

To err is robotic, to tolerate immunological: fault detection in multirobot systems

Danesh Tarapore^{1,2,3}, Pedro U Lima²,
Jorge Carneiro¹, Anders Lyhne Christensen^{4,5}

¹ Instituto Gulbenkian de Ciência, Rua da Quinta Grande 6, 2780-156 Oeiras, Portugal.

² Instituto de Sistemas e Robótica, Instituto Superior Técnico, Av. Rovisco Pais 1, 1049-001 Lisboa, Portugal.

³ Institut des Systèmes Intelligents et de Robotique, Université Pierre et Marie Curie 6, CNRS UMR 7222, F-75252, Paris Cedex 05, France.

⁴ Bio-inspired Computation and Intelligent Machines Lab, Avenida das Forças Armadas, 1649-026 Lisboa, Portugal.

⁵ Instituto de Telecomunicações, Instituto Universitário de Lisboa (ISCTE-IUL), Avenida das Forças Armadas, 1649-026 Lisboa, Portugal.

E-mail: daneshtarapore@gmail.com

Abstract. Fault detection and fault tolerance represent two of the most important and largely unsolved issues in the field of multirobot systems. Efficient, long-term operation requires an accurate, timely detection, and accommodation of abnormally behaving robots. Most existing approaches to fault-tolerance prescribe a characterisation of normal robot behaviours, and train a model to recognise these behaviours. Behaviours unrecognised by the model are consequently labelled abnormal or faulty. Multirobot systems employing these models do not transition well to scenarios involving temporal variations in behaviour (e.g., online learning of new behaviours, or in response to environment perturbations). The vertebrate immune system is a complex distributed system capable of learning to tolerate the organism's tissues even when they change during puberty or metamorphosis, and to mount specific responses to invading pathogens, all without the need of a genetically hardwired characterisation of normality. We present a generic abnormality detection approach based on a model of the adaptive immune system, and evaluate the approach in a swarm of robots. Our results reveal the robust detection of abnormal robots simulating common electro-mechanical and software faults, irrespective of temporal changes in swarm behaviour. Abnormality detection is shown to be scalable in terms of the number of robots in the swarm, and in terms of the size of the behaviour classification space.

1. Introduction

The field of multirobot systems (MRS) has expanded rapidly since its advent in the late 1980s [1, 2], with groups of robots coordinating to perform a wide variety of tasks, ranging from exploration and foraging [3, 4], to site construction, and warehouse-management [5, 6, 7]. The coordination of robots in a MRS is a major challenge because the robot behaviour depends not only on interactions with the immediate environment but also on the behaviour of other robots. A centralised control approach may not always be feasible due to computational and/or communication constraints on robots (e.g., [8, 9]). Distributed control, on the other hand, is often complicated to realise because the behavioural rules for the individual units cannot be easily derived from a desired macroscopic behaviour (e.g., [10, 11, 12]). In the design of large-scale distributed MRS, several researchers have therefore taken inspiration from nature, e.g., aggregation of amoeba into slime mold [13], quorum sensing and communication in bacteria [14], division of labour in social insects such as ants and honey bees (e.g., [4, 3, 15]), and the adaptive immune system in vertebrates (e.g., [16, 17, 18]).

The individual robots of a MRS are susceptible to failure, such as electronic faults on the robot platform, electro-mechanical faults in the robot's sensor and actuation devices, and bugs in the software controlling the robot. Consequent to the wide variety of intricate inter-robot interactions affecting robot behaviour, the prediction and modelling of potential faults to an individual robot is a major challenge. A straightforward approach to fault detection involves the addition of special-purpose hardware to detect some of the common faults (e.g., torque and position sensors for robotic manipulators [19]). However, the additional hardware can greatly increase the cost and complexity of the individual robot platform, and it is therefore preferably avoided in the design of large-scale MRS. Furthermore, the multitude of individual robots constituting a large-scale MRS (e.g., robot swarms), does not inherently result in a fault-tolerant system [20]. Explicit fault-detection systems are therefore critical to enhance the autonomy and operating capacity of MRS.

