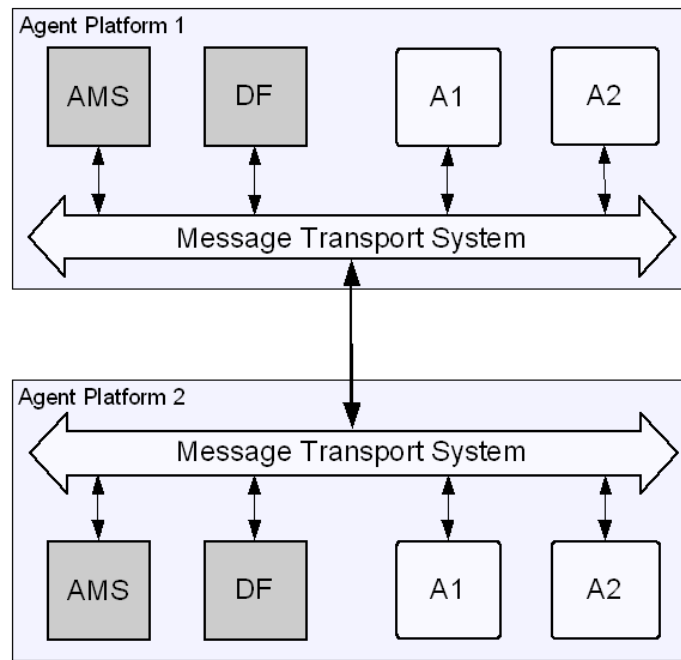


Figure 2.19: FIPA abstract architecture

(see Figure 2.19). The agents exist on a platform that provides some message transport system that allows them to communicate with each other. On each platform there exists two utility agents: a agent management service agent (AMS), and a directory facilitator (DF). The former is compulsory, but the DF is optional. The AMS works as a white pages, keeping a record of all the agents registered on the platform. The DF is a yellow pages, maintaining a record of the services that each agent can provide. The agents communicate with the DF to find agents that can provide the services they require, after which the communication is directed agent-agent.

Contract-net Contract-net protocol was first presented by (116), and provides a more flexible federated architecture. Agents using contract-net protocol dynamically assume one of two roles: manager and contractor. A manager agent is given a task to perform and decides if the task can be decomposed into sub-tasks. The manager then calls for bids from the contractor agents to perform the task/sub-tasks. The contractor agents then reply, communicating how effectively they could perform the task and the manager awards the contracts optimally. The contractor then performs the task and reports to the manager on the tasks results. This, however, is a communication intensive method of agent co-ordination.

Comparison of Agency Architectures Inevitably, the specific application will play a large role in deciding which agency architecture is most effective, but there are some general conclusions one can make regarding these architectures.

Autonomous agency architectures are the ideal embodiment of MAS concepts, providing agents with high independence and flexibility. However, their direct communication links can be inhibiting in agencies that contain agents with heterogeneous communication forms.

In addition, in agencies containing very high numbers of agents, it is foreseeable that direct communication may become cumbersome.

Some degree of the independence afforded by autonomous approaches is lost within federated architectures, as agents must communicate through some interaction controlling agent or a shared data structure and thus agent interaction is now restricted. Delegation of communication routing and some level of task co-ordination present in federated approaches is intended to free the agents, allowing them to concentrate on their other activities. This does not compromise the non-hierarchical nature of MAS concepts, as the intermediary agent does not have control over individual agent decisions. This is particularly true of schemes that implement protocol like contract-net, where the interaction structure of the system is dynamic.

Hierarchical architectures, however, do not strictly adhere to the concepts of multi-agent systems. Nonetheless, their use within existing hierarchical systems is understandable. If the full advantages of MAS concepts are to be realised however, it is necessary that a move to federated or autonomous architectures is made.

2.5.5 Multi-Agent Systems & Fault Tolerant Control

The use of multi-agent concepts in fault tolerant systems is a relatively small, but growing subset of agent applications.

There are innumerable instances where agent techniques have been used in FDI schemes (124; 125; 126; 127; 128; 129; 130). This may be due to the similarities that fault detection has to decision support applications. In many of these FDI schemes, the agents diagnose faults through data analysis and present fault information either to operators or separate reconfiguration mechanisms. Rarely do the agents both diagnose and take measures to compensate the fault within the system, a point acknowledged by (131). This seems perverse, as the multi-agent ethos is that agents should both sense and act in their environment.

There are some examples of agents diagnosing faults, and taking remedial action within power network applications (132). Power networks contain many local controllers in the form of automatic voltage regulators and power system stabilisers which regulate the voltage without consideration of network-wide conditions. Efforts have been made to use agent concepts to distribute and automate the emergency control decisions to these local units, which may shed loads to retain the stability of the network.

Agent techniques have attracted much attention in power network applications (133; 134; 135; 136; 137; 138; 139), and likewise in networked control systems. This is due to the structure and uncertain architecture of these problems suits the localised perspective that agent concepts offer. The use of multi-agent concepts in process control is also quite common (140; 141; 142; 143; 144) as these applications value the modularity, flexibility and openness that agents can provide (144). The issues tackled within these applications are special and somewhat distinct to the challenges in normal fault tolerant applications. Agents are mainly used to provide encapsulation to tackle compatibility issues in these applications, which often require new system elements to be added to old infrastructure, and to manage communication

links in uncertain environments.

However, the basic agent concept has something to offer those control applications which are not necessarily concerned with open architecture issues. Multi-agent concepts can be used to provide structuring, which is often neglected within the field of control engineering, as the problem is usually stated in the form of a single plant model (145). Its localised autonomous qualities can also be used to integrate fault detection and control reconfiguration into the structure that it provides.

Despite this, multi-agent techniques have been under-utilised in control and automation (143), although a few examples do exist (146).

It is possible that agents are not used in fault tolerant control in particular due to the safety criticality these situations. Multi-agent techniques may be perceived as too flexible and unpredictable for use within such applications. However, in the author's opinion, this is not the main cause. As mentioned within Section 2.5.1, flexibility is not an essential characteristic of multi-agent systems, the degree to which the agency is allowed to adapt may be determined by the design. In addition, the safety critical risks of autonomy are just as present in online synthesis methods, which are more wide-spread. A more likely cause of this lack of application may be the fact that fault tolerance measures are often applied separately from the structuring of the control, and thus it is not obvious that both detection and reconfiguration should be distributed in these cases.

(144) suggests that factors such as the difficulty to meet strict real-time requirements in existing agent systems, the complexity of the control problem decomposition, and the rarity of redundant resources, may be at the root of this insufficient development. This lack of redundancy is not an issue within the HRA, and decomposition into sub-systems is also clear. Hence, the HRA seems suitable for use with multi-agent concepts.

Further rationale for the use of multi-agent concepts in the control of HRA is given in Chapter 7.

2.6 Conclusions

Fault tolerance is of growing importance to automated systems in order to provide safe, reliable and ultimately more cost effective operation. The most effective means of achieving fault tolerance is through systematic analysis of the system and its reliability and fault modes, leading to integration of fault tolerance in the design process.

There are several ways fault tolerance can be achieved, either through the strengthening of components, the inclusion of redundant channels, through fault tolerant control, or through a combination of these methods. Fault tolerant actuation for high integrity systems is usually achieved through redundancy, but the control of these redundant elements is also important.

A great deal of research has been undertaken in the fields of fault tolerant control and fault detection and diagnosis. However, the consolidation of both fault detection and fault tolerant control is not usually considered (30). Fault tolerant control methods often assume perfect information regarding the fault state is available without any delays, and similarly, fault detection schemes often assume that the information they produce can be utilised effectively

by any control reconfiguration scheme. Integration of these two elements is important if a fault tolerant control scheme is to operate correctly.

The HRA concept presents an opportunity for integrating redundancy into system design in order to produce a more reliable and efficient solution. By combining the HRA with a multi-agent inspired active fault tolerant control framework, an integrated fault tolerant control design can also be achieved which is tailored to the structure of the system.

An overview of multi-agent systems has been provided, which has demonstrated that agent concepts can be applied to innumerable applications and take a myriad of forms. The suitability of the agent/agency structures and concepts for use with the HRA will be discussed further in Chapter 7.

Chapter 3

Modelling of High Redundancy Actuation

In order to understand and control complex systems, a good mathematical representation of the system is essential. A model should allow the user to form conclusions and control strategies that would be valid for the real system and as such it should describe and predict the behaviour of the system when it is subjected to external disturbances. Yet it is inevitable that the model will not be a perfect representation of the physical system, due to lack of knowledge of certain behavioural aspects or deliberate simplification. Fortunately, the model must only be sufficiently accurate to achieve its purpose, whilst avoiding unnecessary complexity.

The work presented in this Thesis requires two types of model: one for simulation, and another for control synthesis. These two purposes have differing specifications. A relatively simple model is required for control design, to reduce the complexity of the design process and resultant controller. The simulation model, however, should represent the system over a broader range of operation in order to be test-bed for the control designed using the simplified model.

In addition to these requirements, the models must be sufficiently manipulable in order to represent different HRA configurations and sizes, and various fault modes. To satisfy these requirements the following methodology is taken:

1. Single Element Model: Firstly, a linear time-invariant (LTI) model of a single actuation element is derived from first principles. This model is then experimentally verified and its parameters identified.
2. Model Reduction: This LTI model is then reduced through physical assumptions to produce an element model suitable for control design.
3. Non-linearities: The inclusion of some non-linearities to the element model is made, producing the full-order non-linear single element model to be used for simulation.
4. HRA Modelling: A procedure for creating HRA configuration models which can be applied to any of the single element models is developed.

5. Fault Modelling: The modelling of a number of fault modes is then made, and the impact of the various fault modes on certain HRA configurations discussed.

Sections 3.1 and 3.2 cover the first two stages in the modelling procedure. The first section addresses the modelling of a closed-bobbin moving coil actuator, and the second a cut-bobbin actuator. This is as both have been used at different times throughout the project.

In the early stages of the project the moving coil actuator used was the LAL 30 (147). This actuator is unusual as the coil is wound on a closed-bobbin, which forms a closed-loop around the core. This complicates the electrical and magnetic characteristics of the system, and renders the standard series resistance and inductor representation of the electrical characteristics of a standard cut-coil actuator invalid.

The closed-bobbin moving coil actuator has been modelled previously in (8). However, the model presented there approximates the electrical characteristics with a third order transfer function fitted to experimental data. The model derived here developed so from first principles, maintaining the physical relevance of the parameters. This approach has not been found within the literature. Hence, this model and its subsequent reductions and extensions form an original contribution to knowledge.

In the later stages of the project a more standard cut-bobbin actuator was used for the experimental rig. The modelling of this actuator is also presented in this chapter and will be the model used throughout the subsequent control design chapters.

Section 3.3 covers stage 3 of the modelling procedure by introducing non-linearities that can be applied to either actuator model, and their respective approximations. Subsequently, Section 3.4 addresses stage four, producing a modelling procedure for forming HRAs.

Finally, the fault modelling is presented in Section 3.5. This fault modelling also forms some contribution to knowledge. The use of actuators in series is unusual and thus faults in serial actuators are not commonly covered in the literature, whereas this configuration is a particular feature of the HRA concept. The fault types included here in serial actuation have been previously covered in the work of (7). However, these faults were introduced into an electromechanical system, and thus consideration of these fault types within electromagnetic actuation forms a new contribution to the project.

3.1 Modelling of a Closed-Bobbin Moving Coil Actuator

3.1.1 Operating Principles of a Moving Coil Actuator

The application of moving coil actuators was originally limited to voice coils within loudspeakers and similar devices, but became popular in hard disk drives, servo controlled valves, and mirror position for lasers. The short stroke of moving coil actuators limited their use to applications of this kind, where the travel requirement was low. However, technological advances in magnetic materials have made the production of actuators with longer strokes feasible. This has opened up possibilities for use within many fields. In applications requiring precise position control over a moderate stroke, moving coil actuators have advantages in

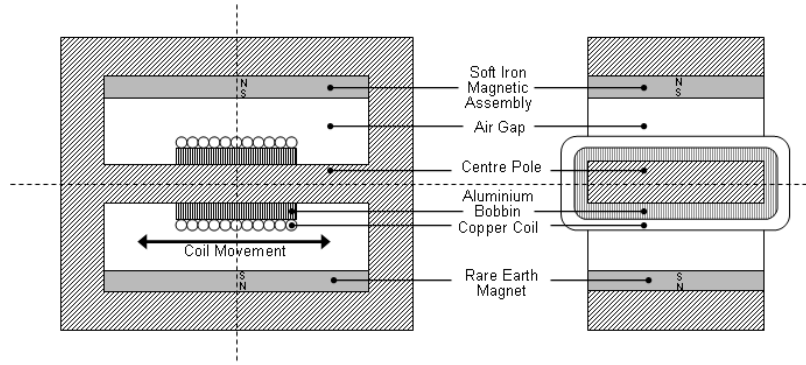


Figure 3.1: Moving coil actuator.

comparison to d.c. and stepper motors, as they do not exhibit the backlash, irregular motion or power loss that result from converting rotary motion to linear motion (148).

A diagram of the moving coil actuator used within these studies can be found in the SMAC data sheet included in Appendix B. Figure 3.1 illustrates the basic components of the actuator. The actuator comprises a moving coil wound round the centre pole of a magnetic assembly that produces a uniform magnetic field perpendicular to the current conducted in the coil. On providing a voltage, a current flows in the coil generating a force which is parallel to the direction of travel. This force causes the coil, and the rod which is mounted on it, to move.

This force is proportional to the current in the coil, the number of turns, and the flux strength.

The copper coil is wound round an aluminium bobbin, which forms part of the piston carriage. This aluminium bobbin surrounds the centre pole of the magnet, forming a circuit, and as such, as it moves within the magnetic field, eddy currents are induced within it. These eddy currents produce magnetic fields that oppose the external magnetic field and thus oppose the movement of the coil causing a damping effect. Eddy currents are also induced within the bobbin by the changing current in the coil, which adds to the damping within the system.

3.1.2 Modelling Methodology

The moving coil actuator is modelled in two stages: firstly the electrical subsystem which characterises the force produced by the electrical input, and then the mechanical subsystem upon which this force acts. The electrical subsystem equations are derived from magnetic principles. These equations are then used to produce an electrical equivalent circuit, using the analogies provided in Table 3.1. This equivalent circuit represents the electrical subsystem, or complete static system. The mechanical subsystem equations are then derived from fundamental equations of motion and finally, the two subsystems are combined using dynamical laws to produce one overall model for the actuation element.

Table 3.1: Analogous electrical and magnetic quantities.

Electrical	Magnetic
Voltage, u	Magnetomotive force, \mathcal{F}
Current, I	Magnetic Flux, Φ
Resistance, R	Reluctance, \mathfrak{R}

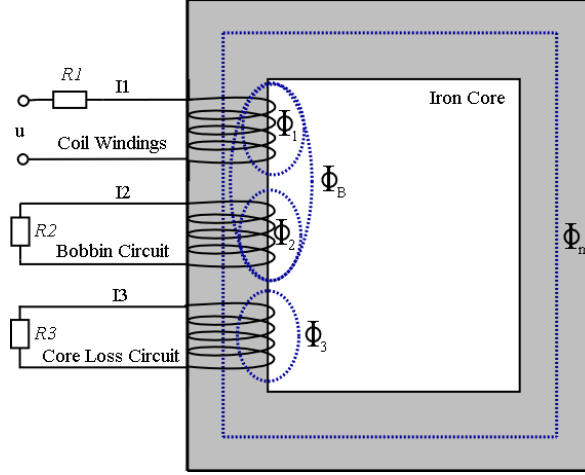


Figure 3.2: Magnetic flux within the static system.

3.1.3 Electrical Subsystem Modelling

Figure 3.2 provides a representation of the flux within the system. The figure shows the iron core with the coil surrounding it. The coil has a voltage input, u and winding resistance R_1 . As the bobbin is effectively a closed turn around the core, it is represented by a second coil within this diagram. The resistance of the bobbin is included as R_2 . The third coil is included to represent the inductive and resistive core losses. The majority of the flux flows in the iron core, and is shown in Figure 3.2 as Φ_M . Φ_1 is the flux linking the coil and Φ_2 is the flux flowing in the bobbin. Φ_b is the flux that links the coil and the bobbin. Finally, the core losses are denoted as Φ_3 .

The magnetomotive force (m.m.f.) that creates this flux is expressed as below, where \mathfrak{R} is the reluctance:

$$\mathcal{F} = \mathfrak{R}\Phi \quad (3.1)$$

The reluctance is analogous to the electrical resistance and is dependant on the dimensions of the core as well as its materials.

Hence, the fluxes denoted in Figure 3.2 can be defined:

$$\mathcal{F}_1 = \mathfrak{R}_1\Phi_1 \quad (3.2)$$

$$\mathcal{F}_2 = \mathfrak{R}_2\Phi_2 \quad (3.3)$$

$$\mathcal{F}_3 = \mathfrak{R}_3\Phi_3 \quad (3.4)$$

$$\mathcal{F}_B = \mathcal{F}_1 + \mathcal{F}_2 = \mathfrak{R}_1\Phi_1 + \mathfrak{R}_2\Phi_2 \quad (3.5)$$

$$\mathcal{F}_1 + \mathcal{F}_2 + \mathcal{F}_3 = \mathfrak{R}_m\Phi_M \quad (3.6)$$

FARADAY's law of electromagnetic induction states that current flowing through the coil not only establishes a magnetic field in the iron, but also creates a voltage across the coil that is proportional to the rate of change of the magnetic flux. This voltage is known as the electromotive force (e.m.f.):

$$E = -N \frac{d\Phi}{dt} \quad (3.7)$$

where N is the number of turns. LENZ's law further states that the induced e.m.f. will oppose the current that created it. Therefore, in the current case, the e.m.f. created by the changing flux is:

$$E_1 = N_1 \frac{d}{dt}(\Phi_M + \Phi_1 + \Phi_B) \quad (3.8)$$

$$E_2 = N_2 \frac{d}{dt}(\Phi_M + \Phi_2 + \Phi_B) \quad (3.9)$$

$$E_3 = N_3 \frac{d}{dt}(\Phi_M + \Phi_3) \quad (3.10)$$

and thus:

$$u = N_1 \frac{d}{dt}(\Phi_M + \Phi_1 + \Phi_B) + R_1 I_1 \quad (3.11)$$

$$0 = N_2 \frac{d}{dt}(\Phi_M + \Phi_2 + \Phi_B) + R_2 I_2 \quad (3.12)$$

$$0 = N_3 \frac{d}{dt}(\Phi_M + \Phi_3) + R_3 I_3 \quad (3.13)$$

and using the m.m.f. law, analogous to AMPERE's law:

$$\Re \Phi = NI \quad (3.14)$$

leads to:

$$\Phi_1 = \frac{N_1 I_1}{\Re_1} \quad (3.15)$$

$$\Phi_2 = \frac{N_2 I_2}{\Re_2} \quad (3.16)$$

$$\Phi_3 = \frac{N_3 I_3}{\Re_3} \quad (3.17)$$

$$\Phi_M = \frac{N_M I_M}{\Re_M} = \frac{N_M (I_1 + I_2 + I_3)}{\Re_M} \quad (3.18)$$

$$\Phi_B = \frac{N_B I_B}{\Re_B} = \frac{N_B (I_1 + I_2)}{\Re_B} \quad (3.19)$$

Thus substituting Equations 3.15-3.19 into Equations 3.11-3.13:

$$u = N_1 \left(\frac{N_M}{\mathfrak{R}_M} \frac{d}{dt} (I_1 + I_2 + I_3) + \frac{N_1}{\mathfrak{R}_1} \frac{dI_1}{dt} + \frac{N_B}{\mathfrak{R}_B} \frac{d}{dt} (I_1 + I_2) \right) + R_1 I_1 \quad (3.20)$$

$$0 = N_2 \left(\frac{N_M}{\mathfrak{R}_M} \frac{d}{dt} (I_1 + I_2 + I_3) + \frac{N_2}{\mathfrak{R}_2} \frac{dI_2}{dt} + \frac{N_B}{\mathfrak{R}_B} \frac{d}{dt} (I_1 + I_2) \right) + R_2 I_2 \quad (3.21)$$

$$0 = N_3 \left(\frac{N_M}{\mathfrak{R}_M} \frac{d}{dt} (I_1 + I_2 + I_3) + \frac{N_3}{\mathfrak{R}_3} \frac{dI_3}{dt} \right) + R_3 I_3 \quad (3.22)$$

and finally the electrical subsystem can be expressed in electrical terms:

$$u = N_1 \left(L_m \frac{d}{dt} (I_1 + I_2 + I_3) + L_1 \frac{dI_1}{dt} + L_B \frac{d}{dt} (I_1 + I_2) \right) + R_1 I_1 \quad (3.23)$$

$$0 = N_2 \left(L_m \frac{d}{dt} (I_1 + I_2 + I_3) + L_2 \frac{dI_2}{dt} + L_B \frac{d}{dt} (I_1 + I_2) \right) + R_2 I_2 \quad (3.24)$$

$$0 = N_3 \left(L_m \frac{d}{dt} (I_1 + I_2 + I_3) + L_3 \frac{dI_3}{dt} \right) + R_3 I_3 \quad (3.25)$$

where:

$$L_m = \frac{N_m}{\mathfrak{R}_m}, L_1 = \frac{N_1}{\mathfrak{R}_1}, L_2 = \frac{N_2}{\mathfrak{R}_2}, L_3 = \frac{N_3}{\mathfrak{R}_3} \text{ and } L_B = \frac{N_B}{\mathfrak{R}_B}$$

These equations describe the actuation element without mechanical movement i.e. when the bobbin is clamped. Hence, the mode represents only the electrical subsystem. From these equations the equivalent circuit shown in Figure 3.3 may be derived.

Some simplifications may be made as L_2 and L_3 are much smaller than L_m and L_B , and thus they may be removed with little affect on the system (149). Using KIRCHHOFF'S current and voltage laws on this simplified circuit (Figure 3.4), the following transfer function and state space model can be derived to describe the electrical subsystem:

$$\frac{I_{R1}}{u_{in}} = \frac{L_B L_m s^2 + (L_B(R_1 + R_2) + L_m(R_2 + R_3))s + R_2 R_3}{L_B L_m L_1 s^3 + c_1 s^2 + c_2 s + R_1 R_2 R_3} \quad (3.26)$$

where

$$c_1 = (L_m(L_B(R_1 + R_2) + L_1(R_2 + R_3)) + L_B L_1 R_3)$$

$$c_2 = (R_2(L_m R_1 + L_1 R_3) + R_3(R_1 + R_2)(L_B + L_m))$$

$$\begin{bmatrix} \dot{I}_{R1} \\ \dot{I}_{LB} \\ \dot{I}_{Lm} \end{bmatrix} = \begin{bmatrix} \frac{-(R_1 + R_2)}{L_1} & \frac{R_2}{L_1} & 0 \\ \frac{R_2}{L_B} & -\frac{(R_2 + R_3)}{L_B} & \frac{R_3}{L_B} \\ 0 & \frac{R_3}{L_M} & -\frac{R_3}{L_M} \end{bmatrix} \bullet \begin{bmatrix} I_{R1} \\ I_{LB} \\ I_{Lm} \end{bmatrix} + \begin{bmatrix} \frac{1}{L_1} \\ 0 \\ 0 \end{bmatrix} \bullet u \quad (3.27)$$

where: I_{Lm} is the current flowing through inductor L_m and is equal to the sum of I_1 , I_2 and I_3 ; I_{LB} is the current flowing through inductor L_B and is equal to the sum of I_1 and I_2 ; and I_{R1} is the current flowing in the winding resistance R_1 and is equal to I_1 .

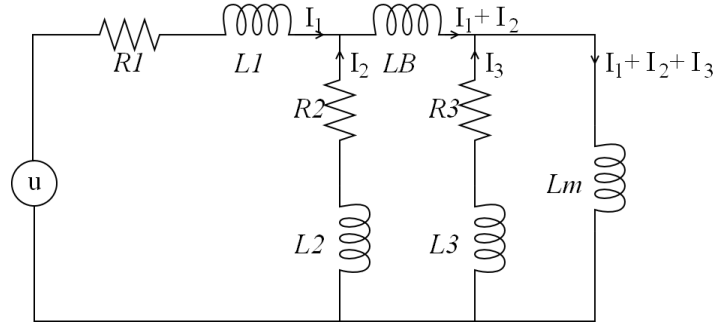


Figure 3.3: Electrical subsystem equivalent circuit.

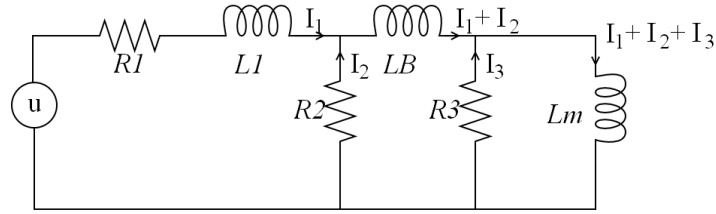


Figure 3.4: Simplified electrical subsystem equivalent circuit.

3.1.4 Mechanical Subsystem Modelling

The mechanical subsystem is a typical second order system consisting of the moving mass of the element and any stiffness and damping within the system with an input force originating from the electrical subsystem. A diagram of the mechanical subsystem alongside its free-body diagram is given in Figure 3.5. Using NEWTON's Law the mechanical subsystem can be described by the equation of motion given in equation (3.29).

$$F_T = m\ddot{x} \quad (3.28)$$

therefore:

$$\ddot{x} = \frac{1}{m}F_e - \frac{d}{m}\dot{x} - \frac{r}{m}x \quad (3.29)$$

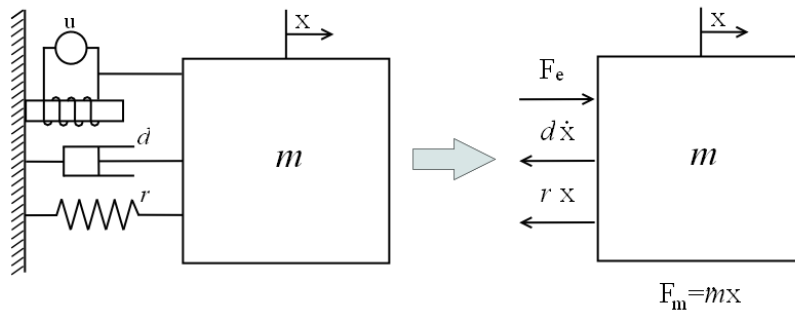


Figure 3.5: Mechanical subsystem and its free-body diagram.

3.1.5 Full Model

Having established the electrical and mechanical subsystems, the full model can now be created by considering the dynamics of the system. There are two equations that describe the flow of force between the two subsystems: the LORENTZ force law and FARADAY's law of induction.

The current flowing perpendicular to the flux density results in a force known as the LORENTZ force:

$$F_e = BNlI \quad (3.30)$$

This force moves both the coil and the bobbin, therefore a force is generated by both I_1 and I_2 as they are the currents associated with the coil and bobbin:

$$F_e = BNlI_1 + BNlI_2 \quad (3.31)$$

$$= k(I_1 + I_2) \quad (3.32)$$

$$= kI_{LB} \quad (3.33)$$

The magnetic flux density, B , is assumed to be constant over the travel of the coil/bobbin. The number of turns, N and the conductor turn length, l are also constant and so BNl may be combined to produce one force constant k . This force is the input to the mechanical subsystem.

As the coil and bobbin are allowed to move in the field, their movement will generate electromotive forces within their circuits in accordance with FARADAY's law. LENZ's law states that this electromotive force will oppose the movement, and hence it is known as the counter-electromotive force and is expressed as below:

$$E = BNl\dot{x} \quad (3.34)$$

$$= k\dot{x} \quad (3.35)$$

The derivative \dot{x} is the perpendicular component of the velocity of the wire relative to the flux lines. BNl is again constant and can be replaced by the force constant k . As I_1 and I_2 are the currents associated with the moving coil and bobbin, E acts as a voltage in the loop containing L_B , and as such features in the expression for \dot{I}_{LB} . Hence, the state-space expression for the full system is stated below:

$$\begin{bmatrix} \dot{I}_{R1} \\ \dot{I}_{LB} \\ \dot{I}_{Lm} \\ \ddot{x} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} -\frac{(R_1+R_2)}{L_1} & \frac{R_2}{L_1} & 0 & 0 & 0 \\ \frac{R_2}{L_B} & -\frac{(R_2+R_3)}{L_B} & \frac{R_3}{L_B} & -\frac{k}{L_B} & 0 \\ 0 & \frac{R_3}{L_m} & -\frac{R_3}{L_m} & 0 & 0 \\ 0 & \frac{k}{m} & 0 & -\frac{d}{m} & -\frac{r}{m} \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \bullet \begin{bmatrix} I_{R1} \\ I_{LB} \\ I_{Lm} \\ \dot{x} \\ x \end{bmatrix} + \begin{bmatrix} \frac{1}{L_1} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \bullet u_{in} \quad (3.36)$$

A block diagram for the moving-coil actuator model is shown in Figure 3.6.

3.1.6 Parameter Identification

Experiments were performed on a single actuation element in order to determine the parameter values and verify the model. The specifications provided for the LAL30 SMAC give the moving mass and force constant (Table 3.2 (147)). However, these parameters were checked experimentally for completeness. The experiment setup, methodology and parameter identification results are discussed within the following subsections.

Table 3.2: SMAC LAL30 moving coil actuator specifications.

Specifications	
Stroke	15 mm
Peak Force	14.5 N
Force Constant	7.4 N/A
Moving Mass	0.15 kg

3.1.6.1 Experiment Setup

The experiment is setup as shown in Figure 3.7. xPC Target alongside Matlab/Simulink is used to provide a real-time environment. xPC Target is a host-target prototyping environment that enables the connection of Simulink models to physical systems and real-time execution on PC compatible hardware (150). Two PCs, a host and target connected via a TCP/IP network are used within the setup. Matlab/Simulink runs on the host PC, where the application is designed in Simulink using xPC target I/O blocks. This Simulink file is then built and compiled within the host computer and downloaded to the target PC where it is executed using the xPC Target real-time kernel. The target PC sends and acquires signals according to the executable from the PCI cards, which are the I/O interface between the target and the experiment hardware. The two PCI cards used in this experiment are:

- NI6024E which has 16 analogue I/Os.
- NI6602 which has 8 up/down counters.

The command signal is output from the NI6024E, amplified and then applied to the actuation element. Four signals are measured from the element: the position and acceleration of its end-effector, the coil current and the amplifier output. The position is measured via an optical glass scale encoder, which is integral in the SMAC actuator and is connected to the NI6602. The acceleration is acquired using an accelerometer (ADXL311) mounted on the end of the rod (a data sheet for this sensor is included in Appendix B). It has an analogue output and as such, is routed to the NI6024E. Coil current is obtained from an output pin on the actuator, which determines the current using a shunt resistor, is connected to the analogue PCI card. The amplifier output is also monitored in order to decouple any amplifier dynamics from the other measurements in the post-experiment analysis.

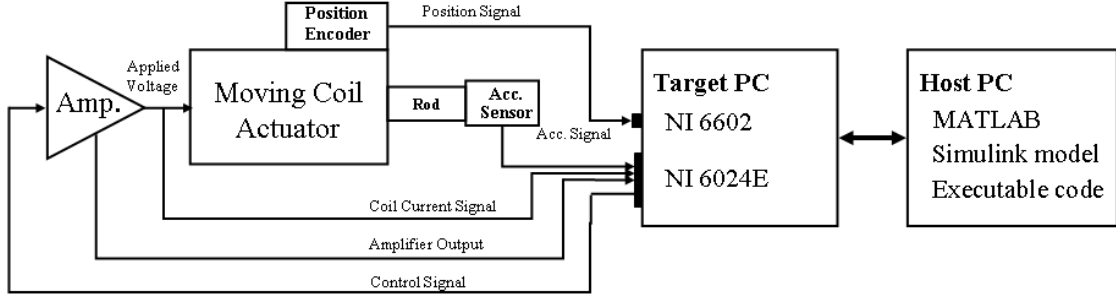


Figure 3.7: Experiment setup.

3.1.6.2 Frequency Response

Frequency responses for the four measured signals were obtained over the frequency range 1-2000 Hz with the coil free-moving. A frequency sweep for coil current and amplifier output was also carried out with the coil clamped mid-way along its travel. This aids the identification process as clamping the coil removes the mechanical dynamics from the system.

In Simulink, a sinusoidal input was applied to the element, and its frequency was manually varied. The frequency response from input to each of the measured quantities was determined by correlating the output signal with the input sinusoid and a cosine of the frequency. In mathematical terms, at frequency point i the input signal is:

$$u_i(t) = U_i \sin(\omega_i t) \quad (3.37)$$

And from the corresponding steady-state response, $y_i(t)$ the following integrals are calculated:

$$S_i = \int_0^{T_e} y_{is}(t) \sin(\omega_i t) dt \quad (3.38)$$

$$C_i = \int_0^{T_e} y_{is}(t) \cos(\omega_i t) dt \quad (3.39)$$

where T_e is the duration of the experiment. Having obtained the integrals, the gain function is now given by:

$$A(\omega_i) = \frac{2}{U_i T_e} \sqrt{S_i^2 + C_i^2} \quad (3.40)$$

and the phase given by:

$$\phi(\omega_i) = \arctan \frac{C_i}{S_i} \quad (3.41)$$

The experiment data (along with offset corrections) is included in Appendix B. The free-moving coil frequency response results are plotted in Figures 3.8 to 3.10. Figure 3.10 also gives the clamped voltage-coil-current response.

The position and acceleration responses are sensible, up to approximately 200Hz, after which unmodelled dynamics appear, most likely caused by mechanical resonances. This is

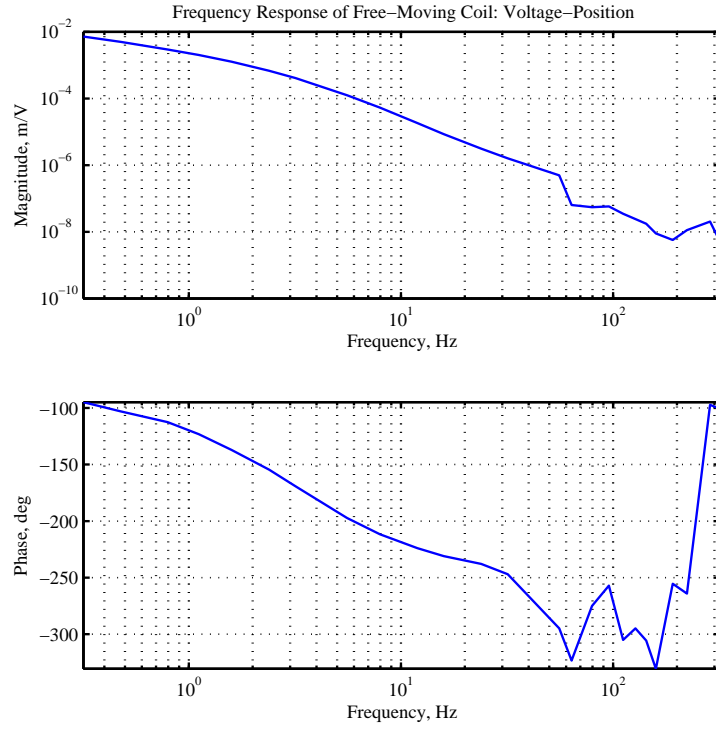


Figure 3.8: Frequency response of free-moving coil: voltage-position.

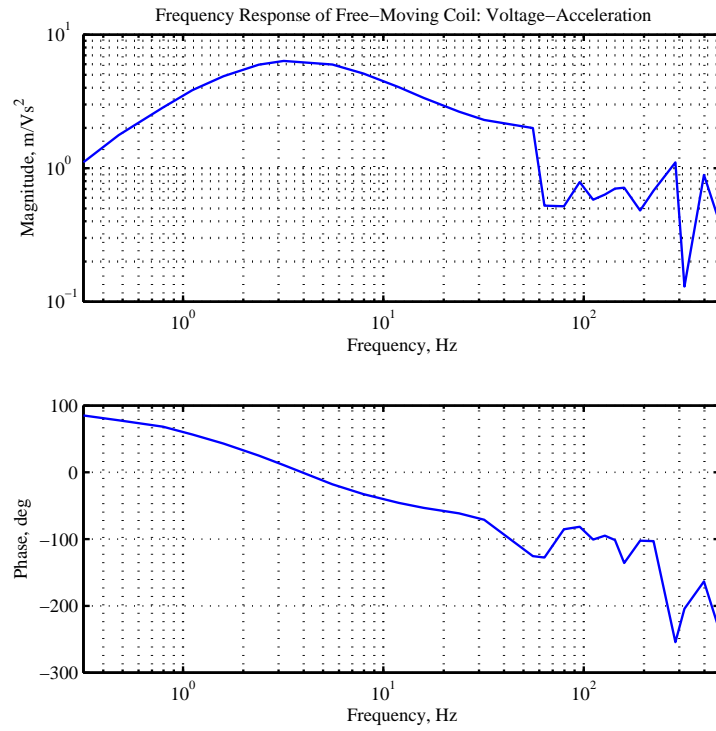


Figure 3.9: Frequency response of free-moving coil: voltage-acceleration.

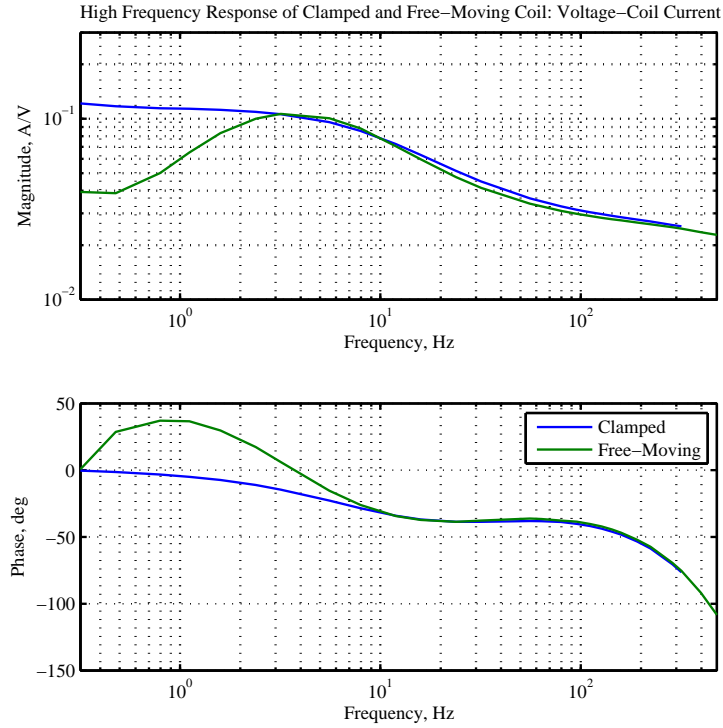


Figure 3.10: Frequency response of clamped and free-moving coil: voltage-current.

not a problem, however, as these dynamics are well above the frequency region used in typical applications.

3.1.6.3 Parameter Fitting

Within the identification process only two parameters could be measured directly: the moving mass, m and the force constant, k . The moving mass was weighed and found to be 0.130 kg. To measure the force constant, a known current was applied to the element and its force measured using scales. The value of k was found to be 7.76 N/A. These values are very close to the stated values of Table 3.2.

The remaining parameters were found by fitting the model to the frequency response data using the optimisation toolbox. The frequency data was entered into Matlab and weights were applied to favour the magnitude response and the 10–100 Hz region. Lower weights were used to remove the influence of the high frequency regions in the position/acceleration responses. Known model values were then set and the remaining parameters defined as parameters to be determined. The model response is then matched to the measured data by defining the difference between them as a scalar function, and using the Matlab function ‘fsolve’ to find a minimum of the function through variation of the parameters, starting from an initial estimate using a non-linear least squares approach. Details of this approach and the weights used can be found in Appendix B where an example Matlab code is provided.

The clamped frequency response was used first in the fitting process, as this system has the fewest parameters. The clamped system transfer function was stated in equation (3.26).

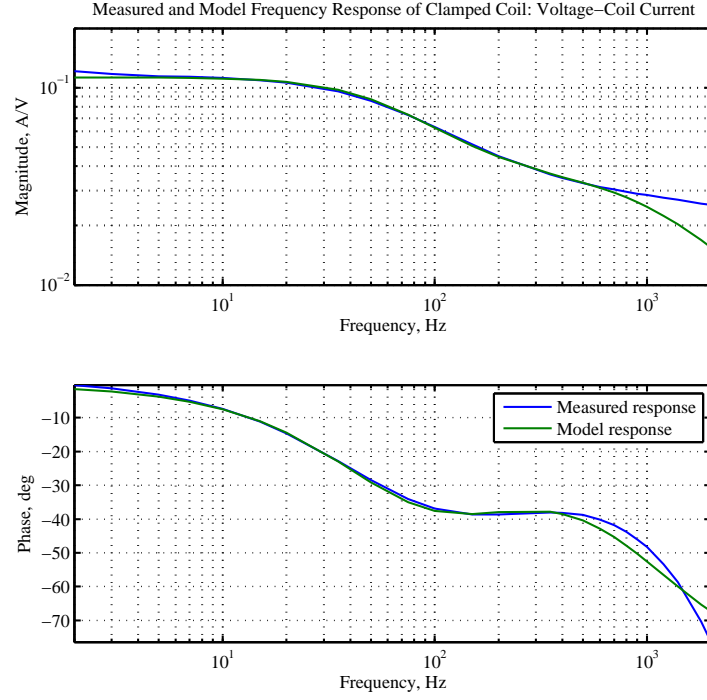


Figure 3.11: Model and experimental frequency response of clamped coil: voltage-current.

The optimisation results suggested that the inductor L_B , symbolising the flux linking the bobbin and the coil, was negligible. Thus, L_B was removed from the model, simplifying it by a degree. The new clamped subsystem model is described by the transfer function in equation 3.42 and the state space expression in 3.43.

$$\frac{I_{R1}}{u_{in}} = \frac{L_m s + R_{2|3}}{L_m L_1 s^2 + (L_m(R_{2|3} + R_1) + L_1 R_{2|3})s + R_1 R_{2|3}} \quad (3.42)$$

where:

$$R_{2|3} = \frac{R_2 R_3}{(R_2 + R_3)}$$

$$\begin{bmatrix} \dot{I}_{R1} \\ \dot{I}_{Lm} \end{bmatrix} = \begin{bmatrix} -\frac{(R_1 + R_{2|3})}{L_1} & \frac{R_{2|3}}{L_1} \\ \frac{R_4}{L_m} & -\frac{R_4}{L_m} \end{bmatrix} \bullet \begin{bmatrix} I_{R1} \\ I_{Lm} \end{bmatrix} + \begin{bmatrix} \frac{1}{L_1} \\ 0 \end{bmatrix} \bullet u \quad (3.43)$$

The parameters L_m , L_1 , R_1 and $R_{2|3}$ were determined from this response. The values of R_2 and R_3 could not be separated at this stage. The determined parameter values are provided in Table 3.3 and the clamped model frequency response is plotted with the experiment data in Figure 3.11.

The free-moving current and position responses were then used to determine the remaining model parameters. The full system state-space model with L_B removed is provided below for

Table 3.3: Parameter values.

Parameter	Value
m	0.13 kg
k	7.76 N/A
L_m	14.14 mH
L_1	4.63 mH
R_1	8.87 Ω
$R_{2 3}$	16.08 Ω
R_2	27.10 Ω
R_3	39.50 Ω
r	not measurable
d	not measurable

reference.

$$\begin{bmatrix} \dot{I}_{R1} \\ \dot{I}_{Lm} \\ \ddot{x} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} -\frac{R_2(R_{2|3}-R_3)-R_1R_3}{L_1R_3} & \frac{R_2(R_3-R_{2|3})}{L_1R_3} & -\frac{kR_{2|3}}{L_1R_3} & 0 \\ \frac{R_2(R_3-R_{2|3})}{L_mR_3} & \frac{R_2(R_{2|3}-R_3)}{L_mR_3} & -\frac{kR_{2|3}}{L_1R_2} & 0 \\ \frac{L_mR_3}{kR_{2|3}} & \frac{L_mR_3}{k(R_3-R_{2|3})} & -\frac{(k^2+d(R_2+R_3))}{m(R_2+R_3)} & -\frac{r}{m} \\ 0 & 0 & 1 & 0 \end{bmatrix} \bullet \begin{bmatrix} I_{R1} \\ I_{Lm} \\ \dot{x} \\ x \end{bmatrix} + \begin{bmatrix} \frac{1}{L_1} \\ 0 \\ 0 \\ 0 \end{bmatrix} \bullet u \quad (3.44)$$

The remaining model parameters determined from these responses are shown in Table 3.3. The ratio between R_2 and R_3 was determined, allowing values for each to be found. The model shows some sensitivity to the distribution between R_2 and R_3 . However, the chosen values provide the best fit. The mechanical parameters r and d were found to be negligible within the optimisation and thus are set at a small value. The frequency responses of the model and measured data for the free-moving system are shown in Figures 3.12 to 3.14.

It can be seen from Table 3.3 that the fitted parameters are of reasonable magnitude. The model provides a good fit to the measured data between 5 – 100 Hz. These discrepancies are present both in the accelerometer and encoder readings, and as such may not be directly attributed to either. The operational temperature and frequencies should be well within the linear limits suggested by the accelerometer documentation. The discrepancies at higher frequencies can be attributed to unmodelled mechanical resonances. Changes in resistance due to thermal effects may have some influence on the current measurements, particularly at high frequency. There may also be some skin effects present in the high frequencies, which could be modelled. However, this would increase the model order significantly, and as these effects are outside the normal operating region, their inclusion in the model is not considered beneficial. In the acceleration frequency response, a discrepancy at low frequencies (2 – 5 Hz) can be observed. This difference is attributable to stiction.

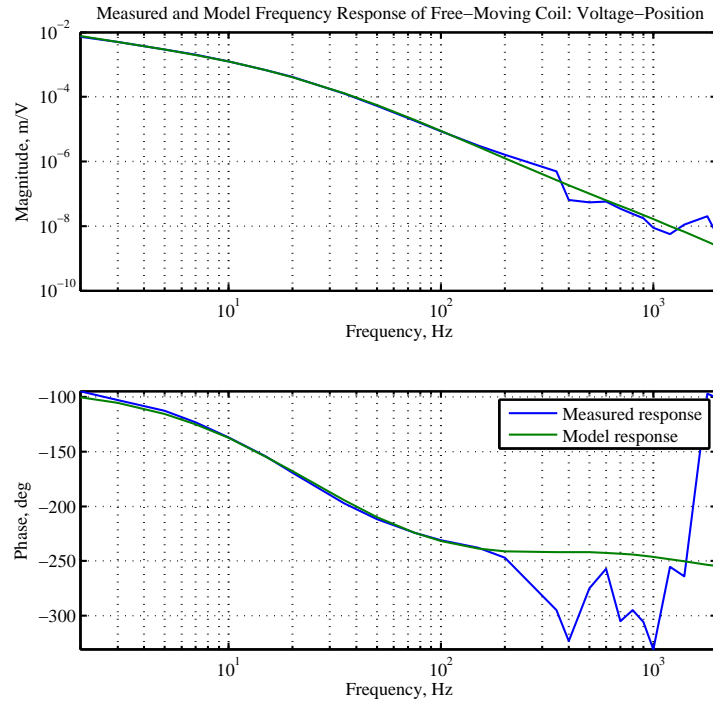


Figure 3.12: Model and experimental frequency response of free-moving coil: voltage-position.

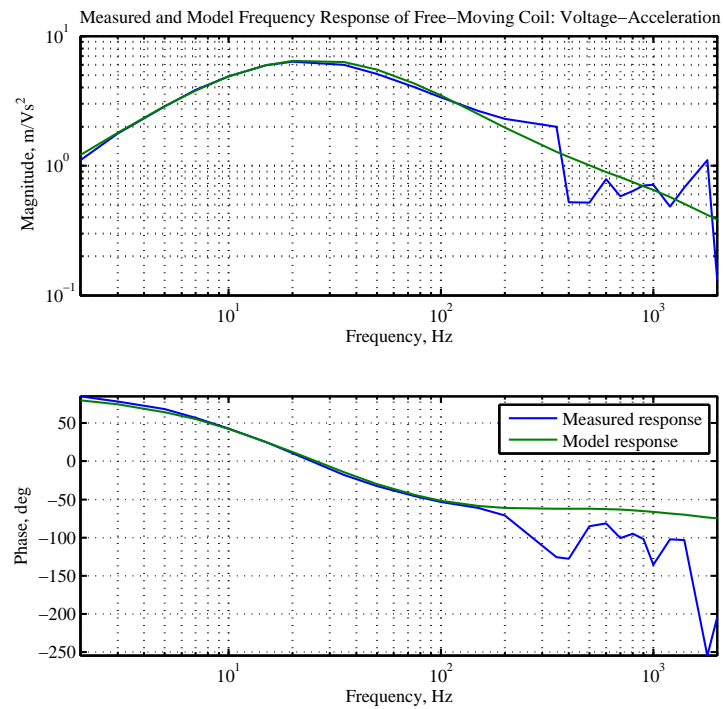


Figure 3.13: Model and experimental frequency response of free-moving coil: voltage-acceleration.

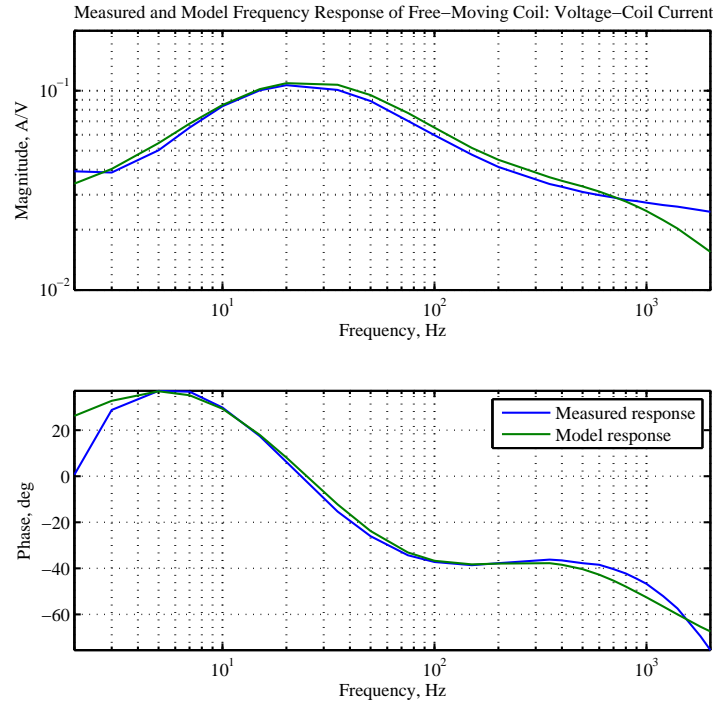


Figure 3.14: Model and experimental frequency response of free-moving coil: voltage-current.

3.1.7 Model Reduction

The model derived and verified in the previous sections is a good linear approximation to the actual moving-coil actuator. The intended use of this model is as an element within multi-element configurations which form the HRA and as this element model is fourth order, the resultant HRA model will be some multiple of this. Figure 3.15 shows the order of various example HRA configurations using the 4th order element. These HRA models are of high order, making them cumbersome for control design purposes. The figure also shows the final model size that could be obtained using a reduced order element. The HRA model sizes using a 2nd order model are much more manageable for control design. Hence, it would be desirable to use a reduced order element that suitably approximates the behaviour of the original system in the design of the control. The control laws created may then be tested on the full order system. This section will detail the reduction of the element model to a 3rd and 2nd order model for this purpose¹.

There are two main approaches to model reduction: analytical reduction and physical reduction. The former is where states are removed or residualised from the original model according to some defined analytical procedure, such as balanced reductions. Analytical reduction is useful when the model is large, and there are no obvious physical assumptions

¹Another option for obtaining a more manageably sized model would be to reduce the HRA configuration model, rather than the single element. This approach has a few drawbacks however, as the reduction process would have to be carried out for each HRA configuration model required and their respective fault modes. Also, this approach is only suitable for centralised design, whereas the main control approach presented here is decentralised, and as such requires the single element model.

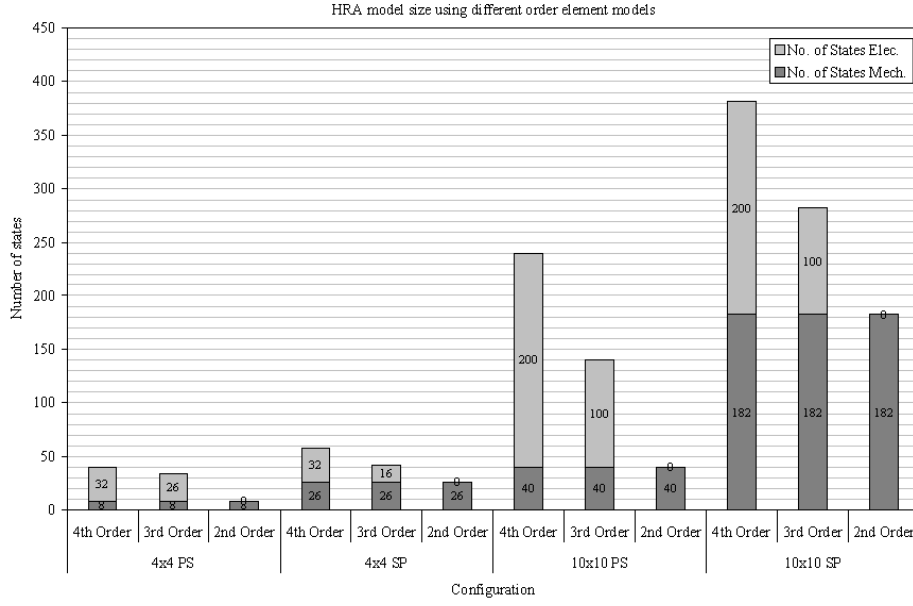


Figure 3.15: HRA model sizes for element models of varying order.

that can be made.

In contrast, a physical model reduction is where the model is reduced through practical knowledge of the actual system. Due to the vastly varied nature of systems, no defined procedure exists to conduct a physical model reduction and as such any results will be as good as the assumptions made within the reduction. However, reduction based on physical knowledge is desirable, as in contrast to analytical procedures, the connection of the model to the physical system and its parameters is maintained. This is important to understanding how parameter or structural changes that occur affect the system, which is crucial for fault tolerant control.

A physical reduction approach is taken here in order to maintain the physical significance of the model. Also, the nature of the model lends itself to this form of reduction. As was illustrated in Figure 3.15, a reduction specifically in the electrical subsystem would be advantageous as this contributes the majority of the states, and simplifications in electrical circuits are common and straightforward. Indeed, a physical reduction from 5th order to 4th has already been made by removing L_B .

In the following subsections the reduction of the model to 3rd and 2nd order is detailed.

3.1.7.1 Third Order Element Model

A 3rd order element model will contain only one electrical state, hence one state must be removed from the electrical subsystem of equation 3.43. The most obvious component to remove from the equivalent circuit is the inductance L_1 , as this constitutes the fastest pole. By

removing this inductor the third order system, G_3 may be expressed as shown in equation 3.45.

$$G_3 : \begin{bmatrix} \ddot{x} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} -\frac{R_1 R_2 R_3}{L_m R_{123}} & \frac{k(R_1 R_2 - R_{123})}{L_m R_{123}} & 0 \\ \frac{k(R_{123} - R_1 R_2)}{m R_{123}} & \frac{k^2(R_1 R_2 - R_{123}) - d R_3 R_{123}}{m R_3 R_{123}} & -\frac{r}{m} \\ 0 & 1 & 0 \end{bmatrix} \bullet \begin{bmatrix} \dot{x} \\ x \end{bmatrix} + \begin{bmatrix} \frac{R_2 R_3}{L_m R_{123}} \\ \frac{k R_2}{m R_{123}} \\ 0 \end{bmatrix} u \quad (3.45)$$

where:

$$R_{123} = R_1 R_2 + R_3(R_1 + R_2) \quad (3.46)$$

3.1.7.2 Second Order Element Model

To reduce the model to second order, the final remaining inductance, L_m must be removed. As this component is in parallel with R_3 , it is necessary to combine their impedances at a certain frequency to simplify the circuit. Their combined impedance Z is:

$$Z = \frac{R_3 \sqrt{(j\omega L_m)^2}}{R_3 + \sqrt{(j\omega L_m)^2}} \quad (3.47)$$

If ω is set to zero, to provide the lowest steady-state error then:

$$Z = \frac{R_3}{R_3 + 1} \quad (3.48)$$

Using this combined impedance in place of R_3 and L_m , the second order model, G_2 can be expressed as below:

$$G_2 : \begin{bmatrix} \ddot{x} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} -\frac{(k^2(R_1 + R_2) - d R_{12Z})}{m R_{12Z}} & -\frac{r}{m} \\ 1 & 0 \end{bmatrix} \bullet \begin{bmatrix} \dot{x} \\ x \end{bmatrix} + \begin{bmatrix} \frac{k R_2}{m R_{12Z}} \\ 0 \end{bmatrix} \bullet u \quad (3.49)$$

where:

$$R_{12Z} = R_1 R_2 + Z(R_1 + R_2) \quad (3.50)$$

3.1.7.3 Model Reduction Quality

The two reduced order models are stable and fully controllable and observable. Figure 3.16 shows the frequency response of the original 4th order system and the 3rd and 2nd order reduced models, and Figure 3.17 the singular values of the additive error between them. It can be observed that the approximations are good within the low-mid frequency region. The high frequency region is less well approximated as the models were reduced through the removal of the high frequency inductors. This is acceptable however, given the original model's uncertainty within this region. The time domain response of the two approximations is also adequate, as shown in Figure 3.18.

Overall, the results indicate that the second order approximation provides an adequately accurate model for the purpose of control design.

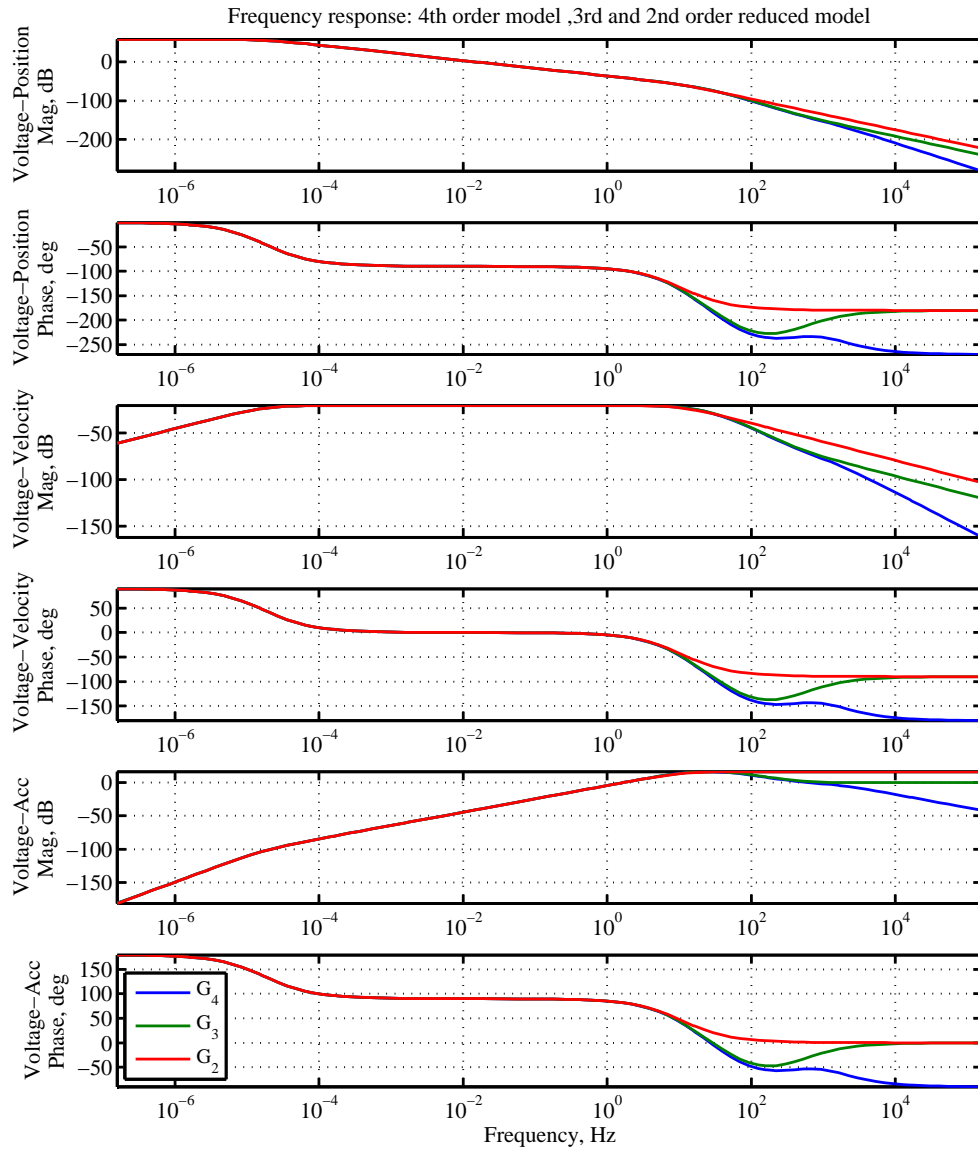


Figure 3.16: Bode diagrams of the 4th, 3rd and 2nd order models.

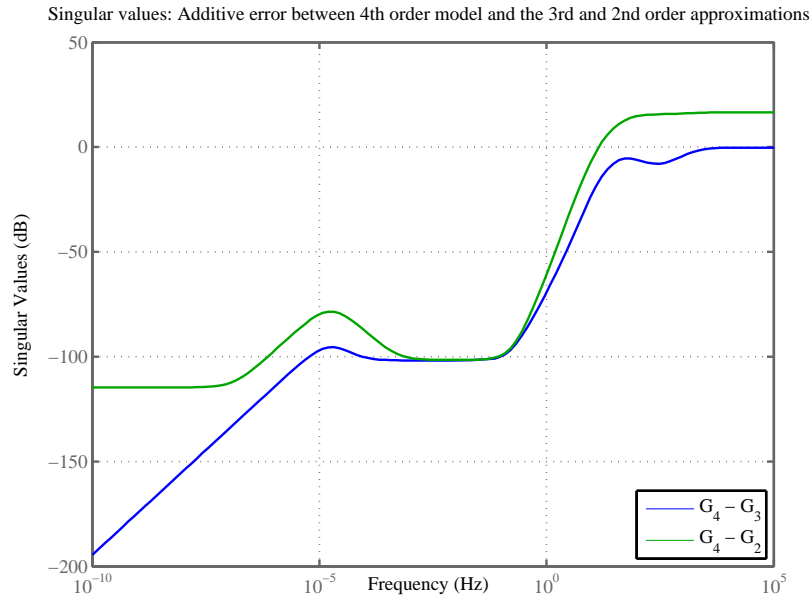


Figure 3.17: Additive error between the 4th, 3rd and 2nd order models.

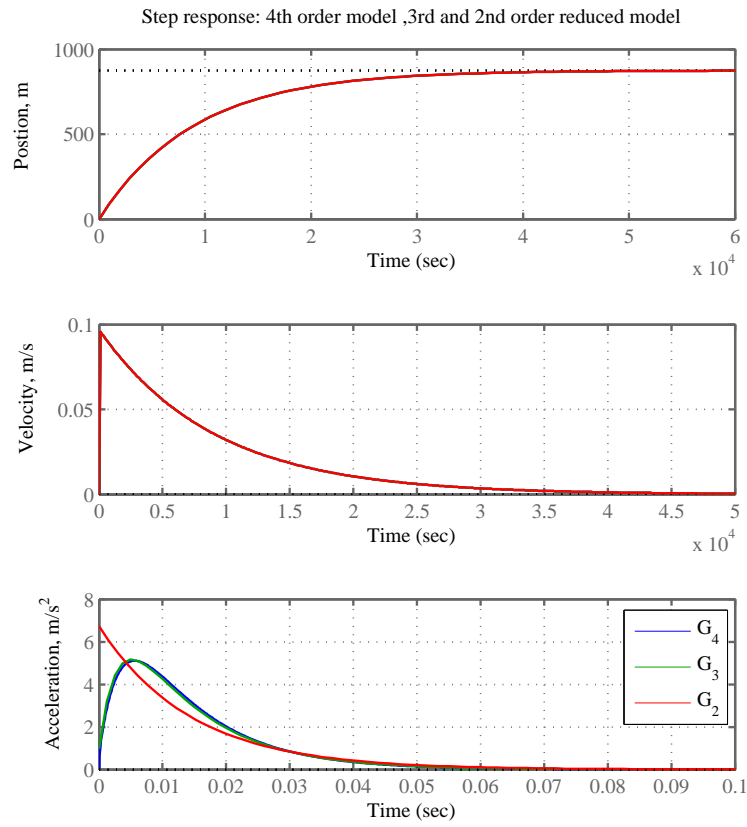


Figure 3.18: Step response of the 4th, 3rd and 2nd order models.

3.2 Modelling of a Cut-Bobbin Moving Coil Actuator

Cutting the bobbin of a moving coil actuator simplifies the electrical dynamics significantly, as no eddy currents are induced in the bobbin.

A model for this system can be derived from the closed-bobbin model that was presented in Section 3.1, by simplifying the electrical circuit. The mechanical subsystem remains unaltered, however, the components associated with the bobbin (R_2 and L_B) in the electrical subsystem equivalent circuit (Figure 3.4) must be removed. Therefore, the circuit shown in Figure 3.19 represents the cut-bobbin electrical subsystem.

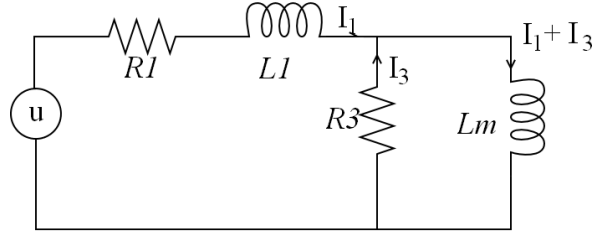


Figure 3.19: Cut-bobbin electrical subsystem equivalent circuit.

This circuit, in combination with the mechanical subsystem, results in the following system model:

$$\begin{bmatrix} \dot{I}_{R1} \\ \dot{I}_{Lm} \\ \ddot{x} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} -\frac{R_1+R_3}{L_1} & \frac{R_3}{L_1} & -\frac{k}{L_1} & 0 \\ \frac{R_3}{L_m} & -\frac{R_3}{L_m} & 0 & 0 \\ \frac{k}{m} & 0 & -\frac{d}{m} & -\frac{r}{m} \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} I_{R1} \\ I_{Lm} \\ \dot{x} \\ x \end{bmatrix} + \begin{bmatrix} \frac{1}{L_1} \\ 0 \\ 0 \\ 0 \end{bmatrix} \bullet u \quad (3.51)$$

3.2.1 Model Reduction

By taking the same approach as described in Section 3.1.7, the model can be reduced to provide a model more suitable for control design of HRA. It may be reduced to third order by removing the inductance L_1 , and to second order by subsequently removing L_m and R_3^2 . This produces the reduced order models expressed in equations 3.52 and 3.53.

$$G_{cut3} : \begin{bmatrix} \dot{I}_{Lm} \\ \ddot{x} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} -\frac{R_{1|3}}{L_m} & \frac{kR_{1|3}}{L_m R_1} & 0 \\ \frac{kR_{1|3}}{m R_1} & \frac{k^2 - d(R_1 + R_3)}{m(R_1 + R_3)} & -\frac{r}{m} \\ 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} I_{Lm} \\ \dot{x} \\ x \end{bmatrix} + \begin{bmatrix} \frac{R_{1|3}}{L_m R_1} \\ 0 \\ 0 \end{bmatrix} u \quad (3.52)$$

$$G_{cut2} : \begin{bmatrix} \ddot{x} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} -\frac{k^2 + dR_1}{m R_1} & -\frac{r}{m} \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \dot{x} \\ x \end{bmatrix} + \begin{bmatrix} \frac{k}{m R_1} \\ 0 \end{bmatrix} \bullet u \quad (3.53)$$

The frequency responses of these approximations, alongside the full-order model, are shown in Figure 3.20 and the additive error between them is shown in Figure 3.21. It can be seen that the approximations are sufficiently accurate within the low and mid frequency regions.

² R_3 is also removed as the combined impedance of these elements needs to be very small in order to achieve a good approximation, as was the case in Section 3.1.7.

The step responses for a unit voltage input are shown in Figure 3.22 also suggest that the reduced order models are adequate for control design use.

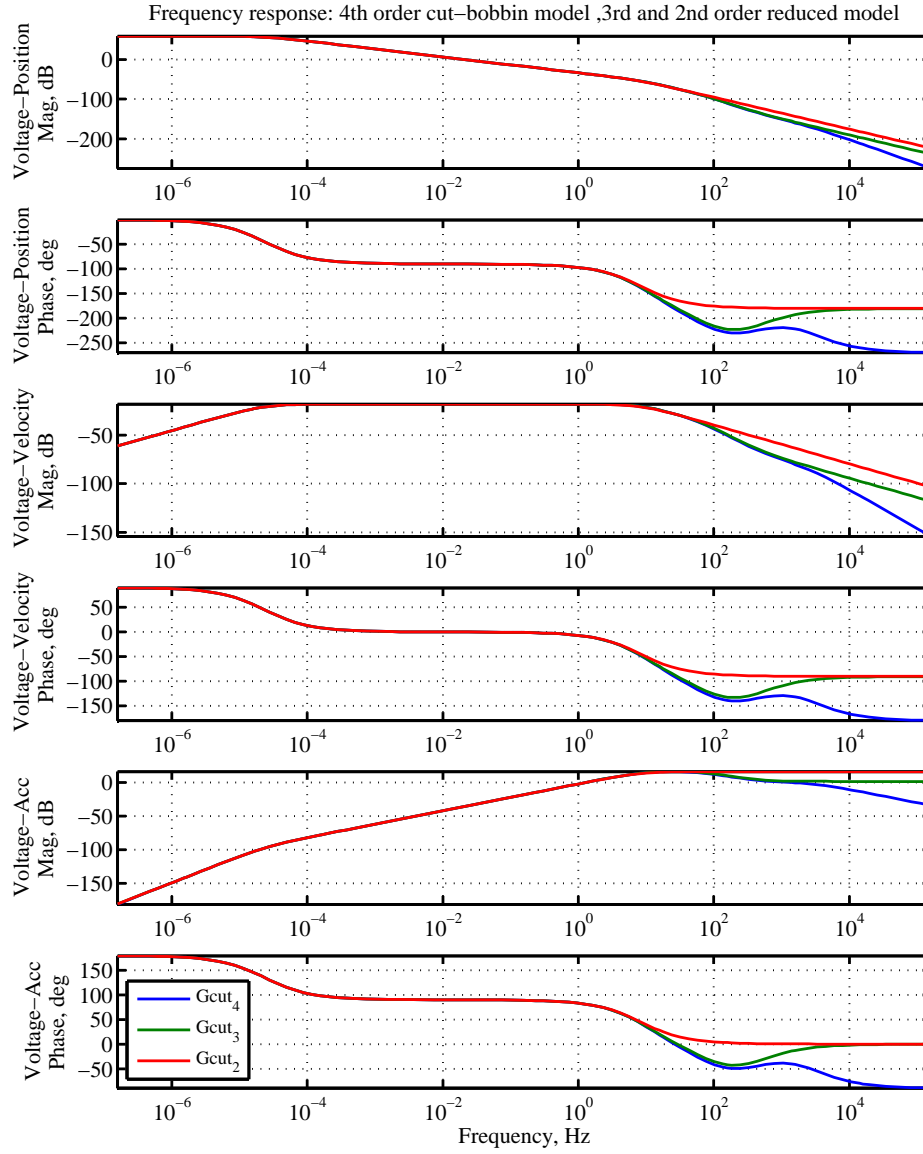


Figure 3.20: Bode diagrams of the 4th, 3rd and 2nd order cut bobbin models.

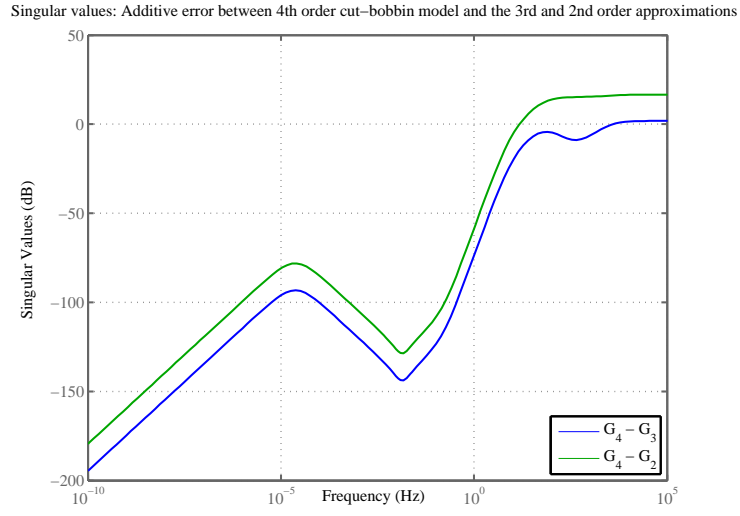


Figure 3.21: Bode diagrams of the 4th, 3rd and 2nd order models.

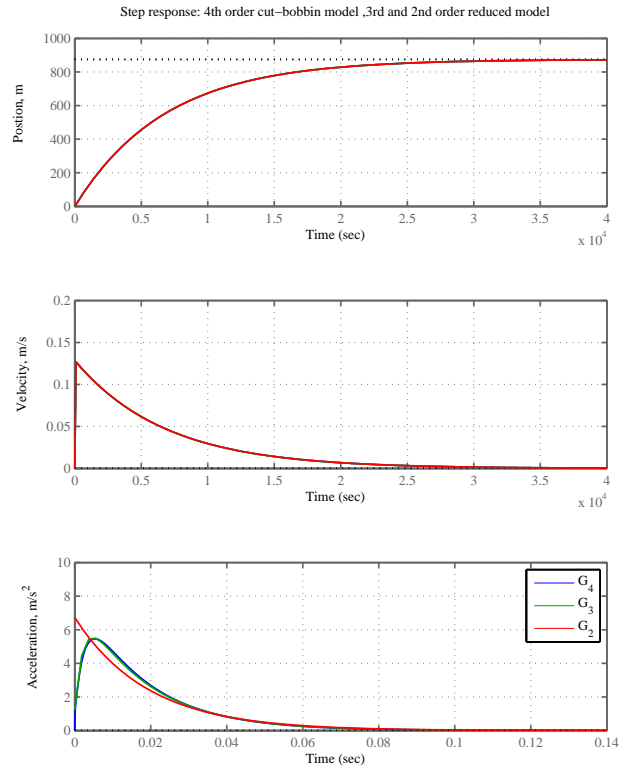


Figure 3.22: Step response of the 4th, 3rd and 2nd order cut-bobbin models with unit voltage input.

3.3 Non-Linearities

Moving coil actuators are relatively linear in comparison to other electromagnetic actuators such as solenoids. The magnetic flux density along the travel of the moving coil is designed to be constant, providing a linear relationship between voltage input and force output over the length of its travel. It also does not suffer from backlash as electromechanical actuators do. However, it is inevitable that non-linearities exist within the system as all physical systems have limits on their magnitude. For example, the moving coil actuator has travel limits, as its motion is transversal, and the input voltage to the system is also limited by the supply voltage. In normal operation, the actuator will be controlled to operate within these limits, making the linear model valid for control design. However, when faults occur the system may reach these limits, and thus it necessary to include them in the model for simulation purposes.

3.3.1 Travel Limits

When the moving-coil reaches the end of its travel it will hit the end stop, abruptly reducing the velocity of the coil and limiting the position. This can be modelled as a stiff spring and damper that occurs at the travel limits, through modification of the mechanical subsystem:

$$m\ddot{x} = \begin{cases} F_e - d\dot{x} - rx, & \forall x_{lim} > |x| \\ F_e - (d + d_{lim})\dot{x} - (r + r_{lim})x, & \forall x_{lim} \leq |x| \end{cases} \quad (3.54)$$

Where x_{lim} is the position limit, and d_{lim} and r_{lim} are the damping and stiffness caused by the end-stop. The values of d_{lim} and r_{lim} are several orders of magnitude larger than the normal damping and stiffness values. This non-linearity can be included in Simulink using logic operators that detect when the coil has reached the travel limit, and switches the end-stop damping and stiffness on accordingly.

Figure 3.23 shows the step response of the element when the travel limits are included using this approach. It can be seen that the end-stop damping and stiffness limits the position and nulls the velocity sufficiently.

3.3.2 Input Saturation

The input voltage to the system is limited to ± 24 V. This can be included in the Simulink model through the introduction of a saturation block at the input, which limits the input to the required region.

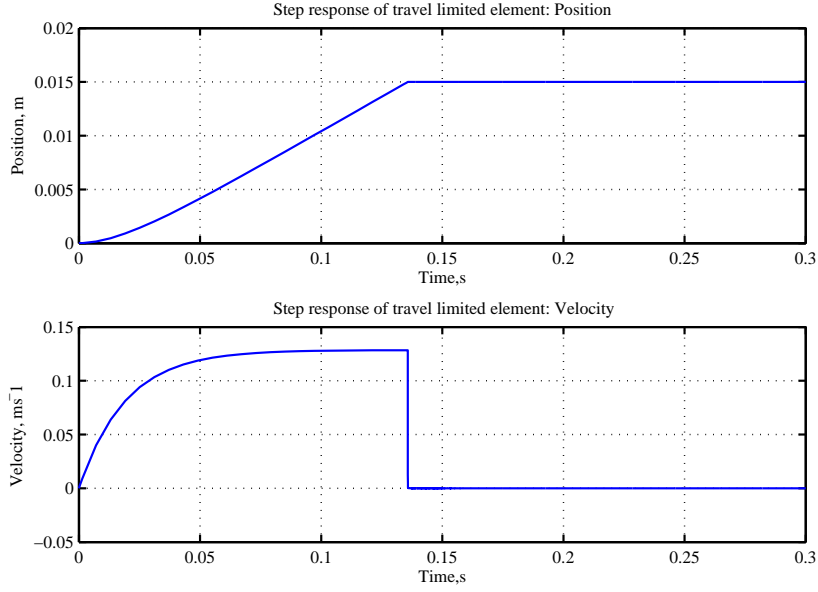


Figure 3.23: Step response of travel limited element.

3.4 Modelling of HRA Configurations

Having developed a model for a single actuation element for simulation and control synthesis purposes, multi-element assemblies can now be constructed. The focus of the current project phase is on planar assemblies and thus the elements are arranged either serially, or in parallel, or in serial/parallel combinations. The optimum configuration of actuation elements, in accordance with the high redundancy actuation concept, is addressed by the work of other HRA researchers, and as such will not be addressed here. This section will address the issue of creating models of possible actuation assemblies using SMAC moving coil actuators as actuation elements. Models of specific configurations that are to be used in the control design chapters are detailed in the next chapter.

Assemblies of actuation elements can be constructed with ease if the model is split into two sections: the electrical/mechanical force model and a separate mass model. This aids the construction of element assemblies, as each actuator element will apply its force to a mass that is dependant on the configuration.

For example, in a two-by-two series in parallel assembly, as shown in Figure 3.24, elements one and two work upon masses m_1 and m_2 respectively. m_1 and m_2 are the combined mass of the moving mass of elements 1 and 2 the casing mass of 3 and 4, respectively. The casing masses of actuator elements 1 and 2 are not included in the diagram as they are fixed to a surface. Actuator elements 3 and 4 both apply their force to m_3 , which is the combined mass of the moving masses of elements 3 and 4 and the load mass. The division between the electrical/mechanical force model and the mass model is distinct within this figure, as each actuation element works upon a mass that is dependant on the assembly's arrangement.

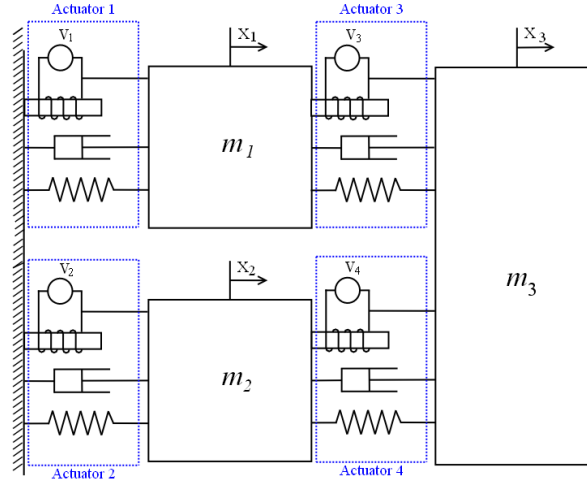


Figure 3.24: Two-by-two series-in-parallel assembly.

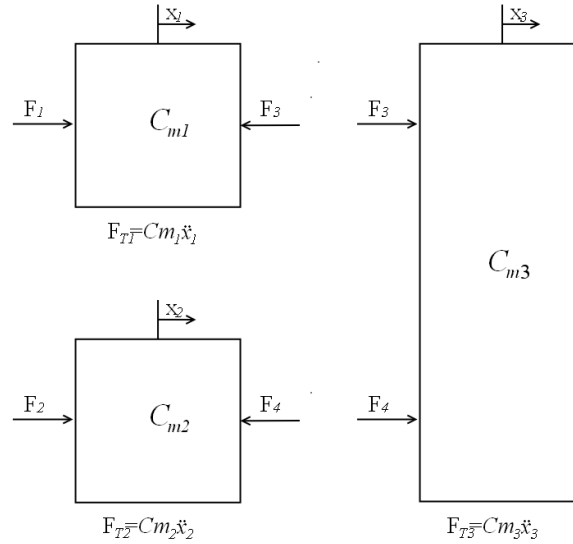


Figure 3.25: Free-body diagram of two-by-two series-in-parallel assembly.

A free-body diagram of the assembly shown in Figure 3.24 is provided in Figure 3.25. Each force within this diagram can be described generically by the following equations (using the 4th order closed-bobbin element model from equation 3.44):

$$F_n = A_{31_n} I_{R1_n} + A_{32_n} I_{Lm_n} + A_{33_n} (\dot{x}_{(i-1)} - \dot{x}_i) + r(x_{(i-1)} - x_i) \quad (3.55)$$

$$\dot{I}_{R1_n} = \frac{1}{L_{1_n}} u_n - A_{11_n} I_{R1_n} + A_{12_n} I_{Lm_n} + A_{13_n} (\dot{x}_{(i-1)} - \dot{x}_i) \quad (3.56)$$

$$\dot{I}_{Lm_n} = A_{21_n} I_{R1_n} + A_{22_n} I_{Lm_n} + A_{23_n} (\dot{x}_{(i-1)} - \dot{x}_i) \quad (3.57)$$

where:

$$\begin{aligned}
 A_{11_n} &= \frac{R_{2_n}(R_{2|3_n}-R_{3_n})-R_{1_n}R_{3_n}}{L_{1_n}R_{3_n}}, & A_{12_n} &= \frac{R_{2_n}(R_{3_n}-R_{2|3_n})}{L_{1_n}R_{3_n}}, & A_{13_n} &= \frac{k_n R_{2|3_n}}{L_{1_n}R_{3_n}}, \\
 A_{21_n} &= \frac{R_{2_n}(R_{3_n}-R_{2|3_n})}{L_{1_n}R_{3_n}}, & A_{22_n} &= \frac{R_{2_n}(R_{2|3_n}-R_{3_n})}{L_{m_n}R_{3_n}}, & A_{23_n} &= \frac{k_n R_{2|3_n}}{L_{1_n}R_{2_n}}, \\
 A_{31_n} &= \frac{k_n R_{2|3_n}}{R_{3_n}}, & A_{32_n} &= \frac{k_n(R_{3_n}-R_{2|3_n})}{R_{3_n}}, & A_{33_n} &= \frac{k_n^2 + d_n(R_{2_n} + R_{3_n})}{R_{2_n} + R_{3_n}}.
 \end{aligned}$$

Where n and i are the actuator element and mass identifier, respectively and $(i-1)$ is the mass that physically (rather than numerically) precedes mass i . A generic state-space expression may be derived from these equations to represent the element subsystem and the mass subsystem:

$$G_e : \begin{bmatrix} \dot{I}_{R1_n} \\ \dot{I}_{Lm_n} \end{bmatrix} = \begin{bmatrix} -A_{11_n} & A_{12_n} \\ A_{21_n} & A_{22_n} \end{bmatrix} \bullet \begin{bmatrix} I_{R1_n} \\ I_{Lm_n} \end{bmatrix} + \begin{bmatrix} \frac{1}{L_{1_n}} & A_{13_n} \\ 0 & A_{23_n} \end{bmatrix} \bullet \begin{bmatrix} u_n \\ (\dot{x}_{(i-1)} - \dot{x}_i) \end{bmatrix} \quad (3.58)$$

$$F_n = \begin{bmatrix} A_{31_n} & A_{32_n} \end{bmatrix} \bullet \begin{bmatrix} I_{R1_n} \\ I_{Lm_n} \end{bmatrix} + \begin{bmatrix} 0 & A_{33_n} \end{bmatrix} \bullet \begin{bmatrix} u_n \\ (\dot{x}_{(i-1)} - \dot{x}_i) \end{bmatrix} \quad (3.59)$$

$$G_m : \begin{bmatrix} \ddot{x}_i \\ \dot{x}_i \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \bullet \begin{bmatrix} \dot{x}_i \\ x_i \end{bmatrix} + \begin{bmatrix} \frac{1}{m_i} \\ 0 \end{bmatrix} F_{T_i} \quad (3.60)$$

Using the generic force equations, the state-space equations for the example assembly are:

$$\dot{I}_{R1_1} = \frac{1}{L_{1_1}}u_1 - A_{11_1}I_{R1_1} + A_{12_1}I_{Lm_1} - A_{13_1}\dot{x}_1 \quad (3.61)$$

$$\dot{I}_{R1_2} = \frac{1}{L_{1_2}}u_2 - A_{11_2}I_{R1_2} + A_{12_2}I_{Lm_2} - A_{13_2}\dot{x}_2 \quad (3.62)$$

$$\dot{I}_{R1_3} = \frac{1}{L_{1_3}}u_3 - A_{11_3}I_{R1_3} + A_{12_3}I_{Lm_3} + A_{13_3}(\dot{x}_1 - \dot{x}_3) \quad (3.63)$$

$$\dot{I}_{R1_4} = \frac{1}{L_{1_4}}u_4 - A_{11_4}I_{R1_4} + A_{12_4}I_{Lm_4} + A_{13_4}(\dot{x}_2 - \dot{x}_3) \quad (3.64)$$

$$\dot{I}_{Lm_1} = A_{21_1}I_{R1_1} + A_{22_1}I_{Lm_1} - A_{23_1}\dot{x}_1 \quad (3.65)$$

$$\dot{I}_{Lm_2} = A_{21_2}I_{R1_2} + A_{22_2}I_{Lm_2} - A_{23_2}\dot{x}_2 \quad (3.66)$$

$$\dot{I}_{Lm_3} = A_{21_3}I_{R1_3} + A_{22_3}I_{Lm_3} + A_{23_3}(\dot{x}_1 - \dot{x}_3) \quad (3.67)$$

$$\dot{I}_{Lm_4} = A_{21_4}I_{R1_4} + A_{22_4}I_{Lm_4} + A_{23_4}(\dot{x}_2 - \dot{x}_3) \quad (3.68)$$

$$F_1 = A_{31_1}I_{R1_1} + A_{32_1}I_{Lm_1} - A_{33_1}\dot{x}_1 + r x_1 \quad (3.69)$$

$$F_2 = A_{31_2}I_{R1_2} + A_{32_2}I_{Lm_2} - A_{33_2}\dot{x}_2 + r x_2 \quad (3.70)$$

$$F_3 = A_{31_3}I_{R1_3} + A_{32_3}I_{Lm_3} + A_{33_3}(\dot{x}_1 - \dot{x}_3) + r(x_1 - x_3) \quad (3.71)$$

$$F_4 = A_{31_4}I_{R1_4} + A_{32_4}I_{Lm_4} + A_{33_4}(\dot{x}_2 - \dot{x}_3) + r(x_2 - x_3) \quad (3.72)$$

$$F_{T1} = m_1 \ddot{x}_1 = F_1 - F_3 \quad (3.73)$$

$$F_{T2} = m_2 \ddot{x}_2 = F_2 - F_4 \quad (3.74)$$

$$F_{T3} = m_3 \ddot{x}_3 = F_3 - F_4 \quad (3.75)$$

This subsystem approach can be used in Simulink to build HRA configurations. The block diagram is split into a element subsystem and a mass subsystem as shown in Figures 3.26 to 3.28. These subsystems can be easily combined to create all possible serial/parallel assemblies. For example, the arrangement of subsystems necessary to create the two-by-two series-in-parallel assembly is shown in Figure 3.29.

