Safety in Numbers: Fault Tolerance in Robot Swarms

Alan FT Winfield*
Intelligent Autonomous Systems laboratory[1],
UWE Bristol, Coldharbour Lane, Bristol BS16 1QY, UK
E-mail: Alan.Winfield@uwe.ac.uk
*Corresponding author

Julien Nembrini
Swarm-Intelligent Systems Group,
EPFL, 1015 Lausanne, Switzerland
E-mail: Julien.Nembrini@epfl.ch

Abstract: The Swarm Intelligence literature frequently asserts that swarms exhibit high levels of robustness. That claim is, however, rather less frequently supported by empirical or theoretical analysis. But what do we mean by a 'robust' swarm? How would we measure the robustness or - to put it another way - fault tolerance of a robotic swarm? These questions are not just of academic interest. If swarm robotics is to make the transition from the laboratory to real-world engineering implementation we would need to be able to address these questions in a way that would satisfy the needs of the world of safety certification. This paper explores fault tolerance in robot swarms through Failure Mode and Effect Analysis (FMEA) and reliability modelling. The work of this paper is illustrated by a case study of a wireless connected robot swarm, employing both simulation and real-robot laboratory experiments.

Keywords: Swarm Intelligence; Swarm Robotics; Swarm Engineering; fault tolerance; safety; reliability.

Reference to this paper should be made as follows: Winfield AFT and Nembrini J (2005) 'Safety in Numbers: Fault Tolerance in Robot Swarms', Int. J. Modelling Identification and Control, Vol. 1, No. 1, pp 00–00

Biographical notes: Alan FT Winfield is Hewlett-Packard Professor of Electronic Engineering in the Faculty of Computing, Engineering and Mathematical Sciences at the University of the West of England (UWE), Bristol, UK. He received his PhD in digital communications from the University of Hull in 1984, then co-founded and led APD Communications Ltd until taking up appointment at UWE, Bristol in 1991. Alan Winfield co-founded UWE's Intelligent Autonomous Systems laboratory in 1993 and the focus of his current research is on the engineering and scientific applications of Swarm Intelligence.
Julien Nembrini completed his PhD in swarm robotics at UWE, Bristol, in 2004. He is currently a post-doctoral research assistant in the Swarm-Intelligent Systems Group at the École Polytechnique Fédérale de Lausanne, Switzerland.

1 INTRODUCTION

From an engineering standpoint the design of complex distributed systems based upon the swarm intelligence paradigm is compellingly attractive but problematical. A distinguishing characteristic of distributed systems based upon swarm intelligence is that they have no hierarchical command and control structure, and hence no common mode failure point or vulnerability. Inspired by social insects, individual agents make decisions autonomously, based upon local sensing and communications, see Bonabeau et al, 1999 (3) and 2000 (4). Systems with these characteristics could, potentially, exhibit very high levels of robustness, in the sense of tolerance to failure of individual agents; much higher levels of robustness

[1] http://www.ias.uwe.ac.uk/

than in complex distributed systems based on traditional design approaches. However, that robustness comes at a price. Complex systems with swarm intelligence might be difficult to control or mediate if they started to exhibit unexpected behaviours. Such systems would therefore need to be designed and validated for a high level of assurance that they exhibit intended behaviours and *equally importantly* do not exhibit unintended behaviours. It seems reasonable to assert that future engineered systems based on the swarm intelligence paradigm would need to be subject to processes of design, analysis and test no less demanding that those we expect for current complex distributed systems.

Some might argue that a 'dependable swarm' is an oxymoron; that the swarm intelligence paradigm is intrinsically unsuitable for application in engineered systems that require a high level of integrity. The idea that overall desired swarm behaviours are not explicitly coded anywhere in the system, but are instead an *emergent consequence* of the interaction of individual agents with each other and their environment, might appear to be especially problematical from a dependability perspective. In a previous paper (16) we argued that this is not so: that systems which employ emergence should, in principle, be no more difficult to validate than conventional complex systems and, indeed, that a number of characteristics of swarm intelligence are highly desirable from a dependability perspective. In that paper we introduced the notion of a 'dependable swarm', that is a robotic swarm engineered to high standards of design, analysis and test, and therefore able to exhibit high levels of safety and reliability; hence 'swarm engineering'. That paper concluded that while some of the tools needed to assure a swarm for dependability exist, most do not, and set out a roadmap of the work that needs to be done before safety-critical swarms become an engineering reality. The present paper is part of that roadmap.