The engineering of fault-detection systems for robots is a well-studied problem, and can be broadly classified into *endogenous* and *exogenous* models. In endogenous fault detection, robots proprioceptively detect and monitor faults in their individual behaviour (e.g., [21, 22, 23, 24]). In most of these models, artificial neural networks

and radial basis functions are trained to detect faults based on the input-output relationship of the focal component. These approaches have been successfully used to detect faults in components such as, actuator joints of a robotic manipulator [19], wheels [22], and treels (wheels with tracks) [21] of a mobile robot. The large majority of endogenous fault detection models are built on the assumption that the *normal* (no faults present) operating behaviour of the robot is known, and can be characterised beforehand. Consequently, the models are trained to recognise prescribed normal behaviour, and behaviours not recognised by the model are labelled *abnormal*. However, while such approaches provide some interesting results of robust fault detection and fault tolerance, they may not easily transition to different and varying characterisations of normality in collective systems. Transitions in normal behaviour can occur in scenarios wherein robots change their behaviour through online learning, or in response to variations in the environment. In addition, the prior information on the characterisation of normal operating behaviour may not always be available to the system designer. In summary, the endogenous fault detection capabilities of the robot tend to be designed rather specifically to the particular behaviour of the target system, under a specific set of task parameters.

Exogenous fault detection (e.g., [25, 26, 27, 28, 29]) refers to the capacity of one robot can detect the presence of faults in other, physically separated robots. Some faults are difficult to detect endogenously by the robot, such as mechanical failures consequent to a broken connection to a power source, or software bugs in the robot controller. Alternatively, while the robot may be able to proprioceptively detect faults, the fault itself might disable the robot's communication and capability to alert other robots or a human operator. Consequently, in exogenous fault detectors, the multiplicity of robots in MRS allows the robots to robustly and explicitly detect the presence of faults in one another. Fault detection in MRS composed of a limited number of relatively complex robots has been extensively studied in the past (e.g., [25, 26, 27, 28, 29]). For instance, a high degree of fault tolerance was achieved by cooperating teams of robots utilising the ALLIANCE software architecture [25], consequent to a mathematical modelling of the robot's motivation to perform different tasks, and an adaptive task selection based on these motivation. In other approaches (e.g., MURDOCH [26, 27] and TraderBots [28]), fault detection and fault tolerance was built into an explicit inter-robot communication process used by the robots to negotiate task allocation. However, in all these approaches, successful fault detection required prior knowledge of the

various tasks to be performed, and their corresponding measures of performance. Furthermore, these fault-detection approaches were designed for MRS consisting of a limited number of tightly-coupled, and relatively complex robots.

The deployment of large-scale MRS would require scalable and completely distributed algorithms for fault detection, if they are to achieve a high level of fault tolerance. Christensen et al. [30] proposed one such approach for MRS swarms, inspired by synchronised flashing behaviour of fireflies, in which failed robots were exogenously detected by other operational members of the swarm. However, by nature of the underlying model, only faults involving a complete robot failure could be detected. The effect of such faults on the MRS collective behaviour has been shown to be relatively benign [31]. By contrast, robots plagued with partial failures have a far greater potential of disrupting the MRS behaviour [20]. In other work, Lau et al. [16, 32] investigated the detection of errors due to faults in a cooperative foraging MRS operating in a dynamic environment. Various statistical classifiers and kernel density estimation functions were applied to the number of pucks foraged by individual robots, to detect outliers (caused by robot faults). While this approach successfully detected errors with a low false-positive rate and latency time, the faults considered were limited to the wheels of the simulated foraging robot. Additionally, a prior knowledge of the behaviour of the robot was required to design the fault-detection system.

An interesting analogy can be made between fault-detection systems and the adaptive immune system. The adaptive immune system too has to allow the body's cells and tissues to function normally, while mounting an immune response or attacking what may be abnormal cells or tissues (e.g., infected cells, and cancerous cells) [33]. The characteristics of these abnormalities and that of the normal cells and tissues are in principle open-ended, and therefore differ from current approaches to endogenous and exogenous fault detection in robots. Experimental evidence indicates that the tolerance exhibited by the immune system results from the dynamics and interactions between specific regulatory and effector T-cells (e.g., [34]). The decentralised nature of these intercellular interactions imparts a high degree of robustness for fault detection, without the need of a genetically hardwired record of what normal tissues should look like.