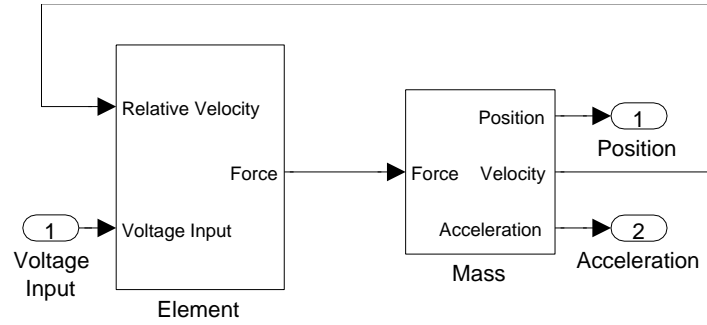


Figure 3.26: Simulink subsystem model.

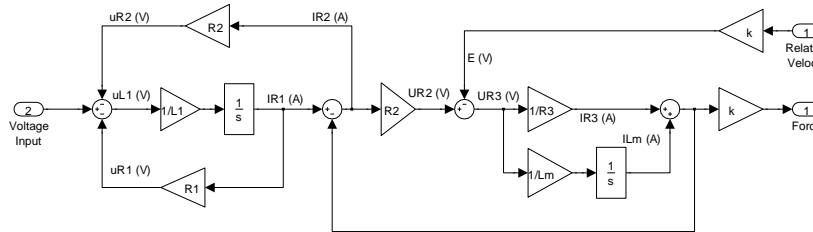


Figure 3.27: Element subsystem from Figure 3.26.

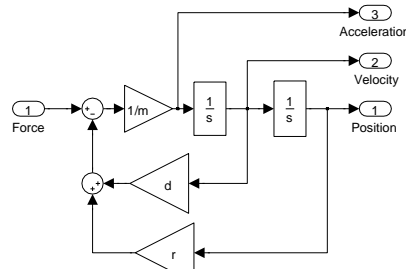


Figure 3.28: Mass subsystem from Figure 3.26.

3.5 Fault Modelling

Having created a nominal model of a single element and a method for creating assembly models, it is necessary to model potential faults that can be injected into the actuation system. Sensor faults are not considered within this thesis as they are not within the scope of this work. There are many methods and application of sensor fault detection, and actuation faults are of much greater import to the HRA concept. Three main fault cases for a moving coil actuator have been identified and modelled, namely:

- Parameter deviations
- Mechanical loose
- Mechanical lock-ups

The simulation of these faults is outlined in the following subsections and some discussion is given on the effect of these faults when elements are arranged to form a HRA.

3.5.1 Parameter Deviations

Heating of the element will change the resistance within the electrical circuit. Likewise inductances will change a little due to geometry changes arising from thermal expansion. These electrical changes may be considered as parameter deviation faults. Parameter deviations can be made by varying the blocks within the simulations or parameter values in the equations. The effect of parameter deviations on some specific HRA examples will be discussed in Chapter 5. A deviation of 10% is considered

3.5.2 Mechanical Loose Faults

An actuator element is described as mechanically loose if it loses the ability to exert force between its two end points. Simulation of the mechanical loose fault can be achieved through severing the force output of the element in Figure 3.27.

Loose faults in parallel systems remove the force generated by an element from the mass(es) upon which it acts. For example, in the parallel arrangement of elements shown in Figure 3.30, if element 2 becomes mechanically loose, then it no longer applies the force F_2 to the mass. However, as elements 1, 3 and 4 are not faulty, then F_1 , F_3 and F_4 are still applied to the mass, retaining its controllability.

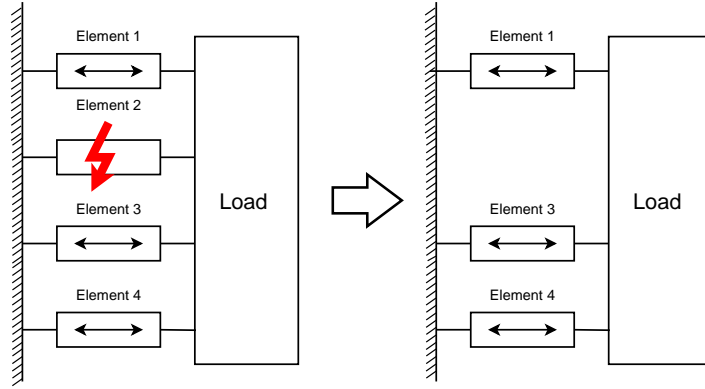


Figure 3.30: Loose fault in parallel actuation elements

Loose faults in serial elements, however, are more problematic. If element 2 becomes loose in the serial elements shown in Figure 3.31, then the connection to ground is lost. Elements 3 and 4 will continue to apply a force to their respective masses, but this force will act equally and oppositely on the loose element's mass and meet no resistance, other than inertial forces, until the coil reached the end of its travel and hits the end-stop. Hence, loose faults render the load of a purely serial arrangement of elements uncontrollable.

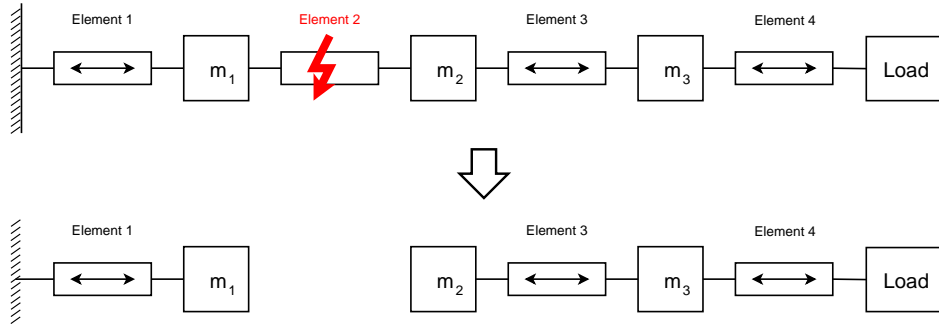


Figure 3.31: Loose fault in serial actuation elements

3.5.3 Mechanical Lock-up Faults

If an element loses the ability to change the length between its two end points then it is said to be mechanically locked-up. This may occur if the coil of the first actuator element is deformed and touches the magnet. This fixes the mass with respect to the reference point, and consequently the relative position is constant and the relative speed is zero.

The mechanical lock-up fault is more challenging in terms of its dynamic effect on the system and its ease of simulation in comparison to the other faults. If an element becomes locked, then the two masses it links will move at the same speed. Therefore, they can be considered as one larger mass. For example, in the serial elements shown in Figure 3.32, if element 2 becomes locked, then m_1 and m_2 can be combined into one mass. This reduces the order of the system by 4 to 2 states (depending on the order of the electrical subsystem used). Force can still be exerted on the load, and thus serial elements increase a HRAs tolerance to lock-up faults.

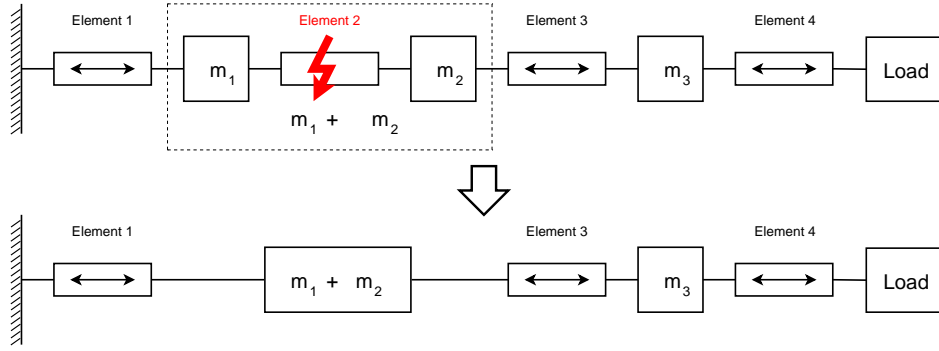


Figure 3.32: Lock-up fault in serial actuation elements

Parallel elements do not improve the system's tolerance to lock-up faults, however. If a lock-up fault occurs in parallel elements, as depicted in Figure 3.33, then this effectively locks all of the parallel actuators, assuming that the elements are not capable of breaking the locked element (which is the case for the HRA concept).

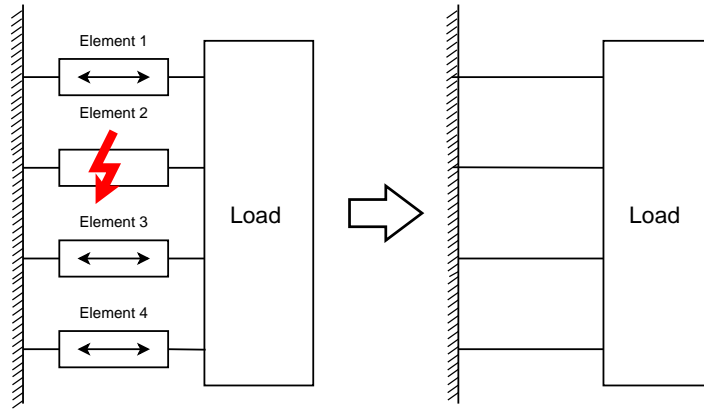


Figure 3.33: Lock-up fault in parallel actuation elements

A lock-up fault may be simulated by changing the friction of an element to a very large value, so that the actuation element is unable to move. This approach incurs solver problems in simulation, as the solutions become very small, simulation run times increase. This issue is particularly problematic in this case as the HRA contains many elements, a number of which could be locked simultaneously.

Hence, lock-up faults have been simulated using an alternative approach here. Lock-up fault simulation can be achieved by using the impulse of the mass. Impulse is the change in momentum and can be expressed as:

$$Fdt = m\dot{x} \quad (3.76)$$

By passing the neighbouring elements the impulse and mass through element blocks in Simulink when they become locked, then effectively the locked element can be removed from the simulation: the masses add to form the locked actuator mass and the momentum in the system is preserved.

This idea is further illustrated in Figures 3.34 and 3.35 where representations of the simulation process for a simple system are given. The impulse and mass is passed upwards and downwards throughout the system between the mass simulation blocks. If the elements are healthy, then these signals are switched off and the sum of the mass and impulse in each mass simulation block is only that of the m_n and Fdt_n . However, lock-up faults activate the mass and impulse links as shown in Figure 3.35, and the summed mass and impulses in the masses attached to a locked element are aligned to form one mass. This may be expressed more formally as:

$$\dot{x}_n = \frac{(Fdt_n + f_{lock(n)}Fdt_{below(n)} + f_{lock(n+1)}Fdt_{above(n+1)})}{(m_n + f_{lock(n)}m_{below(n)} + f_{lock(n+1)}m_{above(n+1)})} \quad (3.77)$$

where

$$m_{below(n)} = m_{(n-1)} + f_{lock(n-1)}m_{below(n-1)} \quad (3.78)$$

$$m_{above(n)} = m_{(n+1)} + f_{lock(n+1)}m_{above(n+1)} \quad (3.79)$$

$$Fdt_{below(n)} = Fdt_{(n-1)} + f_{lock(n-1)}Fdt_{below(n-1)} \quad (3.80)$$

$$Fdt_{above(n)} = Fdt_{(n+1)} + f_{lock(n+1)}Fdt_{above(n+1)} \quad (3.81)$$

and

$$f_{lock(n)} = 0 \text{ when } e_n \text{ is healthy}$$

$$f_{lock(n)} = 1 \text{ when } e_n \text{ is locked}$$

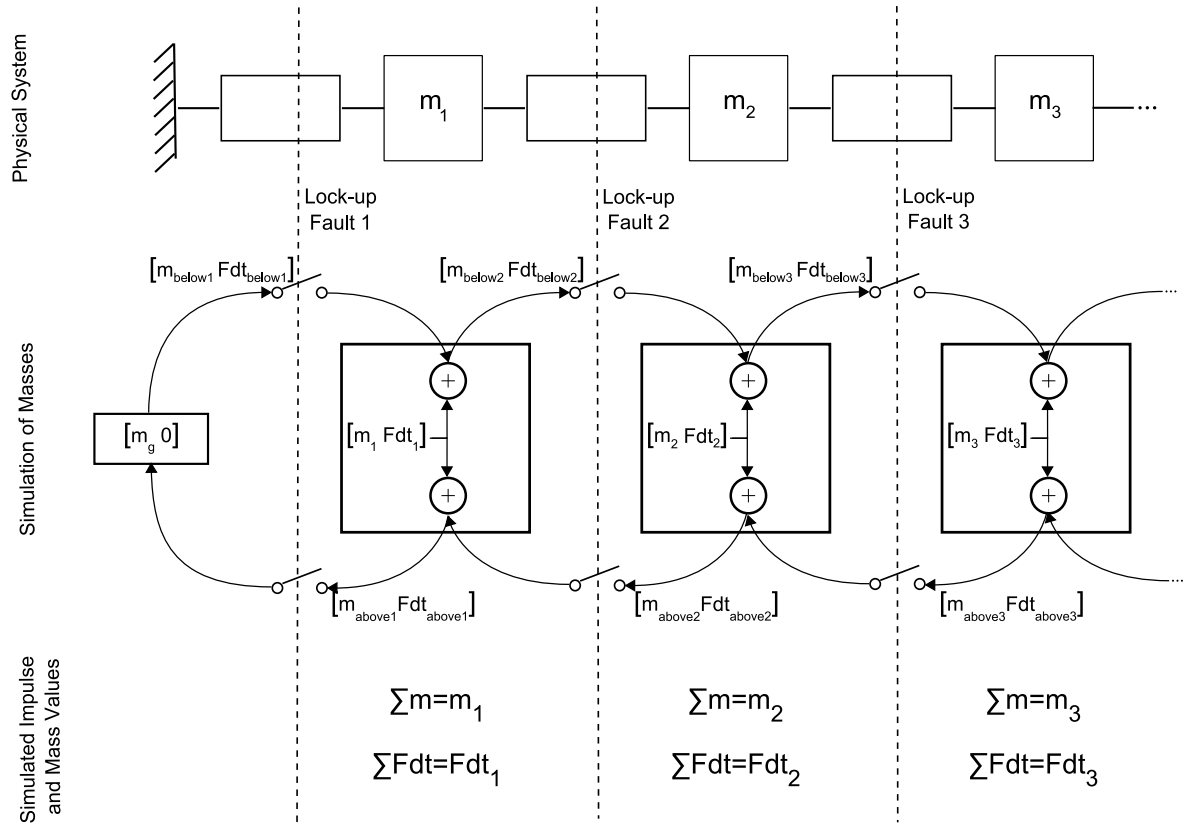


Figure 3.34: Simulation of lock-up faults through passing the mass and impulse - Nominal system

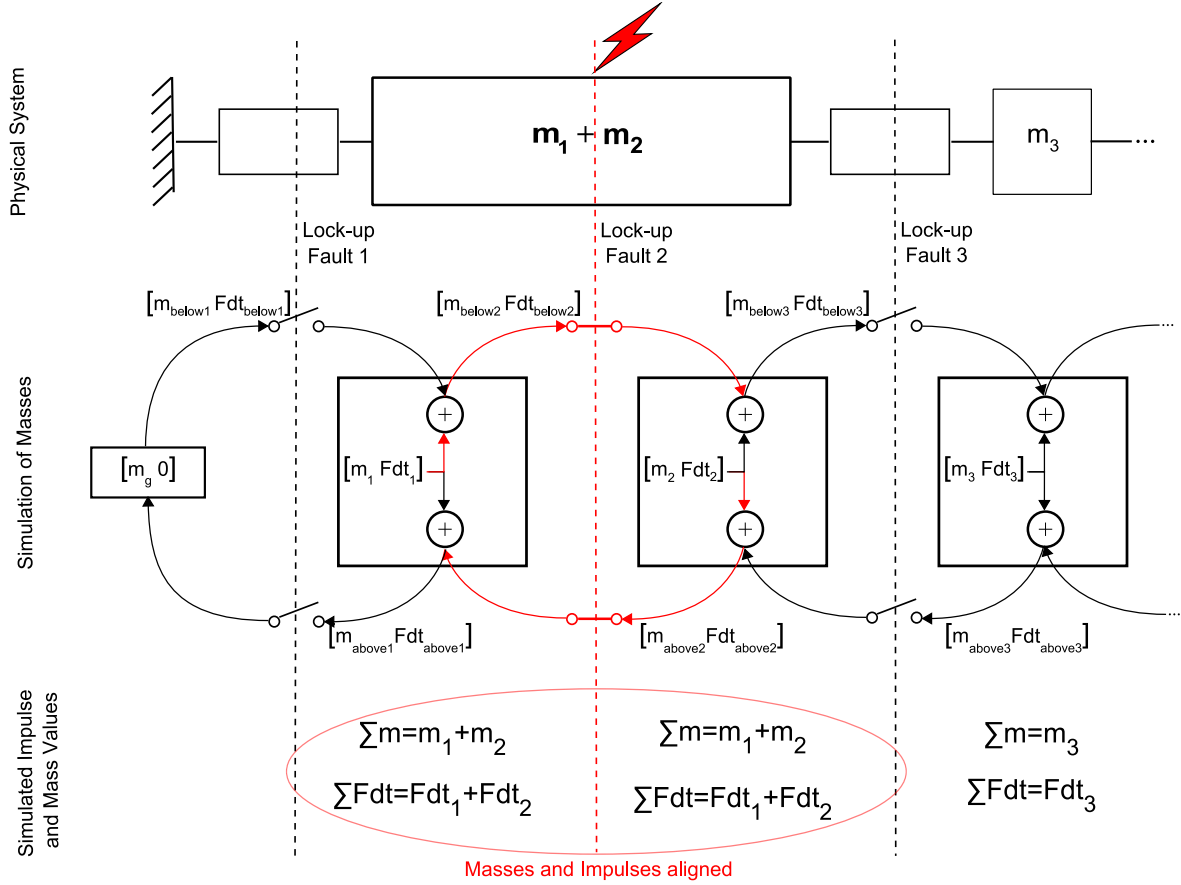


Figure 3.35: Simulation of lock-up faults through passing the mass and impulse - Lock-up in element 2

3.6 Conclusions

The modelling of HRA using moving coil actuation technology has been addressed throughout this chapter. It was established that two types of model were required: a relatively simple LTI model for control design purposes and a more complete model for simulation. To achieve this a LTI model for a single moving coil actuator was derived from first principles and verified. This model was combined with some non-linearities to form the element model for simulation purposes. The element model for control design was then produced by reducing the full-order LTI to second order by physical reduction means.

Having derived an element model, a methodology was presented for creating and simulating assemblies of elements which form HRAs, which could be applied to either element model. Also, the modelling of a number of fault cases was described, and their effect on parallel and serial elements discussed.

Chapter 4

Application Example

4.1 Introduction

The control of HRA has been the primary focus of this phase of the HRA project, and indeed is the focus of the work contributed by this thesis. However, before the control of HRA can be addressed it is important to define the system that is to be controlled. The modelling of a single actuation element has been presented and methods of creating HRA models given. However, the dimensioning of a specific HRA to meet requirements has not been addressed.

Defining the HRA's requirements is problematic at this stage of the project. The HRA is to be used as part of a larger system and thus, its requirements are dependent on the specifications of that system. At present however, the concept is not concentrated on a specific application (as it is applicable to many areas). Also, the current technology used within the project serves to only prove the viability of the HRA concept and is not the final technological solution. The technology used within a HRA synthesised for a real application is likely to be of a much smaller scale if high numbers of elements are to be used¹.

Nonetheless, for the control results that follow to have meaning, it is important to dimension the HRA system studied sensibly and design its performance to meet realistic criteria. Hence, this chapter defines a number of example HRAs for use within the control studies, whose dimensions and control requirements are loosely based upon some typical specifications for possible applications for a HRA. To this end, Section 4.2 provides a general discussion of typical HRA application requirements. Based on these and the current technology specification, requirements for three example HRAs are synthesised in Section 4.3. Finally, having defined some example HRAs and requirements, the capabilities of these systems and their fault tolerance is discussed in Section 4.4.

4.2 Typical Application Requirements

A number of potential applications for the HRA have been studied as part of the HRA project's work. These applications include a flight control surface, railway active suspension, a jet engine inlet guide vane, and a pick-and-place application. The requirement types of these

¹These technological and synthesis issues will be the focus of the next phase of the HRA project as a whole

applications are varied, making comparison difficult. Table 4.1 assembles all the relevant capabilities that are used in more than one application. It can be concluded that these requirements are typical for actuators across different fields, and they are not specific to a certain application.

Table 4.1: Typical application requirements

	Flight control surface (151)	Railway active suspension (152)	Pick & place	Jet engine inlet guide vane (17)
Static performance				
Force capability	13kN	3.3kN	-	55kN
Acceleration	-	2.25m/s ²	150m/s ²	-
Speed	0.1m/s	0.12m/s	1m/s	0.1m/s
Travel	-	18mm	30mm	50mm
Tracking Performance				
Overshoot	-	-	0.01mm	10%
Frequency	0-3Hz	0.1-20Hz	0-15Hz	0-2Hz

4.3 Example HRAs Requirements

Based upon the requirements in Table 4.1 and the specifications of the actuation element currently in use (Table 4.2), requirements for three generic example applications have been created (Table 4.3).

Firstly a 4×4 parallel-in-series (PS) system is chosen, as this is the dimensioning and configuration of the experimental rig. Example system 2 is a series-in-parallel (SP) system of the same dimensioning, and is included as this is the other main configuration for a planar HRA. The requirements for both of these systems are the same.

Example HRAs 1 and 2 contain sixteen elements, which is a relatively low number of elements for a HRA, making the system relatively simple to manage. However, this low number of elements reduces the efficacy of the HRA concept. Fewer elements results in higher levels of over-dimensioning, and faults in individual elements have more affect on the overall system behaviour. Hence, a more realistically dimensioned HRA is provided in the final example system, which is a 10×10 PS HRA.

Table 4.2: Actuation element specifications

Moving-coil actuator specification	
Force	100N
Max. acceleration	80m/s ²
Travel	± 0.015 m
Moving mass	0.13kg
Load	1kg

Table 4.3: Application example requirements

Capability	Application using 4×4 HRA (PS & SP)	Application using 10×10 PS HRA
Force	200N	600N
Acceleration	80m/s^2	80m/s^2
Speed	0.05m/s	0.05m/s
Travel	$\pm 30\text{mm}$	$\pm 90\text{mm}$
Load	2kg	6kg
Overshoot	2%	2%
Frequency	1Hz	1Hz

The application requirements have been scaled between the two example sizes in this case. This is due to the limitations of the current technology. In a real situation, the 10×10 HRA's elements would be of a smaller scale to meet the same requirements as the 4×4 case (or vice-versa). This is not possible in the current technology, so the application requirements have been scaled instead, providing the same effect. The resulting requirements are generally a factor of 10 from the the typical requirements of the real applications of Table 4.1, which is due to the specification of the current element technology.

Models for these examples can be found in Appendix C.

Finally, control requirements can be formulated from this specification, and are included in Table 4.4.

Table 4.4: Control requirements for example systems

Control Requirements	Examples 1 & 2 4×4 PS & SP	Example 3 10×10 PS
Transient response requirements		
Settling time (ST)	1s	1.2s
Rise time (RT)	0.7s	0.75s
Overshoot (OS)	2%	2%
Frequency domain requirements		
Gain margin (GM)	$> 10\text{dB}$	$> 10\text{dB}$
Phase margin (PM)	$> 60\text{deg}$	$> 60\text{deg}$
Frequency	1Hz	1Hz

4.4 Capability & Fault Tolerance

Based on the dimensions of the examples HRAs and the requirements, it is now possible to deduce the nominal capabilities of the HRAs and their levels of fault tolerance to lock-up and loose faults. Capability and fault tolerance are important concepts for the HRA.

The system's capability, as discussed here, is defined as the amount of force or travel the system can potentially produce in comparison to the required force/travel needed to meet the

specification. For example, in a traditional triplex parallel redundancy system, the nominal system has a force capability of 300% (3 times the force required) and a travel capability of 100% (it can travel the required distance but no further). If a fault occurs, this capability will reduce. If it falls below 100%, then the system can no longer achieve the full requirement.

Using this terminology, redundant systems in their nominal state always have capabilities greater than 100%, and as such the nominal HRA must have a capability in excess of 100%. However, this value is also an indicator of the level of over-dimensioning in the system i.e. 300% capability indicates that the system is over-dimensioned 3 times. An aim of the HRA concept is to reduce this over-dimensioning, hence a smaller nominal capability is desirable.

The fault tolerance of the system is defined as the number of faults it can tolerate before the capability falls below 100%. Hence, a triplex parallel redundancy system can tolerate 2 loose faults, and no lock-up faults.

Hence, it is desirable to have a high level of fault tolerance, for a low level of nominal capability. A single indicator of the efficiency with which the fault tolerance is provided may then be derived by dividing the fault tolerance by the nominal capability.

The nominal capabilities and fault tolerances for the example HRAs and the traditional triplex case are provided in Table 4.5.

Table 4.5: HRA capabilities and fault tolerance levels

Capabilities and fault tolerance levels	Parallel triplex system	4 × 4 HRA		10 × 10 HRA
		PS	SP	PS
Nominal force capability	300%	200%	200%	166.66%
Nominal travel capability	100%	200%	200%	166.66%
No. of lock-up faults tolerated	0	2	2-8	4
No. of loose faults tolerated	2	2-8	2	4-40
Lock fault tolerance/nominal capability	0	0.01	0.01-0.04	0.024
Loose fault tolerance/nominal capability	0.0067	0.01-0.04	0.01	0.024-0.24

In the 4 × 4 examples the HRA has a nominal force capability that is twice of that of the requirements. This allows the PS system to tolerate two lock-up faults, as each fault will reduce the system's travel capability by 25%. In the SP case, the capability is reliant on where the fault occurs. If no more than two elements are locked in each serial branch, the system will meet the travel requirement. However, if three faults occur in the same serial branch, then the system can no longer meet this requirement. Hence, the 4x4 SP configuration can

tolerate 2-8 faults.

As the lock-up and loose faults are duals, the opposite is true for loose faults. Only two loose faults are tolerable in the 4×4 SP configuration as each fault will render a serial branch inoperable, and its force will be lost. The PS configuration may tolerate more loose faults, as long as they occur in separate parallel branches.

The nominal capabilities of the 10×10 examples are lower than that of the 4×4 examples. However, the tolerance to faults has increased, as each fault has a smaller influence on the whole system. This is a key feature of the HRA concept: by reducing the element size, and increasing their numbers greater fault tolerance can be achieved with less functional redundancy. This has implications for improving the efficiency, and weight of the system, if a suitable technology can be used.

This is numerically illustrated by the fault tolerance/nominal capability values given in the last two rows of Table 4.5. The values displayed here are derived from dividing the fault tolerance by the nominal capability, and as such, they are a measure of how much tolerance is achieved with the level of functional redundancy. Hence, it is a measure of the efficiency of the system, and a higher value is better as it means more fault tolerance with less functional redundancy. The table shows that the efficiency with which the tolerance of the triplex system is achieved is lower than that of the HRA systems. The 4×4 systems also have a lower level of efficiency/tolerance in comparison to the 10×10 system.

4.5 Conclusions

This chapter produced a number of example HRA configurations that meet requirements that are sensibly dimensioned, in order to provide HRA systems for use in the control studies that follow. To achieve this, firstly, the requirements for some typical HRA applications were found. Based upon these requirements and the specifications of the current HRA technology, some example HRA requirements were established. Two sets of HRA requirements for two sizes of HRA (4×4 and 10×10) were created and based on these requirements, three example HRA configurations were synthesised. The fault tolerance and capabilities of the example configurations was then discussed. It was shown that PS configurations offer greater tolerance to lock-up faults and SP to loose faults. Also, the 10×10 configuration was shown to provide greater fault tolerance using less functional redundancy, which is an important feature of the HRA concept.

Models for the HRA configurations included in this chapter can be found in Appendix C.

Chapter 5

Passive Fault Tolerant Control of High Redundancy Actuation

5.1 Introduction

Control is often integral to providing fault tolerance. The HRA project thus far has focused on using passive Fault Tolerant Control (FTC) to provide fault tolerance. Passive FTC is where a single robust control law is designed, which should provide adequate stability and performance under both nominal and fault conditions¹. Passive FTC been shown to be theoretically viable for fault tolerant control with low levels of redundancy (5; 3; 7; 4), and successful practical testing of these results on a two-by-two electromechanical HRA was achieved (7; 6). More recently Steffen has investigated robust control of HRAs with higher numbers of elements (21; 22). Results indicate that robust control should be a satisfactory method of achieving fault tolerant control of these structures for most applications.

The passive FTC concept with respect to HRA is illustrated by Figure 5.1. The behaviour of the nominal HRA is represented by a point b_n in the diagram. Inevitably, a bound of uncertainty for the system surrounds this point. b_n and its uncertainty bound lies within a region of acceptable behaviours B_{PFT} , within which the system is considered fault tolerant. Passive FTC aims to design a single robust controller that keeps the behaviours of the fault perturbed HRAs (points b_f) within B_{PFT} .

Although the HRA has a capability level in excess of that required by the application, lock-up and loose faults reduce the overall travel or force capability respectively, and as such, there are fault limits dictated by the capability requirement. Thus, HRA under fault conditions in excess of this limit (represented by points b_{gd}) will lie outside B_{PFT} in B_{GD} , a region that represents the HRA graceful degradation operation.

The passive FTC approach is attractive, as its simplicity and constancy make it more easily verifiable for a high integrity application. However, if the region B_{PFT} is restricted, then it can be difficult or impossible to retain $\{b_f\}$ within this region.

Hence, active FTC approaches have also been investigated, which detect element faults and change the control in order to move the points b_f closer to b_n , into a behaviour region

¹A detailed discussion of passive FTC is provided in the literature review in Chapter 2.

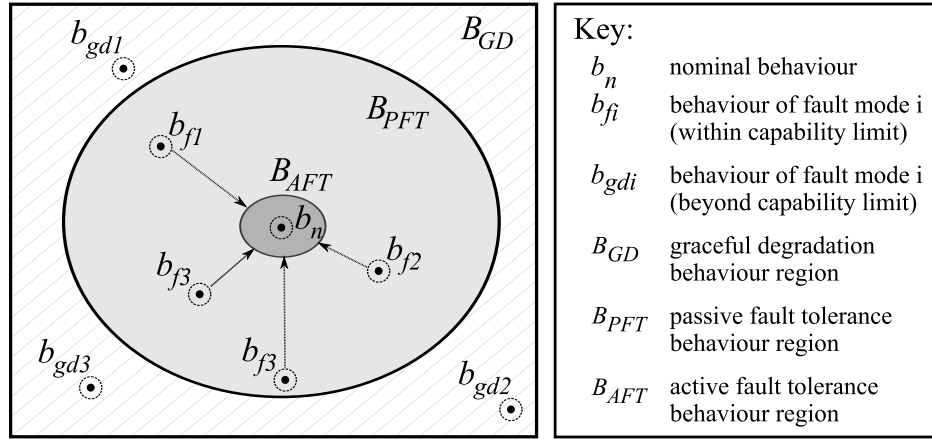


Figure 5.1: Diagrammatic representation of passive and active fault tolerant control of HRA.

B_{AFT} that provides improved performance under fault conditions within the limits of the of the system capability.

This chapter details the design and analysis of basic passive fault tolerant control for the HRA in order to provide a benchmark for comparison with the multi-agent control strategies which will be presented in Chapter 7. In addition, the work presented in this chapter will give an indication of the attainable performance of the HRA using passive methods alone, and allow the demonstration of a number of the key features of HRA in a simple context.

The sections that follow contain the design and analysis of classical passive control for each of the three example HRAs that were detailed in Chapter 4. In each case, the following approach is taken:

1. System analysis - the example system is examined under nominal and fault conditions (according to the fault injection methodology described in Section 5.1.1) to determine the requirement for control. The reduced order HRA models are used in the analysis stage to aid clarity.
2. Control design - based on the results of the system analysis, a global load position controller is designed. The control structure used is illustrated in Figure 5.2. A reference is provided to the system which represents a global travel command for the whole HRA, and feedback of the load position measurement is used. The passive controller provides an identical drive voltage to all actuators in the system. This is a very simple arrangement which requires only one sensor and input for the whole HRA. This simplicity is intentional, as the whole passive approach is an exercise in simplicity, and serves to illustrate what can be achieved using very basic methods with HRA.
3. Fault simulation - having established a control design using the linear, reduced order HRA models, the controller's performance under fault conditions is analysed using the full order, non-linear systems in order to provide a more accurate representation. The efficacy of the control design is subsequently evaluated as described in Section 5.1.2.

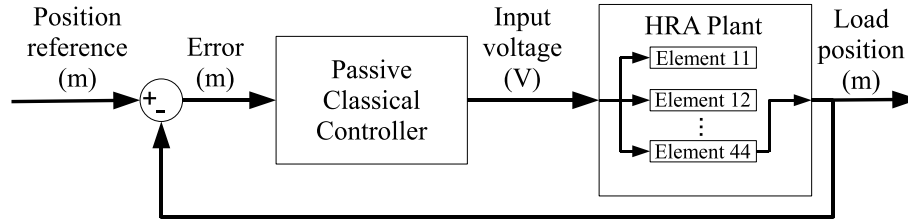


Figure 5.2: Passive fault tolerant control structure.

5.1.1 Fault Injection Methodology

Due to the example HRA's relatively large number of elements, there are many possible locations for fault injection, particularly when one considers multiple faults within the system. However, examining each possible case is impractical, and thankfully unnecessary, as many of these faults have similar effects. Hence, a fault injection methodology is described in this section to provide a sufficiently thorough, yet concise procedure to analyse the system under fault.

Firstly, the influence of parameter uncertainty is considered, then the effects of lock-up faults and loose faults are addressed.

5.1.1.1 Parameter Uncertainties

As stated earlier, the nominal behaviour of the system is subject to a degree of uncertainty (Figure 5.1). The parameters of the reduced order system include the force constant k , the input resistance R , and the mechanical damping and stiffness coefficients d and r respectively, the actual value of which is subject to uncertainty. Hence, considering the affect of their variation on the system is prudent.

A 10% deviation in each parameter is introduced into the system individually, in order to get an appreciation of each parameter's influence. A change of 10% is used in line with previous HRA studies made in (7). As there are many elements in each system, there are many parameters that can be varied. However, varying one element's resistance in a parallel branch of elements is equivalent to varying any of the other elements parallel to it. Hence, where elements are arranged in parallel, only one variation per branch will be considered. Therefore, with reference to Figure 5.3, in both the SP and PS case elements E11, E21, E31 and E41 are injected with parameter deviation faults consecutively, and the same follows for the 10×10 system.

5.1.1.2 Loose Faults

Loose faults are injected into the system up to the limit of the HRA's force requirement i.e. 1-8 in 4x4 PS HRA, 1-2 in 4x4 SP HRA and 1-40 in 10x10 PS HRA. As with parameter deviations, a single loose fault in any parallel branch of elements is very similar to a single loose fault in any other parallel branch. Further analysis shows that the quantity of loose

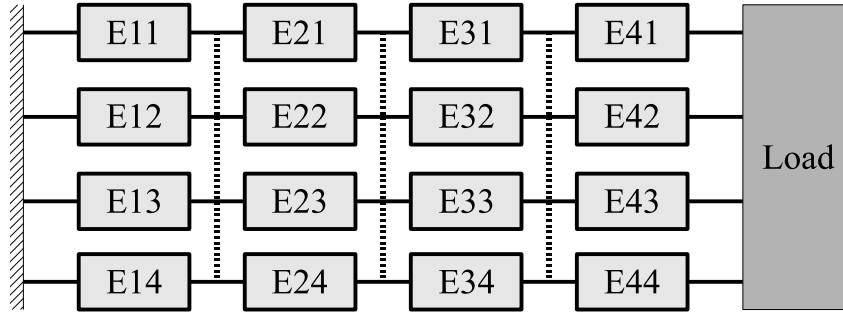


Figure 5.3: Series-in-Parallel and Parallel-in-Series HRA.

Table 5.1: Loose fault injection for 10×10 PS HRA

Fault case	4×4 PS	4×4 SP	10×10 PS
1 loose	$E_{(1,1)}$	$E_{(1,1)}$	$E_{(1,1)}$
2 loose	$E_{(1,1:2)}$	$E_{(1,1:2)}$	-
3 loose	$E_{(1,1:2)}, E_{(2,1)}$	-	-
4 loose	$E_{(1,1:2)}, E_{(2,1:2)}$	-	$E_{(1,1:4)}$
5 loose	$E_{(1,1:2)}, E_{(2,1:2)}, E_{(3,1)}$	-	-
6 loose	$E_{(1,1:2)}, E_{(2,1:2)}, E_{(3,1:2)}$	-	-
7 loose	$E_{(1,1:2)}, E_{(2,1:2)}, E_{(3,1:2)}, E_{(4,1)}$	-	-
8 loose	$E_{(1,1:2)}, E_{(2,1:2)}, E_{(3,1:2)}, E_{(4,1:2)}$	-	$E_{(1:2,1:4)}$
12 loose	-	-	$E_{(1:3,1:4)}$
16 loose	-	-	$E_{(1:4,1:4)}$
20 loose	-	-	$E_{(1:5,1:4)}$
24 loose	-	-	$E_{(1:6,1:4)}$
28 loose	-	-	$E_{(1:57,1:4)}$
32 loose	-	-	$E_{(1:8,1:4)}$
36 loose	-	-	$E_{(1:9,1:4)}$
40 loose	-	-	$E_{(1:10,1:4)}$

faults is decisive in the change of system behaviour, and maximum tolerable faults in the fewest number of parallel branches is the worst case. Hence, the loose faults are injected as described by Table 5.1. Only a selection of the possible 40 loose cases are chosen for the 10×10 example for brevity.

5.1.1.3 Lock-up Faults

As in the loose fault case, lock-up faults are injected into the example systems up to the capability limits of the system (1-2 in 4×4 PS HRA, 1-8 in the 4×4 SP HRA and 1-4 in the 10×10 PS HRA). In the PS arrangement, a lock-up fixes the parallel elements end to end, effectively locking the whole branch. Hence, the lock-up faults in the PS examples are injected into separate branches. The location, serially, has very little affect. However, faults nearer the load are slightly more severe, hence lock-up faults are injected from the load in, as a worst case.

Table 5.2: Lock-up fault injection for example systems

Fault case	4×4 PS	4×4 SP	10×10 PS
1 lock	$E_{(4,1)}$	$E_{(4,1)}$	$E_{(10,1)}$
2 lock	$E_{(3:4,1)}$	$E_{(3:4,1)}$	$E_{(9:10,1)}$
3 lock	-	$E_{(3:4,1)}, E_{(4,2)}$	$E_{(8:10,1)}$
4 lock	-	$E_{(3:4,1:2)}$	$E_{(7:10,1)}$
5 lock	-	$E_{(3:4,1:2)}, E_{(4,3)}$	-
6 lock	-	$E_{(3:4,1:3)}$	-
7 lock	-	$E_{(3:4,1:3)}, E_{(4,4)}$	-
8 lock	-	$E_{(3:4,1:4)}$	-

In the SP example, a maximum of two lock-ups in each serial branch is permissible for the capability requirements, the location of which within the serial branch is negligible. Hence, the faults are injected as described by Table 5.2.

5.1.2 Evaluation Methodology

In order to quantify the affect of faults on the example systems, and establish the degree of accommodation afforded by the designed control scheme, it is pertinent to define some evaluation criteria. Hence, the following characteristics are examined and compared:

- Stability margins - The preservation of stability margins is important for the system under fault and as such, they are examined.
- Additive error – The additive error is the nominal system G minus the faulty system G_f . This indicates the difference between the nominal and perturbed states over the frequency range.
- Infinity norm of additive error – The infinity norm denotes the maximum. Thus, the infinity norm of the additive error is the maximum error, giving an indication of the severity of the fault.
- Transient characteristics - Requirements for each example include overshoot limits and speed requirements, and in addition, as the control objective is set-point tracking, steady-state errors should also be minimised. Hence, the transient characteristics of the nominal and faulty systems are compared.

5.2 Open-Loop System Analysis

This section will examine the three example systems under nominal and fault conditions in order to establish the requirement for control in each case. In addition, this analysis gives an indication of the comparative tolerance to the considered fault types for PS and SP configurations of elements, and for two levels of HRA size (4×4 and 10×10). The models used are the reduced-order representations, full descriptions of which are included in Appendix C.

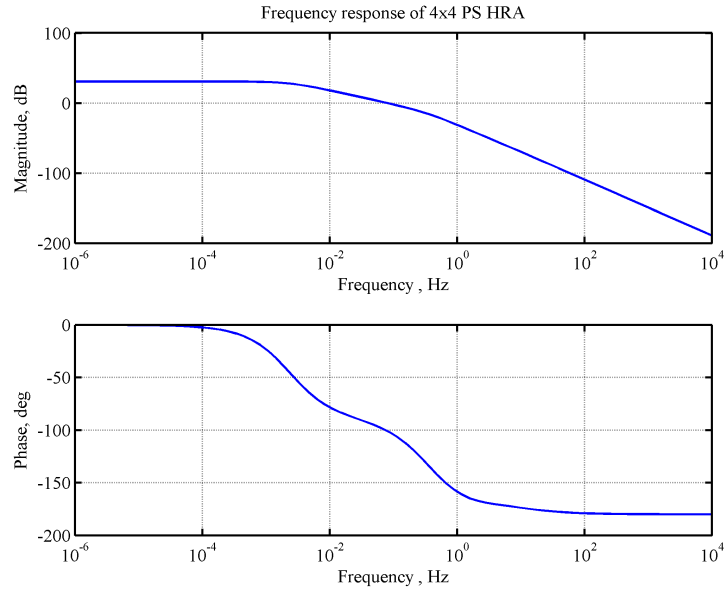


Figure 5.4: Frequency response of 4x4 PS HRA

Table 5.3: Stability margins of nominal example systems

Example HRA	PM (deg)	GM (dB)	BW (rads^{-1})
4x4 PS HRA	79.1	∞	0.51
4x4 SP HRA	79.1	∞	0.51
10x10 PS HRA	40.0	∞	0.79

5.2.1 Nominal system

Figures 5.4 to 5.6 show the frequency response of the three example systems from voltage input to load position output and Table 5.3 contains a summary of their frequency domain characteristics. The 4×4 PS and SP configurations have the same nominal frequency response from the designated input-to-output. The gain and phase margins are adequate in both of these cases, but an increase in bandwidth is required. The 10×10 HRA also requires an increase in bandwidth, but in addition, the phase margin should also be widened.

5.2.2 Parameter Deviations

A 10% parameter deviation is introduced into the example HRAs according to the methodology given in Section 5.1.1. Figures 5.7 to 5.9 gives the resultant singular values of the nominal and parameter deviated systems and Figures 5.10 to 5.12 the additive error between them. Table 5.4 summarises these figures.

The figures show that generally, the perturbation is minimal. The 4×4 examples have similar perturbations. Changes in k and R produce the greatest affect around the bandwidth of the system. Deviations in the damping have little affect over the whole frequency range. However, changes in the stiffness r have the greatest effect at low frequency, resulting in changed steady-state behaviour, which is intuitively correct.

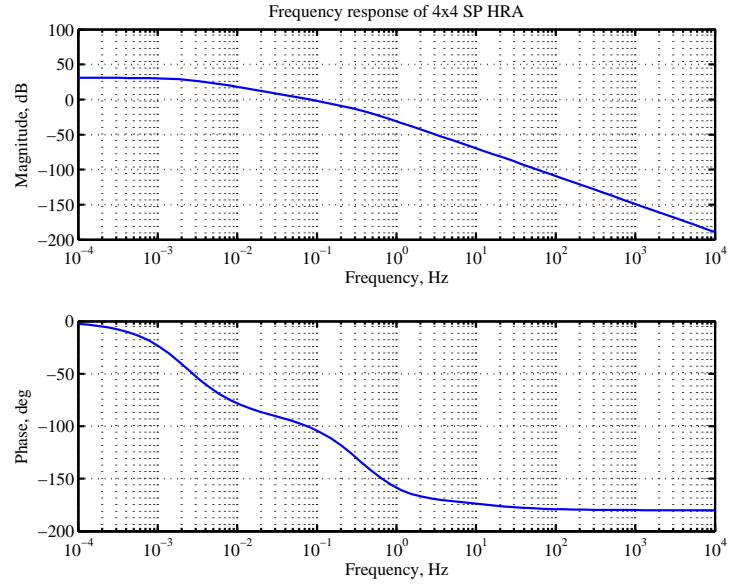


Figure 5.5: Frequency response of 4x4 SP HRA

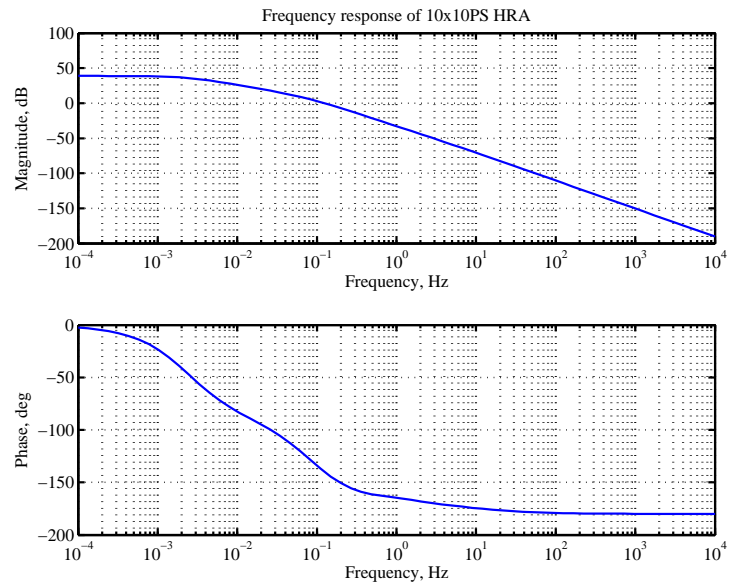


Figure 5.6: Frequency response of 10x10 PS HRA

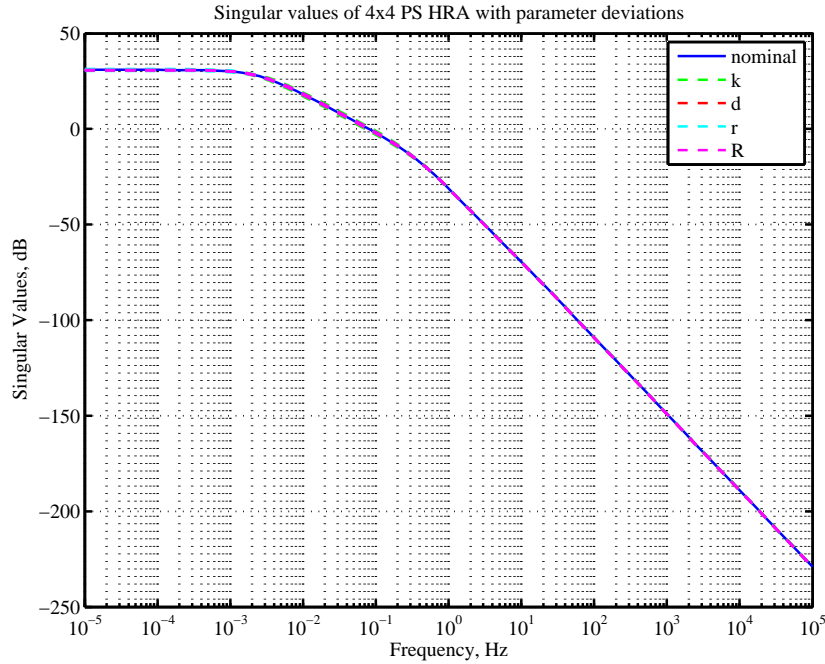


Figure 5.7: Singular values for 4x4 PS HRA with 10% parameter uncertainty.

The parameter changes have a noticeably smaller effect on the 10×10 HRA. Changes in the stability margins and bandwidth are smaller, and the maximum error is considerably lower than the 4×4 systems, a result which reflects an important feature of the HRA concept, where more elements result in decreased effects of faults in individual elements.

5.2.3 Loose Faults

Figures 5.13 to 5.15 give the singular values of the nominal systems and systems with loose faults, injected according to the fault methodology. The additive errors between the nominal and fault cases are also provided in Figures 5.16 to 5.18. Table 5.5 contains the stability margins and additive errors of the loose fault systems.

It is apparent that loose faults have a greater affect on the system in comparison to parameter faults and that more loose faults increase the perturbation. The largest errors occur at low frequency, suggesting that significant steady-state errors are present.

Comparing a single loose fault in each system, it is apparent that, as with the parameter deviations, this fault type has less affect in the system with the most elements. It can also be seen that the loose fault has less influence on the PS configuration than the SP HRA. This is as parallel elements provide greater tolerance to this fault type.

Maximum fault levels in each configuration produce similar changes in bandwidth and phase margin. However, the number of faults required to induce the maximum is different in each case. Up to a 10° decrease in phase margin is induced by the loose faults, hence the control should include at least 10° of phase margin above the minimum requirement to accommodate these changes under fault.

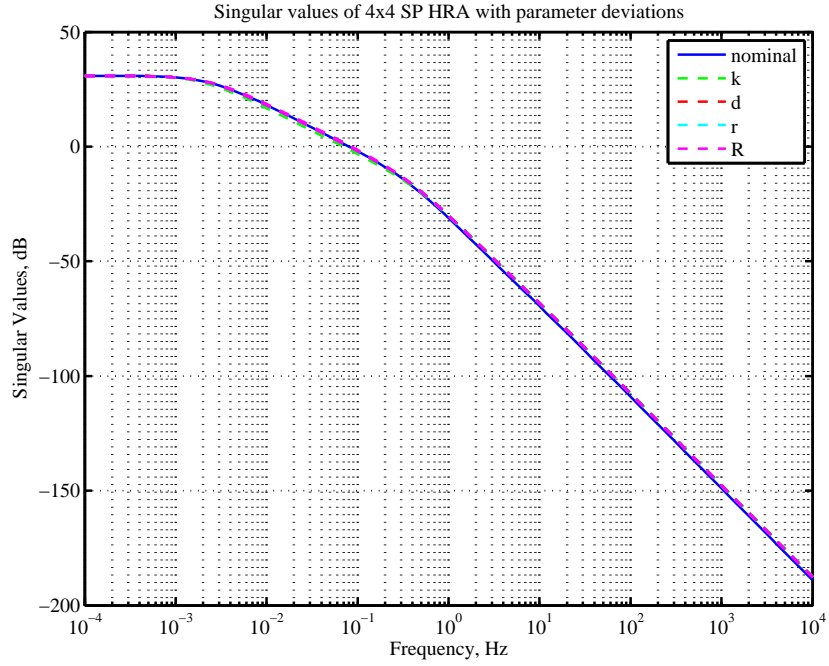


Figure 5.8: Singular values for 4x4 SP HRA with 10% parameter uncertainty.

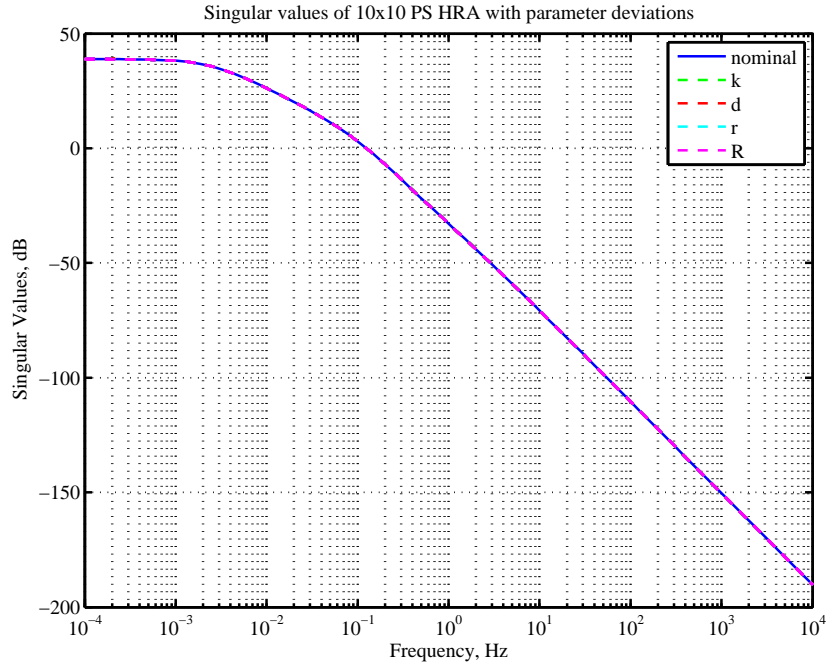


Figure 5.9: Singular values for 10x10 PS HRA with 10% parameter uncertainty.

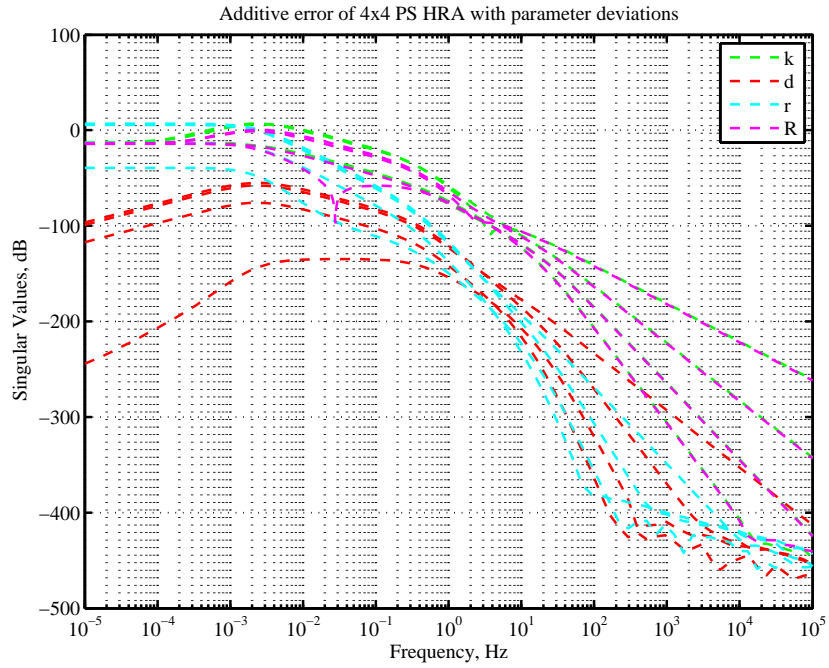


Figure 5.10: Singular values of additive error between nominal and parameter deviated 4x4 PS HRA.

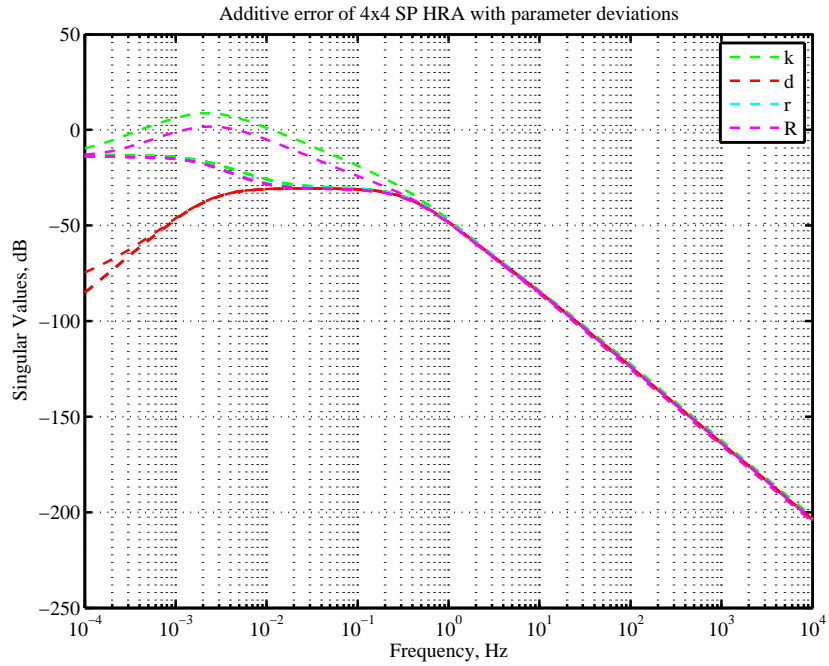


Figure 5.11: Singular values of additive error between nominal and parameter deviated 4x4 SP HRA.

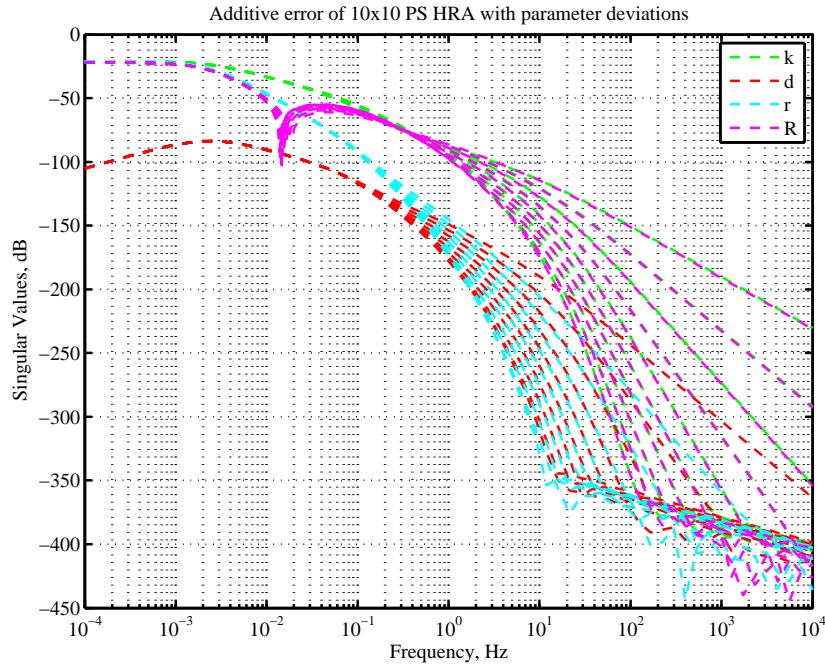


Figure 5.12: Singular values of additive error between nominal and parameter deviated 10x10 PS HRA.

Table 5.4: Stability margins and additive error of 4x4 PS HRA under nominal and fault conditions

System state	PM (deg)	GM (dB)	BW (rads ⁻¹)	$\ G - G_f\ _\infty$ (dB)
4x4 PS HRA				
Nominal	79.2	∞	0.50	-
k	75.8-81.9	∞	0.45-0.56	0.22-2.2
d	79.1	∞	0.50	0.0002-0.0013
r	79.1	∞	0.50	0.011-1.9
R	77.5-80.2	∞	0.48-0.533	0.20-1.05
4x4 SP HRA				
Nominal	79.1	∞	0.51	-
k	80.9-83.6	∞	0.45-0.56	0.22-2.7
d	79.1	∞	0.51	0.030
r	79.1	∞	0.51	0.20
R	79.1-80.6	∞	0.51-0.54	0.20-1.2
10x10 PS HRA				
Nominal	39.98	∞	0.79	-
k	40.02-40.04	∞	0.79	0.088
d	39.98-40.02	∞	0.79	0.001
r	39.98	∞	0.79	0.087
R	39.96-39.98	∞	0.79	0.080

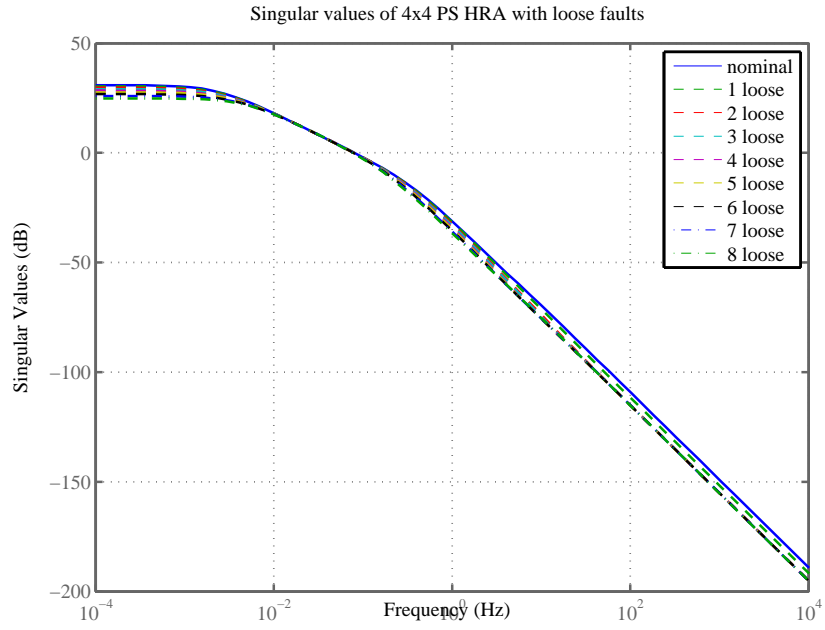


Figure 5.13: Singular values for 4x4 PS HRA with loose faults

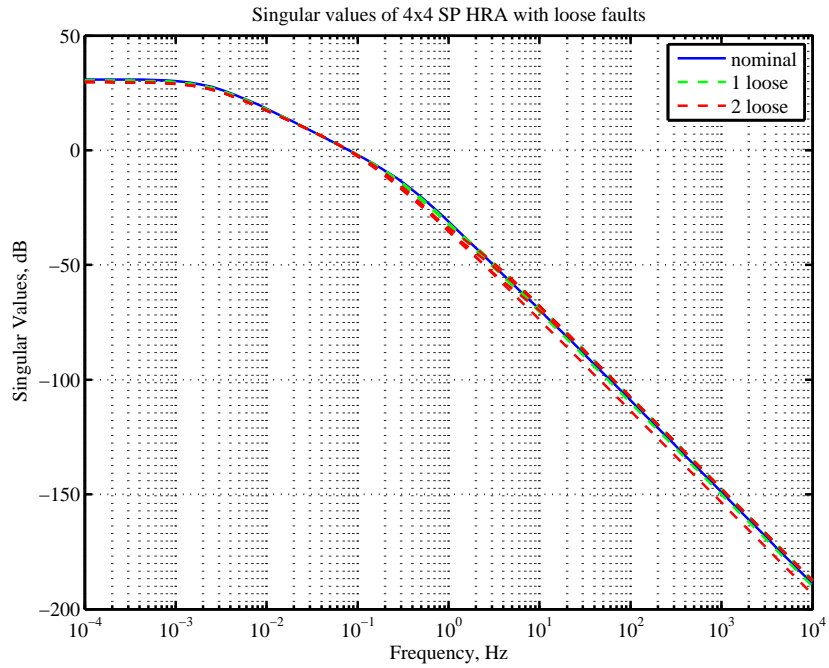


Figure 5.14: Singular values for 4x4 SP HRA with loose faults

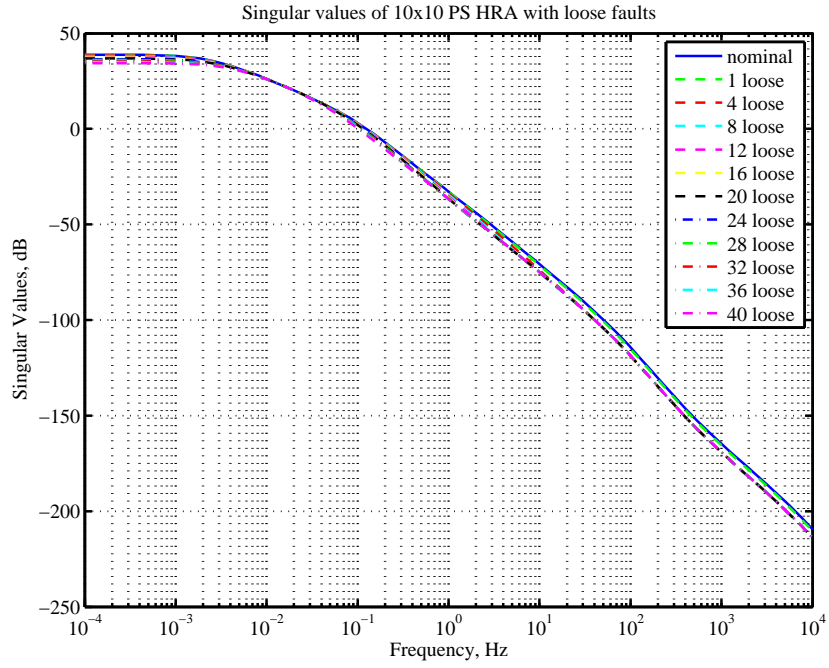


Figure 5.15: Singular values for 10x10 PS HRA with loose faults

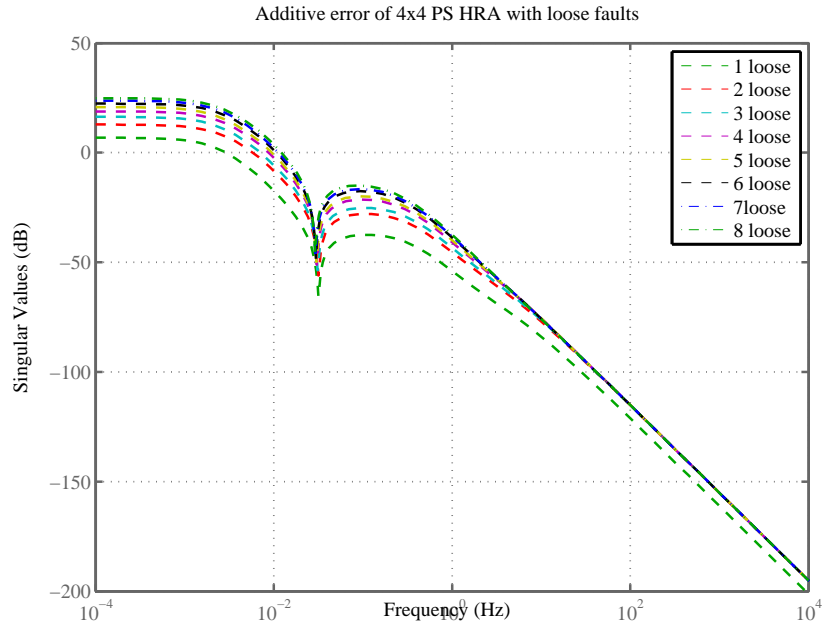


Figure 5.16: Singular values of additive error between nominal and 4x4 PS HRA with loose faults

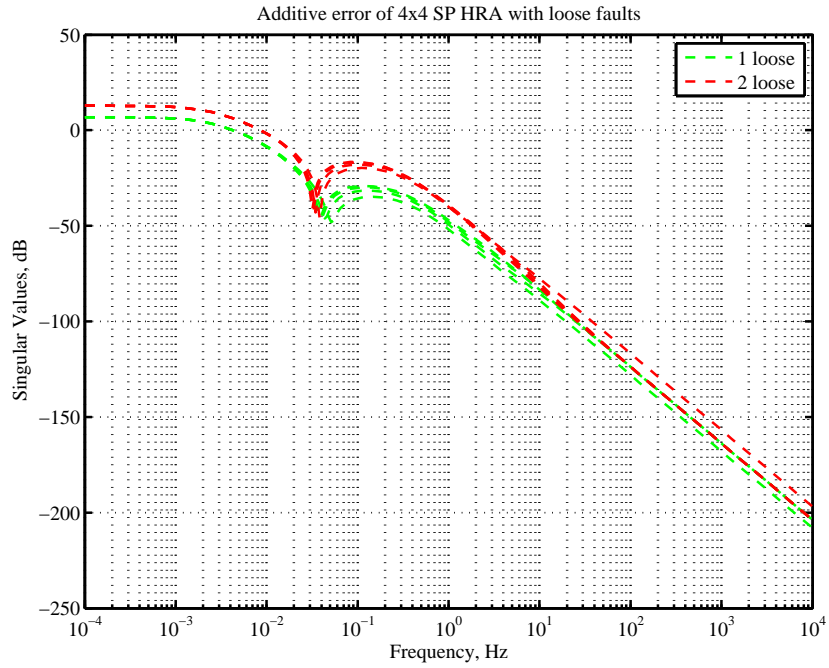


Figure 5.17: Singular values of additive error between nominal and 4x4 SP HRA with loose faults

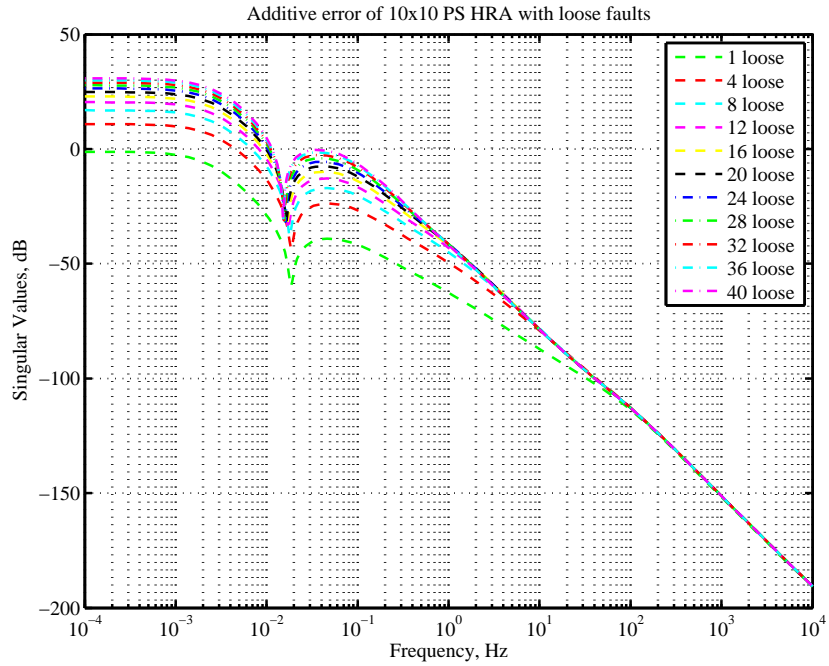


Figure 5.18: Singular values of additive error between nominal and 10x10 PS HRA with loose faults

Table 5.5: Stability margins and additive error of example HRAs under nominal and loose fault conditions

System state	PM (deg)	GM (dB)	BW (rads ⁻¹)	$\ G - G_f\ _\infty$ (dB)
4x4 PS HRA				
Nominal	79.2	∞	0.50	-
1 loose	78.4	75.9	0.50	2.19
2 loose	77.1	49.9	0.50	4.37
3 loose	76.6	49.2	0.50	6.56
4 loose	74.7	49.1	0.50	8.75
5 loose	73.9	52.5	0.50	10.94
6 loose	72.4	∞	0.49	13.12
7 loose	71.6	∞	0.49	15.31
8 loose	70.2	∞	0.48	17.50
4x4 SP HRA				
Nominal	79.1	∞	0.51	-
E11 loose	77.4	∞	0.50	2.19
E11, E12 loose	71.3	∞	0.50	4.37
10x10 PS HRA				
Nominal	40.0	∞	0.79	-
1 loose	39.7	∞	0.79	0.87
4 loose	38.4	∞	0.78	3.50
8 loose	36.9	∞	0.77	7.00
12 loose	35.7	∞	0.75	10.50
16 loose	34.7	∞	0.74	14.00
20 loose	33.9	∞	0.72	17.50
24 loose	33.4	∞	0.71	21.00
28 loose	33.1	∞	0.69	24.50
32 loose	32.9	∞	0.67	28.00
36 loose	32.9	∞	0.66	31.50
40 loose	33.0	∞	0.64	35.00

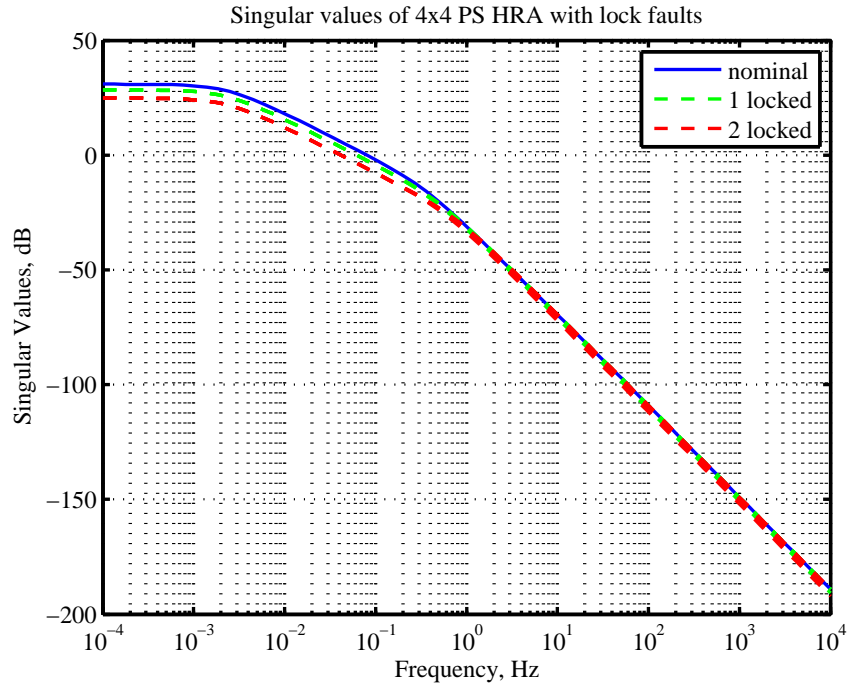


Figure 5.19: Singular values for 4x4 PS HRA with lock-up faults

5.2.4 Lock-up Faults

The singular values of the example systems under lock-up faults are shown in Figures 5.19 to 5.21 and their additive errors in Figures 5.22 to 5.24. Table 5.6 contains the phase margins and infinity norm additive errors for these systems.

In general, the infinity norm of the additive errors are much larger than those incurred by the loose faults, indicating that the lock-up fault is the most severe of the fault types examined within both the PS and SP configurations. The error levels in the low frequency region are comparable to those incurred by loose faults, again suggesting steady-state changes. However, larger errors around the system bandwidth are created by lock-up faults. Significant decreases in the bandwidth indicate a slower system response. This is logical, as lock faults effectively increase the amount of mass in the system. The lock-up faults increase the phase margin of the system, hence do not present a problem for stability.

When comparing the affect of lock-up faults between the example systems, a similar trend to the loose faults can be observed. The maximum PM changes are similar in each case, however, more lock-ups are present in the SP system than in the PS to reach this maximum. This is because serial elements provide more tolerance to lock-up faults. Comparing the PS configurations, it can be seen that a single lock-up fault has approximately half the effect on the 10×10 system than the 4×4 system in terms of maximum additive error. This is due to increased element numbers reducing fault severity.

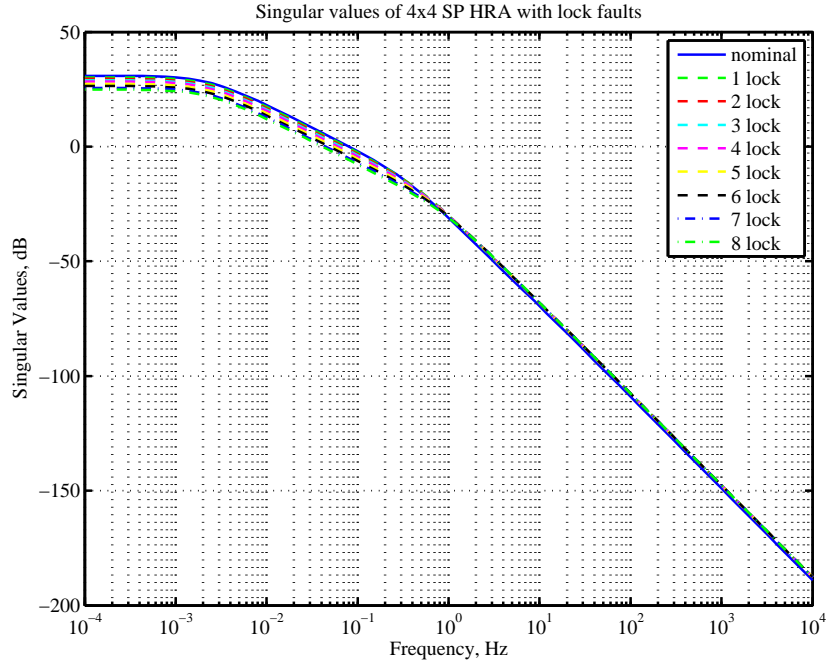


Figure 5.20: Singular values for 4x4 SP HRA with lock-up faults

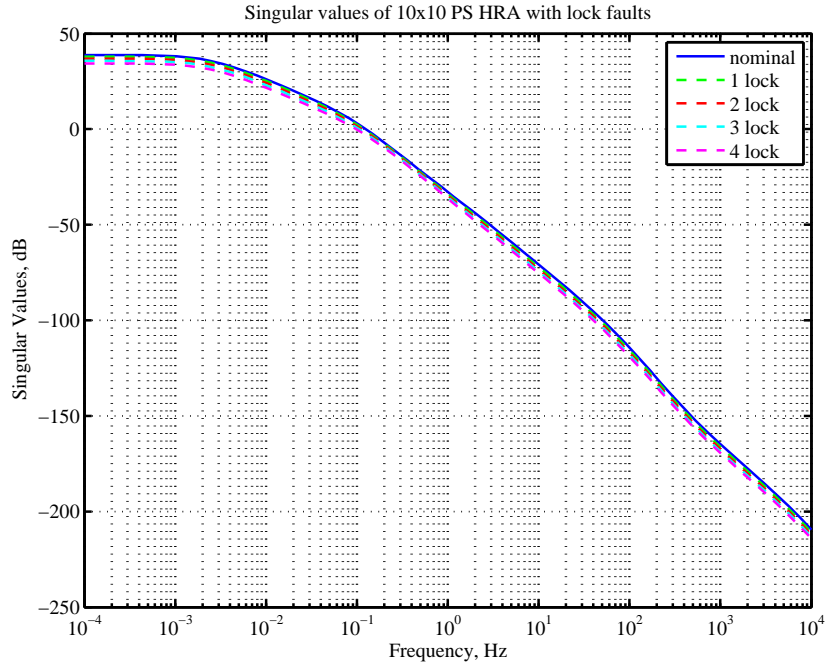


Figure 5.21: Singular values for 10x10 PS HRA with lock-up faults

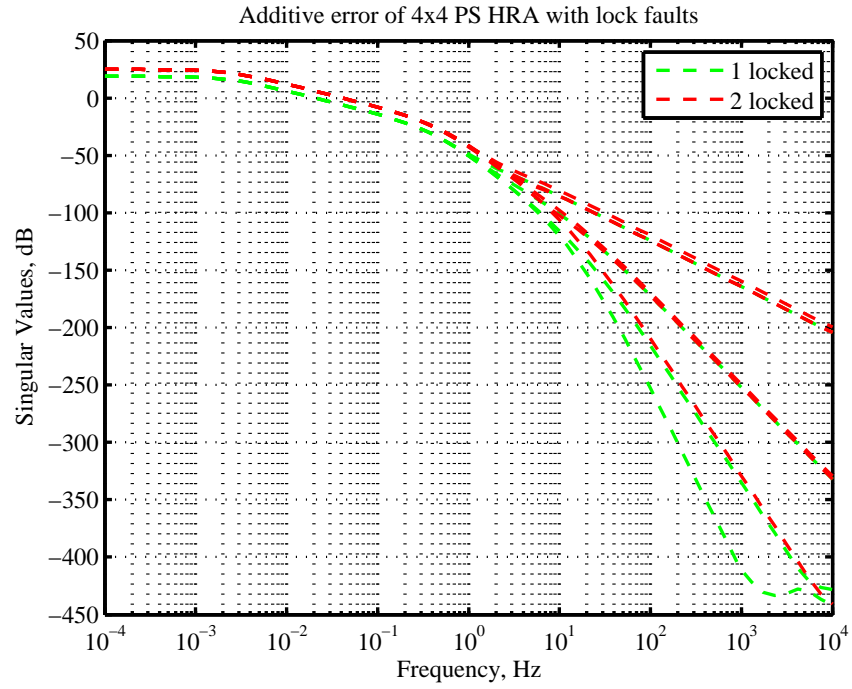


Figure 5.22: Singular values of additive error between nominal 4x4 PS HRA and 4x4 PS HRA with lock-up faults

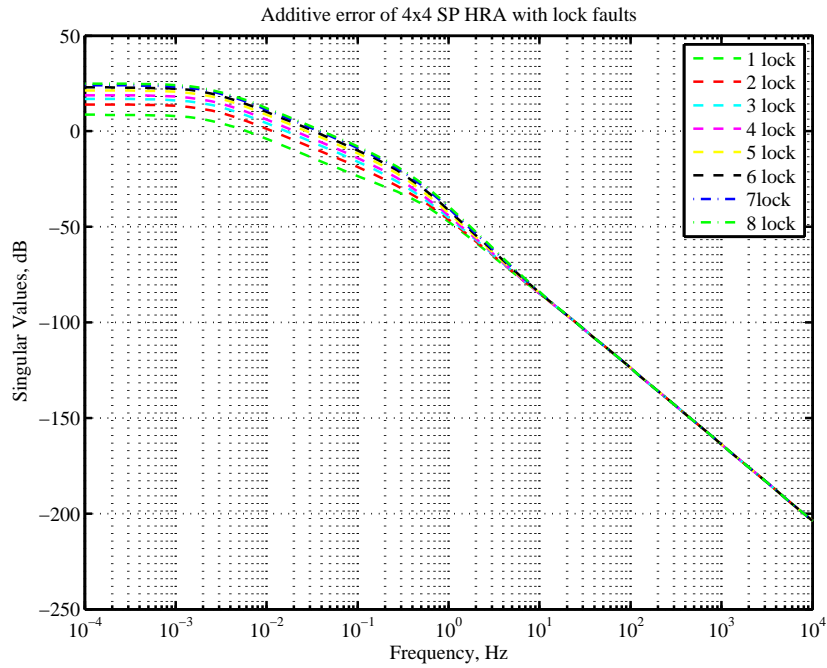


Figure 5.23: Singular values of additive error between nominal 4x4 SP HRA and 4x4 SP HRA with lock-up faults

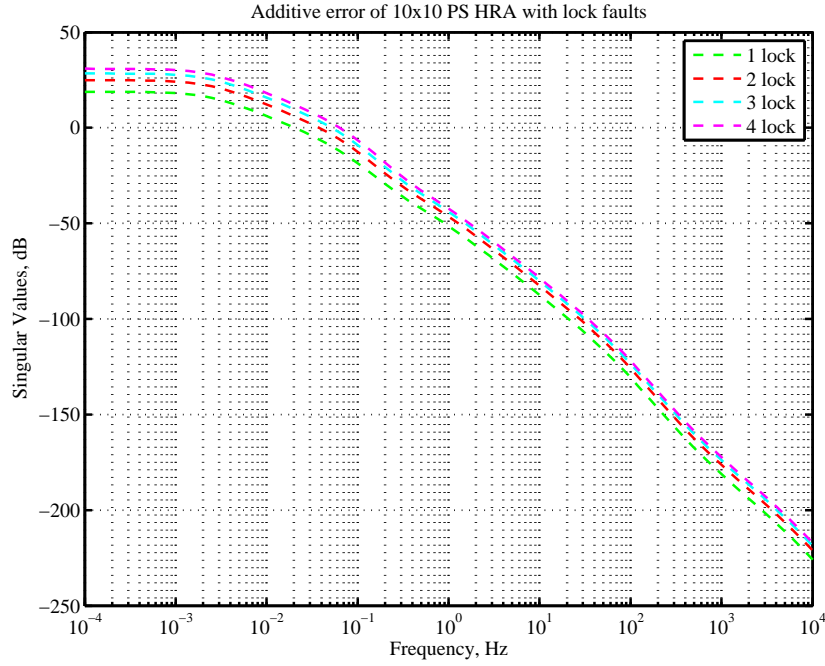


Figure 5.24: Singular values of additive error between nominal 10x10 PS HRA and 10x10 PS HRA with lock-up faults

Table 5.6: Stability margins and infinity norm of additive error for example HRAs under nominal and lock-up fault conditions

System state	PM (deg)	GM (dB)	BW (rads ⁻¹)	$\ G - G_f\ _\infty$ (dB)
4x4 PS HRA				
Nominal	79.2	∞	0.50	-
1 lock	85.4	∞	0.39	18.8
2 lock	90.4	∞	0.26	24.90
4x4 SP HRA				
Nominal	79.1	∞	0.51	-
1 lock	82.5	∞	0.47	3.65
2 lock	84.0	∞	0.44	8.84
3 lock	85.3	∞	0.41	13.17
4 lock	86.3	∞	0.39	18.71
5 lock	87.9	∞	0.34	21.75
6 lock	89.1	∞	0.31	26.14
7 lock	90.0	∞	0.28	38.10
8 lock	91.0	∞	0.26	35.08
10x10 PS HRA				
Nominal	40.0	∞	0.79	-
1 lock	41.7	∞	0.75	8.71
2 lock	44.3	∞	0.71	17.38
3 lock	48.0	∞	0.68	25.70
4 lock	53.9	∞	0.61	35.01

5.2.5 System Analysis Summary

The system analysis has produced a number of points that should be considered in the control design. The nominal 4×4 systems have sufficient stability margins, but an increase in bandwidth is required. The nominal 10×10 system requires both an increase in the phase margin and bandwidth however.

The faults analysed have a varied affect on the example systems. Parameter deviations in each case have a relatively small influence and should not cause issues. Loose faults, however, have a greater influence. They introduce steady-state errors and decrease the phase margin. An extra 10° of phase margin should be provided to accommodate this. Lock-up faults are the most severe fault type. They decrease the bandwidth of the system and increase the phase margin, and as such are not a threat to stability, but will affect the speed of the system.