Probably the most challenging task in dependability assurance, see Anderson et al (1), is proving the *safety* of a system. Formally 'safety' is defined as the property of *not exhibiting undesirable behaviours* or, to put it more simply, not doing the wrong thing. In order to establish this property first requires that we identify and articulate all possible undesirable behaviours. This is called 'hazard analysis'. It is problematical with conventional complex systems, and there is no reason to suppose that identifying the hazards in swarm engineered systems will be any different. Hazards analysis is difficult because there are no formal methods for identifying hazards. It simply has to be done by inspection, typically by 'extreme brainstorming' to try and list all possible hazards (no matter how seemingly implausible or improbable).

Given a reasonably well understood operational environment there are two reasons for undesirable behaviours: random errors, or systematic (design) errors. Random errors are those due to hardware or component faults, and these are typically analysed using techniques such as Failure Mode and Effect Analysis (FMEA), see Dailey (5). The likelihood that random errors cause undesirable behaviours can be reduced, in the first instance, by employing high reliability components. But systems that require high dependability will typically also need to be fault tolerant, through redundancy for example. This is an important point since swarm engineered systems should, in this respect, offer very significant advantages over conventional complex systems. Two characteristics of swarms work in our favour here. Firstly, simple agents with relatively few rules lend themselves to FMEA, and their simplicity facilitates design for reliability. Secondly, swarms consist of multiple robots and hence *by definition* exhibit high levels of redundancy and tolerance to failure of individual agents. Indeed, robot swarms may go far beyond conventional notions of fault tolerance by exhibiting tolerance to individuals who actively thwart the overall desired swarm behaviour. It is the purpose of this paper to explore, by means of a case study, fault tolerance (to random errors) in robot swarms through hazards analysis, FMEA and reliability modelling.

Systematic errors are those aspects of the design that could allow the system to exhibit undesirable behaviours. For swarm engineered systems analysis of systematic errors clearly needs to take place at two levels: in the individual agent and for the swarm as a whole. Analysis of systematic errors in the individual agent should be helped by the relative simplicity of the agents, but is not trivial. Analysis of systematic errors for the swarm as a whole is much more problematical, particularly if the desired behaviours are emergent. However, in (17) we explore the use of the Temporal Logic formalism for specification and possibly proof of correctness of emergent behaviours.

This paper proceeds as follows. Section 2 introduces the case study that will be used to illustrate and underpin the work of the paper, then introduces the concept of robustness in the context of swarm robotics. Section 3 uses the methodology of Failure Mode and Effect Analysis (FMEA) to investigate the fault tolerance of our case study robot swarm. First, we identify all possible hazard (fault) conditions, then analyse the effect (and severity) of such faults occuring in one or more robots of the swarm. Section 4 outlines and discusses a number of possible reliability modelling approaches that could be applied to swarm robotics. Finally in section 5 the paper draws general conclusions, discusses the findings of this work and proposes future work to improve models of fault tolerance in robot swarms.

2 CASE STUDY: SWARM CONTAINMENT

As a case study let us consider a swarm robotics approach to physical containment or encapsulation, as illustrated in figure 1.

Potential applications for such an approach might include a swarm of marine robots that find and then contain oil pollution or *in-vivo* nano-bots that seek and isolate harmful cells in the blood stream (a kind of artificial phagocyte). The latter application is not so far-fetched

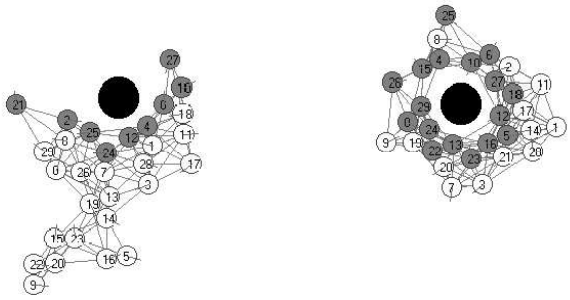

Figure 1: Encapsulation: (l) in progress and (r) complete

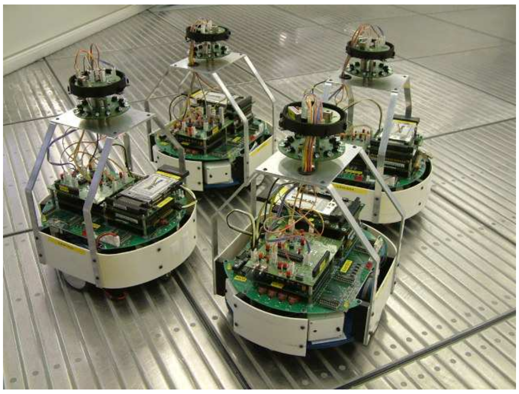

Figure 2: Laboratory Linuxbots

when one considers the rate of progress in the engineering of genetic circuits, see for example Yokobayashi et al, 2003 (18).