In this paper, we present an approach based on the adaptive immune system for exogenous fault detection in large-scale multirobot systems. The main advantages of the proposed approach are that (i) the different behaviours and their characterisation

(normal or abnormal) is learned online and need not be known beforehand, (ii) the approach is completely decentralised and does not rely on task negotiation (or knowledge of tasks), and (iii) detection is not limited to robots experiencing complete failure or any other limited set of faults. Our system is based on the *crossregulation model* (CRM) [35, 36, 37], which is a mathematical model that captures the robust maintenance of immunological tolerance by allowing the system to discriminate between antigens based solely on their density and persistence in the environment. The system is able to tolerate body antigens (the molecular components of body tissues) that are characteristically persistent and abundant, and to mount an immune response to foreign pathogens, that are characterised as being neither persistent nor abundant. In our fault-detection approach, each robots executes an independent instance of the CRM. Previously, we have studied a computationally costly CRM implementation was used directly from [35], for a preliminary assessment of abnormality detection in small-sized MRS [38, 18]. In this study, we present a simplified CRM implementation for decentralised detection of abnormally behaving robots, suitable for large-scale MRS (tested with up to 100 robots). Abnormalities are exhibited with fault-simulating behaviours, such as a robot moving continually in a straight line to mimic a sensor fault. The MRS demonstrates a reliable detection of an introduced abnormally behaving agent, that is surrounded by robots performing typical swarm behaviours (aggregation, flocking, dispersion or homing), and irrespective of multiple temporal changes in the behaviour of the rest of the swarm. Our model is capable of scaling both in terms of the number of robots in the swarm, and in terms of the dimensionality of the space of behavioural features in which the normal and abnormal behaviours are classified. Finally, resilience of the CRM to stochastic variations in behaviour is demonstrated with a comparison of our results with a threshold-based model for normal-abnormal behaviour classification.

The rest of the paper is organised as follows: first, we describe the CRM (Section 2), and approximations to enhance the scalability of the model (Section 3). We then present the distributed embodiment of the CRM in a MRS (Section 4). In Section 5, we report the results of our experiments for different swarm behaviours and under different introduced faults. We vary the swarm size and the number of primitive features used to encode agent behaviour, and we provide a performance comparison of the CRM with a threshold-based model for behaviour classification. Finally, we discuss our approach to fault detection and highlight the conclusions of this study (Section 6).

2. Crossregulation model

The viability of multicellular organisms is consequent to two general principles [39]. Firstly, the persistence of any cell lineage requires that its cells continually interact with the other cell types in the multicellular organism. Cells that fail to interact with other cells eventually die. Secondly, the growth of a cell population involves density-dependent feedback mechanisms controlling individual cell proliferation (division). These feedback mechanisms may involve (i) indirect interactions among cells such as a competition for limited growth factors, and (ii) direct interactions, such as contact inhibition. These two principles are the foundation of the crossregulation model, and have been justified extensively in Carneiro et al. [35]. Below, we outline the model and highlight its key properties that are later embodied distributedly in a multiagent system.

The CRM describes the population dynamics of cells of the adaptive immune system, based on three mutually interacting cell types: (i) antigen presenting cells (APCs) that display the antigen on their surface. Individual APCs have a fixed number of sites (s) on which effector and regulatory cells can form conjugates; (ii) effector cells T_E that can potentially mount an immune response which, depending on receptor specificity, can be directed to foreign pathogens or to self-antigens; and (iii) regulatory cells T_R that suppress proliferation of T_E cells with similar specificities. Furthermore, the APCs are classified into different sub-populations of equivalent APCs, with each APC in a sub-population presenting the same antigen on its surface. Similarly, effector and regulatory cells are also classified into different sub-populations or clones according to their specificity.

The dynamics of the T-cell population is regulated by the following density-dependent feedback mechanisms: (i) effector and regulatory cells that are unable to interact with APCs are slowly lost by cell death; (ii) the proliferation of effector and regulatory cells requires interactions with APCs and depends on interactions these T-cells make with each other. Proliferation of the T_E cell population is promoted by the absence of regulatory cells on the APC. In contrast, T_R can only proliferate following co-conjugation with effector cells on the same APC. Additionally, T_E and T_R cells interact indirectly by competition for access to limited conjugation sites on APCs.

