

# 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  $H_\infty$  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

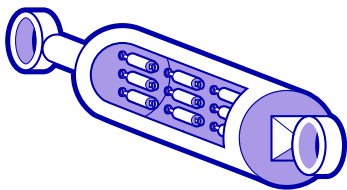


Figure 1. Configuration of a High Redundancy Actuator

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:

$$\frac{d}{dt} \begin{pmatrix} \dot{x} \\ x \end{pmatrix} = \begin{pmatrix} -\frac{d}{m} & -\frac{r}{m} \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \dot{x} \\ x \end{pmatrix} + \begin{pmatrix} \frac{k}{m} \\ 0 \end{pmatrix} i \quad . \quad (2)$$

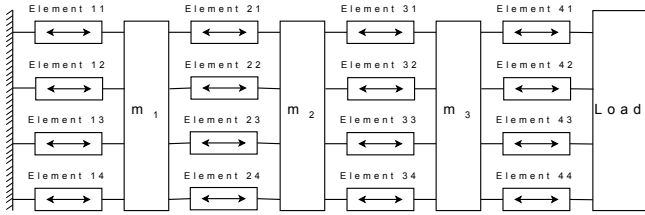


Figure 2. 4x4 Parallel-in-Series (PS) HRA

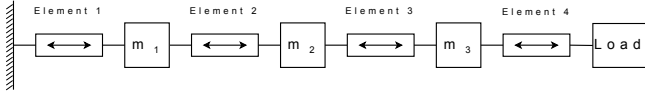


Figure 3. Simplified system of 4 serial elements

*Nominal System*

This paper will address the control of a  $4 \times 4$  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

$$\frac{d}{dt} \mathbf{x} = \mathbf{Ax} + \mathbf{Bu} \tag{3}$$

$$\mathbf{y} = \mathbf{Cx} \tag{4}$$

with

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_{1,2}(m_1) & \mathbf{A}_2(m_1) & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{2,3}(m_2) & \mathbf{A}_{2,3}(m_2) & \mathbf{A}_3(m_2) & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_3(m_3) & \mathbf{A}_{3,4}(m_3) & \mathbf{A}_4(m_3) \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_4(m_3) & \mathbf{A}_{4,5}(m_4) \end{pmatrix} \tag{5}$$

$$\mathbf{B} = \begin{pmatrix} k_1 - k_2 & 0 & k_2 - k_3 & 0 & k_3 - k_4 & 0 & k_4 & 0 \end{pmatrix}^T \tag{6}$$

$$\mathbf{C} = (0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1) \tag{7}$$

where

$$\mathbf{A}_i(m) = \begin{pmatrix} \frac{d_i}{m} & \frac{r_i}{m} \\ 0 & 0 \end{pmatrix} \tag{8}$$

$$\mathbf{A}_{i,j}(m) = \begin{pmatrix} -\frac{d_i + d_j}{m} & -\frac{r_i + r_j}{m} \\ 1 & 0 \end{pmatrix} \tag{9}$$

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

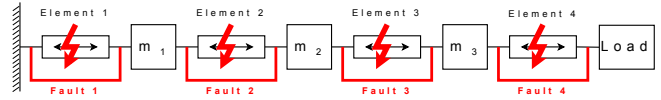


Figure 4. Lock-up faults

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

$$G_0(s) = 10 \frac{1}{(s+2.4)(s+0.104)} \tag{10}$$

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:

$$G_{F4}(s) = 9.58 \frac{s+187.4 \ s+64.9}{s+187.7 \ s+66} \frac{1}{(s+3.03)(s+0.103)} \tag{11}$$

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

$$G_{F34}(s) = 8.93 \frac{s+129.9}{s+131.9} \frac{1}{(s+4.293)(s+0.1024)} \tag{12}$$

III. ROBUST CONTROLLED HRA

The robust controller is designed using  $H_\infty$  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  $H_\infty$  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  $H_\infty$  loop shaping design [6].  $H_\infty$  loop shaping introduces further damping in the system which results to a robust stability radius of  $\epsilon=0.63$ . Of course, different weighting functions will result in different robust stability radius results that either emphasise further (increasing  $\epsilon$ ) or less (decreasing  $\epsilon$ ) robustness to coprime uncertainty.

