Multi-agent Control of High Redundancy Actuation

Jessica Davies[1]   Roger Dixon[2]   Roger M. Goodall [2]   Thomas Steffen[3]

[1]Lancaster Environment Centre, Lancaster University, Lancaster, LE11 3TU, UK
[2]School of Electrical, Electronic & Systems Engineering, Loughborough University, Loughborough, LA1 4YQ, UK
[3]Department of Aeronautical and Automotive Engineering, Loughborough University, Loughborough, LE11 3TU, UK

Abstract: The high redundancy actuator (HRA) concept is a novel approach to fault tolerant actuation that uses a high number of small actuation elements, assembled in series and parallel in order to form a single actuator which has intrinsic fault tolerance. Whilst this structure affords resilience under passive control methods alone, active control approaches are likely to provide higher levels of performance. A multiple-model control scheme for an HRA applied through the framework of multi-agent control is presented here. The application of this approach to a 10×10 HRA is discussed and consideration of reconfiguration delays and fault detection errors are made. The example shows that multi-agent control can provide tangible performance improvements and increase fault tolerance in comparison to a passive fault tolerant approach. Reconfiguration delays are shown to be tolerable, and a strategy for handling false fault detections is detailed.

Keywords: Fault tolerance, actuators, intelligent control, redundancy, intelligent actuators, cooperative systems.

1 Introduction

1.1 Traditional approaches to fault tolerant actuation

In automated processes, faults in hardware or software often produce undesired reactions. Faults can lead to system failures, where expected actions are not completed, possibly resulting in damage to the plant, its environment or people in the vicinity of the plant[1]. A fault tolerant system is able to avoid failure and achieve adequate system performance in the presence of faults.

The majority of fault tolerance research to date has concentrated on sensor faults[2−5]. 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 the signals they produce may be processed or replicated analytically to provide fault tolerance. Actuators, however, must deal with energy conversion. 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[6]. No approach can avoid this fundamental requirement.

The common solution involves straightforward parallel replication of actuators[7−10]. 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. It also can not deal with lock-up (fail-fixed) faults easily.

1.2 High redundancy actuation

High redundancy actuation (HRA) takes a different approach to this problem. 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 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 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 Fig. 1). Faults in elements will affect the maximum 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. 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.
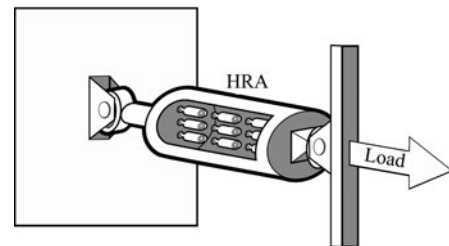

Fig. 1   High redundancy actuation

Through careful design for specific applications, HRA can provide a solution that continues to operate within the system's performance requirements in the presence of multiple faults in the elements, and gracefully degrades after the specific redundancy design limits have been reached.

1.3 Fault tolerant control of high redundancy actuation

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. Research to

date suggests that this is a theoretically and practically viable approach[11−14]. Passive FTC is where a single robust control law is designed, which should provide adequate stability and performance under both nominal and fault conditions.

This concept with respect to HRA is illustrated by Fig. 2. 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} .
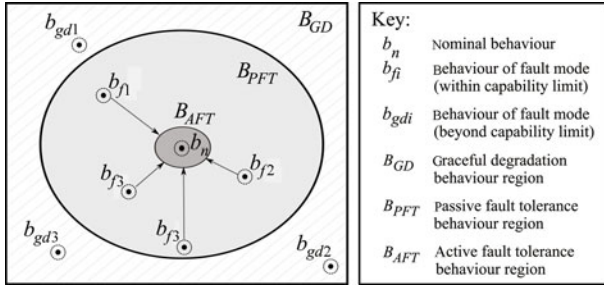

Fig. 2   Representation of passive and active fault tolerant control of HRA

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

This paper details a multi-agent system (MAS) inspired active FTC strategy for the HRA that aims to achieve near-nominal performance under the fault conditions.

1.4   Overview

Section 2 gives a brief overview of MAS and presents a rationale for its use with HRA. Section 3 then goes on to describe the current multi-agent control (MAC) approach and its main features. An example is then presented in Section 4, where the MAC scheme is applied to a 10×10 HRA and its performance is compared to a passive control strategy. Special considerations are made within the simulations for the reconfiguration period and fault detection errors. Finally, conclusions are made in Section 5 and the

future directions of the research are described.

2   Multi-agent control of high redundancy actuation

2.1   Multi-agent control

The concept of agent was first given by Minksy[15]. In his book, 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 when it had 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[16].

There is still some controversy within the agent community regarding what qualities should be included in the definition of an agent. There is a general consensus that autonomy is essential, but the attribution of other qualities is still under debate. However, the following definition, given by[17], conveys the key ideas: 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. An 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 [18]. This means each agent only deals with a local environment, and not the whole plant.