Thus, loose and lock-up faults have conflicting effects, and as such, a trade-off exists between designing for loose fault and lock-up fault performance. The control may be made more conservative to accommodate loose faults more easily, however, this will heighten the negative influences of lock-up faults. Likewise, more phase could be introduced into the control law to make the lock-up faults more tolerable, at the expense of decreasing the tolerance to loose faults. Hence, it is a question of which fault has the highest probability of occurrence in an application as to the adjustments that could be made. However, assuming that they of equal likelihood, designing control based on the nominal system is a simple compromise, and is the approach taken in the following section. Alternatively, an optimisation approach akin to that presented in (23) could be followed to achieve a more suitable controller. However, as passive control is not the main contribution of this work, the simple approach is taken.

The system analysis has also demonstrated some key features of the HRA concept. It has been shown that faults have a smaller affect on the 10×10 system in comparison to the 4×4 configurations, as greater numbers of elements reduce the influence of element faults. Secondly, it has been demonstrated that serial elements increase the accommodation of loose faults, and parallel elements the tolerance to lock-up faults.

5.3 Control Design

Given the conclusions of the system analysis in Section 5.2, the design of control laws (according to the structure of Figure 5.2) for each of the three example systems is detailed in this section.

5.3.1 4x4 PS HRA

A simple phase advance controller is designed to introduce gain and phase into the system. The control law is given in equation 5.1. A zero is placed at -2.25 to cancel a pole in that region, and the pole is placed approximately 12 times faster to increase the phase margin by approximately 10° . This extra phase margin is then used to increase the gain of the system.

$$G_{PA} = 11 \frac{0.45s + 1}{0.038s + 1} \quad (5.1)$$

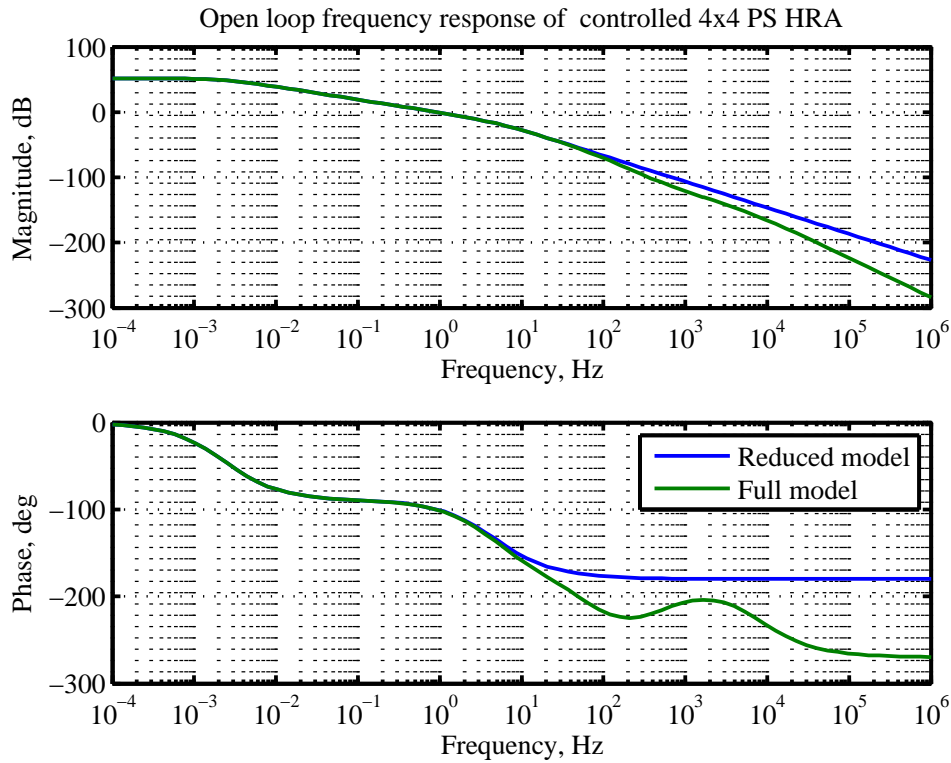


Figure 5.25: Frequency response of 4x4 PS HRA with passive control

The resultant open loop frequency response of the reduced order system, on which the control is based, and the full order system is shown in Figure 5.25 and Table 5.7 gives the associated characteristics. It can be seen that the difference between the reduced order and full order system is very small in the low and mid frequency regions, having little affect on the phase margin, bandwidth or transient characteristics of the system. The phase in the full order system, however, does fall below -180° degree, and as such the full order system has a gain margin of 41dB. This margin is adequate however, and is not considered a problem.

A step response using the full order system is given in Figure 5.26. A step input of the full travel position reference (0.03m) is applied at $t = 0$. The voltage input to the elements is well within the physical limits of the system. The load position and the relative positions of the elements is given in the second subplot. The travel is equally distributed between the elements, however the elements nearer the load move more quickly. This is due to the force produced by those elements effectively working on less mass than those elements nearer the fixed surface. This is a phenomenon that should be less prevalent in HRAs that contain increased inter-mass to load mass ratios. A summary of the load position response transient characteristics is given in Table 5.7.

5.3.2 4x4 SP HRA

As the nominal characteristics of the 4×4 SP HRA are the same as the 4×4 PS system, and the control is designed to suit the nominal case (as it is assumed that loose and lock-up

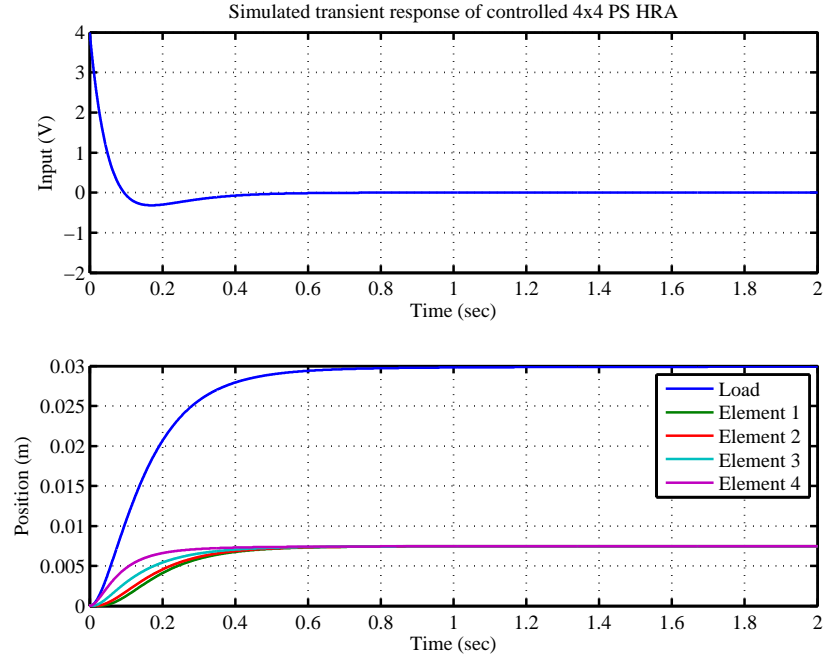


Figure 5.26: Step response of 4x4 PS HRA with passive control

Table 5.7: Stability Margins and transient characteristics of globally controlled example HRAs

	PM (deg)	GM (dB)	BW (rads ⁻¹)	SSerr (%)	RT (s)	ST (s)	OS (%)
4x4 PS HRA	80.0	41.5	5.76	0	0.31	0.58	0
4x4 SP HRA	80.0	41.5	5.71	0	0.31	0.58	0
10x10 PS HRA	73.7	47.5	3.19	0	0.47	0.79	0.05

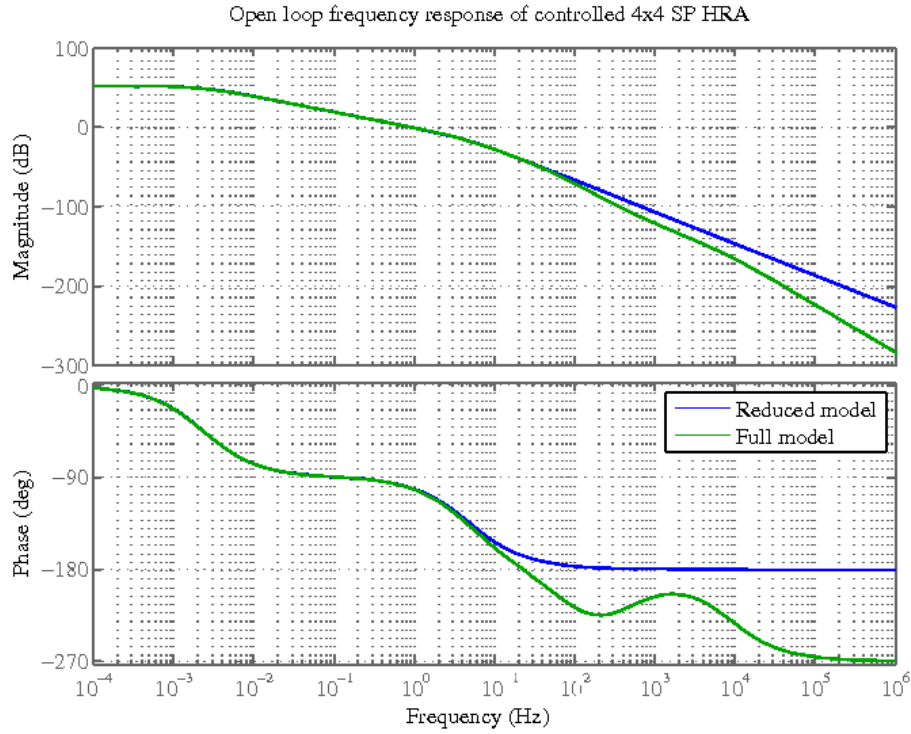


Figure 5.27: Frequency response of 4x4 SP HRA with passive control

faults are equally likely), then the control designed in the PS example (see equation 5.1) may also be applied here.

The resultant open loop frequency response for the reduced order SP system and its full-order representation are shown in Figure 5.27 and the simulated step response of the full order system is given in Figure 5.28. The results are the same as that for the PS system, as further shown by Table 5.7.

5.3.3 10x10 PS HRA

A similar approach to that of the 4×4 systems is taken to controlling the 10×10 PS HRA. The phase advance controller designed is given in equation 5.2. The zero is placed to cancel a pole at -0.5952, and the pole of the phase advance is placed 12 times faster. The gain is then increased to improve the steady-state errors, within the overshoot limit.

$$G_{PA} = 2.5 \frac{1.68s + 1}{0.14s + 1} \quad (5.2)$$

The open loop frequency response of the controlled reduced order and full order system is given in Figure 5.29. Again, the low to mid frequency region are very similar, resulting in little change to the phase margin, bandwidth and transient characteristics of the system. The full order system does not have infinite gain margin however, as shown in Table 5.7. Nevertheless, the gain margin is ample.

The simulated step response of the controlled full order system is given in Figure 5.30.

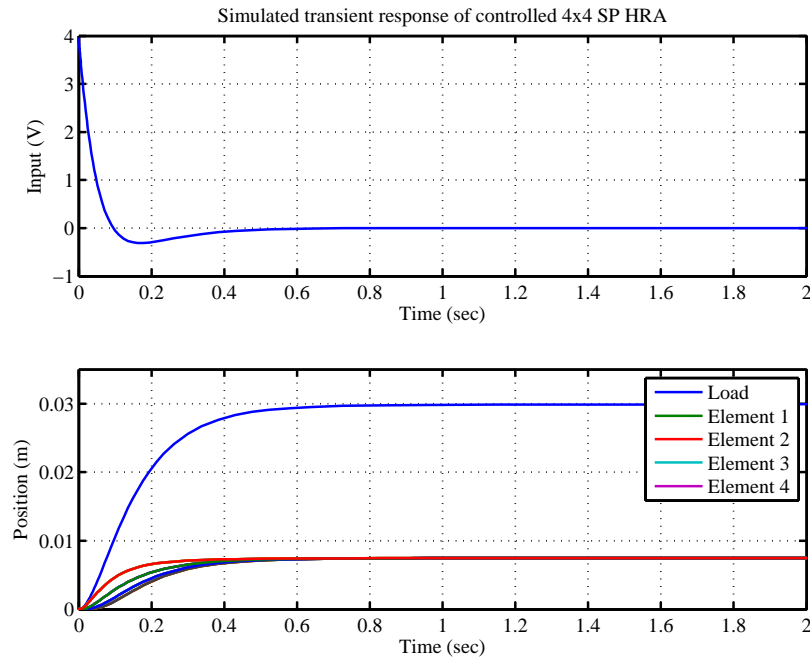


Figure 5.28: Step response of 4x4 SP HRA with passive control

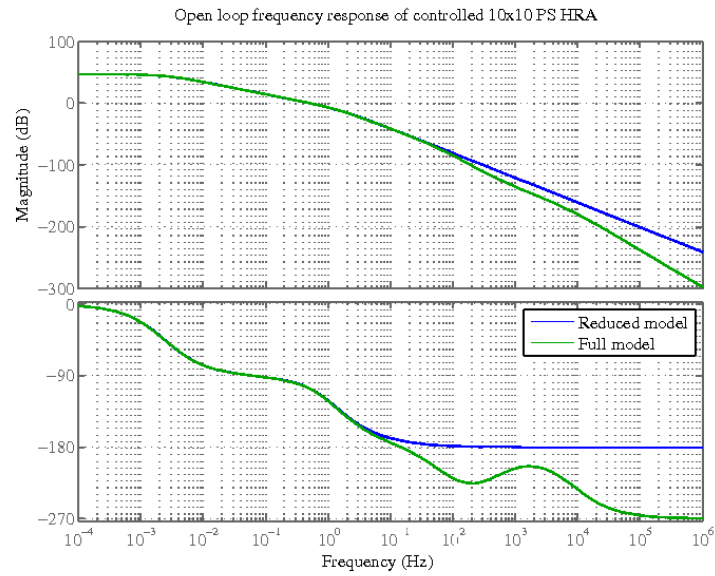


Figure 5.29: Frequency response of 10x10 PS HRA with passive control

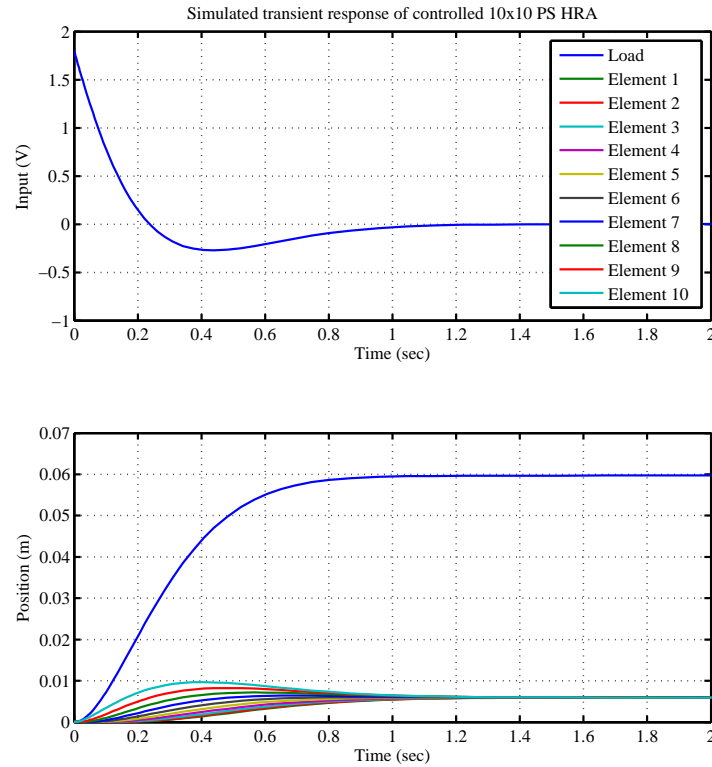


Figure 5.30: Step response of 10x10 PS HRA with passive control

A step input of the full travel position reference (0.06m) is applied at $t = 0$. The resultant voltage input to the system is well within the physical limits of the system. The load position and the relative position of the inter-element masses are given in the second subplot. Equal distribution of the travel is again evident.

5.3.4 Control Design Summary

The design of simple phase advance controllers for the three example HRA systems has been detailed. The controlled systems meet the requirements (Chapter 4) for the nominal system. The next section will examine their performance under fault conditions.

5.4 Fault Simulation

5.4.1 Parameter Deviations

Figures 5.31 to 5.36 show the singular values and simulated step response of the controlled parameter deviated systems in comparison to the nominal case and Table 5.8 summarises these results. The infinity norm of the closed-loop additive error ($\|\Delta G_{CL}\|_{\infty} = \|G_{CL} - G_{CLf}\|_{\infty}$) is greatly reduced in comparison to the uncontrolled system, as one would expect. The effects on system behaviour are minimal in each case, and the system remains within the performance requirements.

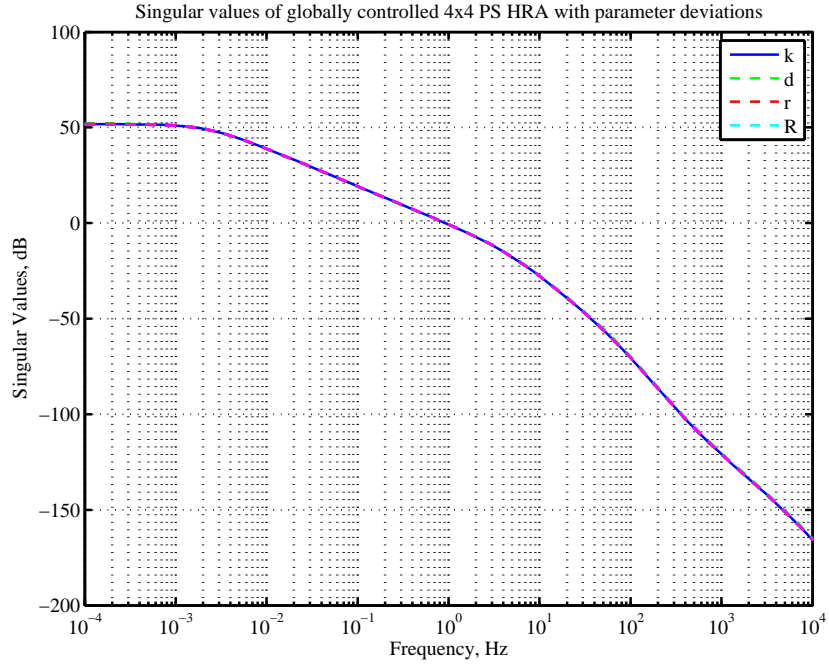


Figure 5.31: Singular values for controlled 4x4 PS HRA with 10% parameter uncertainty

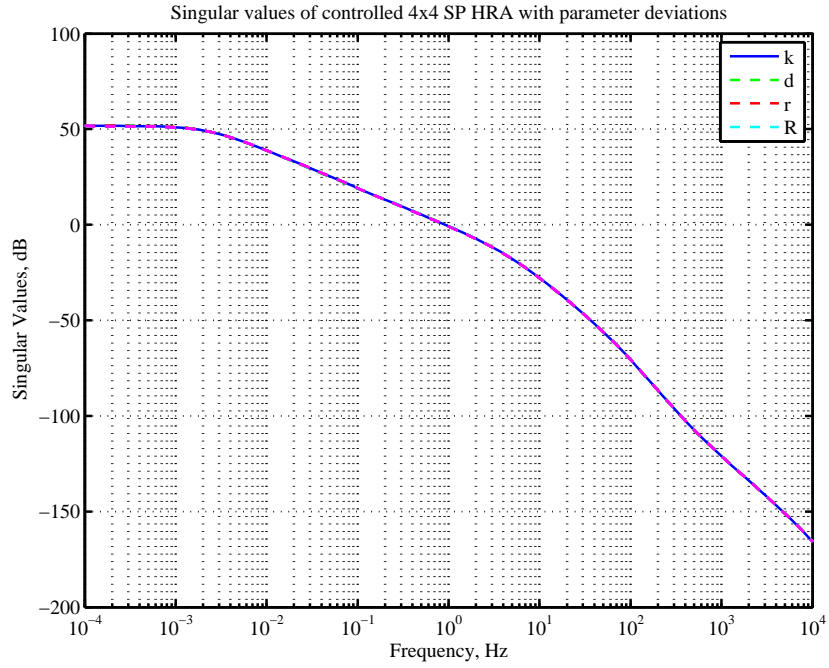


Figure 5.32: Singular values for controlled 4x4 SP HRA with 10% parameter uncertainty

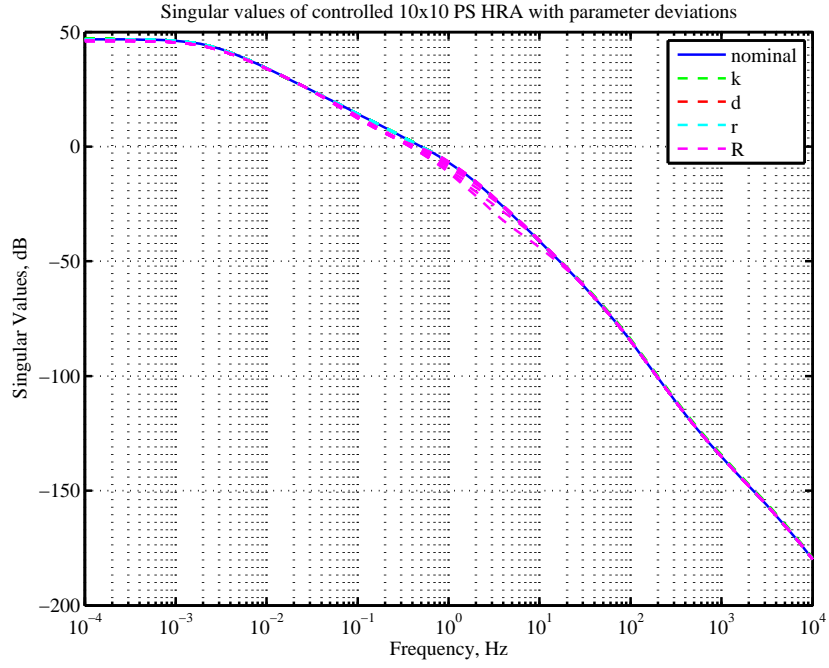


Figure 5.33: Singular values for controlled 10x10 PS HRA with 10% parameter uncertainty

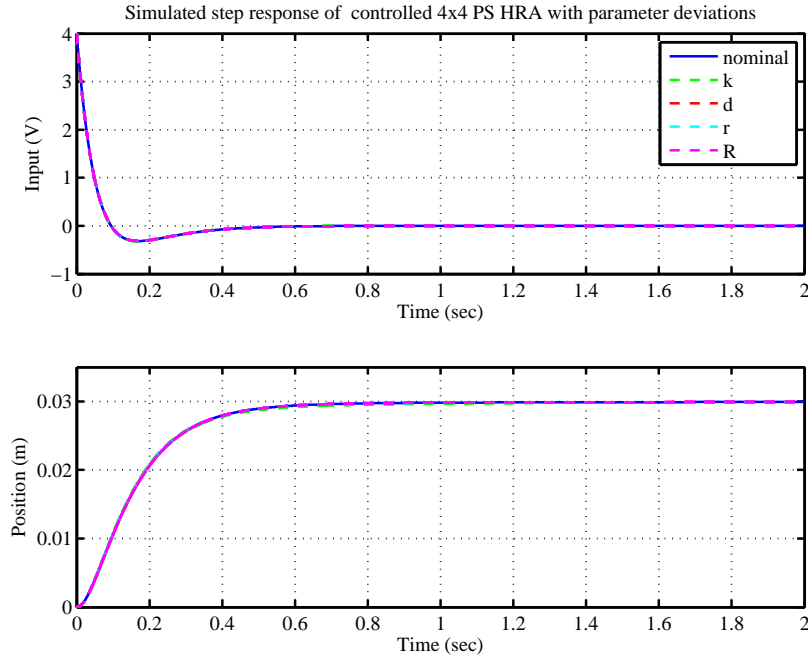


Figure 5.34: Step response of 4x4 PS HRA with passive control and parameter deviations

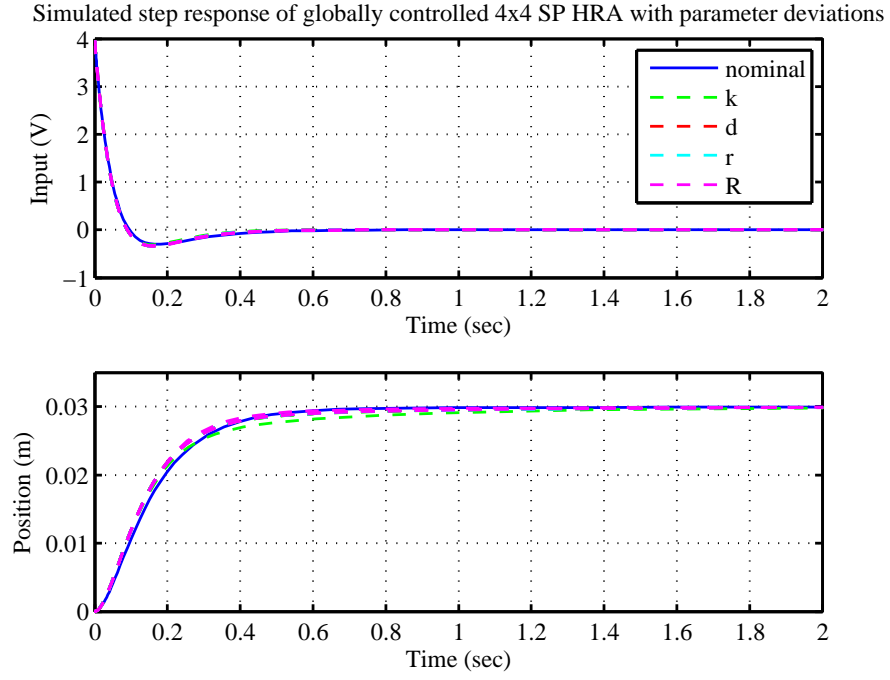


Figure 5.35: Step response of 4x4 SP HRA with passive control and parameter deviations

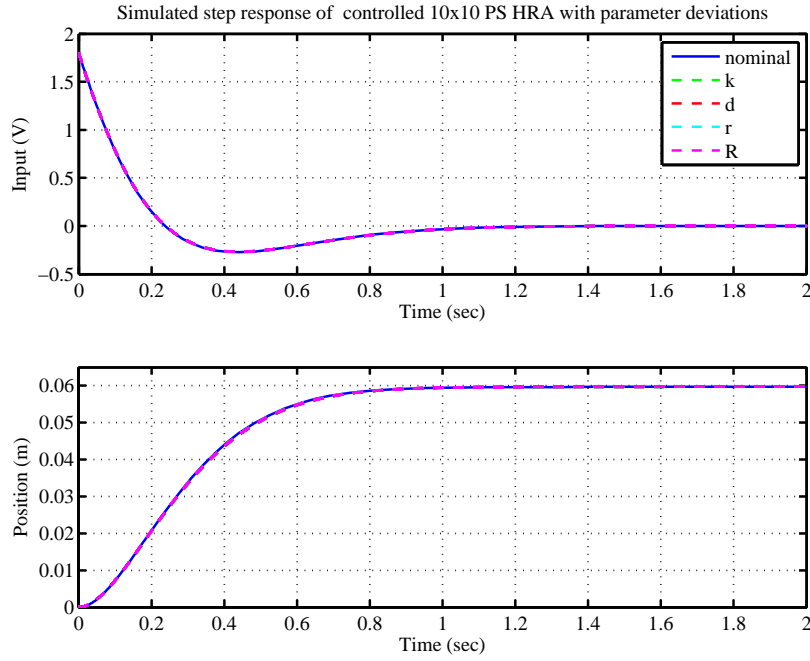


Figure 5.36: Step response of 10x10 PS HRA with passive control and parameter deviations

Table 5.8: Stability margins and transient characteristics of example HRAs under nominal and fault conditions

4x4 PS HRA System state	PM (deg)	GM (dB)	BW (rads ⁻¹)	$\ Gf_{CL}\ _{\infty}$ (dB)	SSerr (%)	RT (s)	ST (s)	OS (%)
4x4 PS HRA								
Nominal	80.0	41.5	5.76	-	0	0.31	0.58	0
k	79.8-80.4	40.5-42.0	5.74-5.79	0.012-0.016	0	0.30-0.31	0.63-0.65	0
d	80	41.5	5.76	0	0	0.31	0.58	0
r	80	41.5	5.76	0	0	0.31	0.58	0
R	78.7-80.8	41.4-41.6	5.72-5.74	0.0034-0.0037	0	0.31	0.58	0
4x4 SP HRA								
Nominal	80.0	41.5	5.71	-	0	0.31	0.58	0
k	81.4-85.2	41.2-41.4	6.29-6.45	0.086-0.090	0	0.30-0.37	0.80-1.16	0
d	81.4	41.5	6.34	0.082	0	0.30	0.77	0
r	81.3	41.5	6.34	0.082	0	0.30	0.77	0
R	81.3-79.7	41.4-41.7	6.31	0.079	0	0.28-0.31	0.58-0.76	0
10x10 PS HRA								
Nominal	73.7	47.5	3.19	-	0	0.47	0.79	0.05
k	73.93-74.05	46.6-49.6	3.16	0.0011	0	0.48	0.80	0.12
d	73.98	47.5	3.16	0	0	0.48	0.80	0.13
r	73.98	47.5	3.16	0.0006	0	0.48	0.80	0.13
R	73.93-74.03	46.6-51.5	3.16	0.0013	0	0.48-0.49	0.80	0.13

5.4.2 Loose Faults

Figures 5.37 to 5.42 give the singular values and simulated step response of the controlled 4x4 PS, SP and 10×10 PS HRAs with loose faults, which are summarised in Table 5.9.

A significant bandwidth decrease is still present and as such the transient characteristics are affected. In particular, the overshoot increase in the system is substantial, exceeding the requirement limit in most cases. One remedy to this issue may be to increase the phase advance in the control. However, this will make the nominal performance more conservative and the phase advance ratio is already relatively high.

The SP HRA transient response exhibits the effects of the non-linearities in the system. As the branches become increasingly loose, then the loose element coils hit the end-stops causing the large transients seen in Figure 5.42. This phenomena can be more clearly seen in Figure 5.43, which gives the SP 4×4 system response to a pulse train, whilst there are two loose faults in elements e_{43} and e_{44} i.e. two branches are loose. The relative positions of all the elements are also given in this figure.

On applying the initial voltage it can be seen that the force exerted by the remaining active elements in the loose branches (cyan), causes the loose elements (red) to hit their nearside end-stop. Likewise, as the elements decelerate, the loose elements continue to travel and hit the far-side end-stop. The uncontrollability of the loose element is exacerbated by the control driving the remaining active elements in the loose branch. This phenomena will be further discussed in Chapter 7.

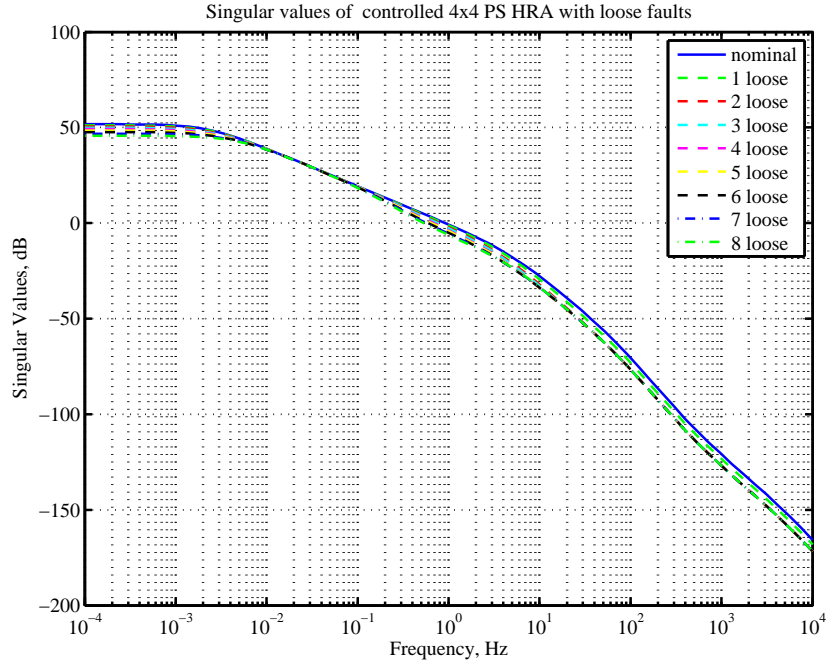


Figure 5.37: Singular values for controlled 4x4 PS HRA with loose faults

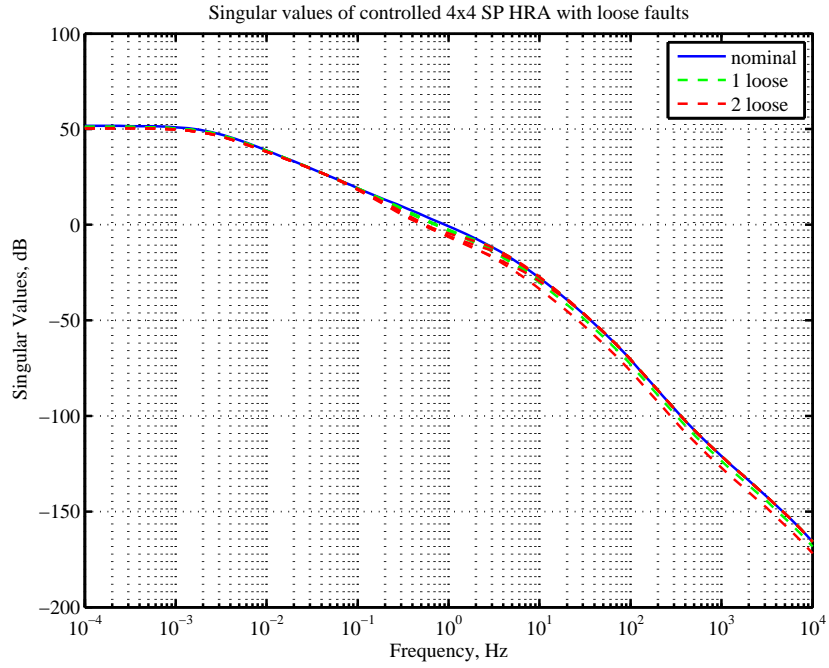


Figure 5.38: Singular values for controlled 4x4 SP HRA with loose faults

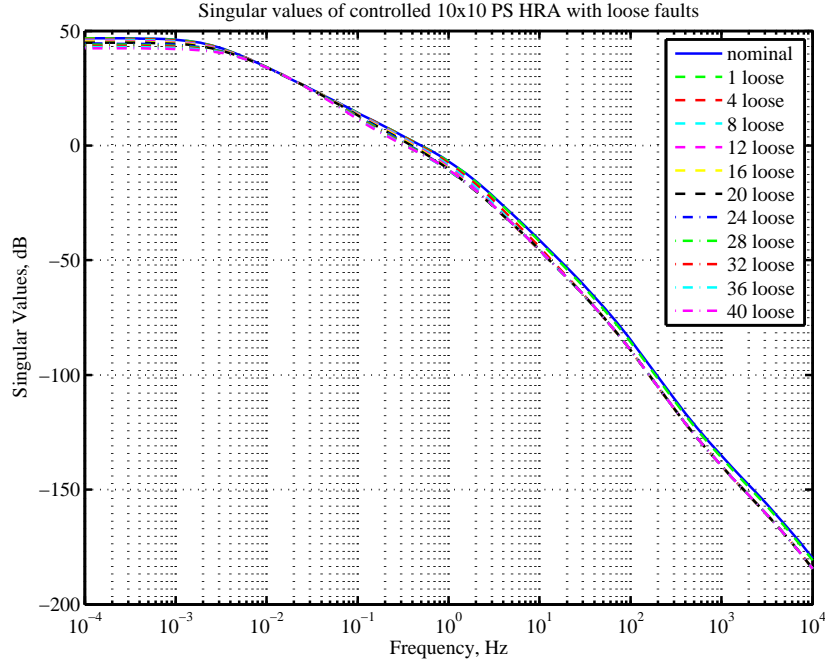


Figure 5.39: Singular values for controlled 10x10 PS HRA with loose faults

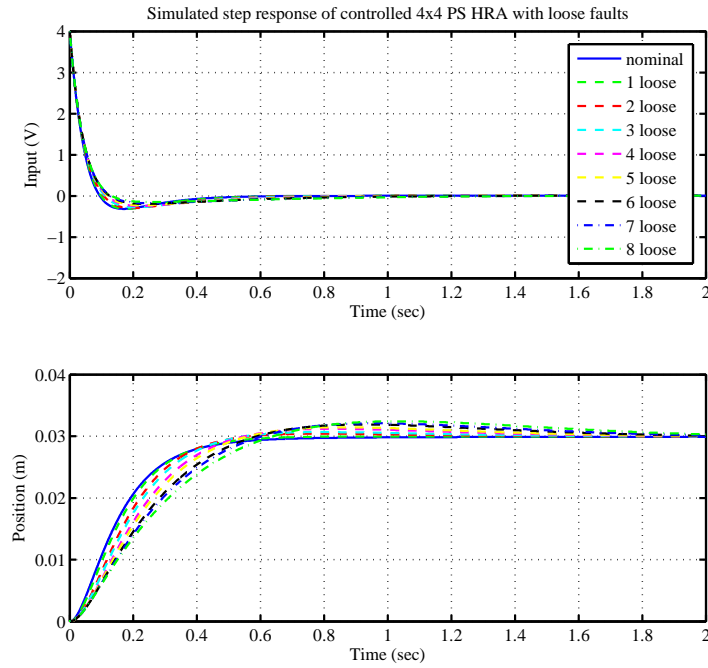


Figure 5.40: Step response of 4x4 PS HRA with passive control and loose faults

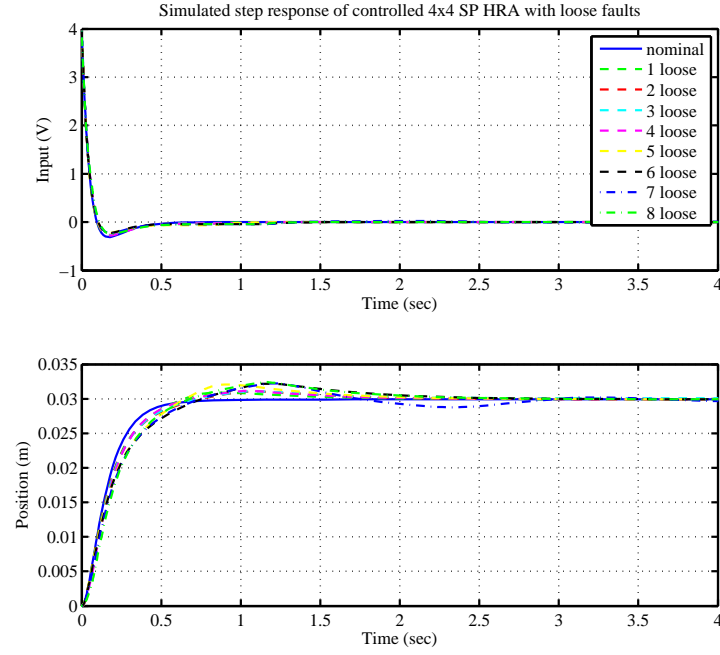


Figure 5.41: Step response of 4x4 SP HRA with passive control and loose faults

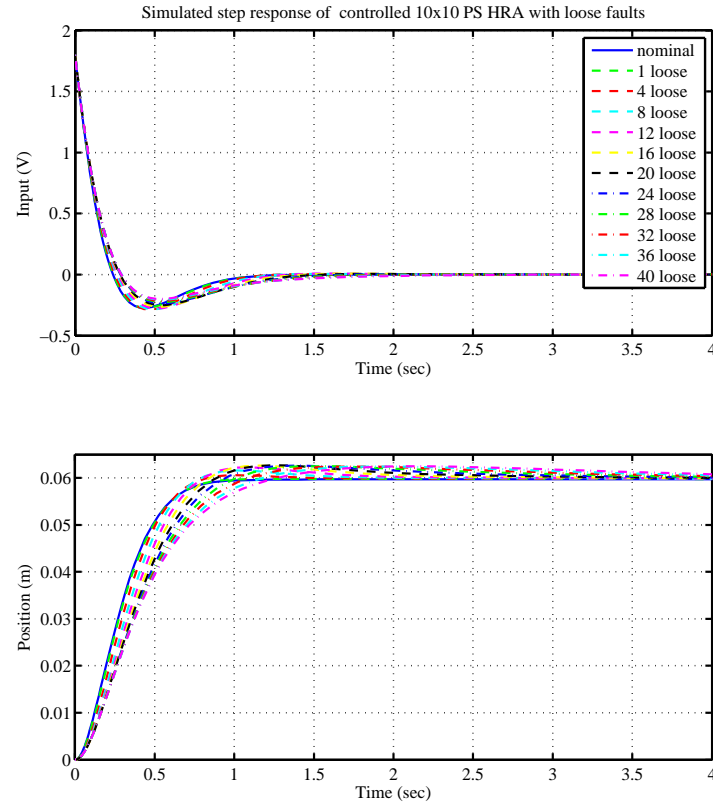


Figure 5.42: Step response of 10x10 PS HRA with passive control and loose faults

Table 5.9: Stability margins and transient characteristics of example HRAs under nominal and loose fault conditions

4x4 PS HRA System state	PM (deg)	GM (dB)	BW (rads ⁻¹)	$\ G_{f_{CL}}\ _{\infty}$ (dB)	SSerr (%)	RT (s)	ST (s)	OS (%)
4x4 PS HRA								
Nominal	80.0	41.5	5.76	-	0	0.31	0.58	0
1 loose	77.4	39.7	5.49	0.06	0	0.30	0.53	0.13
2 loose	74.0	37.8	4.99	0.15	0	0.31	0.56	1.62
3 loose	73.1	40.6	4.73	0.19	0	0.32	1.02	2.57
4 loose	71.6	45.7	4.30	0.26	0	0.34	1.34	4.36
5 loose	71.2	46.1	4.09	0.29	0	0.36	1.46	5.22
6 loose	70.4	46.3	3.75	0.34	0	0.39	1.66	6.75
7 loose	70.2	46.2	3.60	0.36	0	0.40	1.75	7.44
8 loose	70.0	46.1	3.34	0.39	0	0.43	1.91	8.62
4x4 SP HRA								
Nominal	79.3	41.5	5.73	-	0	0.31	0.57	0
1 loose	75.99	44.55	4.44	0.16	0	0.40	1.43	2.99
2 loose	69.48	46.68	3.21	0.38	0	0.53	2.26	8.51
10x10 PS HRA								
Nominal	73.7	47.5	3.19	-	0	0.47	0.79	0.13
1 loose	73.1	45.8	3.14	0.02	0	0.48	0.77	0.16
4 loose	70.1	39.0	3.02	0.10	0	0.47	0.72	1.45
8 loose	68.2	45.2	2.86	0.18	0	0.48	1.33	3.15
12 loose	67.2	50.5	2.71	0.24	0	0.51	1.59	4.30
16 loose	67.2	50.6	2.58	0.27	0	0.57	1.85	4.87
20 loose	67.9	50.3	2.46	0.30	0	0.57	2.18	5.06
24 loose	68.9	50.2	2.35	0.30	0	0.60	2.59	4.98
28 loose	70.4	50.2	2.26	0.31	0	0.63	3.01	4.83
32 loose	72.1	50.2	2.18	0.31	0	0.66	3.40	4.74
36 loose	73.9	50.2	2.11	0.30	0	0.69	3.75	4.80
40 loose	75.7	50.2	2.06	0.30	0	0.72	4.08	4.91

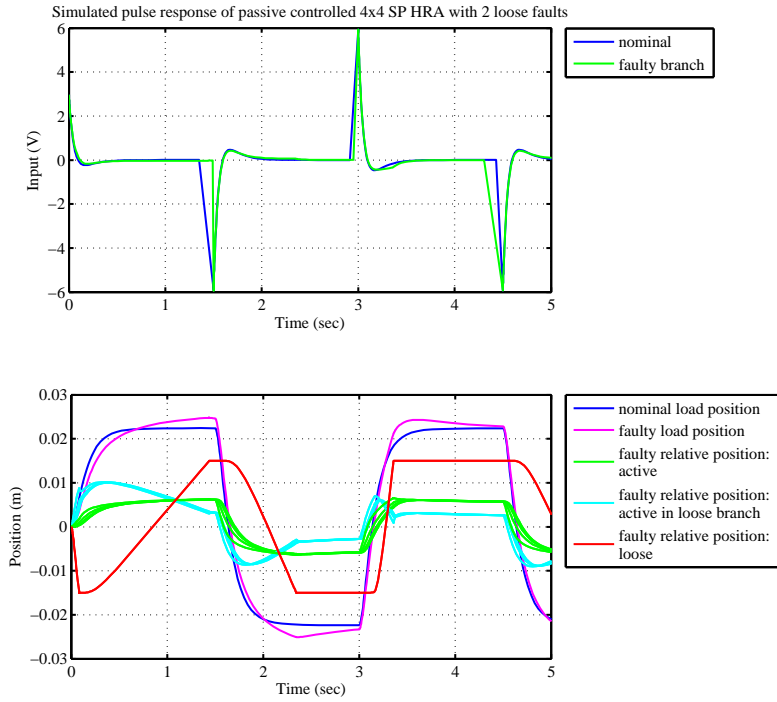


Figure 5.43: Response of SP 4×4 HRA with two loose faults in two branches to pulse train input.

5.4.3 Lock-up Faults

The frequency response and additive error of the locked systems are given in Figures 5.44 and the transient response in Figure 5.47. These plots and Table 5.10 show that the additive error of lock-up faults is reduced in the controlled system. As in the case of loose faults however, the rise time and settling time rise considerably.

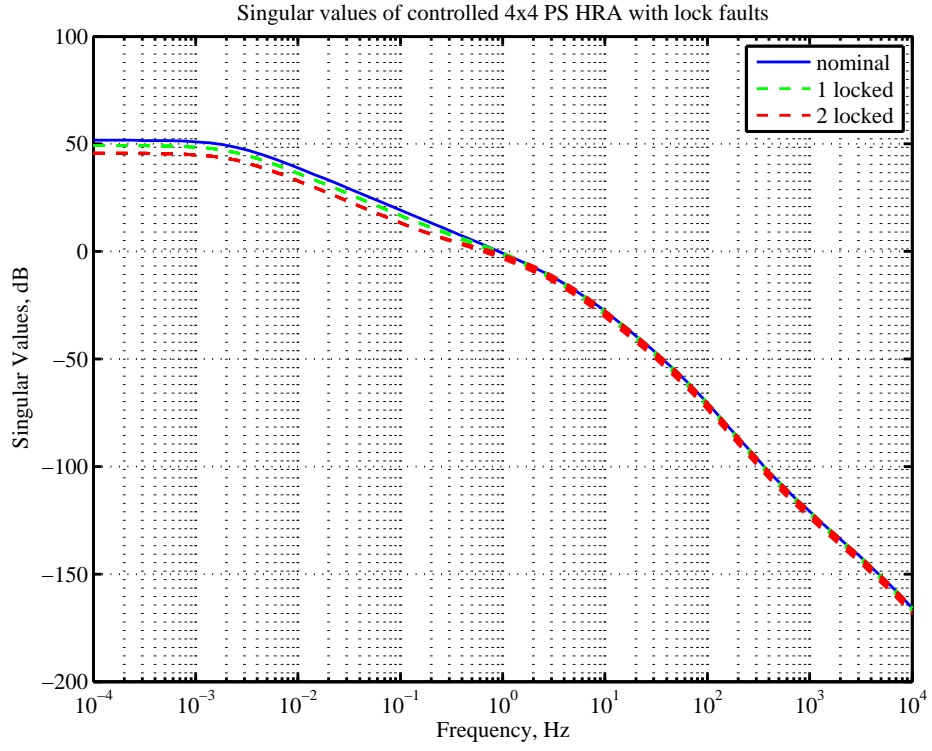


Figure 5.44: Singular values for controlled 4x4 PS HRA with lock-up faults

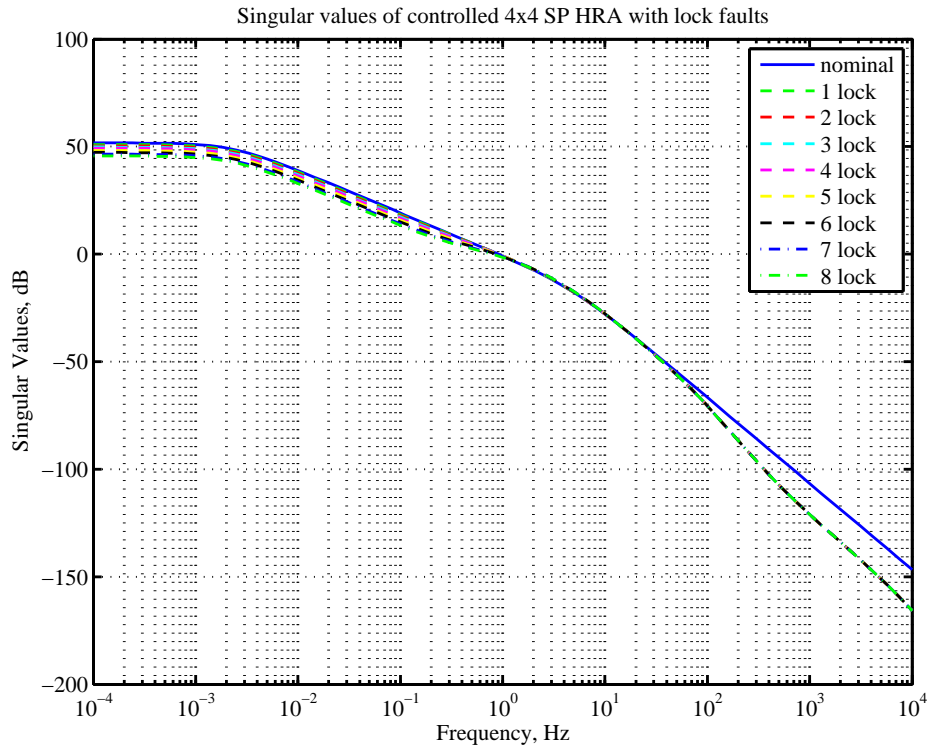


Figure 5.45: Singular values for controlled 4x4 SP HRA with lock-up faults

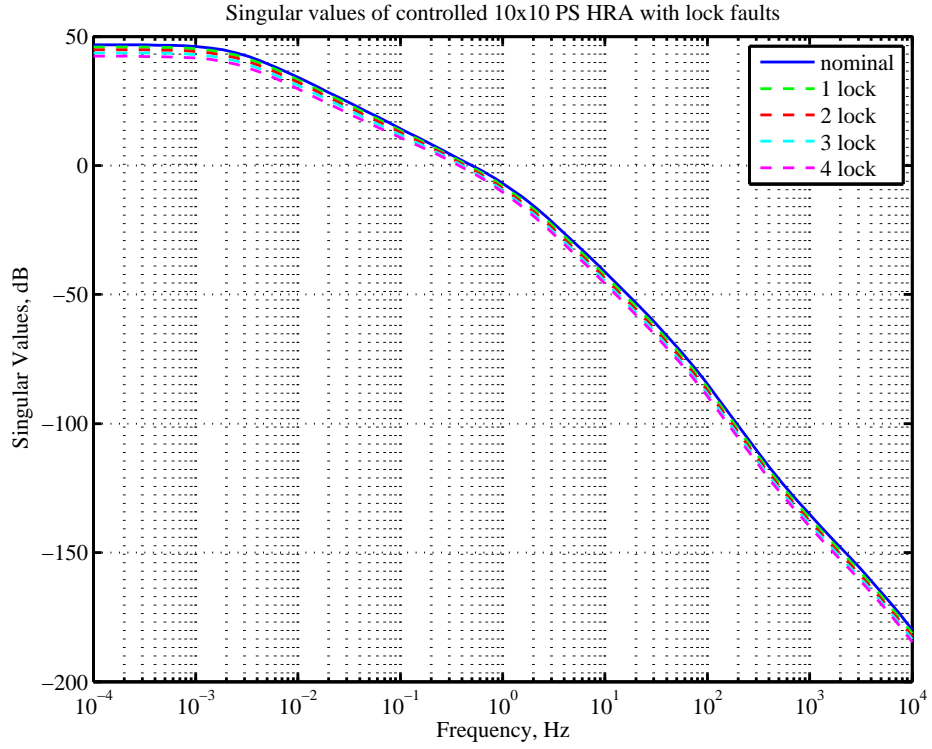


Figure 5.46: Singular values for controlled 10x10 PS HRA with lock-up faults

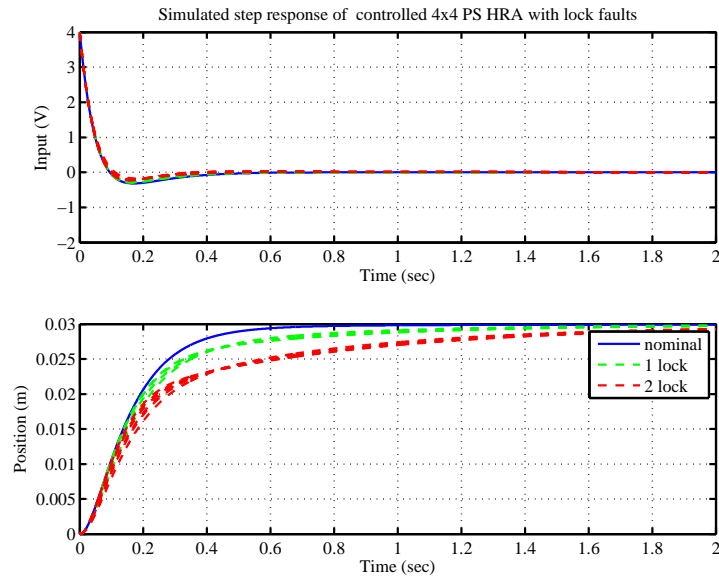


Figure 5.47: Step response of 4x4 PS HRA with passive control and lock-up faults

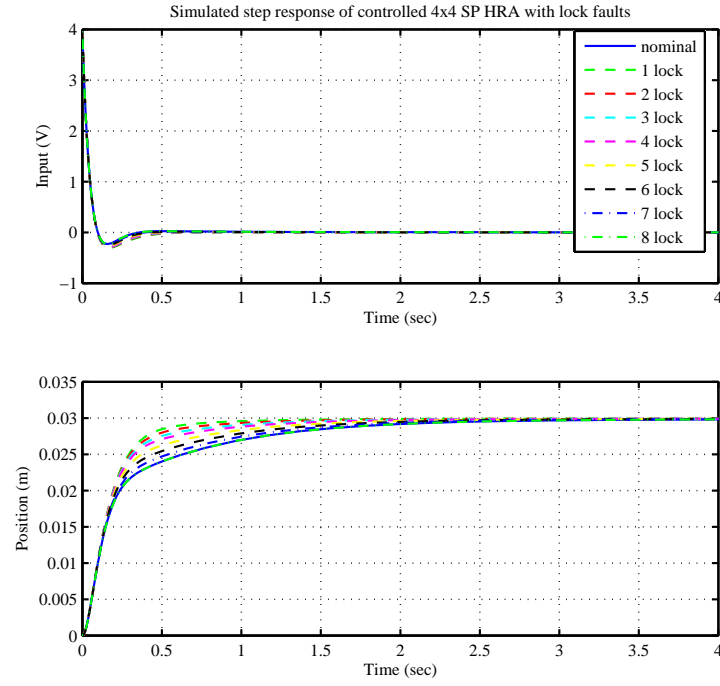


Figure 5.48: Step response of 4x4 SP HRA with passive control and lock-up faults

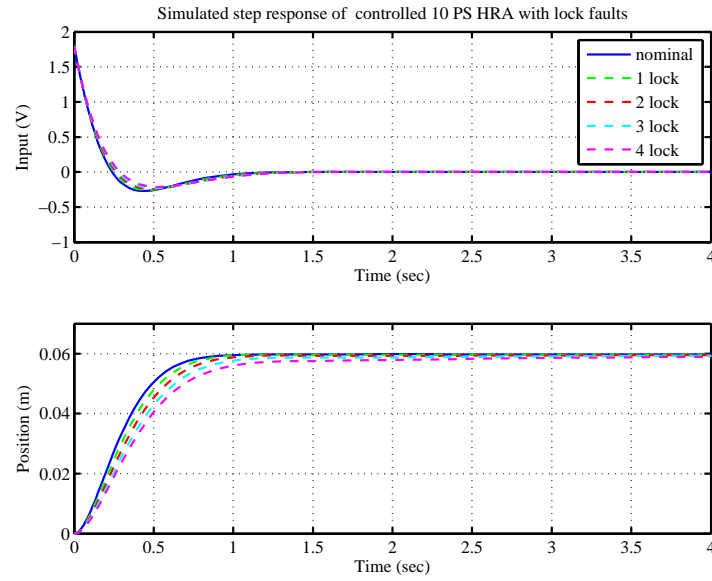


Figure 5.49: Step response of 10x10 PS HRA with passive control and lock-up faults

Table 5.10: Stability margins and transient characteristics of example HRAs under nominal and lock-up fault conditions

System state	PM (deg)	GM (dB)	BW (rads ⁻¹)	$\ G_{f_{CL}}\ _{\infty}$ (dB)	SSerr (%)	RT (s)	ST (s)	OS (%)
4x4 PS HRA								
Nominal	80	41.5	5.76	-	0	0.31	0.58	0
1 lock	86.5	42.5	5.02	0.09	0	0.45	1.32	0.17
2 lock	95.5	43.5	3.99	1.00	0	0.94	2.09	0
4x4 SP HRA								
Nominal	80.0	41.5	5.71	-	0	0.31	0.58	0
1 lock	83.1	41.5	6.42	0.090	0	0.33	0.76	0
2 lock	84.9	41.5	6.42	0.098	0	0.35	0.99	0.03
3 lock	86.6	41.5	6.42	0.11	0	0.39	1.16	0.09
4 lock	88.4	41.5	6.41	0.12	0	0.44	1.28	0.17
5 lock	92.2	41.5	6.25	0.14	0	0.58	1.56	0
6 lock	96.1	41.4	6.06	0.16	0	0.72	1.75	0
7 lock	100.1	41.4	5.82	0.19	0	0.83	1.92	0
8 lock	104.1	41.4	5.53	1.00	0	0.94	2.08	0
10x10 PS HRA								
Nominal	73.7	47.5	3.19	-	0	0.47	0.79	0.13
1 lock	73.4	48.2	2.90	0.082	0	0.66	1.09	0
2 lock	73.7	48.8	2.69	0.15	0	0.77	1.12	0
3 lock	74.7	49.3	2.52	0.20	0	0.85	2.16	0
4 lock	76.7	49.9	2.36	0.24	0	0.93	3.49	0

5.4.4 Fault Simulation Summary

The fault simulations have demonstrated the affect of faults on the closed-loop systems. Parameter deviations have a negligible influence in the closed-loop. Loose faults however, result in increased settling times and overshoots. Likewise, lock-up faults produce increased rise and settling times. Stability margins are maintained in each system under the faults considered. The transient characteristics of the results are summarised in Figures 5.50 to 5.52.

In these figures the settling time, rise time and overshoot of the nominal and faulty systems are represented alongside the performance requirements (denoted by the blue area).

It can be seen that the parameter deviated systems for each example all lie within the performance boundary. Although up to 8 loose faults are theoretically tolerable for the 4×4 PS HRA, only the systems with 1-2 loose faults lie within the desired performance region, due to the large increases in overshoot and settling time. Likewise, up to 2 lock-up faults are tolerable from a capability perspective, however, neither of these systems meet the performance requirements.

The results are similar for the SP system. Only 4 of the theoretically possible 8 lock-up faults are tolerated within the SP system, due to increases in rise time and settling time, and no loose faults lie within the required performance boundary due to the the overshoot and settling time introduced.

The 10×10 PS HRA is capable of tolerating 1-4 loose faults and 1 lock-up before the performance criteria is exceeded.

Overall, a large number of the tolerable fault situations (from a capability perspective) do not satisfy the performance criteria. There are a number of ways in which this could be improved:

- Relax the performance criteria - It may be acceptable in certain applications to define a nominal performance criteria and accept a performance degraded beyond this point under higher levels of fault, and in this way increase the number of permissible faults within the system. However, this is not an ideal concession.
- Improve the passive control - The control approach taken within this section is very basic. A more sophisticated approach or a more complex control law could provide greater performance in fault conditions.
- Active fault tolerant control - Changing the control law in response to fault conditions will improve performance, at the expense of a reliance on fault detection and reconfiguration.

The next chapter investigates an active fault tolerant approach to controlling HRA and will aim to quantify the advantages in terms of performance and fault tolerance. Issues associated with the introduction of active FTC such as the reliance on fault detection will also be addressed within this chapter.

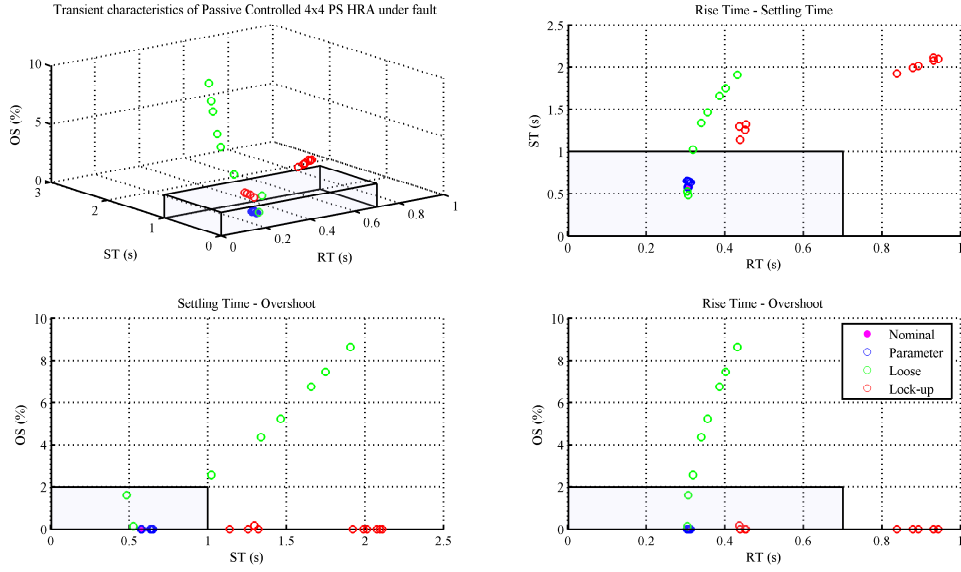


Figure 5.50: Transient characteristics of passively controlled 4x4 PS HRA under fault

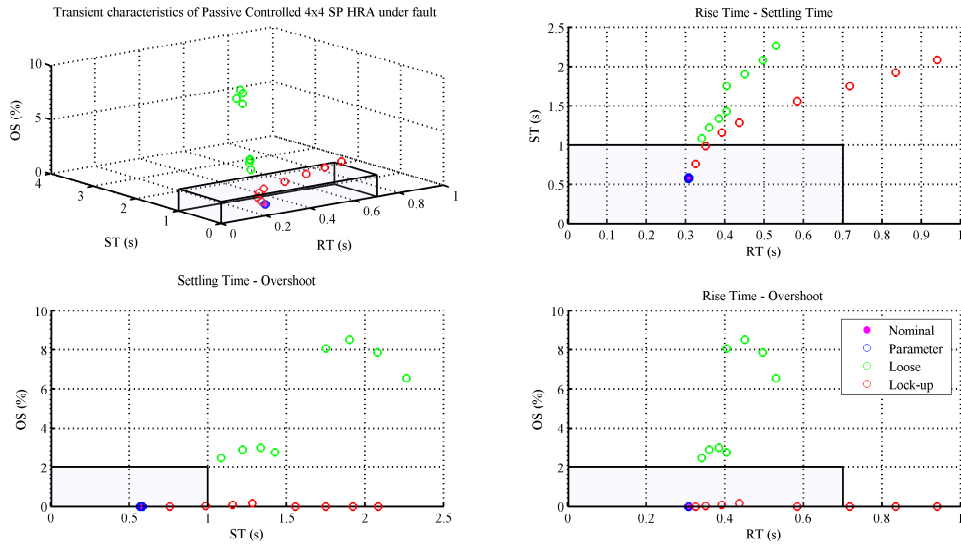


Figure 5.51: Transient characteristics of passively controlled 4x4 SP HRA under fault

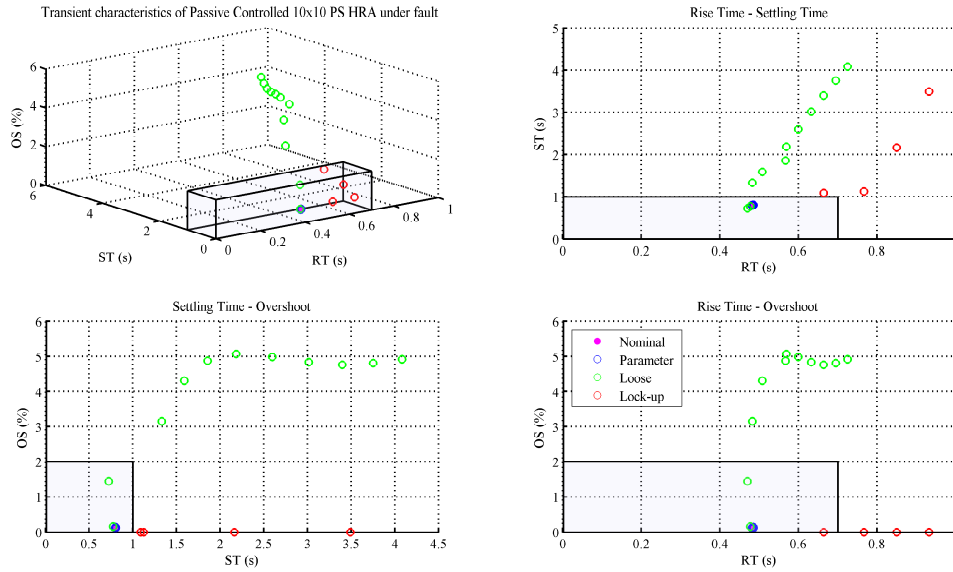


Figure 5.52: Transient characteristics of passively controlled 10×10 PS HRA under fault

5.5 Conclusions

The design and simulation of passive fault tolerant control for three HRA examples have been presented in this chapter.

Analyses of the systems under nominal and fault conditions were initially conducted, in order to establish the control requirements and quantify the effects of faults on the systems. Generally, it was shown that faults have less affect in larger systems, and that serial actuation provides greater tolerance to loose faults and parallel actuation to lock-ups. Parameter deviations were found to have a relatively small influence in both the SP and PS configurations. Loose faults and lock-up faults however, have greater affect, with the lock-up fault being the more severe.

Loose and lock-up faults were found to have conflicting effects from a control design perspective, and as such, a trade-off exists between designing for loose fault and lock-up fault performance. It was assumed that the occurrence of loose and lock-up faults were of equal likelihood, and thus designing control based on the nominal system was considered the most suitable compromise.

Hence, passive classical control laws were designed for each system based on the nominal mode. This control was tested under simulation using the full order, non-linear system. Faults were injected into each system up to the theoretical capability limits of the system. Results of these simulations indicate that, using the single basic control law designed in 5.3, performance is significantly degraded, leading to a large number of the theoretically tolerable faults exceeding the performance requirements.

The active fault tolerant control of the example HRAs will be addressed within Chapter 7 in order to assess the benefits accrued in terms of fault tolerance and performance with the introduction of control reconfiguration.

Chapter 6

Fault Detection & Health Monitoring

6.1 Introduction

One of the major advantages of passive fault tolerant control (Chapter 5) is that fault state knowledge i.e. Fault Detection and Isolation (FDI) is not required from a control perspective. However, from an operational perspective, some form of health monitoring is a necessity. It is envisaged that the HRA will continue to operate within an acceptable performance region under element fault conditions, until the capability (be it travel or force) falls below that required by the application. At this point, or just before it, maintenance will be required to replace the HRA unit. Hence, health monitoring is needed to provide an indication of the capability of the HRA in order to schedule this maintenance or health information could be used for operational purposes.

The Active Fault Tolerant Control (AFTC) method presented in the next chapter also requires the fault state of the system to be detected. However, in AFTC, the control is dependent on the fault state and thus more detailed information is required.

This chapter approaches the issues of fault detection for general health monitoring purposes and for AFTC. To this end, two methods of fault detection are presented and applied to the 4×4 PS example HRA system¹: an Interacting Multiple-Model (IMM) method for general health monitoring and a rule-based approach to be used in the AFTC method.

Two different approaches are described as FDI for general health monitoring and FDI for the AFTC methods presented in this work have differing requirements and resources at their disposal. These characteristics are summarised in Table 6.1.

As mentioned previously, the explicitness of the fault information required is different, however, there also other distinctions. The speed of detection is more critical within the AFTC set-up, as the effectiveness of the control relies upon the diagnosis. Simplicity is also a major requirement for the particular AFTC method presented in this work. The AFTC method in question is inspired by multi-agent concepts, and as such it operates on a localised basis². Hence, faults need to be detected in small sub-sets of the HRA's elements. This information

¹For the sake of brevity, the fault detection methods are only applied to the 4×4 PS system, and not all three example HRAs. The methods apply equally to the other example systems and indeed the rule-based method is applied to the other HRAs in the Chapter 7.

²The multi-agent active fault tolerant control approach is described in detail in Chapter 7.

Table 6.1: Fault detection and health monitoring requirements and resources

	Fault Detection for AFTC	General health monitoring
Requirements	Fault type and location Fast detection Localised operation	Indication of HRA's capability Timely detection Centralised operation
Resources	Many local sensors Local input information	Minimal sensors Single input information

can then be aggregated throughout the agency, providing a very detailed representation of the fault state of the HRA. Consequently, there are numerous fault detection units, and as such a simple approach is desirable in order to reduce complexity in the system. Indeed, a simple approach is all that should be required as detecting faults on this local scale is more straightforward than taking a centralised perspective.

Another consequence of the localised AFTC is that there is local sensory information available to each FDI unit. In contrast, the FDI for health monitoring should use the limited sensory information available in the passively controlled systems, as added sensors will increase cost. As the scheme is centralised, however, then a more complex approach may be permissible as it is a one-off in the system.

6.2 Rule-Based Fault Detection for AFTC

A rule-based approach to fault detection was chosen for use with the multi-agent inspired control scheme of Chapter 7 as it satisfies the requirements outlined in Table 6.1. The rule-based detection is very simple, and provides a finite-state representation of the fault condition.

The composition of the rule-based FDI for the PS 4×4 HRA example is represented in Figure 6.1. The figure shows the internal control structure of the agent that controls the bank of parallel elements nearest the load. There is an identical agent for each parallel bank of elements. Details of the control architecture are given in Chapter 7, and as such they are omitted here. All that is necessary to note is that in each agent there is a fault detection unit which has access to all the quantities within the agent, such as the local measurements of element position and coil current, the actuator voltage input and local reference, and with these quantities it is necessary for the FDI unit to determine whether the elements are healthy, locked, or if a proportion of them are loose using some logical relations. The lock-up and loose fault detection processes have been separated within this layer, and operate concurrently to optimise detection times. Also, a loose detection algorithm is used per element within the agent's subsystem, thus separating the detection algorithms allows easier extension to different element levels. The detection algorithms are discussed in the subsections that follow.

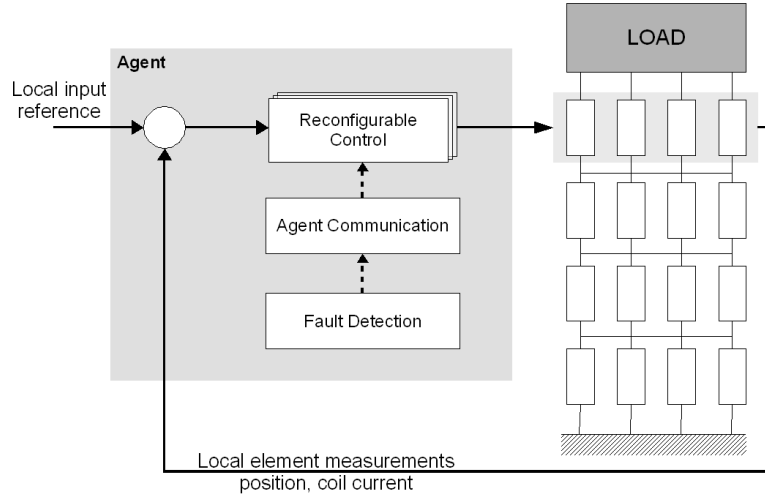


Figure 6.1: Rule-based fault detection in multi-agent control of HRA

6.2.1 Lock-up Fault Detection

Symptoms: If an element is locked, then its relative velocity is zero. Other occasions where an element may be stationary include when the element is at its reference point, or when it has reached the limit of travel.

Diagnosis: Hence, the algorithm should firstly check if the element is moving. If it is not, then it should check if there is a position error in order to determine if it is at the reference point. If there is an error, then the position of the element should be checked to determine if the element is at the limit of travel, and the input is larger than the element travel limit.

Rules: Hence, to diagnose a lock-up fault, the following rules can be used:

$$\begin{aligned}
 F_{lock} &\rightarrow L_{lock1} \wedge L_{lock2} \\
 L_{lock1} &: \dot{x} < \dot{x}_{threshold} \\
 L_{lock2} &: e > e_{threshold} \wedge (x < x_{limit} \vee (x = x_{limit} \wedge u < x_{limit}))
 \end{aligned}$$

where L_{lock} represents the rules to be checked, e is the local input error and $e_{threshold}$, $\dot{x}_{threshold}$ and x_{lim} represent the position error threshold, the velocity threshold and the travel limit respectively.

The inputs and outputs of the detection algorithm are shown in Figure 6.2 and the process is described in the flow chart of Figure 6.3. The position and error are locally available, and the velocity is approximated using a difference equation.

A conviction factor is used within the algorithm to decide whether the actuation element is locked. If the rules are valid for enough iterations, then the fault detection will enter the lock-up state. If however the actuation element moves, the conviction factor is reset. This

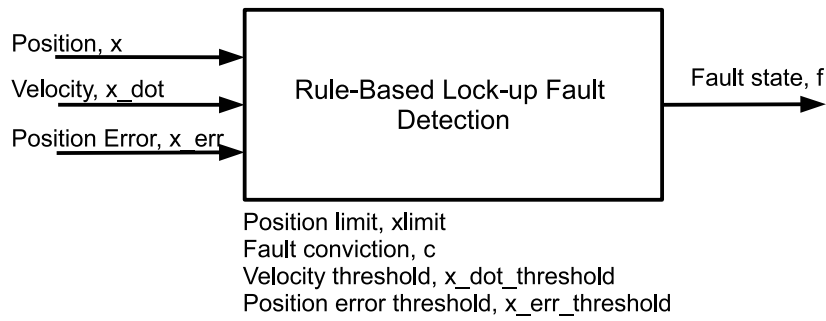


Figure 6.2: Lock-up fault detection

conviction factor and the thresholds are manually tuned based on the clock rate and the system.

6.2.2 Loose Fault Detection

Symptoms: If an element is loose, then the current flowing within the coil is zero. The current may also be zero whilst the actuator is healthy if the element is stationary.

Diagnosis: Thus, to diagnose a loose fault the algorithm should firstly check if a current flows in the coil. If there is no current, then the velocity should be checked, a non-zero value indicated that the element is moving whilst not being driven, i.e. it is loose.

Rules: Thus, diagnosis of this fault type may be achieved with the following rules:

$$\begin{aligned}
 F_{loose} &\rightarrow L_{loose1} \wedge L_{loose2} \\
 L_{loose1} &: I_c < I_{c\text{threshold}} \\
 L_{loose2} &: \dot{x} > \dot{x}_{\text{threshold}}
 \end{aligned}$$

where L_{loose} represent the rules to be checked, and I_c is the measured coil current. Figure 6.4 shows the inputs and outputs of the loose fault detection algorithm and Figure 6.5 provides a flow chart. Again, a conviction factor is used within the algorithm to decide whether the actuation element is loose.

6.2.3 Fault Simulations

The rule-based FDI approach described in the previous sub-section is simulated here. The cascaded nominal control architecture that will be described in Chapter 7 is used. A sine wave input reference provides a constant source of excitation and Gaussian white noise is added to the HRA plant input and measured values. A sampling frequency and clock rate of 20 Hz is used within the fault detection element of the simulation.

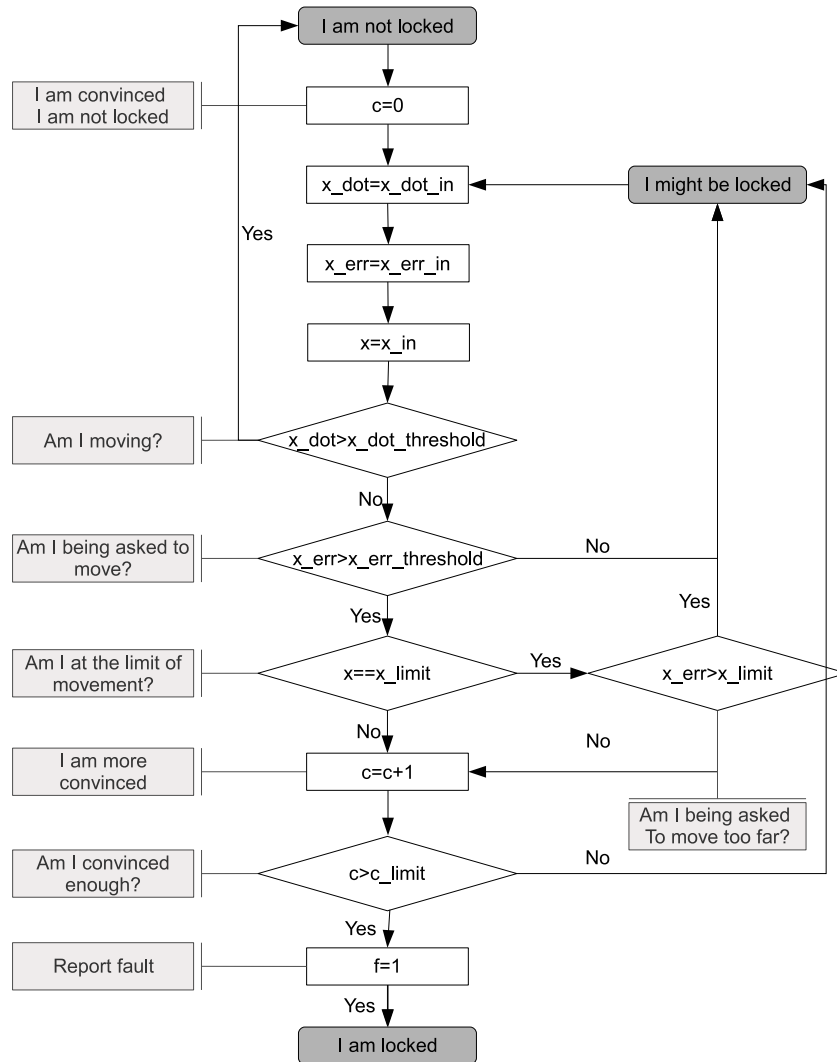


Figure 6.3: Flow chart for lock-up fault detection

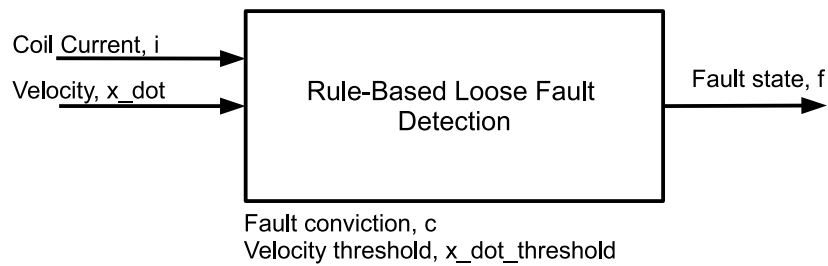


Figure 6.4: Loose fault detection

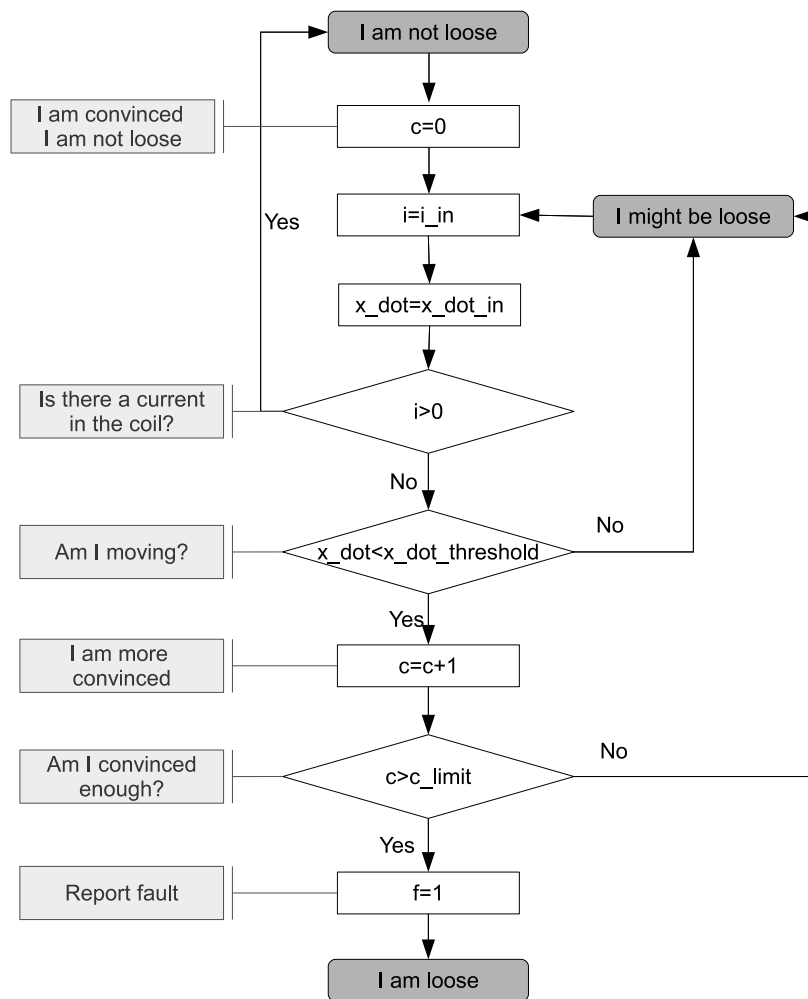


Figure 6.5: Flow chart for loose fault detection

Table 6.2: Loose fault simulation profile (refer to Figure C.1 for element numbers)

Fault State	Loose Elements	Time
Nominal	-	0-2s
1 Loose	E11	2-4s
2 Loose	E11 E21	4-6s
3 Loose	E11 E21 E31	6-8s
4 Loose	E11 E21 E31 E41	8-10s
5 Loose	E11-2 E21 E31 E41	10-12s
6 Loose	E11-2 E21-2 E31 E41	12-14s
7 Loose	E11-2 E21-2 E31-2 E41	14-16s
8 Loose	E11-2 E21-2 E31-2 E41-2	16-18s

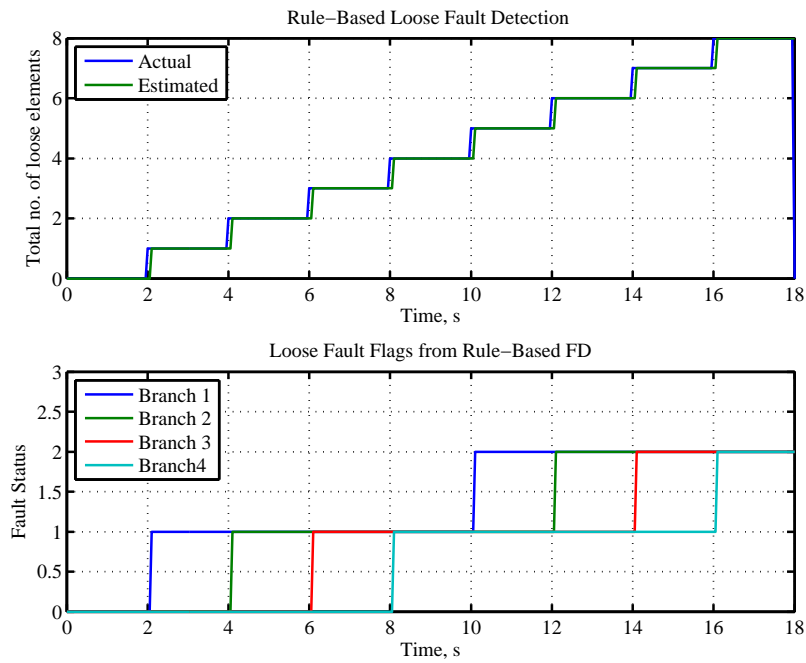


Figure 6.6: Rule-based loose fault detection simulation

6.2.3.1 Loose Faults

Figure 6.6 gives the fault detection signals produced by the rule-based FDI where the fault status changes from nominal to 8 loose faults (the total number tolerable in this configuration) as described in Table 6.2. The input and output to the HRA during this period is shown in Figure 6.7. The rate at which faults occur in this profile is unrealistically high. However, it is simulated in this fashion as long-term, realistic timescales are not practical.

Each fault is detected accurately within 0.1s. However, it should be noted that if the input type is changed to a less constant excitation, detection times can rise.

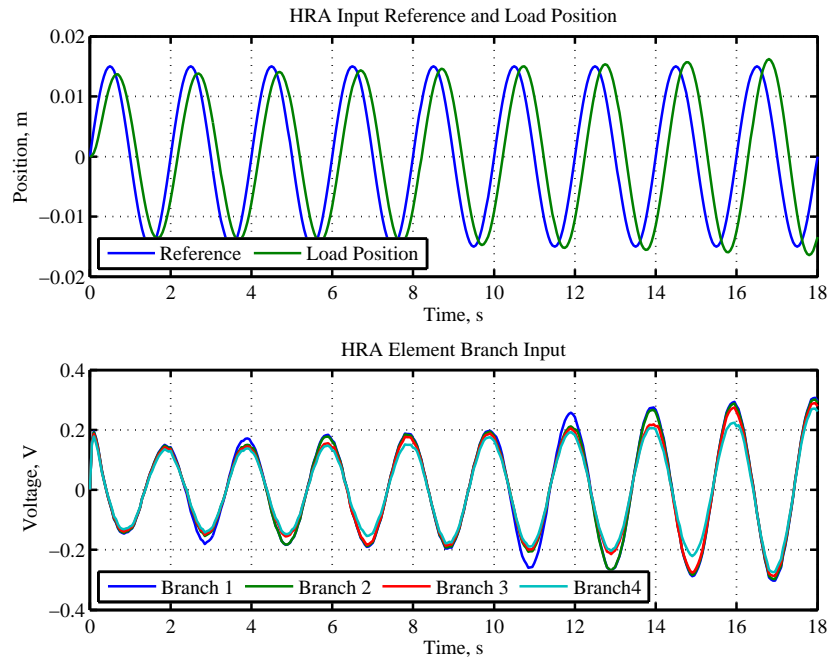


Figure 6.7: HRA input and output during loose fault profile.