The overall controller transfer function is

$$G_{opt}(s) = 305 \frac{s+102}{s+100} \frac{s+3}{s+2.996} \frac{s+2.88}{s+144} s+0.104. \quad (14)$$

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

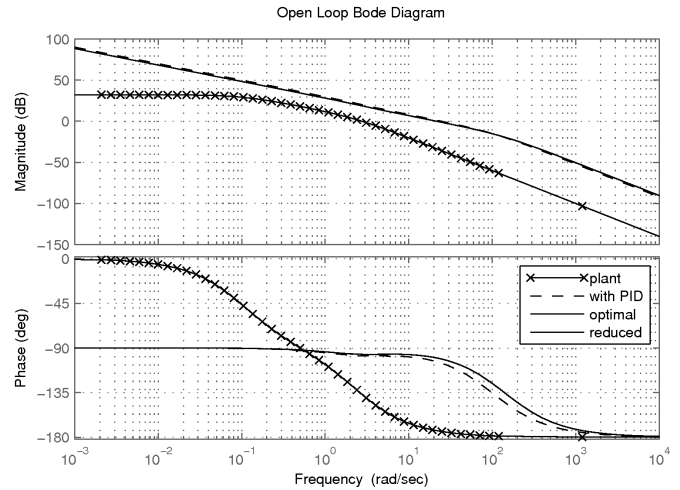


Figure 5. Behaviour with Different Controllers

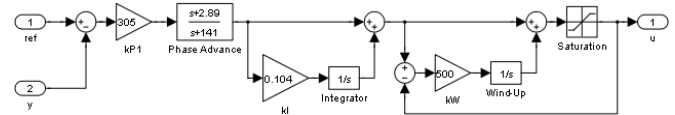


Figure 6. Implementation of the robust controller

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.

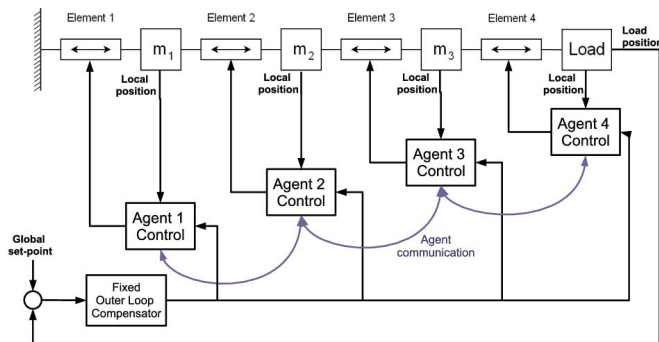


Figure 7. MAC scheme

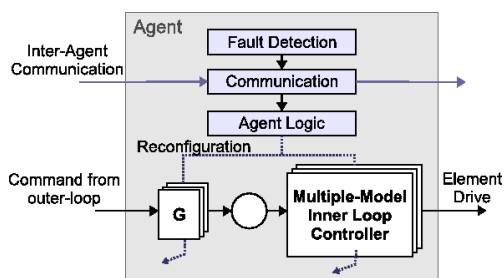


Figure 8. Agent architecture

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,  $\tau$  is decreased to 80% of it’s 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  $\tau$ , 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,

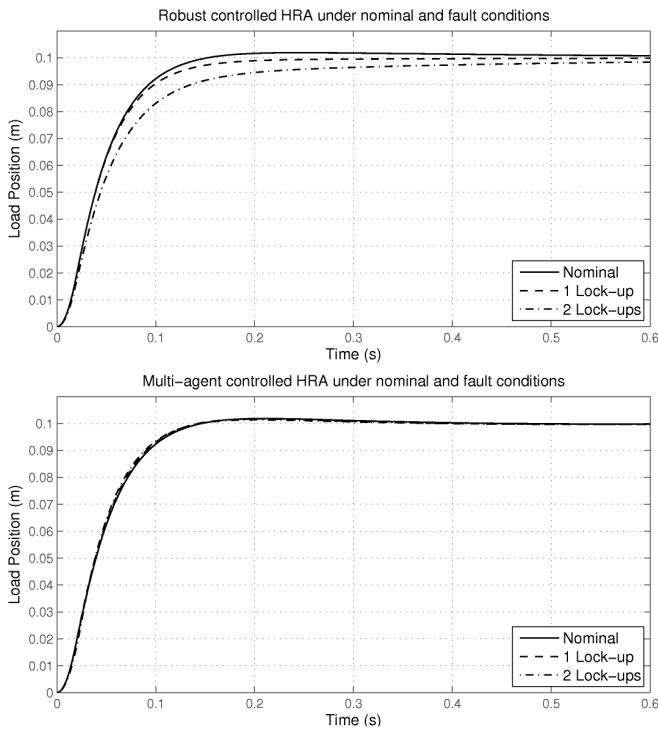


Figure 9. Step response of robust controlled and multi-agent controlled HRA in nominal and faulty conditions