2.2   Rationale for multi-agent control of HRA

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

MAC and HRA are structurally similar (Fig. 3). 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 mode[19]. The process industry ac-

knowledges that the structuring of control is an important issue in complex systems. Thus it is given more attention in this field and numerous MAC system have been proposed within this application area[20].
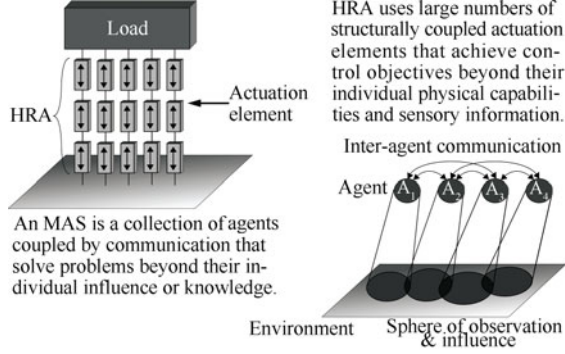

Fig. 3   HRA and MAS

Equally, a structured approach to control may be achieved through use of decentralised control techniques[21, 22]. 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 multi-agent 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 direct measured quantities[18]. 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 conventional active FTC techniques. Localisation of decisional 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.

3   MACHRA scheme

HRAs have two main configurations of elements: series-in-parallel (SP) configuration, where serial branches of elements are connected to a load in parallel; and parallel-in-series (PS) configuration, where parallel elements are connected serially. These two configurations have equal nominal system capabilities, but differing tolerance to the two major fault types: lock-up and loose faults. The scheme described here refers to the PS configuration, which is the most severely affected by lock-up faults. However, the con-

cept is equally applicable to the SP or mixed configurations.

The MAC of HRA scheme is essentially a decentralised multiple-model active FTC solution. The resultant control solution has two main properties:

1) It can provide near nominal performance under fault conditions (if the fault level is within the fault limit $F_{\lim}$) after a reconfiguration period T_r has elapsed.

2) It can provide near nominal performance in the case of false fault detections (up to the fault limit $F_{\lim}$).

The first feature is achieved as a control law is designed for each possible fault condition. The HRA is decomposed into similar physical subsystems. It is assumed that these subsystems operate in a finite number of modes, representing the nominal and fault conditions. For each mode, a classical control law is designed offline to provide a performance that is near to that of the nominal case. An agent-based framework is then applied to the decomposed HRA to detect faults locally and implement these pre-designed controllers. It would also be possible to apply adaptive control using this approach. However, a multiple-model based approach was favoured as this aids the certification of robustness and stability that would be necessary for high integrity applications for which HRA is intended.

The reconfiguration of control through the agent-based framework will not be instantaneous. There will be some delay incurred by fault detection T_{fd}. Communication of the fault throughout the agency will also take a finite period (T_{com}) and finally the reconfigured control will require certain time to settle (T_s). Hence, the reconfiguration time T_r may be expressed as

$$T_r = T_{fd} + T_{com} + T_s. \tag{1}$$

The fault limit $F_{\lim}$ for an HRA is the number of lock-up faults that is designed to accommodate. As lock-up faults essentially remove the travel capability of a parallel branch of elements, then the fault limit is the number of serial branches of elements, minus the number of serial elements required to achieve the required travel.

3.1   MACHRA architecture

Matlab/Simulink is used to create and simulate HRA assemblies, details of which can be found in [23]. State flow 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.

3.1.1   Agency architecture

Fig. 4 displays the MACHRA scheme′s agency architecture for an $m \times n$ HRA PS configuration. In the figure, the extensibility of the architecture is indicated by the dashed lines. 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 multi-agent concepts where no hierarchy should exist. A fixed outer control loop provides each agent with an identical set-point. Communication between agents is transmitted via point to point links connecting neighbouring elements. This means that agents only consider messages from their structural neighbours in the first instance. However, if lock-up faults occur, the

agents′ structural neighbours will change and thus other messages become relevant. It is the job of the affected agent to forward messages as necessary. This particular (structural neighbour) approach is taken as it limits the amount of inputs and outputs (I/O) per agent regardless of the number of agents in the overall HRA. Whilst this is a significant benefit, it also has the disadvantage of adding some delay in communication.
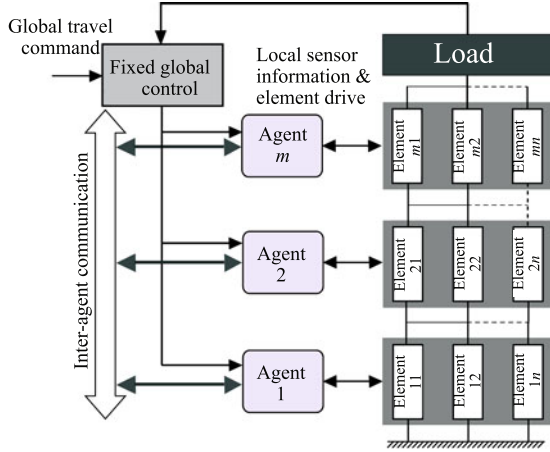