The emergent encapsulation behaviour of figure 1 is one of a number of emergent properties of a class of algorithms that we have developed, which make use of local wireless connectivity information alone to achieve swarm aggregation; see Nembrini et al, 2002 (13) and Nembrini 2004 (14). Wireless connectivity is linked to robot motion so that robots within the swarm are wirelessly 'glued' together. This approach has several advantages: firstly the robots need neither absolute or relative positional information; secondly the swarm is able to maintain aggregation (i.e. stay together) even in unbounded space, and thirdly, the connectivity needed for and generated by the algorithm means that the swarm naturally forms an *ad hoc* communications network. Such a network would be a requirement in many swarm robotics applications. The algorithm requires that connectivity information is transmitted only a single hop. Each robot broadcasts its ID and the IDs of its immediate neighbours only, and since the maximum number of neighbours a real robot can have is physically constrained and the same for a swarm of 100 or 10,000 robots, the algorithm scales linearly for increasing swarm size. The algorithm thus meets the criteria for swarm robotics, articulated by Şahin, 2005 (15) and Beni, 2005 (2). We have (we contend) a highly *robust* and *scalable* swarm of homogeneous and relatively incapable robots with only local sensing and communication capabilities, in which the required swarm behaviours are truly emergent.

The lowest level swarm behaviour is 'coherence' which, in summary, works as follows. Each robot has range-limited wireless communication and, while moving, periodically broadcasts an 'I am here' message (which also contains the IDs of its neighbours). The message will of course be received only by those robots that are within wireless range. If a robot loses a connection to robot r *and* the number of its remaining neighbours still connected to r is less than or equal to the threshold β then it assumes it is moving out of the swarm and will execute a 180° turn. When the number of connections rises (i.e. when the swarm is regained) the robot chooses a new direction at random. We say that the swarm is coherent if any break in its overall connectivity lasts less than a given time constant C. Coherence gives rise to the two emergent behaviours of swarm aggregation and a (coherent) connected *ad hoc* wireless network. Each robot also has short-range avoidance sensors and a long-range beacon sensor. When a robot senses the beacon it sets its β threshold to ∞ (the normal value of β is low, typically 2 or 3). This creates a differential motion in the swarm and gives rise to the emergent beacon taxis behaviour. Swarm obstacle avoidance and beacon encapsulation behaviours follow naturally. Table 1 summarises the complete set of emergent swarm behaviours.

	Case Study Swarm Behaviours
1	Swarm aggregation
2	Coherent *ad hoc* network
3	Beacon taxis
4	Obstacle avoidance
5	Beacon encapsulation

Table 1: Summary of Emergent Swarm Behaviours

Our algorithms for coherent swarming of wireless networked mobile robots have been tested extensively in simulation and, rather less extensively, using a fleet of physical laboratory robots. A group of these robots (Linuxbots) are shown in figure 2. The real robot implementation does not, however, constitute a real-world application. It is instead an 'embodied' simulation, shown in figure 3, whose main purpose is to verify that algorithms tested in computer simulation will transfer to the real world of non-ideal and noisy sensors and actuators.

2.1 Robustness and task completion

In the swarm intelligence literature, robustness sometimes refers to the simplicity and hence functional and mechanical reliability of the simple, even minimalist robots that comprise a swarm, Melhuish 2001, (11). Sometimes it refers to the ability of the swarm to cope with a demanding operational environment, Mondada et al 2002, (12), but most often robustness refers to the swarm's tolerance

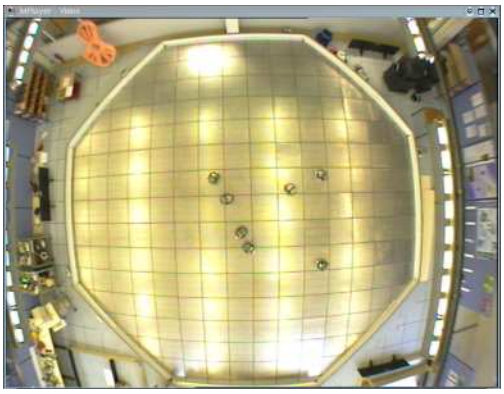

Figure 3: A Linuxbot Embodied Simulation

to the failure of one or more individual robots, Kazadi et al 2004, (8). Actually, this lack of precision about what is meant by robustness is understandable when we consider that a multi-robot system built upon the principles of swarm intelligence may well exhibit all of these forms of robustness, and more.