Table 6.3: Lock-up fault simulation profile

Fault State	Locked branch	Time
Nominal	-	0-5s
1 Locked	E2	5-10s
2 Locked	E2, E4	10-15s

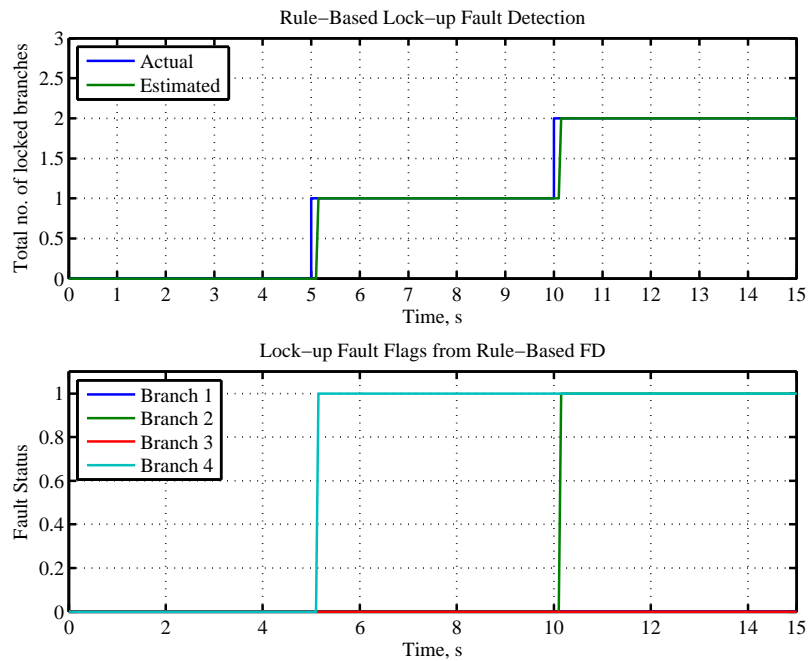


Figure 6.8: Rule-based lock-up fault detection simulation

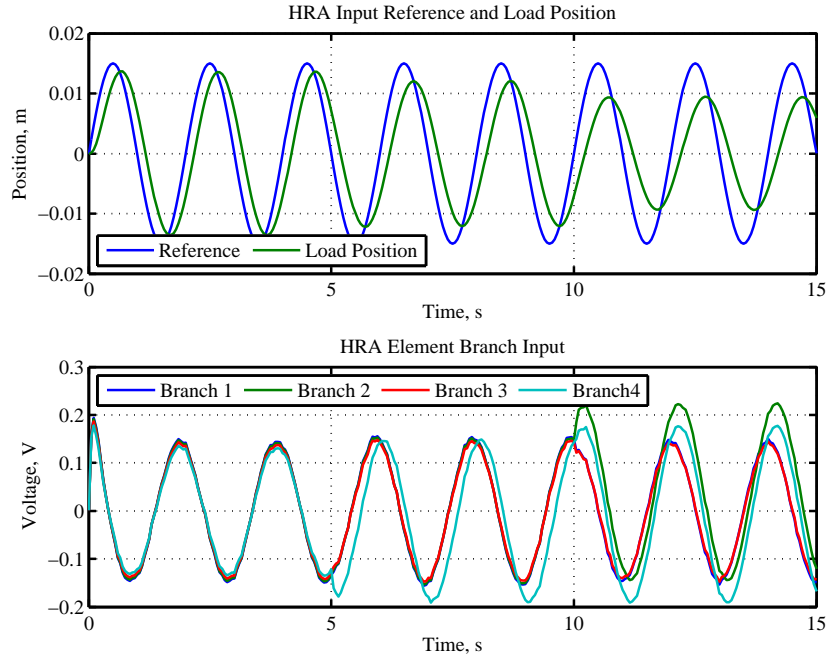


Figure 6.9: HRA input and output during lock-up fault profile.

6.2.3.2 Lock-up Faults

Simulation of the lock-up fault profile represented in Table 6.3 results in the fault detection waveforms shown in Figure 6.8 with input and output signals of that shown in Figure 6.9. These results are typical of other fault location profiles.

Again, the detection of faults is accurate and quick, with a maximum detection time of 0.15 seconds. As in the loose fault detection, the speed of detection is decreased if the excitation to the system is reduced, as faults cannot be detected when the element is stationary.

6.2.3.3 Summary

The fault simulations show that lock-up and loose faults are detected quickly and accurately within this rule-based FDI approach. Detailed information regarding quantity and location of fault is provided. However, a large number of sensors are required, and this approach is only really practical when combined with a highly decentralised control strategy. Hence a more centralised approach is discussed in the next section.

6.3 Interacting Multiple-Model Fault Detection for Health Monitoring

As stressed in Section 6.1, a health monitoring scheme for use with passive robust control should use the measurands that are already available to produce an indication of the overall capability of the HRA. This can be a difficult task as the measured information is from the load alone i.e. nothing is explicitly known about the element-level dynamics. However, this

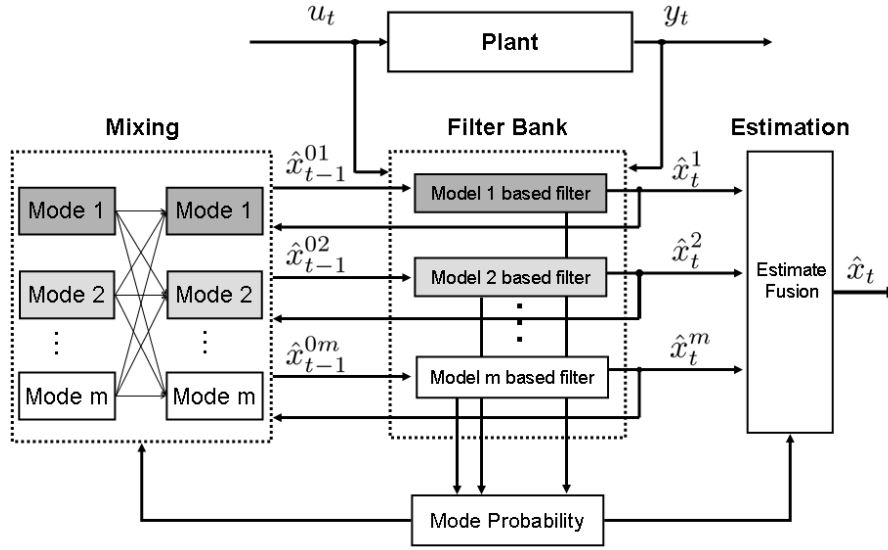


Figure 6.10: IMM estimation

limited measured information can be combined with what is known about the behavioural dynamics of the system under nominal and fault conditions i.e. the system models, to produce an estimate of the system health state. The IMM FDI approach provides one possible means of achieving this.

Conventional multiple-model estimation methods use a bank of filters, each of which is based on a model of the system when it is in a particular mode. The outputs of these filters are combined with a probabilistically weighted sum to achieve an overall state estimate.

However, there is no interaction between the filters, and as such the approach is not suited to situations where the parameters or structure of the system changes (82). Nonetheless, non-interacting methods of multiple-model estimation have been applied to FDI applications, where sudden parameter and structural changes to the system occur using ad hoc solutions (153; 154).

The interacting multiple-model method, developed in the field of tracking (155; 156) deals with these issues. In the IMM approach, the initial estimate at the beginning of each iteration is a mixture of recent estimates from the filters. As a result the accuracy of estimation is increased and dependency on the previous mode history is introduced. This increases its suitability to detecting faults and thus it has been applied within this field (157; 82; 158; 159; 160).

6.3.1 IMM Estimation Algorithm

A depiction of the IMM estimation algorithm is shown in Figure 6.10. A number of filters (in this case Kalman filters) are designed based on m models of the system modes.

Also, a mode transition probability matrix p_{ij} is defined where the element ij represents the probability of transition from mode i to mode j in the next time frame. This may be based on knowledge of fault type frequency and likelihood when the system is in a certain state.

The IMM algorithm has four main stages:

- Mixing
- Mode matched filtering
- Mode probability calculation
- Combination of estimates

Mixing

The first stage of the IMM algorithm involves the mixing of all the filters estimated values and covariances from the previous iteration ($\hat{x}_{(t-1)}^i$ and $P_{(t-1)}^i$ for $i = 1 : m$) and the mixed probability, $\rho_{i|j(t-1)}$ to produce the input to the filters:

$$\hat{x}_{(t-1)}^{0j} = \sum_{i=1}^m \hat{x}_{(t-1)}^j \rho_{i|j(t-1)}, \quad j = 1, \dots, m \quad (6.1)$$

$$P_{(t-1)}^{0j} = \sum_{i=1}^m \rho_{i|j(t-1)} \left\{ \left[\hat{x}_{(t-1)}^j - \hat{x}_{(t-1)}^{0j} \right] \right. \quad (6.2)$$

$$\left. \cdot \left[\hat{x}_{(t-1)}^j - \hat{x}_{(t-1)}^{0j} \right]^T \right\} \quad (6.3)$$

where $\rho_{i|j(t)}$ in the previous time step was calculated by:

$$\rho_{i|j(t-1)} = \frac{1}{\bar{c}_j} p_{ij} \rho_{i(t-1)}, \quad i, j = 1, \dots, m \quad (6.4)$$

$$\bar{c}_j = \sum_{i=1}^m p_{ij} \rho_{i(t-1)}, \quad j = 1, \dots, m \quad (6.5)$$

Mode matched filtering

The Kalman filter algorithms are then obtained based on the discrete system. For a discrete system:

$$x_{(t+1)} = Fx_{(t)} + Gu_{(t)} + w_{(t)} \quad (6.6)$$

$$y_{(t)} = Hx_{(t)} + Lu_{(t)} + v_{(t)} \quad (6.7)$$

where $w_{(t)}$ and $v_{(t)}$ are the plant and measurement noise respectively with covariances of Q and R . Both are assumed to be white Gaussian with zero mean. The Kalman filter

algorithms can then be expressed as:

$$\hat{x}_{(t/t-1)}^j = F^j(\hat{x}_{(t-1/t-1)}^{0j}) + D^j u_{(t-1)} \quad (6.8)$$

$$\hat{x}_{(t/t)}^j = \hat{x}_{(t/t-1)}^j + K_{(t)}^j \left[y_{(t)} - (H^j(\hat{x}_{(t/t-1)}^j) + L^j u_{(t)}) \right] \quad (6.9)$$

$$K_{(t)}^j = P_{(t/t-1)}^j H_{(t/t-1)}^{jT} S_{(t)}^{j-1} \quad (6.10)$$

$$S_{(t)}^{j-1} = H_{(t/t-1)}^j P_{(t/t-1)}^j H_{(t/t-1)}^{jT} + R_{(t-1)}^j \quad (6.11)$$

$$P_{(t/t-1)}^j = F_{(t-1)}^j P_{(t/t-1)}^{0j} F_{(t-1)}^{jT} + G_{(t-1)}^j Q_{(t-1)}^j G_{(t-1)}^{jT} \quad (6.12)$$

$$P_{(t/t)}^j = P_{(t/t-1)}^j - K_{(t)}^j S_{(t)}^j K_{(t)}^{jT} \quad (6.13)$$

Mode Probability Calculation

The mode probability, $\rho_{j(t)}$ (for mode j at time t) is then updated based on the likelihood function Λ for each mode filter:

$$\rho_{jt} = \frac{\Lambda_{j(t)} \bar{c}_j}{\sum_{i=1}^m \Lambda_{i(t)} \bar{c}_i} \quad (6.14)$$

$$\Lambda_{j(t)} = \left| 2\pi S_{(t)}^j \right|^{-\frac{1}{2}} \exp \left[-\frac{1}{2} \left(y_{(t)} - \left(H^j \hat{x}_{(t/t-1)}^j + L^j u_{(t)} \right) \right)^T \right. \quad (6.15)$$

$$\left. \cdot \left(S_{(t)}^j \right)^{-1} \left(y_{(t)} - \left(H^j \hat{x}_{(t/t-1)}^j + L^j u_{(t)} \right) \right) \right] \quad (6.16)$$

The mode probabilities give a time-varying estimate on the likelihood of the system state being one of the model-based modes and thus they are used in the indication of fault type for FDI applications. The probabilities are smoothed using a moving average window.

Combination of Estimates

Finally, the combined state estimate $\hat{x}_{(t)}$ and covariance $P_{(t)}$ are derived by weighting the estimated state and the mixed covariance for each mode with the mode probabilities:

$$\hat{x}_{(t)} = \sum_{j=1}^m \hat{x}_{(t)}^j \rho_{j(t)} \quad (6.17)$$

$$P_{(t)} = \sum_{j=1}^m \rho_{j(t)} \left[P_{(t)}^j + \left[\hat{x}_{(t)}^j - \hat{x}_{(t)} \right] \cdot \left[\hat{x}_{(t)}^j - \hat{x}_{(t)} \right]^T \right] \quad (6.18)$$

6.3.2 IMM Mode Allocation

The choice of modes on which to base the filters is important within this method, as the output of the IMM algorithm is the likelihood of each mode being the active mode in the system, with respect to each other (eqn. 6.16).

One allocation option is to use a filter for every possible fault scenario, as this should provide the most accurate estimation. This approach was taken in (161) for a 3×3 PS

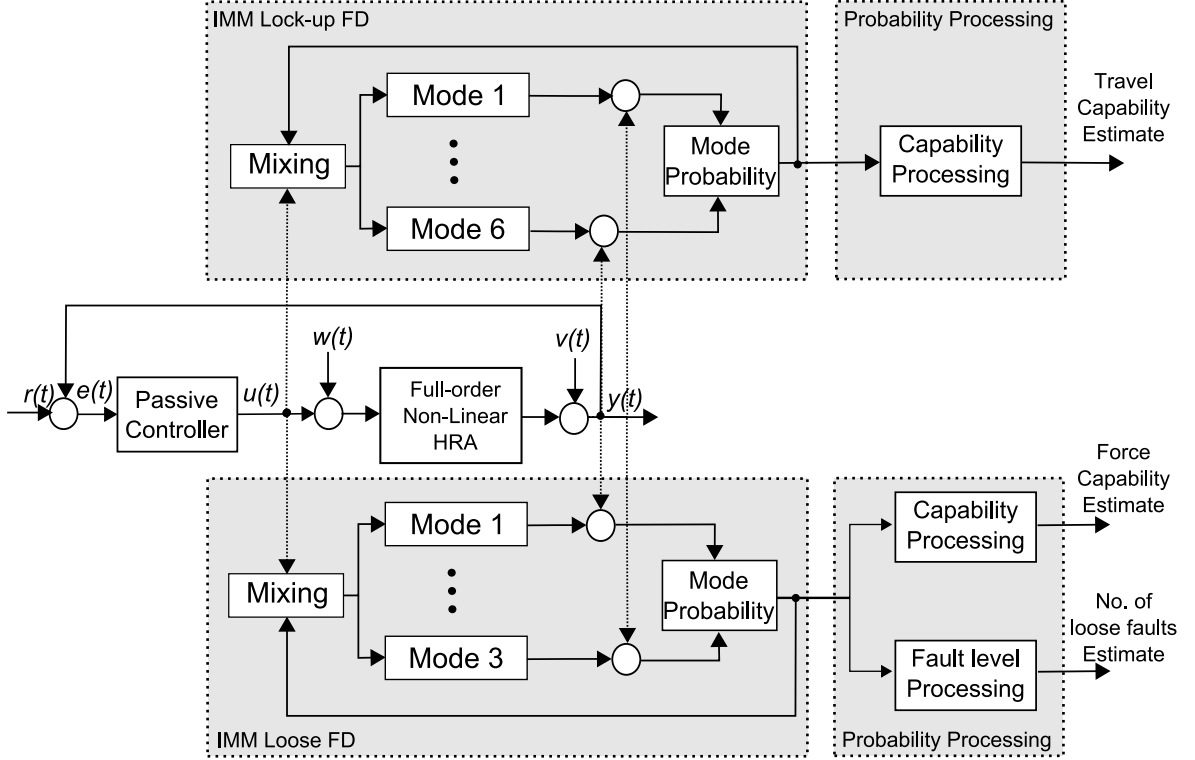
system where 23 modes were required to cover all possibilities up to 2 loose faults and 2 lock-ups. Whilst the performance of this approach was good, it was noted that this is a large number of filters considering the relatively low number of elements within the system, and for larger HRAs the number of filters required explodes. This approach also required use of all the relative element positions and full order representations of the system. As the example systems are larger than the system addressed within this paper, and the available measurements are limited in this case, then this approach is not taken here. However, the interested reader is directed towards Appendix F where a copy of the paper can be found.

Since the work of (161), another approach to mode allocation has been taken, which can reduce the number of modes and measurements required. Figure 6.11 represents the IMM FDI approach taken for the 4×4 PS HRA example within this study. The modes used are:

- IMM Loose FDI:
 - Mode 1: Nominal system
 - Mode 2: 1 element loose in each branch
 - Mode 3: 2 elements loose in each branch
- IMM Lock-up FDI
 - Mode 1: Nominal system
 - Mode 2: Lock-up in parallel element branch 1
 - Mode 3: Lock-up in parallel element branch 2
 - Mode 4: Lock-up in parallel element branch 3
 - Mode 5: Lock-up in parallel element branch 4
 - Mode 6: 2 lock-up faults in parallel element branches 1 and 2
 - Mode 7: 2 lock-up faults in parallel element branches 3 and 4

Two separate IMM algorithms are used, one for detecting loose faults and one for lock-ups and their outputs are combined post-IMM to provide a full health state. This reduces the number of filters required as modes representing combinations of these faults are no longer needed. Both of these IMM algorithms use 8th order filters based on the reduced order systems, and use only one system output to base the estimation upon: the load position. Consequently, it is now difficult to know where the fault is actually located. Fault location has important implications for determining the likelihood of the next fault decreasing the capability significantly, or causing a failure. Hence for detailed health prognostics, the information provided by this scheme may not be sufficient. However, an estimate of the capability can be detected, and this this can be used as an important indicator the current health of the element. In addition, information regarding the timing of faults, and the progression from one fault state to the next can also be implied and used within the health monitoring.

The IMM loose FDI uses 3 modes. Mode 2 represents the system with one loose fault in each parallel branch (4 loose faults in the system) and mode 3 represents the system with 2


 Figure 6.11: IMM FDI for 4×4 PS HRA

loose faults in each branch (8 loose faults in all). As a consequence, when there are fewer actual loose faults in the HRA, the likelihood produced for these modes does not approach unity. In fact, the likelihood is directly proportional to the number of faults present. For example, if there is one loose fault in the HRA, then the likelihood of mode 3 is found to be close to 0.25 i.e. a quarter of the 4 loose faults that mode 3 represents is present. This can be exploited in the post-IMM processing of the probabilities and an estimate of both the number of loose faults \hat{F}_{loose} and the force capability of the system can be produced from these three modes. \hat{F}_{loose} is derived as follows:

$$\hat{F}_{loose} = 4\Lambda_2(t) + 8\Lambda_3(t) \quad (6.19)$$

The IMM lock-up FDI uses more modes, 7 in all. This is because the location of the lock-up fault has a stronger influence on the overall system behaviour. Modes 2 to 5 represent a lock-up fault in each branch respectively. The algorithm does not reliably indicate the correct single fault mode due to the limited measured outputs. However, as the fault location is not of interest, it is possible to simply sum the probability of modes 2 to 5 to produce a more robust probability of there being a single lock-up (or a single reduction in travel capability) in the system. Likewise, the sum of likelihoods of modes 6 to 7 provides a probability of there being two branches locked in the system, where these modes represent the two extreme location possibilities (two locked branches near the base, and two locked branches near the load). Hence, the probability of travel capabilities 4 to 2 (4 representing 4 times the single

element capability etc.) may be expressed as:

$$\Lambda_{travelcap4}(t) = \Lambda_1(t) \quad (6.20)$$

$$\Lambda_{travelcap3}(t) = \sum_{i=2}^5 \Lambda_i(t) \quad (6.21)$$

$$\Lambda_{travelcap2}(t) = \sum_{i=6}^7 \Lambda_i(t) \quad (6.22)$$

6.3.3 Fault Simulation

The simulation set-up is shown in Figure 6.11. The system is given a sine wave input reference to provide a constant source of excitation. The transition matrix p_{ij} is set as shown in equation 6.23 such that the probability of no transition from the current state (i.e. where $i = j$) is 0.999 and transitions to the other modes (i.e. where $i \neq j$) are set at equi-likelihood.

$$p_{ij} = \begin{bmatrix} \alpha & \beta & \cdots & \beta \\ \beta & \alpha & \ddots & \vdots \\ \vdots & \ddots & \ddots & \beta \\ \beta & \cdots & \beta & \alpha \end{bmatrix} \in \mathbb{R}^{n \times n} \quad (6.23)$$

where:

$$\begin{aligned} \alpha &= 0.999 \\ \beta &= \frac{1 - \alpha}{(n - 1)} \end{aligned}$$

A very small value of covariance is used for the noise on the measured position ($5 \times 10^{-12} \text{m}^2$), as the glass encoder that is used to measure position has an rms noise value of $1 \mu\text{m}$. The plant noise covariance Q is set at $1 \times 10^{-5} \text{V}^2$, as this gives a noise level in the order of mV.

6.3.3.1 Loose Faults

Figure 6.12 gives the mode probabilities produced by the IMM loose FDI where the fault status changes as described in Table 6.2.

Figure 6.12 shows that the nominal state in the first fault period is detected clearly. In subsequent fault periods, where loose faults are in effect, the mode probabilities behave as discussed in Section 6.3.2. Mode 2 only approaches unity when four loose faults in separate branches are present, but for fault states before this point, the probability is roughly proportional to the actual number of loose faults. The same applies to mode 3.

Using equation 6.19, the number of loose faults is estimated as shown in Figure 6.13. It can be observed that the limited number of filters and measurements used can provide a rough estimate of the number of loose faults in the system. The capability is more important for the health monitoring of the HRA, however, and Figure 6.14 gives the estimated capability in this case. After 4 seconds the first loose fault causes a decrease in force capability from 3 to 4.

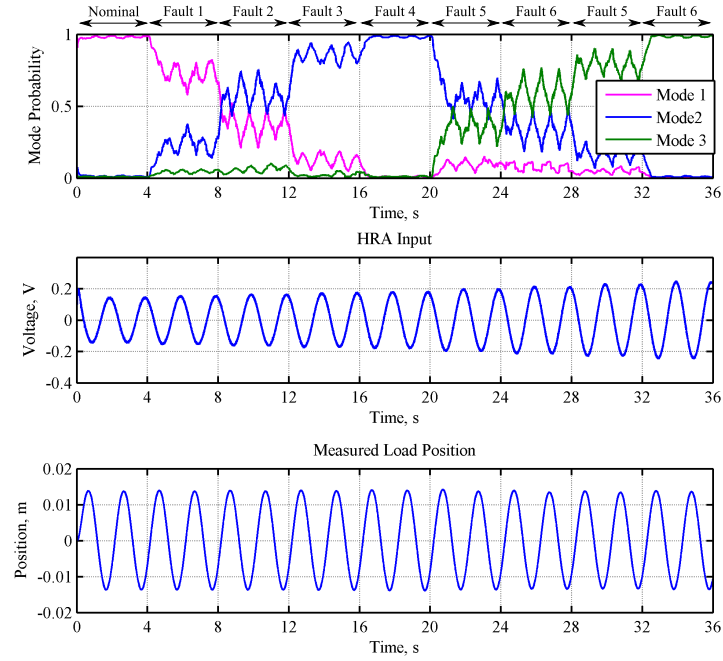


Figure 6.12: Mode probabilities produced by IMM loose FDI from measured load position output and known voltage input.

Table 6.4: Loose fault simulation profile

Fault State	Loose Elements	Time
Nominal	-	0-4s
1 Loose	E11	4-8s
2 Loose	E11 E21	8-12s
3 Loose	E11 E21 E31	12-16s
4 Loose	E11 E21 E31 E41	16-20s
5 Loose	E11-2 E21 E31 E41	20-24s
6 Loose	E11-2 E21-2 E31 E41	24-28s
7 Loose	E11-2 E21-2 E31-2 E41	28-32s
8 Loose	E11-2 E21-2 E31-2 E41-2	32-36s

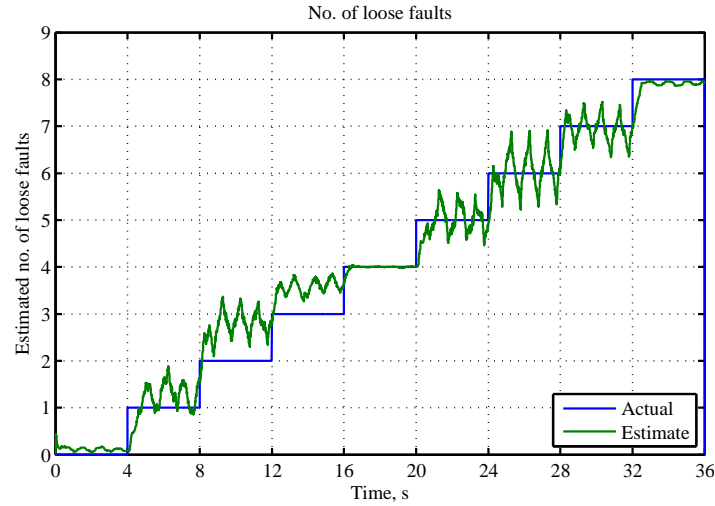


Figure 6.13: Estimated number of loose faults.

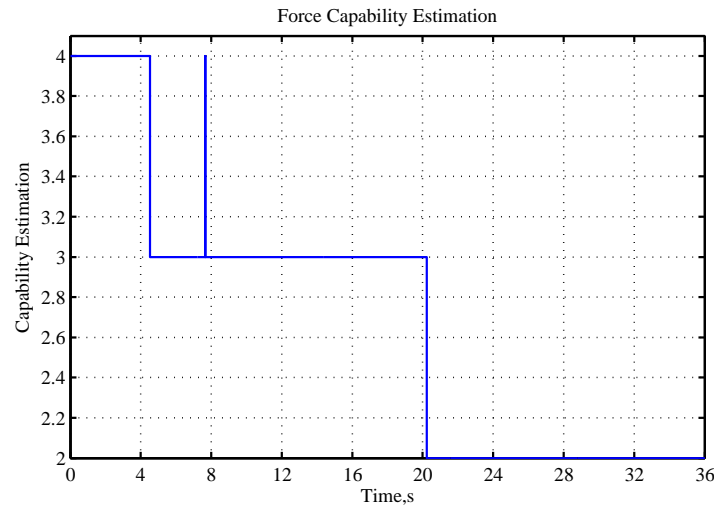


Figure 6.14: Estimated force capability.

This is detected in 0.528 seconds. At 20 seconds the fifth loose fault causes the capability to decrease from 3 to 2, which is detected in 0.23 seconds. There is a small spike in the capability at around 8 seconds, however, this could be smoothed out.

6.3.3.2 Lock-up Faults

The fault profile described in Table 6.5 was simulated and the resultant fault modes and capability estimation is shown in Figure 6.15. the inputs and outputs of the system are also given in Figure 6.16. These results are typical for all lock-up fault locations.

The summed mode probabilities indicate the correct system fault state throughout the fault profile. In this instance, the correct single locked element branch mode is chosen as the most probable during the single lock-up period (2-4 seconds). This location accuracy is not always the case with single lock-up faults, however, a single fault is always indicated.

When the second fault occurs in element branch 4, there is a temporary rise in mode 5

Table 6.5: Lock-up fault simulation profile

Fault State	Locked branch	Time
Nominal	-	0-2s
1 Locked	E2	2-4s
2 Locked	E2, E4	4-6s

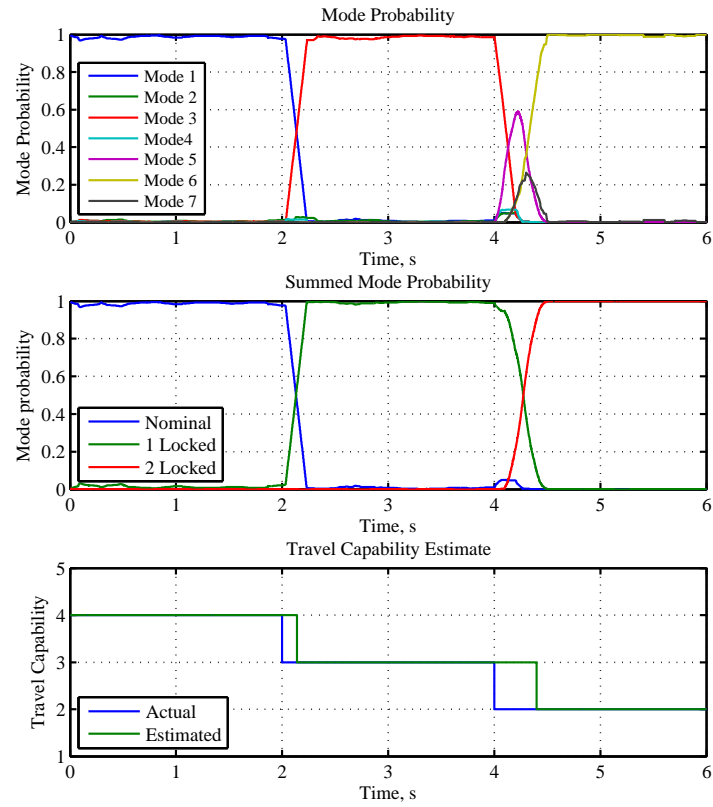


Figure 6.15: Mode probabilities and estimated travel capability.

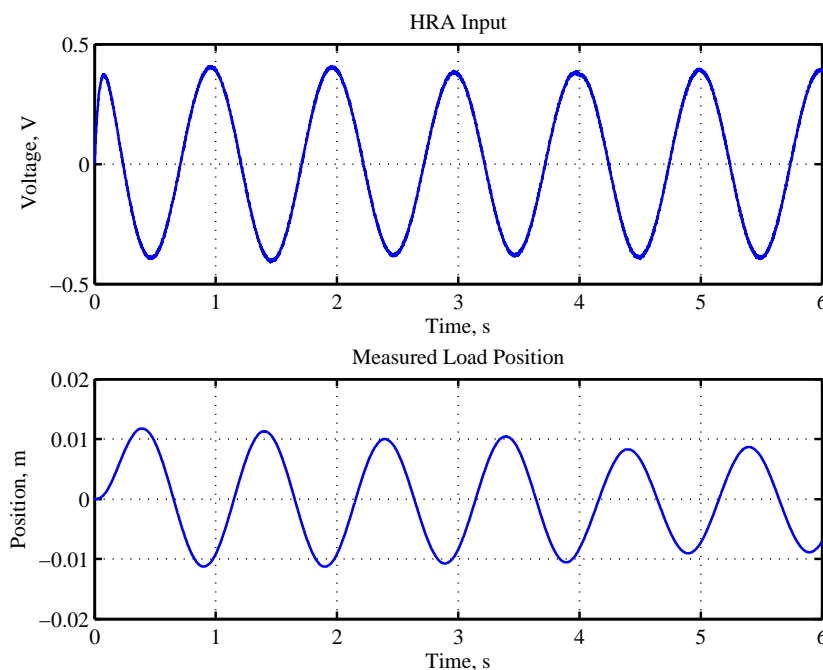


Figure 6.16: Measured load position output and known voltage input for HRA during lock-up faults.

(its corresponding single lock-up mode), however, the double lock-up mode is converged upon reasonably quickly.

The first reduction in capability is detected after 0.142 seconds, and the second reduction is confirmed by the FDI 0.436 seconds after the second fault occurs.

6.3.3.3 Summary

The simulation results show that the IMM FDI method used produces accurate diagnosis of capability levels in lock-up and loose fault condition. This result, like the rule-based approach, is sensitive to the input of the system. However, low excitation in this case can cause false diagnosis, rather than just delayed diagnosis. One solution to this problem may be to combine the IMM output with some logic that holds the previous estimated state if the excitation falls below a threshold.

6.4 Conclusions

This chapter has discussed the requirement for fault detection in the HRA for use in control reconfiguration, or for general health monitoring purposes. These two applications of FDI in the HRA have distinct requirements and resources, and as such two methods of FDI were discussed: a rule-based approach for use in AFTC and an interacting multiple-model method for health monitoring.

The rule-based method detected the location and nature of faults quickly and accurately using localised simple algorithms. However, the distributed nature of this method means

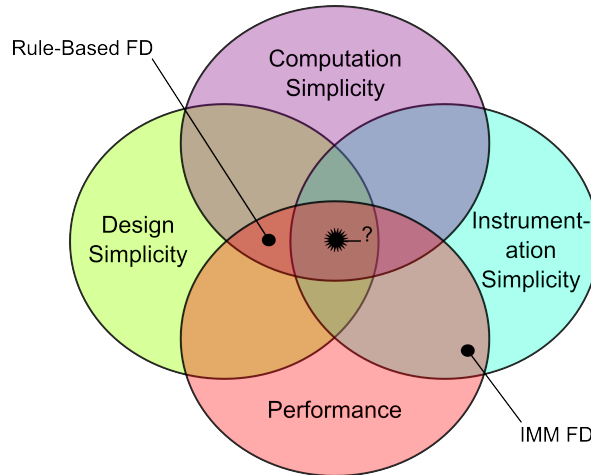


Figure 6.17: Fault detection properties.

that localised sensors are required, which is appropriate to the multi-agent control method presented in Chapter 7, but is unlikely to be feasible for general health monitoring in a passive FTC scheme such as that of Chapter 5.

Hence, the IMM FDI approach described in Section 6.3 aimed at providing a simplified diagnosis (indicating the capability of the HRA) utilising less sensory information. Whilst the detection rates using this approach were reduced slightly in comparison to the rule-based FDI, the correct travel and force capability was estimated, and detection time is not as critical in general health monitoring.

Despite the simplifications provided by the reduced-order filters and mode allocation method (in comparison to the full mode representation presented in (161)), the IMM FDI is still far more complex in terms of the design effort and computational demands in comparison to the rule-based approach.

A trade-off seems to exist in these two methods between where complexity in the system lies i.e. if further complexity in the system (such as extra sensors) can not be justified, than complexity within the FDI scheme itself is required. This idea is illustrated in Figure 6.17. An ideal fault method should be simple in terms of its design, instrumentation and computational demands, whilst providing performance in terms of accuracy, speed and reliability. This is demanding, however, and in reality one must consider which aspects have the most rewards, and which the most penalties for the purpose in mind. The rule-based method has many of these desirable properties, however, justifying the instrumentation cost is difficult if it is for health monitoring alone, and thus the IMM approach may be considered more suitable.

These two approaches are by no means the only fault detection methods that could be applied to the HRA, as was illustrated in Chapter 2. The field of fault detection is diverse, and it is likely that other methods which satisfy more of these properties exist. However, health monitoring is not the main concern of the current project phase, and the rule-based detection presented here can now be used in conjunction with control reconfiguration to form an AFTC scheme in the next chapter.

Chapter 7

Active Fault Tolerant Control of High Redundancy Actuation

7.1 Introduction

One of the key objectives of the work carried out is to investigate active fault tolerant control approaches based upon multi-agent concepts for the HRA and quantify any benefits that can be achieved in terms of performance and fault tolerance, whilst also evaluating the associated increases in complexity, and uncertainty. To this end, Chapter 5 provided a passive fault tolerant control performance for HRA, which acts as a benchmark. In this chapter, an active fault tolerant control strategy is outlined, the results of which are compared to the passive benchmark.

A multi-agent control approach was chosen as the active FTC strategy. A detailed rationale for this choice is given in Section 7.2. The initial impetus for taking this approach however, was concerns with complexity. The HRA is complex, as it contains many moving masses making the system high-order. In addition, this system is changeable due to the occurrence of faults. Although Chapter 5 has shown that a very simple passive approach can provide some degree of fault tolerance, previous to the current stage of the HRA project, concerns were raised regarding whether passive FTC could provide adequate control for such a complex and changeable system. It was thought that one way in which to deal with this complexity may be a divide-and-conquer approach: providing active control and fault detection on a localised basis. Multi-agent concepts offer ideas for implementing such a scheme, and thus this approach was investigated.

Chapter 2 provided an introduction to multi-agent systems (Section 2.5), and a discussion of concepts central to multi-agent systems (Section 2.5.4). These form a background and basis upon which the multi-agent control strategy for the HRA is designed, details of which are given in this Chapter (Section 7.3).

7.2 Rationale for Multi-Agent Control of HRA

Taking a multi-agent based perspective on HRA control design can provide two key features:

- Structuring.
- Flexibility.

Multi Agent Control (MAC) and HRA are structurally similar. Both are inspired by natural mechanisms which utilise large numbers of relatively simple cells/processes to form complex structures/behaviours. The HRA, viewed as a whole is a complex, changeable system. An unstructured approach to applying active FTC to this system is likely to make control reconfiguration complicated and fault diagnosis difficult. However, if the HRA is viewed as a collection of simpler (if not similar) subsystems, then simple control reconfiguration and simple fault detection can be applied on a local level, and MAC can provide a framework for this.

The structuring of control is often neglected within the field of control engineering, as the problem is stated in the form of a single plant model (145). The process industry acknowledges that the structuring of control is an important issue in complex systems, thus it is given more attention in this field and numerous MAC systems have been proposed within this application area, for example (162).

Equally, a structured approach to control may be achieved through use of decentralised control techniques (163; 164). However, these techniques do not necessarily facilitate the application of localised control reconfiguration and fault detection. In addition, the abstract approach to the control problem offered by MA concepts frees the design from the usual conventions. For example, the sharing of system parameters, capabilities and intentions are possibilities that may be derived from the multi-agent concept, but would not be considered within conventional distribution of control, as signals tend to be directly measured quantities (96). This interaction between the agents is important as it implicitly acknowledges the interaction between the HRA elements.

The flexibility and structuring provided by MAC also has advantages over more conventional active FTC techniques. Localisation of decision-making capabilities avoids the issue of single point-of-failure incurred by active FTC schemes that employ centralised fault detection or supervisors. The flexibility afforded by the communication involved in the agent approach also offers complex active control strategies to be employed with greater ease.

Hence, it is the combination of both structuring and flexibility that motivates the use of MAC above conventional decentralised control and centralised active FTC techniques. Nonetheless, there are a number of potential issues associated with MASs that require careful attention such as deliberation, communication and negotiation delays, agent non-consensus and communication failure.

7.3 Design of Multi-Agent Control of HRA

The discussion of concepts central to multi-agent systems given in Chapter 2 illustrated the diversity that exists in the field and demonstrated that there is no definitive agent or agency architecture, communication structure or protocol that must be abided by in order for a system to be considered a MAS. Rather, a multi-agent application is simply one that encompasses

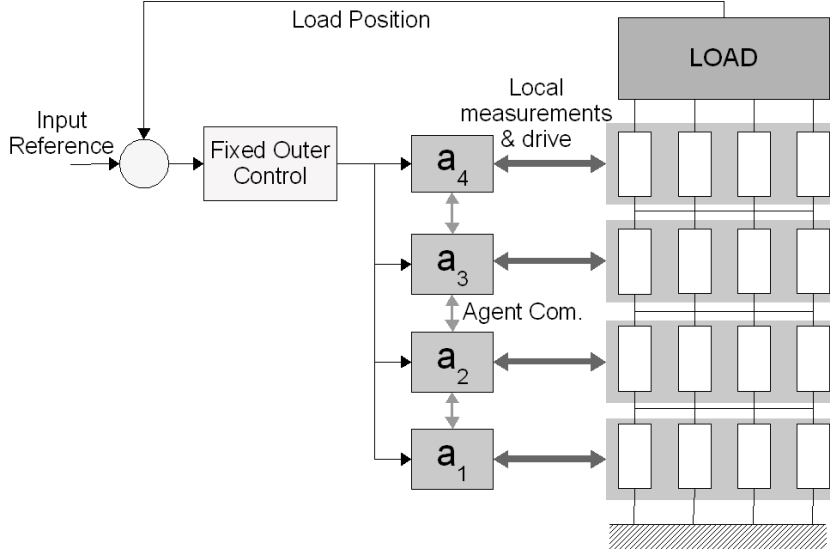


Figure 7.1: Agency structure for PS HRA.

some of the defining characteristics of the concept of agents. Thus, it is pertinent to consider which aspects of these agent architectures and agency structures are most useful for the problem at hand: Multi-Agent Control of HRA (MACHRA), and on this basis form an appropriate MACHRA strategy.

7.3.1 MACHRA Agency Structure

The first consideration is allocation of agents in the HRA. How should the role and remit of each agent be defined? As the control of the HRA's actuation resources during health state changes is the main requirement, then the most obvious allocation of agents is according to the physical resources i.e. per actuation unit. In PS configurations, an agent is assigned to each parallel bank of actuators and to each serial bank in the SP configuration. Thus, in a $m \times n$ HRA an agency A consists of a set of agents:

$$A_{PSm \times n} = \{a_1, a_2, \dots, a_m\} \quad (7.1)$$

$$A_{SPm \times n} = \{a_1, a_2, \dots, a_n\} \quad (7.2)$$

This structure is illustrated in Figures 7.1 and 7.2.

An agent is used per bank due to the nature of the faults considered. A lock-up fault effectively locks all parallel elements, and likewise a loose fault effects all serial actuators, and as such the resultant control changes made will be the same in this branch regardless of which

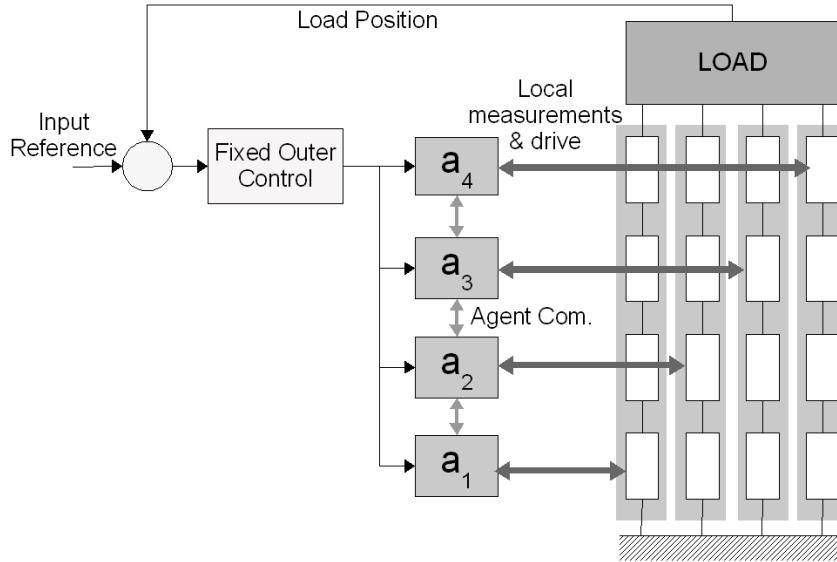


Figure 7.2: Agency structure for SP HRA.

element fails. In more mixed configurations, agents may be allocated per parallel and serial branch or per individual actuation unit. However, these configurations are not within the scope of this work.

Having assigned agents to subsystems, it is necessary to define their responsibilities. Localisation is a key characteristic of MASs, thus each agent should use local sensory information and exert local influence. Each moving coil actuation element has an integral glass-slide encoder providing local position information and the coil current of each element is readily available. Using these local measurements, each agent must provide local control drive to its actuation element(s) and detect faults in their actuators i.e. sense the state of their subsystem. In order to compensate for reduced capability in other actuation elements, it is necessary that the agents have knowledge regarding the overall state of the system. Through communication, they can build up a representation of the overall health of the system and reconfigure their control as necessary.

Communication of simple information regarding the agent's believed state of the system needs to be quick and efficient. Considering the types of communication discussed in Section 2.5.4.2, then direct or message-based communication seems the most appropriate. Indirect forms of communication are not suitable as they are too slow and the resultant consequences too ill-defined for a safety-critical application. High level language is also not suitable as such complexity and flexibility is not needed. Direct communication is chosen for use in this case as there are a limited number of situations to communicate throughout the system, which can be finitely defined.

The chosen structure of the communications is point-to-point. Each agent communicates with its direct neighbours, a_{n-1} and a_{n+1} . This approach is preferred to a broadcast method as it reduces the communication traffic in the system. Broadcast structures are useful when the presence of agents in the system is uncertain, but this is less relevant for the HRA as the HRA starts in a fixed configuration with a fixed agent structure: no more resources or agents will be introduced into the system as is the case in applications such as distributed power control (135; 136). Although, in a real application some provision may need to be made to salvage communication links in the event of an individual agent malfunction. This could be achieved by implementing a white pages agent similar to that discussed in the FIPA abstract architecture (Figure 2.19) where a record is kept of the agents in the system and their communication links, and, on loss of communication, an agent can consult these white pages to re-establish a communication link with the next available agent.

The final agency-related consideration is co-ordination. In the MACRHA scheme, co-ordination is achieved through built-in standardisation. This approach is chosen as it will provide the most reactivity, as no context establishing communication or mediation needs to be conducted. The built-in nature of the standardisation avoids problems with hierarchical supervision, which can cause single points-of-failure. It also avoids the uncertainty associated with over-flexibility, which can be an issue for high integrity application verification.

7.3.2 MACHRA Agent Architecture

Due to the fault types and system dynamics, the HRA requires a control strategy that will respond quickly to faults, making reactivity a key requirement. For this reason, a hybrid agent architecture is chosen. Logic based approaches, where complex models of the environment are formed and reasoned upon, are not appropriate due to this requirement for speed. Similarly, practical reasoning methods such as BDI may also be inappropriate due to the delays caused by reasoning. Additionally, the HRA generally does not have changing goals: the goal is to provide a performance that is as close to nominal as possible. Thus, having separate desires and intentions is superfluous. A hybrid approach is preferred over a simply reactive approach, as this allows the retention of state. Simple internal state and system state models are retained in each agent and the control action is changed in accordance with these.

The agent architecture is vertically layered, containing fault detection, communication and control layers as shown in Figure 7.3. These layers are discussed individually below.

7.3.2.1 Fault Detection Layer

The top layer contains the fault detection for the agent, where loose faults and lock-up faults are detected. The input to this layer consists of internally available quantities such as the local position error and the local measurements available. In the PS configuration case this is a single local position measurement, and n coil current signals (for n parallel actuation elements). In the SP configuration, n local position measurements and n coil current signals are used (where n is the number of serial actuation elements in this case).

The detection algorithms use the simple rule-based approach described in Chapter 6. The

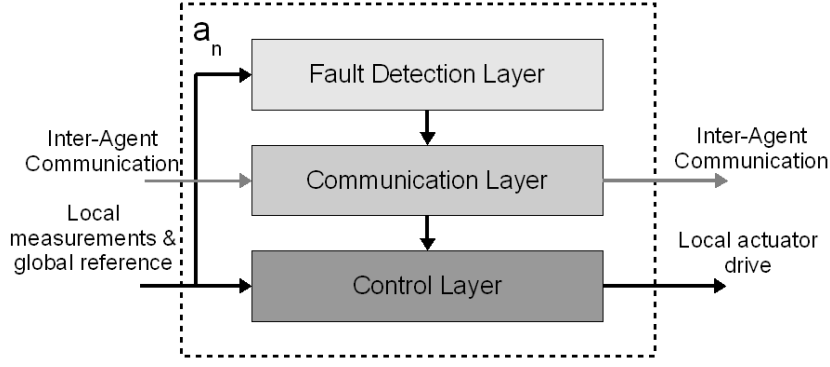


Figure 7.3: Agent architecture

outputs of these fault detection algorithms are combined to provide a three term model of the internal fault state, which is passed to the next layer:

$$F_0 = \begin{bmatrix} f_{loose} & f_{unloose} & f_{lock} \end{bmatrix} \quad (7.3)$$

where f_{loose} is the number of loose faults detected (with a maximum of n in n parallel elements and 1 in n serial elements) and f_{lock} is the number of lock-up faults detected (with a maximum of n in n serial elements and 1 in n parallel elements). There are maximums of 1 due to the effect of the fault on in the parallel and serial elements. A single lock-up in parallel elements essentially locks all the elements in that branch, and similarly a single loose fault in serial elements is effectively a loose serial branch.

The middle term, $f_{unloose}$, is only used in the PS configurations and represents the number of loose faults that previously occurred in a branch before that branch became locked. If the branch locks after a loose fault has occurred in that branch, then it effectively cancels the effect of that loose fault, and it is necessary to communicate this to other agents.

7.3.2.2 Communication Layer

The output of the fault detection layer is fed into the communication layer along with the messages F_l and F_r from the agents a_{n-1} and a_{n+1} respectively. The communication layer takes these messages and combines them with the internal state F_0 to form two messages representing the cumulative faults to the left F_l and the right F_r , which are, in turn, passed to a_{n+1} and a_{n-1} respectively. The communication is formed from three values:

$$F_l = \begin{bmatrix} f_{lloose} & f_{ldeloose} & f_{llock} \end{bmatrix} \quad (7.4)$$

$$F_r = \begin{bmatrix} f_{rloose} & f_{rdeloose} & f_{rlock} \end{bmatrix} \quad (7.5)$$

where each quantity is the number of fault types to the left or right. From F_l , F_r , and the internal fault status F_0 , a fault state model for the whole system can be formed. In the PS case this is stated as:

$$F_T = \begin{bmatrix} f_{tloose} & f_{tlock} \end{bmatrix} \quad (7.6)$$

where the first term is the total number of loose elements in the system and the second is the total number of lock-ups. In the PS case these values are defined as:

$$f_{tloose} = F_0(1) + F_l(1) + F_r(1) - F_0(2) - F_l(2) - F_r(2) \quad (7.7)$$

$$f_{tlock} = F_0(3) + F_l(3) + F_r(3) \quad (7.8)$$

And in the SP case:

$$f_{tloose} = F_0(1) + F_l(1) + F_r(1) \quad (7.9)$$

$$f_{tlock} = F_0(3) + F_l(3) + F_r(3) \quad (7.10)$$

This communication procedure is illustrated by an example in Figure 7.4. A loose fault occurs in the elements associated with a_2 and this is communicated to the other agents.

As there are a finite number of internal and system fault states, based upon these two quantities, a finite set of perceptions P for the system can be formed:

$$P = \{p_{11}, p_{12}, \dots, p_{ij}\} \quad (7.11)$$

where:

$$p_{mn} = \begin{bmatrix} F_0(m) & F_T(n) \end{bmatrix} \quad (7.12)$$

As mentioned in Section 2.5.4.2, in general P does not necessarily map to the actual system state S , which is certainly true in this case. Each perception only represents the effective number of faults in the system. There is no representation of the location of these faults. This simplification that occurs between S and P is not necessarily an issue, however. Reflecting back to the work on passive control detailed in Chapter 5, it was shown that the number of effective faults was the major influence on system behaviour. Thus, this representation is mostly¹ adequate for control purposes and reduces the complexity of the communication and internal models significantly.

Each perception p maps to a pre-designed control action c :

$$P \rightarrow C \quad (7.13)$$

where:

$$C = \{c_{11}, c_{12}, \dots, c_{ij}\} \quad (7.14)$$

Hence, F_0 and F_T are passed to the control layer where the control action is implemented.

7.3.2.3 Control Layer

Figure 7.5 is a representation of the control layer for PS configurations. The input to the control layer is from a global control loop (Figures 7.1 and 7.2). This outer control loop provides integral action to ensure any steady-state error is removed. As this control law is

¹In the case of lock-up faults in the 10×10 system this representation is not sufficient to satisfy all possible fault modes. This is discussed further in Section 7.4.

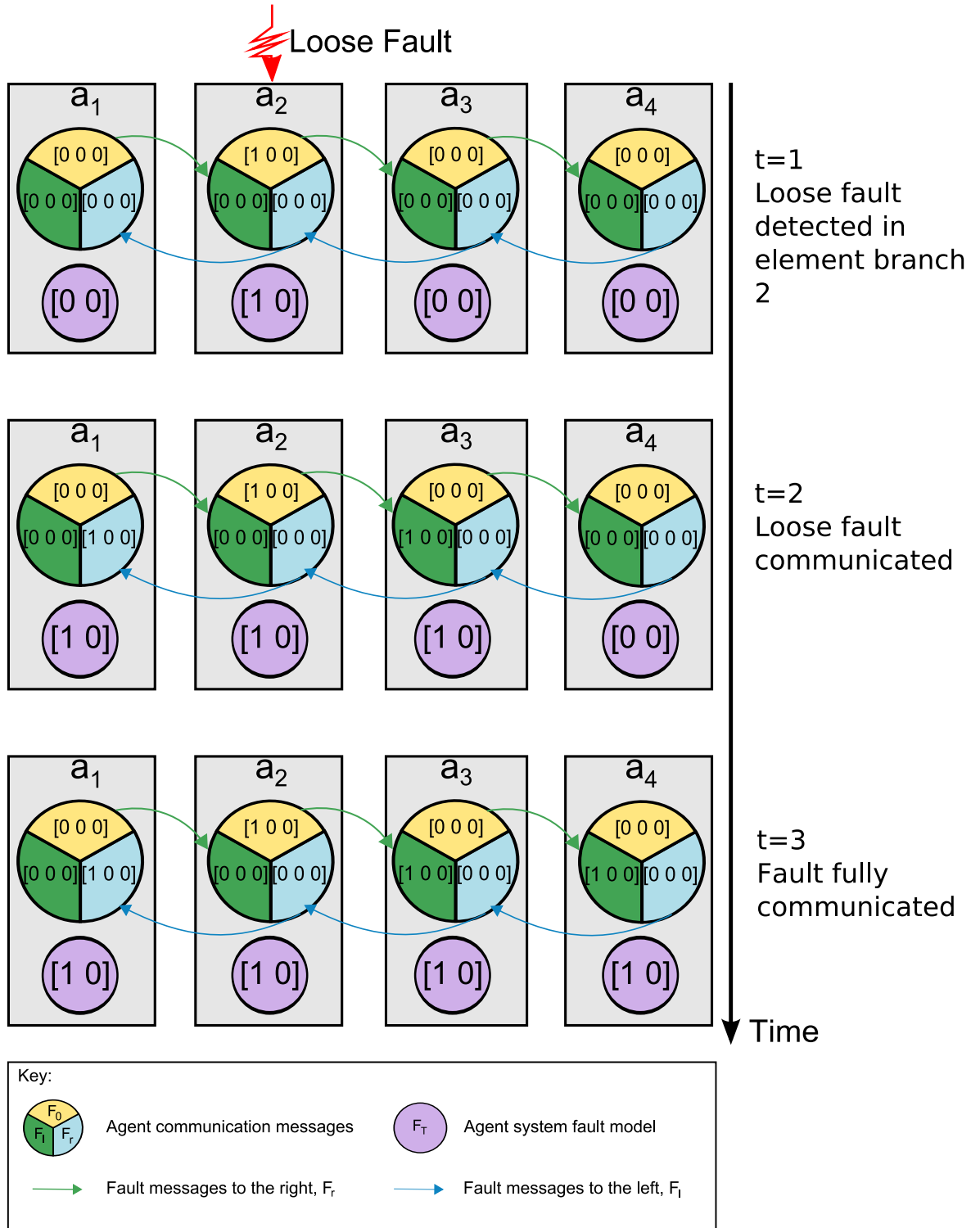


Figure 7.4: Fault communication example

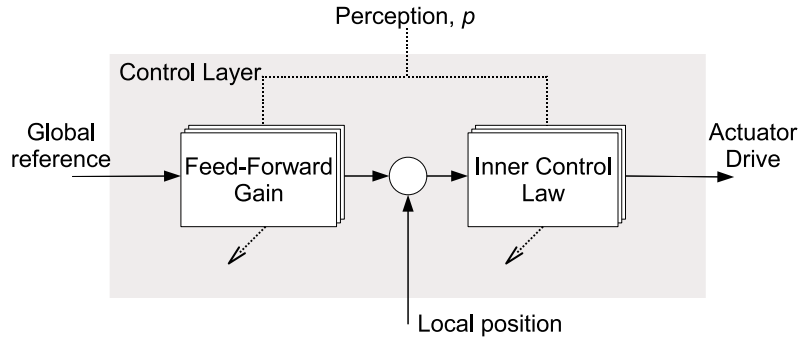


Figure 7.5: Control layer

fixed, it is not considered part of the active control or a supervisor of the agents. This is important, as this removes any decision-making single-point-of failure from the agency, and retains an autonomous agency architecture.

The control within the layer consists of a feed-forward gain, which is scheduled in order to keep the gain in the system equal under fault conditions, and an inner local position control loop with a classical controller. The control parameters are chosen from a bank of pre-designed controllers depending on the perception input, p . In the SP case, there are actually four identical inner-loop controllers, one for each local position measurement.

The overall control effect of this scheme is a decentralised gain scheduling and multiple-model active FTC strategy. In this case, 'multiple-model' means that a number of control laws are based on a number of fault models for the system. This form of offline design, multiple-model control, has been chosen for use within this scheme as a pre-designed control law is more easily verifiable for high integrity applications. The stability of online control synthesis is more difficult to ensure, and the uncertainty and complexity involved can be a barrier to industrial implementation. Whilst the MAC approach described here is complex in that there are effectively many control loops, the stability of the control can be verified offline, and this sort of multi-loop complexity is used widely in application areas such as aerospace.

7.3.3 Control Design

As stated in equation 7.13, for each perception a control action exists, which must be pre-designed. The approach to design is discussed within this section. The actual controller values are available within Appendix D.

7.3.3.1 Nominal Control

The nominal inner-control law used is the phase advance controller designed in the passive FTC case. The feed-forward gain is related to the travel capability. In the nominal case, the 4×4 configurations have a travel capability of 4 actuation elements, and the 10×10 configuration has a travel capability of 10. The feed-forward gain spreads the input reference to the inner-loops between the remaining capability, hence, the nominal feed-forward gain is

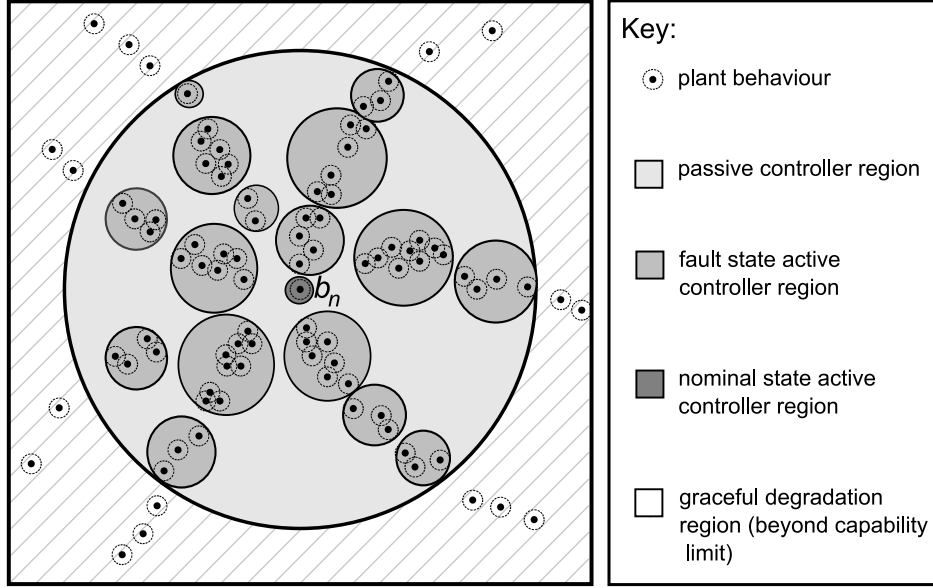


Figure 7.6: Multiple-model and passive control design

1/4 for the 4×4 HRAs and 1/10 for the 10×10 HRA. The fixed outer-loop PI controller is then designed to provide a transient response and stability margins that are similar to the benchmark passive FTC presented in Chapter 5.

7.3.3.2 Fault Control

For each perceived fault case, new inner-loop control parameters are designed based on fault models of the system. The value of the phase advance time constant, τ , is adjusted to re-cancel the pole that was cancelled by the nominal controller. This brings the response of the faulty system nearer to the nominal response (this will be illustrated in Section 7.4). In addition, the feed-forward gain is also re-scheduled when reductions in travel capability occur.

However, as discussed in Section 7.3.2, a new controller is not designed for each actual system state, but rather for each perceived state. This may be visualised using the same approach as discussed in Chapter 5 (Figure 5.1). Figure 7.6 shows the behavioural space that the plant state behaviours B inhabit. The passive approach designed a controller that would encompass as many of B as possible, but the results were shown to be somewhat conservative. Active multiple-model control provides the possibility of designing more controllers to suit each member of B . Within this multi-agent approach however, a controller is designed only for each perception p , which encompasses a number of fault modes. Each controller must be optimised to satisfy the requirements with each member of p . Optimisation has been achieved here in a very basic way, by manually tuning each controller. Nonetheless, a more complex/automated approach such as genetic algorithms could also be taken.

The closed-loop behaviour may then be represented as in Figure 7.7. The active control performance region, that represents the behaviour of the system under fault, is closer to the nominal performance in comparison that of the the passive controller performance. It is apparent that a trade-off exists here between the number of fault model controllers (or per-

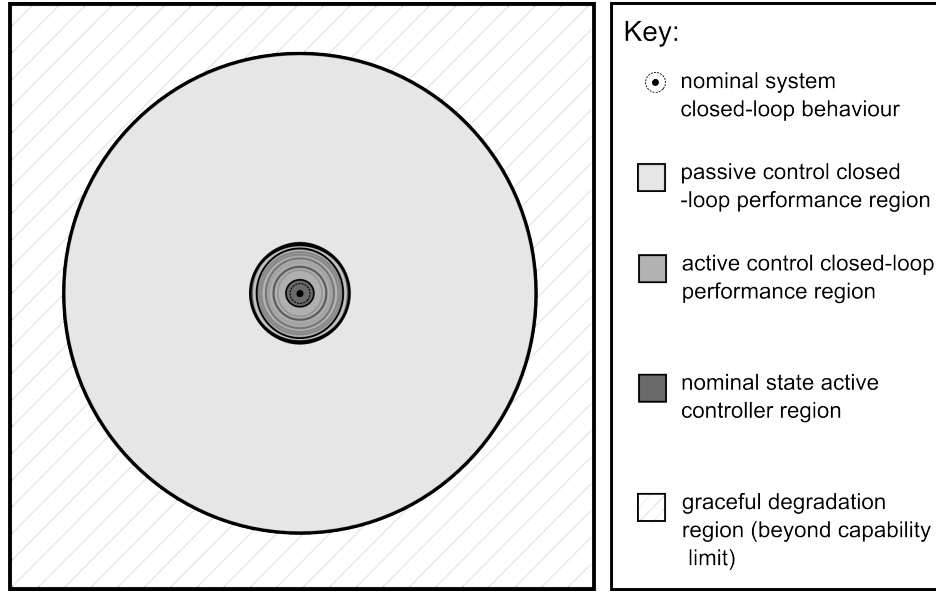


Figure 7.7: Multiple-model and passive control performance

ceptions in the multi-agent control approach) and the performance. As the ratio of controllers to fault behaviours approaches 1:1, the conservativeness of the controllers are also reduced and a performance nearer to that of the nominal system can be achieved. However, more controllers increases complexity in the system and increases communication demands. Hence, a balance must be struck between achieving an adequate performance and the practicalities of design and implementation.

7.4 Fault Simulations

Having described the multi-agent approach taken and the control design process, simulations of the systems under fault are presented here. Firstly, the static fault control performance is assessed and compared to the passive control performance by simulating the same fault profiles as presented in Chapter 5. The term static is used here to represent the control reconfiguration state in the system. These static simulations represent the system performance after the fault has been detected, communicated, and the control reconfigured, and any post-reconfiguration transients in the system settled.

The control performance in a more fault dynamic sense is addressed in Section 7.4.2, where faults are injected and the influence of detection and reconfiguration time considered.

7.4.1 Static Fault State Simulations

The static fault condition performance of the MAC controlled HRA is examined and compared to the passive FTC approach within this section.

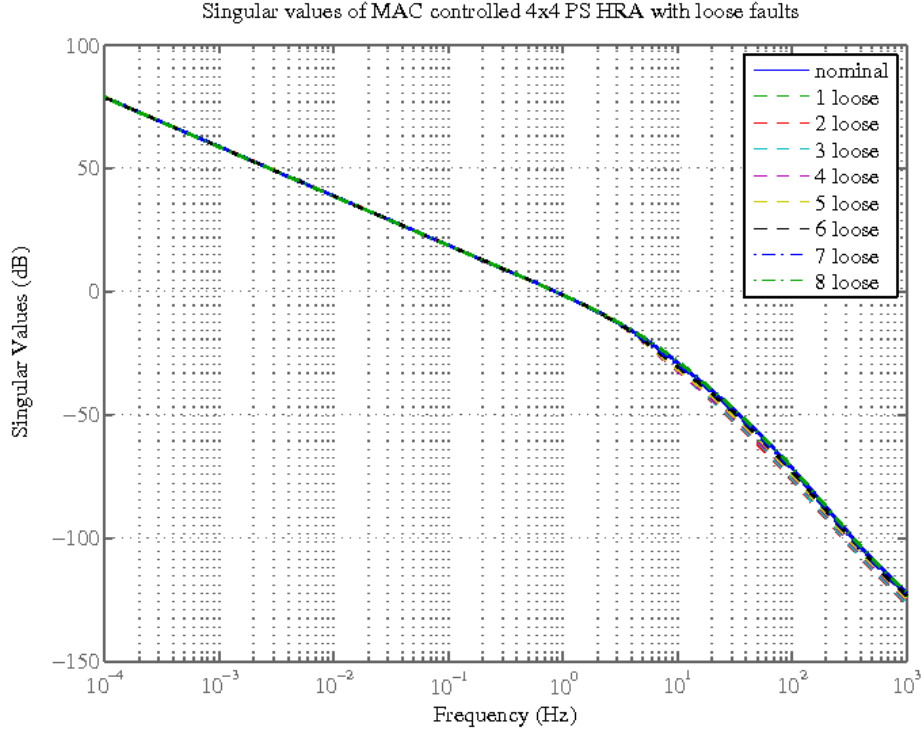


Figure 7.8: Singular values of nominal and loose MAC controlled 4×4 PS HRA.

7.4.1.1 Loose faults

Figures 7.8 to 7.10 show the frequency response of the open loop systems under loose fault conditions with MAC and Figures 7.11 to 7.13 provide the transient responses. Table 7.1 summarises these figures.

The PS HRA's performance under loose fault conditions is very close to that of the nominal system. The overshoot problems that were witnessed with passive control have been eradicated with the MAC. Consequently, the system meets the performance criteria under all the loose fault conditions.

The SP HRA transient response exhibits similar non-linear effects to those in the passively controlled case (Figure 5.41) due to the loose elements in the system hitting the end-stops. This is more clearly illustrated in Figure 7.14, where the relative positions of the elements are also provided for the SP 4×4 system response to a pulse train, whilst there are two loose faults in elements e_{43} and e_{44} . As before, the force exerted by the remaining active elements in the loose branches (cyan), is not translated by the loose elements (red), causing the loose elements to hit the end-stops. The uncontrollability of the loose element is exacerbated by the control driving the remaining active elements in the loose branch.

There is potential for suppression of this non-linear behaviour in the multi-agent control approach, as the control scheme in the loose branch can also be reconfigured. If the input to these elements is set to zero, or in other words if they are 'de-activated', then the affect on the remaining healthy elements is minimised (Figure 7.15). The loose elements still hit the end-

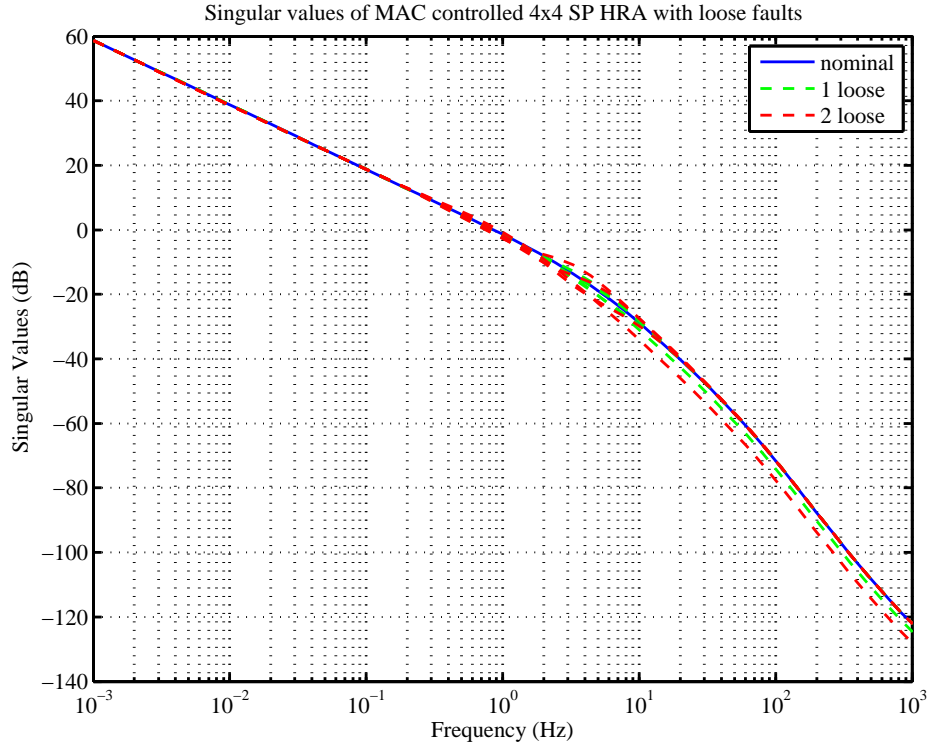


Figure 7.9: Singular values of nominal and loose MAC controlled 4×4 SP HRA.

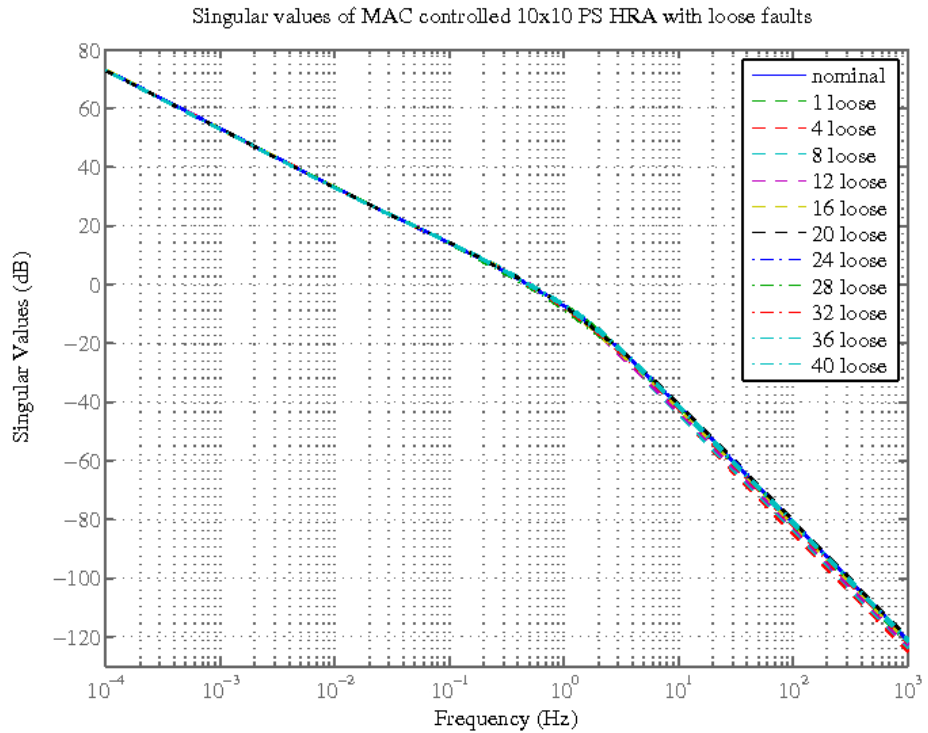


Figure 7.10: Singular values of nominal and loose MAC controlled 10×10 PS HRA.

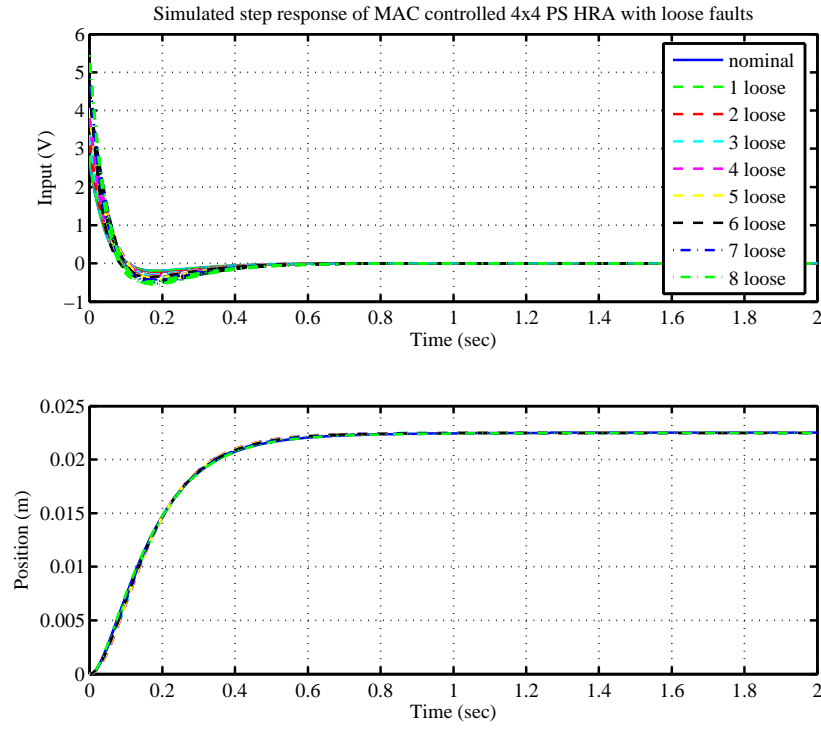


Figure 7.11: Step response of nominal and loose MAC controlled 4×4 PS HRA.

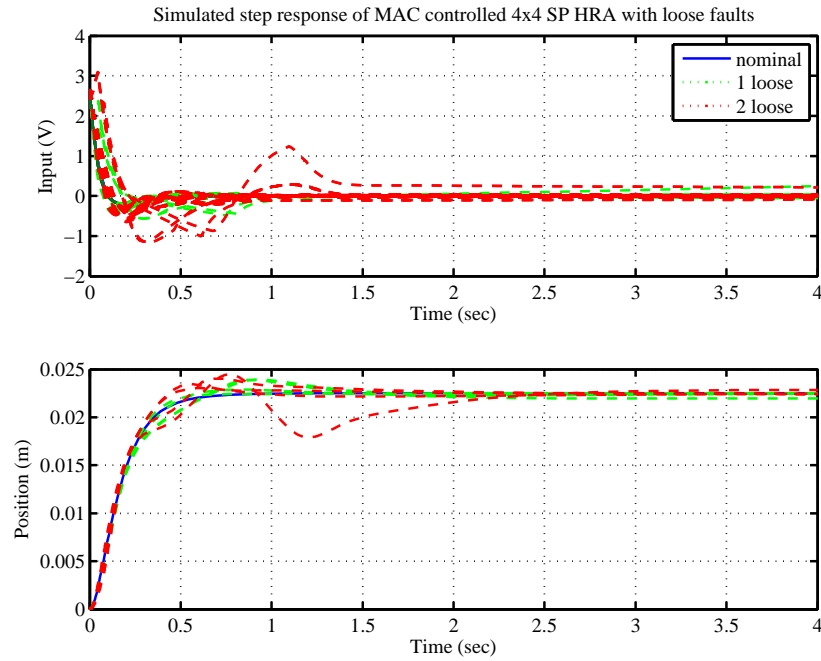


Figure 7.12: Step response of nominal and loose MAC controlled 4×4 SP HRA.

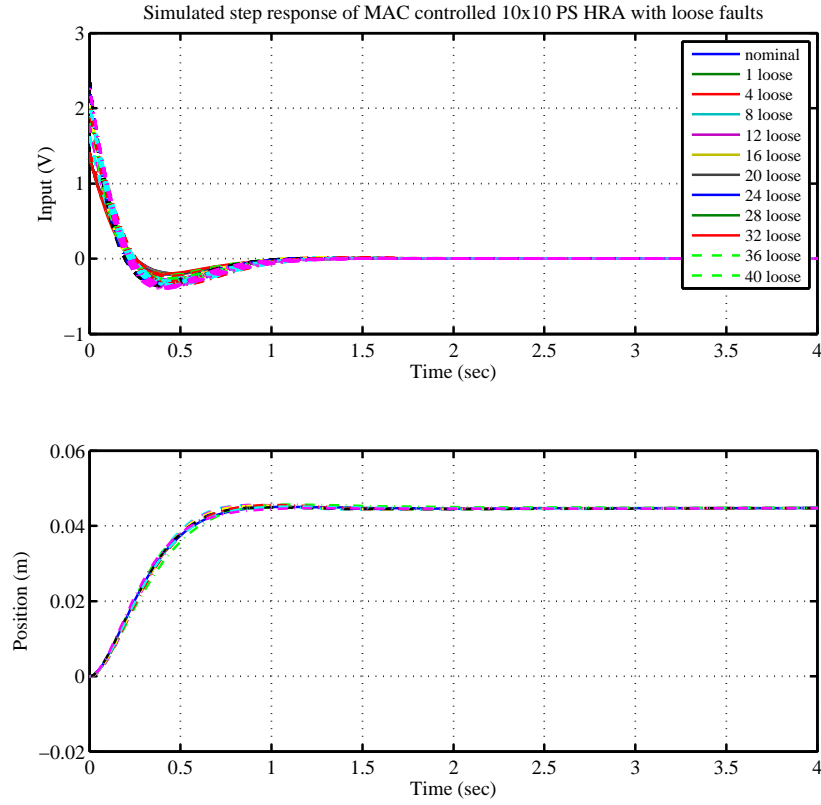


Figure 7.13: Step response of nominal and loose MAC controlled 10×10 PS HRA.

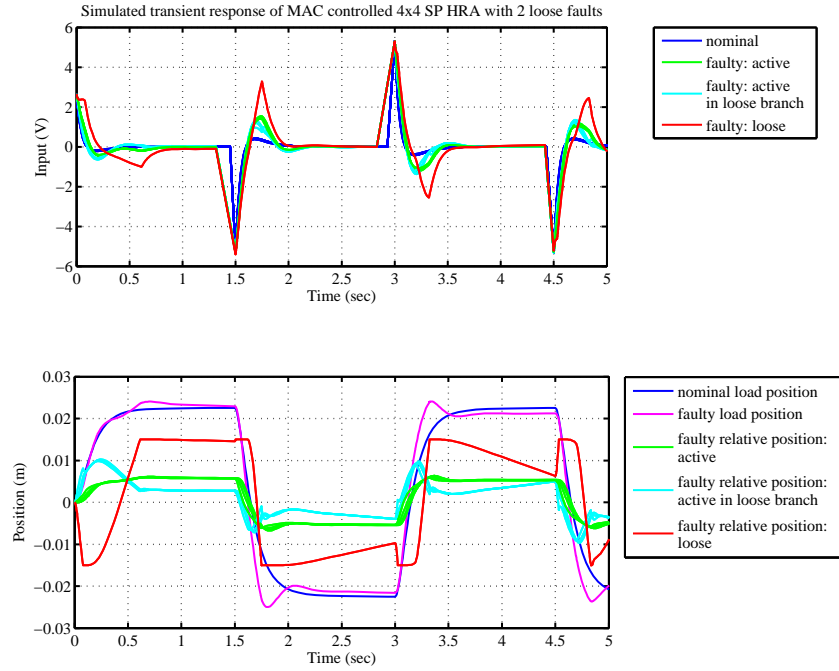


Figure 7.14: Response of SP 4×4 HRA with two loose faults in two branches to pulse train input.

Table 7.1: Stability margins and transient characteristics of example MAC controlled HRAs under nominal and loose fault conditions

4x4 PS HRA System state	PM (deg)	GM (dB)	BW (rads ⁻¹)	SSerr (%)	RT (s)	ST (s)	OS (%)
4x4 PS HRA							
Nominal	78.9	42.1	5.37	0	0.32	0.60	0.06
1 loose	77.7	39.7	5.32	0	0.32	0.57	0
2 loose	76.2	35.8	5.35	0	0.30	0.55	0
3 loose	76.1	38.5	5.34	0	0.30	0.54	0
4 loose	76.2	44.0	5.37	0	0.30	0.55	0
5 loose	76.5	43.8	5.38	0	0.31	0.56	0
6 loose	77.3	42.9	5.36	0	0.31	0.57	0
7 loose	78.0	42.2	5.39	0	0.33	0.58	0
8 loose	79.3	41.0	5.40	0	0.33	0.61	0
4x4 SP HRA							
Nominal	78.9	42.1	5.39	0	0.32	0.60	0.06
1 loose	76.92	42.0	5.08	0	0.33	0.53	0.72
2 loose	73.03	41.07	5.21	0	0.32	0.62	1.75
10x10 PS HRA							
Nominal	72.5	47.5	3.11	0	0.50	0.80	0.04
1 loose	71.9	45.2	3.24	0	0.52	0.79	0
4 loose	69.2	37.8	3.17	0	0.50	0.73	0.68
8 loose	68.1	44.1	3.27	0	0.45	0.68	1.06
12 loose	68.3	49.0	3.21	0	0.49	0.71	1.16
16 loose	69.5	48.2	3.21	0	0.46	0.70	0.69
20 loose	71.5	47.3	3.20	0	0.48	0.75	0
24 loose	70.9	47.1	3.16	0	0.49	0.75	0.34
28 loose	71.1	47.7	2.87	0	0.53	0.80	1.39
32 loose	70.0	46.9	3.07	0	0.49	0.74	1.13
36 loose	72.2	46.6	3.06	0	0.51	0.80	0.23
40 loose	74.3	45.8	3.20	0	0.51	0.90	0

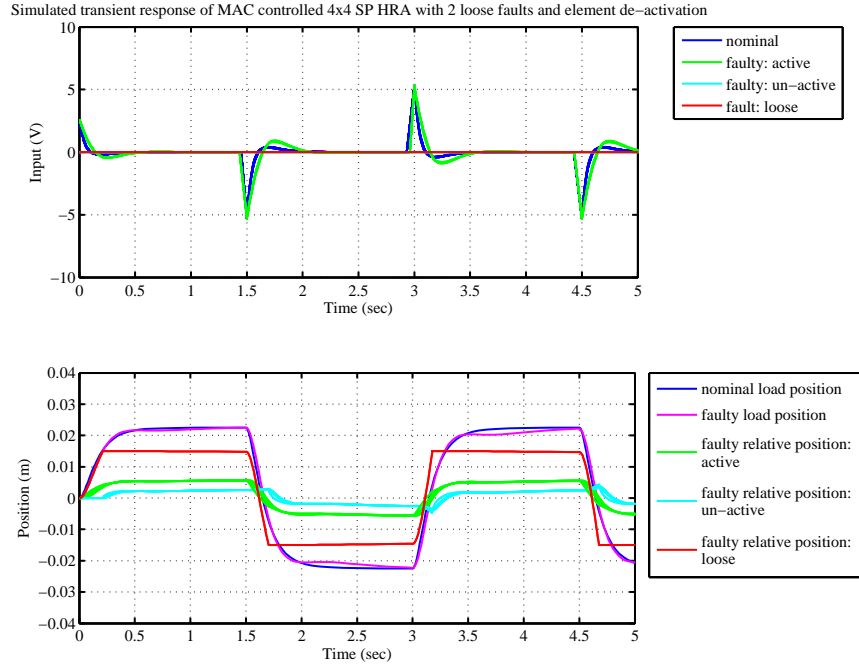


Figure 7.15: Response of SP 4×4 HRA with two loose faults in two branches to pulse train input with element de-activation.

stops, but not as frequently and they hit the end-stop in the direction of overall travel. There is also less energy being used within the loose branch elements, which may be considered advantageous. This sort of strategy is very simple to achieve in the MAC framework.

If this approach is implemented, then the transient response shown in Figure 7.16 can be achieved. Non-linear effects are still present, however, the large overshoot and steady-state transients are reduced.

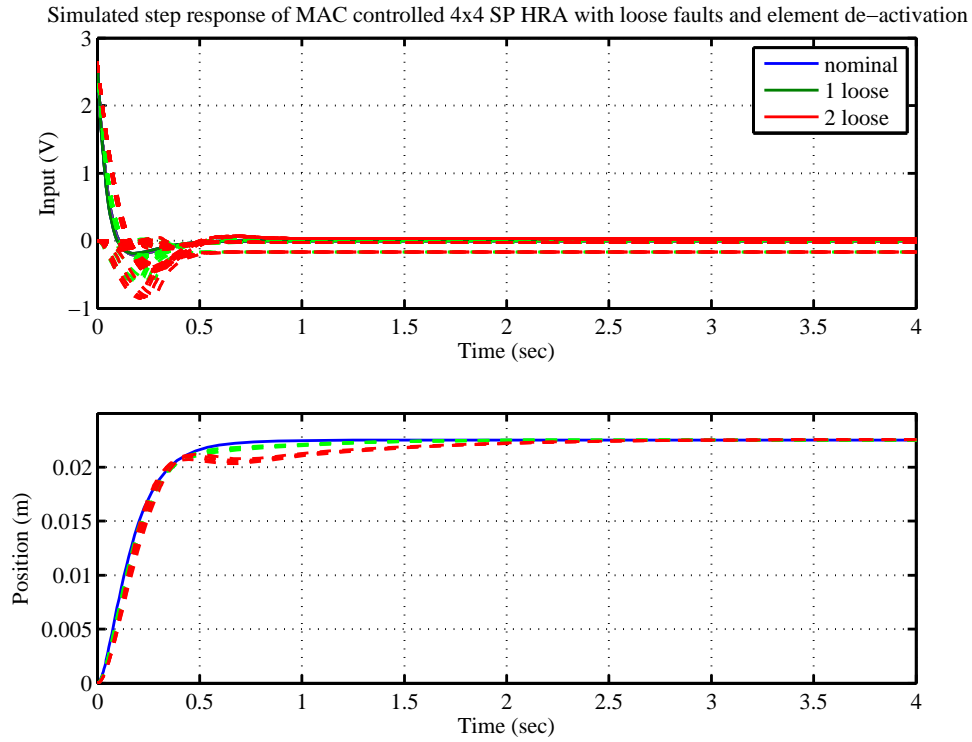


Figure 7.16: Step response of nominal and loose MAC controlled 4×4 SP HRA with element de-activation.

7.4.1.2 Lock-up faults

Figures 7.17 to 7.19 show the frequency response of the open loop systems under lock-up fault conditions with MAC and Figures 7.20 to 7.22 provide the transient responses. Table 7.2 summarises these figures.