Fig. 4   MACHRA agency architecture

3.1.2   Agent architecture

The agent architecture is illustrated in Fig. 5. This architecture has similarities with subsumption, first introduced by [24], which uses behaviours layered in order of abstraction to produce more complex emergent behaviours in a reactive time frame. This reactivity is the 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.
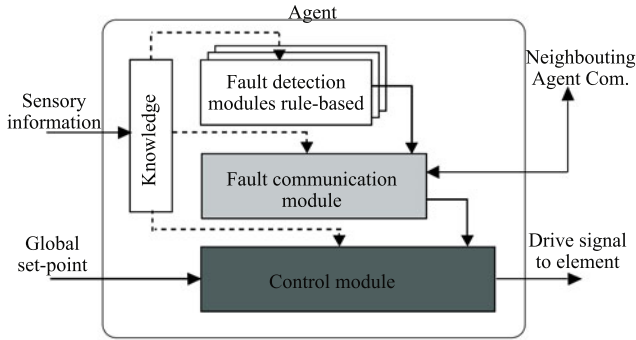

Fig. 5   MACHRA agent architecture

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 simple rule-based logic which determines the fault status of the element based on sensory information and internal knowledge. Firstly, it is checked whether the elements are moving. If they are, then

the system is not locked. However, if they are stationary, it is then determined whether they should be moving according to the agent′s input command and the mechanical limits of the system. Hence, faults are only detected during transient periods of operation. This is sufficient as elements that lock during steady-state have no effect on the system in that period.

If a fault is detected, the agent updates its internal fault state: FS_i. 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 agents communicate two values: the cumulative faults in element branches below the current branch FS_b; and cumulative faults above the current branch FS_a. The sum of FS_a, FS_b and FS_i, gives the total number of faults FS_T in the system, hence providing a simple, scalable method for communicating faults throughout the agency. Fig. 6 shows the process of recording faults over three communication intervals (top to bottom), for a four agent HRA. At the 3rd interval all four agents have the correct total faults.
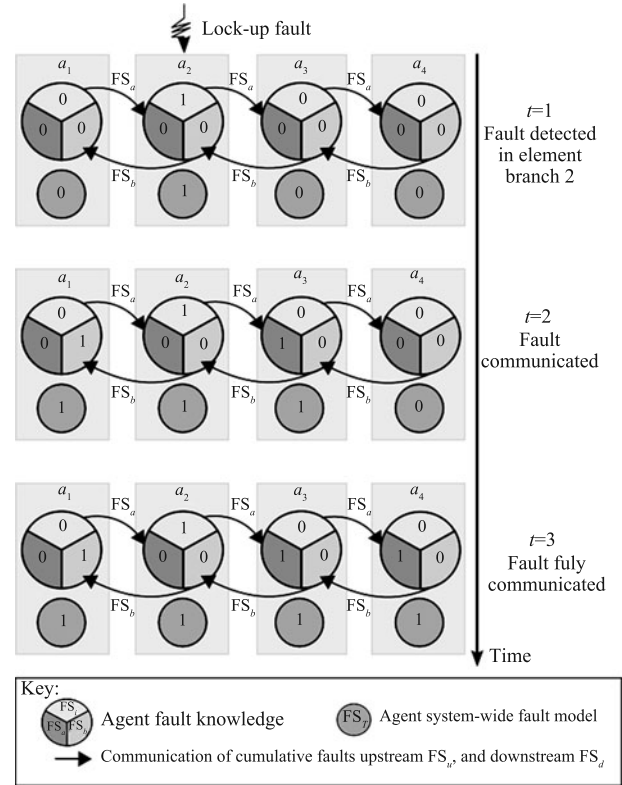

Fig. 6   Inter-agent fault communication and counting

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. The set-point from the global controller is initially fed through a feed-forward gain which is scheduled according to the active number of elements in the system. Then an inner control loop using local element position is implemented where the multiple-model control scheme is employed. The controller is a classical design, the parameters of which are chosen from a set of pre-computed values, again based on the num-

ber of active elements in the HRA.

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 MACHRA example

This section describes the results of applying this MAC scheme to an HRA in simulation. A 10×10 HRA is used as an example, as this is a non-trivial size system comprising a relatively large number of elements and as such the effect of faults is of a realistic dimension. The MAC approach is compared to passive fault tolerant control methods in order to determine if the addition of active FTC is beneficial.

4.1 System description

The 10×10 PS HRA is structured as shown in Fig. 4, with ten branches of ten parallel elements arranged serially. The actuation elements currently being used within the project are SMAC electromagnetic actuators[25], which have been configured to form a lab-scale concept demonstrator HRA system. The modelling of these actuators was considered in [23], and will not be detailed here. A simplified two state element model is used in this example, making the overall system 20th order. This model is included in the Appendix.

The position of the load is set as the control objective, and some transient requirements are defined for the system, suitable to the system's technology with good stability margins (Table 1). In this case, the system load is six times the mass of the inter-element masses and 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.