We could say that a robot swarm is robust because:

1. it is a completely distributed system and therefore has no common-mode failure point;
2. it is comprised of simple and hence functionally and mechanically reliable individual robots;
3. it may be tolerant to noise and uncertainties in the operational environment;
4. it may be tolerant to the failure of one or more robots without compromising[2] the desired overall swarm behaviours, and
5. it may be tolerant to individual robots who fail in such a way as to thwart the overall desired swarm behaviour.

When we speak of failure of the swarm to achieve the desired overall swarm behaviour we need to ask "failure to do what, exactly?". One of the defining characteristics of robotic swarms is that task completion is hard to pin down for two reasons. Firstly, because task completion is generally only in the eye of the beholder; the robots themselves often cannot know when the task is complete either because their simplicity precludes the sensing or computational mechanisms to detect the condition of task completion or because their limited localised sensing means they cannot see enough of the environment to be able to get the big picture (or both). In the case of puck clustering, for instance, robots that are left to run after they have done their work may even, in time, disturb and uncluster the pucks, only to then form them in a different place (11). Secondly, in swarm robotics, task completion is often defined by some statistical measure rather than a hard determined outcome. In considering the object encapsulation of figure 1, for instance, the swarm would need to reach an acceptable threshold of density of robots uniformly surrounding the target object (beacon).

3 ANALYSIS OF SWARM FAILURE

In this section we undertake a Failure Mode and Effect Analysis (FMEA) for our case study of a wireless connected robot swarm. The methodology is straightforward, see Dailey (5). We attempt to identify all of the possible hazards, which could be faults in robots or robot subsystems (internal hazards), or environmental disturbances (external hazards). Then, in each case, we analyse the effect of the hazard on each of the overall swarm behaviours. Thus, we build up a picture of the tolerance of the swarm to both types of hazard and begin to understand which hazards are the most serious in terms of compromising the overall desired swarm behaviours. FMEA is, at this stage, essentially qualitative. In this paper we consider only internal hazards. External hazards (i.e. communications noise) were investigated in Nembrini, 2004 (14).

First we identify the internal hazards. In keeping with the swarm intelligence paradigm our robot swarm contains no system-wide components or structures, thus the only internal hazards that can occur are faults in individual robots. Since, in our case, the robots of the swarm are all identical, then (internal) hazards analysis requires us to consider only the faults that could occur in one or more individual robots, and then consider their effect on the overall swarm behaviours. Table 2 identifies the fault conditions for an individual robot.

Hazard	Description
H_1	Motor failure
H_2	Communications failure
H_3	Avoidance sensor(s) failure
H_4	Beacon sensor failure
H_5	Control systems failure
H_6	All systems failure

Table 2: Internal hazards for a single robot

Table 2 makes the assumption that failures of robot subsystems can occur independently. This is a reasonable assumption, given that our mobile robots are in reality an assembly of complex but relatively self-contained subsystems. Hazard H_1 motor failure, covers the possibility of mechanical or motion-controller failure in one or both of the motors in our differential drive mobile robot, such that the robot is either unable to move at all or can only turn on the spot (which from an overall swarm point of view amounts to the same thing). Hazard H_2 represents a failure of the robot's wireless network communication system such that the robot is unable to receive or transmit messages. Hazards H_3 and H_4 represent failure of the robot's avoidance and beacon sensors respectively; the former will

[2] although the desired swarm behaviours might well be delayed or impaired in some way.

render the robot incapable of detecting and hence avoiding robots or environmental obstacles, the latter means that the robot cannot sense the target beacon, i.e. the object to be encapsulated. Hazard H_5 represents a failure of the robot's control system (typically implemented in software). Finally hazard H_6 represents a total failure of the robot; failure of the robot's power supply would, for instance, bring about this terminal condition.

Let us now consider the effects of each the hazards enumerated above on the overall swarm behaviours. We will consider here the effect on the overall swarm of the hazard occuring in one or *a small number* of the individuals in the swarm. Of course the question of how many is a small number in this context is important, and later in the paper we will refer again to the question of what proportion of robots need to fail in order to *seriously* compromise the desired overall swarm behaviours. Simulation studies suggest we can (very conservatively) estimate *a small number* here as under 5% of the swarm. Those studies also show that a higher proportion of failures can still be tolerated.