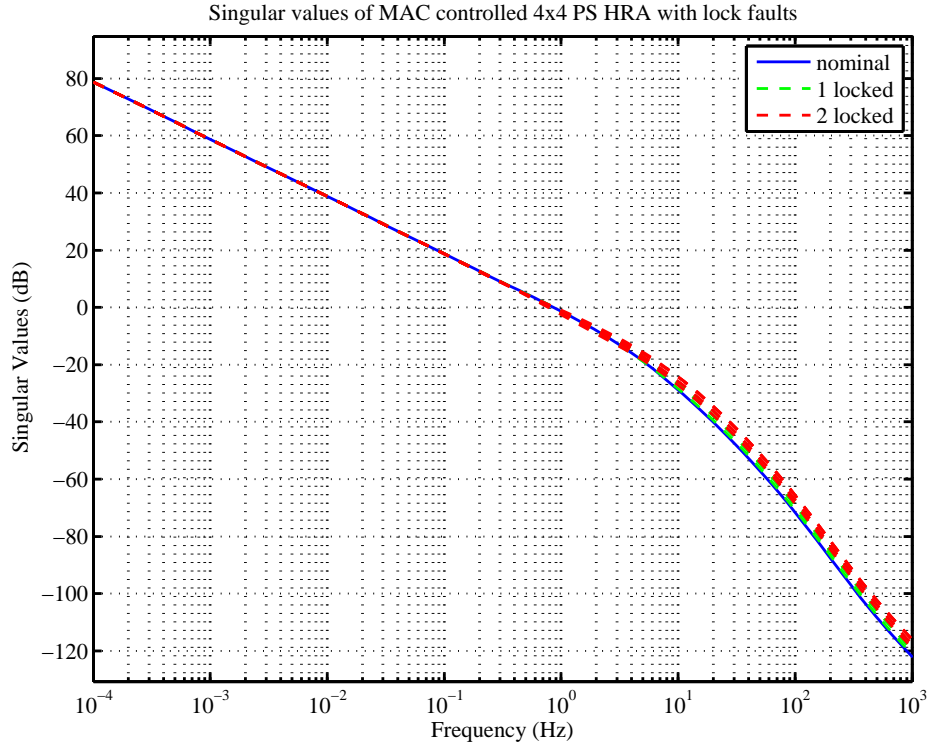


Figure 7.17: Singular values of nominal and locked up MAC controlled 4×4 PS HRA.

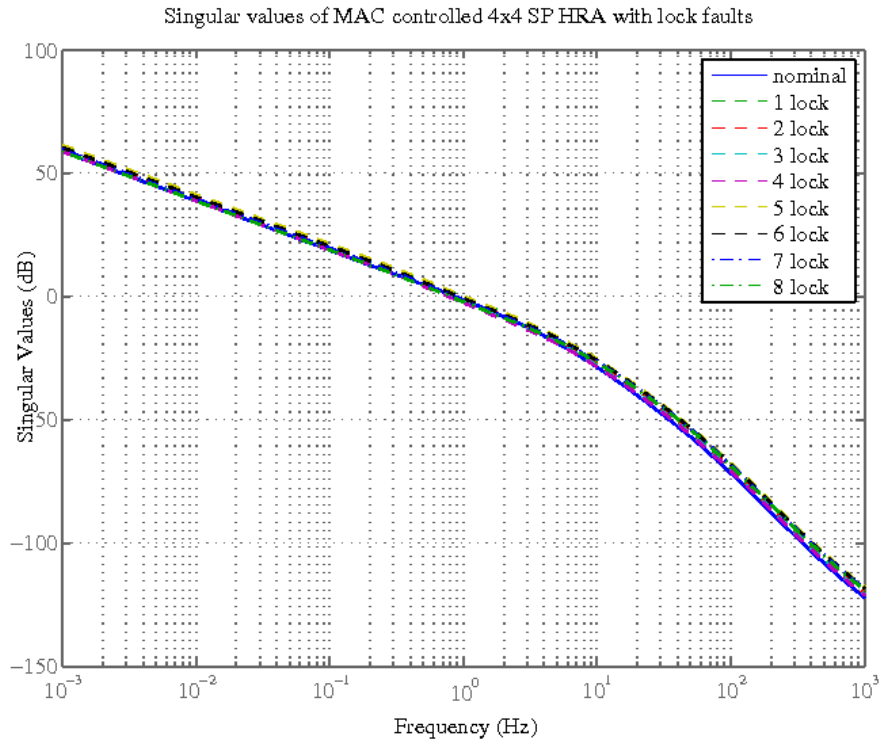


Figure 7.18: Singular values of nominal and locked up MAC controlled 4×4 SP HRA.

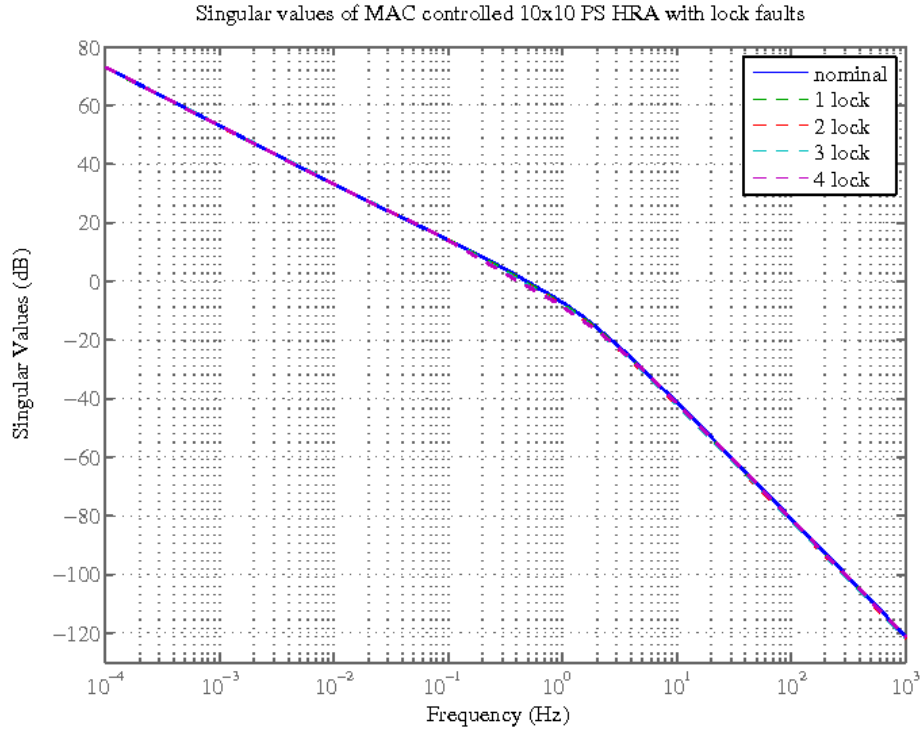


Figure 7.19: Singular values of nominal and loose MAC controlled 10×10 PS HRA.

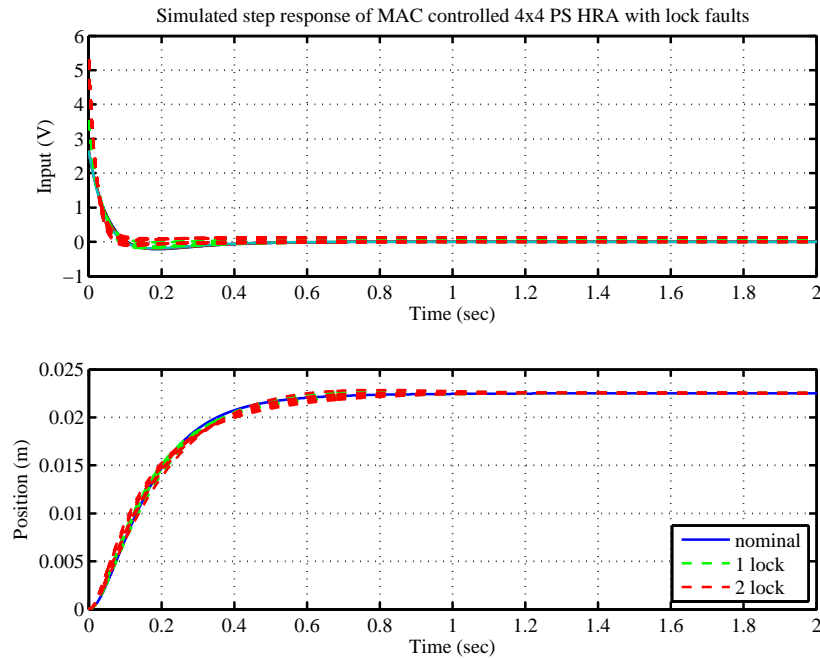


Figure 7.20: Step response of nominal and locked up MAC controlled 4×4 PS HRA.

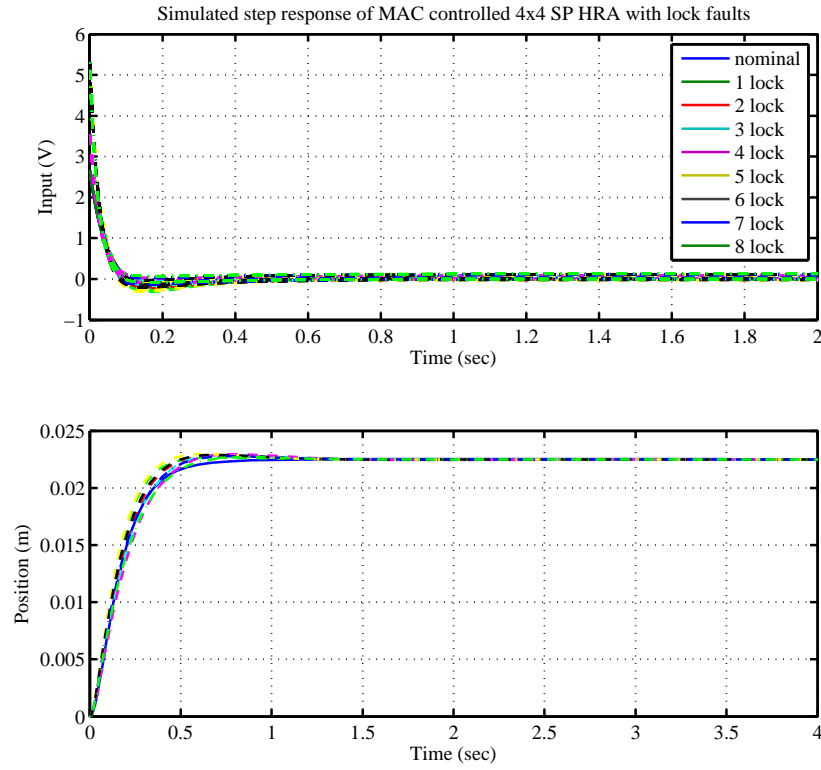


Figure 7.21: Step response of nominal and locked up MAC controlled 4×4 SP HRA.

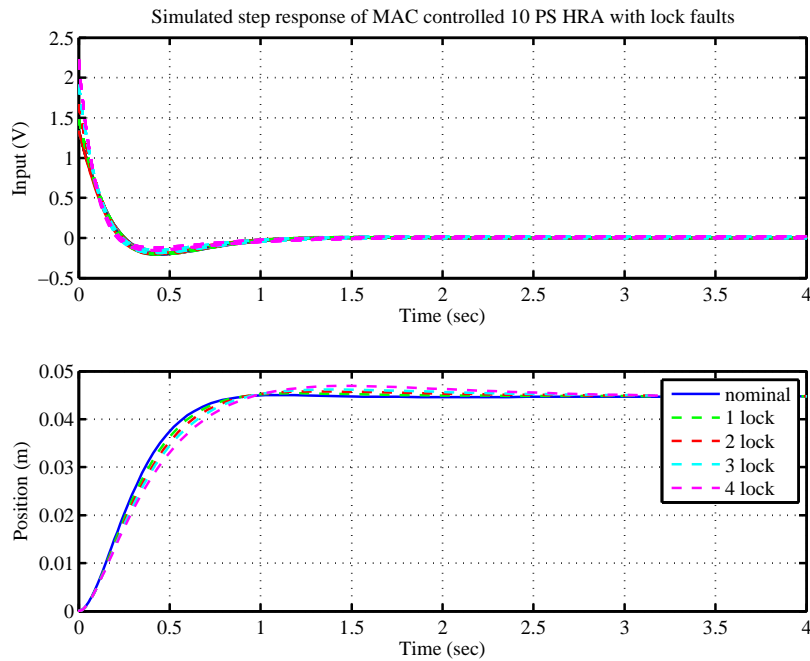


Figure 7.22: Step response of nominal and locked up MAC controlled 10×10 PS HRA.

Table 7.2: Stability margins and transient characteristics of example MAC controlled HRA under nominal and lock-up fault conditions

System state	PM (deg)	GM (dB)	BW (rads ⁻¹)	SSerr (%)	RT (s)	ST (s)	OS (%)
4x4 PS HRA							
Nominal	78.9	42.1	5.37	0	0.32	0.60	0.06
1 lock	81.1	42.2	5.27	0	0.36	0.63	0.39
2 lock	83.5	42.3	5.23	0	0.38	0.65	0.54
4x4 SP HRA							
Nominal	78.9	42.1	5.39	0	0.32	0.60	0.06
1 lock	76.4	40.7	6.23	0	0.27	0.44	0.63
2 lock	76.1	42.0	5.53	0	0.31	0.47	1.61
3 lock	76.2	42.8	5.12	0	0.34	0.49	1.80
4 lock	76.0	43.6	4.78	0	0.36	0.80	2.01
5 lock	76.2	40.8	6.33	0	0.27	0.49	1.96
6 lock	76.7	41.8	5.76	0	0.31	0.45	1.71
7 lock	78.1	42.6	5.35	0	0.34	0.50	1.17
8 lock	79.2	43.2	4.98	0	0.36	0.56	0.79
10x10 PS HRA							
Nominal	72.5	47.5	3.11	0	0.50	0.80	0.04
1 lock	72.5	47.8	2.93	0	0.53	0.82	0.91
2 lock	72.8	48.1	2.77	0	0.56	0.85	1.77
3 lock	72.9	48.4	2.64	0	0.58	1.77	2.61
4 lock	72.4	49.1	2.45	0	0.62	2.24	4.27

The dominantly serial nature of the SP configuration makes it more tolerant to lock-up faults. Nevertheless, in the passive control case, only up to 4 lock-up faults were accommodated. The MAC results show that with simple control reconfiguration, a performance within the requirements can be achieved under the full set of potentially tolerable faults.

The 4×4 PS HRA's performance under lock-up fault conditions is very close to that of the nominal system. The settling time issues that occurred in the passive control case have been significantly reduced, and both 1 and 2 lock-up faults are now tolerable within the performance specifications.

In the 10×10 PS HRA simulations, the system meets the requirements with 1-2 lock-up faults. However, the settling time and overshoot limits are exceeded with both 3 and 4 lock-up faults. This is attributable to the ratio between the number of fault mode controllers and actual fault modes within the system. As the perceptions only cover the total number of faults, and not their location, the number of real system fault modes that need to be accommodated by each controller is large: there are 120 unique, effective² fault location combinations for 3 faults, and 210 for 4 lock-ups. In addition, the variation in behaviour between high proportions of lock up faults in the grounded half of the assembly and the load-side half is significant. This makes control design to satisfy all of these modes difficult. The controllers in this case

²There are less effective fault locations from a system representation and performance stand-point than actual fault locations i.e. a fault in a parallel set of redundant actuators will produce the same behaviour, regardless of the actual location within the parallel arrangement.

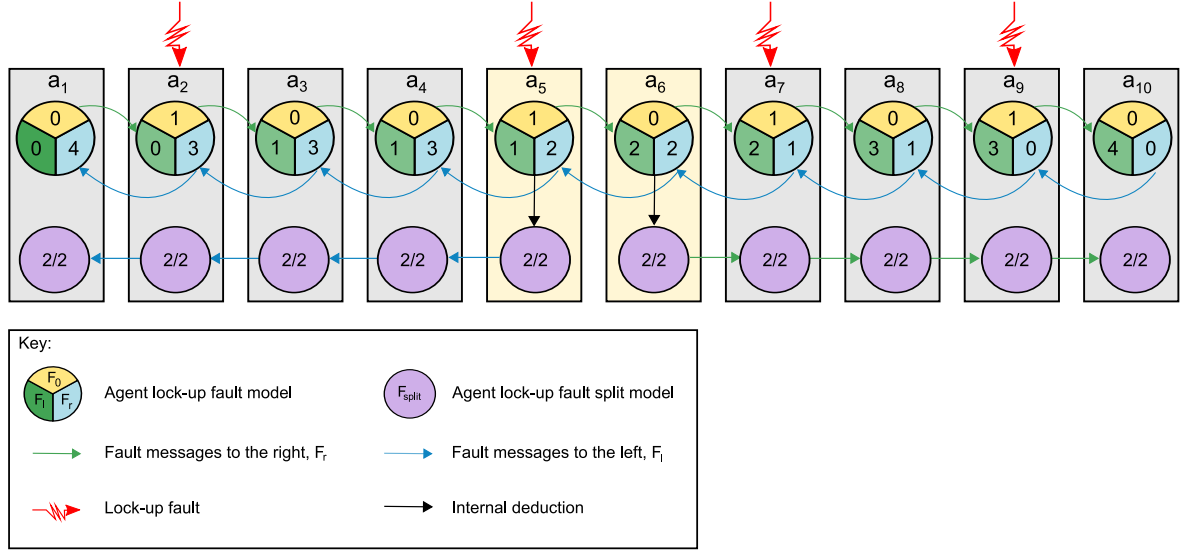


Figure 7.23: Communication of location specific fault information.

only satisfy 90 of the possible 120 fault scenarios for the 3 lock-up fault case, and 160 of the possible 210 for the 4 fault case.

This level of tolerance is still a large improvement on that of the passive case. However, it is possible to improve this performance by including more perception states.

Two more perceptions are added if agents communicate at which end of the HRA the faults have occurred (in the half nearest the grounded connection, or the half nearest the load). Changes to the agency's communication set-up are required in order to effect these location specific perceptions. However, a relatively simple approach is available as the information regarding how many faults are in each side of the HRA already exists in the agent system. f_{lock} and f_{rlock} in a_5 and a_6 respectively give the number of lock-ups in the left half and right half side, once they are added to their personal fault state. If these values are attached to the messages travelling left in communications on the left hand side, and to the right in communications the right hand side, then the location-specific fault information is distributed. This is further illustrated by Figure 7.23, which gives an example where the system has three lock-up faults. In addition to this added message, a further complication is added to the system, as it is necessary for each agent to know their location in the assembly, and agents 5 and 6 have to take a slightly different role from the other agents, as they pass on their perception of the fault locations.

These added perceptions improve the performance of the system as shown in Table 7.3. The mean and maximum settling times and overshoots are significantly reduced and the number of tolerable modes increased. This number could be increased further through more advanced control tuning, or the addition of further perception states, both at the cost of added complexity within the design process.

Table 7.3: Transient characteristics and no. of systems within the requirements under 3-4 lock up faults

System state	RT (s)		ST (s)		OS (%)		No. of systems within requirements
	mean	max	mean	max	mean	max	
1 Perception per lock-up fault							
3 lock	0.56	0.62	1.09	1.78	0.22	2.60	96/120
4 lock	0.60	0.66	1.15	2.24	0.56	4.26	160/210
2 Perceptions per lock-up fault							
3 lock	0.49	0.52	0.81	1.39	0.32	2.53	119/120
4 lock	0.60	0.65	1.02	1.82	0.34	2.71	207/210

7.4.1.3 Comparison with Passive Fault Tolerant Control Performance

The control results of the preceding two sub-sections are compared to the passive fault tolerant control results of Chapter 5 in Figures 7.24 and 7.25. Figure 7.24 compares the transient response characteristics of the passively controlled and actively controlled example systems and Figure 7.25 provides those under lock-up fault conditions. The settling times and rise times are expressed as a percentage change from the nominal value, whereas the overshoot is the actual value. The requirement threshold for each example system is also given on these figures, and is expressed as a percentage increase on the nominal value.

It can be seen from these figures that the active fault tolerant control approach offers substantial benefits in terms of performance. The percentage increase of rise times and settling times is greatly decreased in all three example systems under the influence of loose faults. The change in rise time is up to 12 times smaller with active fault tolerant control in the 4×4 PS example, and settling time is over 130 times smaller with 8 loose fault case. The overshoot increases induced by loose faults are also significantly reduced by the active FTC method, and remain within the requirement boundary.

Large improvements to rise and settling time changes are also witnessed in the lock-up fault scenarios. The overshoot, however, is increased by the active control changes, although this stays within the requirements for the 4×4 systems. The overshoot increases in the 10×10 system exceed the requirement in the 3 and 4 lock-up fault cases, however, as mentioned in Section 7.4.1.2, the performance can be improved by introducing more fault perceptions and control modes.

7.4.1.4 Static Performance Summary

The static fault simulations discussed in the preceding subsections have shown that the MAC approach taken allows a dramatic improvement in system performance. Figures 7.26 to 7.29 illustrate this further. For each system, all possible faults combinations have been simulated and their transient performance is summarised in these figures. They show that all the fault scenarios are accommodated within the performance requirements in the PS and SP 4×4 HRAs. In the 10×10 system, a number of lock-up fault scenarios fall outside the transient requirements. More faults can be accommodated by adding more perceptions, as stressed by

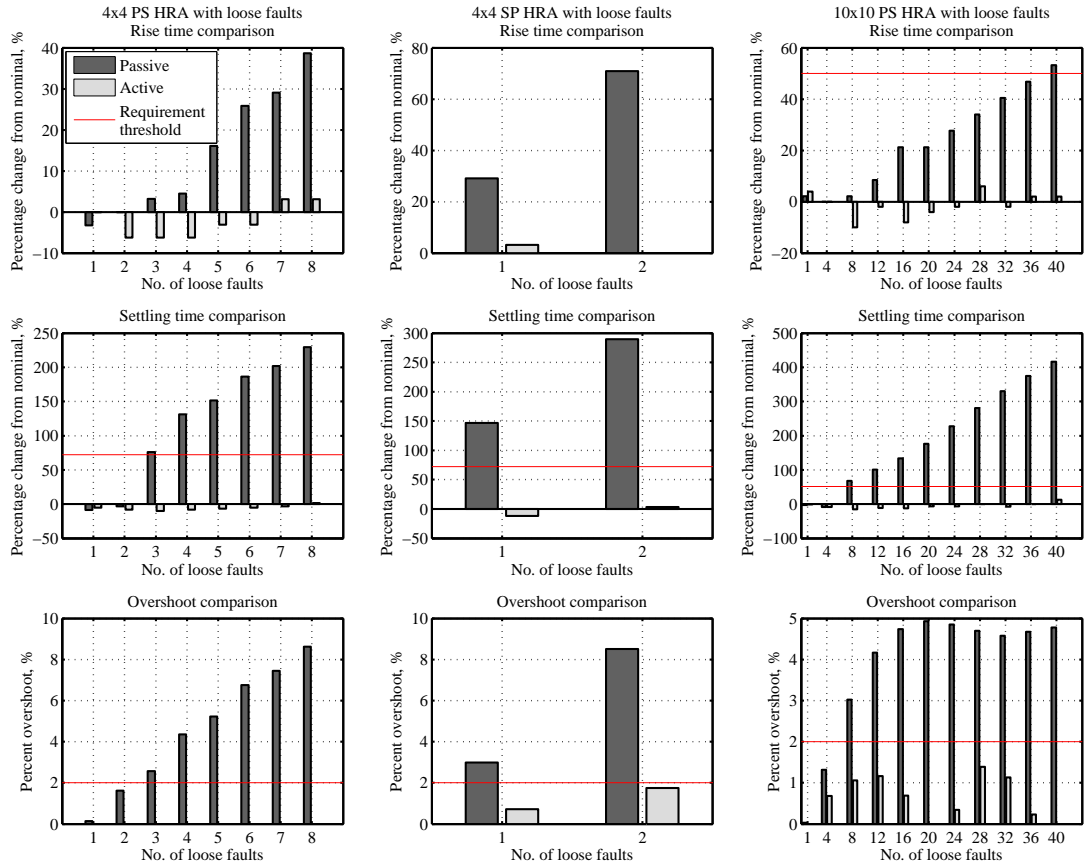


Figure 7.24: Comparison of active and passive fault tolerant control performance with loose faults.

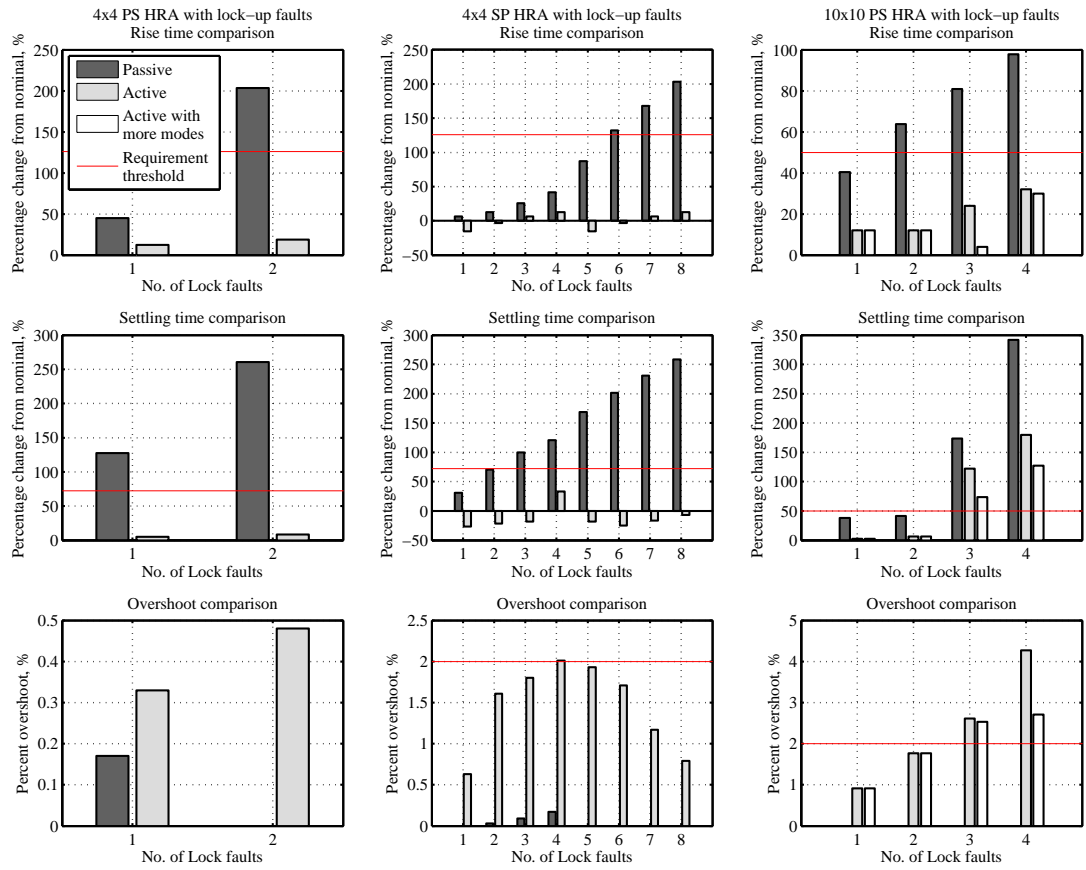


Figure 7.25: Comparison of active and passive fault tolerant control performance with lock-up faults.

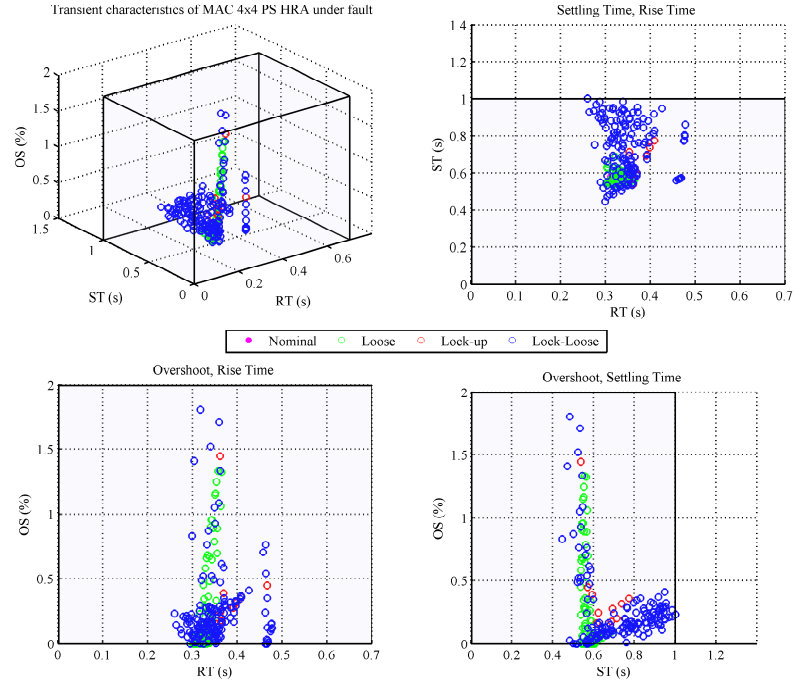


Figure 7.26: Transient characteristics of MAC controlled 4x4 PS HRA under all fault combinations

Figure 7.29.

Overall, the results are a considerable improvement on the passive control case. However, this performance is only achieved in the static fault state. Performance during fault detection and reconfiguration is considered in the next subsection.

7.4.2 Dynamic Fault Injection Simulations

Although performance during the static fault state has been shown to satisfactory, it is important to consider the affects of fault detection delays, communication and control switching within this multi-agent approach.

Analysis of the systems under fault, whilst controlled by their corresponding control law has shown that these systems are stable. However, the reconfiguration process in the MAC involves switching between control laws when faults are detected, and it is well known that unconstrained switching, even between asymptotically stable systems, can induce instability (165), although, the nature of this instability is limited as the signals cannot escape to infinity in the finite switching times.

Stability analysis of switched linear systems has been an active area of research in the last decades and, as such, there are several analytical ways to guarantee stability within an unconstrained switched system. If a common Lyapunov function exists for all the possible switched systems, then the system is asymptotically stable for any switching signal. A common Lyapunov function may be found in a number of ways, however, this is a non-trivial task for a

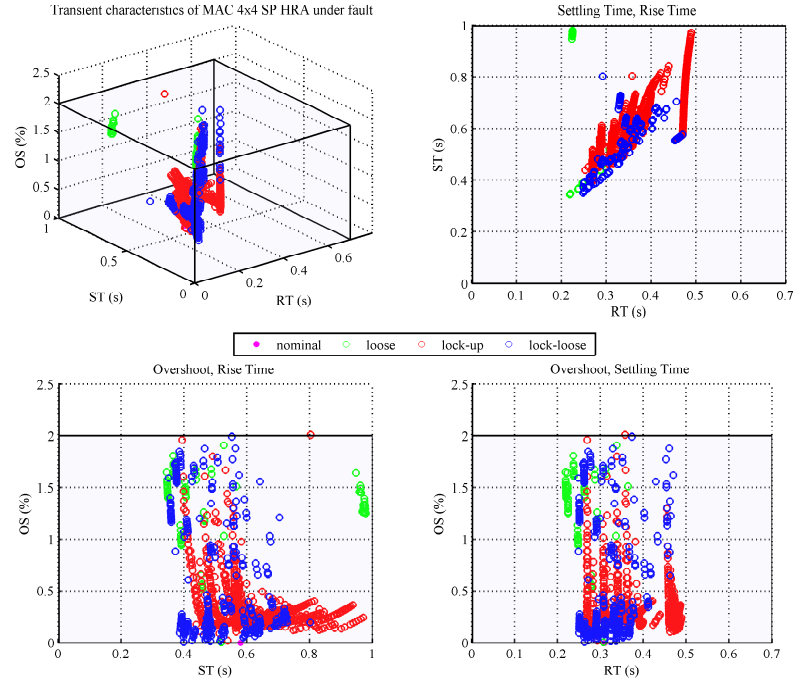


Figure 7.27: Transient characteristics of MAC controlled 4x4 SP HRA under all fault combinations

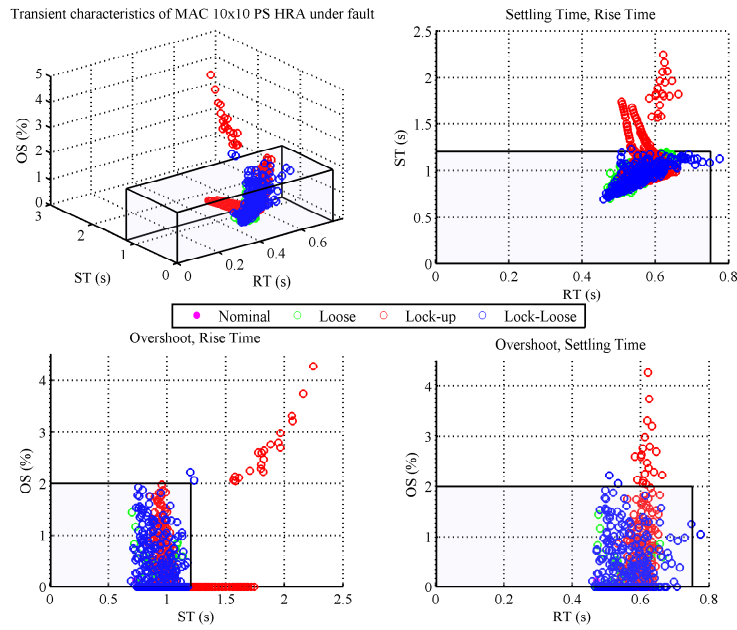


Figure 7.28: Transient characteristics of MAC controlled 4x4 SP HRA under all fault combinations

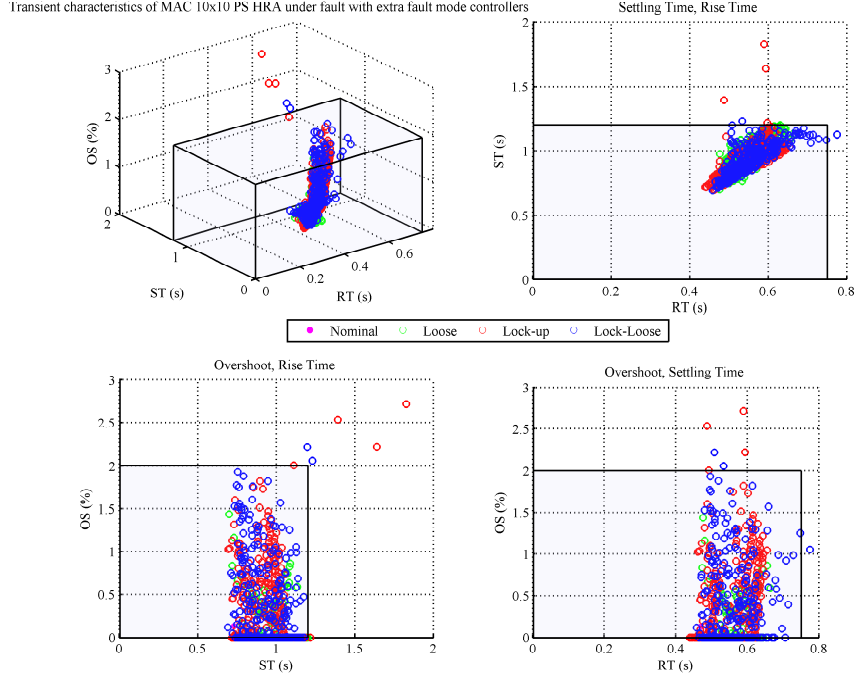


Figure 7.29: Transient characteristics of MAC controlled 4x4 SP HRA under all fault combinations

large family of switched systems. Alternatively, if the admissible switching path is restricted then stability may be ensured through the use of multiple Lyapunov functions (166). Another, simpler way to guarantee stability within a switched system is to constrain the switching. This is usually done by enforcing a dwell time (167). Finding the minimum dwell time, however, is also demanding for large sets of switched controllers.

These issues are not imperative for the MAC presented here, as the switching in this case is already severely constrained. The fault detection within each agent effectively determines the switching characteristic of the system. The detection algorithms used are state-based i.e. the subsystem is either nominal or faulty. Once the fault detection has entered the faulty state, then it cannot return to the nominal state, unless the fault detection is reset. This constrains the switching within the system considerably. A conservative upper-limit on the potential number of switches exists, which is determined by the maximum number of faults that can occur.

In addition, once a fault has been detected, and the control has been reconfigured, the control law will remain constant until the fault state changes. This period in a real application is likely to be very long, particularly in comparison to the speed of the system. Hence, the effective average dwell time of the switching signal is very large, and would be far in excess of any value derived analytically from (168). Hence, a formal analysis of the stability under switching is not addressed here.

Whilst switching is not considered problematic with respect to stability, the switching characteristic does have an effect on the output of the system during the reconfiguration.

These effects are witnessed and discussed further in the dynamic fault injection simulations that follow.

7.4.2.1 Single Fault Injection

The occurrence of a single fault at any given time instance is the most likely scenario in the HRA. Dynamic results for the MAC controlled PS 4×4 HRA with 1 lock-up are shown in Figure 7.30. A square-wave is used as the system load reference and a lock-up fault is injected in branch 4 at $t=0$. The input reference and the load position for the system under nominal conditions, and fault conditions with passive and MAC control, are given in the first sub-plot. The second plot provides the HRA voltage input, and shows that the input remains well within the input limits. The third sub-plot shows the summed fault detection residuals from the agents. It can be seen that the fault is detected at 0.25 seconds. The final plot gives the internal control mode of each agent. All agents have reconfigured their control to mode 1 (which represents the mode designed for 1 lock-up fault) by 0.3 seconds.

The response is the same as the passive case before the fault is detected, after which the gain is increased and the response quickens. By the latter half of the first square-wave period the effects of the reconfiguration transients and fault state of the system have settled and the response is very close to that of the nominal system. The effects of control switching and detection delays are minimal in this single fault case and these results are typical of the three example HRAs with various single faults types and locations.

7.4.2.2 Multiple Fault Injection

Whilst single faults occurring at an instant are most probable, it is pertinent to consider worst-case scenarios where many faults occur at once. Multiple faults increase detection and communication times as well as control switching in the system. Hence, scenarios where the system transitions from the nominal state to a maximum fault level in a single instance are considered in this subsection.

Figure 7.31 gives the response of the 4×4 PS HRA where 2 lock-up faults and 4 loose faults are injected into the system at $t=0$. (representing the maximum permissible faults in this system). The loose faults are detected at 0.125 seconds and the lock-up faults at 0.4 seconds. A large transient in load position follows the reconfiguration after detection of the lock-up faults and a significant amount of overshoot is incurred ($\approx 11\%$). This large transient, however, is not due to the fast switching between several control modes incurred after the detection (8 switches occur in the 0.4-0.45 second time frame, Figure 7.32 shows this more clearly). Rather, it is attributable to the large change in gain at this point.

This is further demonstrated by Figure 7.33, which shows the response of the system where the number of switches are falsely reduced, so that all control is switched at the instance of lock-up fault detection. The transient is still induced regardless of the reduced switching.

The MAC architecture offers a solution to these gain-change induced transients, however. The control was reconfigured as quickly as the fault detection and agent communication would permit in the simulation of Figure 7.31. As the reconfiguration is distributed, it is

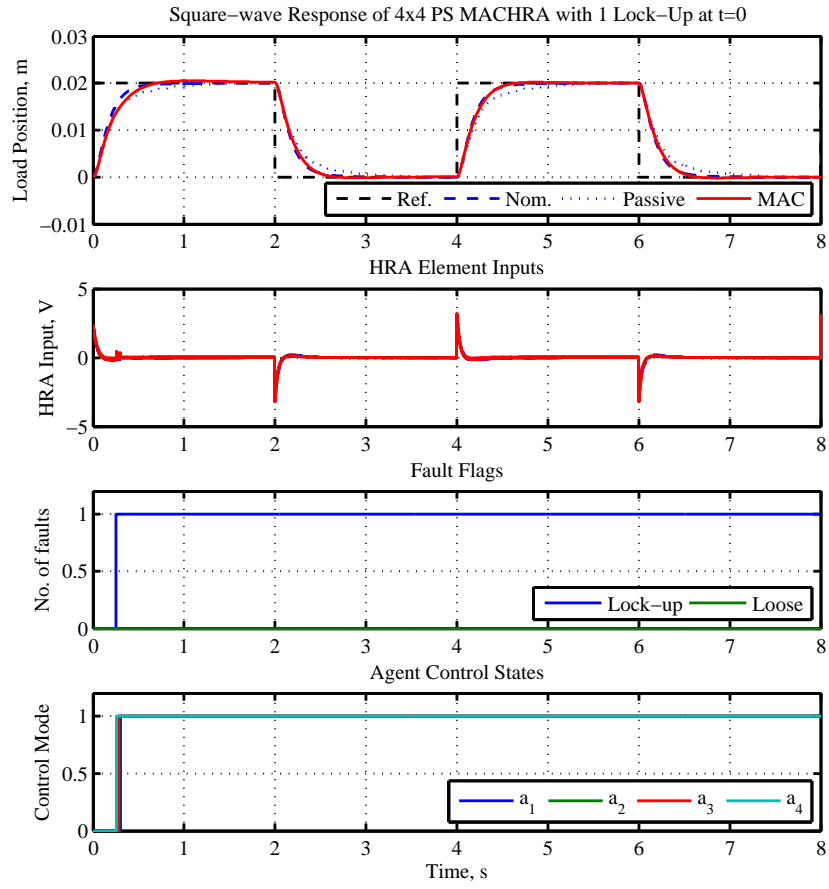


Figure 7.30: Dynamic response of 4×4 PS MACHRA with 1 lock-up fault at t=0.

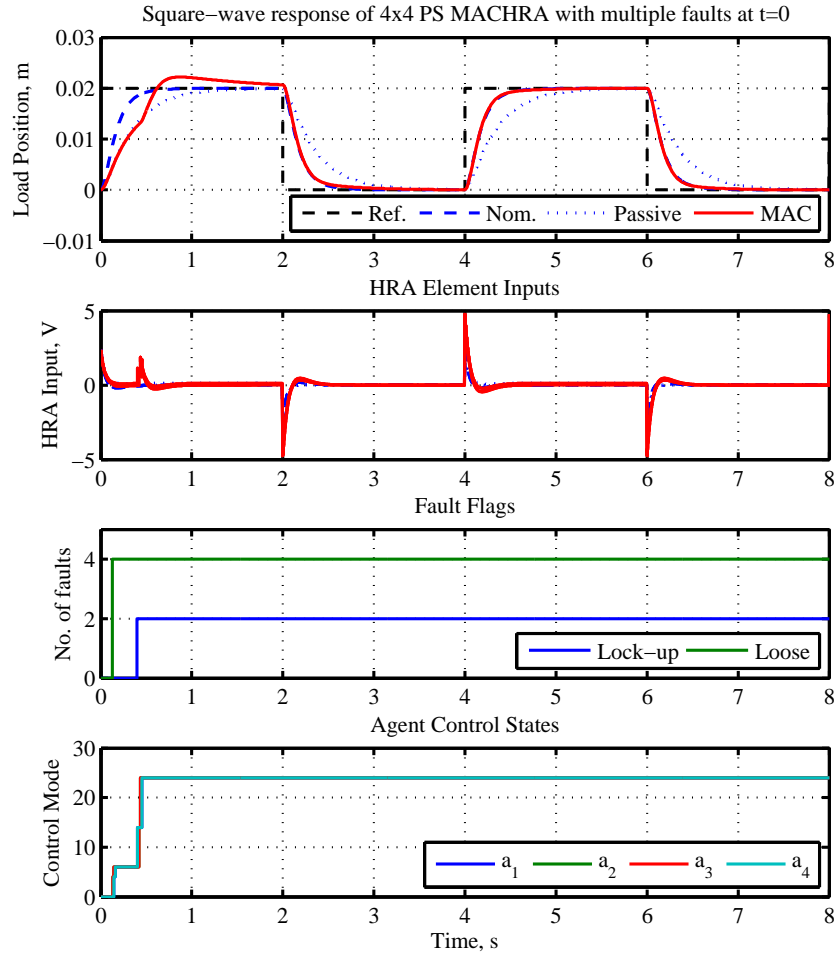


Figure 7.31: Dynamic response of 4×4 PS MACHRA with multiple faults at t=0.

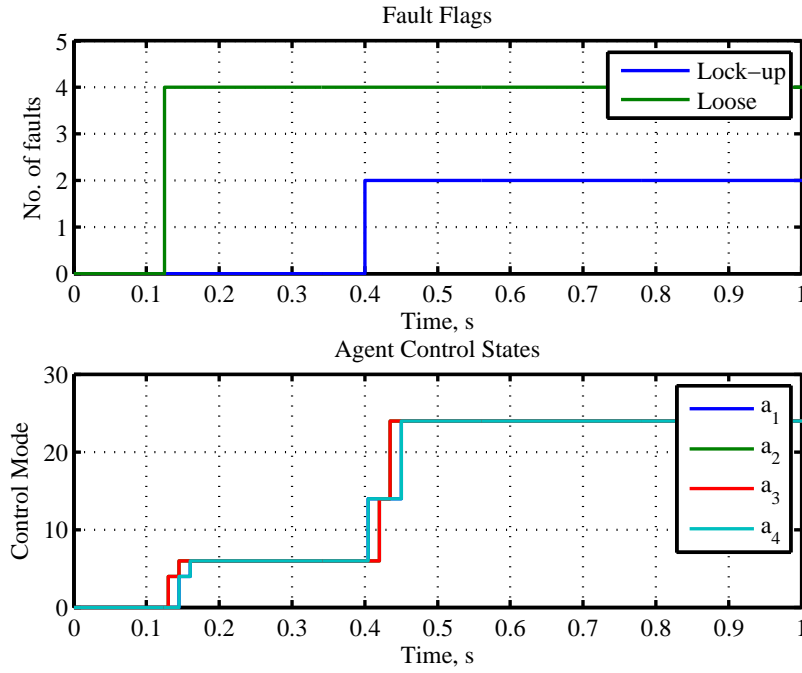


Figure 7.32: Fault flags and control modes of agents in 4×4 PS HRA in response to multiple faults.

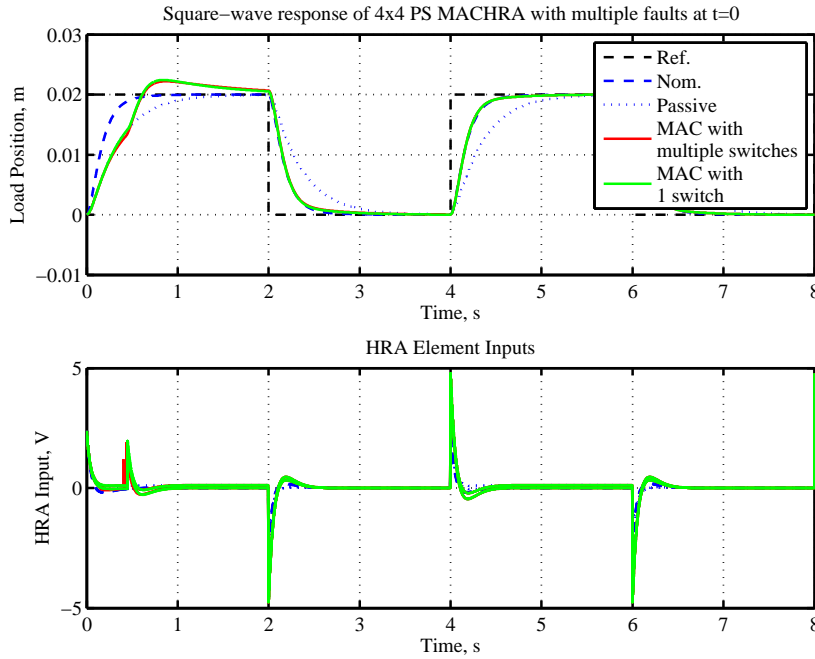


Figure 7.33: Response of 4×4 PS HRA with multiple faults and multiple or single control switches.

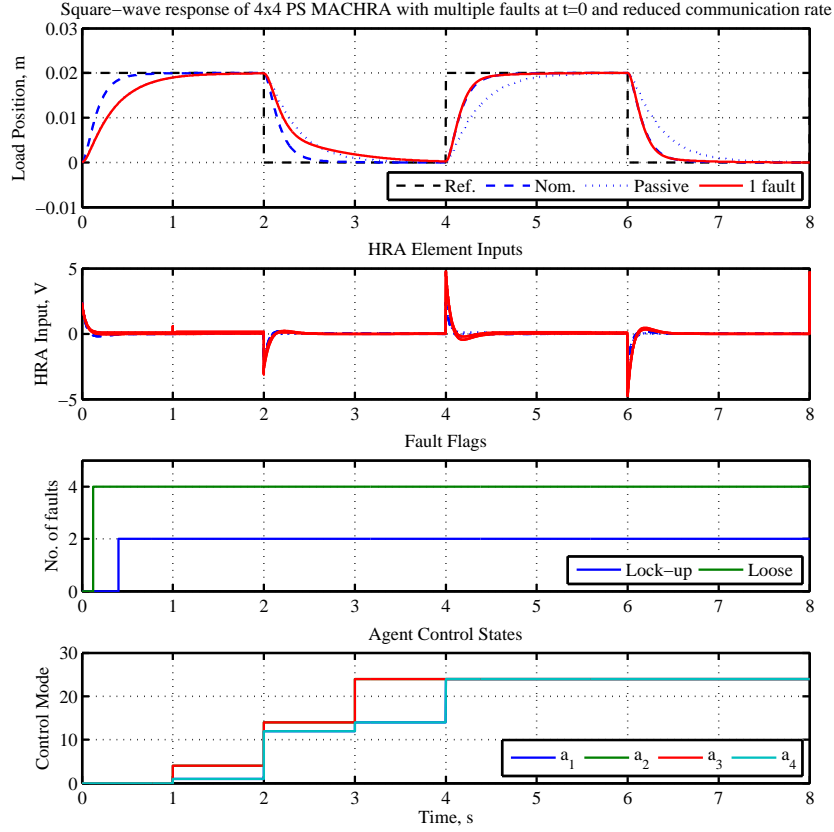


Figure 7.34: Response of 4×4 PS HRA with multiple faults and a reduced reconfiguration rate.

possible to reconfigure the system more gradually if this process is delayed. This can be easily achieved by reducing the speed of communication, which also has obvious benefits for any implementation of this scheme, as demands on the communication bandwidth will be reduced. If the communication frequency is decreased, then the results in Figure 7.34 for the same fault profile can be achieved. The transient has now been removed from the system. However, the time taken for the system to achieve total reconfiguration and the output to converge upon the nominal performance is increased. Hence, it is apparent that a trade-off exists here between limiting transient reconfiguration effects and reducing the time that elapses between fault occurrence and total reconfiguration. The specific application will dictate which aspect of performance is most important, and consequently dictate the reconfiguration rate.

7.5 Conclusions

This chapter has presented an active fault tolerant control method for high redundancy actuation. Multi-agent concepts have been used to provide a structured approach to active FTC design that deals with the complexity of HRA through the use of simple localised reconfigurable control and fault detection.

An outline of the MAC scheme has been provided and simulation results of its application to the three HRA example systems have been given. It was shown that MAC of HRA can provide significant benefits in comparison to passive fault tolerant control, under the full range of fault levels. Changes in transient characteristics are decreased substantially, whilst stability margins are preserved, and near nominal performance can be maintained in the majority of fault scenarios.

Control switching and reconfiguration delay effects were considered, and generally their influence on system stability and performance was found to be minimal. In extreme fault cases, where many faults occur in one instance, changes in the control gain can cause large transients, which may be reduced by slowing the reconfiguration process.

Chapter 8

Software Demonstrator

8.1 Introduction

The HRA is a new concept in fault tolerant actuation, and one whose configuration and operation can be difficult to visualise for those outside of the project. The development of hardware demonstrators aids this issue somewhat, and proves that these systems are controllable. A 2×2 SP HRA utilising electromechanical actuators (Figure 8.1) was developed in an earlier stage of the project, and a 4×4 PS electromagnetic HRA is also in the final stages of development.

However, the current number of elements within these hardware demonstrators is far lower than that envisaged for a true HRA being in the order of 1-10 rather than 100+. The physical arrangement of these demonstrators are also limited to one configuration, and it has been shown that this has a significant affect on the reliability and characteristics of the HRA. Hence, a software demonstrator of the HRA has also been developed in order to aid visualisation and performance assessment of larger HRAs and multiple configurations. The key aims of this demonstration are:

- to illustrate how large numbers of small elements may form a single actuator.
- to simulate a range of HRA sizes and configurations.
- to demonstrate key concepts such as system capability.

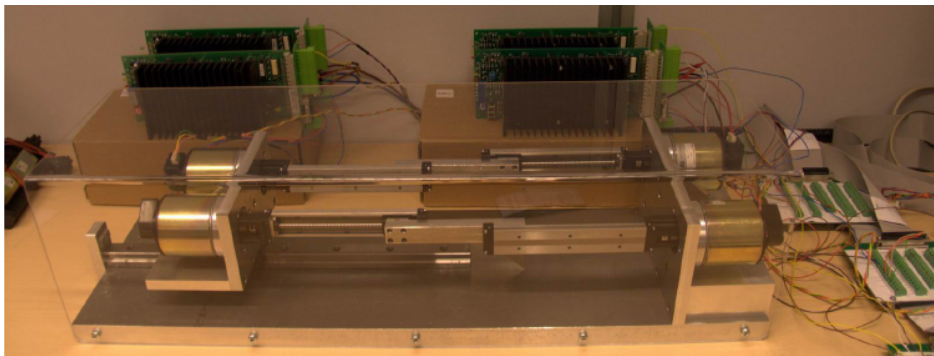


Figure 8.1: 2×2 SP HRA electromechanical experiment.

- to illustrate the effects of the main fault types: lock-up and loose faults.
- to display the performance of a variety of control laws.

This chapter details the development of this HRA software demonstrator.

Firstly, some requirements for the demonstrator are established. The implementation and functionality of the resultant software demonstrator is then described and finally, some conclusions are made

8.2 Requirements

Having stated the main aims within the introduction, the required features for meeting those objectives are discussed within this section. The requirements of the demonstrator may be generally divided into a number of categories:

- Functional requirements,
- Visualisation requirements,
- Usability, portability and extendability requirements.

A full list of requirements is given in Table 8.1, and each category is discussed briefly in the following sub-sections

8.2.1 Functional Requirements

The software demonstrator must offer the simulation of a selection of configurations and sizes to meet the principal objectives. Each of these simulated systems must be based upon the real dynamics of a HRA in order to provide validity to the results it illustrates.

The simulation must operate in a real-time manner, so that the user can see the operation of the HRA during simulation, and change simulation parameters, such as the fault state, mid-simulation. The user should also have a choice of control options and limited control of system inputs and noise parameters in order to see their affect within the system.

8.2.2 Visualisation Requirements

The visual elements of the demonstrator should show the structure of the system clearly, so that parties unfamiliar with control can understand the operation of the system. Also, a representation of the HRA's structure should be given, and this should be dynamic during simulation. A real-time indication of faults within elements should also be given in this visualisation of the HRA.

In order to demonstrate the concepts of capability levels and system health, a visualisation that demonstrates these features is required. Again, this should operate real-time within the simulation.

A final visualisation requirement is the plotting of system references, inputs and measurands to inform the user of the system's operation in a more precise manner.

Table 8.1: Software Demonstrator Requirements

1. Functional Requirements	
1.1	Real-time simulation of high redundancy actuators using real models of the system.
1.2	Simulation of several configurations and HRA sizes.
1.3	Choice of a number of pre-defined control laws.
1.4	Real-time fault injection of lock-up and loose faults.
1.5	User control of system inputs, noise and disturbances.
2. Visualisation Requirements	
2.1	Clear visualisation of the system level structure.
2.2	Element level, real-time animation of the physical structure and dynamics of the HRA.
2.3	Indication of faults within the real-time physical representation of the system.
2.4	Dynamic real-time display of system fault state and health.
2.5	Real-time plotting of references, inputs, states and measurands.
3. Useability, Portability and Extendability Requirements	
3.1	Useable by parties who are relatively unfamiliar with the simulation environment.
3.2	Encapsulation of coding and simulation elements to avoid unintentional editing.
3.3	Operational in Matlab/Simulink environment.
3.4	Operational using standard amounts of processing capability.
3.5	Portable to other machines.
3.6	Control law extendability by HRA researchers.

8.2.3 Usability, Portability and Extendability Requirements

The software demonstrator must be relatively useable for those unassociated with the project, and be intuitive to operate. Encapsulation of subsystems and functions should help achieve this goal, and help prevent unintentional editing. However, the underlying simulation should not be so inaccessible to prevent future extensions.

Matlab/Simulink is chosen as the operating environment for the demonstrator, for reasons discussed within the following section. Hence, the demonstrator should operate in the current version of Matlab/Simulink and should be operational using a system that meets the standard Matlab/Simulink requirements (which are currently 680MB of disk space, 1024 MB RAM).

8.3 Implementation

The software demonstrator is implemented within the Matlab/Simulink environment as this package accommodates a number of the requirements. Firstly, Simulink has been used within the modelling and control studies of the HRA under nominal and fault conditions, and simulation libraries for this purpose were developed. Thus producing real-time simulations for HRAs for demonstration purposes is relatively straightforward within this environment. This may be incorporated with Matlab's GUI development environment and subsystem masking capabilities within Simulink to increase the clarity and usability of the simulations. Matlab/Simulink also has extensive facilities for graphical visualisation and plotting. This integration of the mathematical representation of the system, the user interface and visualisation elements makes Matlab/Simulink an attractive choice. In addition, its wide use throughout

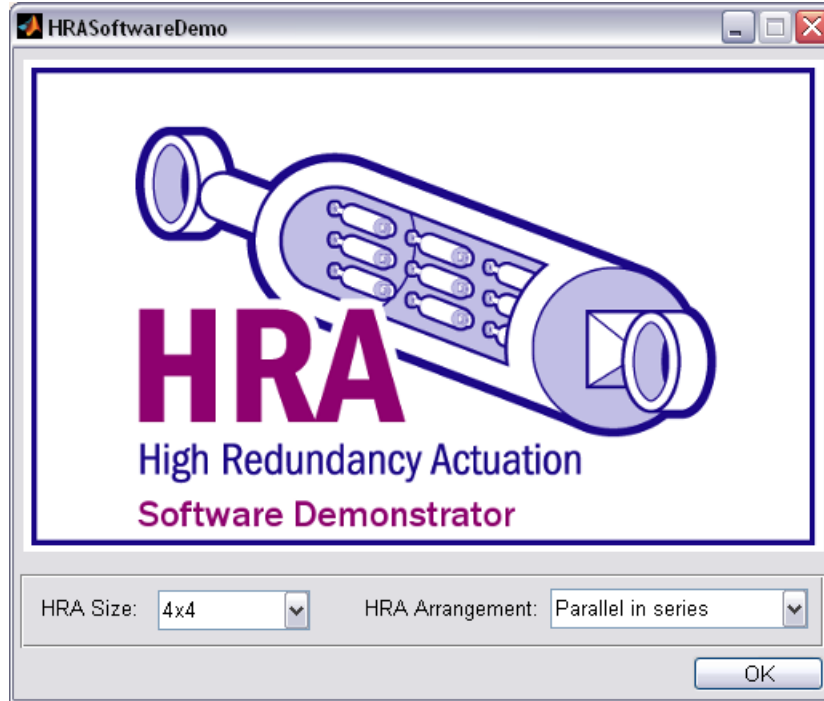


Figure 8.2: Software Demonstrator navigation window.

academia and industry, its platform independence and open functions makes Matlab/Simulink suitable for use in this case.

The software demonstrator's operation is discussed in the following sub-sections.

8.3.1 Navigation Window

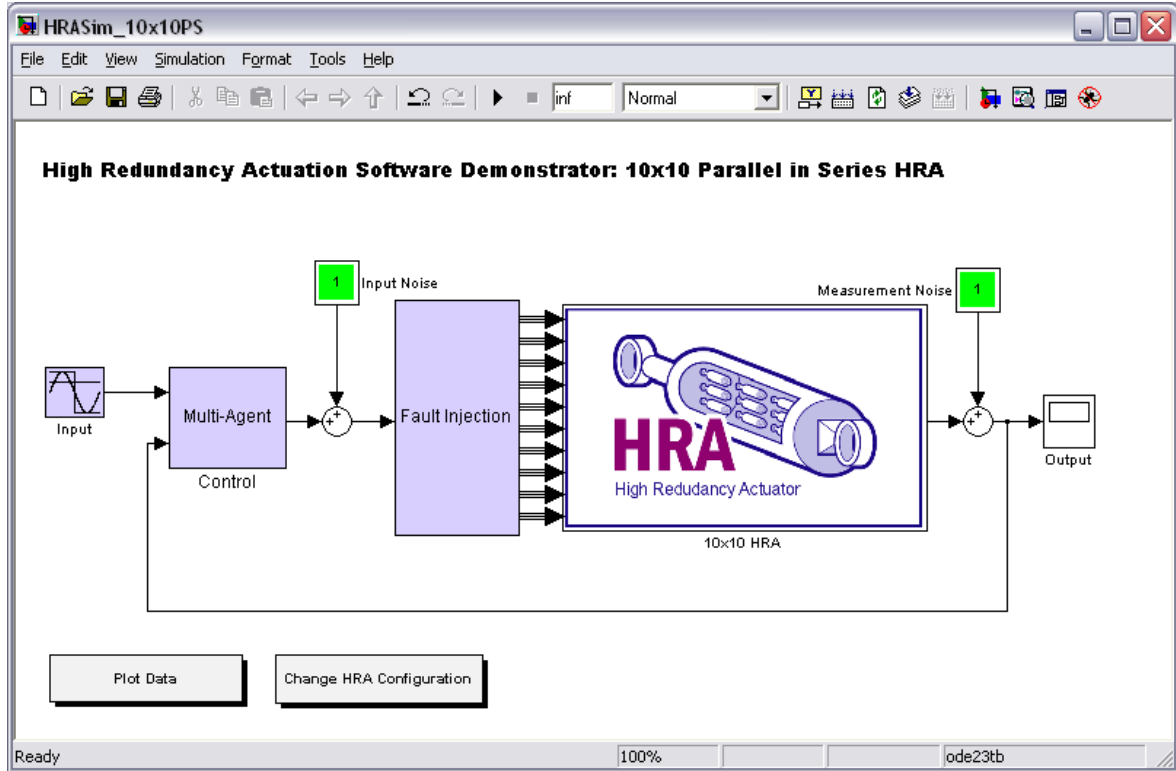
On starting the demonstrator, by running the `HRA Software Demo` file, the user is presented with the navigation window shown in Figure 8.2. The user may choose the dimensions of the system as either 4×4 or 10×10 , and the configuration of the system as parallel in series or series in parallel from the drop-downs. On pressing 'OK' the corresponding simulation is initialised. This window was created using Matlab's GUIDE tool.

8.3.2 Simulation Window

Figure 8.3 gives the simulation window that is called on selecting a 10×10 parallel in series system from the navigator. This is a Simulink model of the system and the demonstrator is run and configured from this screen. The user is provided with a representation of the simulated process from system input to output.

8.3.2.1 Configuration

The blocks within this system are interactive, and allow the user to configure the simulation. Each block is discussed below:

Figure 8.3: Main simulation window for 10×10 PS system.

Input block The input of system may be configured by clicking on the input block, activating the pop-up shown in Figure 8.4. The input type drop-down offers a choice of a square wave, sine wave, or constant input. The edit boxes below then allow the user to modify the amplitude and frequency (for square wave and sine wave inputs). This pop-up is created by masking a sub-system which switches between input types, and controlling the switching signal through the mask parameters. The input may be edited before or during simulation.

Control block The control block gives the user a selection of control algorithms to simulate (Figure 8.5). The control may be set to open loop, classical control, or multi-agent control. The algorithms used are those described in Chapters 5 and 7. On selecting a control option, the control type is displayed on the block, and the correct controller is switched to in the subsystem which the block masks. The control law may only be configured before simulation, as changes in control mid-simulation are not recommended. The masking of this subsystem discourages the user from editing the underlying control law, as the simulation is for demonstration, not control development purposes. However, the control options are easily extendable if further control laws are designed.

Input and sensor noise blocks The input and measurement noise blocks control the noise within the system. On double-clicking either of the blocks, a pop-up similar to that shown in Figure 8.6, which allows the noise to be toggled. The block displays a '1' and a green background if the noise is switched on, and conversely a '0' and a red background if the noise

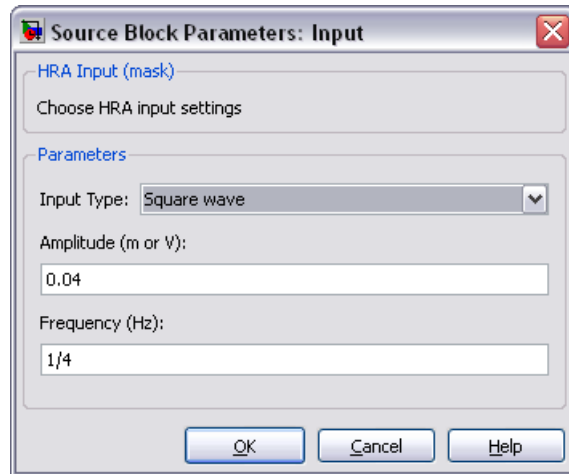


Figure 8.4: Input selection pop-up.

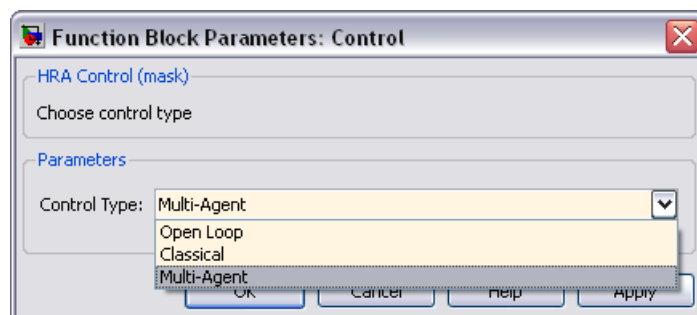


Figure 8.5: Control selection pop-up.

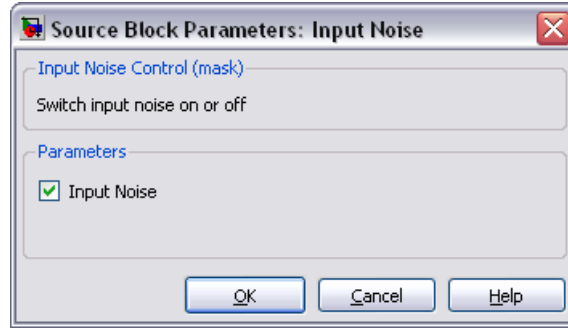


Figure 8.6: Input noise pop-up.

is off, allowing the user to know the state of the block without checking it. These blocks may be edited pre and mid-simulation.

Fault injection block The final configuration element is the fault injection block. On double-clicking this block the user is presented with the fault injection control panel shown in Figure 8.7. This figure shows the control panel for a 10×10 PS system, however, other configurations will have a slightly different layout. The fault state of the 10 parallel branches may be controlled from this panel. The healthy state of a parallel branch is denoted by a fully green block with the number 10 displayed (representing the number of healthy parallel elements). On clicking on one of the blocks, a loose fault is injected, which is signified by a decrease of 1 in the number displayed and a drop in the green area. On reaching '0', the next click will inject a lock-up fault, which is denoted by a red cross and a '-1'. Hence, the particular display in Figure 8.7 shows that the system has two loose faults in its first parallel branch, 3 in its second, a lock-up in branch 5 etc. Faults may be injected real-time within the system, however, if lock-ups are injected it is recommended that the simulation is paused whilst the fault status is changed by the user to prevent the system stepping through fault types whilst the simulation is running.

8.3.2.2 Simulation Visualisation

On running the simulation (by pressing the play button situated within the task bar of the main simulation window) a number of visualisation windows are activated. A typical screenshot for the demonstrator during operation is shown in Figure 8.8. Each of the simulation visualisation windows are discussed here.

HRA Animation An animation figure is activated which illustrates the HRA's operation during simulation. Figure 8.9 provides a still from the 10×10 PS simulation. Each element is represented as a simplified actuator within this figure, and the inter-element masses are shown in grey. The load mass is the final grey mass at the right hand side. On providing an input to the simulation, the actuation elements move transversely and injected faults are represented by colour changes of the actuator, where red signifies a lock-up and yellow a loose fault.

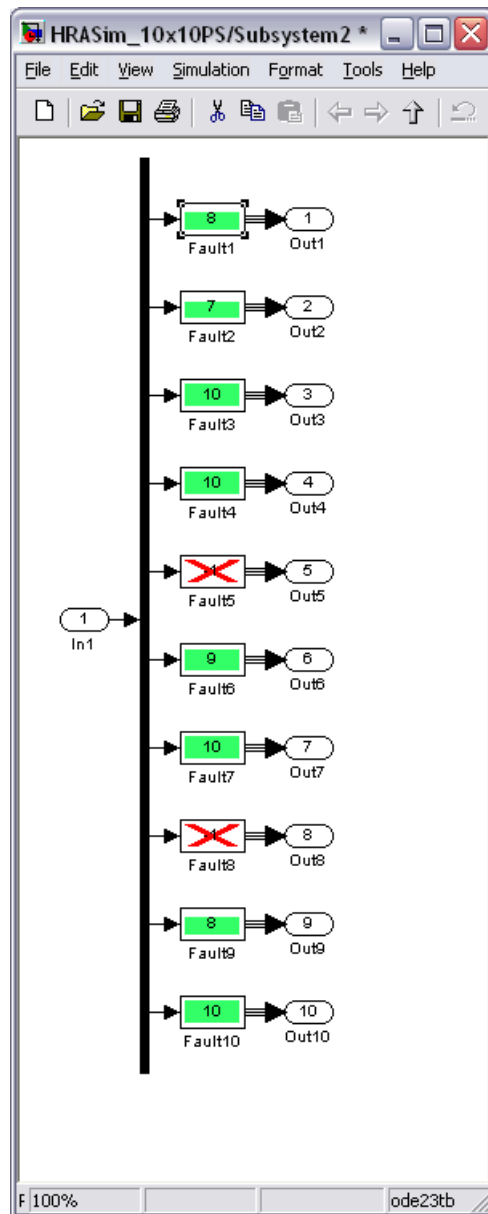


Figure 8.7: Fault injection controls.

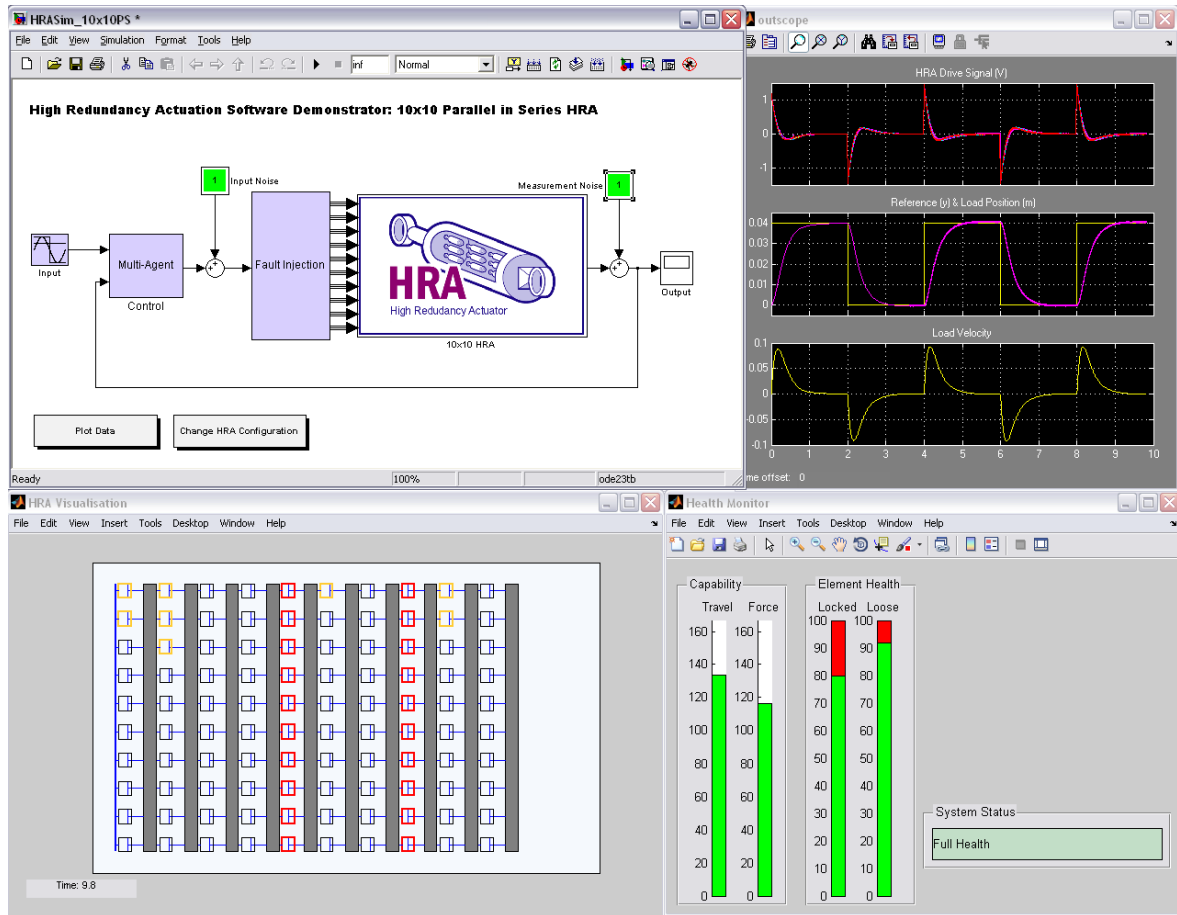


Figure 8.8: Typical simulation screen.

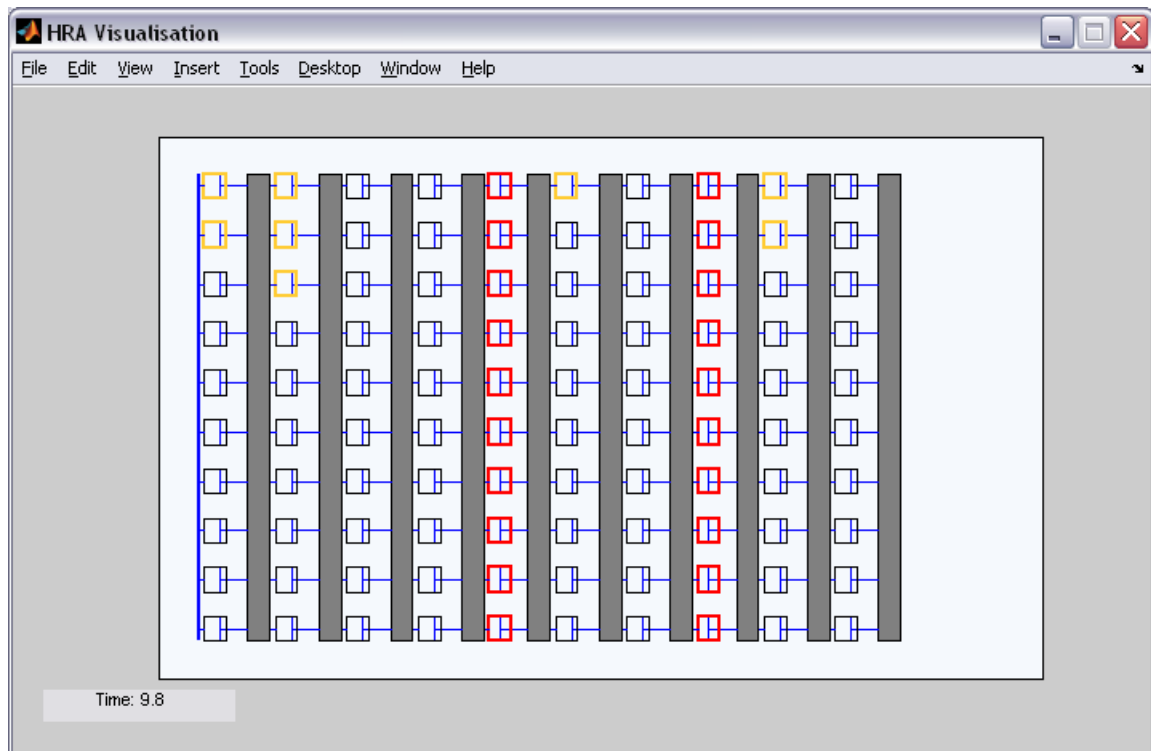


Figure 8.9: HRA animated display.

Animated figures can be created in Matlab by drawing a figure with handle commands, and updating the position of those handles every time step and re-drawing the figure. In order for this animation to run real-time, however, and allow for real-time fault injection, the animation m-code is implemented within an s-function in the simulation. S-functions are dynamically linked subroutines that the Matlab interpreter can automatically load and execute. The relative positions of the elements and their fault status are passed to the s-function, which initialises the figure on simulation start up, and updates it at every time step. A general flow chart of this process (which is applicable to all the s-functions described in this chapter) is shown in Figure 8.10 and the s-function for the 4×4 PS HRA animation (as this is the simplest example, having the fewest moving handles) is included in Appendix E.

Health Monitoring One of the aims of the software demonstrator was to demonstrate key concepts such as system capability. The system's capability and fault state is dynamically displayed within the health monitoring panel. Figure 8.11 provides a screen-shot of the health monitoring gauge for the 10×10 PS HRA. This window offers one potential way in which fault information could be displayed and used if fault detection and health monitoring algorithms are used. The element health gauges illustrate the number of loose and locked elements within the system, and the capability gauges provide the resultant travel and force capability as a percentage of the respective required capability. This dual gauge system allows the user to further appreciate the affect of faults on the system, and understand the influence of fault location and configuration on capability reductions. A system status box also offers advice on the system's health. It has three states:

- 'Full Health' - The system has travel and force capability in excess of 100% i.e. it has more capability than required. This state may be in effect in the presence of faults due to the compartmentalised redundancy within the system.
- 'Critical Health - Schedule Maintenance' - This state is displayed when the system is at 100% capability for either force or travel. In this situation, one more fault in the correct location will result in a capability below that required. Hence, whilst the system is fully operational, maintenance should be planned.
- 'Restricted Capability - Maintenance ASAP' - If the travel or force capability falls below 100%, than this message is displayed. The system now has a capability below that required, and the actuator should be replaced as soon as possible. However, the system will still have some capability, as displayed within the capability gauge, and operates in a graceful degradation region.

This health display is achieved through a combination of Matlab's GUI and animation functions executed within a s-function. Fault information is sent from the simulation to the s-function, which initialises the window using GUI functions, and updates the gauges at each time step. The fault information is derived from the user input, not fault detection algorithms. This is due to the limited scope of the health monitoring studies within the work of this thesis. The aims of the software demonstrator do not include an illustration of health monitoring

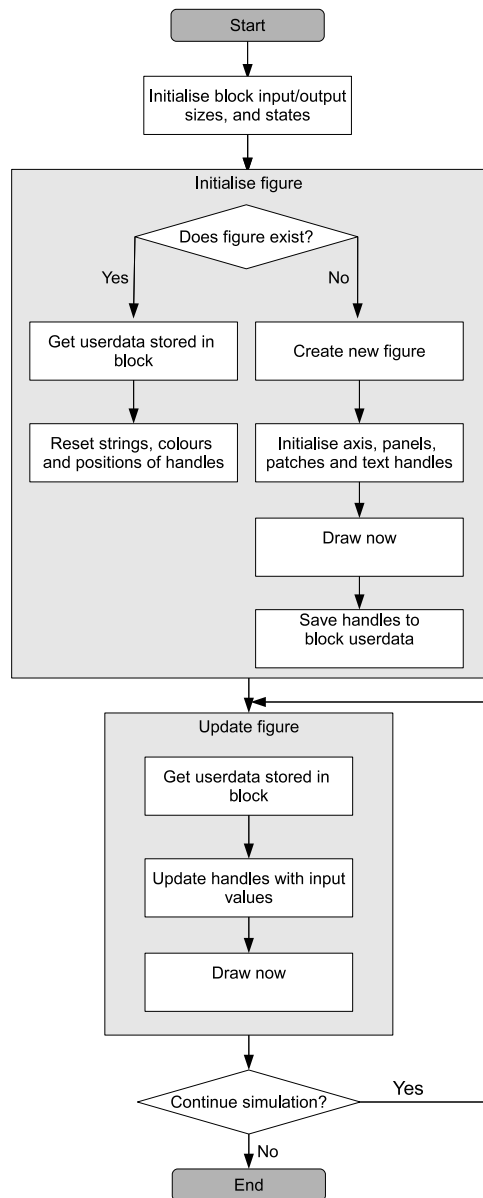


Figure 8.10: Flow chart describing operation of s-functions within the software demonstrator that provide real-time visualisations.

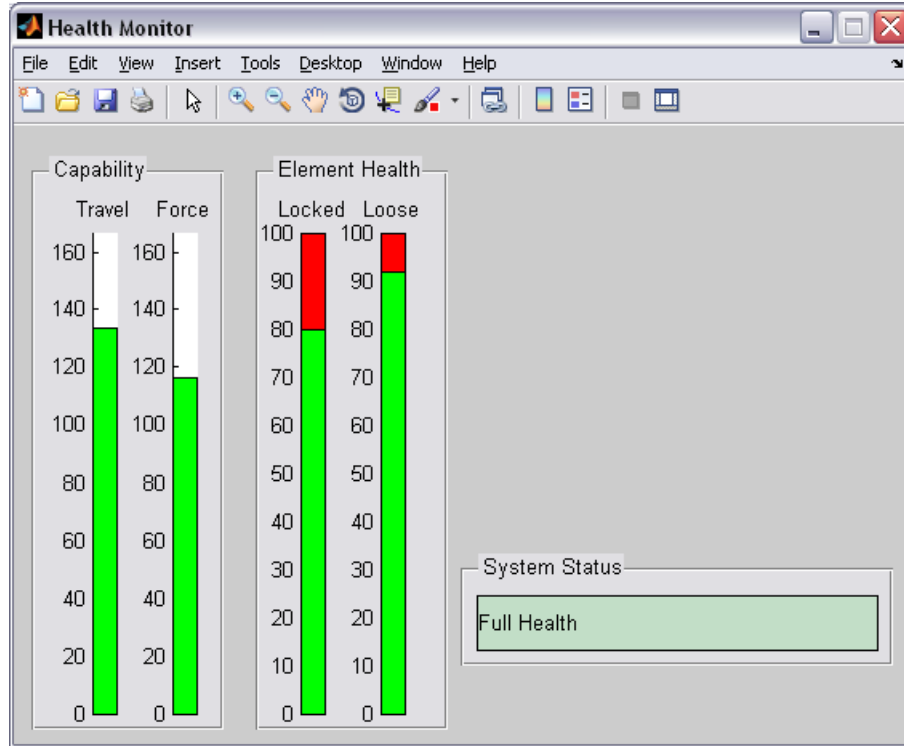


Figure 8.11: Health monitor window.

techniques, as this particular area is not a priority at this project phase. Nevertheless, the rule-based algorithms are executed in the software demonstrator for use within the multi-agent control. However, their output is not used within the health monitoring gauges as it is not designed for all the control structure options. The IMM studies are also not implemented within this demonstrator for similar reasons. If further methods of fault detection/health monitoring are studied, however, they may be incorporated within this display with relative ease.

Output scope Plots of key signals within the simulation is also provided during run-time through the output scope (Figure 8.12). The system input, output and velocity are displayed within this scope. These plots allow the user to appreciate performance changes in fault conditions during run time. A plot output option is also offered at the bottom of the screen, which runs an m-file that plots the same signals over the entire period of the last simulation, so that the system performance may be observed more clearly and saved if necessary.

Agent State Display The final window activated during run-time is the agent state display, a screen-shot of which for the 10×10 PS system is shown in Figure 8.13. This screen displays the state of each agent in a manner similar to that shown in Figure 7.4. Each agent's local fault state is displayed, as well as their personal model of the larger system state. This display allows the user to see when agents have detected faults, and how the knowledge of the fault spreads through the system. The speed of communication may be controlled by changing the communication times in the mask dialogue of the MAC block underneath the control block,

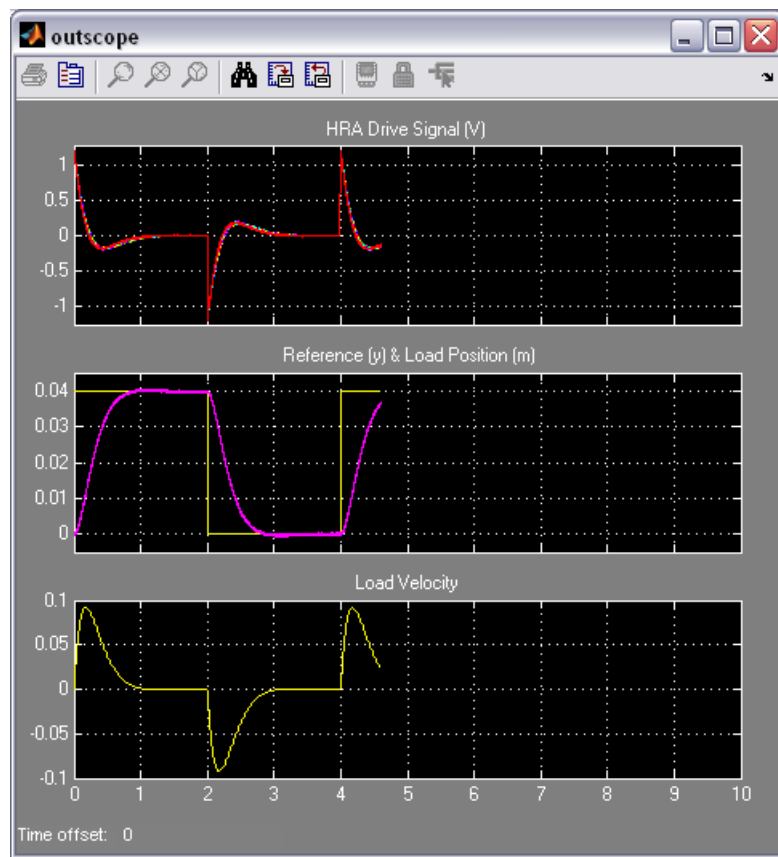


Figure 8.12: Simulation traces.



Figure 8.13: Agent state display window.

although it is set at a default time which allows the user to view state changes.

Again, this display is achieved through the use of a s-function which contains plotting commands that are updated every time step with the fault information provided to the function. This data is derived from the rule-based algorithms embedded in the Stateflow realisation of the multi-agent system within the simulation.

8.4 Conclusions

This chapter has detailed the development of a software demonstrator for the HRA concept. The main objectives of this demonstrator were to:

- illustrate how large numbers of small elements may form a single actuator.
- simulate a range of HRA sizes and configurations.
- demonstrate key concepts such as system capability.
- illustrate the effects of the main fault types: lock-up and loose faults.
- display the performance of a variety of control laws.

A number of requirements for the demonstrator were made in Section 8.2 in order to achieve these objectives.

The implementation of the software demonstrator was then described in Section 8.3. This demonstrator meets the requirements set-out in Section 8.2 as detailed in Table 8.2.