Table 1　Requirements

Requirements	Example 2 (10×10 HRA)
Travel window	± 0.06 m ($6 \times$ element travel)
Overshoot	$< 2\%$
Steady state error	0%
Settling time	<1.2 s

As the PS assembly is naturally tolerant 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 10×10 system of this dimensioning may incur up to 40 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, four lock-ups will bring the travel capability to critical point. The actual location of these faults, provided they are located in by separate branches, has very little effect on the resultant fault behaviour[26]. Thus, from 1 to 4 faults are injected into the simulation in a worst-case manner (in separate branches), as described in Table 2.

4.1.1 Control scheme

Fig. 7 portrays both the passive and MAC control schemes. The passive scheme has cascaded classical controllers designed to meet the control objectives in nominal conditions. These control laws are included in the Appendix. The inner loops contain 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's load. Proportional-integral (PI) control is used in the outer loop to achieve the steady state requirements.

Table 2　Fault cases

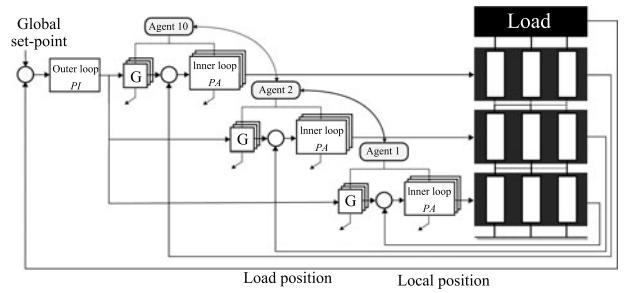| Case	Description	HRA state
Nominal	All elements are healthy	Healthy and capable
FC1	Branch nearest load locked	Faulty, but capable
FC2	2 branches nearest load locked	Faulty, but capable
FC3	3 branches nearest load locked	Faulty, but capable
FC4	4 branches nearest load locked	Critical fault level

Fig. 7　Passive and multi-agent control schemes

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. However, four more sets of inner-loop control laws are designed based on the four fault modes of the system, where six to nine out of ten parallel branches of elements are active.

Thus, on detection of a fault, this is communicated to the agents with healthy elements and their inner-loop phase advance controller parameters are changed according to a look-up table of pre-computed control laws (included in the appendix). The feed-forward gain in the agent's control module is also changed to redistribute the travel demand of the system, i.e., if the system is nominal and one element locks then the gain would be changed from 1/10 to 1/9, as there are nine active parallel element branches remaining. This keeps the gain in the system constant.

The outer-loop controller is not reconfigured as this would compromise the localisation of fault detection and reconfiguration decision, producing a potential single point of failure.

4.1.2 Simulation of fault cases

Fig. 8 displays the response of the passively controlled and MAC 10 × 10 HRA under nominal and faulty conditions (Table 2), when a step change of 0.05 m in the reference was applied at $t = 0$. All faults were introduced at the beginning of the simulation. Table 3 gives the stability margins and transient characteristics of these responses.

In the passive control case, the simulations show that, as faults occur, the increasing load slows the response. Nevertheless, the passive control case shows some tolerance to

faults, as the steady-state criteria is met under each fault condition due to the integral action of the outer loop control. However, the rise and settling time requirements are not met when two or more faults are present in the system.

In contrast, the MAC case produces fault responses that are very similar to the nominal case. The requirements are met under each fault condition.

Table 3 Simulation results for each fault case giving overshoot (OS), rise time (RT), settling time (ST), steady state error (SSE), gain margin (GM) and phase margin (PM)

Fault case	OS (%)	RT (s)	ST (s)	SSE (%)	GM (dB)	PM (°)
Nominal	1.85	0.68	1.02	0	∞	74
Passive FC1	1.01	0.75	1.19	0	∞	76
Passive FC2	1.01	0.88	1.47	0	∞	79
Passive FC3	1.01	1.05	1.88	0	∞	82
Passive FC4	1.10	1.33	2.26	0	∞	86
MAC FC1	1.66	0.68	1.04	0	∞	74
MAC FC2	1.69	0.68	1.05	0	∞	75
MAC FC3	1.93	0.71	1.07	0	∞	75
MAC FC4	1.82	0.73	1.12	0	∞	77
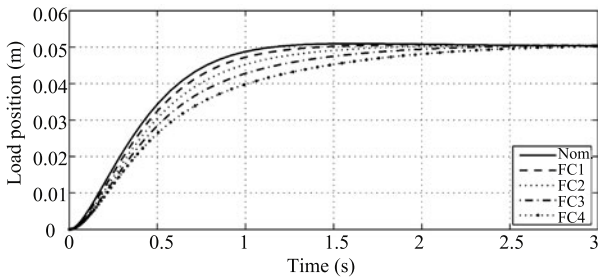
(a) Step response of 10×10 HRA with passive cascaded control
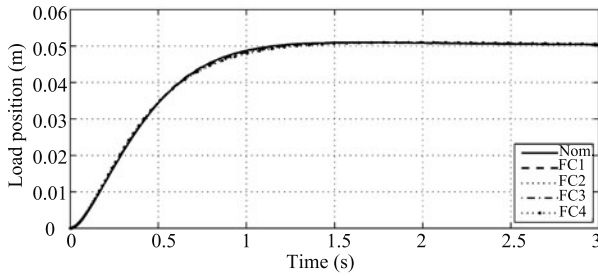
(b) Step responsse of 10×10 MACHRA