3.1 Hazard H_1: motor failure

The effect of motor failure in a single robot, or small number of robots, is interesting. Robot(s) with fault H_1 become - in effect - stationary but, given that their wireless communication and other electronic systems continue to function, they remain within the wireless *ad hoc* network of the swarm. These robots continue to fully contribute to the swarm aggregation and *ad hoc* network emergent behaviours. It is only when the swarm needs to physically translate its position. i.e. for the beacon taxis, obstacle avoidance and beacon encapsulation behaviours, that hazard H_1 becomes a serious problem. In these cases robots with motor failure will have the effect of physically anchoring the swarm, either impeding or, at worst, actually preventing the swarm from moving toward its target[3]. This is a potentially serious hazard since one or a small number of robots with motor failure could seriously compromise our top-level desired 'beacon encapsulation' behaviour. We shall label this fault effect as E_1, with an upper-case E to denote that it is potentially serious.

Effect E_1: Motor failure anchoring the swarm

3.2 Hazard H_2: communications failure

Failure of the network communications sub-system in one or a small number of mobile robots means that those robots become disconnected from the swarm. Given that the basic swarm aggregation mechanism depends upon wireless network communication, then robots with fault H_2 will become physically lost to the swarm and will wander off at random. As far as the swarm is concerned these robots simply become (moving) obstacles to be avoided. The overall swarm behaviours are, however, essentially unaffected. This hazard has, therefore, a relatively benign effect, except of course that the failed robots remain mobile within the environment and - in some circumstances - this may be undesirable. We label this effect e_2, with a lower-case e to denote that it is a non-serious fault.

Effect e_2: Lost robot(s) loose in the environment

3.3 Hazard H_3: avoidance sensor failure

Failure of the avoidance sensor(s) in one or a small number of robots has little effect on overall swarm behaviour. A single robot with failed avoidance sensors will be avoided by the other robots in the swarm and hence have no overall effect. In the unlikely event that two or more robots with failed avoidance sensors collide with each other then physical damage might result from such collisions, but the overall swarm behaviours remain unaffected. It is only when we consider overall swarm behaviours: obstacle avoidance and beacon encapsulation that hazard H_3 becomes a potential problem. Clearly, in these cases, the robots with failed avoidance sensors could collide with either static obstacles in the environment or with the beacon (target). Although not serious this effect might be a problem in some circumstances, we thus label this effect e_3.

Effect e_3: Robot collisions with obstacles or target.

3.4 Hazard H_4: beacon sensor failure

Failure of the beacon sensor in one or a small number of robots has a practically undetectable effect on the overall swarm behaviour. This is because the emergent beacon taxis behaviour results from a differential motion between that part of the swarm whose robots can 'see' the beacon, and the rest of the swarm that cannot (see Nembrini (14) for a detailed explanation of this mechanism). Since this differential is created between two substantive parts of the swarm, the effect of one or a small number of robots with failed beacon sensors is negligible. At worst we might observe a slight slow-down in the swarm taxis behaviour, but this is not judged sufficient to merit labelling as a fault effect.

3.5 Hazard H_5: control systems failure

A control systems failure in one or a small number of robots is very difficult to characterise in terms of its effect on overall swarm behaviour. However, there is every reason to suppose that the likelihood of a failed control system will not seriously compromise the overall swarm behaviours. This is because the control software is very simple; each robot has three behavioural layers: 'forward' (the default behaviour), 'avoidance' (triggered by the avoidance sensors), and 'coherence' (as defined in section 2). Figure 4 shows the control system Finite State Machine (FSM). A

[3] Simulation studies suggest that swarm taxis 'pull' can overcome the anchoring force, in some cases. This phenomenon needs further investigation