Table 8.2: Software demonstrator requirement status

	Status	Comments	Extendability?
1. Functional Requirements			
1.1	✓	Real-time simulation based upon system model dynamics.	N/A
1.2	✓	4×4 and 10×10 PS and SP options available.	More sizes and configurations.
1.3	✓	Open-loop, passive and multi-agent control.	More control options.
1.4	✓	Real-time injection of lock-up and loose faults.	N/A
1.5	✓	Input and sensor noise control.	Add user-defined force disturbance.
2. Visualisation Requirements			
2.1	✓	Schematic of system provided.	N/A
2.2	✓	Real-time animation of elements.	More advanced representation.
2.3	✓	Fault indicated by colour changes to elements.	N/A
2.4	✓	Health monitor provides status information.	Use fault detection algorithms with display
2.5	✓	Plots of key signals provided.	N/A
3. Useability, Portability and Extendability Requirements			
3.1	~	Demonstrator is very visual, however, no testing by outside parties has been undertaken.	Quantitative/qualitative assessment of useability could be undertaken.
3.2	✓	Key Simulink subsystems have been masked, and embedded s-functions not easily editable.	N/A
3.3	✓	Operational in Matlab/Simulink environment.	N/A
3.4	✓	Operational on standard PC.	N/A
3.5	✓	The demonstrator has standard requirements and is portable.	N/A
3.6	✓	Control functions are easily extendable for researchers involved with the project.	N/A

The demonstrator simulates a range of HRA sizes and configurations, and allows the user to choose between a selection of control laws, and system inputs. Dynamic visualisations of the physical system and key signals illustrates how the HRA is structured and how it operates. Faults may be injected within the system and the effects of these faults are summarised through a health monitoring panel. This aspect of the demonstrator should improve understanding of system capability and reliability within HRA structures. Visualisation of the multi-agent fault detection, communication and internal states is also realised within an agent state panel.

The overall package is clear in its operation and should be useable for those relatively familiar with Matlab/Simulink. However, this useability has not been quantified, and hence, studies of these kind could be undertaken. Much of the underlying system models, control algorithms etc. are hidden using masks, which makes the top-level operation clearer, whilst allowing extensions to be made with relative ease in the future. The demonstrator operates on a standard machine with a standard Matlab/Simulink set-up and thus, is portable.

Chapter 9

Conclusions & Future Work

9.1 Conclusions

This thesis has presented research relating to the high redundancy actuator concept, which is a novel approach to achieving fault tolerant actuation through the use of large numbers of small actuators in a highly modular, low capability topology.

Firstly, the background to fault tolerant systems was presented and it was concluded that current actuator redundancy schemes, whilst providing fault tolerance, increase the cost, weight and inefficiency of the system, and also limit the types of technology that can be feasibly used due to their susceptibility to lock-up failures. The HRA offers a solution to these issues, potentially increasing reliability and system efficiency and avoiding lock-up fault mode issues. However, the HRA is also a complex control problem, as the system is high order, containing many moving masses, and is changeable as the system must continue to operate in the presence of multiple element faults.

A survey of fault tolerant control methods and fault detection and diagnosis strategies was given. A great deal of research has been made in these areas resulting in a diverse range of methods that may be implemented. Passive fault tolerant control, where a single robust control law is designed which must provide adequate control under all fault conditions, is the simplest approach. However, the number of faults which may be tolerated can be limited using this approach and resultant performance is conservative. In application to HRA however, these issues are alleviated somewhat, as the faults have a reduced effect on the overall system.

In contrast, active fault tolerant control is where the control law is changed in response to faults. Changes to the control law may be pre-designed or synthesised online in response to information provided by a fault detection scheme. There are numerous methods of active fault tolerant control and fault detection, however, the consideration of these areas is all too often separate, with assumptions of perfect fault information and zero delays.

The objectives of this thesis were:

1. to investigate the use of moving coil actuators as elements in the high redundancy actuation scheme, including the modelling of an element and assemblies, model verification and fault modelling.

2. to investigate an active fault tolerant control approach for use within the HRA, and compare its performance to a passive fault tolerant controlled system.
3. to explore possibilities for fault detection and health monitoring methods for the high redundancy actuator.
4. to develop a software demonstrator which illustrates the operation of high redundancy actuation systems that comprise of many elements.

To these ends, the main content of the thesis began in Chapter 3 with the modelling of moving coil actuators from first principles. These models were verified experimentally, and full order, non-linear simulation models and reduced order linear control design models were formed. A methodology for combining these actuator element models into HRA structures was presented and their fault modes considered.

Chapter 4 presented the formulation of three HRA benchmark systems which were to be used within the control studies that followed. The dimensioning and requirements of these systems were based upon real applications. The three benchmark systems allow the demonstration of the two main HRA configuration types, and two levels of element redundancy, forming the most thorough analysis of high redundancy actuation configurations to date. The respective fault tolerance and efficiency of these systems was then considered.

Subsequently, the benchmark systems were used in the design of passive fault tolerant control in Chapter 5. A very simple approach to fault tolerant control was taken after an analysis of the system fault modes, based upon designing a classical controller to meet requirements in the nominal system, as the two major fault modes (loose and lock-up faults) have opposite effects. The resultant performance afforded some fault tolerance, but the performance requirements were not met for a large proportion of the potentially tolerable fault set dictated by the capability requirements of the system.

Chapter 7 considered an active fault tolerant strategy for the HRA based on multi-agent principles. Multi-agent systems consist of numerous simplified autonomous entities which operate in a localised fashion and collaborate to achieve system-wide goals. This concept is similar to that of high redundancy actuation, as the HRA consists of a large number of low capability elements which form an actuator that achieves control objectives greater than the capability of individual elements. Through application of multi-agent concepts, a control strategy may be constructed which uses simple fault detection and control reconfiguration concepts on a localised basis to achieve a more complex control result. It also allows integration of both fault detection and reconfiguration in a structured manner. The rule-based fault detection used was initially described within Chapter 6, alongside a more centralised method of health monitoring. The control reconfiguration used was a gain scheduling, multiple model method. The results of the scheme showed that active fault tolerant control of this nature can achieve much higher levels of performance, in comparison to the passive fault tolerant control method presented. The large majority of theoretically tolerable faults were accommodated by the scheme. However, this performance improvement comes at a cost. The resultant control strategy is much more complex than the passive fault tolerant controller, presenting

issues for real system implementation. Also, the performance of the system is linked with the performance of the fault detection, and as such, the risk of system malfunction is increased to some extent through the introduction of active fault tolerant control. Ultimately, it is the criticality and stringency of performance requirements of the application that will dictate the suitability of passive and active fault tolerant control. However, it has been shown that both approaches are feasible with the HRA scheme, and the essence of the trade-offs between the two approaches has been given.

Finally, the development of a software demonstrator was presented in Chapter 8. the purpose of this demonstrator was to visually illustrate the operation of HRAs of varying configuration and size, to complement the hardware demonstrators of the project which have limited configuration and limited numbers of elements. This demonstrator was developed within the Matlab/Simulink environment. The resulting demonstrator meets the requirements set out in that chapter.

9.2 Suggestions for Further Work

The work presented within this thesis has provided a significant contribution to high redundancy actuation research. However, there is much to be considered before real industrial implementation is achieved.

Technology is one of the major development areas that needs to be addressed. The actuation technology used within this thesis is for concept demonstration only. It is likely that a different technology will be required for effective implementation of HRAs containing many more elements. Future research may lead to piezoelectric (a discussion of which is given in Appendix A), micro-hydraulic, or MEMs technology solutions.

The health monitoring of the HRA is an area that must also be addressed. The health monitoring studies within this thesis are limited, as they were not the main focus of this work. Many more methods of condition monitoring and fault detection exist which could be implemented within this scheme, and studies should be made regarding the suitability of these approaches.

There is much potential for extension of the multi-agent control scheme presented in this thesis. Preliminary work has been conducted on limiting the effect of misdiagnosis. If agent's reconfigure their local control on detection of faults to force that fault behaviour in the local sub-system, then the control reconfigurations made agency-wide will be appropriate to the overall system behaviour. This may be achieved in the case of lock-up faults by changing the control to fix the position of the diagnosed element, for example through providing the controller with a fixed position reference and changing the controller to a strong integral loop. Other schemes such as distribution of inputs to maximise the reliability of elements could also be achieved through negotiation between agents.

The more general control challenges associated with the HRA have more or less been answered by research conducted within this phase of the project. However, further control methods could be researched and applied, perhaps to more application specific systems. Implementation of the control studies within this thesis is also possible, once the electro-magnetic

4×4 PS HRA is completed.

Finally, the formulation of design synthesis methodologies is also an area that requires research. The development of tools and standard procedures for moving from system requirement to HRA realisation, considering factors such as required reliabilities, capabilities and dimensioning, should be formed to aid and promote industrial implementation.

References

- [1] R. J. Patton, “Fault detection and diagnosis in aerospace systems using analytical redundancy,” *Computing and Control Engineering Journal*, vol. 2, no. 3, pp. 127–136, 1991.
- [2] T. S. Steffen, J. Davies, R. Dixon, R. M. Goodall, J. Pearson, and A. C. Zolotas, “Failure modes and probabilities of a high redundancy actuator,” in *IFAC World Congress, Seoul*, 2008.
- [3] X. Du, R. Dixon, R. M. Goodall, and A. C. Zolotas, “Modelling and control of a highly redundant actuator,” 2006.
- [4] ———, “Assessment of strategies for control of high redundancy actuators,” 2006.
- [5] ———, “Lqg control of a highly redundant actuator,” in *Conference for Advanced Intelligent Mechatronics*, 2007.
- [6] X. Du, R. Dixon, R. M. Goodall, and T. Steffen, “Validation of a two by two high redundancy actuator experimental results,” in *In proceedings of Actuator, International Exhibition on Smart Actuators and Drive Systems*, 2008.
- [7] X. Du, “Doctoral thesis, high redundancy actuation,” 2008.
- [8] B. M. Hanson, M. D. Brown, and J. Fisher, “Self sensing: closed-loop estimation for a linear electromagnetic actuator,” *American Control Conference, 2001. Proceedings of the 2001*, vol. 2, 2001.
- [9] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki, *Diagnosis and fault-tolerant control*. Germany: Springer New York, 2006.
- [10] R. Isermann and P. Balle, “Trends in the application of model-based fault detection and diagnosis of technical processes,” *Control Engineering Practice*, vol. 5, no. 5, pp. 709–719, 1997.
- [11] M. Staroswiecki and A. L. Gehin, “From control to supervision,” *Annual Reviews in Control*, vol. 25, pp. 1–11, 2001, compilation and indexing terms, Copyright 2006 Elsevier Inc. All rights reserved. [Online]. Available: [http://dx.doi.org/10.1016/S1367-5788\(01\)00002-5](http://dx.doi.org/10.1016/S1367-5788(01)00002-5)

-
- [12] T. Steffen, *Control reconfiguration of dynamical systems: linear approaches and structural tests*. New York: Springer, 2005.
 - [13] N. Aeronautics and S. Administration, "Fault tolerant design," National Aeronautics and Space Administration, Tech. Rep. PD-ED-1246, 1995.
 - [14] H. Hecht, *Systems reliability and failure prevention*. Artech House Publishers, 2003.
 - [15] G. Betta and A. Pietrosanto, "Instrument fault detection and isolation: state of the art and new research trends," *IEEE Transactions on Instrumentation and Measurement*, vol. 49, no. 1, pp. 100–107, 2000.
 - [16] I. Moir and A. Seabridge, *Aircraft Systems: Mechanical, Electrical and Avionics Subsystems Integration*. Wiley, 2008.
 - [17] R. Dixon, N. Gifford, C. Sewell, and M. C. Spalton, "Reacts: reliable electrical actuation systems," 1999, pp. 5/1–5/16.
 - [18] S. Cutts, "A collaborative approach to the more electric aircraft," in *Power Electronics, Machines and Drives, 2002. International Conference on (Conf. Publ. No. 487)*, June 2002, pp. 223–228.
 - [19] P. Janker and F. Claeysen, "New actuators for aircraft and space applications," in *Proc Actuator*, 2006, pp. 325–330.
 - [20] D. Patterson, G. Gibson, and R. Katz, "A case for redundant arrays of inexpensive disks (raid)," in *SIGMOD '88: Proceedings of the 1988 ACM SIGMOD international conference on Management of data*, 1988, pp. 109–116.
 - [21] T. Steffen, R. Dixon, R. M. Goodall, and A. C. Zolotas, "Robust control of a high redundancy actuator," in *UKACC Control 08*, 2008 2008.
 - [22] T. S. Steffen, R. Dixon, R. M. Goodall, and A. C. Zolotas, "Multi-variable control of a high redundancy actuator," in *In proceedings of Actuator, International Exhibition on Smart Actuators and Drive Systems*, 2008.
 - [23] T. S. Steffen, K. Michail, R. Dixon, A. C. Zolotas, and R. M. Goodall, "Optimal passive fault tolerant control of a high redundancy actuator," in *Safeprocess, Barcelona*, 2009.
 - [24] M. Blanke, R. Izadi-Zamanabadi, S. A. Bogh, and C. P. Lunau, "Fault-tolerant control systems - a holistic view," *Control Engineering Practice*, vol. 5, no. 5, pp. 693–702, 1997/5.
 - [25] M. Blanke, M. Staroswiecki, and N. E. Wu, "Concepts and methods in fault-tolerant control," *American Control Conference, 2001.Proceedings of the 2001*, vol. 4, 2001.
 - [26] R. Patton, "Robustness issues in fault-tolerant control," *Fault Diagnosis and Control System Reconfiguration, IEE Colloquium on*, p. 1, 1993.

-
- [27] R. J. Patton, "Fault-tolerant control systems: The 1997 situation," *IFAC Symposium on Fault Detection Supervision and Safety for Technical Processes*, vol. 3, 1997.
- [28] R. F. Stengel, "Intelligent failure-tolerant control," *Control Systems Magazine, IEEE*, vol. 11, no. 4, pp. 14–23, 1991.
- [29] R. V. White and F. M. Miles, "Principles of fault tolerance," *Applied Power Electronics Conference and Exposition, 1996.APEC'96.Conference Proceedings 1996., Eleventh Annual*, vol. 1, 1996.
- [30] Y. Zhang and J. Jiang, "Bibliographical review on reconfigurable fault-tolerant control systems," in *5th IFAC Symposium on fault detection, supervision and safety for technical processes*, 2003, pp. 265–276.
- [31] M. Blanke, M. Kinnaert, J. Schroder, J. Lunze, and M. Staroswiecki, *Diagnosis and fault-tolerant control*. Springer Verlag, 2003.
- [32] M. Mahmoud, J. Jiang, and Y. Zhang, *Active fault tolerant control systems: stochastic analysis and synthesis*. Springer Verlag, 2003.
- [33] H. Noura, D. Theilliol, J. Ponsart, and A. Chamseddine, *Fault-Tolerant Control Systems: Design and Practical Applications*. Springer Verlag, 2009.
- [34] H. Niemann and J. Stoustrup, "Reliable control using the primary and dual youla parameterizations," *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, vol. 4, 2002.
- [35] —, "Passive fault tolerant control of a double inverted pendulum - a case study," *Control Engineering Practice*, vol. 13, no. 8, pp. 1047–1059, 2005.
- [36] H. Noura, D. Sauter, F. Hamelin, and D. Theilliol, "Fault-tolerant control in dynamic systems: application to a winding machine," *Control Systems Magazine, IEEE*, vol. 20, no. 1, pp. 33–49, 2000.
- [37] F. Liao, J. L. Wang, and G. H. Yang, "Reliable robust flight tracking control: an lmi approach," *Control Systems Technology, IEEE Transactions on*, vol. 10, no. 1, pp. 76–89, 2002.
- [38] V. Kulatliumani, A. Arora, Y. M. Kim, P. Shankar, and R. K. Yedavalli, "Reliable control system design despite byzantine actuators," in *DETC2005: ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Sep 24-28 2005*, ser. Proceedings of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference - DETC2005. American Society of Mechanical Engineers, New York, NY 10016-5990, United States, 2005.
- [39] R. J. Veillette, "Reliable linear-quadratic state-feedback control," *Automatica*, vol. 31, no. 1, pp. 137–143, 1995.

-
- [40] R. J. Veillette, J. B. Medanic, and W. R. Perkins, "Design of reliable control systems," *Automatic Control, IEEE Transactions on*, vol. 37, no. 3, pp. 290–304, 1992.
 - [41] Q. Zhao and J. Jiang, "Reliable state feedback control system design against actuator failures," *Automatica*, vol. 34, no. 10, pp. 1267–1272, 1998.
 - [42] Y. W. Liang, D. C. Liaw, and T. C. Lee, "Reliable control of nonlinear systems," *Automatic Control, IEEE Transactions on*, vol. 45, no. 4, pp. 706–710, 2000.
 - [43] K. Zhou, J. Doyle, and K. Glover, *Robust and optimal control*. Prentice Hall Englewood Cliffs, NJ, 1996.
 - [44] M. Basseville, "Detecting changes in signals and systems a survey," *Automatica*, vol. 24, no. 3, pp. 309–326, 1988.
 - [45] R. N. Clark, "Instrument fault detection," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 14, no. 3, pp. 456–465, 1978.
 - [46] C. Dailly, "Fault monitoring and diagnosis," *Computing and Control Engineering Journal*, vol. 1, no. 2, pp. 57–62, 1990.
 - [47] J. J. Gertler, "Survey of model-based failure detection and isolation in complex plants," *Control Systems Magazine, IEEE*, vol. 8, no. 6, pp. 3–11, 1988.
 - [48] R. Isermann, "Process fault detection based on modeling and estimation methods - a survey," *Automatica*, vol. 20, no. 4, pp. 387–404, 1984/7.
 - [49] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri, "A review of process fault detection and diagnosis part i: Quantitative model-based methods," *Computers and Chemical Engineering*, vol. 27, no. 3, pp. 293–311, 2003.
 - [50] V. Venkatasubramanian, R. Rengaswamy, and S. N. Kavuri, "A review of process fault detection and diagnosis part ii: Qualitative models and search strategies," *Computers and Chemical Engineering*, vol. 27, no. 3, pp. 313–326, 2003.
 - [51] V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri, and K. Yin, "A review of process fault detection and diagnosis part iii: Process history based methods," *Computers and Chemical Engineering*, vol. 27, no. 3, pp. 327–346, 2003.
 - [52] A. S. Willsky, "A survey of design methods for failure detection in dynamic systems," *Automatica*, vol. 12, no. 6, pp. 601–611, 1976/11.
 - [53] E. Chow and A. Willsky, "Analytical redundancy and the design of robust failure detection systems," *Automatic Control, IEEE Transactions on*, vol. 29, no. 7, pp. 603–614, 1984.
 - [54] W. Chen and M. Saif, "An actuator fault isolation strategy for linear and nonlinear systems," 2005, pp. 3321–3326 vol. 5.

-
- [55] M. Basseville and A. Benveniste, *Detection of abrupt changes in signals and dynamical systems*. Springer, 1986.
 - [56] P. Young, "Parameter estimation for continuous-time models-a survey." Pergamon, 1980, p. 17.
 - [57] F. Hermans and M. Zarrop, "Parameter estimation using sliding mode principles," in *Proc. IFAC Symposium SAFEPROCESS*.
 - [58] B. Walker and K. Huang, "Fdi by extended kalman filter parameter estimation for an industrial actuator benchmark," *Control Engineering Practice*, vol. 3, no. 12, pp. 1769–1774, 1995.
 - [59] A. Bernieri, G. Betta, A. Pietrosanto, and C. Sansone, "A neural network approach to instrument fault detection and isolation," *Instrumentation and Measurement, IEEE Transactions on*, vol. 44, no. 3, pp. 747–750, 1995.
 - [60] P. M. Frank, S. X. Ding, and T. Marcu, "Model-based fault diagnosis in technical processes," *Transactions of the Institute of Measurement and Control*, vol. 22, no. 1, p. 57, 2000.
 - [61] C. T. Kowalski and T. Orłowska-Kowalska, "Neural networks application for induction motor faults diagnosis," *Mathematics and Computers in Simulation*, vol. 63, no. 3-5, pp. 435–448, 2003.
 - [62] M. R. Napolitano, Y. An, and B. A. Seanor, "A fault tolerant flight control system for sensor and actuator failures using neural networks," *Aircraft Design*, vol. 3, no. 2, pp. 103–128, 2000.
 - [63] B. Samanta, "Gear fault detection using artificial neural networks and support vector machines with genetic algorithms," *Mechanical Systems and Signal Processing*, vol. 18, no. 3, pp. 625–644, 2004/5.
 - [64] V. N. Vapnik, "An overview of statistical learning theory," *Neural Networks, IEEE Transactions on*, vol. 10, no. 5, pp. 988–999, 1999.
 - [65] M. Fouladirad and I. Nikiforov, "Optimal statistical fault detection with nuisance parameters," *Automatica*, vol. 41, no. 7, pp. 1157–1171, 2005/7.
 - [66] P. M. Frank, "Enhancement of robustness in observer-based fault detection," *International Journal of Control*, vol. 59, no. 4, pp. 955–981, 1994.
 - [67] R. J. Patton and J. Chen, "Robust fault detection of jet engine sensor systems using eigenstructure assignment," *Journal of Guidance, Control, and Dynamics*, vol. 15, no. 6, pp. 1491–1497, 1992.
 - [68] —, "On eigenstructure assignment for robust fault diagnosis," *International journal of robust and nonlinear control*, vol. 10, no. 14, pp. 1193–1208, 2000.

-
- [69] P. M. Frank and X. Ding, "Frequency domain approach to optimally robust residual generation and evaluation for model-based fault diagnosis," *Automatica (Journal of IFAC)*, vol. 30, no. 5, pp. 789–804, 1994.
- [70] J. J. Gertler and M. M. Kunwer, "Optimal residual decoupling for robust fault diagnosis," *International Journal of Control*, vol. 61, no. 2, pp. 395–421, 1995.
- [71] F. Hamelin and D. Sauter, "Robust fault detection in uncertain dynamic systems," *Automatica*, vol. 36, no. 11, pp. 1747–1754, 2000.
- [72] Z. Han, W. Li, and S. L. Shah, "Fault detection and isolation in the presence of process uncertainties," *Control Engineering Practice*, vol. 13, pp. 587–599, 2005.
- [73] M. Zhong, S. X. Ding, J. Lam, and H. Wang, "An lmi approach to design robust fault detection filter for uncertain lti systems," *Automatica*, vol. 39, no. 3, pp. 543–550, 2003.
- [74] P. Frank, "Analytical and qualitative model-based fault diagnosis-a survey and some new results," *European Journal of Control*, vol. 2, no. 1, pp. 6–28, 1996.
- [75] J. Chen, C. Roberts, and P. Weston, "Fault detection and diagnosis for railway track circuits using neuro-fuzzy systems," *Control Engineering Practice*, vol. 16, no. 5, pp. 585 – 596, 2008.
- [76] C. Roberts, H. P. B. Dassanayake, N. Lehrsab, and C. J. Goodman, "Distributed quantitative and qualitative fault diagnosis: railway junction case study," *Control Engineering Practice*, vol. 10, no. 4, pp. 419 – 429, 2002.
- [77] S. Yoon and J. MacGregor, "Fault diagnosis with multivariate statistical models part i: using steady state fault signatures," *Journal of Process Control*, vol. 11, no. 4, pp. 387–400, 2001.
- [78] P. S. Maybeck and R. D. Stevens, "Reconfigurable flight control via multiple model adaptive control methods," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 27, no. 3, pp. 470–480, 1991.
- [79] D. Moerder, N. Halyo, J. Broussard, and A. Caglayan, "Application of precomputed control laws in a reconfigurable aircraft flight control system," 1989.
- [80] C. D. Ormsby, J. F. Raquet, and P. S. Maybeck, "A new generalized residual multiple model adaptive estimator of parameters and states," *Mathematical and Computer Modelling*, vol. 43, no. 9-10, pp. 1092–1113, 2006, compilation and indexing terms, Copyright 2006 Elsevier Inc. All rights reserved. [Online]. Available: <http://dx.doi.org/10.1016/j.mcm.2005.12.003>
- [81] H. E. Rauch, "Autonomous control reconfiguration," *Control Systems Magazine, IEEE*, vol. 15, no. 6, pp. 37–48, 1995.

-
- [82] Y. Zhang and J. Jiang, "Integrated active fault-tolerant control using imm approach," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 37, no. 4, pp. 1221–1235, 2001.
- [83] Z. Gao and P. J. Antsaklis, "On the stability of the pseudo-inverse method for reconfigurable control systems," 1989, pp. 333–337 vol.1.
- [84] S. H. Lane and R. F. Stengel, "Flight control design using non-linear inverse dynamics," *Automatica*, vol. 24, no. 4, pp. 471–483, 1988.
- [85] J. W. Choi, D. Y. Lee, and M. H. Lee, "Reconfigurable control via eigenstructure assignment," in *Proceedings of the 1998 37th SICE Annual Conference, Jul 29-31 1998*, ser. Proceedings of the SICE Annual Conference. Society of Instrument and Control Engineers (SICE), Tokyo, Japan, 1998, pp. 1041–1045.
- [86] I. Konstantopoulos and P. Antsaklis, "An eigenstructure assignment approach to control reconfiguration," in *Proceedings of 4th IEEE Mediterranean Symposium on Control and Automation. Greece*, 1996.
- [87] Y. M. Zhang and J. Jiang, "Active fault-tolerant control system against partial actuator failures," *Control Theory and Applications, IEE Proceedings-*, vol. 149, no. 1, pp. 95–104, 2002.
- [88] M. R. Napolitano, Y. Song, and B. Seanor, "On-line parameter estimation for restructurable flight control systems," *Aircraft Design*, vol. 4, no. 1, pp. 19–50, 2001.
- [89] D. Shin, G. Moon, and Y. Kim, "Design of reconfigurable flight control system using adaptive sliding mode control: Actuator fault," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 219, no. 4, pp. 321–328, 2005.
- [90] C. P. Tan and C. Edwards, "Sliding mode observers for detection and reconstruction of sensor faults," *Automatica*, vol. 38, no. 10, pp. 1815–1821, 2002.
- [91] M. Maki, J. Jiang, and K. Hagino, "A stability guaranteed active fault-tolerant control system against actuator failures," *International Journal of Robust and Nonlinear Control*, vol. 14, no. 12, pp. 1061–1077, 2004.
- [92] M. Guler, S. Clements, L. M. Wills, B. S. Heck, and G. J. Vachtsevanos, "Transition management for reconfigurable hybrid control systems," *IEEE Control Systems Magazine*, vol. 23, no. 1, pp. 36–49, 2003.
- [93] M. Minsky, *The society of mind*. Chichester Sussex: William Heinemann Ltd., 1986.
- [94] G. Weiss, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. MIT Press, 1999.

-
- [95] N. R. Jennings, K. Sycara, and M. Wooldridge, "A roadmap of agent research and development," *Autonomous Agents and Multi-Agent Systems*, vol. 1, no. 1, pp. 7–38, 1998.
- [96] J. Ferber, *Multi-agent systems: an introduction to distributed artificial intelligence*. United States: Addison-Wesley, 1999.
- [97] M. P. Georgeff and A. L. Lansky, "Reactive reasoning and planning," *Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI-87)*, p. 677, 1987.
- [98] M. Cao, A. Morse, C. Yu, B. Anderson, and S. Dasgupta, "Controlling a triangular formation of mobile autonomous agents," *Decision and Control, 2007 46th IEEE Conference on*, pp. 3603–3608, 2007.
- [99] H. Shi, "Flocking coordination of multiple mobile autonomous agents with asymmetric interactions and switching topology," *Intelligent Robots and Systems, 2005. IEEE International Conference on*, pp. 771–776, 2005.
- [100] M. Hadzic, "Holonc multi-agent system complemented by human disease ontology supporting biomedical community," *Computational Intelligence Methods and Applications, 2005 ICSC Congress on*, p. 5 pp., 2005.
- [101] J. A. Hossack, J. Menal, S. D. J. McArthur, and J. R. McDonald, "A multiagent architecture for protection engineering diagnostic assistance," *Power Systems, IEEE Transactions on*, vol. 18, no. 2, pp. 639–647, 2003.
- [102] C. Kenfack, "Modeling community of practices using intelligent agents," *Computer Supported Cooperative Work in Design, 2007. CSCWD 2007. 11th International Conference on*, pp. 377–382, 2007.
- [103] M. Ljungberg and A. Lucas, "The oasis air-traffic management system," *Proceedings of the Second Pacific Rim International Conference on Artificial Intelligence, PRICAI*, vol. 92, 1992.
- [104] R. H. Guttman, A. G. Moukas, and P. Maes, "Agent-mediated electronic commerce: a survey," *The Knowledge Engineering Review*, vol. 13, no. 02, pp. 147–159, 2001.
- [105] H. V. D. Parunak, A. D. Baker, and S. J. Clark, "The aaria agent architecture: an example of requirements-driven agent-based system design," *Proceedings of the first international conference on Autonomous agents*, pp. 482–483, 1997.
- [106] R. Roberts, *Zone Logic: A Unique Method of Practical Artificial Intelligence*. Computer Books, 1989.
- [107] S. Britain, A. Gibb, and C. Roberts, "Automatic reconfiguration of a robotic arm using a multi-agent approach," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 222, no. 2, pp. 127–135, 2008.

-
- [108] L. Overgaard, H. G. Petersen, and J. W. Perram, "Motion planning for an articulated robot: A multi-agent approach," *Proceedings of Modelling Autonomous Agent in a Multi-Agent World*, pp. 171–182.
- [109] A. Newell and H. A. Simon, "Computer science as empirical enquiry," *Communications of the ACM*, vol. 19, no. 3, pp. 113–126, 1976.
- [110] R. Brooks, "A robust layered control system for a mobile robot," *Robotics and Automation, IEEE Journal of*, vol. 2, no. 1, p. 14, 1986.
- [111] W. Shen, J. P. A. Barth-Åss, and D. H. Norrie, *Multi-Agent Systems for Concurrent Intelligent Design and Manufacturing*. London: Taylor and Francis, 2001.
- [112] A. S. Rao and M. P. Georgeff, "Modeling agents within a bdi-architecture," *Proc. of the 2nd International Conference on Principles of Knowledge Representation and Reasoning*, p. 473.
- [113] M. R. Genesereth, M. L. Ginsberg, and J. S. Rosenschein, *Cooperation without communication*. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 1988.
- [114] M. Fenster, S. Kraus, and J. S. Rosenschein, "Coordination without communication: Experimental validation of focal point techniques," *Readings in Agents*, 1997.
- [115] C. Hewitt, P. Bishop, and R. Steiger, "A universal modular actor formalism for artificial intelligence," *3rd International Joint Conference on Artificial Intelligence*, pp. 235–245, 1973.
- [116] R. G. Smith, "Communication and control in problem solver," *IEEE Transactions on Computers*, vol. 29, no. 12, 1980.
- [117] "The foundation for intelligent physical agents," 2008. [Online]. Available: <http://www.fipa.org/>
- [118] N. R. Jennings, "Commitments and conventions: The foundation of coordination in multi-agent systems," *The Knowledge Engineering Review*, vol. 8, no. 3, pp. 223–250, 1993.
- [119] E. H. van Leeuwen, D. Norrie, B. H. P. Res, and V. Mulgrave, "Holons and holarchies [intelligent manufacturing systems]," *Manufacturing Engineer*, vol. 76, no. 2, pp. 86–88, 1997.
- [120] K. Fischer, "Agent-based design of holonic manufacturing systems," *Robotics and Autonomous Systems*, vol. 27, no. 1-2, pp. 3–13, 1999.
- [121] J. Butler and H. Ohtsubo, "Addyms: Architecture for distributed dynamic manufacturing scheduling," *Artificial Intelligence Applications in Manufacturing*, p. 199, 1992.
- [122] R. Englemore and T. Morgan, *Blackboard systems*. Addison-Wesley, 1988.

-
- [123] FIPA, "Fipa abstract architecture specification," 2002. [Online]. Available: <http://www.fipa.org/repository/architecturespecs.html>
- [124] S. Eo, T. Chang, D. Shin, and E. Yoon, "Cooperative problem solving in diagnostic agents for chemical processes," *Computers and Chemical Engineering*, vol. 24, no. 2-7, pp. 729–734, 2000.
- [125] R. Ferrari, T. Parisini, and M. Polycarpou, "A fault detection scheme for distributed nonlinear uncertain systems," in *2006 IEEE Computer Aided Control System Design*.
- [126] T.-J. Li, Y.-Q. Peng, H.-W. Zhao, and K. Li, "Application of multi-agent in control and fault diagnosis systems," in *Proceedings of 2004 International Conference on Machine Learning and Cybernetics, Aug 26-29 2004*, vol. 1, Res. Inst. of Robotics and Automat., Hebei University of Technology, Tianjin, 300130, China. Shanghai, China: Institute of Electrical and Electronics Engineers Inc., New York, NY 10016-5997, United States, 2004, pp. 194–197.
- [127] S. Lee, S. Yang, and B. Song, "Hierarchical Fault Detection and Diagnosis for Unmanned Ground Vehicles," *Proceedings of 7th Asian Control Conference, Hong Kong, China*, 2009.
- [128] R. Marzi and P. John, "Supporting fault diagnosis through a multi-agent architecture," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 216, no. 4, pp. 627–631, 2002.
- [129] Y. Murphey, J. Crossman, Z. Chen, and J. Cardillo, "Automotive fault diagnosis-part ii: a distributed agent diagnostic system," *IEEE Transactions on Vehicular Technology*, vol. 52, no. 4, pp. 1076–1098, 2003.
- [130] G. Niu, T. Han, B. Yang, and A. Tan, "Multi-agent decision fusion for motor fault diagnosis," *Mechanical Systems and Signal Processing*, vol. 21, no. 3, pp. 1285–1299, 2007.
- [131] L. Liu, K. Logan, D. Cartes, and S. Srivastava, "Fault detection, diagnostics, and prognostics: software agent solutions," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 4 Part 1, pp. 1613–1622, 2007.
- [132] W. Du, Z. Chen, H. Wang, and R. Dunn, "Feasibility of online collaborative voltage stability control of power systems," *Generation, Transmission & Distribution, IET*, vol. 3, no. 2, pp. 216–224, 2009.
- [133] E. M. Davidson, S. D. J. McArthur, J. R. McDonald, T. Cumming, and I. Watt, "Applying multi-agent system technology in practice: automated management and analysis of scada and digital fault recorder data," *Power Systems, IEEE Transactions on*, vol. 21, no. 2, pp. 559–567, 2006.

-
- [134] J. Hossack, J. Menal, S. McArthur, and J. McDonald, "A multiagent architecture for protection engineering diagnostic assistance," *Power Systems, IEEE Transactions on*, vol. 18, no. 2, pp. 639–647, 2003.
- [135] S. D. J. McArthur, M. Davidson, M. Catterson, L. Dimeas, D. Hatziargyriou, F. Ponci, and T. Funabashi, "Multi-agent systems for power engineering applications part i: Concepts, approaches, and technical challenges," *Power Systems, IEEE Transactions on*, vol. 22, no. 4, pp. 1743–1752, 2007.
- [136] S. D. J. McArthur, E. M. Davidson, V. M. Catterson, A. L. Dimeas, N. D. Hatziargyriou, F. Ponci, and T. Funabashi, "Multi-agent systems for power engineering applications part ii: Technologies, standards, and tools for building multi-agent systems," *Power Systems, IEEE Transactions on*, vol. 22, no. 4, pp. 1753–1759, 2007.
- [137] S. D. J. McArthur, S. M. Strachan, and G. Jahn, "The design of a multi-agent transformer condition monitoring system," *Power Systems, IEEE Transactions on*, vol. 19, no. 4, pp. 1845–1852, 2004.
- [138] T. Nagata and H. Sasaki, "A multi-agent approach to power system restoration," *IEEE Transactions on Power Systems*, vol. 17, no. 2, pp. 457–462, 2002.
- [139] C. Rehtanz, *Autonomous systems and intelligent agents in power system control and operation*. Springer Verlag, 2003.
- [140] Y. N. Guo, J. Cheng, D. W. Gong, and J. H. Zhang, "A novel multi-agent based complex process control system and its application," *Lecture notes in control and information sciences*, vol. 344, pp. 319–330, 2006.
- [141] Y. Li, T. Yang, J. X. Xu, and F. Lv, "Multiagent-based monitoring-fault diagnosis-control integrated system for chemical process," *Machine Learning and Cybernetics, 2005.Proceedings of 2005 International Conference on*, vol. 3, 2005.
- [142] I. Seilonen, "Reactive and deliberative control and cooperation in multi-agent system based process automation," *Computational Intelligence in Robotics and Automation, 2005. CIRA 2005. Proceedings. 2005 IEEE International Symposium on*, pp. 469–474, 2005.
- [143] R. Schneider and W. Marquardt, "Information technology support in the chemical process design life cycle," *Chemical Engineering Science*, vol. 57, no. 10, pp. 1763–1792, 2002.
- [144] J. Velasco, J. Gonzalez, L. Magdalena, and C. Iglesias, "Multiagent-based control systems: A hybrid approach to distributed process control," *Control Engineering Practice*, vol. 4, no. 6, pp. 839–845, 1996.
- [145] A. van Breemen and T. de Vries, "An agent-based framework for designing multi-controller systems."

-
- [146] A. J. N. van Breemen and T. J. A. de Vries, "Design and implementation of a room thermostat using an agent-based approach," *Control Engineering Practice*, vol. 9, no. 3, pp. 233 – 248, 2001.
- [147] SMAC, "Actuator data sheet - electric linear/rotary moving coil actuators la190-50," 2004.
- [148] P. Campbell, *Permanent Magnet Materials and their Application*. Cambridge: Cambridge University Press, 1996.
- [149] H. D. Chai, *Electromechanical motion devices*. United States: Prentice Hall, 1998.
- [150] Mathworks, "xpc target 4 data sheet," 2008. [Online]. Available: <http://www.mathworks.com/products/xpctarget.html>
- [151] R. Pratt, *Flight control systems: practical issues in design and implementation*. Iet, 2000.
- [152] R. Goodall, J. Pearson, and I. Pratt, "Actuator technologies for active secondary suspensions," in *Proceedings of the International Conference On Railway Speed-up Technology*, vol. 2, pp. 377–382.
- [153] T. E. Menke and P. S. Maybeck, "Sensor/actuator failure detection in the vista f-16 by multiplemode adaptive estimation," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 31, no. 4, pp. 1218–1229, 1995.
- [154] M. Napolitano and R. Swaim, "New technique for aircraft flight control reconfiguration," *Journal of Guidance, Control, and Dynamics*, vol. 14, no. 1, p. 184190, 1991.
- [155] H. A. P. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with markovswitching coefficients," *Automatic Control, IEEE Transactions on*, vol. 33, no. 8, pp. 780–783, 1988.
- [156] Y. Bar-Shalom, X. R. Li, T. Kirubarajan, and J. Wiley, *Estimation with applications to tracking and navigation*. Wiley New York, 2001.
- [157] R. Mehra, C. Rago, S. Seereeram, S. S. Co, and M. A. Woburn, "Autonomous failure detection, identification and fault-tolerant estimation with aerospace applications," in *Aerospace Conference, 1998. Proceedings., IEEE*, vol. 2, 1998.
- [158] Y. Hayashi, H. Tsunashima, and Y. Marumo, "Fault detection of railway vehicles using multiple model approach," in *SICE-ICASE, 2006. International Joint Conference*, 2006, pp. 2812–2817.
- [159] M. Hashimoto, S. Watanabe, and K. Takahashi, "Sensor fault detection and isolation for a powered wheelchair," in *Advanced Intelligent Mechatronics, 2007 IEEE/ASME International Conference on*, 2007, pp. 1–6.

-
- [160] Y. Hayashi, H. Tsunashima, and Y. Marumo, "Fault detection of railway vehicle suspension systems using multiple-model approach," *Journal of Mechanical Systems for Transportation and Logistics*, vol. 1, no. 1, pp. 88–99, 2008.
- [161] J. Davies, T. S. Steffen, R. Dixon, and R. M. Goodall, "Fault detection in high redundancy actuation using an interacting multiple-model approach," in *Safeprocess, Barcelona*, 2009.
- [162] H. Wang and C. Wang, "Intelligent agents in the nuclear industry," *Computer*, vol. 30, no. 11, pp. 28–31, 1997.
- [163] D. D. Siljak, *Large-scale dynamic systems: stability and structure*. North Holland, 1978.
- [164] ———, *Decentralized control of complex systems*. London: Academic Press Limited, 1991, vol. 184.
- [165] D. Liberzon, *Switching in systems and control*. Springer, 2003.
- [166] M. Branicky, "Multiple lyapunov functions and other analysis tools for switched and hybrid systems," *Automatic Control, IEEE Transactions on*, vol. 43, no. 4, pp. 475–482, Apr 1998.
- [167] A. Morse, "Supervisory control of families of linear set-point controllers—part 1: exact matching," *IEEE Trans. Automat. Contr*, 1996.
- [168] J. Hespanha and A. Morse, "Stability of switched systems with average dwell-time," in *Decision and Control, 1999. Proceedings of the 38th IEEE Conference on*, vol. 3, 1999, pp. 2655–2660 vol.3.

Appendix A

Piezo Technology Report

Piezoelectric actuation and its suitability for use within high redundancy actuation

Report following the Piezoelectric Actuation Workshop, IMechE, 2009.

Jessica Davies (j.davies@lboro.ac.uk)

1. Introduction

Piezoelectric actuation is an emerging technology based upon the converse piezoelectric effect, where a mechanical strain is produced in response to an applied electric field in materials such as Lead Zirconate Titanate (PZT). This report provides a brief summary of current piezoelectric actuation technology, followed by a discussion of the impact and relevance to high redundancy actuation.

2. Current Piezoelectric Actuator Technology

Piezoelectric actuators encompass a number of different configurations which tend to fall into one of the following categories:

- **Stack actuators** – Stack actuators comprise of a number of ceramic disks arranged mechanically in series and electrically in parallel. This arrangement allows increased displacement at a reduced voltage. Stack actuators can withstand high pressures and exhibit the highest stiffness.
- **Amplified piezo actuators** – Although stack actuators provide high forces, their travel capability is small. Travel can be increased if the stack is combined with some mechanical elements that amplify the motion, at the expense of a reduction in force capability. Hydraulic amplification is also possible [2, 6].
- **Piezoelectric benders** – Bender actuators are formed by gluing a piezoceramic strip to a passive metal substrate. The ceramic expands or contracts in proportion to the applied voltage, whilst the metal substrate does not change in length. This results in a deflection proportional to the voltage input. As in amplified piezo actuators, the movement is amplified and the force reduced.
- **Piezoelectric motors** – Piezoelectric motors are usually rotational devices that operate through frictional transmission of small, repetitive movements. They are often driven at a resonant frequency in order to extract the maximum mechanical output. Linear, non-resonant motors, such as inchworm motors are also available. Generally, piezo motors are low speed and high torque in comparison to DC motors.

A more detailed discussion of these piezo actuation types can be found in [4].

3. Implications for High Redundancy Actuation

HRA requires small-scale actuation technology. Piezoelectric actuators meet this requirement with unit sizes typically in the order of mm-cm, however, micro-scale actuators are possible [5]. Other aspects of their suitability for use within HRA are considered in the following subsections.

3.1. Operational Capabilities

A general indication of the travel and force capabilities of current piezo actuators is given in Figure 1. A wide range of operational capabilities are available, which should make dimensioning for any given HRA application possible. Travel capabilities are effectively unlimited if linear translation of piezo motors is implemented. Whereas high forces could be provided by stacked actuators for applications such as active railway suspension.

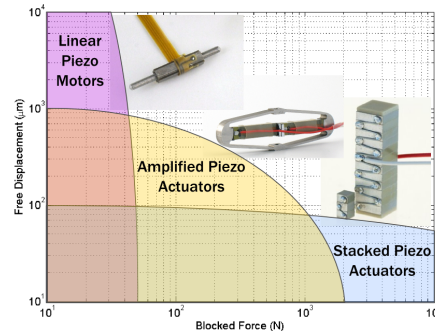


Figure 1 - Travel and force capabilities of piezo electric actuators.

3.2. Fault Modes

The piezoceramic itself has no moving parts, and as such, the likelihood of failure is reduced. Most failures occur due to excessive mechanical stress. PZT ceramics are brittle, and unable to withstand tensile or shear forces, torque or mechanical shock. Measures must be taken to protect the actuators from these forces.

Temperature is also often stated as an issue in piezoelectric actuators. PZT materials have a hard operational limit dictated by the Curie temperature, above which the material permanently loses polarisation. This temperature is in the order of 200-300°C. However, in practise the operating temperature must be lower than this to avoid de-polarisation effects (approx. 100 °C). Temperature changes also induce thermal expansion within the material, changing the displacement capabilities of the piezo actuator. Significant positioning variations are possible if operating over a wide temperature region.

PZT ceramics must be encapsulated to protect them from moisture. Failures can occur if humidity or conductive materials such as metal dust degrade this insulation, leading to irreparable dielectric breakdown.

The likelihood of these fault types will not be reduced through packaging many piezoelectric actuators together in a HRA, as they are common mode failures.

However, faults are also possible within the electric drive of the system (i.e. short-circuit, open-circuit) and in any mechanical element that translates the piezoceramic's movement (i.e. jamming faults, loose faults). These faults are conceivably accommodated by a HRA structure.

3.3. Configuration

As piezo actuators are small with low displacement capabilities, multi-actuator arrangements are common. Piezoceramics are routinely used in serial configurations in actuation stacks. Parallel deployment to increase force capabilities is also suggested by Cedrat (Figure 2).

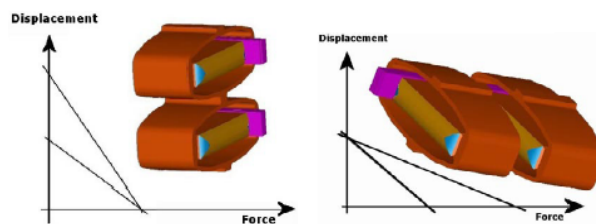


Figure 2 - Serial and parallel arrangement of amplified piezo actuators taken from [1].

An architecture that combines both series and parallel actuation is shown in Figure 3. Two sets of quasi-serial actuators are arranged in a parallel push-pull configuration. Whilst, the push-pull configuration is used in an attempt to eliminate thermal effects, it has the added effect of increased force. Travel is also amplified as the serial stacks push against each other. Configurations of this sort could be extended to form a HRA.

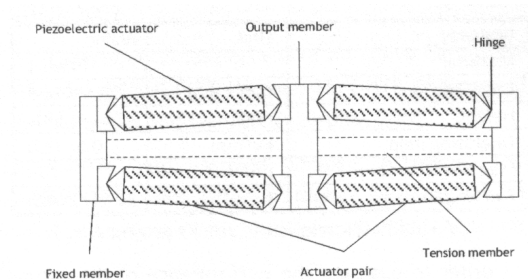


Figure 3 - Diamond frame piezo actuator [3].

3.4. Control

In contrast to many other types of actuation, piezo actuators are commonly operated open-loop, as displacement is approximately equal to the drive voltage. Hysteresis, non-linearity and creep effects limit accuracy, however. Closed-loop control is necessary to ensure long-term stability and linear performance. Charge or current control is preferred to voltage control, as this eliminates hysteresis effects.

3.5. Costs

Generally, the piezo actuators cost in the region of a few hundred pounds. However, the main expense seems to be in the required amplifiers, which cost in the region of £1000. Piezoelectric actuators require high voltage drivers that can deliver 200-2000 Volts peak-to-peak at high bandwidths.

4. Conclusions

Piezo electric actuators offer a compact solution for high speed, accurate operation, that can be tailored for a wide range travel and force requirements, making this an attractive technology for HRA. However, consideration should be given to issues such as thermal operation limits, mechanical stress, and costs. If piezo materials are to be used within a HRA, an integrated design approach is important. The electrical drive circuit; mechanical positioning and coupling; and environment protection need to be incorporated into the design if a reliable solution is to be achieved. Given this, collaboration with piezo specialists, such as Cedrat, may be considered.

5. References

- [1] Cedrat. Cedrat Piezo Products Catalogue, Version 3.2, 2007.
- [2] Keogh, P., 'The application potential of piezoelectric actuation for back-up bearing in magnetic bearing systems', IMechE Piezoelectric Actuation Workshop, London, 2009.
- [3] Mangeot, C., Andersen, B., Hilditch, R., 'New actuators for aerospace', IMechE Piezoelectric Actuation Workshop, London, 2009.
- [4] Jons, J.L., 'Emerging actuator technologies: A micromechatronic approach', John Wiley & Sons Ltd., London, 2005.
- [5] Wilson, S., 'Ultrasonic micromotors, piezoelectric microvalves and micropumps', IMechE Piezoelectric Actuation Workshop, London, 2009.
- [6] Zulch, S., Cooke, M., Hardy, M., Nebojsa, M., 'An innovative state-of-the-art direct acting diesel injector', IMechE Piezoelectric Actuation Workshop, London, 2009.

Appendix B

Modelling

Linear and Linear/Rotary Moving Coil Actuators

Two Axis Moving Coil Designed for Pick, Orient and Place

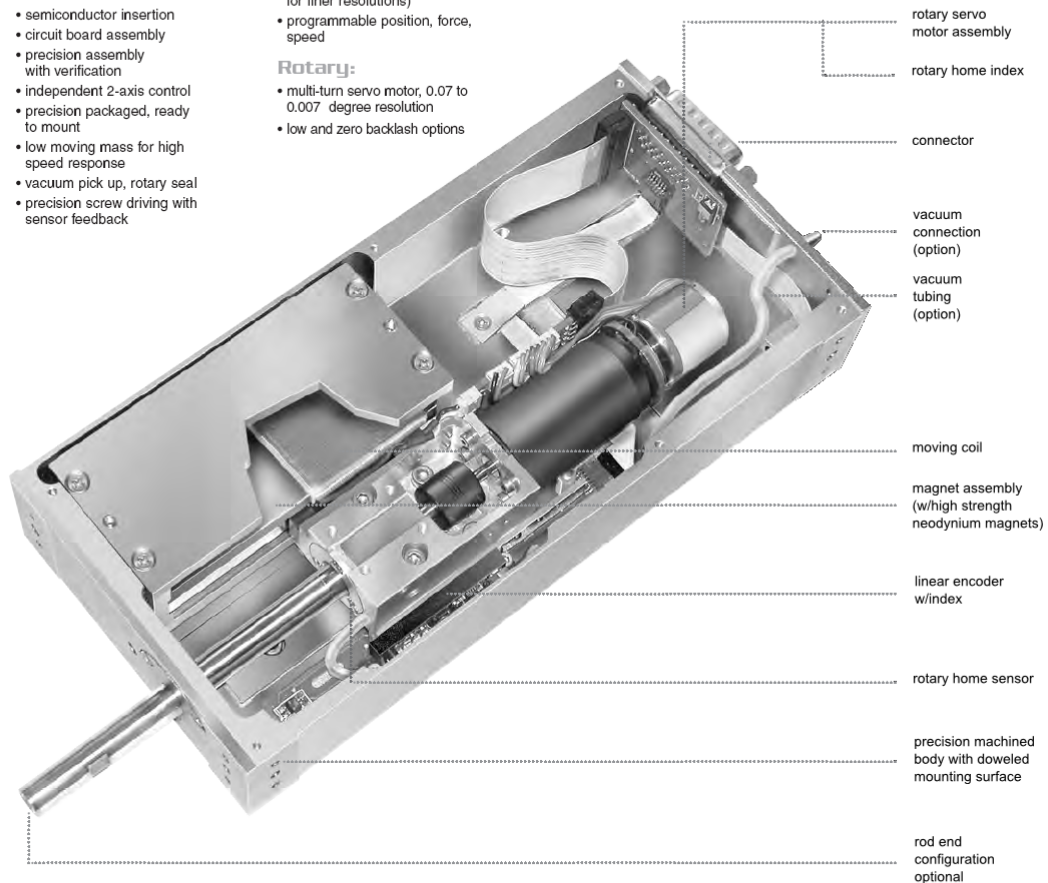
- semiconductor insertion
- circuit board assembly
- precision assembly with verification
- independent 2-axis control
- precision packaged, ready to mount
- low moving mass for high speed response
- vacuum pick up, rotary seal
- precision screw driving with sensor feedback

Linear:

- up to 150mm stroke, 5, 1, 0.5 micron resolution (consult factory for finer resolutions)
- programmable position, force, speed

Rotary:

- multi-turn servo motor, 0.07 to 0.007 degree resolution
- low and zero backlash options



Linear and Linear/Rotary Actuators

	LAL, LAS 10-005	LAL-LAS 15-015	LAR 15-015	LAL,LAS 20	LAR 20	LAL, LAS LAR 30-015	LAL,LAS LAR 30-025
Volts	12	24	24	24	24	24	24
Size: LxWxH	45 x 70 x 10	120x 58 x 15	120x 58 x 15	85x65x20	115x65x20	125x83x31	150x83x34
Stroke, mm	5	15	15	10 / 15 / 25	15	15	25
Peak Force, N	3.8	3.8	5	8 / 7 / 4	7	14.5	11.3
Continuous Force, N	2.5	2.7	4	5.5 / 5 / 3	5	7.4	5.9
Force Constant, N/A	2.5	2.7	4	5.5 / 5 / 3	5	7.4	5.9
Weight, kg	.10	.23	.25	0.34	0.41	0.69	0.74 (LAR 0.8)
Moving Mass, kg	.02	.05	.07	0.07	0.09	0.15 (LAR 0.21)	0.15 (LAR 0.21)
Rod Diameter, mm	4	6	6	6	6	8	8
Runout, micron	50	50	50	50	50	50	50
Rotary	No	No	Yes	No	Yes	Yes	Yes
Torque, N-m			.0084		.008	0.1	0.1
Gear Ratio			direct		Direct drive	76:1	76:1
Rotary Encoder Counts			20,000		14K	4864	4864
Speed, rpm			5000		5000	150	150
Shaft Thread F/M,mm	NA/3	3/4	3/4	3/4	3/4	4/6	4/6





Low Cost, Ultracompact $\pm 2\text{ g}$ Dual-Axis Accelerometer

ADXL311

FEATURES

Low cost

High resolution

Dual-axis accelerometer on a single IC chip

5 mm \times 5 mm \times 2 mm CLCC package

Low power < 400 μA (typ)

X-axis and Y-axis aligned to within 0.1° (typ)

BW adjustment with a single capacitor

Single-supply operation

High shock survival

APPLICATIONS

Tilt and motion sensing in cost-sensitive applications

Smart handheld devices

Computer security

Input devices

Pedometers and activity monitors

Game controllers

Toys and entertainment products

GENERAL DESCRIPTION

The ADXL311 is a low cost, low power, complete dual-axis accelerometer with signal conditioned voltage outputs, all on a single monolithic IC. The ADXL311 is built using the same proven iMEMS[®] process used in over 100 million Analog Devices accelerometers shipped to date, with demonstrated 1 FIT reliability (1 failure per 1 billion device operating hours).

The ADXL311 will measure acceleration with a full-scale range of $\pm 2\text{ g}$. The ADXL311 can measure both dynamic acceleration (e.g., vibration) and static acceleration (e.g., gravity). The outputs are analog voltages proportional to acceleration.

The typical noise floor is $300\text{ }\mu\text{g}/\sqrt{\text{Hz}}$ allowing signals below 2 mg (0.1° of inclination) to be resolved in tilt sensing applications using narrow bandwidths (10 Hz).

The user selects the bandwidth of the accelerometer using capacitors C_X and C_Y at the X_{FILT} and Y_{FILT} pins. Bandwidths of 1 Hz to 2 kHz may be selected to suit the application.

The ADXL311 is available in a 5 mm \times 5 mm \times 2 mm 8-terminal hermetic CLCC package

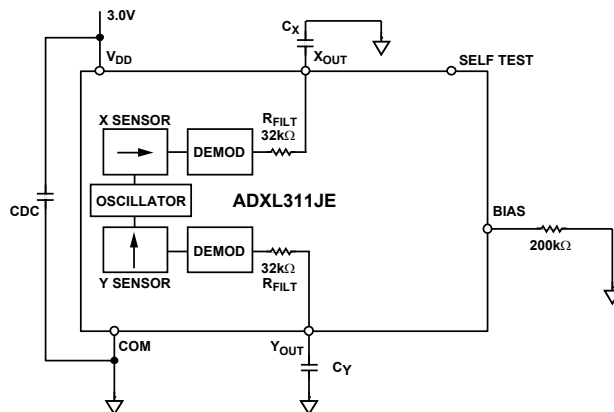


Figure 1. Functional Block Diagram

Rev. A

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective companies.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781.329.4700 Fax: 781.326.8703
www.analog.com
© 2003 Analog Devices, Inc. All rights reserved.

ADXL311

TABLE OF CONTENTS

Specifications.....	3	Pin Configuration and Functional Descriptions.....	9
Absolute Maximum Ratings.....	4	Outline Dimensions	10
Typical Performance Characteristics	5	Ordering Guide	10
Theory of Operation	7		
Applications.....	7		
Design Trade-Offs for Selecting Filter Characteristics: The Noise/BW Trade-Off.....	7		
Using the ADXL311 as a Dual-Axis Tilt Sensor.....	8		

REVISION HISTORY

7/03—Data sheet changed from Rev. 0 to Rev. A. Change to OUTLINE DIMENSIONS.....	10
Revision 0: Initial Version	

SPECIFICATIONS

Table 1. $T_A = 25^\circ\text{C}$, $V_{DD} = 3\text{ V}$, $R_{BIAS} = 125\text{ k}\Omega$, Acceleration = 0 g , unless otherwise noted.)

Parameter	Conditions	Min	Typ	Max	Units
SENSOR INPUT	Each Axis				
Measurement Range			± 2		g
Nonlinearity	Best Fit Straight Line		0.2		% of FS
Alignment Error ¹			± 1		Degrees
Alignment Error	X Sensor to Y Sensor		0.01		Degrees
Cross Axis Sensitivity ²			± 2		%
SENSITIVITY	Each Axis				
Sensitivity at X_{FILT} , Y_{FILT}	$V_{DD} = 3\text{ V}$	140	167	195	mV/g
Sensitivity Change due to Temperature ³	Delta from 25°C		-0.025		%/ $^\circ\text{C}$
ZERO g BIAS LEVEL	Each Axis				
0 g Voltage X_{FILT} , Y_{FILT}	$V_{DD} = 3\text{ V}$	1.2	1.5	1.8	V
0 g Offset vs. Temperature	Delta from 25°C		2.0		mg/ $^\circ\text{C}$
NOISE PERFORMANCE					
Noise Density	@ 25°C		300		$\mu\text{g}/\sqrt{\text{Hz}}$ RMS
FREQUENCY RESPONSE					
3 dB Bandwidth	At Pins X_{FILT} , Y_{FILT}		6		kHz
Sensor Resonant Frequency			10		kHz
FILTER					
R_{FILT} Tolerance	32 k Ω Nominal		± 15		%
Minimum Capacitance	At Pins X_{FILT} , Y_{FILT}	1000			pF
SELF TEST					
X_{FILT} , Y_{FILT}	Self Test 0 to 1		45		mV
POWER SUPPLY					
Operating Voltage Range		2.7		5.25	V
Quiescent Supply Current			0.4	1.0	mA
Turn-On Time			$160 \times C_{FILT} + 0.3$		ms
TEMPERATURE RANGE					
Operating Range		0		70	$^\circ\text{C}$

¹ Alignment error is specified as the angle between the true and indicated axis of sensitivity (Figure 1).

² Cross axis sensitivity is the algebraic sum of the alignment and the inherent sensitivity errors.

³ Defined as the output change from ambient to maximum temperature or ambient to minimum temperature.

ADXL311

ABSOLUTE MAXIMUM RATINGS

Table 2.

Parameter	Rating
Acceleration (Any Axis, Unpowered)	3,500 g, 0.5 ms
Acceleration (Any Axis, Powered, $V_{DD} = 3\text{ V}$)	3,500 g, 0.5 ms
V_{DD}	-0.3 V to +0.6 V
Output Short-Circuit Duration, (Any Pin to Commom)	Indefinite
Operating Temperature Range	-55°C to +125°C
Storage Temperature	-65°C to +150°C

Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational section of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Table 3. Package Characteristics

Package Type	θ_{JA}	θ_{JC}	Device Weight
8-Lead CLCC	120°C/W	TBD°C/W	<1.0 gram

TYPICAL PERFORMANCE CHARACTERISTICS

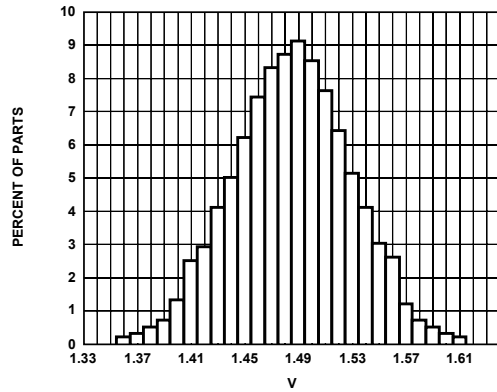


Figure 2. X-Axis Zero g BIAS Output Distribution

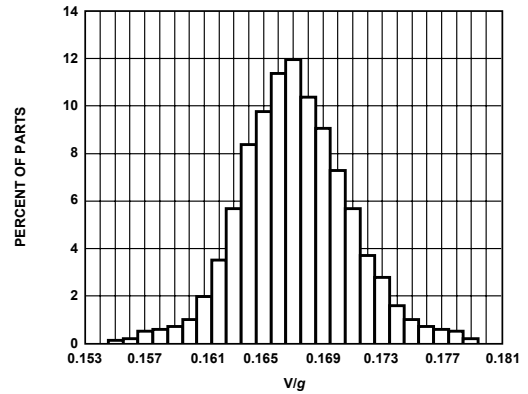


Figure 5. Y-Axis Sensitivity Distribution at Y_{OUT}

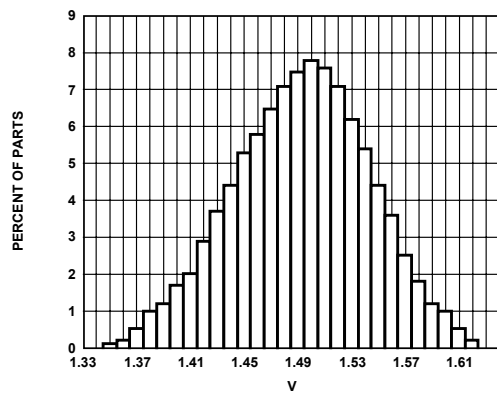


Figure 3. Y-Axis Zero g BIAS Output Distribution

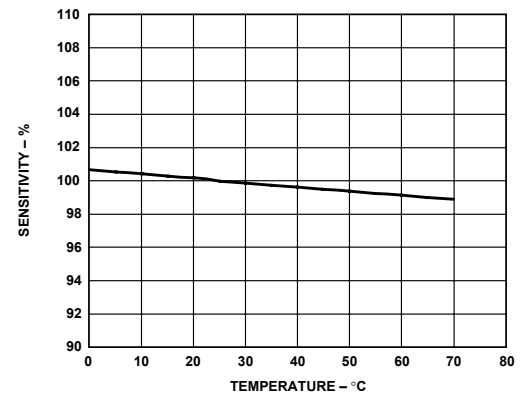


Figure 6. Normalized Sensitivity vs. Temperature

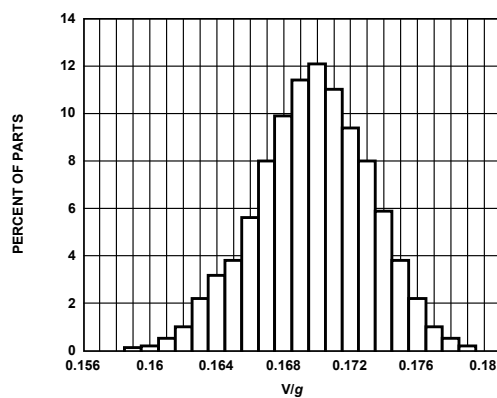


Figure 4. X-Axis Output Sensitivity Distribution at X_{OUT}

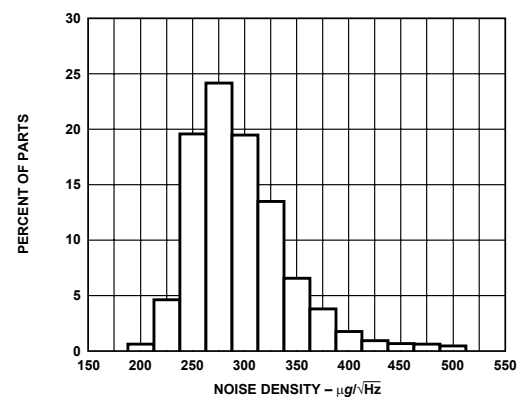


Figure 7. Noise Density Distribution

ADXL311

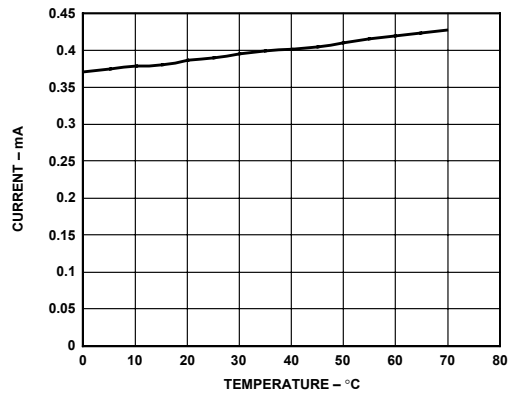


Figure 8. Typical Supply Current vs. Temperature

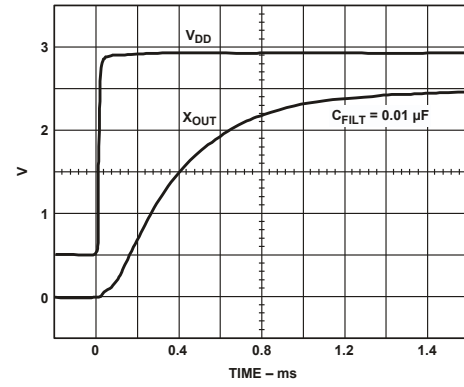


Figure 9. Typical Turn-On Time

THEORY OF OPERATION

The ADXL311 is a complete, dual-axis acceleration measurement system on a single monolithic IC. It contains a polysilicon surface-micromachined sensor and signal conditioning circuitry to implement an open-loop acceleration measurement architecture. The output signals are analog voltage proportional to acceleration. The ADXL311 is capable of measuring both positive and negative accelerations to at least $\pm 2g$. The accelerometer can measure static acceleration forces, such as gravity, allowing it to be used as a tilt sensor.

The sensor is a surface-micromachined polysilicon structure built on top of the silicon wafer. Polysilicon springs suspend the structure over the surface of the wafer and provide a resistance against acceleration forces. Deflection of the structure is measured using a differential capacitor that consists of independent fixed plates and central plates attached to the moving mass. The fixed plates are driven by 180° out of phase square waves. Acceleration will deflect the beam and unbalance the differential capacitor, resulting in an output square wave whose amplitude is proportional to acceleration. Phase sensitive demodulation techniques are then used to rectify the signal and determine the direction of the acceleration.

The output of the demodulator is amplified and brought off-chip through a $32\text{ k}\Omega$ resistor. At this point, the user can set the signal bandwidth of the device by adding a capacitor. This filtering improves measurement resolution and helps prevent aliasing.

Applications

POWER SUPPLY DECOUPLING

For most applications, a single $0.1\text{ }\mu\text{F}$ capacitor, CDC, will adequately decouple the accelerometer from noise on the power supply. However, in some cases, particularly where noise is present at the 100 kHz internal clock frequency (or any harmonic thereof), noise on the supply may cause interference on the ADXL311 output. If additional decoupling is needed, a $100\text{ }\Omega$ (or smaller) resistor or ferrite beads may be inserted in the supply line of the ADXL311. Additionally, a larger bulk bypass capacitor (in the $1\text{ }\mu\text{F}$ to $4.7\text{ }\mu\text{F}$ range) may be added in parallel to CDC.

SETTING THE BANDWIDTH USING C_X AND C_Y

The ADXL311 has provisions for bandlimiting the X_{OUT} and Y_{OUT} pins. Capacitors must be added at these pins to implement low-pass filtering for antialiasing and noise reduction. The equation for the 3 dB bandwidth is

$$F_{-3\text{dB}} = 1 / (2\pi(32\text{ k}\Omega) \times C_{(X,Y)})$$

or, more simply

$$F_{-3\text{dB}} = 5\text{ }\mu\text{F} / C_{(X,Y)}$$

The tolerance of the internal resistor (R_{FILT}) can vary typically as much as $\pm 15\%$ of its nominal value of $32\text{ k}\Omega$; thus, the bandwidth will vary accordingly. A minimum capacitance of 1000 pF for C_X and C_Y is required in all cases.

Table 4. Filter Capacitor Selection, C_X and C_Y

Bandwidth	Capacitor (μF)
10 Hz	0.47
50 Hz	0.10
100 Hz	0.05
200 Hz	0.027
500 Hz	0.01
5 kHz	0.001

SELF TEST

The ST pin controls the self-test feature. When this pin is set to V_{DD} , an electrostatic force is exerted on the beam of the accelerometer. The resulting movement of the beam allows the user to test if the accelerometer is functional. The typical change in output will be 270 mg (corresponding to 45 mV). This pin may be left open circuit or connected to common in normal use.

R_{BIAS} SELECTION

A bias resistor (R_{BIAS}) must always be used. If no resistor is present, the ADXL311 may appear to work but will suffer degraded noise performance. The value of the resistor used is not critical. Any value from $50\text{ k}\Omega$ to $2\text{ M}\Omega$ can be used. Using a $2\text{ M}\Omega$ resistor rather than a $50\text{ k}\Omega$ will save roughly $25\text{ }\mu\text{A}$ of supply current.

Design Trade-Offs for Selecting Filter Characteristics: The Noise/BW Trade-Off

The accelerometer bandwidth selected will ultimately determine the measurement resolution (smallest detectable acceleration). Filtering can be used to lower the noise floor, which improves the resolution of the accelerometer. Resolution is dependent on the analog filter bandwidth at X_{OUT} and Y_{OUT} .

The output of the ADXL311 has a typical bandwidth of 5 kHz . The user must filter the signal at this point to limit aliasing errors. The analog bandwidth must be no more than half the A/D sampling frequency to minimize aliasing. The analog bandwidth may be further decreased to reduce noise and improve resolution.

The ADXL311 noise has the characteristics of white Gaussian noise that contributes equally at all frequencies and is described in terms of $\mu\text{g}/\sqrt{\text{Hz}}$, i.e., the noise is proportional to the square

ADXL311

root of the bandwidth of the accelerometer. It is recommended that the user limit bandwidth to the lowest frequency needed by the application, to maximize the resolution and dynamic range of the accelerometer.

With the single pole roll-off characteristic, the typical noise of the ADXL202E is determined by

$$RMS\ NOISE = (300\ \mu g / \sqrt{Hz}) \times (\sqrt{BW} \times 1.6)$$

At 100 Hz the noise will be

$$RMS\ NOISE = (300\ \mu g / \sqrt{Hz}) \times (\sqrt{100} \times 1.6) = 3.8\ mg$$

Often the peak value of the noise is desired. Peak-to-peak noise can only be estimated by statistical methods. Table 5 is useful for estimating the probabilities of exceeding various peak values, given the rms value.

Table 5. Estimation of Peak-to-Peak Noise

Peak-to-Peak Value	% of Time That Noise Will Exceed Nominal Peak-to-Peak Value
2 × RMS	32
4 × RMS	4.6
6 × RMS	0.27
8 × RMS	0.006

The peak-to-peak noise value will give the best estimate of the uncertainty in a single measurement. Table 6 gives the typical noise output of the ADXL311 for various C_X and C_Y values.

Table 6. Filter Capacitor Selection (C_X , C_Y)

Bandwidth (Hz)	C_X , C_Y (μF)	RMS Noise (mg)	Peak-to-Peak Noise Estimate (mg)
10	0.47	1.2	7.2
50	0.1	2.7	16.2
100	0.047	3.8	22.8
500	0.01	8.5	51

USING THE ADXL311 WITH OPERATING VOLTAGES OTHER THAN 3 V

The ADXL311 is tested and specified at $V_{DD} = 3\ V$; however, it can be powered with V_{DD} as low as 2.7 V or as high as 5.25 V. Some performance parameters will change as the supply voltage is varied.

The ADXL311 output is ratiometric, so the output sensitivity (or scale factor) will vary proportionally to supply voltage. At $V_{DD} = 5\ V$ the output sensitivity is typically 312 mV/g.

The zero g bias output is also ratiometric, so the zero g output is nominally equal to $V_{DD}/2$ at all supply voltages.

The output noise is not ratiometric but absolute in volts; therefore, the noise density decreases as the supply voltage increases. This is because the scale factor (mV/g) increases while the noise voltage remains constant.

The self-test response is roughly proportional to the square of the supply voltage. At $V_{DD} = 5\ V$, the self-test response will be approximately equivalent to 800 mg (typical).

The supply current increases as the supply voltage increases. Typical current consumption at $V_{DD} = 5\ V$ is 600 μA.

Using the ADXL311 as a Dual-Axis Tilt Sensor

One of the most popular applications of the ADXL311 is tilt measurement. An accelerometer uses the force of gravity as an input vector to determine the orientation of an object in space.

An accelerometer is most sensitive to tilt when its sensitive axis is perpendicular to the force of gravity, i.e., parallel to the earth's surface. At this orientation, its sensitivity to changes in tilt is highest. When the accelerometer is oriented on axis to gravity, i.e., near its +1 g or -1 g reading, the change in output acceleration per degree of tilt is negligible. When the accelerometer is perpendicular to gravity, its output will change nearly 17.5 mg per degree of tilt, but at 45° degrees, it is changing only at 12.2 mg per degree and resolution declines.

DUAL-AXIS TILT SENSOR: CONVERTING ACCELERATION TO TILT

When the accelerometer is oriented so both its X-axis and Y-axis are parallel to the earth's surface, it can be used as a two axis tilt sensor with a roll axis and a pitch axis. Once the output signal from the accelerometer has been converted to an acceleration that varies between -1 g and +1 g , the output tilt in degrees is calculated as follows:

$$PITCH = \text{ASIN}(A_X / 1\ g)$$

$$ROLL = \text{ASIN}(A_Y / 1\ g)$$

Be sure to account for overranges. It is possible for the accelerometers to output a signal greater than ±1 g due to vibration, shock, or other accelerations.

Experiment Data

Table B.1: Frequency Sweep Data for Free-Moving Actuator Before Offsetting (03/08/07).

Freq. (Hz)	Frequency Response: Magnitude (dB) and Phase (deg)			
	Coil-Current	Position	Acceleration	Actuator Input
2	-47.07 \angle - 7.43	18.16 \angle 257	-35.51 \angle 77.00	-25.82 \angle - 8.149
3	-40.67 \angle - 4.31	21.51 \angle 224	-24.9 \angle 45.00	-19.28 \angle - 33.00
5	-31.80 \angle - 24.20	23.59 \angle 186.12	-14.12 \angle 7.00	-12.65 \angle - 61.27
7	-26.16 \angle - 41.70	3.70 \angle 158.50	-8.15 \angle - 21.60	-9.28 \angle - 78.30
10	-21.03 \angle - 66.20	22.60 \angle 127.00	-3.09 \angle - 52.90	-6.30 \angle - 95.89
15	-16.80 \angle - 97.30	19.80 \angle 90.90	1.27 \angle - 89.50	-3.66 \angle - 114.60
20	-14.90 \angle - 120.70	16.80 \angle 63.90	3.20 \angle - 116.00	-2.28 \angle - 126.80
35	-13.79 \angle - 162.60	8.12 \angle 15.51	4.27 \angle - 165.10	-0.69 \angle - 147.20
50	-14.50 \angle - 183.00	1.04 \angle - 8.60	3.3 \angle - 189.70	-0.29 \angle - 157.00
75	-16.25 \angle - 198.80	-8.00 \angle - 28.47	1.51 \angle - 210.20	-0.065 \angle - 164.50
100	-17.65 \angle - 205.70	-14.60 \angle - 39.39	-0.07 \angle - 221.70	-0.01 \angle - 168.40
150	-19.56 \angle - 210.80	-23.20 \angle - 49.90	-2.07 \angle - 233.46	0.04 \angle - 172.10
200	-20.76 \angle - 211.80	-29.00 \angle - 61.00	-3.30 \angle - 245.00	0.04 \angle - 174.10
350	-22.48 \angle - 212.70	-39.25 \angle - 111.30	-4.50 \angle - 302.00	0.065 \angle - 176.50
400	- - 22.76 \angle - 213.40	-57.00 \angle - 140.00	-16.11 \angle - 304.50	0.067 \angle - 176.80
500	-23.25 \angle - 214.70	-58.30 \angle - 92.00	-16.30 \angle - 262.29	0.078 \angle - 177.00
600	-23.58 \angle - 216.30	-57.80 \angle - 74.90	-12.56 \angle - 259.37	0.083 \angle - 177.80
700	-23.82 \angle - 218.30	-62.29 \angle - 123.00	-15.20 \angle - 278.70	0.083 \angle - 178.00
800	-24.02 \angle - 220.40	-65.47 \angle - 113.10	-14.43 \angle - 273.00	0.088 \angle - 178.20
900	-24.17 \angle - 222.80	-68.25 \angle - 124.00	-13.53 \angle - 279.90	0.092 \angle - 178.40
1000	-24.30 \angle - 225.30	-74.00 \angle - 149.00	-13.39 \angle - 314.20	0.096 \angle - 178.50
1200	-24.53 \angle - 230.70	-78.00 \angle - 74.00	-16.49 \angle - 281.00	0.1042 \angle - 178.60
1400	-24.70 \angle - 236.30	-72.00 \angle - 83.00	-13.81 \angle - 282.00	0.1212 \angle - 178.90
1800	-25.02 \angle - 248.20	-66.90 \angle 84.00	-9.58 \angle - 433.00	0.1307 \angle - 178.80
2000	-25.18 \angle 105.70	-78.8 \angle 80.21	-28.16 \angle - 383.10	0.129 \angle - 178.90

Table B.2: Frequency Sweep Data for Clamped Actuator Before Offsetting (06/08/07).

	Frequency Response: Magnitude (dB) and Phase (deg)	
Freq. (Hz)	Coil-Current	Actuator Input
2	$-42.68 \angle 26.65$	$-31.22 \angle 26.97$
3	$-34.10 \angle 2.16$	$-22.30 \angle 33.517$
5	$-25.29 \angle -35.44$	$-13.30 \angle -32.21$
7	$-20.90 \angle -62.35$	$-8.85 \angle -57.41$
10	$-17.54 \angle -90.66$	$-5.37 \angle -83.31$
15	$-15.18 \angle -120.50$	$-2.792 \angle -109.40$
20	$-14.31 \angle -139.50$	$-1.66 \angle -124.90$
35	$-14.08 \angle -170.10$	$-0.55 \angle -147.30$
50	$-14.75 \angle -185.40$	$-0.26 \angle -156.90$
75	$-16.05 \angle -198.50$	$-0.104 \angle -164.40$
100	$-17.19 \angle -205.10$	$-0.04 \angle -168.20$
150	$-18.91 \angle -210.70$	$0.002 \angle -172.00$
200	$-20.06 \angle -212.60$	$0.03 \angle -173.90$
350	$-21.90 \angle -214.50$	$0.05 \angle -176.40$
400	$-22.23 \angle -215.00$	$0.056 \angle -176.80$
500	$-22.78 \angle -216.20$	$0.059 \angle -177.40$
600	$-23.14 \angle -217.90$	$0.071 \angle -177.70$
700	$-23.41 \angle -219.80$	$0.073 \angle -178.00$
800	$-23.62 \angle -221.90$	$0.082 \angle -178.10$
900	$-23.81 \angle -224.30$	$0.084 \angle -178.30$
1000	$-23.93 \angle -226.70$	$0.1045 \angle -178.40$
1200	$-24.20 \angle -232.00$	$0.1028 \angle -178.60$
1400	$-24.41 \angle -237.50$	$0.1028 \angle -178.80$
1800	$-24.78 \angle -249.30$	$0.1313 \angle -178.90$
2000	$-24.88 \angle -255.40$	$0.1671 \angle -179.00$

Table B.3: Measurement Offsets.

	Offset
Current:	
Input to PCI cards has a 0.2Ω resistance.	+13.98 dB
Actuator Input Voltage:	
Amplifier has a factor of 1/11.	+20.83 dB
Position Encoder:	
$0.5\mu m$ per count, plus factor of 0.01 in Simulink setup.	+66.02 dB
Accelerometer:	
0.528 V per 2 g or 26.91 mV per $\frac{m}{s^2}$.	+31.4 dB

Parameter Identification

Algorithm B.1 Parameter identification Matlab code

```

%------%
%   Parameter Identification
%------%
% solve identification criteria, with initial parameter estimates X0
fsolve(@ident_criteria,[X0])
%------%
function [c] = ident_criteria(p)
% Function creates a criteria c which evaluates to zero when the frequency
% response of the model matches that of the experiment data
%
% This function is for matching to the clamped coil voltage-coil current
% response. The input are initial values for [L1 Lm R23 R1]
% Define system parameters - assign value or parameter number
L1=p(1);    %Determine L1
Lm=p(2);    %Determine Lm
R23=p(3);   %Determine R2|3
R1=p(4);    %Determine R1
% Define system volt-coil current
sys=tf([Lm R23],[Lm*L1 (Lm*(R23+R1)+L1*R23) R1*R23]);
% Experiment Frequency Response Data
freq=[2 3 5 7 10 15 20 35 50 75 100 150 200 350 400 500 600 700 800 900 ...
      1000 1200 1400 1800 2000]';
magcurc=[-18.3100 -18.6200 -18.8400 -18.9000 -19.0200 -19.2380...
          -19.5000 -20.3801 -21.3400 -22.7960 -24.0000 -25.7620 -26.9400...
          -28.8000 -29.1360 -29.6890 -30.0610 -30.3330 -30.5520 -30.7440...
          -30.8845 -31.1528 -31.3628 -31.7613 -31.8971]';
phascurc=[-0.3200 -1.3530 -3.2300 -4.9400 -7.3500 -11.1000...
          -14.6000 -22.8000 -28.5000 -34.1000 -36.9000 -38.7000 -38.7000...
          -38.1000 -38.2000 -38.8000 -40.2000 -41.8000 -43.8000 -46.0000...
          -48.3000 -53.4000 -58.7000 -70.4000 -76.4000]';
% Model response
sys=frd(sys,freq,'Units','Hz');
response=squeeze(sys.response);
% Difference between experimental response and model response
damp = magcurc - 20.*log10(abs(response));
dph = phascurc - 180./ pi .* phase(response);
% Frequency Weights
w = [ ones(3,1) .* 0.1; ones(9,1) ; ones(3,1).*0.1 ; ones(10,1)*0.001];
% Criteria - minimise difference
c = sum(abs(damp.*w).^2)*10 + sum (abs(dph.*w).^2);

```

Appendix C

Example Models

Using the element models derived in the Chapter 3, and the methodology for modelling HRA configurations also presented there, models of the three application examples described in Chapter 4 are provided here. The models provided use the reduced order LTI element model, as this produces the overall model used in the control design.

4×4 Parallel-in-Series HRA Model

The 4×4 PS HRA is arranged as shown in Figure C.1. The system has 4 moving masses, hence the reduced order cut-bobbin model, stated in equation C.1 has 8 states.

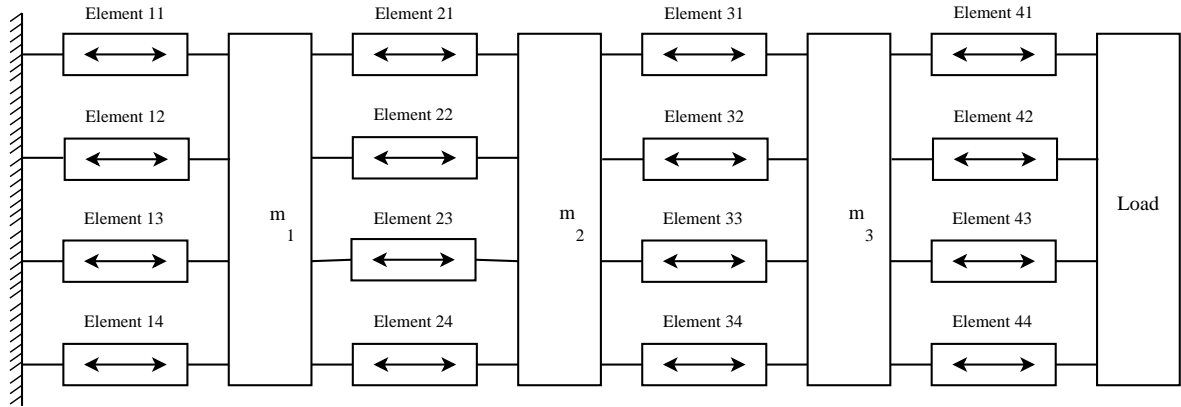


Figure C.1: 4×4 PS HRA

$$\begin{aligned}
 \begin{bmatrix} \ddot{x}_1 \\ \dot{x}_1 \\ \ddot{x}_2 \\ \dot{x}_2 \\ \ddot{x}_3 \\ \dot{x}_3 \\ \ddot{x}_4 \\ \dot{x}_4 \end{bmatrix} &= \begin{bmatrix} -\frac{(F_{a1}+A_{a2})}{m_1} & -\frac{(F_{b1}+A_{b2})}{m_1} & \frac{F_{a2}}{m_1} & \frac{F_{b2}}{m_1} & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{F_{a2}}{m_2} & \frac{F_{b2}}{m_2} & -\frac{(F_{a2}+F_{a3})}{m_2} & -\frac{(A_{b2}+A_{b3})}{m_2} & \frac{F_{a3}}{m_2} & \frac{F_{b3}}{m_2} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{F_{a3}}{m_3} & \frac{F_{b3}}{m_3} & -\frac{(F_{a3}+F_{a4})}{m_3} & -\frac{(F_{b3}+F_{b4})}{m_3} & \frac{F_{a4}}{m_3} & \frac{F_{b4}}{m_3} \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{F_{a4}}{m_4} & \frac{F_{b4}}{m_4} & -\frac{F_{a4}}{m_4} & -\frac{F_{b4}}{m_4} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \cdots \\
 &\bullet \begin{bmatrix} \dot{x}_1 \\ x_1 \\ \dot{x}_2 \\ x_2 \\ \dot{x}_3 \\ x_3 \\ \dot{x}_4 \\ x_4 \end{bmatrix} + \begin{bmatrix} \frac{U_1}{m_1} & -\frac{U_2}{m_1} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{U_2}{m_2} & -\frac{U_3}{m_2} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{U_3}{m_3} & -\frac{U_4}{m_3} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{U_4}{m_4} \\ 0 & 0 & 0 & 0 \end{bmatrix} \bullet \begin{bmatrix} u_1' \\ u_2' \\ u_3' \\ u_4' \end{bmatrix}
 \end{aligned} \tag{C.1}$$

where:

$$F_{ai} = \sum_{n=1:4} \left(\frac{(k_{i,n}^2 + d_{i,n} R_{i,n})}{R_{i,n}} \right), \quad F_{bi} = \sum_{n=1:4} r_{i,n}$$

$$U_i = \begin{bmatrix} \frac{k_{i1}}{R_{i1}} & \frac{k_{i2}}{R_{i2}} & \frac{k_{i3}}{R_{i3}} & \frac{k_{i4}}{R_{i4}} \end{bmatrix}, \quad u_i' = \begin{bmatrix} u_{i1} & u_{i2} & u_{i3} & u_{i4} \end{bmatrix}'$$

In this example the values of m_1 to m_3 are four times the usual element mass (0.52 kg) and m_4 is 2 kg. The other parameters are equal to those stated in Chapter 3.