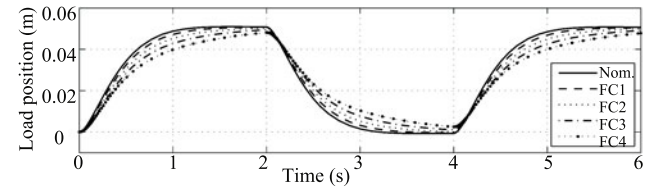Fig. 8 Step response of passive and MAC 10×10 HRA

4.2 Reconfiguration delays

The MAC results given in the previous section assumed that faults were detected and communicated instantaneously throughout the agency. As acknowledged in Section 3, this is not a realistic assumption. The effect of reconfiguration delays T_r must be considered in the simulation if the results are to resemble reality. Fig. 9 shows the transient responses of the idealised MAC 10×10 HRA and one that includes these reconfiguration delays. The fault detection, communication and control reconfiguration are all

simulated using state flow, which introduces delays into the system.

A square-wave input is applied to the system and all faults 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.

Fig. 10 shows the initial response in more detail. Total reconfiguration of the system was attained after 0.35 s. 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 limit is 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 lessened if the faults did not occur simultaneously, which is likely to be the case in a real situation.

(a) Transient response of 10×10 HRA with passive cascaded control
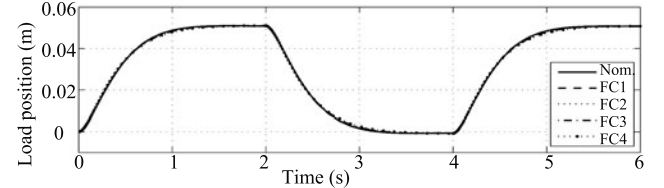
(b) Transient response of ideal 10×10 MACHRA
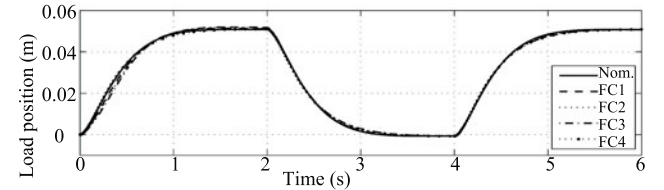
(c) Transient response of 10×10 MACHRA with delays

Fig. 9 Transient response of passive, ideal MAC and MAC with delays

These simulations show that in this case, moderate detection, communication and reconfiguration delays in the MACHRA have a limited influence on the performance of the system during reconfiguration, which is likely to be acceptable in application.

4.3 Fault detection errors in MACHRA

The benefit of using MAC witnessed in the examples is attained at the cost of a dependency on fault detection. As the HRA is an intended solution for high integrity applications, it is necessary to consider what would happen if this fault detection failed.

As mentioned previously, fault detection errors 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 ind-
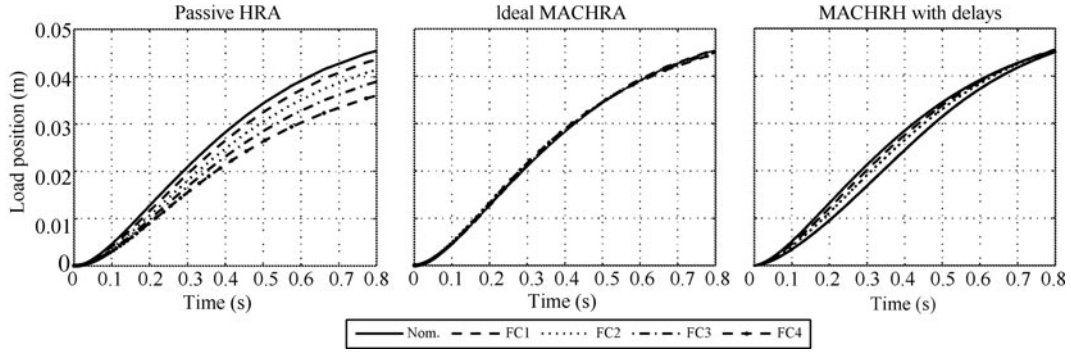
Fig. 10   Initial response of passive, ideal MAC and MAC with delays

uce instability. Equally, if the system's control relies upon faults being detected and a fault is not detected, then the results could be similar. Fault detection errors in this particular system will be considered here.

4.3.1   Undetected faults

Undetected faults should not cause stability problems in this case. At worst, the system's response will be that of the passive case, i.e., the system will become slower, but stability will be maintained.