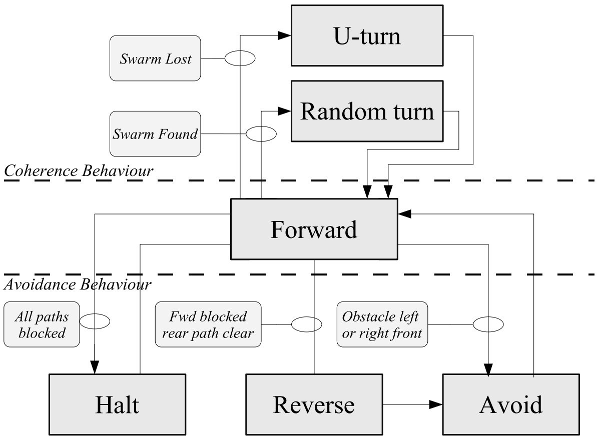

Figure 4: Controller Finite State Machine

control systems failure, if it occurs at all, is most likely to manifest itself as incorrect motor actions. Consider the case that a control systems failure means that the the robot is stuck in its default (forward) behaviour regardless of sensors or network communications; this robot will quickly leave the swarm and be lost in the environment (effect e_2). Its effect on the swarm will be transient and can therefore be ignored in this analysis. A more serious situation would be a control system failure that leaves the robot either stationary or turning on the spot, i.e. effect E_1. This is unlikely, given that the default behaviour is forward, but - to be cautious - let us assume the possibility of this worst case hazard. Logically, E_1 will only compromise beacon taxis and higher order swarm behaviours, thus control systems failure could result (worst case) in e_2 during aggregation and *ad hoc* network behaviours or E_1 during all others.

3.6 Hazard H_6: total systems failure

Complete failure of one or a small number of robots caused, for instance, by power failure will clearly render the robot(s) stationary and inactive. They will be wirelessly disconnected from the swarm and will be treated, by the swarm, as static obstacles to be avoided. Ironically, given that this is the most serious failure at the level of an individual robot, it is the most benign as far as the overall swarm is concerned. Apart from the loss of the failed robots from the swarm, none of the overall swarm behaviours are compromised by this hazard. It is, in fact, the least serious hazard.

To summarise this section, table 3 shows the swarm fault effects, as defined above, generated by one or a small number of robots with hazards $H_1 ... H_6$, for each of the five emergent swarm behaviours defined in section 2. Table 3 clearly shows that the serious swarm failure effect E_1 only occurs in 6 out of 30 possible combinations of robot hazard and swarm behaviour. 15 out of the 30 hazard scenarios have no effect at all on swarm behaviour, and the remaining 9 have only minor, non-serious, effects.

Swarm Behaviour	H_1	H_2	H_3	H_4	H_5	H_6
Aggregation	-	e_2	-	-	e_2	-
Ad hoc network	-	e_2	-	-	e_2	-
Beacon taxis	E_1	e_2	-	-	E_1	-
Obstacle avoidance	E_1	e_2	e_3	-	E_1	-
Encapsulation	E_1	e_2	e_3	-	E_1	-

Table 3: Summary of Failure Modes and Effects

4 SWARM RELIABILITY MODELLING

In this section we explore a number of possible reliability models for a robot swarm. The purpose of a reliability model is to enable the estimation of overall system reliability, given the (known) reliability of individual components of the system, see Elsayed (6). Reliability R is defined as the probability that the system will operate without failure, thus the unreliability (probability of failure) of the system, $P_f = 1 - R$. In our case the overall system is the robot swarm and its components are the individual robots of the swarm.

From a reliability modelling perspective a swarm of robots is clearly a parallel system of N components (robots). If the robots are independent, with equal probability of failure p, then the system probability of failure is clearly the product of robot probabilities of failure. Thus, for identical robots,

$$R = 1 - p^N \tag{1}$$

p can be estimated using a classical reliability block diagram approach on the individual sub-systems of the robot. Since the individual robot does not internally employ parallelism or redundancy then its reliability will be modelled as a series system, giving p less than the worst sub-system in the robot, which is likely to be its motor drive system (refer to hazard H_1 in the previous section).

However, this simplistic modelling approach makes a serious and incorrect assumption, which is that the overall system remains fully operational if as few as one of its components remains operational. This is certainly not true of our case study wireless connected swarm. The desired emergent swarm behaviours require the interaction of multiple robots; our swarm beacon taxis behaviour is a dramatic example: with one robot only the behaviour simply cannot emerge. It is a general charactersitic of swarm robotic systems that the desired overall swarm behaviours are not manifest with just one or a very small number of robots. However, the question of how many (or few) robots are needed in order to guarantee a required emergent behaviour in a particular swarm and for a particular behaviour is often not straightforward.

4.1 A load-sharing approach

This leads us to suggest that a robot swarm should be reliability-modelled as a parallel *load-sharing* system since, in a sense, the overall workload of the swarm is shared between its members. A reliability model of a parallel load-sharing system takes the approach that if one component fails then the probability of failure of the remaining $N-1$ components increases; if a second component fails then the probability of the remaining $N-2$ failing further increases, and so on; see Lee et al (9). While such a model is certainly appropriate for conventional load-sharing systems (think of a 4 engined aircraft with one failed engine, flying on its remaining 3 engines), its applicability is arguable in the case of a robot swarm. Consider our case study. The failure of one or more robots does not intrinsically increase the workload - and hence reduce the reliability - of the remaining, operational, robots. Only in the limited sense that failed robots might increase the task completion (beacon encapsulation) time of the remaining robots might there be an impact on reliability, in that the remaining robots are operational for a longer time. In a robot swarm that does perform work, for example sorting or manipulating physical objects, as in Martinoli et al 2004 (10), then it may be the case that the failure of one or more robots does increase the workload on the remaining robots; in these cases the load-sharing reliability model may be applicable.