2.1. Mathematical formulation of the model

The dynamics of the interactions between effector and regulatory cells, with APCs is described by a set of ordinary differential equations in the following variables: (i) the number of effector E_i and regulatory R_i T-cells of clonal type i , where $i \in \{1, 2 \dots N\}$ and N is the number of T-cell clones; (ii) the number of APCs A_j , where $j \in \{1, 2 \dots M\}$ and M is the number of different antigen types; (iii) the number of conjugates C_{ij} formed between effector and regulatory cells from clone i and APC from population j .

For the effector E_i and regulatory R_i cells of clone i , we have:

$$\frac{d E_i}{d t} = \pi_E E_i^* - \delta E_i \quad (1)$$

$$\frac{d R_i}{d t} = \pi_R R_i^* - \delta R_i \quad (2)$$

where the involved quantities are defined in Table 1. In this parameter table, the CRM parameters representing the number of binding sites on an APC (s), and the seed population of T-cells (E_0 and R_0), are set based on our previous experiments in using the model for robust and online classification [38, 40, 18]. The remaining CRM parameters of T-cell-APC interactions (γ_c and γ_d), and T-cell proliferation and death rates (π_E , π_R and δ), can be reduced to two key composite parameters representing effective growth rates of effector and regulatory T-cells, that define the parameter regimes of the model (details in Sec. 2.2, and [36]). These model parameters are set from knowledge of the underlying biology; wherein, the T-cell-APC interaction rates are known to be orders of magnitude faster than T-cell proliferation and death rates [35].

The equations for E_i (eq. 1) and R_i (eq. 2) have two terms. The first term represents the proliferation of activated effector E_i^* and regulatory R_i^* cells, and the second term accounts for the death of T-cells, which is assumed to be at constant rate. In the simulations, we generate all T-cell clones with similar initial conditions i.e., $\forall i, E_i(0) = E_0$ and $R_i(0) = R_0$.

The density of activated T_E and T_R cells of each clone are computed in a stepwise manner. Let us consider the interactions between the i -th T-cell clone and the j -th APC population. The dynamics of the conjugates C_{ij} is described by the following equation:

Table 1: Parameters of the crossregulation model.

Param.	Description	Value (a.u.)
A_j	Density of APCs of sub-population j	–
s	Maximum number of T-cells that can bind to an APC	3
E_0	Seed density of effector cells	10
R_0	Seed density of regulatory cells	10
E_i	Density of effector cells of clone i	–
R_i	Density of regulatory cells of clone i	–
T_i	Density of T-cells of clone i	$E_i + R_i$
C_{ij}	Density of conjugates between T_i and A_j	–
γ_c	Conjugation rate of T-cells to APCs	10^{-1}
γ_d	Deconjugation rate of T-cells from APCs	10^{-1}
π_E	Proliferation rate of effector cells	10^{-3}
π_R	Proliferation rate of regulatory cells	0.7×10^{-3}
δ	Death rate of effector and regulatory cells	10^{-6}

$$\frac{dC_{ij}}{dt} = \gamma_c \theta_{ij} \left(T_i - \sum_{j=1}^M C_{ij} \right) \left(A_j s - \sum_{i=1}^N C_{ij} \right) - \gamma_d C_{ij} \quad (3)$$

where $T_i = E_i + R_i$, and γ_c and γ_d involve the conjugation and deconjugation rates between APCs and T-cells, respectively (parameters in Table 1). In the above equation, new conjugates are formed by the free T-cells of clone i with the available sites on APCs of population j at rate γ_c . The conjugation rate is also controlled by the affinity θ_{ij} , between the T-cells and APCs. The existing conjugates dissociate at rate γ_d . As the conjugation and deconjugation of T-cells from the APCs is a fast process with respect to the overall T-cell population dynamics, we numerically integrate to the quasi-steady state values of the conjugates, by the Euler-Heun adaptive step method [41].