4.3.2   False detection of faults

False detection in this MACHRA approach 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. This is a high number of false detections, and one would not expect a well-designed fault detection scheme to perform so badly. However, it is worthwhile considering such worst-case scenarios.

Table 4   False detections

No. of false detects	1	2	3	4
Settling time (s)	1.85	1.92	1.93	1.66
OS (%)	1.88	4.04	6.16	8.18
PM (°)	72	70	68	66

When false detections are made, the phase margin decreases, but the system remains stable. The overshoot and settling time, however, rise significantly.

As proposed in Section 3, the flexibility of MAC can handle this problem through further reconfiguration. On triggering of the FDM, the input reference of the agent is fixed to the local position at time of detection. The controller within the agent is replaced with the PI compensator given in the appendix. Given sufficient gain is achievable (which is the case within the physical limits of this system), then the subsystem is forced to behave as the detected fault case.

The simulation results of this approach are shown in Table 5. Subsequently, the phase margin is not eroded and the overshoot and settling time limit achieved. This approach will have no effect if the fault detected is actually present.

Table 5   False detection with reconfiguration

No. of false detects	1	2	3	4
Settling time (s)	1.04	1.05	1.08	1.13
OS (%)	1.58	1.51	1.66	1.57
PM (°)	74	74	74	75

5   Conclusions

This paper 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 an example of its application to a 10×10 HRA has 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. Near nominal performance can be maintained in worst case fault scenarios.

It was shown that reconfiguration delays in MAC can affect the response until full reconfiguration has been achieved. However, these effects may be considered acceptable, due to their short lasting nature. Fault detection errors were also considered and it was shown that MAC has the flexibility to counteract the negative effects of false detections.

Future work on the HRA technology concept itself will include practical testing and control of a 3×4 electro-mechanical actuator. This will extend the work to give an indication of performance in a real-world situation. In addition, future work concerning the reconfiguration/communication delays should focus on the search for a general analytical solution.

Appendix

A1.   Nomenclature

B_{AFT}:   Active fault tolerant control behaviour region
b_f: HRA fault mode behaviour
B_{GD}:   Graceful degradation behaviour region
b_{gd}:   Behaviour of HRA when capability limit is exceeded, i.e., a graceful degradation behaviour
b_n: Nominal behaviour of the HRA as represented

within the behavioural space

B_{PFT}: Acceptable behaviour region

d: Damping constant (Ns/m)

F: Force (N)

$F_{\lim}$: The fault limit, i.e., the number of lock-up faults an HRA is designed to accommodate

FS_a/FS_b: Cumulative faults in element branches above/below the current branch

FS_i: Agent internal fault state

FS_T: Total faults in HRA system

k: Force constant (N/A)

m_n: Inter-mass element (kg)

r: Spring constant (N/m)

$R1$: Input resistance

T_{com}: Fault communication delay

T_{fd}: Fault detection delay

T_r: Reconfiguration time

T_s: Settling time

u_n: Element input voltage (V)

x_n: Absolute position (m)

A2. 10×10 PS HRA model

The nominal 10×10 HRA can be described as follows:

$$
\begin{bmatrix} \dot{c}_1 \\ \dot{c}_2 \\ \vdots \\ \dot{c}_{10} \end{bmatrix} = A \cdot \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{10} \end{bmatrix} + B \cdot \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{10} \end{bmatrix}
$$

$$
y = A \cdot \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{10} \end{bmatrix} + 0
$$

$$
A = \begin{bmatrix}
A_{b1} & A_{c1} & 0 & 0 & 0 & 0 & 0 \\
A_{a2} & A_{b2} & A_{c2} & 0 & 0 & 0 & 0 \\
0 & A_{a3} & A_{b3} & A_{c3} & 0 & 0 & 0 \\
0 & 0 & \ddots & \ddots & \ddots & 0 & 0 \\
0 & 0 & 0 & A_{a8} & A_{b8} & A_{c8} & 0 \\
0 & 0 & 0 & 0 & A_{a9} & A_{b9} & A_{c9} \\
0 & 0 & 0 & 0 & 0 & A_{a10} & A_{b10}
\end{bmatrix}
$$

$$
B = \begin{bmatrix}
U_1^1 & U_2^1 & 0 & 0 & 0 & 0 & 0 \\
0 & U_2^2 & U_3^2 & 0 & 0 & 0 & 0 \\
0 & 0 & U_3^3 & U_4^3 & 0 & 0 & 0 \\
0 & 0 & 0 & \ddots & \ddots & 0 & 0 \\
0 & 0 & 0 & 0 & U_8^8 & U_9^8 & 0 \\
0 & 0 & 0 & 0 & 0 & U_9^9 & U_9^{10} \\
0 & 0 & 0 & 0 & 0 & 0 & U_{10}^{10}
\end{bmatrix}
$$