4.2 A multi-state approach

Finally let us consider a multi-state approach. The FMEA analysis of the previous section showed that individual robots are not always either fully functioning or completely failed, but could be in one of a number of hazard states that we labelled as $H_1...H_6$. States $H_1...H_5$ correspond to partial failure states, state H_6 is completely failed. The FMEA revealed that the most critical hazard state is H_1, giving rise to swarm failure effect E_1: robot(s) with motor failure 'anchoring' the swarm. (Hazard state H_5 we argued was, in some conditions, equivalent.) Thus, from a reliability point-of-view let us make the simplifying assumption that robots are in one of three states: fully operational, state H_1 or state H_6 completely failed.

If the probability of failure of a robot in state H_1, $p_1 = P(H_1)$ and the probability of failure in state H_6 is $p_6 = P(H_6)$, then clearly the reliability of one robot $r = 1 - p_1 - p_6$. For N robots in the swarm, the reliability of the swarm could be modelled as,

$$R = (1 - p_1)^N - p_6^N \qquad (2)$$

In fact plotting equation 2 for a range of values of N interestingly gives us an optimum value for swarm size in order to maximise the swarm reliability. It is trivial to find the optimum N for given values of p_1 and p_6 by taking the derivative of equation 2 with respect to N and equating to 0 (see (6)). Clearly we expect $p_1 \ll p_6$, but this analysis does give surprisingly low 'optimum' values for swarm size N. For example if $p_6 = 0.1$ (which is rather unreliable) and $p_1 = 0.001$ we find the optimum swarm size is between 3 and 4 robots!

Although this may appear to be a meaningless result, it does tell us, firstly, that with the rather larger swarm sizes that we need in order to bring about the desired emergent swarm behaviours we are operating with a sub-optimal swarm size in terms of reliability. Secondly, and perhaps more importantly, this analysis strengthens the conclusion of the FMEA of the previous section, that we need to endeavour to minimise, or ameliorate, the likelihood of hazard H_1.

5 CONCLUSIONS AND DISCUSSION

This paper has explored fault tolerance in robot swarms by means of Failure Mode and Effect Analysis (FMEA) and reliability modelling. One overall conclusion of this paper is that robot swarms do indeed merit the general characterisation of 'robust', although not just because of their inherent parallelism and redundancy. The high level of robustness is a result of several factors: parallelism of multiple robots; redundancy characterised by a sub-optimal approach to the desired overall swarm functionality (in common with the natural systems from which swarm intelligence takes its inspiration); the fully distributed approach with no 'system-wide' vulnerability to hazards; the functional simplicity of individual robots, and the swarm's unusual tolerance to failure in individual robots. It is useful to reflect on the fact that this level of fault-tolerance comes for free with the swarm intelligence paradigm, that is, without special efforts to achieve fault tolerance by the designer. Contrast this with conventional complex distributed systems that require considerable design effort in order to achieve fault tolerance.

The FMEA case study of this paper has showed that our robot swarm is remarkably tolerant to the complete failure of robot(s) but - perhaps counter-intuitively - is less tolerant to partially failed robots. For the swarm of our case study a robot with failed motors, but all other sub-systems functioning, can have the effect of anchoring the swarm and hindering or preventing swarm motion (taxis toward the target). This leads us to two conclusions (1) analysis of fault tolerance in swarms critically needs to consider the consequence of partial robot failures, and (2) future safety-critical swarms would need designed-in measures to counter the effect of such partial failures. For example, we could envisage a new robot behaviour that identifies neighbours who have partial failure, then 'isolates' those robots from the rest of the swarm: a kind of built-in immune response to failed robots.

Furthermore, we have demonstrated that FMEA is a valuable methodology for swarm robotics systems, and that the use of the swarm intelligence paradigm simplifies FMEA considerably.