4 × 4 Series-in-Parallel HRA Model

Figure C.2 depicts the 4x4 SP HRA. This system has many more moving masses, in comparison to the PS 4x4 HRA, and hence the model has more states (26 in all).

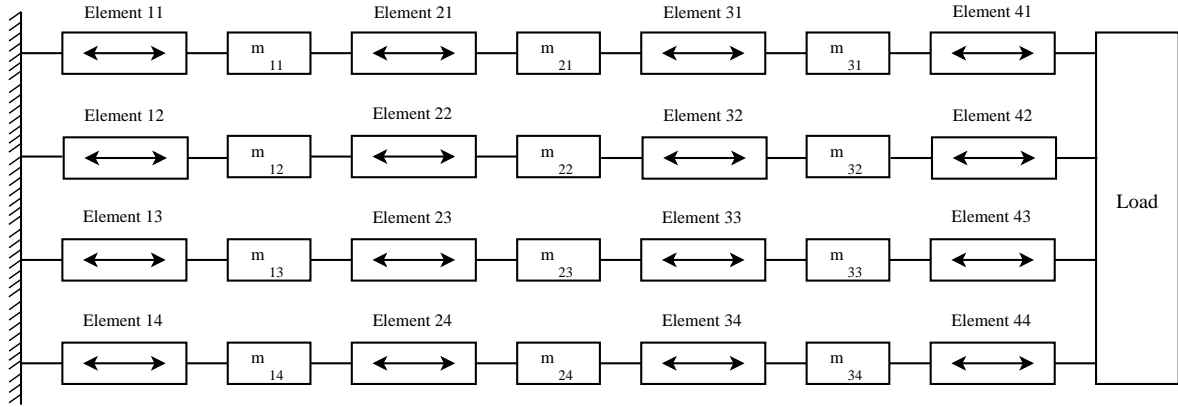


Figure C.2: 4x4 SP HRA

$$\begin{bmatrix} \dot{c}_1 \\ \dot{c}_2 \\ \dot{c}_3 \\ \ddot{x}_4 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} A_{b1} & A_{c1} & 0 & 0 & 0 \\ A_{a2} & A_{b2} & A_{c2} & 0 & 0 \\ 0 & A_{a3} & A_{b3} & I(A_{c3}) & 0 \\ 0 & 0 & A_4 & -\frac{\sum F_{a1:4}}{m_4} & -\frac{\sum F_{b1:4}}{m_4} \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \dot{x}_4 \\ x_4 \end{bmatrix} \dots$$

$$+ \begin{bmatrix} U_1^1 & -U_2^1 & 0 & 0 \\ 0 & U_2^2 & -U_3^2 & 0 \\ 0 & 0 & U_3^3 & -U_4^3 \\ 0 & 0 & 0 & U_4^4 \\ 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} u_1' \\ u_2' \\ u_3' \\ u_4' \end{bmatrix}$$

where:

$$c_n = \begin{bmatrix} \dot{x}_{n1} & x_{n1} & \dot{x}_{n2} & x_{n2} & \dot{x}_{n3} & x_{n3} & \dot{x}_{n4} & x_{n4} \end{bmatrix}'$$

$$\begin{aligned}
 A_{an} &= \begin{bmatrix} A_{an1} & 0 & 0 & 0 \\ 0 & A_{an2} & 0 & 0 \\ 0 & 0 & A_{an3} & 0 \\ 0 & 0 & 0 & A_{an4} \end{bmatrix}, \quad A_{ani} = \begin{bmatrix} \frac{F_{ani}}{m_{ni}} & \frac{F_{bni}}{m_{ni}} \\ 0 & 0 \end{bmatrix} \\
 A_{bn} &= \begin{bmatrix} A_{bn1} & 0 & 0 & 0 \\ 0 & A_{bn2} & 0 & 0 \\ 0 & 0 & A_{bn3} & 0 \\ 0 & 0 & 0 & A_{bn4} \end{bmatrix}, \quad A_{bni} = \begin{bmatrix} -\frac{(F_{ani}+F_{ani+1})}{m_{ni}} & -\frac{(F_{bni}+F_{bni+1})}{m_{ni}} \\ 1 & 0 \end{bmatrix} \\
 A_{cn} &= \begin{bmatrix} A_{cn1} & 0 & 0 & 0 \\ 0 & A_{cn2} & 0 & 0 \\ 0 & 0 & A_{cn3} & 0 \\ 0 & 0 & 0 & A_{cn4} \end{bmatrix}, \quad A_{ani} = \begin{bmatrix} \frac{F_{ani+1}}{m_{11}} & \frac{F_{bni+1}}{m_{11}} \\ 0 & 0 \end{bmatrix} \\
 A_4 &= \begin{bmatrix} \frac{F_{a41}}{m_4} & \frac{F_{b41}}{m_4} & \frac{F_{a43}}{m_4} & \frac{F_{b42}}{m_4} & \frac{F_{a43}}{m_4} & \frac{F_{b43}}{m_4} & \frac{F_{a44}}{m_4} & \frac{F_{b44}}{m_4} \end{bmatrix} \\
 U_n^p &= \begin{bmatrix} \frac{k_{n1}}{R_{n1}m_{pi}} & 0 & 0 & 0 \\ 0 & \frac{k_{n2}}{R_{n2}m_{pi}} & 0 & 0 \\ 0 & 0 & \frac{k_{n3}}{R_{n3}m_{pi}} & 0 \\ 0 & 0 & 0 & \frac{k_{n3}}{R_{n3}m_{pi}} \end{bmatrix}, \quad U_4 = \begin{bmatrix} \frac{k_{41}}{R_{41}m_4} & \frac{k_{42}}{R_{42}m_4} & \frac{k_{43}}{R_{43}m_4} & \frac{k_{44}}{R_{44}m_4} \end{bmatrix}
 \end{aligned}$$

In this SP configuration m_1 to m_3 represent the mass of each element, and as such are equal to the single element mass stated in Chapter 3 (0.13 kg). As in the PS example, the load mass is 2 kg and the other parameters are equal to those stated in Chapter 3.

10 × 10 Parallel-in-Series HRA Model

The structure of the 10 × 10 Parallel in Series HRA is the same as the 4 × 4 PS system (Figure C.1) but with 10 serial banks of 10 parallel elements. A state-space description is given below.

$$\begin{bmatrix} \dot{c}_1 \\ \dot{c}_2 \\ \vdots \\ \dot{c}_{10} \end{bmatrix} = \mathbf{A} \bullet \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{10} \end{bmatrix} + \mathbf{B} \bullet \begin{bmatrix} u'_1 \\ u'_2 \\ \vdots \\ u'_{10} \end{bmatrix} \quad (\text{C.2})$$

$$\mathbf{A} = \begin{bmatrix} A_{b1} & A_{c1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ A_{a2} & A_{b2} & A_{c2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & A_{a3} & A_{b3} & A_{c3} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & A_{a4} & A_{b4} & A_{c4} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & A_{a5} & A_{b5} & A_{c5} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & A_{a6} & A_{b6} & A_{c6} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & A_{a7} & A_{b7} & A_{c7} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & A_{a8} & A_{b8} & A_{c8} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_{a9} & A_{b9} & A_{c9} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_{a10} & A_{b10} \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} U_1^1 & -U_2^1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & U_2^2 & -U_3^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & U_3^3 & -U_4^3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & U_4^4 & -U_5^4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & U_5^5 & -U_6^5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & U_6^6 & -U_7^6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & U_7^7 & -U_8^7 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & U_8^8 & -U_9^8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & U_9^9 & -U_{10}^9 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & U_{10}^{10} \end{bmatrix}$$

where:

$$\mathbf{c}_n = \begin{bmatrix} \dot{\mathbf{x}}_n & \mathbf{x}_n \end{bmatrix}', \quad A_{an} = \begin{bmatrix} \frac{F_{an}}{m_n} & \frac{F_{bn}}{m_n} \\ 0 & 0 \end{bmatrix},$$

$$A_{bn} = \begin{bmatrix} -\frac{(F_{an} + A_{an+1})}{m_n} & -\frac{(F_{bn} + A_{bn+1})}{m_n} \\ 1 & 0 \end{bmatrix}, \quad A_{cn} = \begin{bmatrix} \frac{F_{an+1}}{m_n} & \frac{F_{bn+1}}{m_n} \\ 0 & 0 \end{bmatrix},$$

$$U_i = \begin{bmatrix} \frac{k_{i1}}{R_{i1}} & \frac{k_{i2}}{R_{i2}} & \cdots & \frac{k_{i10}}{R_{i10}} \end{bmatrix}, \quad \mathbf{u}_i' = \begin{bmatrix} u_{i1} & u_{i2} & \cdots & u_{i10} \end{bmatrix}'$$

The values of m_1 to m_9 in this example are ten times that of the usual element mass (1.3 kg) and the load mass m_{10} is 6 kg. The other parameters are equal to those stated in Chapter 3.

Appendix D

Multi-Agent Control

Fixed Outer-Loop Control Values

The control values of the PI controllers (which take the form shown in equation D.1) for the three example systems are given in Table D.1.

$$C_{pi} = G_{pi} \frac{\tau_{pi}s + 1}{s} \quad (D.1)$$

Table D.1: Fixed PI control gains for MAC of HRA examples

System	G_{pi}	τ_{pi}
PS 4×4 HRA	5.5	0.65
SP 4×4 HRA	5.5	0.65
PS 10×10 HRA	3.0	3.3

Look-up Table Control Parameters

The look-up tables containing the control parameters of the inner loop multi-agent control (Figure D.1) are provided in Tables D.2 to D.4.

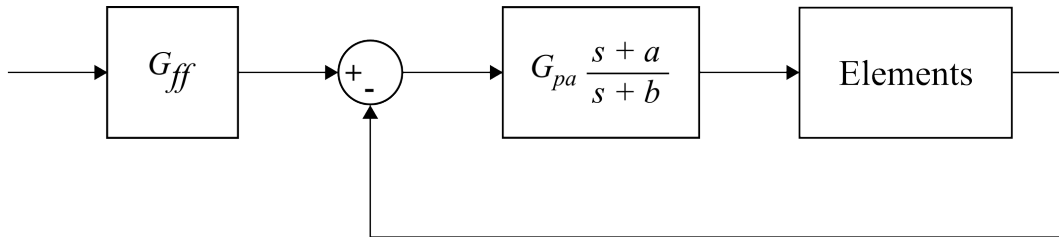


Figure D.1: Inner-loop multi-agent control.

Table D.2: Look-up table of control parameters for MAC of PS 4×4 HRA.

Perception State		Control Parameters			
Loose faults	Lock-up faults	G_{ff}	G_{pa}	τ_{pa}	α_{pa}
0	0	1/4	11	0.038	0.456
1	0	1/4	11	0.037	0.47
2	0	1/4	11	0.037	0.525
3	0	1/4	11	0.037	0.562
4	0	1/4	11	0.037	0.637
5	0	1/4	11	0.037	0.68
6	0	1/4	11	0.037	0.76
7	0	1/4	11	0.037	0.81
8	0	1/4	11	0.037	0.91
0	1	1/3	11	0.028	0.336
0	2	1/2	11	0.019	0.228
1	1	1/3	11	0.03	0.36
2	1	1/3	11	0.04	0.48
3	1	1/3	11	0.043	0.52
4	1	1/3	11	0.05	0.60
5	1	1/3	11	0.05	0.60
6	1	1/3	11	0.06	0.72
1	2	1/2	11	0.023	0.28
2	2	1/2	11	0.03	0.36
3	2	1/2	11	0.035	0.42
4	2	1/2	11	0.042	0.50

Table D.3: Look-up table of control parameters for MAC of SP 4×4 HRA.

Perception State		Control Parameters			
Loose faults	Lock-up faults	G_{ff}	G_{pa}	τ_{pa}	α_{pa}
0	0	1/4	11	0.038	0.46
1	0	1/4	11	0.048	0.57
2	0	1/4	11	0.077	0.92
0	1	1/3	11	0.034	0.41
0	2	1/3	11	0.030	0.36
0	3	1/3	11	0.028	0.34
0	4	1/3	11	0.026	0.31
0	5	1/2	11	0.024	0.29
0	6	1/2	11	0.023	0.28
0	7	1/2	11	0.021	0.25
0	8	1/2	11	0.020	0.24
1	1	1/3	11	0.044	0.53
1	2	1/3	11	0.036	0.43
1	3	1/2	11	0.034	0.41
1	4	1/2	11	0.032	0.38
1	5	1/2	11	0.027	0.32
1	6	1/2	11	0.024	0.29
2	1	1/3	11	0.066	0.79
2	2	1/3	11	0.060	0.71
2	3	1/2	11	0.050	0.59
2	4	1/2	11	0.042	0.50

Table D.4: Look-up table of control parameters for MAC of PS 10×10 HRA.

Perception State		Control Parameters			
Loose faults	Lock-up faults	G_{ff}	G_{pa}	τ_{pa}	α_{pa}
0	0	1/10	2.5	0.14	1.68
0	1	1/9	2.5	0.13	1.50
0	2	1/8	2.5	0.11	1.32
0	3	1/6	2.5	0.096	1.15
0	4	1/7	2.5	0.078	0.94
1-4	0	1/10	2.5	0.14	1.77
5-8	0	1/10	2.5	0.13	1.90
9-12	0	1/10	2.5	0.12	1.94
13-16	0	1/10	2.5	0.11	2.04
17-20	0	1/10	2.5	0.10	2.14
21-24	0	1/10	2.5	0.11	2.25
25-28	0	1/10	2.5	0.12	2.15
29-32	0	1/10	2.5	0.14	2.49
33-36	0	1/10	2.5	0.14	2.60
37-40	0	1/10	2.5	0.14	2.84
1-4	1	1/9	2.5	0.14	1.67
5-8	1	1/9	2.5	0.13	1.70
9-12	1	1/9	2.5	0.12	1.70
13-16	1	1/9	2.5	0.11	1.75
17-20	1	1/9	2.5	0.10	1.90
21-24	1	1/9	2.5	0.11	1.90
25-28	1	1/9	2.5	0.12	2.10
29-32	1	1/9	2.5	0.14	2.25
33-36	1	1/9	2.5	0.14	2.40
1-4	2	1/8	2.5	0.13	1.35
5-8	2	1/8	2.5	0.12	1.41
9-12	2	1/8	2.5	0.11	1.54
13-16	2	1/8	2.5	0.1	1.60
17-20	2	1/8	2.5	0.11	1.75
21-24	2	1/8	2.5	0.12	1.80
25-28	2	1/8	2.5	0.14	1.95
29-32	2	1/8	2.5	0.14	2.10
1-4	3	1/7	2.5	0.12	1.15
5-8	3	1/7	2.5	0.11	1.30
9-12	3	1/7	2.5	0.10	1.54
13-16	3	1/7	2.5	0.11	1.60
17-20	3	1/7	2.5	0.12	1.75
21-24	3	1/7	2.5	0.14	1.80
25-28	3	1/7	2.5	0.14	1.95
1-4	4	1/6	2.5	0.11	1.43
5-8	4	1/6	2.5	0.10	1.54
9-12	4	1/6	2.5	0.11	1.72
13-16	4	1/6	2.5	0.12	1.89
17-20	4	1/6	2.5	0.14	2.04
21-24	4	1/6	2.5	0.14	2.17

Appendix E

Software Demonstrator

Animation Coding

```
function [sys,x0,str,ts] = HRAAnimation_4x4PS(t,x,u,flag,pausetime)
%=====
%
% 4x4 HRA Animation S-function.
% Jessica Davies, Loughborough University 01/02/2009
%
%=====
switch flag,
%=====
% Initialization %
%=====
    case 0,
        [sys,x0,str,ts]=mdlInitializeSizes;

%=====
% Update %
%=====
    case 2,
        sys=mdlUpdate(t,x,u,pausetime);

%=====
% Not use %
%=====
    case { 1, 3, 4, 9 },
        sys = [];

%=====
% Unexpected flags %
%=====
    otherwise
        error(['Unhandled flag = ',num2str(flag)]);
end

%=====
```

```
% mdlInitializeSizes
% Return the sizes, initial conditions, and sample times for the S-function.
%=====

function [sys,x0,str,ts]=mdlInitializeSizes
sizes = simsizes;
sizes.NumContStates = 0;
sizes.NumDiscStates = 0;
sizes.NumOutputs = 0;
sizes.NumInputs = 8;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1;
sys = simsizes(sizes);

%
% initialize the initial conditions
%
x0 = [];

%
% str is always an empty matrix
%
str = [];

%
% initialize the array of sample times, for the pendulum demo,
% the animation is updated every 0.1 seconds
%
ts = [0.1 0];

%
% create the figure, if necessary
%
LocalHRAInit;

% end mdlInitializeSizes

%=====
% mdlUpdate
% Update the animation.
%=====
function sys=mdlUpdate(t,x,u,pausetime)
Fig = get_param(gcbh,'UserData');
if ishandle(Fig),
    if strcmp(get(Fig,'Visible'),'on'),
        ud = get(Fig,'UserData');
        LocalHRASets(t,ud,u,pausetime);
    end
end;
sys = [];
% end mdlUpdate
```

```

%=====
% LocalHRASets
%=====
function LocalHRASets(time,ud,u,pausetime)

u      = u([1 2 3 4 5 6 7 8]);
xA     = u(1);
xB     = u(2);
xC     = u(3);
xD     = u(4);
TxA    = makehgtform('translate',[xA 0 0]);
TxB    = makehgtform('translate',[xB 0 0]);
TxC    = makehgtform('translate',[xC 0 0]);
TxD    = makehgtform('translate',[xD 0 0]);

ElementLetter=['A' 'B' 'C' 'D'];
for i=5:8;
    if i==5;
        if u(i)==4; %All elements healthy
            set(ud.GndElement1,'EdgeColor','k','linewidth',1)
            set(ud.GndElement2,'EdgeColor','k','linewidth',1)
            set(ud.GndElement3,'EdgeColor','k','linewidth',1)
            set(ud.GndElement4,'EdgeColor','k','linewidth',1)
        elseif u(i)==3; %One element loose
            set(ud.GndElement1,'EdgeColor',[1 0.8 0.2],'linewidth',2)
            set(ud.GndElement2,'EdgeColor','k','linewidth',1)
            set(ud.GndElement3,'EdgeColor','k','linewidth',1)
            set(ud.GndElement4,'EdgeColor','k','linewidth',1)
        elseif u(i)==2; %Two elements loose
            set(ud.GndElement1,'EdgeColor',[1 0.8 0.2],'linewidth',2)
            set(ud.GndElement2,'EdgeColor',[1 0.8 0.2],'linewidth',2)
            set(ud.GndElement3,'EdgeColor','k','linewidth',1)
            set(ud.GndElement4,'EdgeColor','k','linewidth',1)
        elseif u(i)==1; %Three elements loose
            set(ud.GndElement1,'EdgeColor',[1 0.8 0.2],'linewidth',2)
            set(ud.GndElement2,'EdgeColor',[1 0.8 0.2],'linewidth',2)
            set(ud.GndElement3,'EdgeColor',[1 0.8 0.2],'linewidth',2)
            set(ud.GndElement4,'EdgeColor','k','linewidth',1)
        elseif u(i)==0; %Four elements loose
            set(ud.GndElement1,'EdgeColor',[1 0.8 0.2],'linewidth',2)
            set(ud.GndElement2,'EdgeColor',[1 0.8 0.2],'linewidth',2)
            set(ud.GndElement3,'EdgeColor',[1 0.8 0.2],'linewidth',2)
            set(ud.GndElement4,'EdgeColor',[1 0.8 0.2],'linewidth',2)
        elseif u(i)==-1; %Elements locked
            set(ud.GndElement1,'EdgeColor','r','linewidth',2)
            set(ud.GndElement2,'EdgeColor','r','linewidth',2)
            set(ud.GndElement3,'EdgeColor','r','linewidth',2)
            set(ud.GndElement4,'EdgeColor','r','linewidth',2)
        end
    else

```

```

if u(i)==4; %All elements healthy
    eval(['set(ud.Case' ElementLetter(i-5) '1','...
        ''EdgeColor'', 'k', ''linewidth'',1)'])
    eval(['set(ud.Case' ElementLetter(i-5) '2','...
        ''EdgeColor'', 'k', ''linewidth'',1)'])
    eval(['set(ud.Case' ElementLetter(i-5) '3','...
        ''EdgeColor'', 'k', ''linewidth'',1)'])
    eval(['set(ud.Case' ElementLetter(i-5) '4','...
        ''EdgeColor'', 'k', ''linewidth'',1)'])
elseif u(i)==3; %One element loose
    eval(['set(ud.Case' ElementLetter(i-5) '1','...
        ''EdgeColor'', [1 0.8 0.2], ''linewidth'',2)'])
    eval(['set(ud.Case' ElementLetter(i-5) '2','...
        ''EdgeColor'', 'k', ''linewidth'',1)'])
    eval(['set(ud.Case' ElementLetter(i-5) '3','...
        ''EdgeColor'', 'k', ''linewidth'',1)'])
    eval(['set(ud.Case' ElementLetter(i-5) '4','...
        ''EdgeColor'', 'k', ''linewidth'',1)'])
elseif u(i)==2; %Two elements loose
    eval(['set(ud.Case' ElementLetter(i-5) '1','...
        ''EdgeColor'', [1 0.8 0.2], ''linewidth'',2)'])
    eval(['set(ud.Case' ElementLetter(i-5) '2','...
        ''EdgeColor'', [1 0.8 0.2], ''linewidth'',2)'])
    eval(['set(ud.Case' ElementLetter(i-5) '3','...
        ''EdgeColor'', 'k', ''linewidth'',1)'])
    eval(['set(ud.Case' ElementLetter(i-5) '4','...
        ''EdgeColor'', 'k', ''linewidth'',1)'])
elseif u(i)==1; %Three elements loose
    eval(['set(ud.Case' ElementLetter(i-5) '1','...
        ''EdgeColor'', [1 0.8 0.2], ''linewidth'',2)'])
    eval(['set(ud.Case' ElementLetter(i-5) '2','...
        ''EdgeColor'', [1 0.8 0.2], ''linewidth'',2)'])
    eval(['set(ud.Case' ElementLetter(i-5) '3','...
        ''EdgeColor'', [1 0.8 0.2], ''linewidth'',2)'])
    eval(['set(ud.Case' ElementLetter(i-5) '4','...
        ''EdgeColor'', 'k', ''linewidth'',1)'])
elseif u(i)==0; %Four elements loose
    eval(['set(ud.Case' ElementLetter(i-5) '1','...
        ''EdgeColor'', [1 0.8 0.2], ''linewidth'',2)'])
    eval(['set(ud.Case' ElementLetter(i-5) '2','...
        ''EdgeColor'', [1 0.8 0.2], ''linewidth'',2)'])
    eval(['set(ud.Case' ElementLetter(i-5) '3','...
        ''EdgeColor'', [1 0.8 0.2], ''linewidth'',2)'])
    eval(['set(ud.Case' ElementLetter(i-5) '4','...
        ''EdgeColor'', [1 0.8 0.2], ''linewidth'',2)'])
elseif u(i)==-1; %Elements locked
    eval(['set(ud.Case' ElementLetter(i-5) '1','...
        ''EdgeColor'', 'r', ''linewidth'',2)'])
    eval(['set(ud.Case' ElementLetter(i-5) '2','...
        ''EdgeColor'', 'r', ''linewidth'',2)'])
    eval(['set(ud.Case' ElementLetter(i-5) '3','...

```

```

        ''EdgeColor'', ''r'', ''linewidth'', 2)']])
eval(['set(ud.Case' ElementLetter(i-5) '4,'...
        ''EdgeColor'', ''r'', ''linewidth'', 2)']])
    end

end

end

set(ud.ElementA,...
    'Matrix',TxA);
set(ud.ElementB,...
    'Matrix',TxB);
set(ud.ElementC,...
    'Matrix',TxC);
set(ud.ElementD,...
    'Matrix',TxD);
set(ud.TimeField,...
    'String',num2str(time));

% Force plot to be drawn
pause(pausetime);
drawnow;
% end LocalHRASets

%
%=====
% LocalHRAInit
% Local function to initialize the animation. If the animation window already
% exists, it is brought to the front. Otherwise, a new figure window is
% created.
%=====
function LocalHRAInit
sys = get_param(gcs,'Parent');
TimeClock = 0;
RefSignal = 0;

Tx0      =makehgtform('translate',[0 0 0]);
xA       = 0;
xB       = 0;
xC       = 0;
xD       = 0;
TxA      =makehgtform('translate',[xA 0 0]);
TxB      =makehgtform('translate',[xB 0 0]);
TxC      =makehgtform('translate',[xC 0 0]);
TxD      =makehgtform('translate',[xD 0 0]);

```

```
% The animation figure handle is stored in the HRA block's UserData.
% If it exists, initialize the elements
```

```
Fig = get_param(gcbbh,'UserData');
if ishandle(Fig),
    FigUD = get(Fig,'UserData');
    set(FigUD.TimeField,...
        'String',num2str(TimeClock));
    set(FigUD.ElementA,...
        'Matrix',TxA);
    set(FigUD.ElementB,...
        'Matrix',TxB);
    set(FigUD.ElementC,...
        'Matrix',TxC);
    set(FigUD.ElementD,...
        'Matrix',TxD);
    set(FigUD.Ground,...
        'Matrix',Tx0)
```

```
% bring it to the front
figure(Fig);
return
end
```

```
% the animation figure doesn't exist, create a new one and store its
% handle in the animation block's UserData
```

```
FigureName = 'HRA_Visualisation';
Fig = figure(...
    'Units',          'pixel',...
    'Name',           FigureName,...
    'NumberTitle',    'off',...
    'IntegerHandle',  'off',...
    'Resize',         'off',...
    'Position',       [2      33    711    415]);%[25 25 560 420]);
```

```
AxesH = axes(...
    'Parent', Fig,...
    'Units',   'pixel',...
    'xtick',   [],...
    'xticklabel', '',...
    'ytick',   [],...
    'yticklabel', '',...
    'visible', 'on',...
    'box',     'on',...
    'color',   [0.9608    0.9765    0.9922]);
axis([-5 30 -5 5]*0.011)
```

```
%Set Ground Elements
GndOrigin=-5.5;
```

```

Ground=hgtransform;

Gnd=line([-1.5 -1.5]*0.011,[-3.5 3.5]*0.011,...
    'Parent',Ground,'erasemode','none','linewidth',2);
for i=1:4,
    eval(['GndElement' num2str(i) '=rectangle(''Position'',...
        '[-1_GndOrigin+i*2_2_1]*0.011,...
        ''Parent'',Ground,''erasemode'', ''none'', ''linewidth'',1.2);']);
    eval(['GndElementConnet' num2str(i) '=...
        'line([-1.5_1]*0.011,[-5+i*2_-5+i*2]*0.011,...
        ''Parent'',Ground,''erasemode'', ''none'', ''linewidth'',1.2);']);
end

% Moving Elements A
ElementA=hgtransform;
set(ElementA,'Parent',AxesH)
Aorigin=0;

%Set up elements, cases and connections for branch A
for i=1:4,
    eval(['ElementABar' num2str(i) '=...
        'line([Aorigin_Aorigin]*0.011,[-5.5+i*2_-4.5+i*2]*0.011,...
        ''Parent'',ElementA,''erasemode'', ''none'', ''linewidth'',1.2);']);
    eval(['ElementAConnect' num2str(i) '=...
        'line([Aorigin_Aorigin+2.5]*0.011,[-5+i*2_-5+i*2]*0.011,...
        ''Parent'',ElementA,''erasemode'', ''none'', ''linewidth'',1.2);']);
    eval(['CaseA' num2str(i) '=...
        'rectangle(''Position'',_1[Aorigin+5_-5.5+i*2_2_1]*0.011,...
        ''Parent'',ElementA,''erasemode'', ''none'', ''linewidth'',1.2);']);
    eval(['CaseAConnect' num2str(i) '=...
        'line([Aorigin+4.5_Aorigin+5]*0.011,[-5+i*2_-5+i*2]*0.011,...
        ''Parent'',ElementA,''erasemode'', ''none'', ''linewidth'',1.2);']);
end

%Set Mass A
MassA=rectangle('Position', [Aorigin+2.5 -3.5 2 7]*0.011,...
    'Parent',ElementA,'erasemode','none','linewidth',1.2,...
    'facecolor',[0.5 0.5 0.5]);

% Moving Elements B
ElementB=hgtransform;
set(ElementB,'Parent',AxesH)
Borigin=6;

%Set up elements, cases and connections for branch B
for i=1:4,
    eval(['ElementBBar' num2str(i) '=...
        'line([Borigin_Borigin]*0.011,[-5.5+i*2_-4.5+i*2]*0.011,...
        ''Parent'',ElementB,''erasemode'', ''none'', ''linewidth'',1.2);']);
    eval(['ElementBConnect' num2str(i) '=...
        'line([Borigin_Borigin+2.5]*0.011,[-5+i*2_-5+i*2]*0.011,...
        ''Parent'',ElementB,''erasemode'', ''none'', ''linewidth'',1.2);']);

```



```

eval(['CaseB' num2str(i) '=' ...
    'rectangle(''Position'', [Borigin+5 -5.5+i*2 2]*0.011, ...
    ''Parent'', ElementB, 'erasemode'', 'none'', 'linewidth'', 1.2);']);
eval(['CaseBConnect' num2str(i) '=' ...
    'line([Borigin+4.5 Borigin+5]*0.011, [-5+i*2 -5+i*2]*0.011, ...
    ''Parent'', ElementB, 'erasemode'', 'none'', 'linewidth'', 1.2);']);
end

% Set Mass B
MassB=rectangle('Position', [Borigin+2.5 -3.5 2 7]*0.011, ...
    'Parent', ElementB, 'erasemode', 'none', 'linewidth', 1.2, ...
    'facecolor', [0.5 0.5 0.5]);

% Moving Elements C
ElementC=hgtransform;
set(ElementC, 'Parent', AxesH)
Corigin=12;

% Set up elements, cases and connections for branch C
for i=1:4,
    eval(['ElementCBar' num2str(i) '=' ...
        'line([Corigin Corigin]*0.011, [-5.5+i*2 -4.5+i*2]*0.011, ...
        ''Parent'', ElementC, 'erasemode'', 'none'', 'linewidth'', 1.2);']);
    eval(['ElementCConnect' num2str(i) '=' ...
        'line([Corigin Corigin+2.5]*0.011, [-5+i*2 -5+i*2]*0.011, ...
        ''Parent'', ElementC, 'erasemode'', 'none'', 'linewidth'', 1.2);']);
    eval(['CaseC' num2str(i) '=' ...
        'rectangle(''Position'', [Corigin+5 -5.5+i*2 2]*0.011, ...
        ''Parent'', ElementC, 'erasemode'', 'none'', 'linewidth'', 1.2);']);
    eval(['CaseCConnect' num2str(i) '=' ...
        'line([Corigin+4.5 Corigin+5]*0.011, [-5+i*2 -5+i*2]*0.011, ...
        ''Parent'', ElementC, 'erasemode'', 'none'', 'linewidth'', 1.2);']);
end

% Set Mass C
MassC=rectangle('Position', [Corigin+2.5 -3.5 2 7]*0.011, ...
    'Parent', ElementC, 'erasemode', 'none', 'linewidth', 1.2, ...
    'facecolor', [0.5 0.5 0.5]);

% Moving Elements D
ElementD=hgtransform;
set(ElementD, 'Parent', AxesH)
Dorigin=18;

% Set up elements, cases and connections for branch D
for i=1:4,
    eval(['ElementDBar' num2str(i) '=' ...
        'line([Dorigin Dorigin]*0.011, [-5.5+i*2 -4.5+i*2]*0.011, ...
        ''Parent'', ElementD, 'erasemode'', 'none'', 'linewidth'', 1.2);']);
    eval(['ElementDConnect' num2str(i) '=' ...
        'line([Dorigin Dorigin+2.5]*0.011, [-5+i*2 -5+i*2]*0.011, ...
        ''Parent'', ElementD, 'erasemode'', 'none'', 'linewidth'', 1.2);']);

```

```
end
    %Set Mass D
    MassD=rectangle('Position', [Dorigin+2.5 -3.5 2 7]*0.011,...
        'Parent',ElementD,'erasemode','none','linewidth',1.2,...
        'facecolor',[0.5 0.5 0.5]);

uicontrol(...
    'Parent',          Fig,...
    'Style',           'text',...
    'Units',           'pixel',...
    'HorizontalAlignment','right',...
    'String',          'Time:□',...
    'Position',        [20 20 60 20])

TimeField = uicontrol(...
    'Parent',          Fig,...
    'Style',           'text',...
    'Units',           'pixel', ...
    'HorizontalAlignment','left',...
    'String',          num2str(TimeClock),...
    'Position',        [80 20 60 20]);

FigUD.GndElement1 = GndElement1;
FigUD.GndElement2 = GndElement2;
FigUD.GndElement3 = GndElement3;
FigUD.GndElement4 = GndElement4;

FigUD.CaseA1 = CaseA1;
FigUD.CaseA2 = CaseA2;
FigUD.CaseA3 = CaseA3;
FigUD.CaseA4 = CaseA4;

FigUD.CaseB1 = CaseB1;
FigUD.CaseB2 = CaseB2;
FigUD.CaseB3 = CaseB3;
FigUD.CaseB4 = CaseB4;

FigUD.CaseC1 = CaseC1;
FigUD.CaseC2 = CaseC2;
FigUD.CaseC3 = CaseC3;
FigUD.CaseC4 = CaseC4;

FigUD.ElementA = ElementA;
FigUD.ElementB = ElementB;
FigUD.ElementC = ElementC;
FigUD.ElementD = ElementD;
```

```
FigUD.TimeField      = TimeField;
FigUD.Ground         = Ground;

FigUD.Block          = get_param(gcbh,'Handle');
set(Fig,'UserData',FigUD);
drawnow
% store the figure handle in the animation block's UserData
set_param(gcbh,'UserData',Fig);
% end LocalHRAInit
```

Appendix F

Publications

Modelling of High Redundancy Actuation Utilising Multiple Moving Coil Actuators

J. Davies* T. Steffen* R. Dixon* R.M. Goodall*
A.C. Zolotas* J. Pearson**

* Control Systems Group, Loughborough University, Loughborough,
LE11 3TU, UK, <http://www.lboro.ac.uk/departments/el/research/scg>

** SEIC, BAE Systems, Holywell Park, Loughborough, LE11 3TU, UK,
<http://www.seic-loughborough.com>

Abstract: This paper presents the modelling of a moving coil actuator for use as an element in a High Redundancy Actuator (HRA). A single element model is derived from first principles and verified using experimental data. This model is subsequently used to describe an approach to deriving models of multi-element HRAs and determine the effect of a variety of faults, chosen to be appropriate for the electro-magnetic technology, on the behaviour of multi-element assemblies.

1. INTRODUCTION

1.1 Fault Tolerant Control and Actuator Redundancy

A fault may be defined as a defect or imperfection that occurs in the hardware or software of a system. Faults in automated processes will often cause undesired reactions which could manifest as failures, where an expected action is not completed by the overall system. The consequences of failures could include damage to the plant, its environment, or people in the vicinity of that plant [Blanke et al., 2001]. Fault tolerant control aims to prevent failures and achieve adequate system performance in the presence of faults.

The majority of research to date has concentrated on sensor faults. Significant advances have been made in this area, however, most of these strategies are not applicable to actuator faults. This is attributable to the fundamental differences between actuators and sensors. Sensors deal with information, and measurements may be processed or replicated analytically to provide fault tolerance. However, actuators must deal with energy conversion, and as a result actuator redundancy is essential if fault tolerance is to be achieved in the presence of actuator faults. Actuation force will always be required to keep the system in control and bring it to the desired state [Patton, 1991]. No approach can avoid this fundamental requirement.

The common solution is to use some form of over-actuation in which the fault-free system has more control action than needed. For critical systems, the normal approach involves straightforward replication of the actuators, e.g. 3 or 4 actuators are used in parallel for aircraft flight control systems. Each redundant actuator must be capable of performing the task alone and possibly override the other faulty actuators. This over-engineering however, incurs penalties as cost and weight are increased and subsequently efficiency is reduced.

1.2 High Redundancy Actuation

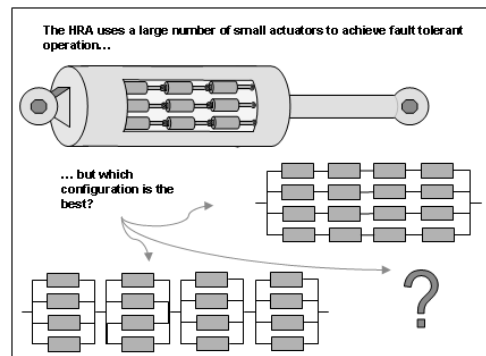


Figure 1. High Redundancy Actuator.

The High Redundancy Actuator (HRA) concept is a novel approach to actuator fault tolerance, inspired by human musculature. A muscle is composed of many individual muscle cells, each of which provides a minute contribution to the force and the travel of the muscle. These properties allow the muscle, as a whole, to be highly resilient to individual cell damage.

The HRA project aims to use the same principle of cooperation to provide intrinsic fault tolerance using existing technology. To achieve this, a high number of small actuator elements are assembled in parallel and in series to form one highly redundant actuator (see Figure 1). Faults within the actuator will affect the maximum capability, but through robust control, full performance can be maintained without either adaptation or reconfiguration.

The HRA is an important new approach within the overall area of fault-tolerant control. When applicable, it can provide actuators that gracefully degrade, and that continue

to operate at close to nominal performance in the presence of multiple faults in the actuator elements. The HRA research project has already studied the use of electro-mechanical technology [Du et al., 2007] in order to assess the concept's viability. Progress towards an electro-magnetic HRA is also under way [Steffen et al., 2007a].

1.3 Overview

This paper presents the modelling of a moving coil actuator that is intended to be the building block of an electro-magnetic HRA. Derivation of the element model from operating principles and equivalent circuits is provided in Section 2. Section 3 describes the experimental identification of the parameters and verification of the model. The modelling of faults in a single element is given in Section 4. Section 5 details the modelling of element assemblies, the effect of faults in which is discussed in Section 6. Finally, the paper's conclusions are made in Section 7 which includes comments on the future direction of this research.

2. MODELLING OF A SINGLE ACTUATION ELEMENT

In order to construct a multi-element actuation system, it is first necessary to model a single actuation element i.e. a moving coil actuator provided by SMAC UK Ltd. [SMAC, 2004]. This modelling will be addressed here.

2.1 Operating Principles

Figure 2 illustrates the basic components a moving coil

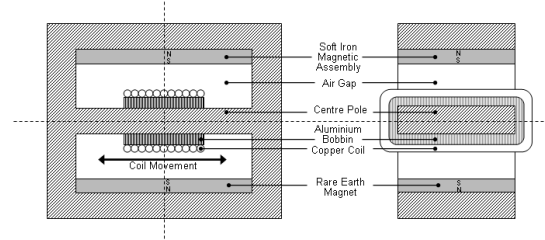


Figure 2. Moving Coil Actuator.

actuator. It comprises a moving coil wound round the centre pole of a magnetic assembly that produces a uniform magnetic field perpendicular to the current conducted in the coil. On providing a voltage, a current flows in the coil generating a force which is parallel to the direction of travel. This force causes the coil, and the rod which is mounted to it, to move. The force is proportional to the current in the coil, the number of turns, and the flux strength.

The copper coil is wound round an aluminium bobbin, which forms part of the piston carriage. This aluminium bobbin surrounds the centre pole of the magnet, forming a circuit, and as such, as it moves within the magnetic field, eddy currents are induced within it. These eddy currents produce magnetic fields that oppose the external magnetic field and thus oppose the movement of the coil causing a

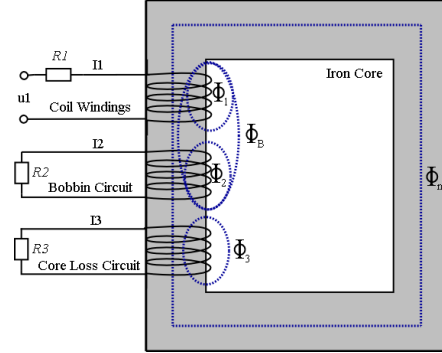


Figure 3. Magnetic flux within the static system.

damping effect. In addition, eddy currents are also induced within the bobbin by the changing current in the coil. This aspect of the moving coil actuator in question complicates the modelling procedure as the usual resistor inductor circuit that is used to model the electrical characteristics is no longer appropriate. Hence, a different approach, based on magnetic principles is taken to the modelling of this system.

As the system contains electrical, magnetic and mechanical elements, electrical analogies will be used to derive one homogeneous model. The actuation element will be modelled in two stages: firstly the electrical subsystem which characterises the force produced by the electrical input, and then the mechanical subsystem upon which this force is acting. Equivalent circuits will be formulated for both subsystems and then they will be combined using dynamical laws to produce one overall equivalent circuit for the element.

2.2 Electrical Subsystem

Figure 3 illustrates the flux within the system. The figure shows the iron core surrounded by three coil circuits: the moving coil with its voltage input u_1 and winding resistance R_1 ; a second circuit representing the bobbin, which is effectively a closed-turn with resistance R_2 ; and a third coil representing the inductive and resistive core losses. The majority of the flux flows in the iron core, and is shown in Figure 3 as Φ_M . Φ_1 is the flux linking the coil and Φ_2 is the flux flowing in the bobbin. Φ_b is the flux that links the coil and the bobbin. Finally, the core losses are denoted as Φ_3 .

Using the following expressions for the magnetomotive force (m.m.f.) that creates the flux and the electromotive force (e.m.f.) created across the coils by the changing flux:

$$\mathcal{F} = \mathcal{R}\Phi \quad (1)$$

$$E = N \frac{d\Phi}{dt} \quad (2)$$

The three circuit equations can be defined:

$$u_1 = N_1 \frac{d}{dt} (\Phi_M + \Phi_1 + \Phi_B) + R_1 I_1 \quad (3)$$

$$0 = N_2 \frac{d}{dt} (\Phi_M + \Phi_2 + \Phi_B) + R_2 I_2 \quad (4)$$

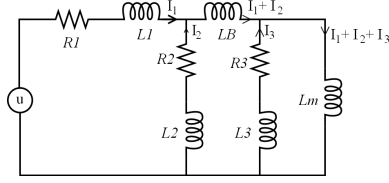


Figure 4. Electrical subsystem equivalent circuit.

$$0 = N_3 \frac{d}{dt} (\Phi_M + \Phi_3) + R_3 I_3 \quad (5)$$

and using the m.m.f. law, analogous to AMPERE's law, $\mathcal{R}\Phi = NI$ and substituting terms in $\frac{N}{\mathcal{R}}$ for inductances gives:

$$u_1 = N_1 \left(L_M \frac{dI_m}{dt} + L_1 \frac{dI_1}{dt} + L_B \frac{d}{dt} (I_1 + I_2) \right) + R_1 I_1 \quad (6)$$

$$0 = N_2 \left(L_M \frac{dI_m}{dt} + L_2 \frac{dI_2}{dt} + L_B \frac{d}{dt} (I_1 + I_2) \right) + R_2 I_2 \quad (7)$$

$$0 = N_3 \left(L_M \frac{dI_m}{dt} + L_3 \frac{dI_3}{dt} \right) + R_3 I_3 \quad (8)$$

where $I_m = I_1 + I_2 + I_3$. These equations describe the element without mechanical movement i.e. when the bobbin is clamped. Hence, the mode represents only the electrical subsystem. Some simplifications may be made as L_2 and L_3 are much smaller than L_m and L_B and thus they may be removed with little affect on the system [Chai, 1998]. The resultant equivalent circuit is shown in Figure 4, and the following transfer function may be derived:

$$\frac{I_{R1}}{u_{in}} = \frac{L_B L_m s^2 + (L_B R_3 + L_m R_5)s + R_2 R_3}{L_B L_m L_1 s^3 + c_1 s^2 + c_2 s + R_1 R_2 R_3} \quad (9)$$

where:

$$\begin{aligned} R_4 &= (R_1 + R_2), \quad R_5 = (R_2 + R_3) \\ c_1 &= (L_m(L_B R_4 + L_1 R_5) + L_B L_1 R_3) \\ c_2 &= (R_2(L_m R_1 + L_1 R_3) + R_3 R_4(L_B + L_m)) \end{aligned}$$

2.3 Mechanical Subsystem

The mechanical subsystem is a typical second order system consisting of the moving mass of the element and any stiffness and damping within the system with an input force originating from the electrical subsystem. Using NEWTON's Law the mechanical subsystem can be described by the equation of motion given in equation (10).

$$\ddot{x} = \frac{1}{m} F - \frac{d}{m} \dot{x} - \frac{r}{m} x \quad (10)$$

Using the current-force analogy, this mechanical subsystem can also be described by an equivalent circuit that has a current input analogous to the electrical force supplying three parallel components:

- a capacitance, C_m representing the moving mass,
- a resistor, R_d representing the damping within the mechanical system, as well as the damping caused by the velocity induced eddy currents,

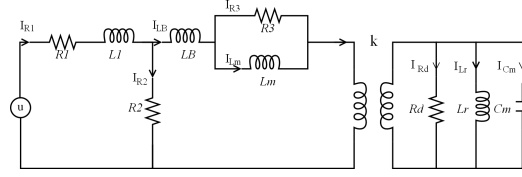


Figure 5. Final equivalent circuit.

- and an inductor, L_r representing any stiffness within the system.

2.4 Full Model

The full model can be created by combining the two subsystems with the dynamics of the system. There are two equations that describe the flow of energy between the two subsystem circuits: the LORENTZ force law and FARADAY's law of induction.

The current flowing perpendicular to the flux density results in a force known as the LORENTZ force:

$$F = BNlI \quad (11)$$

This force moves both the coil and the bobbin, therefore a force is generated by both I_1 and I_2 :

$$F = BNlI_1 + BNlI_2 = k(I_1 + I_2) \quad (12)$$

The magnetic flux density, B is assumed to be constant over the travel of the coil/bobbin. The number of turns N and the conductor turn length l are also constant and so BNl may be combined to produce one force constant k . This force is the input to the mechanical subsystem.

As the coil and bobbin are allowed to move in the field, their movement will generate counter-electromotive forces within their circuits which can be expressed as below:

$$E = BNl\dot{x} = k\dot{x} \quad (13)$$

The derivative \dot{x} is the perpendicular component of the velocity of the wire relative to the flux lines. The voltage equations (6) and (7) are augmented as below to include the counter-electromotive force:

$$u_1 = N_1 \left(L_M \frac{dI_m}{dt} + L_1 \frac{dI_1}{dt} + L_B \frac{d}{dt} (I_1 + I_2) \right) + R_1 I_1 + \dot{x}k \quad (14)$$

$$0 = N_2 \left(L_M \frac{dI_m}{dt} + L_B \frac{d}{dt} (I_1 + I_2) \right) + R_2 I_2 + \dot{x}k \quad (15)$$

In the mechanical-electrical analogy, velocity is equivalent to voltage, and thus the voltage across the capacitor in the mechanical circuit is \dot{x} . As both the counter-electromotive force equation and the LORENTZ force equation have a factor of k , the transfer of force between the two subsystem circuits is equivalent to a transformer with a turns ratio of \sqrt{k} . The mechanical components can be transferred to the primary electrical side by multiplying them by the square of the turn ratio, producing the final equivalent circuit as shown in Figure 5. The final system parameters are displayed in Table 1.

Table 1. System Parameters

Symbol	Meaning
R_1	Coil winding resistance
L_1	Coil inductance
R_2	Bobbin (eddy current) Resistance
R_3	Core loss resistance
L_B	Bobbin-coil inductance
L_m	Mutual inductance
R_d	Resistor equivalent of mechanical damping
L_r	Inductor equivalent of mechanical friction
C_m	Capacitor equivalent of moving mass
k	Force constant

From the equivalent circuit, the following state-space expression can be formed:

$$\begin{bmatrix} \dot{I}_{R1} \\ \dot{I}_{LB} \\ \dot{I}_{L3} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} \frac{-R_4}{L_1} & \frac{R_2}{L_1} & 0 & 0 & 0 \\ \frac{R_2}{L_1} & \frac{-R_5}{L_1} & \frac{R_3}{L_1} & \frac{-k}{L_B} & 0 \\ \frac{R_3}{L_1} & \frac{R_2}{L_1} & \frac{-R_5}{L_1} & \frac{R_3}{L_B} & 0 \\ 0 & \frac{L_m}{k} & \frac{L_m}{k} & 0 & 0 \\ 0 & \frac{C_m}{k} & 0 & \frac{-k}{C_m R_d} & \frac{-k}{C_m L_r} \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} I_{R1} \\ I_{LB} \\ I_{L3} \\ x \end{bmatrix} + \begin{bmatrix} 1 \\ L_1 \\ 0 \\ 0 \\ 0 \end{bmatrix} u_{in} \quad (16)$$

3. MODEL VERIFICATION & PARAMETER IDENTIFICATION

Frequency sweeps were made on a single actuation element in order to determine the parameter values and verify the model. Three signals were measured: the position and acceleration of the rod, and the coil current. Frequency responses for these signals were obtained over the frequency range 1-2000Hz with the coil free-moving. A frequency sweep for coil current was also carried out with the coil clamped mid-way along its travel. This aids the identification process as clamping the coil removes the mechanical dynamics from the system.

Only two parameters could be measured directly: the moving mass, C_m and the force constant, k . The moving mass was weighed and the force constant determined by applying a known current to the element and measuring its force using a scales. These two parameters determine the capacitance C_m as $C_m = mk^2$.

The remaining parameters were found by fitting the model to the frequency response data using the optimisation toolbox. The frequency data was entered into Matlab and weights were applied to favour the magnitude response and the 10 – 100 Hz region and remove the influence of the high frequency regions in the position/acceleration responses. Known model values were set and the remaining parameters defined as values to be determined. The model response was then matched to the measured data by defining the difference between them as a scalar function, and using the Matlab function 'fsolve' to find a minimum of the function through variation of the parameters, starting from an initial estimate.

The clamped frequency response was used first in the fitting process, as this system has fewer parameters. The clamped system transfer function was stated in equation (9). The results suggested that the effect of the inductance L_B (symbolising the flux linking the bobbin

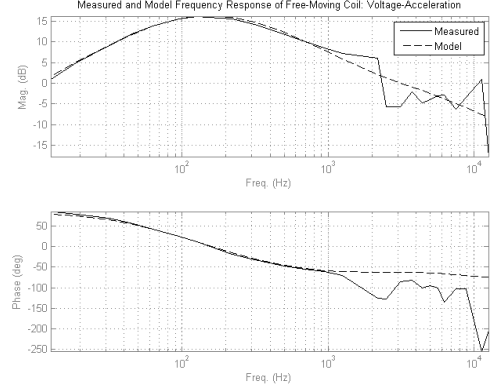


Figure 6. Model and Experimental Frequency Response of Free-Moving Coil: Voltage-Acceleration.

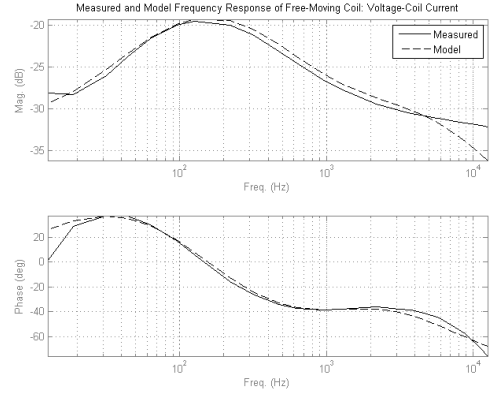


Figure 7. Model and Experimental Frequency Response of Free-Moving Coil: Voltage-Current.

and the coil) on the system was negligible. Thus, L_B was removed from the model, simplifying it by a degree. The new clamped subsystem model is as described in equation (17). Hence, the parameters L_m , L_1 , R_1 and $R_{2|3}$ were determined from this response.

$$\frac{I_{R1}}{u_{in}} = \frac{L_m s + R_{2|3}}{L_m L_1 s^2 + (L_m (R_{2|3} + R_1) + L_1 R_{2|3}) s + R_1 R_{2|3}} \quad (17)$$

where:

$$R_{2|3} = \frac{R_2 R_3}{(R_2 + R_3)}$$

The free-moving current and position responses were used to determine the remaining model parameters. The ratio between R_2 and R_3 was determined, allowing values for each to be found. The mechanical parameters L_r and R_d did not have a significant affect on the system and thus are set very high. The frequency responses of the model and measured data for the free-moving system are shown in Figures 6 and 7.

The model provides a good fit to the measured data between 5 – 100 Hz, which is the critical frequency range.

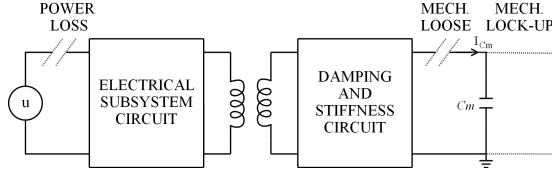


Figure 8. Faults represented in the equivalent circuit a single actuation elements.

The discrepancies present in the acceleration and position response at higher frequencies are due to unmodelled mechanical resonances. There may also be some skin effects present in the high frequencies, which could be modelled. However, this would increase the model order significantly.

In the acceleration frequency response, a discrepancy below 5 Hz can also be observed. This difference is attributable to stiction. Again, the inclusion of the stiction dynamics in the model is not considered worthwhile as this would introduce non-linearities into the system. However, transient response data suggests that the stiction is significant and that its inclusion in the model may be necessary in the future.

4. FAULT MODELLING IN A SINGLE ELEMENT

As the HRA is being developed in the interest of fault tolerance, it is necessary to model potential faults that can be injected into the system. Three main fault cases have been identified and modelled to date, namely:

- Mechanical Loose - A mechanically loose actuation element loses the ability to exert force between its two end points. Thus, a mechanically loose element behaves as if it is not there.
- Mechanical Lock-up - An element loses the ability to change the length between its two end points. This may occur if the coil of the first actuation element is deformed and touches the magnet. This fixes the mass with respect to the reference point, and consequently the relative position and the speed are constant.
- Power Loss - This fault is where the electrical input to the actuation element is lost, or the coil circuit becomes open circuit, but the mechanical subsystem continues to operate.

These faults are easily represented in the electrical equivalent circuit format. Figure 8 illustrates where the equivalent circuit for an actuation element needs to be shorted or severed to represent the given faults. A loss of power is realised by breaking the circuit so that the electrical power supply is disconnected. The mechanical loose fault is similar, as the force applied to the mass is lost. As current is equivalent to force in the current analogy, and the capacitance C_m represents the mass, the current supply to this component I_{Cm} must be removed, and thus the circuit is opened at this point. The mechanical lock-up fault requires the capacitor to be short circuited: the force applied to the mass is bypassed, fixing the masses velocity and position relative to the preceding element or surface.

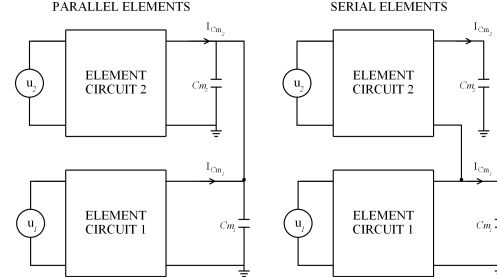


Figure 9. Parallel and serial assemblies using circuit analogies.

5. MODELLING OF ACTUATION ASSEMBLIES

Having developed a model for a single actuation element, multi-element assemblies can now be constructed to form a HRA. The current focus of the project is on planar assemblies and thus the elements are arranged either serially, or in parallel, or in serial/parallel combinations. The optimum configuration of actuation elements, in accordance with the high redundancy actuation concept, is discussed in [Steffen et al., 2007a], and hence will not be addressed here. This section will merely address the issue of creating models of possible actuation assemblies using SMAC moving coil actuators as actuation elements.

The use of electrical analogies in the model derivation allows multi-element actuator models to be created by replicating and interconnecting the equivalent circuit according to the assembly structure. For example, if two actuation elements are arranged in parallel and act upon a common load, their forces add and act upon one moving mass. Thus, the element equivalent circuit capacitor branch currents I_{Cm1} and I_{Cm2} add and flow through one combined capacitor, or alternatively each current flows separately through two parallel capacitors that add to make the moving mass as shown in Figure 9.

If actuators are connected in series, the first moving mass has the force of the first element and an opposing force from the second element acting upon it, and the second moving mass has the second element force applied to it. In electrical equivalence terms this means the first capacitor C_{m1} has the current $I_{Cm1} - I_{Cm2}$ and the second capacitor C_{m2} has the current I_{Cm2} and thus the two circuits are connected as shown in Figure 9.

More complicated assemblies can be modelled based on these two fundamental circuits.

6. FAULTS IN ACTUATION ASSEMBLIES

The equivalent circuit representation of the model provides an intuitive insight into the effect of the faults on multi-element assemblies. To illustrate this point, the equivalent circuit fault model for a two-by-two series-in-parallel system, as shown in Figure 10, is given in Figure 11.

In the example assembly, elements one and two work upon masses m_1 and m_2 respectively. m_1 and m_2 are the combined mass of the moving mass of elements 1 and 2 the casing mass of 3 and 4, respectively. The casing masses of

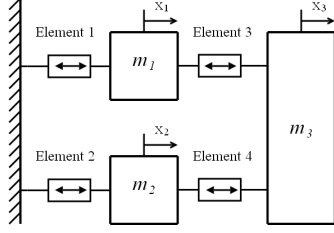


Figure 10. Two-by-two series-in-parallel assembly.

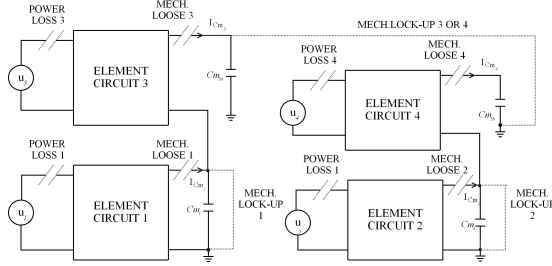


Figure 11. Faults represented in the equivalent circuit for a two-by-two series-in-parallel assembly.

actuation elements 1 and 2 are not included in the diagram as they are fixed to a surface. actuation elements 3 and 4 both apply their force to m_3 , which is the combined mass of the moving masses of elements 3 and 4 and the load mass. The effect of faults on this system, with reference to the equivalent circuit diagram, is discussed below.

6.1 Power Loss

A power loss fault in any of the actuation elements removes the influence of the electrical subsystem on the associated mechanical subsystem, but does not affect the flow of force between the systems. Hence theoretically, this HRA can withstand up to three power losses without complete loss of force to the load.

6.2 Mechanical Loose

A mechanical loose fault in actuation elements 1 or 2 results in loss of current to C_{m1} or C_{m2} respectively. In mechanical terms, this means that the force on m_1 or m_2 is lost, rendering that serial branch inoperable. The same applies to elements 3 and 4, as a loose fault in either will result in a reduction of current to $C_{m3(a+b)}$. However, the remaining un-loose serial branch will continue to provide force to the load mass in either case, resulting in a theoretically operational system.

If both serial branches suffer a loose fault however, current to $C_{m3(a+b)}$ will be lost completely, resulting in the failure of this HRA configuration.

6.3 Mechanical Lock-up

A mechanical lock-up in elements 3 or 4 results in the short-circuiting of the capacitance $C_{m3(a+b)}$, hence fixing x_3 and \dot{x}_3 with respect to the previous masses. If elements

1 or 2 lock-up, then their respective capacitor is short-circuited and thus the states of the associated mass are fixed with respect to the fixed surface. The system will remain theoretically operational as long as one element in each serial branch remains un-locked.

These observations confirm the logical deductions that parallel elements reduce the effect of loose mechanical faults, but do not aid fault tolerance in the case of mechanical lock-ups. Conversely, serial elements improve fault tolerance to mechanical lock-ups, as the other element remains effective, but are vulnerable mechanically loose faults. The quantification of fault tolerance within high redundancy actuators is further discussed in [Steffen et al., 2007b].

7. CONCLUSIONS

In this paper a model for a moving coil actuator has been derived with the intention of using it as a single element within an electro-magnetic HRA. The model was verified using experimental data and the parameter values were identified. The modelling of faults appropriate to moving coil technology in a single element was considered and the formulation of assembly models using equivalent circuits was discussed. Finally, the effect of faults on multi-element assemblies was considered and an example given. This modelling provides a foundation for the control studies planned for the future. The control studies planned within the project take two directions: robust control strategies and self-organising control.

ACKNOWLEDGEMENTS

This project is a cooperation of the Control Systems group at Loughborough University, the Systems Engineering and Innovation Centre (SEIC), and the actuator supply SMAC UK Ltd.. The project is funded by the UK's Engineering and Physical Sciences Research Council (EPSRC) under reference EP/D078350/1.

REFERENCES

- M. Blanke, M. Staroswiecki, and N. E. Wu. Concepts and methods in fault-tolerant control. *American Control Conference, 2001. Proceedings of the 2001*, 4, 2001.
- H. D. Chai. *Electromechanical motion devices*. Prentice Hall, United States, 1998.
- X. Du, R. Dixon, R. M. Goodall, and A. C. Zolotas. Lqg control of a highly redundant actuator. In *Conference for Advanced Intelligent Mechatronics*, 2007.
- R. J. Patton. Fault detection and diagnosis in aerospace systems using analytical redundancy. *Computing and Control Engineering Journal*, 2(3), 1991.
- SMAC. Actuator data sheet - electric linear/rotary moving coil actuators lal90-50, 2004.
- T. Steffen, J. Davies, R. Dixon, R. M. Goodall, and A. C. Zolotas. Using a series of moving coils as high redundancy actuator. In *IEEE Conference for Advanced Intelligent Mechatronics*, Zurich, 2007a.
- T. Steffen, R. Dixon, R. M. Goodall, and A. C. Zolotas. Quantifying the fault tolerance of high redundancy actuator assembly. *Submitted to Mechatronics*, 2007b.

Multi-Agent Control of High Redundancy Actuation

J. Davies* T. Steffen* R. Dixon* R.M. Goodall*

* Control Systems Group, Loughborough University, Loughborough,
LE11 3TU, UK, <http://www.lboro.ac.uk/departments/el/research/scg>

Abstract: The High Redundancy Actuator (HRA) project investigates the use of a relatively high number of small actuation elements, assembled in series and parallel in order to form a single actuator which has intrinsic fault tolerance. Both passive and active methods of control are planned for use with the HRA. This paper presents progress towards a multiple model control scheme for the HRA applied through the framework of multi-agent control.

1. INTRODUCTION

1.1 Fault Tolerant Control and Actuator Redundancy

A fault may be defined as a defect or imperfection that occurs in the hardware or software of a system. Faults in automated processes will often cause undesired reactions which could manifest as failures, where an expected action is not completed by the overall system. The consequences of failures could include damage to the plant, its environment, or people in the vicinity of that plant [Blanke et al., 2001]. Fault tolerant control aims to prevent failures and achieve adequate system performance in the presence of faults.

The majority of research to date has concentrated on sensor faults. Significant advances have been made in this area, but most of these strategies are not applicable to actuator faults. This is attributable to the fundamental differences between actuators and sensors. Sensors deal with information, and measurements may be processed or replicated analytically to provide fault tolerance. Actuators, however, must deal with energy conversion, and as a result actuator redundancy is essential if fault tolerance is to be achieved in the presence of actuator faults. Actuation force will always be required to keep the system in control and bring it to the desired state [Patton, 1991].

The common solution for critical systems involves straightforward parallel replication of actuators. Each redundant actuator must be capable of performing the task alone and possibly override the other faulty actuators. This over-engineering incurs penalties as cost and weight are increased and subsequently, efficiency is reduced.

1.2 High Redundancy Actuation and Multi-Agent Control.

The High Redundancy Actuator (HRA) concept is a novel approach to actuator fault tolerance, inspired by human musculature. A muscle is composed of many individual cells, each of which provides a minute contribution to the force and the travel of the muscle. These properties allow the muscle, as a whole, to be highly resilient to individual cell damage.

The HRA project aims to use the same principle of co-operation to provide intrinsic fault tolerance using existing technology. To achieve this, a high number of small actuator elements are assembled in parallel and series to form one high redundancy actuator (see Figure 1).

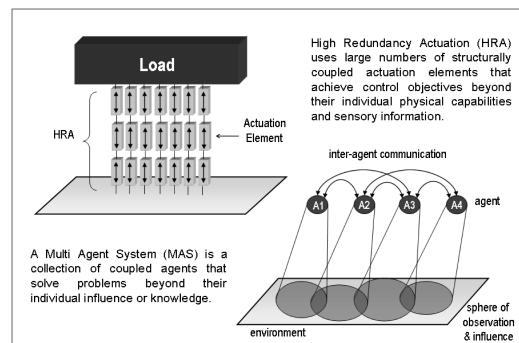


Figure 1. HRA and MAS.

Faults in elements will affect the maximum capability, but through control techniques, required performance can be maintained.

The HRA is an important new approach within the overall area of fault-tolerant control. When applicable, it can provide actuators that have graceful degradation, and that continue to operate at close to nominal performance even in the presence of multiple faults in the elements.

The main focus of the HRA project thus far has utilised robust control techniques. These techniques have been shown to be theoretically viable for fault tolerant control of low levels of redundancy [Du et al., 2007], and testing of these results on a two-by-two electromechanical rig is in its final stages [Du et al., 2008].

Electro-magnetic actuation is now being considered as a candidate element within the HRA, the modelling of which in both nominal and fault condition has been detailed in [Davies et al., 2008]. Research is ongoing into the robust control of these elements at higher levels of redundancy [Steffen et al., 2007]. Results to date suggest that robust

control should be a satisfactory method of achieving fault tolerant control of these structures. Indeed, the robust, passive¹ control approach is attractive, as its simplicity and constancy mitigate many of the associated problems with active control methods. However, research into more intelligent, active approaches is also an objective of the HRA project, to ascertain the levels of fault tolerance and nominal performance attainable in comparison to passive methods.

Multi-Agent Systems (MAS) are the focus of this active fault tolerance scheme. MAS was chosen as an intelligent approach to controlling the HRA as the two concepts are strongly related (Figure 1). Both are inspired by natural mechanisms which utilise vast numbers of relatively simple cells/processes to form complex structures/behaviours.

1.3 Overview

This paper presents the concepts and objectives of applying MAS to an electro-magnetic HRA with an example of Multi-Agent Control (MAC) applied to a parallel-series (PS) HRA. Section 2 briefly introduces agent concepts and discusses the rationale behind MAC of HRA. The current MAC scheme is described in Section 3, from both an agent and agency architecture perspective. Section 4 details simulation results of a MA controlled 4x4 HRA in comparison to a global control scheme with the injection of faults into the system. The further development of the multi-agent scheme is discussed in Section 5 and conclusions are made in Section 6.

2. MULTI-AGENT CONTROL OF A HIGH REDUNDANCY ACTUATOR (MACHRA)

2.1 Concepts of Multi-Agent Systems

An agent is a physical or virtual entity situated in its environment, which acts autonomously and flexibly within its purview to achieve goals in a real-time manner [Jennings et al., 1998]. A MAS therefore, is a collection of agents that are socially coupled and collaborate to achieve some objective, which in the case of MAC is the control of a system.

These agent characteristics resemble the concept of closed-loop control, which achieves objectives through sensing and acting. However, there are important differences within the agent concept. The most obvious difference is the social interaction and negotiation that exists between agents. Also, the agent philosophy is strongly associated with localisation, a point emphasised by [Ferber, 1999].

According to [Weiss, 1999], agent concepts are most beneficial and applicable in applications that have one or more of the following attributes:

Modularity/Decentralisation - A physically or functionally modular system is naturally identifiable with an agent structure. Similarly, agents are useful in a decentralised system, as they may be associated with distributed subsystems and their pro-active capabilities allow low-level

decisions to be made locally, facilitating the management of large systems.

Changability/Ill-structure - The modular and decentralised nature of agents allow the structure of the agency, or agents themselves, to be changed with minimum impact to the system, providing a robust adaptable solution. This is important in systems that are likely to change frequently, or ill-structured systems where the domain structure is not completely specified or static.

Complexity - A complex system, with many interacting elements and behaviours, can be served well by an agent approach, as problems may be solved in a more efficient and timely manner.

2.2 Why Take a Multi-Agent Approach to HRA Control?

In addition to the inherent similarities of HRA and MAS, the HRA is likely to benefit from a multi-agent approach as it displays many of the properties in Section 2.1.

The key rationale for combining MA concepts with HRA, however, is the structuring of both of these concepts. The HRA, viewed as a whole is a complex system, but if viewed as a collection of simpler, similar (if not identical), physically distributed modules, the complexity and changeable nature of the system's dynamics and structure can be handled at a local level, allowing objectives to be met with greater speed and efficiency. MASs facilitate the control of such decompositions, and due to their communicative and flexible qualities, potentially provide greater robustness and adaptability in fault situations.

The structuring of control is often neglected in the field of control engineering as the problem is stated in the form of a single plant model [van Breemen and de Vries]. The HRA is a complex, highly structured system, with well defined interactions between simple elements. An unstructured approach will have difficulties dealing with this complexity. MAC can replicate the structure of the HRA, which should simplify the individual control algorithms. The process industry acknowledges that the structuring of control is an important issue when applied to a decomposed system, thus it is given more attention in this field and numerous MACS have been proposed in this application area e.g. [Wang and Wang, 1997].

The actual control technique implemented in the agent is peripheral to the MA scheme. Classic or modern designs based on multiple model approaches can be implemented, with the MA concept providing the mechanism for intelligently deciding which controller to employ locally. Adaptive controllers could also be applied, again with the agents providing the decomposition of the problem.

Essentially, agent methods provide a framework to apply active control, fault detection and health monitoring to the HRA, whilst avoiding some of the issues associated with active control. Multiple model control schemes often have one active global controller, and a supervisor that decides which controller should be active. This centralisation can create problems with bumpless transfer, as large control signal changes can occur when switching between schemes and the supervisor becomes a single point of failure, increasing the systems reliance upon fault detection. In

¹ In this context, passive refers to a static control structure and algorithm.

addition to this, a global view on the system can make faults more complex to diagnose. These centralisation issues are negated by MAC, as are issues associated with adaptive control.

The unpredictability of centralised adaptive control schemes should be alleviated somewhat by the decentralisation MAC offers. Undesirable changes within modules will have less affect on the system as a whole, perhaps even with other agents adapting to counter-balance the unwanted behaviours. Localisation of control may also improve on response speed issues associated with adaptive control.

Nonetheless, MA concepts are not without their own potential disadvantages. Delays incurred by deliberation or communication and negotiation procedures may cause issues, as can the non-consensus of agents, which can lead to incompatible actions being taken or decisional instability. Issues can also arise in situations where agents fail to communicate or if they fail completely.

2.3 MACHRA Objectives

The objectives for the use of MAC in this project include those made for the control of the HRA with robust techniques, namely:

- Control of the elements resulting in a unified dynamic for the HRA.
- Nominal or acceptable behaviour of the HRA in element fault conditions.
- Graceful degradation of the HRA as fault levels increase beyond their critical point.
- Health and capability monitoring of the elements for maintenance/operator use.

If the inclusion of intelligence within the control scheme is to be justified then the MA controlled HRA must achieve tangibly more in comparison to passive methods. Thus, the objective for MAC of an HRA also include:

Increased reliability - Robust techniques can be limited in the number of faults or fault types they can accommodate. The structure of the HRA alleviates this problem somewhat, as the number of elements reduces the overall affect of faults on the system. Nevertheless, a more intelligent scheme, such as multi-agents, may accommodate even greater fault levels and fault types.

Improved nominal performance - Passive fault accommodation methods require the controller design to be robust enough to produce adequate performance during faulty conditions. This can lead to conservative performance in nominal conditions. An active control scheme can offer an increase in nominal performance as the control action can be changed in fault situations. Agent schemes may also provide performance enhancement due to their potential to pre-empt situations.

3. MACHRA SCHEME

The MACHRA scheme is currently in the investigative stage, concentrating on parallel in series (PS) configurations with lock-up and loose faults. Initial agent architectures and agency structures have been designed and simulated.

At present, Matlab/Simulink is used to create and simulate HRA assemblies, details of which can be found in [Davies et al., 2008]. Stateflow is used to simulate the inner rule-based logic of the agents and their communication. This provides a fast prototyping tool of the agents for use with Matlab/Simulink.

3.1 Agency Architecture

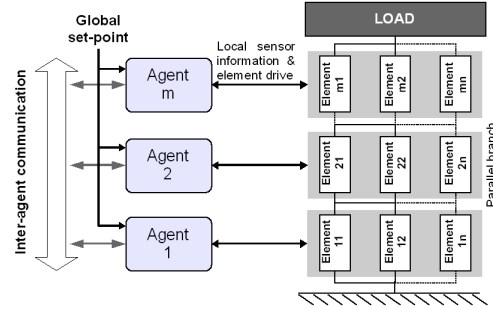


Figure 2. MACHRA agency architecture

The term agency architecture refers to the configuration of multiple agents on the macro scale. Figure 2 displays the MACHRA scheme's agency architecture for a $m \times n$ HRA PS configuration. There is an agent per parallel branch of elements, each of which is responsible for the control and detection of faults within its elements. In this configuration, lock-up faults will render the parallel branch inoperative, adding that branch's weight to the load of its neighbouring branches. However, loose faults will not affect the travel capability of that branch, as long as one operational element remains: PS assemblies have inherent fault tolerance to this fault type. Different configurations will provide more or less inherent tolerance to these faults, hence the configuration must be chosen to suit the application.

All agents within this scheme are identical and peers, consistent with the spirit of MAC where no hierarchy should exist. A global set-point for the whole HRA is given to each agent, as well as local position sensory input from its branch. Communication between agents is broadcast to all, however each message is addressed and the recipient agent only reads the messages that are from its immediate structural neighbours to save time when large numbers of elements are used. If lock-up faults occur, the agent's structural neighbours will change and thus different messages become relevant.

3.2 Agent Architecture

The current agent architecture is illustrated in Figure 3. This architecture has similarities with subsumption, first introduced by [Brooks, 1986], that uses behaviours layered in order of abstraction to produce more complex emergent behaviors in a reactive time-frame. This reactivity is key in the HRA as, due to the fast dynamics of the electromagnetic elements, a purely deliberative architecture may not provide the response times needed.

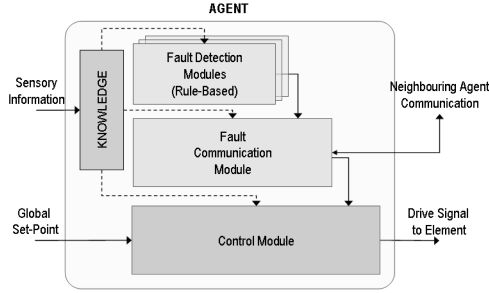


Figure 3. MACHRA agent architecture

The most reactive, basal behaviors are situated on the bottom layer, in this case the Control Module (CM), which provides the drive signal to the element based on the global set-point. A multiple model control scheme is employed within this design as the CM contains a look-up table with simple classical control designs based on the number of active agents structurally surrounding the agent e.g. more control action is required if an element's neighbour locks.

The Fault Detection Module (FDM) is the most abstracted layer, and thus affects those below it. As its name suggests, the FDM detects faults within its element. At the current stage of development, one fault type (lock-up faults) is detected. Future agents will have more than one module, arranged either as peers in a single layer or as separate layers ordered by the severity of the fault type. The module contains rule-based logic which determines the fault status of the element based on sensory information and internal knowledge.

The middle layer is the Fault Communication Module (FCM). This module communicates the fault status and global capability estimate to other agents, receiving messages of the same content from other agents. This information is also passed to the CM, where it is used to choose a controller and decide what portion of the overall global set-point to make its objective. In the absence of this information, the CM assumes nominal conditions if no previous communication with the FCM has been made, and last known conditions in the presence of communication history.

One shortcoming of the subsumption architecture, according to [van Breemen and de Vries], is its inability to combine information from different layers. This problem is negated in the MACHRA agent structure, as information is stored within an inter-accessible knowledge module. This module contains both knowledge given to the agent on start-up and that deduced within the individual modules.

Another commonly cited inadequacy of the subsumption scheme is the lack of consideration of previous events. This is not the case within the MACHRA architecture, as the modules have previous state-based behaviours, making the agent not purely reactive, but hybrid in nature.

4. MAC VS GLOBAL CONTROL SIMULATIONS

An example that illustrates the potential of MAC of HRA, in comparison to centralised passive control, is provided

Table 1. Fault Cases

Case	Description	HRA State
Nom.	All elements are healthy	Healthy & capable
F1	Branch nearest load locked	Faulty, but capable
F2	2 branches nearest load locked	Faulty and critical
F3	3 branches nearest load locked	Graceful degrade

Table 2. Requirements

Performance Requirements	
Travel Window	$\pm 0.015\text{m}$ ($2 \times \text{element travel}$)
Overshoot	$< 2\%$
Rise Time	$< 0.5\text{s}$
Settling Time	$< 0.75\text{s}$

in this section. The HRA system chosen is a 4x4 PS configuration with an overall travel control objective. The elements work upon a load that is twice as large as the inter-element masses and for the purposes of this example it is assumed that the HRA is over-dimensioned by a factor of two i.e. the maximum required travel is twice the travel of a single element².

As the PS assembly has natural tolerance to loose faults in terms of travel control, they will not be considered here. However, element lock-ups immobilise the parallel branches, and thus will be considered. Theoretically, a 4x4 system, of this dimensioning, may incur up to eight lock-up faults and still be capable of meeting its travel requirement. However, in a worst-case scenario, where single lock-ups occur in different branches, two lock-ups will bring the travel capability to critical point. Hence, faults will be injected in this worst-case manner, as described in Table 1.

4.1 Control Schemes

Figure 4 portrays the global and MAC control schemes. In the global scheme, a single, phase advance controller is designed to meet arbitrary transient requirements, displayed in Table 2, with good stability margins

Two MAC control schemes are included in this paper. The first scheme, MACS1 has set-point redistribution only. Each agent has a phase advance controller³ designed to meet the requirements in the nominal case. In fault conditions, the reduction in capability is communicated to other agents, and the extra travel required is distributed amongst the remaining active elements, minus the lock-up position of the faulty element. The control algorithm in each agent remains unchanged.

The second scheme, MACS2 utilises the same control algorithms as MACS1 under nominal conditions. The set-point redistribution described in MACS1 is also present. However, another controller per fault condition is designed and implemented when faults occur. Hence, this scheme illustrates set-point redistribution with a multiple model control scheme.

² This is an unrealistically low element mass-to-load ratio and high level of over-dimensioning for a HRA due to the relatively low level of redundancy used in this example.

³ The controllers in each agent are identical, as this reduces design time and aids verification for high integrity applications.

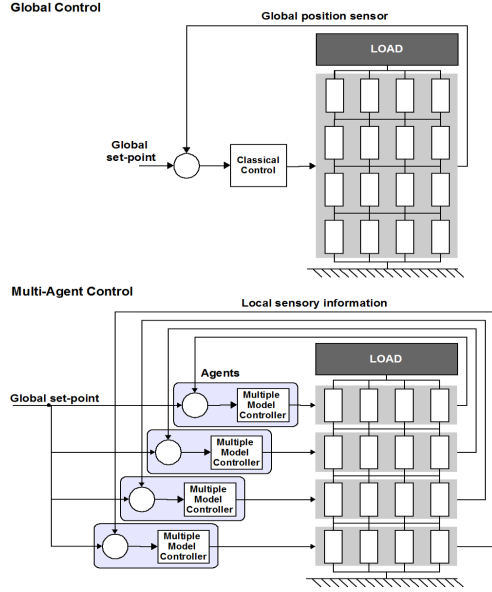


Figure 4. Global and agent control schemes

4.2 Comparison of Control under Fault Conditions

All faults were injected at $t=0$ and the set-point was an attainable travel in the worst fault case. Figure 5 shows a step response of the three control schemes under these conditions and their characteristics are summarised in Table 3. The simulations show that as faults occur, the increasing load slows the response.

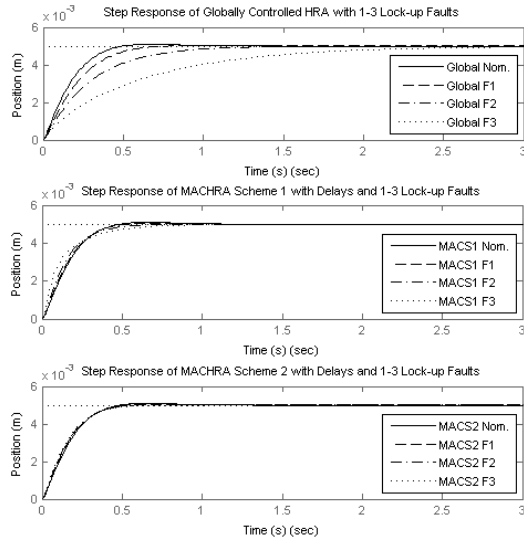


Figure 5. Step response for global control, MACS1 and MACS2

Table 3. Global control, MACS1 and MACS2 simulations

Fault Case	Overshoot	Rise Time	Settling Time
Glob. Nom.	1.80%	0.30s	0.45s
Glob. F1	0.37%	0.40s	0.62s
Glob. F2	0%	0.63s	1.14s
Glob. F3	0%	1.38s	2.48s
MACS1 Nom.	1.83%	0.30s	0.44s
MACS1 F1	0.95%	0.30s	0.45s
MACS1 F2	0%	0.30s	0.54s
MACS1 F3	0%	0.36s	0.73s
MACS2 Nom.	1.83%	0.30s	0.44s
MACS2 F1	1.01%	0.30s	0.45s
MACS2 F2	0.42%	0.30s	0.46s
MACS2 F3	0%	0.30s	0.49s

Comparing the global scheme to MACS1, it can be observed that the transient performance of the globally controlled HRA degrades to a greater extent than that of the MA controlled system with input redistribution only. The rise time and settling time increase significantly with each fault in the global case, whereas very little change is observed until three of the four branches are locked in the MAC case. This result illustrates that the localisation of control in this manner is favourable in comparison to the global approach.

The results shown in Table 3 for MACS2 are an improvement on those for MACS1. The rise time does not lengthen under any fault condition, and the change in settling time is significantly reduced. In addition to these marginal improvements, it can be seen from Figure 5, that the transient response envelope is tightened in general when using a multiple model control algorithm.

The MACHRA schemes simulated in this example offer significantly improved performance under fault conditions in comparison to a simple global control technique. However, it must also be acknowledged that the fault levels and over-dimensioning present in this example are much higher than those conceived by the HRA concept, and less distinction between the control schemes will be present with higher order configurations. Ultimately, the necessity for inclusion of active control strategies such as MAC will be dictated by the stringency of the requirements on a specific application.

4.3 Fault Detection and Control Reconfiguration Delays

The results discussed in Section 4.2 assumed that fault detection and control reconfiguration in the MA schemes was instantaneous, which is unrealistic. If Stateflow is used to simulate the multi-agent type fault detection, communication and control reconfiguration then a delay is introduced, providing a more realistic representation.

Figure 6 is a pulse train simulation of the previous MAC scheme's, simulated with Stateflow. As previously, all faults were injected at $t=0$ and an attainable travel demand was made.

The reconfiguration of the MACSs can be observed in the first pulse rise, resulting in a slower response. This is particularly pronounced where three faults need to be detected and communicated. However, in the remaining operational period, where no faults occur, there are no delay

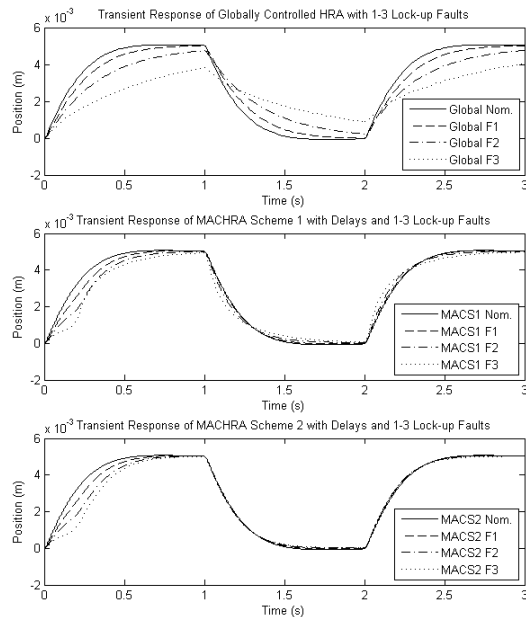


Figure 6. Pulse train for global control, MACS1 and MACS2 with delays

affects, and as such the response obtained in the previous section applies. This ephemeral performance degradation on fault injection may be considered acceptable in a real application, as faults are likely to be an infrequent event.

5. PLANNED DEVELOPMENTS

The development of a MAC scheme for the HRA is in its early stages and as such, many developments are envisaged. These include the improvement and extension of current fault detection schemes, and control in terms of the controlled variables and design techniques. The configurations to which the MAC is applied must also be extended to series-parallel and mixed configurations.

Changes to the architecture of the MAC scheme may be investigated and robust testing procedures are required, as many questions exist to the scheme's reliability. In addition to the robustness checking usually associated with control system design, situations where faults are misdiagnosed, communication is lost, and agents malfunction must be considered. Also, issues regarding feasibility at very high levels of redundancy must be addressed.

The practical testing of MAC on a experimental electromagnetic HRA is also planned, which should give an indication of such a scheme's performance in a real-world situation.

6. CONCLUSIONS

MAC potentially provides an active fault tolerant solution to controlling the HRA that improves on the nominal and faulty performance of passive schemes, whilst negating some of the issues associated with control reconfiguration.

The simulation of a suggested agent/agency architecture was included within an example that compared MAC with a more traditional global scheme. This example illustrated that requirements can be met under greater fault levels with the assistance of MA concepts, using input redistribution alone or in conjunction with localised multiple model control.

Many further developments for the MACHRA are planned including the practical testing on an experimental HRA rig, which will help ascertain the feasibility of MAC for this application.

ACKNOWLEDGEMENTS

This project is a cooperation of the Control Systems group at Loughborough University, the Systems Engineering and Innovation Centre (SEIC), and the actuator supply SMAC UK Ltd.. The project is funded by the UK's Engineering and Physical Sciences Research Council (EPSRC) under reference EP/D078350/1.