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

where

$$
c_n = \begin{bmatrix} \dot{x}_n & x_n \end{bmatrix}^{\mathrm{T}}, \quad A_{an} = \begin{bmatrix} \dfrac{F_{an}}{m_n} & \dfrac{F_{bn}}{m_n} \\ 0 & 0 \end{bmatrix}
$$

$$
A_{bn} = \begin{bmatrix} -\dfrac{F_{an} + A_{an+1}}{m_n} & -\dfrac{F_{bn} + A_{bn+1}}{m_n} \\ 1 & 0 \end{bmatrix}
$$

$$
A_{cn} = \begin{bmatrix} \dfrac{F_{an} + 1}{m_n} & \dfrac{F_{bn} + 1}{m_n} \\ 0 & 0 \end{bmatrix}
$$

$$
F_{ai} = \left\{ \dfrac{k_{i,n}^2 + d_{i,n} R_{i,n}}{R_{r,n}} \right\}, \quad F_{bi} = \sum_{n=1:10} r_{i,n}
$$

$$
U_i^n = \begin{bmatrix} \dfrac{1}{m_n} \dfrac{\sum k_{i1:10}}{\sum R_{i1:10}} & 0 \end{bmatrix}.
$$

A methodology for providing the fault condition models is provided in [26].

A3. Control algorithms

1) Nominal and fault mode control

The outer-loop, load position controller (see Fig. 7) in the passive FTC system and MAC case takes the form of PI_{global}. The inner loop, local position controller also used in these cases takes the form of PA_{norm}. The control parameter values are given for each fault case in Table A1. It can be seen that the global controller parameters are stationary throughout, whereas the inner loop controllers are adjusted in response to faults in the MAC cases.

$$
PI_{\mathrm{global}} = G_{GL} \frac{a_{gl}s + 1}{s}
$$

$$
PA_{\mathrm{nom}} = G_{in} \frac{a_{in}s + 1}{b_{in}s + 1}.
$$

Table A1   Control parameters under fault cases

Fault case	G_{gl}	a_{gl}	G_{in}	a_{in}	b_{gl}
Nominal	3	0.51	10	1.3	0.1
Passive FC1	3	0.51	10	1.3	0.1
MAC FC1	3	0.51	10	1.268	0.098
MAC FC2	3	0.51	10	1.203	0.093
MAC FC3	3	0.51	10	1.105	0.085
MAC FC4	3	0.51	10	1.021	0.078

2) Faulty element control

The set-point of the faulty element's on detection of a fault is fixed to the position at time of detection and the local control law is changed to the following PI controller:

$$
PI_{\mathrm{local}} = 500 \frac{s + 1}{s}.
$$

References

[1] M. Blanke, M. Staroswiecki, N. E. Wu. Concepts and methods in fault-tolerant control. In *Proceedings of 2001 American Control Conference*, IEEE, Arlington, VA, vol. 4, pp. 2606–2620, 2001.

[2] G. Betta, 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.

[3] P. M. Frank, S. X. Ding, T. Marcu. Model-based fault diagnosis in technical processes. *Transactions of the Institute of Measurement and Control*, vol. 22, no. 1, pp. 57–101, 2000.

[4] R. T. Rausch, K. F. Goebel, N. H. Eklund, B. J. Brunell. Integrated in-flight fault detection and accommodation: A model-based study. *Journal of Engineering for Gas Turbines and Power*, vol. 129, no. 4, pp. 962–969, 2007.

[5] G. H. Yang, D. Ye. Adaptive fault-tolerant H_∞ control against sensor failures. *IET Control Theory and Applications*, vol. 2, no. 2, pp. 95–107, 2008.

[6] R. J. Patton. Fault detection and diagnosis in aerospace systems using analytical redundancy. *IEE Computing and Control Engineering Journal*, vol. 2, no. 3, pp. 127–136, 1991.

[7] S. Osder. Practical view of redundancy management application and theory. *Journal of Guidance, Control, and Dynamics*, vol. 22, no. 1, pp. 12–21, 1999.

[8] S. Ravenscroft. Actuation systems. *Flight Control Systems: Practical Issues in Design and Implementation*, R. Pratt, Ed., IET, 2000.

[9] J. Jiang, Q. Zhao. Design of reliable control systems possessing actuator redundancies. *Journal of Guidance, Control, and Dynamics*, vol. 23, no. 4, pp. 709–718, 2000.

[10] W. Chen, J. Jiang. Fault-tolerant control against stuck actuator faults. *IEE Proceedings – Control Theory and Applications*, vol. 152, no. 2, pp. 138–146, 2005.

[11] X. Du, R. Dixon, R. M. Goodall, A. C. Zolotas. LQG control of a high redundancy actuator. In *Proceedings of IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, IEEE, Zurich, pp. 1–6, 2007.

[12] T. Steffen, J. Davies, R. Dixon, R. M. Goodall, A. C. Zolotas. Using a series of moving coils as a high redundancy actuator. In *Proceedings of IEEE/ASME International Conference for Advanced Intelligent Mechatronics*, IEEE, Zurich, pp. 1–6, 2007.