This paper's study of reliability models is perhaps less conclusive. The most interesting conclusion is that a multi-state reliability model is needed in order to account for the partially failed robots identified by FMEA. We have shown that a multi-state reliability model can have interesting implications for optimum swarm size (from a reliability perspective), although this finding comes with a clear health warning. Further work is clearly needed to study reliability models for swarm systems including, for instance, study of the k-out-of-n reliability model, in which k would be the minimum number of robots needed for acceptable overall swarm functionality, although how we would determine k in this context is by no means clear. Further work should also investigate combined multi-state k-out-of-n reliability approaches, as in Huang et al, 2000 (7).

In summary, future work should include:

- the study of more quantitative measures of robustness and fault tolerance in robotic swarms;
- the extension of this work to other swarm system case studies in an effort to generalise the approach;
- the development of increased fault tolerance in robot swarms based on an auto immune-response model, and
- further study of reliability models for swarm robotic systems.

ACKNOWLEDGMENTS

The authors gratefully acknowledge formative discussions with Alcherio Martinoli, Guy Théraulaz, Nicolas Reeves and members of the Swarm-Intelligent Systems (SWIS) Group at EPFL during the Summer Research Institute, 2005.

REFERENCES

[1] Anderson T, Avizienis A and Carter WC: Dependability: Basic Concepts and Terminology, Series: Dependable Computing and Fault-Tolerant Systems Volume 5, Laprie, J-C (ed), Springer-Verlag, New York (1992)

[2] Beni G: From swarm intelligence to swarm robotics, in Şahin E and Spears WM (eds) SAB'04 workshop on Swarm Robotics, LNCS 3342, pp 1-9 (2005)

[3] Bonabeau E, Dorigo M, and Théraulaz G: Swarm Intelligence: from natural to artificial systems, Oxford University Press (1999)

[4] Bonabeau E and Théraulaz G: Swarm Smarts, Scientific American, pp 72-79, March (2000)

[5] Dailey KW: The FMEA Handbook, DW Publishing (2004)

[6] Elsayed EA: Reliability Engineering, Addison Wesley Longman (1996)

[7] Huang J, Zuo MJ and Wu Y: Generalised Multi-state k-out-of-n:G Systems, IEEE Trans. on Reliability, **48**(1), (2000)

[8] Kazadi S, Kondo E and Cheng A: A Robust Centralized Linear Spatial Flock, Proc. IASTED International Conference on Robotics and Applications, Hawaii (2004)

[9] Lee SJ, Durham SD and Lynch JD (1995): On the calculation of the reliability of a general load sharing system, Journal of Applied Probability, **32**, pp 777-792 (1995)

[10] Martinoli A, Easton K and Agassounon W: Modeling swarm robotic systems: A case study in collaborative distributed manipulation, Int. Journal of Robotics, **23**(4), pp 415-436 (2004)

[11] Melhuish C: Strategies for Collective Minimalist Mobile Robots, Professional Engineering Publishing (2001)

[12] Mondada F, Guignard A, Colot A, Floreano D, Deneubourg J-L, Gambardella L, Nolfi S and Dorigo M: SWARM-BOT: A new concept of Robust All-Terrain Mobile Robotic System. Technical report, LSA2-I2S-STI, EPFL, Lausanne (2002)

[13] Nembrini J, Winfield A and Melhuish C: Minimalist Coherent Swarming of Wireless Connected Autonomous Mobile Robots, Proc. Simulation of Artificial Behaviour '02, Edinburgh, August (2002)

[14] Nembrini J: Minimalist Coherent Swarming of Wireless Networked Autonomous Mobile Robots, PhD Thesis, University of the West of England, Bristol (2004) download http://swis.epfl.ch/people/julien/

[15] Şahin, E: Swarm Robotics: from sources of inspiration to domains of application, in Şahin E and Spears WM (eds) SAB'04 workshop on Swarm Robotics, LNCS 3342, pp 10-20 (2005)

[16] Winfield AFT, Harper CJ and Nembrini J: Towards Dependable Swarms and a New Discipline of Swarm Engineering, in Şahin E and Spears WM (eds) SAB'04 workshop on Swarm Robotics, LNCS 3342, pp 126-142 (2005)

[17] Winfield AFT, Sa J, Gago MC, Fisher M: Using Temporal Logic to Specify Emergent Behaviours in Swarm Robotic Systems, Proc. Towards Autonomonous Robotic Systems (TAROS), London, September (2005)

[18] Yokobayashi Y, Collins CH, Leadbetter JR, Arnold FH and Weiss R: Evolutionary Design of Genetic Circuits and Cell-Cell Communications, Advances in Complex Systems, **6**(1), pp 37-45 (2003)