REFERENCES

- M. Blanke, M. Staroswiecki, and N. E. Wu. Concepts and methods in fault-tolerant control. *American Control Conference, Proceedings of the 2001*, 4, 2001.
- R.A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1): 14–23, 1986.
- J. Davies, T. Steffen, R. Dixon, R. M. Goodall, A. C. Zolotas, and J. Pearson. Modelling of high redundancy actuation utilising multiple moving coil actuators. *IFAC World Congress, Seoul*, 2008.
- X. Du, R. Dixon, R. M. Goodall, and A. C. Zolotas. Lqg control of a highly redundant actuator. In *Conference for Advanced Intelligent Mechatronics*, 2007.
- X. Du, R. Dixon, R.M. Goodall, and A.C. Zolotas. Assessment of strategies for control of high redundancy actuators. *Actuator 08, Bremen*, 2008.
- J. Ferber. *Multi-agent systems: an introduction to distributed artificial intelligence*. Addison-Wesley, United States, 1999.
- N. R. Jennings, K. Sycara, and M. Wooldridge. A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems*, 1(1):7–38, 1998.
- R. J. Patton. Fault detection and diagnosis in aerospace systems using analytical redundancy. *Computing and Control Engineering Journal*, 2(3), 1991.
- T. Steffen, J. Davies, R. Dixon, R. M. Goodall, and A. C. Zolotas. Using a series of moving coils as high redundancy actuator. In *IEEE Conference for Advanced Intelligent Mechatronics*, Zurich, 2007.
- A. van Breemen and T. de Vries. An agent-based framework for designing multi-controller systems. *International Conference on The Practical Applications of Intelligent Agents and Multi-Agent Technology, Manchester*, 2000.
- H. Wang and C. Wang. Intelligent agents in the nuclear industry. *IEEE Computer*, 30(11):28–34, 1997.
- G. Weiss. *Multiagent Systems A Modern Approach to Distributed Artificial Intelligence*. The MIT Press, United States of America, 1999.

Multi-Agent Control of a 10x10 High Redundancy Actuator

J. Davies* T. Steffen* R. Dixon* R.M. Goodall*

* Control Systems Group, Loughborough University, Loughborough,
LE11 3TU, UK, <http://www.lboro.ac.uk/departments/el/research/scg>

Abstract: The High Redundancy Actuator (HRA) project investigates the use of a relatively high number of small actuation elements, assembled in series and parallel in order to form a single actuator which has intrinsic fault tolerance. Both passive and active methods of control are planned for use with the HRA. This paper presents a multiple-model control scheme for a 10x10 HRA applied through the framework of multi-agent control.

1. INTRODUCTION

1.1 Traditional Approaches to Fault Tolerant Actuation

In automated processes, faults in hardware or software will often produce undesired reactions. These faults could result in failures, where the system as a whole does not complete an expected action. Failures can cause damage to the plant, its environment, or people in the vicinity of that plant [Blanke et al., 2001]. Fault Tolerant Control (FTC) aims to prevent failures and their consequences by providing adequate system performance in the presence of faults.

The majority of FTC research to date has concentrated on sensor faults. Significant advances have been made in this area, but most of these strategies are not applicable to actuator faults. This is attributable to the fundamental differences between actuators and sensors. Sensors deal with information, and measurements may be processed or replicated analytically to provide fault tolerance. Actuators, however, must deal with energy conversion, and as a result actuator redundancy is essential if fault tolerance is to be achieved in the presence of actuator faults. Actuation force will always be required to keep the system in control and bring it to the desired state [Patton, 1991].

The common solution for fault tolerant actuation in critical systems involves straightforward parallel replication of actuators. Each redundant actuator must be capable of performing the task alone and possibly override the other faulty actuators. This solution is over-engineered, reducing the efficiency of the system i.e. in triplex systems 200% more capability, cost and weight than required is introduced to ensure a certain level of reliability.

1.2 High Redundancy Actuation

High Redundancy Actuation (HRA) is a novel approach to actuator fault tolerance that aims to reduce the over-engineering incurred by traditional approaches. The HRA concept is inspired by musculature, where the tissue is composed of many individual cells, each of which provides a minute contribution to the overall contraction of the muscle. These characteristics allows the muscle, as a whole, to be highly resilient to individual cell damage.

This principle of co-operation in large numbers of low capability modules can be used in fault tolerant actuation to provide intrinsic fault tolerance. The HRA uses a high number of small actuator elements, assembled in parallel and series to form one high redundancy actuator (see Figure 1). Faults in elements will affect the maximum

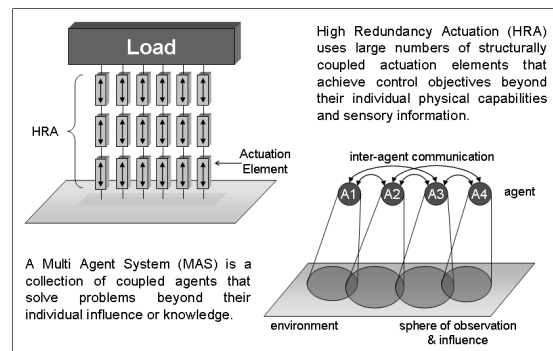


Figure 1. HRA and MAS.

capability, but through control techniques, the required performance can be maintained. This allows the same level of reliability to be attained in exchange for less over-dimensioning.

The HRA is an important new approach within the overall area of fault tolerant control. When applicable, it can provide actuators that have graceful degradation, and that continue to operate at close to nominal performance even in the presence of multiple faults in the elements.

1.3 Control of High Redundancy Actuation

The main focus of the HRA project thus far has utilised robust control methods. These techniques have been shown to be theoretically viable for fault tolerant control of low levels of redundancy [Du et al., 2007], and successful practical testing of these results on a two-by-two electromechanical HRA was achieved.

More recently, electromagnetic actuation has been used as elements of the HRA, the modelling of which in both

nominal and fault condition has been detailed in [Davies et al., 2008b]. Research is ongoing into the robust control of these elements at higher levels of redundancy [Steffen et al., 2007]. Results to date suggest that robust control should be a satisfactory method of achieving fault tolerant control of these structures. Indeed in most cases, the robust, passive¹ control approach is attractive, as its simplicity and constancy mitigate many of the associated problems with active control methods. However, research into more intelligent, active approaches is also an objective of the HRA project, to ascertain the levels of fault tolerance and nominal performance attainable in comparison to passive methods.

Multi-Agent Systems (MAS) are the focus of this active fault tolerance scheme. MAS was chosen as an intelligent approach to controlling the HRA as the two concepts are strongly related (Figure 1).

1.4 Overview

[Davies et al., 2008a] presented a Multi-Agent Control (MAC) scheme for a 4x4 HRA, which was found to be advantageous in terms of fault tolerance in comparison to a passive approach. However, it was questioned whether the approach would still provide tangible benefits at higher, more realistic levels of redundancy. Hence, this paper extends the application of MAC concepts to a 10x10 HRA to address this issue. In addition, the possibility of fault misdiagnosis is also considered. Section 2 briefly introduces agent concepts and discusses the rationale behind MAC of HRA. The current MAC scheme is described in Section 3. Section 4 then provides details of the control of a 10x10 HRA using passive and MAC means.

2. MULTI-AGENT CONTROL OF A HIGH REDUNDANCY ACTUATOR (MACHRA)

2.1 Multi-Agent Control

An agent is a physical or virtual entity situated in its environment, which acts autonomously and flexibly within its purview to achieve goals in a real-time manner [Jennings et al., 1998]. A MAS, therefore, is a collection of agents that are socially coupled and collaborate to achieve objectives, which in the case of MAC are the control objectives of the application.

These agent characteristics resemble the concept of closed-loop control, which achieves objectives through sensing and acting. However, there are important differences within the agent concept. The most obvious difference is the social interaction and negotiation that exists between agents. Also, the agent philosophy is strongly associated with localisation, a point emphasised by [Ferber, 1999].

2.2 Rationale for Multi-Agent Control of HRA

MAS and HRA are conceptually similar (Figure 1). Both are inspired by natural mechanisms which utilise vast numbers of relatively simple cells/processes to form complex structures/behaviours.

¹ In this context, passive refers to a static control structure and algorithm.

This similarity in their structuring is the key rationale for combining MA ideas with HRA. The structuring of control is often neglected in the field of control engineering as the problem is stated in the form of a single plant model [van Breemen and de Vries, 2000]. The process industry acknowledges that the structuring of control is an important issue when applied to a decomposed system, thus it is given more attention in this field and numerous MACS have been proposed in this application area e.g. [Wang and Wang, 1997].

The HRA is a complex, highly structured system, with well-defined interactions between simple elements. An unstructured approach will have difficulties dealing with this complexity. However, if the HRA is viewed as a collection of simpler, similar (if not identical), physically distributed modules, the complexity and changeable nature of the system's dynamics and structure can be handled at a local level, allowing objectives to be met with greater speed and efficiency. MASs facilitate the control of such decompositions, allowing simple control algorithms in conjunction with simple fault detection methods at a local level to achieve greater robustness and adaptability in fault situations.

Agents also avoid some of the issues associated with active control. Multiple-model control schemes often have one active global controller, and a supervisor that decides which controller should be active. A centralised supervisor becomes a single point of failure, increasing the systems reliance upon fault detection. In addition to this, a global view on the system can make faults more complex to diagnose. These centralisation issues are negated by MAC, as are issues associated with adaptive control.

The unpredictability of centralised adaptive control schemes should be alleviated by the decentralisation MAC offers. Undesirable changes within modules will affect the system as a whole to a lesser extent, perhaps even with other agents adapting to counter-balance the unwanted behaviours. Localisation of control may also improve on response speed issues associated with adaptive control.

Nonetheless, there are a number of potential issues associated with MASs that require careful attention such as deliberation, communication and negotiation delays, agent non-consensus and communication failure.

2.3 MACHRA Objectives

The HRA project's objectives include:

- Control of the elements resulting in a unified dynamic for the HRA.
- Nominal or acceptable behaviour of the HRA in element fault conditions.
- Graceful degradation of the HRA as fault levels increase beyond their critical point.

If the inclusion of intelligence in the control scheme is to be justified then the MA controlled HRA must achieve tangibly more in comparison to passive methods. Thus, the objective for MAC of an HRA also include:

Increased reliability - Robust techniques can be limited in the number of faults or fault types they can accommodate. The structure of the HRA alleviates this problem, as

the number of elements reduces the overall affect of faults on the system. Nevertheless, a more intelligent strategy may accommodate even greater fault levels and fault types.

Improved nominal performance - Passive fault accommodation methods require the controller design to be robust enough to produce adequate performance during faulty conditions. This can lead to conservative performance in nominal conditions. An active control scheme can offer an increase in nominal performance as the control action can be changed in fault situations.

3. MACHRA SCHEME

The MACHRA scheme is currently in the investigative stage, concentrating on parallel in series (PS) configurations with lock-up and loose faults. Initial agent architectures and agency structures have been designed and simulated.

At present, Matlab/Simulink is used to create and simulate HRA assemblies, details of which can be found in [Davies et al., 2008b]. Stateflow is used to simulate the inner rule-based logic of the agents and their communication. This provides a fast prototyping tool of the agents for use with Matlab/Simulink.

The agent configuration and internal structuring was detailed in [Davies et al., 2008a]. A brief overview of the MACHRA scheme is provided here.

3.1 Agency Architecture

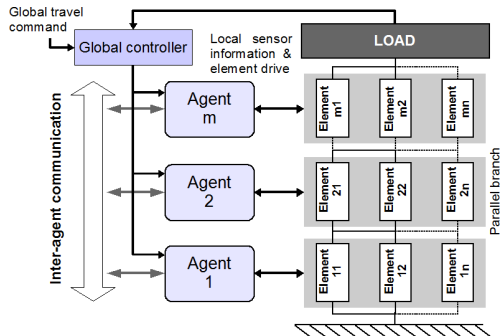


Figure 2. MACHRA agency architecture

The architecture of an agency is the configuration of multiple agents on a macro scale. Figure 2 displays the MACHRA scheme's agency architecture for a $m \times n$ HRA PS configuration. There is an agent per parallel branch of elements, each of which is responsible for the control and detection of faults within its elements and communication of faults to other agents.

All agents within this scheme are identical and peers, consistent with the spirit of MAC where no hierarchy should exist. A fixed outer control loop provides each agent with an identical set-point. Communication between agents is broadcasted via a bus. However, agents only consider messages from structural neighbours. If lock-up faults occur, the agent's structural neighbours will change and thus different messages become relevant.

3.2 Agent Architecture

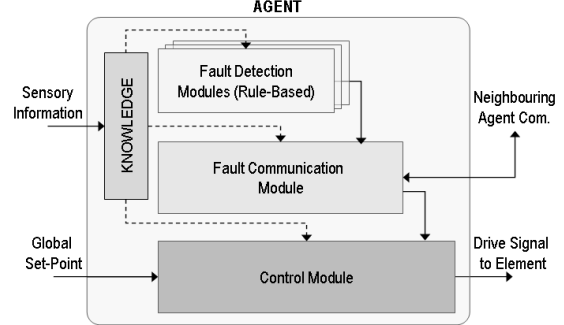


Figure 3. MACHRA agent architecture

The current agent architecture is illustrated in Figure 3. This architecture has similarities with subsumption, first introduced by [Brooks, 1986], that uses behaviours layered in order of abstraction to produce more complex emergent behaviors in a reactive time-frame. This reactivity is key in the HRA as, due to the fast dynamics of the electromagnetic elements, a purely deliberative architecture may not provide the response times needed.

The Fault Detection Module (FDM) is the most abstracted layer, and thus affects those below it. As its name suggests, the FDM detects faults in its elements. Currently, only one fault type (lock-up faults) is detected. Future agents will have more than one module, arranged either as peers in a single layer or as separate layers ordered by the severity of the fault type. The module contains rule-based logic which determines the fault status of the element based on sensory information and internal knowledge.

If a fault is detected, this information is passed to the Fault Communication Module (FCM) where it is relayed to other agents. Fault status messages from other agents are also received here.

The most reactive layer is the Control Module (CM), which provides the drive signal to the element based on the set-point, and its knowledge of the system status. A multiple-model control scheme is employed, as the CM contains a look-up table with simple classical control designs based on the number of active elements in the system.

Finally, a knowledge module containing both knowledge given to the agent on start-up and that deduced within the individual modules links the layers.

4. CONTROL OF A 10X10 HRA

This section will consider the control of a 10x10 HRA using MAC concepts and a passive control approach for comparison. [Davies et al., 2008a] gave an example of MAC applied to a 4x4 system. As this system had a relatively low level of modular redundancy in terms of the HRA concept, the effects of faults on the system were relatively large. A 10x10 system is a more appropriate level of redundancy for the HRA concept and thus it is worthwhile reconsidering the effectiveness of active FTC in a system where faults have less affect. In addition, the effects of reconfiguration

Table 1. Requirements

Performance Requirements	
Travel Window	$\pm 0.06\text{m}$ ($6 \times \textit{element travel}$)
Overshoot	$< 2\%$
Rise Time	$< 0.75\text{s}$
Settling Time	$< 1.20\text{s}$

Table 2. Fault Cases

Case	Description	HRA State
Nom.	All elements are healthy	Healthy & capable
FC1	Branch nearest load locked	Faulty, but capable
FC2	2 branches nearest load locked	Faulty, but capable
FC3	3 branches nearest load locked	Faulty, but capable
FC4	4 branches nearest load locked	Critical fault level

delays and fault detection errors will be considered in the MAC scheme.

4.1 Case Study System

The HRA system considered in this paper is, as previously stated, a 10x10 system in parallel-series (PS) configuration, which is structured as shown in Figure 2, with ten branches of ten parallel elements arranged serially.

The actuation elements currently being used within the project are SMAC electromagnetic actuators [SMAC, 2004]. The modelling of these actuators was considered in [Davies et al., 2008b], and will not be detailed here. A simplified 2 state element model is used in this example, making the overall system 20th order.

The control is designed to meet some transient requirements, suitable to the system's technology with good stability margins. These requirements are given in Table 1.

The PS configuration of HRA is most affected by lock-up faults, as a locked element will fix its whole parallel branch of elements from the preceding surface to the next. Loose faults are naturally accommodated by this structure, as parallel elements compensate for loose elements in the branch. Thus, lock-up faults are considered in this example.

It is assumed that this system is designed for an application with travel requirements that need at least 6 of the 10 parallel branches to be operational. Hence, up to 4 lock-up faults in separate branches would be tolerable in this case and this level of faults will be considered here. 1-4 faults are injected in a worst-case manner (in separate branches), as described in Table 2.

4.2 Control Schemes

Figure 4 represents both the passive control and MAC schemes.

The passive scheme has cascaded classical controllers designed to meet the control objectives in nominal conditions. The inner loops have a phase advance compensator controlling the local position of each parallel branch of elements. This spreads the travel between the elements equally. An outer loop controller is then included to control the overall travel of the HRA as a whole. Proportional-

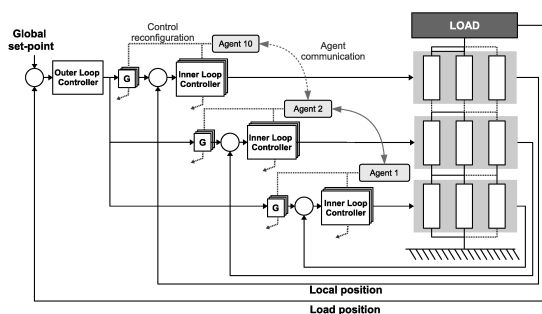


Figure 4. Global and agent control schemes

integral control is used in the outer loop to achieve the steady state requirements.

This passive control scheme is used as the base for the MAC approach. Under nominal conditions, the MA controlled system is identical to the passively controlled system. When a fault is detected by an agent, however, this fault is communicated throughout the agency and the control laws are changed. The outer loop is not reconfigured, as this would compromise the localisation of fault detection and reconfiguration decision, producing a single point of failure, as mentioned previously.

The feed-forward gain in the agent's control module is changed to redistribute the travel demand of the system i.e. if the system was nominal and one element locks then the gain would be changed from $1/10$ to $1/9$, as there are nine active element branches remaining.

In addition to this, the parameters in the local phase advance controller are also reconfigured. This is necessary as lock-ups in the system effectively increase the mass of the system: operational elements now have to work upon the dead mass of the faulty actuator as well as the load. An increase in the speed of the local controller can improve the performance of the remaining operational elements. Hence, in the agent's control module there is a look-up table of pre-computed control parameters based on the number of locked element branches in the system. In effect, this is a decentralised multiple-model control scheme, as there are a number of local controller designs based on fault models of the system.

It would also be possible to apply adaptive control using this approach. However, a multiple-model based approach was favoured as this aids verification of robustness and stability that would be necessary for high integrity applications for which HRA is intended for.

4.3 Simulation of Fault Cases

Figure 5 displays the response of the passively controlled and MAC schemes under nominal and faulty conditions as previously described in Table 2, when a step change of 0.05m in the reference was applied at $t=0$. All faults were introduced at the beginning of the simulation. Table 3 gives the gain margins and transient characteristics of these responses.

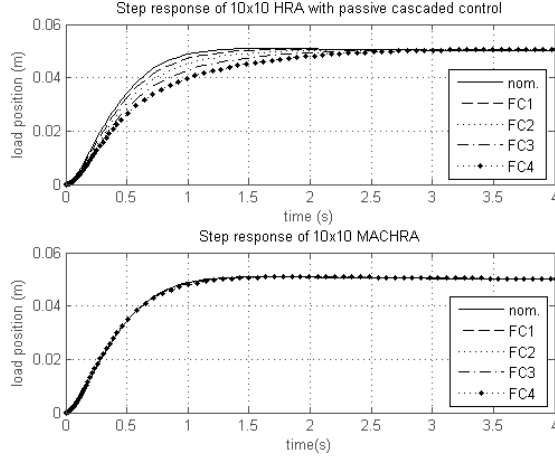


Figure 5. Step response of passive and MAC 10x10 HRA

Table 3. Passive control and MAC HRA

Fault Case	Over-shoot	Rise Time	Settling Time	Gain Margin	Phase Margin
Nominal	1.88%	0.68s	1.03s	∞	74deg
Passive FC1	1.01%	0.75s	1.20s	∞	76deg
Passive FC2	1.01%	0.88s	1.48s	∞	79deg
Passive FC3	1.01%	1.05s	1.85s	∞	82deg
Passive FC4	1.10%	1.33s	2.26s	∞	86deg
MACS FC1	1.68%	0.68s	1.04s	∞	74deg
MACS FC2	1.70%	0.68s	1.04s	∞	75deg
MACS FC3	1.96%	0.71s	1.07s	∞	75deg
MACS FC4	1.94%	0.73s	1.29s	∞	77deg

It can be observed that lock-up faults cause the system to slow. In the passive control case, the rise time and settling time increase significantly, and the requirements are not met when two or more actuation branches are locked.

In the MAC case, the increase in rise time and settling time can be reduced significantly, producing a response that is very similar to nominal conditions. The transient requirements are met in all fault conditions, apart from the settling time requirement in FC4.

These results illustrate that a MAC approach can provide near nominal performance in a realistically scaled HRA under realistic fault levels. This is an improvement on the passive control case.

4.4 Reconfiguration Delays

The MAC results given in Section 4.3 assumed that faults were detected and communicated instantaneously within the MAC architecture. This is not a realistic assumption. The detection of faults will take some finite period, as will the communication of these faults to the other agents. In addition, on receiving fault messages, the agents will take time to change their control parameters, and if multiple faults occur simultaneously, multiple messages get passed throughout the agency, and an agent will effectively step through these parameters until the final fault status has settled.

All of these effects must be considered in the simulation if the results are to resemble reality. Figure 6 shows the transient responses of a more realistic MAC 10x10 HRA in comparison to the previous passive control case and MAC without delays. The fault detection, communication and control reconfiguration are all simulated using Stateflow, which introduced delays into the system.

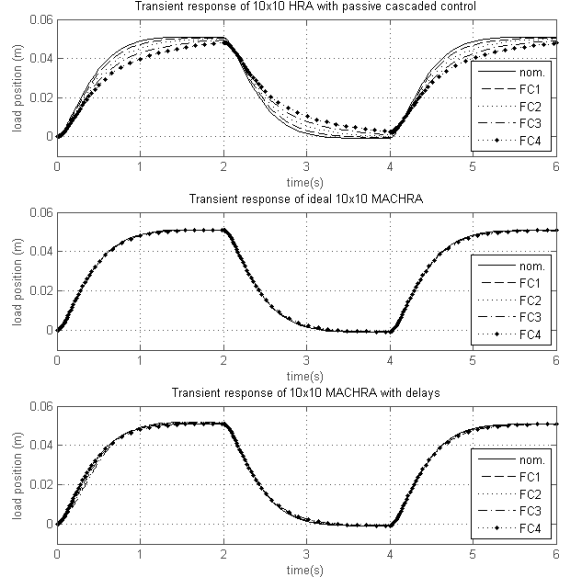


Figure 6. Transient response of passive, ideal MAC and MAC with delays

A square-wave input is applied to the system and all faults were injected at $t=0$. The response shows that in the first half period of the input, delay effects are present in the more realistic MAC scheme. However, after all faults are detected, communicated and control reconfigured the system's behaviour returns to that of the ideal MAC case.

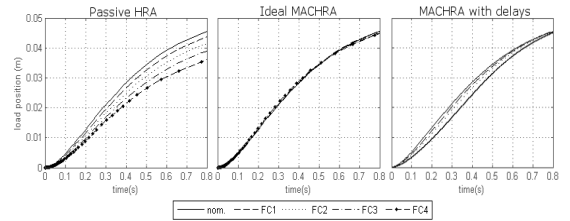


Figure 7. Initial response of passive, ideal MAC and MAC with delays

Figure 7 shows the initial response in more detail. Total reconfiguration of the system was attained after 0.35s. This delay increases the settling time and overshoot of the response in the first half period. The overshoot limit is exceeded in FC1, FC2 and FC3. If this was critical, then the agent's control reconfiguration could be adjusted to slow down reconfiguration, or reduce control gains until the fault state is stable. The effects of delays would also be

Table 4. False detection

No.of False Detects	1	2	3	4
Overshoot	1.88%	4.04%	6.16%	8.18%
Phase Margin	72deg	70deg	68deg	66deg

Table 5. False detection with reconfiguration

No.of False Detects	1	2	3	4
Overshoot	1.58%	1.51%	1.66%	1.57%
Phase Margin	74deg	74deg	74deg	75deg

lessened if the faults did not occur simultaneously, which is likely to be the case in a real situation.

4.5 Misdiagnosis in MACHRA

Misdiagnosis of faults in active FTC systems can be problematic. If the system adapts to a change that has not actually occurred in the system, then the results could degrade performance, cause faults or induce instability. Equally, if the system's control relies upon faults being detected and a fault is not detected then the results could be similar. Misdiagnosis of faults in this particular system will be considered briefly here.

Undetected faults should not cause problems in this particular scheme. At worst, the system's response will be that of the passive case. The system will become slower, but stability will be maintained. This is due to the outer loop control. If no outer loop was in place, the same response under working fault detection conditions could be achieved. However, an undetected fault would result in a significant steady state error for the overall HRA as the feed-forward agent control gains are not reconfigured.

False detection of faults in this MAC scheme will result in gain and inner control law changes, which could lead to instability. Table 4 gives the overshoot, gain and phase margins in the case of 1-4 false lock-up detections. The phase margin decreases, but the system retains stability. The overshoot, however, rises significantly. This is unlikely to be acceptable in an application, however four false detections may also be unlikely given a robust fault detection algorithm.

The flexibility of a MAC scheme can handle this problem through further reconfiguration. If the control law of the 'locked' agent is changed to force those elements into a locked state at time of detection, then this decrease of the stability margins can be avoided. This approach was applied and simulation results are shown in Table 5. On the triggering the FDM, the input reference of the agent is fixed to the local position at time of detection and the controller is changed to a PI compensator. This forces the system to behave as the detected fault case. Subsequently, the phase margin is not eroded and the overshoot limit achieved.

5. CONCLUSIONS AND OUTLOOK

The case for MAC of HRA has been made and the current MAC scheme described. It has been shown that, at this moderately high level of modular redundancy for HRA, MAC still provides significant benefits in comparison to passive control under realistic fault levels. Near nominal

performance can be maintained in worst case fault scenarios.

Reconfiguration delays in MAC can affect the response until full reconfiguration has been achieved. These effects may be considered acceptable, due to their ephemeral nature. Non-detection will result in the performance of a passive system. However, false detections will result in decreases in the stability margins. MAC offers a solution to this problem, by reconfiguring the control of agents that have detected a fault.

Practical testing of MAC on an experimental electromagnetic HRA is planned, which should give an indication of such a scheme's performance in a real-world situation.

ACKNOWLEDGEMENTS

This project is a cooperation of the Control Systems group at Loughborough University, the Systems Engineering and Innovation Centre (SEIC), and the actuator supply SMAC UK Ltd.. The project is funded by the UK's Engineering and Physical Sciences Research Council (EPSRC) under reference EP/D078350/1.

REFERENCES

- M. Blanke, M. Staroswiecki, and N. E. Wu. Concepts and methods in fault-tolerant control. *American Control Conference, Proceedings of the 2001*, 4, 2001.
- R.A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1): 14–23, 1986.
- J. Davies, T. Steffen, R. Dixon, and R. M. Goodall. Multi-agent control of a high redundancy actuator. *Control 08, Manchester*, 2008a.
- J. Davies, T. Steffen, R. Dixon, R. M. Goodall, A. C. Zolotas, and J. Pearson. Modelling of high redundancy actuation utilising multiple moving coil actuators. *IFAC World Congress, Seoul*, 2008b.
- X. Du, R. Dixon, R. M. Goodall, and A. C. Zolotas. Lqg control of a highly redundant actuator. In *Conference for Advanced Intelligent Mechatronics*, 2007.
- J. Ferber. *Multi-agent systems: an introduction to distributed artificial intelligence*. Addison-Wesley, United States, 1999.
- N. R. Jennings, K. Sycara, and M. Wooldridge. A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems*, 1(1):7–38, 1998.
- R. J. Patton. Fault detection and diagnosis in aerospace systems using analytical redundancy. *Computing and Control Engineering Journal*, 2(3), 1991.
- SMAC. Actuator data sheet - electric linear/rotary moving coil actuators la190-50, 2004.
- T. Steffen, J. Davies, R. Dixon, R. M. Goodall, and A. C. Zolotas. Using a series of moving coils as high redundancy actuator. In *IEEE Conference for Advanced Intelligent Mechatronics*, Zurich, 2007.
- A. van Breemen and T. de Vries. An agent-based framework for designing multi-controller systems. *International Conference on The Practical Applications of Intelligent Agents and Multi-Agent Technology, Manchester, 2000*, 2000.
- H. Wang and C. Wang. Intelligent agents in the nuclear industry. *IEEE Computer*, 30(11):28–34, 1997.

Fault Detection in High Redundancy Actuation using an Interacting Multiple-Model Approach

J. Davies * H. Tsunashima ** R.M. Goodall * R. Dixon * T. Steffen *

* Control Systems Group, Loughborough University, Loughborough, LE11 3TU, UK, <http://www.lboro.ac.uk/departments/el/research/scg>

** Department of Mechanical Engineering, College of Industrial Technology, Nihon University, 1-2-1 Izumi-cho, Narashino-shi, Chiba 275-8575, Japan

Abstract: The High Redundancy Actuation (HRA) project investigates the use of a relatively high number of small actuation elements, assembled in series and parallel in order to form a single actuator which has intrinsic fault tolerance. Both passive and active methods of fault tolerant control are being considered for use with the HRA. In either approach, some form of health monitoring is required to indicate the requirement for reconfiguration in the latter case and the need for maintenance in the former. This paper presents a method of detecting faults in a HRA using an Interacting multiple-model (IMM) algorithm.

Keywords: Fault detection, Fault diagnosis, Kalman filters, Multiple model

1. INTRODUCTION

1.1 Traditional Approaches to Fault Tolerant Actuation

In automated processes, faults in hardware or software will often produce undesired reactions. These faults can result in failures, where the system as a whole does not complete an expected action, possibly causing damage to the plant, its environment, or people in the vicinity of that plant [Blanke et al., 2001]. Fault Tolerant Control (FTC) aims to prevent failures and their consequences by providing adequate system performance in the presence of faults.

The majority of FTC research to date has concentrated on sensor faults. Significant advances have been made in this area, but most of these strategies are not applicable to actuator faults. This is attributable to the fundamental differences between actuators and sensors. Sensors deal with information, and measurements may be processed or replicated analytically to provide fault tolerance. Actuators, however, must deal with energy conversion, and as a result actuator redundancy is essential to keep the system in control and bring it to the desired state in the presence of actuator faults [Patton, 1991].

The common solution for fault tolerant actuation in critical systems involves straightforward parallel replication of actuators. Each redundant actuator must be capable of performing the task alone and possibly override the other faulty actuators. This solution is over-engineered, reducing the efficiency of the system i.e. in triplex systems 200% more capability, cost and weight than required is introduced to ensure a certain level of reliability.

1.2 High Redundancy Actuation

High Redundancy Actuation (HRA) is a novel, state-of-the-art approach to actuator fault tolerance that aims to reduce the over-engineering incurred by traditional approaches. The HRA concept is inspired by musculature, where the tissue is

composed of many individual cells, each of which provides a minute contribution to the overall contraction of the muscle. These characteristics allows the muscle, as a whole, to be highly resilient to individual cell damage.

This principle of co-operation in large numbers of low capability modules can be used in fault tolerant actuation to provide intrinsic fault tolerance. The HRA uses a high number of small actuator elements, assembled in parallel and series to form one high redundancy actuator (see Figure 1). Faults in elements will affect the maximum capability, but through control techniques, the required performance can be maintained. This concept allows the same level of reliability to be attained in exchange for less over-dimensioning.

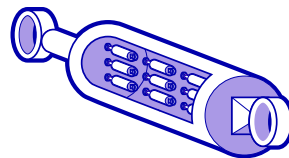


Figure 1. High Redundancy Actuation.

The HRA is an important new approach within the overall area of fault tolerant control. When applicable, it can provide actuators that operate at the desired level of performance in the presence of multiple faults in actuator elements, and gracefully degrade after the designed level of fault tolerance has been exceeded.

1.3 High Redundancy Actuation and Fault Detection

The project thus far has investigated two methods of controlling the HRA: robust control (passive fault tolerance) and reconfigured control (active fault tolerance) [Dixon et al., 2009, Davies et al., 2008a, Steffen et al., 2008]. Both of these approaches, to

different extents, require some form of fault detection (FD). In the latter case, a clear indication of the HRA's remaining capability, and thus its fault state is required in order to reconfigure the control laws appropriately. In passive control, the controller is static and thus not reliant on the fault state. However, health monitoring of the system is still required to indicate to a user the remaining capability of the HRA or indicate requirement for maintenance if fault levels approach the performance limits.

1.4 Overview

This paper presents an approach to fault detection for a HRA using an Interacting Multiple-Model (IMM) methods. Section 2 describes the modelling of HRA that uses electromagnetic actuation technology. The IMM algorithm is outlined in Section 3, and simulation results of its application to parallel, serial and mixed configuration elements are discussed in Section 4. Finally, conclusions are made in Section 5 and future work is considered.

2. HRA MODELLING

This paper assumes that the underlying technology of the HRA is electromagnetic, moving coil actuation, which is similar to a voice-coil in operation. Many other technologies are possible and indeed, the next stage of the project aims to address which technology will be best suited to manufacturing HRAs with large numbers of elements. However, many technologies will lead to a model with a similar structure to that presented here.

2.1 Single Element

The full order modelling of a moving coil actuator and of HRA configurations using these actuation elements is presented in [Davies et al., 2008b]. However, this paper will utilise a simplified version of this model.

A moving coil actuator typically comprises a coil wound round the centre pole of a magnetic assembly that produces a uniform magnetic field perpendicular to the current conducted in the coil. On providing a voltage, a current flows in the coil (inversely proportional to the input resistance R_{in}), which generates a force known as the LORENTZ force:

$$F = BNIl = kI = \frac{k}{R_{in}}v \quad (1)$$

Where B is the magnetic flux density, N is the number of turns and l is the length of the conductor. These are all constant and thus may be combined to form a single force constant, k . This force causes the coil, and the rod which is mounted to it, to move. The movement of the coil in the field generates a counter-electromotive force which can be expressed as below:

$$E = BNl\dot{x} = k\dot{x} \quad (2)$$

Where the derivative \dot{x} is the perpendicular component of the velocity of the wire relative to the flux lines i.e. the velocity of the coil.

The force produced by the electrical/magnetic part of the system acts upon the mechanical part which consists of the moving mass of the element and any stiffness and damping. Hence,

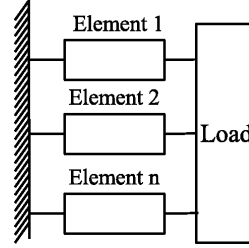


Figure 2. Parallel elements

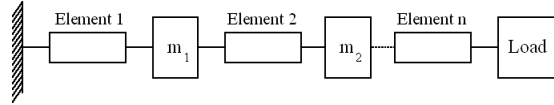


Figure 3. Serial elements

using NEWTON's second law of motion, the following second order model for the actuator can be derived:

$$m\ddot{x} = \frac{k}{R_{in}}v - \frac{k^2 + R_{in}d}{R_{in}}\dot{x} - rx \quad (3)$$

where v is the input voltage, m is the moving mass, d is the damping factor, and r is the stiffness. Choosing \dot{x} and x as states leads to the following state space model:

$$\begin{bmatrix} \ddot{x} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} -\frac{k^2 + R_{in}d}{R_{in}m} & -\frac{r}{m} \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ x \end{bmatrix} + \begin{bmatrix} \frac{k}{R_{in}m} \\ 0 \end{bmatrix} v \quad (4)$$

2.2 Parallel Elements

When elements are arranged in parallel (Figure 2), their forces act on a combined mass. Hence, assuming a common input voltage, the model for n parallel elements is:

$$\begin{bmatrix} \ddot{x} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} -\frac{\sum_{i=1}^n K_i}{m} & -\frac{\sum_{i=1}^n r_i}{m} \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ x \end{bmatrix} + \begin{bmatrix} \frac{\sum_{i=1}^n K_{in_i}}{m} \\ 0 \end{bmatrix} v \quad (5)$$

where:

$$K_i = \frac{k_i^2 + R_{in(i)}d_i}{R_{in(i)}} \text{ and } K_{in_i} = \frac{k_i}{R_{in(i)}}$$

2.3 Serial Elements

If a number of elements n are arranged serially (Figure 3) then the system contains n moving masses. Forces produced by the elements act not only on their moving mass, but also counter-act upon the preceding moving mass. The model of three elements in series is then:

$$\begin{bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \\ \ddot{x}_3 \\ \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} a_{13} & a_{14} & a_{15} & a_{16} & 0 & 0 \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} \\ 0 & 0 & a_{31} & a_{32} & a_{33} & a_{34} \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ 0 \\ 0 \\ 0 \end{bmatrix} v \quad (6)$$

where:

$$\begin{aligned}
a_{i1} &= \frac{K_{i-1}}{m_i}, a_{i2} = \frac{r_{i-1}}{m_i} \\
a_{i3} &= -\frac{K_i + K_{i+1}}{m_i}, a_{i4} = -\frac{r_i + r_{i+1}}{m_i} \\
a_{i1} &= \frac{K_{i+1}}{m_i}, a_{i2} = \frac{r_{i+1}}{m_i}, b_i = \frac{Kin_i - Kin_{i+1}}{m_i}
\end{aligned}$$

Models of higher numbers of serial elements follow this model's structure. Also, models of mixed configuration arrangements which are necessary for creating HRAs can be constructed using these basic equations.

2.4 Element faults

Three fault types are considered within this paper; overheating, loose faults and lock-up faults. Overheating of an actuation element may be represented as an increase in the resistance i.e. an increase in R_m .

A loose fault is where the actuation element loses the ability to translate force between its end points. Hence, a loose fault in a parallel assembly will reduce the force exerted on the mass. In serially connected elements, this fault is terminal as it effectively fails the whole serial branch. Thus, it is only useful to consider loose faults where elements are arranged in parallel.

A lock-up fault is where an element loses the ability to change the length between its two end points. This may occur if the coil of an actuation element is deformed and touches the magnet. This fixes the mass with respect to the reference point, and consequently the relative position and the speed are constant. In serially connected elements, this adds the mass of the locked element to the preceding element, and removes the mechanical states of that element from the system model. In parallel arrangements, this fault locks the whole assembly from end-to-end. Therefore, this fault type will only be considered where there are serial elements.

3. INTERACTING MULTIPLE-MODEL APPROACH

Conventional multiple-model estimation methods use a bank of filters, each of which is based on a model of the system when it is in a particular mode. The outputs of these filters are combined with a probabilistically weighted sum to achieve an overall state estimate.

However, there is no interaction between the filters, and as such the approach is not suited to situations where the parameters or structure of the system changes [Zhang and Jiang, 2001]. Nonetheless, non-interacting methods of multiple-model estimation have been applied to FD applications, where sudden parameter and structural changes to the system occur using ad hoc solutions [Menke and Maybeck, 1995, Napolitano and Swaim, 1991].

The Interacting multiple-model (IMM) method, developed in the field of tracking [Blom and Bar-Shalom, 1988, Bar-Shalom et al., 2001] deals with these issues. In the IMM approach, the initial estimate at the beginning of each iteration is a mixture of recent estimates from the filters. As a result the accuracy of estimation is increased and dependency on the previous mode history is introduced. This increases its suitability to detecting faults and thus it has been applied within this field [Mehra et al., 1998, Zhang and Jiang, 2001, Hayashi et al., 2006, Hashimoto et al., 2007, Hayashi et al., 2008].

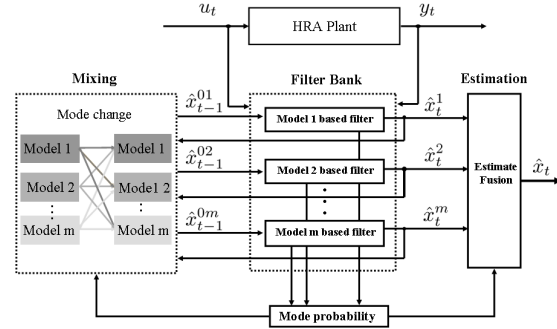


Figure 4. IMM estimation

3.1 IMM Estimation Algorithm

A depiction of the IMM estimation algorithm is shown in Figure 4. A number of filters (in this case Kalman filters) are designed based on m models of the system modes.

Also, a mode transition probability matrix p_{ij} is defined where the element ij represents the probability of transition from mode i to mode j . This may be based on knowledge of fault type frequency and likelihood when the system is in a certain state. The IMM algorithm has four main stages:

- Mixing
- Mode matched filtering
- Mode probability calculation
- Combination of estimates

Mixing The first stage of the IMM algorithm involves the mixing of all the filters estimated values and covariances from the previous iteration ($\hat{x}_{(t-1)}^i$ and $P_{(t-1)}^i$ for $i = 1 : m$) and the mixed probability, $\rho_{i|j(t-1)}$ to produce the input to the filters:

$$\hat{x}_{(t-1)}^{0j} = \sum_{i=1}^m \hat{x}_{(t-1)}^i \rho_{i|j(t-1)}, \quad j = 1, \dots, m \quad (7)$$

$$P_{(t-1)}^{0j} = \sum_{i=1}^m \rho_{i|j(t-1)} \left\{ \left[\hat{x}_{(t-1)}^i - \hat{x}_{(t-1)}^{0j} \right] \cdot \left[\hat{x}_{(t-1)}^i - \hat{x}_{(t-1)}^{0j} \right]^T \right\} \quad (8)$$

where $\rho_{i|j(t)}$ in the previous time step was calculated by:

$$\rho_{i|j(t-1)} = \frac{1}{\bar{c}_j} p_{ij} \rho_{i|j(t-1)}, \quad i, j = 1, \dots, m \quad (10)$$

$$\bar{c}_j = \sum_{i=1}^m p_{ij} \rho_{i|j(t-1)}, \quad j = 1, \dots, m \quad (11)$$

Mode matched filtering The Kalman filter algorithms are then obtained based on the discrete system. For a discrete system:

$$x_{(t+1)} = Fx_{(t)} + Gu_{(t)} + w_{(t)} \quad (12)$$

$$y_{(t)} = Hx_{(t)} + Lu_{(t)} + v_{(t)} \quad (13)$$

where $w_{(t)}$ and $v_{(t)}$ are the plant and measurement noise respectively with covariances of Q and R . Both are assumed to be

white Gaussian with zero mean. The Kalman filter algorithms can then be expressed as:

$$\hat{x}_{(t/t-1)}^j = F^j(\hat{x}_{(t-1/t-1)}^j) + D^j u_{(t-1)} \quad (14)$$

$$\hat{x}_{(t/t)}^j = \hat{x}_{(t/t-1)}^j + K_{(t)}^j [y_{(t)} - (H^j(\hat{x}_{(t/t-1)}^j) + L^j u_{(t)})] \quad (15)$$

$$K_{(t)}^j = P_{(t/t-1)}^j H_{(t/t-1)}^{jT} S_{(t)}^{j-1} \quad (16)$$

$$S_{(t)}^{j-1} = H_{(t/t-1)}^j P_{(t/t-1)}^j H_{(t/t-1)}^{jT} + R_{(t-1)}^j \quad (17)$$

$$P_{(t/t-1)}^j = F_{(t-1)}^j P_{(t-1)}^{0j} F_{(t-1)}^{jT} + G_{(t-1)}^j Q_{(t-1)}^j G_{(t-1)}^{jT} \quad (18)$$

$$P_{(t/t)}^j = P_{(t/t-1)}^j - K_{(t)}^j S_{(t)}^j K_{(t)}^{jT} \quad (19)$$

Mode Probability Calculation The mode probability, $p_{j(t)}$ (for mode j at time t) is then updated based on the likelihood function Λ for each mode filter:

$$p_{jt} = \frac{\Lambda_{j(t)} \bar{c}_j}{\sum_{i=1}^m \Lambda_{i(t)} \bar{c}_i} \quad (20)$$

$$\Lambda_{j(t)} = \left| 2\pi S_{(t)}^j \right|^{-\frac{1}{2}} \exp \left[-\frac{1}{2} \left(y_{(t)} - \left(H^j \hat{x}_{(t/t-1)}^j + L^j u_{(t)} \right) \right)^T \right. \quad (21)$$

$$\left. \cdot \left(S_{(t)}^j \right)^{-1} \left(y_{(t)} - \left(H^j \hat{x}_{(t/t-1)}^j + L^j u_{(t)} \right) \right) \right] \quad (22)$$

The mode probabilities give a time-varying estimate on the likelihood of the system state being one of the model-based modes and thus they are used in the indication of fault type for FD applications. The probabilities are smoothed using a moving average window.

Combination of Estimates Finally, the combined state estimate $\hat{x}_{(t)}$ and covariance $P_{(t)}$ are derived by weighting the estimated state and the mixed covariance for each mode with the mode probabilities:

$$\hat{x}_{(t)} = \sum_{j=1}^m \hat{x}_{(t)}^j p_{j(t)} \quad (23)$$

$$P_{(t)} = \sum_{j=1}^m p_{j(t)} \left[P_{(t)}^j + [\hat{x}_{(t)}^j - \hat{x}_{(t)}] \cdot [\hat{x}_{(t)}^j - \hat{x}_{(t)}]^T \right] \quad (24)$$

4. SIMULATION EXAMPLES

This section discusses the simulation results of the IMM approach when applied to first purely parallel and serial configurations to illustrate that overheating, loose and lock-up faults can be diagnosed in these structures. The diagnosis of faults in a mixed configuration of parallel and serial elements will then be considered briefly.

In each case the simulation is set-up as shown in Figure 5. The elements receive a shared input from a classical controller, designed for good transient characteristics and frequency margins from voltage input to load position. The system is given a sine wave input reference with an amplitude that uses its full range of travel ($\pm 15\text{mm}$). The known input and measured output is passed to the IMM algorithm which produces mode probabilities and a mixed state estimate.

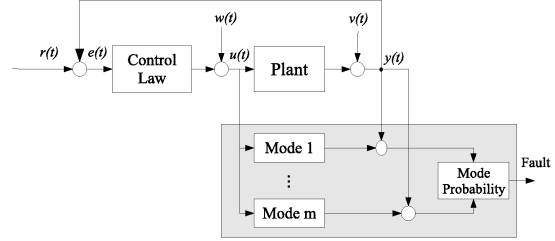


Figure 5. IMM Simulation

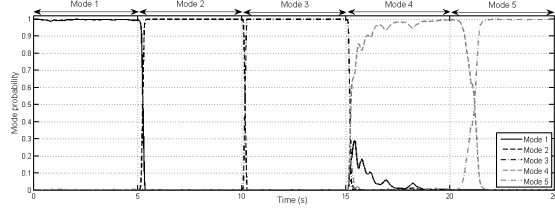


Figure 6. Mode probabilities for parallel elements

4.1 Parallel Elements

IMM FD is applied to three parallel elements here. The IMM estimator is designed based on the following modes:

- Mode 1: Nominal system
- Mode 2: Overheating, a resistance increase of 20%
- Mode 3: Overheating, a resistance increase of 50%
- Mode 4: 1 Loose element
- Mode 5: 2 Loose elements

The transition matrix p_{ij} is set such that the probability of no transition from the current state is 0.999 and 2.5×10^{-5} for transitions to the other modes.

The measured output is the position of the load. A very small value of covariance is used for the noise on the measured position ($5 \times 10^{-12}\text{m}$), as the glass encoder used has an rms noise value of $1\mu\text{m}$. The plant noise covariance Q is set at $1 \times 10^{-5}\text{V}$, as this gives a noise level in the order of mV.

Simulation Results The simulation results shown in Figure 6 are the mode probabilities produced from the IMM algorithm for the parallel elements system with changing fault state. At $t = 0$, the system is nominal, after which at 5s intervals the system fault state is changed from mode 2 through to mode 5.

It can be seen that during each fault state, the correct mode is diagnosed with a high probability after approximately 0.5s except mode 4. The probability of mode 4 takes longer to rise due to its similarity to the nominal state.

In a realistically scaled HRA, the levels of parallel redundancy (used in conjunction with serial redundancy) will be higher e.g. 10 or more parallel elements. A greater similarity between the nominal system and small proportions of loose faults will exist. This may make the clear diagnosis of low numbers of loose faults more difficult. However, if the behaviour of a HRA with a very low proportion of loose elements is sufficiently near the nominal behaviour, then detection of these faults at this fault level is not crucial, as the health status of the HRA will be high and control reconfiguration will not be necessary.

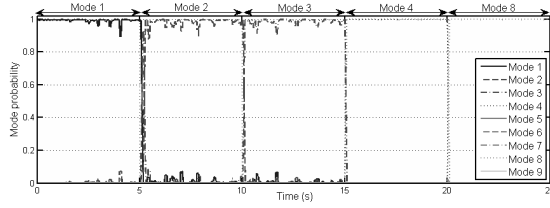


Figure 7. Mode probabilities for serial elements

4.2 Serial Elements

Three serial elements with overheating and lock-up faults are used in this example. As the location of a lock-up fault in the system will result in slightly different fault models, more modes are needed to diagnose lock-ups within serial elements. Therefore, the IMM estimator in this instance uses the following nine modes:

- Mode 1: Nominal system
- Mode 2: Overheating, a resistance increase of 20%
- Mode 3: Overheating, a resistance increase of 50%
- Mode 4: Element 1 lock-up
- Mode 5: Element 2 lock-up
- Mode 6: Element 3 lock-up
- Mode 7: Elements 1 and 2 lock-up
- Mode 8: Elements 1 and 3 lock-up
- Mode 9: Elements 2 and 3 lock-up

The transition matrix p_{ij} is set such that the probability of no transition from the current state is 0.999 and 1.25×10^{-5} for transitions to the other modes. Relative positions were used as the measured quantities in the simulation. Relative measurements were chosen over absolute as the HRA rig in development will have position encoders on each element. The noise covariance for each sensor the same as that used in the parallel example. The plant noise covariance is also the same as that used in the parallel case.

Estimation in the presence of actuator lock-ups presents a special issue, particularly when actuators are arranged in series. The fault model for a serial assembly of n actuation elements with one locked element will effectively be a model for $n-1$ elements with one mass augmented with the locked element's mass (if it is not the ground connected mass). When actuators lock at a non-zero point along their travel, this unknown position is not included in the fault model and thus position estimation and correct mode identification (without velocity information) becomes difficult.

One solution to this problem is to include in the fault model a high damping factor in the faulty element's dynamics. This will incorporate the locked position into the estimation resulting in a more accurate overall estimation and accurate mode identification. This approach is used within this simulation.

Simulation Results The resulting mode probabilities for an example fault profile simulation are shown in Figure 7. The mode probabilities in the example are typical of all fault profiles. The correct mode is clearly indicated in each time period. More fluctuation of the mode probabilities is present during nominal conditions and overheating in comparison with the parallel element results. This may be explained by the increased number of sensors in the system. In this case three sensors are

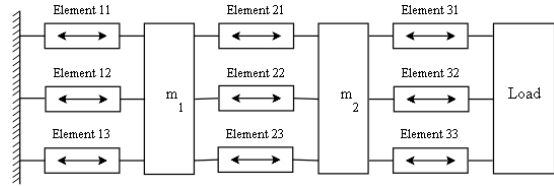


Figure 8. Parallel in Series 3×3 system

used, each measuring a smaller quantity than the one measurement in the parallel case. However, the same noise covariance is present on each sensor. Thus there is more noise present in the system. These fluctuations are less prevalent when actuation elements lock, as this fault mode is more removed from the nominal behaviour.

4.3 Parallel in Series HRA

Having illustrated that it is possible to diagnose overheating, loose faults and lock-up faults in purely parallel or serial arrangements of elements, the application of IMM FD to a system that contains both parallel and serial elements is briefly considered. A 3×3 Parallel in Series (PS) system (Figure 8) is used as an example.

This configuration has relatively high intrinsic tolerance to loose faults. Loose faults in the parallel branches will have little affect on the system until there are loose faults in every branch i.e. one loose fault in every parallel branch is equivalent to one loose branch in a purely parallel system. Hence, at least 2 loose faults (but at maximum 4 if they are divided equally between two branches) can occur before a reduction in force capability is observed.

The system has less tolerance to lock-up faults, however. A locked element will lock a whole parallel branch, reducing its travel capability by a third and thus the same fault tolerance is achieved as in a purely serial arrangement.

Many more mode filters are required to cover all the possible fault combinations within this system. As before, 3 modes are required to diagnose nominal conditions and two levels of overheating; 2 for diagnosing 2 reductions in force capability (i.e. loose faults within the system); and 6 modes for diagnosing travel capability reductions (lock-up faults). However, if we were to consider occasions where both force and travel capabilities are reduced, as would be necessary for a HRA, then the required number of modes rises to 23. Considering that this is a quite low level of redundancy, then this number is high. In a higher order, more realistically dimensioned HRA containing, for example 10×10 elements, then using the current approach to mode allocation, 2286 modes would be needed to diagnose all the fault combinations for up to 50% reduction in force and travel capabilities.

Simulation Results The higher number of modes in this example does not affect the diagnosis quality. An example simulation for the 3×3 PS system is shown in Figure 9. The system is nominal for the first 5s period, followed by a lock-up in element 1 at $t=5$. Loose faults resulting in a 1/3 loss of force capability are injected at $t=10$, and element 2 locks at $t=15$. Finally, another loose fault in the remaining unlocked branch occurs at $t=20$. In each case, the correct mode is diagnosed with short detection delays. The state in 15s-20s is more difficult

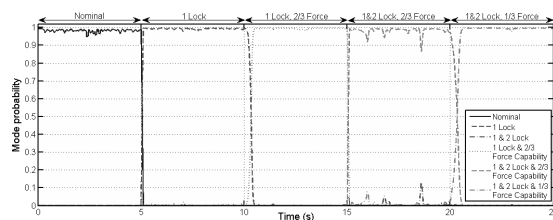


Figure 9. PS system mode probabilities (modes with low probabilities removed from plot for clarity)

to diagnose as it is similar to no loose faults and 1/3 force capability, but the correct mode is still clearly indicated. The higher number of modes, however, does affect the required simulation run-time for the IMM algorithm.

5. CONCLUSIONS

This paper discussed the utilisation of IMM techniques to achieve fault detection in a HRA. Simulation results of the IMM method applied to parallel elements, serial elements and a small mixed configuration HRA were presented. These results suggested that, using a comprehensive set of mode filters, it is possible to detect overheating faults; location independent loose element faults; and location specific lock-up faults. However, the required number of modes for detection in low level redundancy HRA is relatively large, and for more realistically dimensioned HRA (100+ elements) the required number becomes much greater. This may make real-time diagnosis unfeasible.

However, in HRA applications, the location of the locked element is not of interest. Only the actuator's remaining travel or force capability is required to give an indication of health, or reconfigure global control laws¹. Hence, if a simplified model of the system is used with the IMM algorithm, where each mode filter represents a level of capability, the number of required modes would be limited dramatically. This approach will be the focus of the next stage of work on this specific area of the HRA project's research. Also, the development of a 4×4 experimental rig for the HRA is underway and application of these fault detection techniques to further assess their feasibility is planned.

This approach to fault detection is by no means the only one that can be taken to meet the requirements of this application. Indeed, the project aims to examine other fault detection and health monitoring methods in the future. A comparison may then be made between the fault detection types to further assess the effectiveness and feasibility of using this IMM approach with the HRA.

ACKNOWLEDGEMENTS

This project is a cooperation of the Control Systems group at Loughborough University, the Systems Engineering and Innovation Centre (SEIC), and the actuator supply SMAC UK Ltd.. The project is funded by the UK's Engineering and Physical Sciences Research Council (EPSRC) under reference EP/D078350/1.

¹ Another solution to this problem is to decentralise the estimation. Localised element-level control and fault detection can be applied, which limits the required number of modes.

REFERENCES

- Y. Bar-Shalom, X. R. Li, T. Kirubarajan, and J. Wiley. *Estimation with applications to tracking and navigation*. Wiley New York, 2001.
- M. Blanke, M. Staroswiecki, and N. E. Wu. Concepts and methods in fault-tolerant control. *American Control Conference, Proceedings of the 2001*, 4, 2001.
- H. A. P. Blom and Y. Bar-Shalom. The interacting multiple model algorithm for systems with markovianswitching coefficients. *Automatic Control, IEEE Transactions on*, 33(8): 780–783, 1988.
- J. Davies, T. Steffen, R. Dixon, and R. M. Goodall. Multi-agent control of a high redundancy actuator. *Control 08, Manchester*, 2008a.
- J. Davies, T. Steffen, R. Dixon, R. M. Goodall, A. C. Zolotas, and J. Pearson. Modelling of high redundancy actuation utilising multiple moving coil actuators. *IFAC World Congress, Seoul*, 2008b.
- R. Dixon, T. Steffen, J. Davies, R.M. Goodall, A.C. Zolotas, and J. Pearson. Hra - intrinsically fault tolerant actuation through high redundancy. *SAFEPROCESS 09, Barcelona*, 2009.
- M. Hashimoto, S. Watanabe, and K. Takahashi. Sensor fault detection and isolation for a powered wheelchair. In *Advanced Intelligent Mechatronics, 2007 IEEE/ASME International Conference on*, pages 1–6, 2007.
- Y. Hayashi, H. Tsunashima, and Y. Marumo. Fault detection of railway vehicles using multiple model approach. In *SICE-ICASE, 2006. International Joint Conference*, pages 2812–2817, 2006.
- Y. Hayashi, H. Tsunashima, and Y. Marumo. Fault detection of railway vehicle suspension systems using multiple-model approach. *Journal of Mechanical Systems for Transportation and Logistics*, 1(1):88–99, 2008.
- R. Mehra, C. Rago, S. Seereeram, S. S. Co, and M. A. Woburn. Autonomous failure detection, identification and fault-tolerant estimation with aerospace applications. In *Aerospace Conference, 1998. Proceedings., IEEE*, volume 2, 1998.
- T. E. Menke and P. S. Maybeck. Sensor/actuator failure detection in the vista f-16 by multiplemodel adaptive estimation. *Aerospace and Electronic Systems, IEEE Transactions on*, 31(4):1218–1229, 1995.
- M. Napolitano and R. Swaim. New technique for aircraft flight control reconfiguration. *Journal of Guidance, Control, and Dynamics*, 14(1):184–190, 1991.
- R. J. Patton. Fault detection and diagnosis in aerospace systems using analytical redundancy. *Computing and Control Engineering Journal*, 2(3), 1991.
- T. S. Steffen, R. Dixon, R. M. Goodall, and A. C. Zolotas. Multi-variable control of a high redundancy actuator. In *Actuator, Bremen*, 2008.
- Y. Zhang and J. Jiang. Integrated active fault-tolerant control using imm approach. *Aerospace and Electronic Systems, IEEE Transactions on*, 37(4):1221–1235, 2001.

Active Versus Passive Fault Tolerant Control of a High Redundancy Actuator

Jessica Davies*, Thomas Steffen*, Roger Dixon, Roger Goodall, Argyrios Zolotas

(* joint first authors)

Control Systems Group,
Department of Electronic and Electrical Engineering,
Loughborough University, Loughborough, LE11 3TU, UK,
<http://www.lboro.ac.uk/departments/el/research/scg/>
j.davies, t.steffen, r.dixon, r.m.goodall, a.c.zolotas@lboro.ac.uk

Abstract— The High Redundancy Actuator (HRA project investigates the use of large numbers of small actuation elements to achieve fault tolerance. The large number of components involved poses a unique challenge from a control perspective. This paper presents the two main options to control the HRA: using robust control (passive fault tolerance), and reconfigurable control (active fault tolerance). The robust controller is designed using H_∞ methods, and handles the different system behaviours of the HRA with only small changes to the closed-loop system. In contrast, control reconfiguration detects the fault and changes the control laws accordingly. Multi-Agent System (MAS) concepts are used to apply localised multiple-model control and fault detection on an individual element level. The results of both approaches are compared to illustrate the trade-off between the complexity of the control approach and the resulting performance under different fault situations.

Index Terms—high redundancy actuator, fault-tolerant control, active fault tolerance, passive fault tolerance, fault accommodation, robust control, control reconfiguration, multi-agent systems.

I. HIGH REDUNDANCY ACTUATION

High Redundancy Actuation (HRA) is a new approach to fault tolerant actuation, where an actuator comprises a large number of actuation elements (see Figure 1). Faults in the individual elements can be accommodated without resulting in a failure of the complete actuation system.

The concept of the HRA is inspired by musculature. A muscle is composed of many individual cells, each of which provides only a minute contribution to the force and the travel of the muscle. The aim of this project is to use the same principle of co-operation of high levels of low capability elements to provide intrinsic fault tolerance.

An important feature of the HRA is that the actuator elements are connected both in parallel and in series. Serial elements allow the HRA to tolerate element lock-ups whilst parallel

elements allow tolerance of 'loose' failure modes. Clearly, a combination of serial and parallel elements will have a degree of tolerance to both. However, the post fault performance depends on how robust the HRA controller is to faults.

HRA model complexity presents a problem for typical multi-variable control approaches (see [1], [2]). Models that include each actuation element explicitly will inevitably be high order, particularly for the envisioned levels of modular redundancy e.g. 10x10 or more.

This paper presents two control concepts to deal with both the complexity of the system and with the occurrence of faults. The first concept uses robust control. The design of the robust controller can be performed with a reduced model, leading to a low complexity controller.

The second method is to use Multi-Agent System concepts to apply a decentralised active control and fault detection scheme. Each actuation element is controlled by an individual agent. Again, this leads to a low complexity controller, as only the dynamics of single element have to be considered. By detecting and communicating faults, this structure is able to respond to faults, and compensate their effect on the overall behaviour of the HRA.

Section 2 presents the model of the HRA. Sections 3 and 4 present the robust control and the multi-agent control approach. Example results are compared in Section 5, leading to the conclusion and outlook in Section 6.

II. HRA MODEL

This paper assumes that the underlying technology of the actuator is electromagnetic actuation, which is similar to a voice-coil in operation. Other technologies are possible, many of which lead to a similar model.

Single Element

An individual actuation element can be modelled as a spring-damper system, following Newton's second law of motion (see [3] for full details):

$$m\ddot{x} = ki - d\dot{x} - rx \quad , \quad (1)$$

where x is the position, m is the moving mass, k is the input coefficient, d is the damping factor (accounting for mechanical and electrical damping), r is the elasticity of the spring, i is the current input and \dot{x} is the position of the mass. Choosing x and \dot{x} as states leads to the following state space model:

$$\frac{d}{dt} \begin{pmatrix} \dot{x} \\ x \end{pmatrix} = \begin{pmatrix} -\frac{d}{m} & -\frac{r}{m} \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \dot{x} \\ x \end{pmatrix} + \begin{pmatrix} \frac{k}{m} \\ 0 \end{pmatrix} i \quad . \quad (2)$$

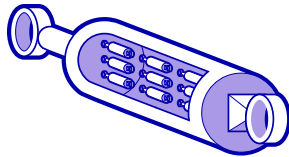


Figure 1. Configuration of a High Redundancy Actuator

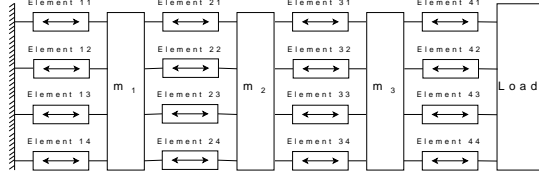


Figure 2. 4x4 Parallel-in-Series (PS) HRA

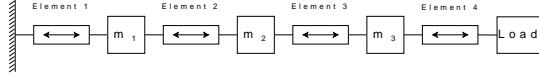


Figure 3. Simplified system of 4 serial elements

Nominal System

This paper will address the control of a 4×4 system arranged in parallel and series as shown in (Figure 2). As mentioned previously, the inclusion of each element's dynamics will increase the size of the model. However, each group of four parallel elements can be simplified to one equivalent (stronger) element, because they all act on the same moving mass (Figure 3). Using this simplification, the state-space model of the fault-less SISO system is

$$\frac{d}{dt}\mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \quad (3)$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} \quad (4)$$

with

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_{1,2}(m_1) & \mathbf{A}_2(m_1) & \mathbf{O} & \mathbf{O} \\ \mathbf{A}_2(m_2) & \mathbf{A}_{2,3}(m_2) & \mathbf{A}_3(m_2) & \mathbf{O} \\ \mathbf{O} & \mathbf{A}_3(m_3) & \mathbf{A}_{3,4}(m_3) & \mathbf{A}_4(m_3) \\ \mathbf{O} & \mathbf{O} & \mathbf{A}_4(m_3) & \mathbf{A}_{4,5}(m_4) \end{pmatrix} \quad (5)$$

$$\mathbf{B} = \begin{pmatrix} \frac{k_1 - k_2}{m_1} & 0 & \frac{k_2 - k_3}{m_2} & 0 & \frac{k_3 - k_4}{m_3} & 0 & \frac{k_4}{m_4} & 0 \end{pmatrix}^T \quad (6)$$

$$\mathbf{C} = (0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1) \quad (7)$$

where

$$\mathbf{A}_i(m) = \begin{pmatrix} \frac{d_i}{m} & \frac{r_i}{m} \\ 0 & 0 \end{pmatrix} \quad (8)$$

$$\mathbf{A}_{i,j}(m) = \begin{pmatrix} -\frac{d_i + d_j}{m} & -\frac{r_i + r_j}{m} \\ 1 & 0 \end{pmatrix} \quad (9)$$

are submatrices, $\mathbf{x} = (\dot{x}_1 \ x_1 \ \dot{x}_2 \ x_2 \ \dot{x}_3 \ x_3 \ \dot{x}_4 \ x_4)^T$ is the state, \mathbf{u} is the input, and \mathbf{y} is the output of the system. The parameter vectors used here are

$$\begin{aligned} \mathbf{m} &= (0.2 \ 0.2 \ 0.2 \ 1)^T \text{ kg} \\ \mathbf{d} &= (13 \ 12.5 \ 11.5 \ 10 \ 0)^T \text{ Ns/m} \\ \mathbf{r} &= (1.3 \ 1.25 \ 1.15 \ 1 \ 0)^T \text{ N/m} \\ \mathbf{k} &= (13 \ 12.5 \ 11.5 \ 10 \ 0)^T \text{ N/V} . \end{aligned}$$

The choice of the slightly different coefficients is deliberate, to compensate for the higher mass that the lower elements have

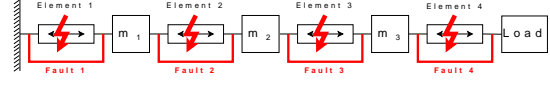


Figure 4. Lock-up faults

to move. This choice aligns the dynamics of the elements, so that they move synchronously (see [4] for further details). If all inputs receive the same value, this gives the simple SISO system transfer function

$$G_0(s) = 10 \frac{1}{(s + 2.4)(s + 0.104)} \quad (10)$$

The remaining six poles at -205 , -119 , -38.1 , -0.1003 , -0.1001 and -0.1 are cancelled with input de-coupling zeros due to the specific parameter choice.

Behaviour with Faults

This paper only addresses lock-up faults, as they have the more challenging effect on the dynamics compared to loose faults or faults in the electrical circuit of the coil for this configuration. If an element locks up, this means that the two masses it links are moving at the same speed, so that they can be considered as one larger mass (Figures 4).

The resulting 6 state model follows the same structure of the nominal model, but the parameters have to be rearranged according to the position of the fault. As a result, the pole-zero cancellation of the nominal system is no longer perfect, which means that the higher order modes have some limited influence on the behaviour. This is in addition to the obvious change in system amplification and in the position of the faster pole. For example, if element 4 locks, then the resulting transfer function is:

$$G_{F4}(s) = 9.58 \frac{s + 187.4 \ s + 64.9}{s + 187.7 \ s + 66} \frac{1}{(s + 3.03)(s + 0.103)} \quad (11)$$

In each fault location case, the resultant behaviour is very similar. See [5] for a more detailed discussion of the deviation introduced by the fault.

If two elements within the system lock, then the effect on the system will be greater, because the structure of the model changes. Essentially two pairs of states are unified, and all interactions with the remainder of the model have to be updated. However, as in the single fault case the resultant behaviours are similar regardless of the fault location. The SISO transfer functions for the lock-up of elements 3 and 4 is

$$G_{F34}(s) = 8.93 \frac{s + 129.9}{s + 131.9} \frac{1}{(s + 4.293)(s + 0.1024)} \quad (12)$$

III. ROBUST CONTROLLED HRA

The robust controller is designed using H_∞ loop shaping. This is a two-step process: first a classical controller is designed following rather conservative design rules, and then this controller is used as a weighting function for the design of

an H_∞ optimal controller, which further robustifies the initial controller.

The classical controller is designed as a PI controller with phase advance for the nominal model G_p . (This structure has the advantage of PID of being realisable, so no approximation is necessary in the implementation.) The two zeros are placed to cancel the two poles of the system. Both the nominal and the fault models are taken into account, so the average values -3 and -0.104 are chosen. The free pole is put at -100 to force a fast system response, and the gain is set to achieve critical damping. This leads to the PID controller

$$G_{PID}(s) = 252 \frac{s+3}{s+100} \frac{s+0.104}{s}. \quad (13)$$

While both the gain and the span of the phase advance compensator may seem rather high, this is not a practical problem, because the position (given in m) is measured optically with a resolution of $1 \mu\text{m}$.

This controller is then used as a weighting function for the H_∞ loop shaping design [6]. H_∞ loop shaping introduces further damping in the system which results to a robust stability radius of $e=0.63$. Of course, different weighting functions will result in different robust stability radius results that either emphasise further (increasing e) or less (decreasing e) robustness to coprime uncertainty.

The overall controller transfer function is

$$G_{opt}(s) = 305 \frac{s+102}{s+100} \frac{s+3}{s+2.996} \frac{s+2.88}{s+141} s+0.104. \quad (14)$$

The extra elements, compared to the original PID weight, introduced from the loop-shaping design are evident. Applying balanced truncation (or even by inspection) the controller size can be reduced down to a PID form. The reduced controller is thus

$$G_{red}(s) = 305 \frac{s+2.89}{s+141} \frac{s+0.104}{s}, \quad (15)$$

Such a reduction is possible when starting with an appropriate PID weighting function (i.e. appropriately robust behaviour). For completeness, Figure 5, illustrates the plant, target loop and actual loop designs. The difference between the optimal controller and reduced controller is very small.

The reduced controller is working as expected with the linear system. However, control saturation is an issue due to the fast speed of response. The phase advance filter creates a short but very high spike, and standard anti-windup measures are not sufficient to maintain a good step response. Shaping of the reference trajectory could prevent this, but a simpler solution is proposed here: the unrealised input is saved in an integrator and released later when the limits allow it. The resulting controller shown in Figure 6 achieves a good step response across different amplitudes.

IV. MULTI-AGENT CONTROLLED HRA

An agent is a physical or virtual entity situated in its environment, which acts autonomously and flexibly within its purview to achieve goals in a real-time manner [7]. A Multi-Agent System (MAS), therefore, is a collection of agents

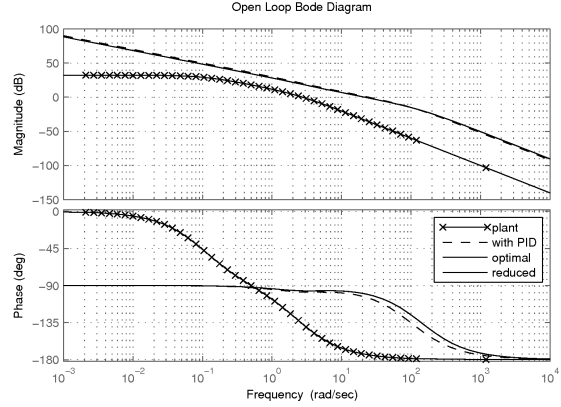


Figure 5. Behaviour with Different Controllers

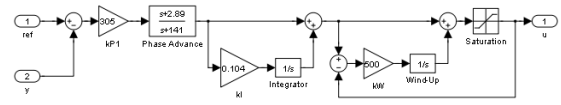


Figure 6. Implementation of the robust controller

that are socially coupled and collaborate to achieve objectives, which in the case of Multi-Agent Control (MAC) are the control objectives of the application.

These agent characteristics resemble the concept of closed-loop control. However, there are important differences within the agent concept such as social interaction and negotiation. Also, the agent philosophy is strongly associated with localisation, a point emphasised by [8].

MAS concepts are used as an intelligent approach to controlling the HRA as the two concepts are strongly related. They both use large numbers of simple elements/processes, coupled structurally or by communication, to achieve objectives that are beyond the capability and sensory knowledge of the individual parts.

This similarity in their structuring is the key rationale for combining MAS ideas with HRA (a fuller discussion is given in [9]). The complexity and changeable nature of HRA can be handled at a local level if it is viewed as a collection of simpler, similar, physically distributed modules. MASs facilitate the control of such decompositions, allowing the application of simple control algorithms in conjunction with simple fault detection methods at a local level to achieve greater robustness and adaptability in fault situations.

This decentralisation also provides advantages in comparison to other active fault tolerant control methods. There is no single point of failure as in systems with supervisors, and the affects of possible mis-reconfiguration are reduced.

MAC Structure

Figure 7 shows the control configuration of the MAC scheme used in this example.

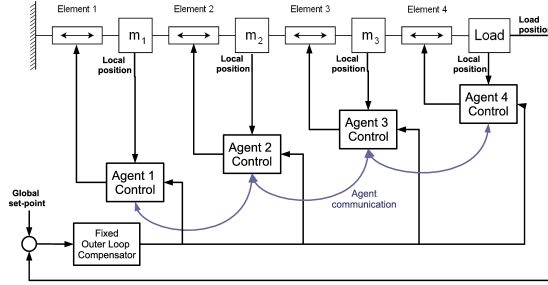


Figure 7. MAC scheme

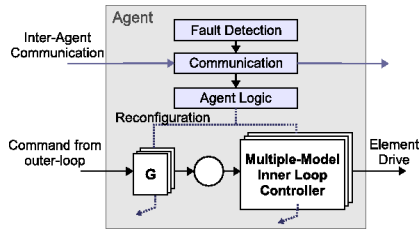


Figure 8. Agent architecture

There is a fixed outer-loop compensator which controls the overall position of the HRA using a load position measurement. This provides a command to the inner-loop agent controllers, which control the local position of each element (or bank of parallel elements). Figure 8 illustrates the agent architecture. There is a feedforward gain that distributes the command between the active agents and an inner-loop controller based on a number of fault models of the system. Hence, this MAC scheme is effectively a decentralised gain scheduling and multiple-model control scheme.

The agent uses its local sensory information to detect faults in its element using simple rule-based logic. On detecting a fault, this is communicated to the other agents neighbour-to-neighbour. If a fault message is received, the agent updates its health status knowledge and reconfigures its control. The feedforward gain is adjusted to redistribute the input between the remaining active agents and the inner-loop compensator is reconfigured using a look-up table of pre-computed controller parameters based on the number of active elements in the system.

MAC Design

The MAC controllers are designed to match the behaviour of the robust control approach under nominal conditions. The final phase advance compensator used in the robust approach is used as the inner-loop control under nominal conditions. A PI controller for the outer loop is then designed to match the behaviour of the robust control scheme through manual tuning. The following PI controller was found to provide a

Table I
TRANSIENT CHARACTERISTICS

	Rise Time	Settling Time	Over-shoot	Steady-State Error
Robust Nom.	0.082s	0.14s	1.26%	0
Robust 1 Fault	0.088s	0.19s	0%	0
Robust 2 Faults	0.13s	0.56s	0%	0
MAC Nom.	0.084s	0.14s	0.96%	0
MAC 1 Fault	0.084s	0.14s	0.41%	0
MAC 2 Faults	0.089s	0.15s	0.40%	0

good match:

$$G_{MACPI}(s) = 22.5 \frac{0.165s + 1}{s}, \quad (16)$$

When a fault is detected, the remaining agents reschedule the feedforward gain from 1/4 to 1/3. The inner-loop phase advance controller's time constant, τ is decreased to 80% of its original value. This reconfiguration retains the nominal system performance. If two faults are detected the feedforward gain is changed to 1/2, as two active elements remain, and τ , is decreased to 60%. Again, this reconfiguration provides a response very close to that of the nominal case.

As in the robust control case, whilst this control scheme works well with a linear system, control saturation is an issue. This can be remedied by applying the same approach described in Figure 6 to each inner-loop branch, or by using a simple rate limiter with each inner-loop compensator.

V. EXAMPLE SIMULATION

This section will consider the performance of the two proposed control approaches in nominal and faulty conditions. Two fault cases are considered, the first where one element is locked (thus locking its entire parallel branch of elements), and the second where two separate element branches become locked. The location of the locked elements within the HRA is of little importance, as was discussed in Section II. The faults were introduced into the system at $t=0$ and a constant input reference of 0.1m was applied. The limits of the system are included within the simulation, and the anti-wind up strategies described earlier were in place.

Simulation Results

Figure 9 gives the resulting response of these simulations and Table I summarises their transient characteristics. The individual travels of each element in the HRA are also given in Figure 10.

It can be observed from Figure 9 that the response of the passive robust controller and the MAC scheme are very similar. Figure 10 shows that, in the nominal case, the individual elements in both control approaches move in unison.

This unified dynamic is lost in both cases when faults occur. The distribution of travel amongst the elements is equal, however, their velocity differs as the pole-zero cancellation in the model is not perfect in the fault case. The MAC scheme has the potential to compensate for this, but, this would increase the number of pre-computed control laws needed significantly,

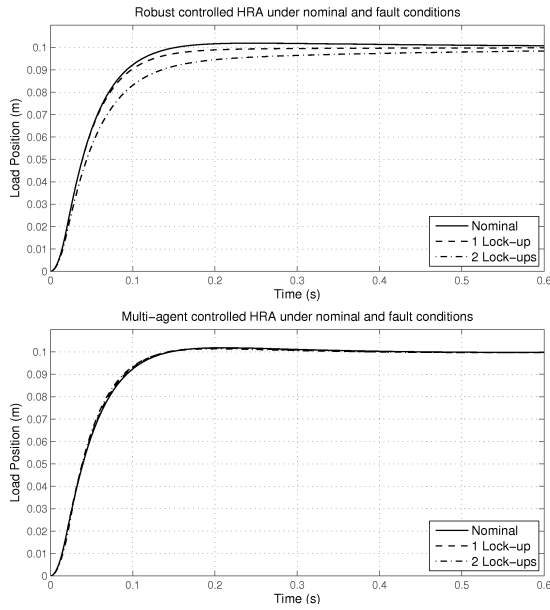


Figure 9. Step response of robust controlled and multi-agent controlled HRA in nominal and faulty conditions

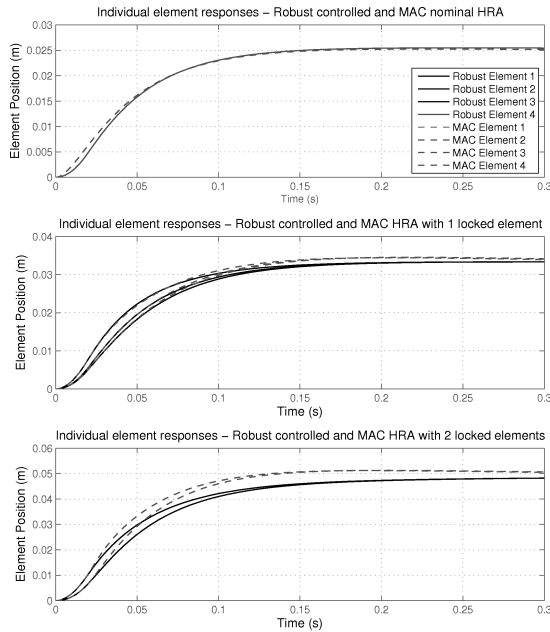


Figure 10. Individual element response of robust controlled and multi-agent controlled HRA in nominal and faulty conditions

making the scheme more complicated and increasing the verification effort required for high integrity applications.

The overall dynamic of the system is a more pertinent consideration in the HRA concept. When one element lock-up occurs in the HRA, the effective load within the system increases slowing the response. In the robust control case, a slight increase in the rise and settling time occurs, and the overshoot diminishes. The change in behaviour is very small, and most likely tolerable in an application. However, two lock-up faults within the system causes a more dramatic rise in the settling time of the robust control scheme.

The MAC approach provides a response that is very close to the nominal behaviour under both fault conditions.

Delays in Active Control

The simulation of multi-agent control provided in this example is idealised. No delays were incurred in the detection of faults, their communication or reconfiguration. This is not a realistic assumption as these tasks will require a finite amount of time. Figure 11 gives the transient response to a square-wave input of the passive robust controlled HRA, the idealised MAC scheme and a MAC scheme with its fault detection and reconfiguration delays simulated in the Stateflow toolbox.

With one lock-up in the non-ideal MAC, the fault is detected at $t=0.05s$ and all communication and reconfiguration is completed by $0.1s$. This delay causes a slight deterioration in the performance in the first transient. The subsequent behaviour is that of the ideal MAC, as no delays are incurred when the fault status is unchanging and reconfiguration complete.

The effects of delay are more pronounced in the two fault case. This is because two faults now have to be detected and communicated, and the active agents will step through the control algorithms as the fault messages spread through the system. Both faults were detected at $t=0.06s$, but the full communication of faults and reconfiguration took longer than previously, and is completed by $0.125s$. The control algorithm changes during this period cause the 'bend' present in the response, and some overshoot occurs. It can be observed that the response after this first transient is that of the ideal case.

If the overshoot induced was critical, then the agent's control reconfiguration could be adjusted to slow down the control algorithm changes, or reduce control gains until the fault state is stable. The affects of delays would also be lessened if the faults did not occur simultaneously, which is likely to be the case in a real situation.

Simulation Conclusions

The simulation results show that both robust control and MAC provide fault tolerance to lock-up faults within a HRA. The robust control is simple to implement, however there is a slight difference in closed loop behaviour between the performance under nominal and one fault conditions. This deviation may be tolerable in a real application. However, in the case of two lock-up faults, the difference in the closed loop behaviour becomes more pronounced, and may not be acceptable. The effects of faults could be reduced by tightening

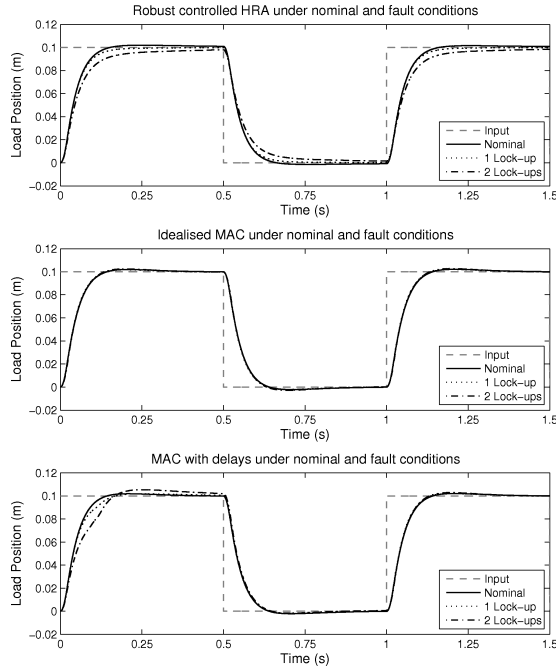


Figure 11. Response to a square-wave input of robust controlled HRA and idealised and delayed MAC

the control loop (which would require further and more accurate sensor readings), but it cannot be eliminated completely.

In contrast, nominal behaviour after faults have occurred can be restored with MAC (within the mechanical limits of the system). This advantage is gained in exchange for reliance on fault detection and a more complex control structure. However, the control algorithms remain simple and the criticality of fault detection can be reduced as discussed in [10].

Delays in the MAC scheme cause a deterioration in the performance during reconfiguration and for a short period after. These effects are minimal with one fault, but become more pronounced when there are two. Again, these effects can be reduced if the number of elements in the HRA is increased.

The suitability of the proposed control approaches to a given application depends on the dimension of the system, its performance requirements under nominal and fault conditions and the required extent of fault tolerance. The passive robust control approach is attractive in its simplicity and constancy, allowing it to be easily verifiable for high integrity applications. However, nominal performance levels will not be achievable in the presence of faults, and the extent to which faults are tolerated may be restricted. Nonetheless, this is sufficient for HRA, as the concept does not require the actuator to be fully operational with fault levels above its designed level of redundancy.

If performance was critical at higher fault levels, then active

control could provide this as well as near nominal performance at low fault levels. This adds complexity and makes verification more difficult however. Although, with an agent approach, high levels of modular redundancy in the HRA will reduce some of the negative effects produced by control reconfiguration.

VI. CONCLUSIONS AND FURTHER RESEARCH

Two approaches to controlling a High Redundancy Actuator have been described: a passive, robust control approach and an active, Multi-Agent inspired control scheme. These approaches were applied to an example 4×4 HRA, and simulations showed that both methods can provide fault tolerance. However, the level of fault tolerance provided differs as does their level of complexity.

The trade-off between complexity and control performance under faults does not just include these two options, but it is almost a continuous field of increasingly complex control structures. An adaptive controller for example could be used as an active fault tolerant approach. This would compensate for most of the behavioural differences introduced by the fault, without requiring a decentralised control or fault detection.

Further research will continue to explore this compromise, especially for higher order systems (such as a 10×10 configuration), and from the perspective of health monitoring.

VII. ACKNOWLEDGEMENTS

This project is a cooperation of the Control Systems group at Loughborough University, the Systems Engineering and Innovation Centre (SEIC), and the actuator supplier SMAC Europe limited. The project is funded by the Engineering and Physical Sciences Research Council (EPSRC) of the UK under reference EP/D078350/1.

REFERENCES

- [1] X. Du, R. Dixon, R. M. Goodall, and A. C. Zolotas, "Assessment of strategies for control of high redundancy actuators," in *Proceedings of the ACTUATOR 2006*, 2006.
- [2] X. Du, R. Dixon, R. M. Goodall, and A. C. Zolotas, "Lqg control for a high redundancy actuator," in *Proceedings of the 2007 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2007.
- [3] J. Davies, T. Steffen, R. Dixon, R. M. Goodall, A. C. Zolotas, and J. Pearson, "Modelling of high redundancy actuation utilising multiple moving coil actuators," in *Proceedings of the IFAC World Congress 2008*, Jul 6-11 2008.
- [4] T. Steffen, R. Dixon, R. M. Goodall, and A. Zolotas, "Multi-variable control of a high redundancy actuator," in *Actuator 2008 - International Conference and Exhibition on New Actuators and Drive Systems - Conference Proceedings*, pp. 473-476, HVG, 2008.
- [5] T. Steffen, R. Dixon, R. M. Goodall, and A. C. Zolotas, "Robust control of a high redundancy actuator," in *Proceedings of the UKACC Control Conference 2008*, Sep 2-4 2008. Tu09.06/p149.
- [6] D. McFarlane, K. Glover, B. H. P. Res, and V. Clayton, "A loop-shaping design procedure using hinf synthesis," *Automatic Control, IEEE Transactions on*, vol. 37, no. 6, pp. 759-769, 1992.
- [7] N. R. Jennings, K. Sycara, and M. Wooldridge, "A roadmap of agent research and development," *Autonomous Agents and Multi-Agent Systems*, vol. 1, no. 1, pp. 7-38, 1998. ID: 194.
- [8] J. Ferber, *Multi-agent systems: an introduction to distributed artificial intelligence*. United States: Addison-Wesley, 1999. ID: 212.
- [9] J. Davies, T. Steffen, R. Dixon, and R. M. Goodall, "Multi-agent control of high redundancy actuation," in *Proceedings of UKACC International Conference on Control*, 2008 2008.
- [10] J. Davies, T. Steffen, R. Dixon, and R. M. Goodall, "Multi-agent control of a 10×10 high redundancy actuator," in *Submitted to 23rd IAR Workshop on Advanced Control and Diagnosis*, 2008.