[13] X. Du. High Redundancy Actuation, Ph. D. dissertation, Loughborough University, Loughborough, UK, 2008.

[14] X. Du, R. Dixon, R. M. Goodall, A. C. Zolotas. Assessment of strategies for control of high redundancy actuators. In *Proceedings of the Actuator'08*, Bremen, Germany, 2008.

[15] M. Minsky. *The Society of Mind*, New York: Simon & Schuster, 1988.

[16] G. Weiss. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, New York: MIT Press, 1999.

[17] N. R. Jennings, K. Sycara, M. Wooldridge. A roadmap of agent research and development. *Autonomous Agents and Multi-agent Systems*, vol. 1, no. 1, pp. 7–38, 1998.

[18] J. Ferber. *Multi-agent Systems: An Introduction to Distributed Artificial Intelligence*, United States: Addison-Wesley, 1999.

[19] A. J. N. van Breemen, T. de Vries. An agent based framework for designing multi-controller systems. In *Proceedings of the 5th International Conference on the Practical Application of Intelligent Agents and Multi-agent Technology*, Manchester, UK, pp. 219–236, 2000.

[20] H. Q. Wang, C. Wang. Intelligent agents in the nuclear industry. *Computer*, vol. 30, no. 11, pp. 28–31, 1997.

[21] D. D. Šiljak. *Large-scale Dynamic Systems: Stability and Structure*, New York: North-Holland, 1978.

[22] D. D. Šiljak. *Decentralized Control of Complex Systems*, vol. 184. London: Academic Press Ltd., 1991.

[23] J. Davies, T. Steffen, R. Dixon, R. M. Goodall, A. C. Zolotas, J. T. Pearson. Modelling of high redundancy actuation utilising multiple moving coil actuators. In *Proceedings of IFAC World Congress*, Seoul, Korea, vol. 17, pp. 3228–3233, 2008.

[24] R. A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, vol. 2, no. 1, pp. 14–23, 1986.

[25] SMAC, Actuator Data Sheet-electric Linear/Rotary Moving Coil Actuators, LAL90–50, 2004.

[26] J. Davies, T. Steffen, R. Dixon, R. M. Goodall, A. C. Zolotas. Active versus passive fault tolerant control of a high redundancy actuator. In *Proceedings of the European Control Conference*, pp. 1–6, 2009.

Jessica Davies obtained her M. Eng. degree in electronic and electrical engineering at Loughborough University, UK in 2006, after which she joined Loughborough Control Group. She received her Ph. D. degree in 2010 and she currently works at Lancaster University, UK as a senior research associate.

Her research interests include fault tolerant control design, intelligent control and model order reduction techniques.

E-mail: j.davies4@lancaster.ac.uk

Roger Dixon was awarded a B. Eng. degree in mechanical engineering in 1992 and an M. Sc. degree in mechatronics in 1993, from Lancaster University, UK. He studied for a Ph. D. degree in discrete time modelling and control at the Centre for Research in Environmental Systems and Statistics (CRES), graduating in 1996. In 1996, he took a position at Alstom in Leicester and he joined Loughborough as a senior lecturer in September 2003. He is head of the Control Systems Research Group at Loughborough University, School of Electronic and Electrical Engineering and the director of the Systems Engineering Doctorate Centre. He is a fellow of the Higher Education Academy, a registered Chartered Engineer and fellow of the Institution of Mechanical Engineers.

His research interests focus on various aspects of control systems engineering including: application of model-based control systems design, model-based fault detection and isolation, system condition/health monitoring and fault tolerant design (of controllers and actuators).

E-mail: R.Dixon@lboro.ac.uk (Corresponding author)

Roger M. Goodall holds the BA and MA degrees in engineering from the University of Cambridge, and Ph. D. degree from Loughborough University, UK. From 1970–1982, he worked at British Rail's Research Division in Derby, UK. He moved to Loughborough University in 1982 and became a full professor in 1994. He is an elected fellow of the Royal Academy of Engineering, and also a fellow of the professional engineering institutions IET and IMechE. He is professor of Control Systems Engineering in the School of Electronic, Electrical and Systems Engineering at Loughborough University in the UK.

His research interests include active control of railway dynamic systems, the controllability of super-conducting magnets for electromagnetic actuation and state estimation for aircraft.

E-mail: R.M.Goodall@lboro.ac.uk

Thomas Steffen graduated from the Technical University of Ilmenau with a diploma (B. Eng.) in electrical engineering/automation and control in 1999. He worked as a research assistant at the Technical University of Hamburg-Harburg (TUHH), Germany and at the Ruhr-Universität Bochum, Germany, where he obtained his Ph. D. degree in 2004. He joined Loughborough University as a research associate in January 2007, and now works as a lecturer at the Department of Aeronautical and Automotive Engineering, Loughborough University.

His research interests include engine, powertrain and aftertreatment control, high speed / high performance control and fault tolerant control.

E-mail: T.Steffen@lboro.ac.uk