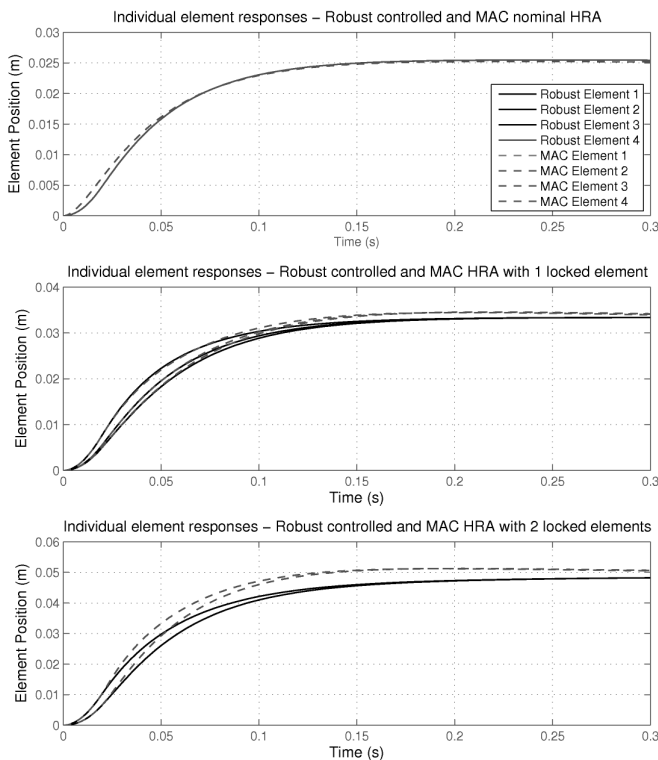


Figure 10. Individual element response of robust controlled and multi-agent controlled HRA in nominal and faulty conditions

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

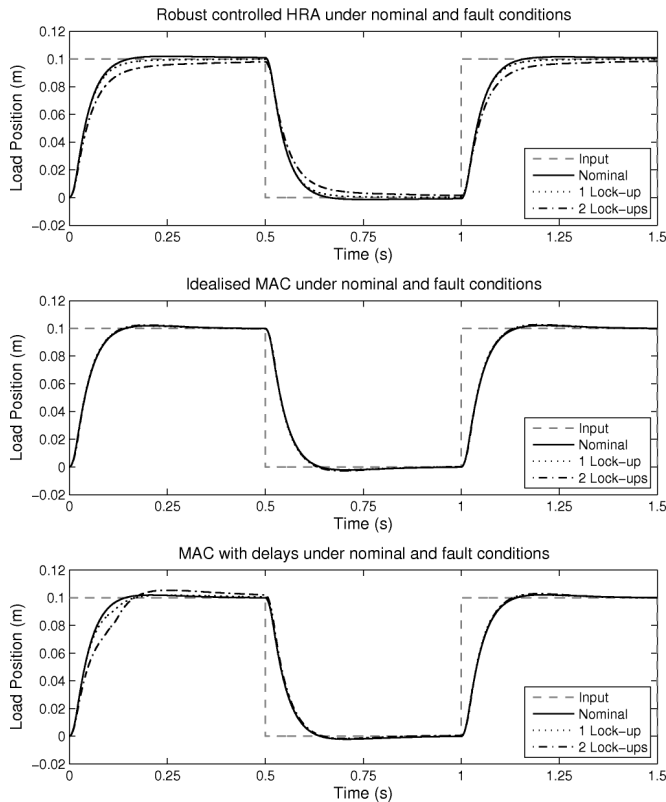


Figure 11. Response to a square-wave input of robust controlled HRA and idealised and delayed MAC

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  $4 \times 4$  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  $10 \times 10$  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  $10 \times 10$  high redundancy actuator," in *Submitted to 23rd IAR Workshop on Advanced Control and Diagnosis*, 2008.