The densities of activated effector E_i^* and regulatory R_i^* cells can now be calculated, for all T-cell clones i . Conjugated effector cells are activated in the absence of regulatory cells on the same APC. In contrast, conjugated regulatory cells can only be activated if at least one effector cell is simultaneously conjugated to the same APC. Given the conjugate density C_{ij} at quasi-steady state, the density of conjugated effector and regulatory cells is calculated proportional to the relative frequency of T_E and T_R cells in the clone. For the conjugated effector $E_{C_{ij}}$ and

regulatory Rc_{ij} cells of clone i at APC population j , we have:

$$Ec_{ij} = \frac{C_{ij}E_i}{T_i} \quad \text{and} \quad Rc_{ij} = \frac{C_{ij}R_i}{T_i}$$

Finally, for the number of activated effector E_i^* and regulatory R_i^* cells, we have:

$$E_i^* = \sum_{j=1}^M P_e(A_j, Ec_j, Rc_j) Ec_{ij} \quad (4)$$

$$R_i^* = \sum_{j=1}^M P_r(A_j, Ec_j, Rc_j) Rc_{ij} \quad (5)$$

where the function P_e is the probability that an effector cell is conjugated with no neighbouring regulatory cell at the same APC. P_r is the probability that a regulatory cells is conjugated with an APC that has at least one effector cell conjugated simultaneously. Additionally, Ec_j and Rc_j are the total number of conjugated effector and regulatory cells on the APC sub-population j :

$$Ec_j = \sum_{i=1}^N Ec_{ij} \quad \text{and} \quad Rc_j = \sum_{i=1}^N Rc_{ij}$$

The probability functions P_e and P_r can be reduced to the following expressions, based on a multinomial approximation [42] that is valid given that the total number of sites (summed over all the APCs) is much larger than the number of sites per APC. For three binding sites ($s = 3$) on each APC, we have:

$$P_e(A_j, Ec_j, Rc_j) = \frac{(Rc_j - 3A_j)^2}{9A_j^2} \quad (6)$$

$$P_r(A_j, Ec_j, Rc_j) = \frac{(6A_j - Ec_j)Ec_j}{9A_j^2} \quad (7)$$

Utilising the probability functions P_e and P_r , the density of activated effector and regulatory cells is calculated (eq. 4 and 5). Finally, for each T-cell clone i , the population of effector E_i (eq. 1), and the population of regulatory R_i (eq. 2) cells, are updated.

2.2. Behaviour of the cell population

The population dynamics behaviour exhibited by the CRM is governed by two key composite parameters representing the effective growth rates of T_E and T_R cell populations [36]. These two parameters are directly proportional to the basic parameters controlling population growth i.e., conjugation constant (γ_c), affinity between T-cell and APCs (θ_{ij}), proliferation rates of effector and regulatory cells (π_E and π_R), and the density of APCs (A_j). The effective growth rate of the T-cells is also inversely proportional to the death rate (δ) of the corresponding population. The composite T_E and T_R growth parameters define four parameter regimes according to the resulting cell population behaviour. Three parameter regimes result in a single stable state that may correspond to either: (i) extinction of all T-cells ($T_E = 0$, $T_R = 0$), (ii) immune state ($T_E > T_R$, $T_R = 0$), or (iii) tolerant state ($T_E < T_R$, $T_E > 0$). The fourth parameter regime corresponds to a bistable system where both immune and tolerant states are stable. A detailed analysis of these four parameter regimes is provided in Leon et al. [36].

In the present study, the parameter values of the CRM have been set so that at very low APC densities, the stable state corresponds to extinction of all T-cells ($A < a_E$, Figure 1a). Furthermore, at low APC densities, the system evolves into a single immune state, composed only of effector cells ($a_E \leq A < a_R$, Figure 1a). Finally, at a relatively high density of APCs, the system is bistable ($A \geq a_R$, Figure 1a) and can evolve either into an equilibrium state consisting predominantly of effector cells (immune state), or into a state composed largely of regulatory cells (tolerant state). The system develops into the regulatory cell dominated tolerant state, provided that the seeding population has sufficient T_R cells. By contrast, if T_R cells are initially underrepresented, the T_E cells will competitively exclude the former from the system. Consequently, the CRM is able to discriminate between antigens based on their abundance in the environment.

The CRM is also able to discriminate between antigens based on their persistence in the environment, wherein the outcome of the model depends on the speed and magnitude of the rise in APC density, relative to the proportion of T_R cells present in the seed population. In the bistable parameter regime, whatever the proportion of T_R cells, they may be diluted in forming co-conjugates with T_E cells, if the influx of APCs is sufficiently fast and large. Consequently, the T_E cells would dominate the population, resulting in an immune response (e.g., T_R cell dilution at APC densities

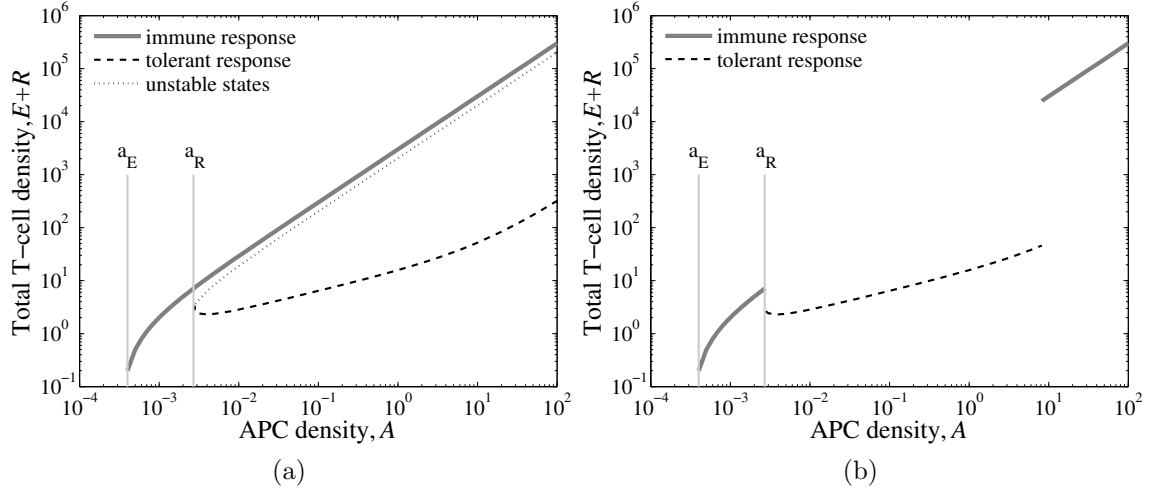


Figure 1: Equilibrium densities of E and R cell populations as a function of APC density. (a) Bifurcation diagram of the CRM, representing all possible equilibrium densities of T-cells (sum of variables, $E + R$), as a function of the APC density (parameter A). (b) Equilibria that are actually reached by solving the system with a fixed seed T-cell population ($E_0 = 4, R_0 = 2$, remaining parameter values in Table 1).

exceeding 10, see Figure 1b). By contrast, if the APC influx is slow and gradual enough, a fraction of T_R cells in the responding population may slowly adapt its size and control the expansion of T_E cells, allowing for a tolerance response.

3. Simplified implementation of the crossregulation model

The numerical integration of the T-cell population is a recurrent two-step process comprising of, (i) the integration of T-cell–APC conjugates to the quasi-steady state, and (ii) utilising the conjugated T-cell densities to update the population of T_E and T_R cells. The numerical integration of the conjugated cells can be avoided, and consequently the computation burden of the model reduced using an approximated analytic expression of the steady-state values of the conjugated cells C_{ij} . This analytic expression is valid in the special condition that T-cells are always in excess to APCs, as proposed by [43]. In this section, we derive such an analytical expression, and

highlight the properties of the resulting CRM implementation.

The dynamics of the conjugates C_{ij} , between the i -th T-cell clone and the j -th APC sub-population can be simplified by assuming that the total T-cell density is in excess of the density of conjugated cells (i.e., for all T-cell clonal types i in the population, $T_i \gg \sum_{j=1}^M C_{ij}$). Consequently, eq. 3 can be reduced to the following equation:

$$\frac{dC_j}{dt} = \gamma_c \left(\sum_{i=1}^N \theta_{ij} T_i \right) (A_j s - C_j) - \gamma_d C_j \quad (8)$$

From eq. 8, the quasi-steady state density of the conjugated cells is calculated as the following function, for each existing APC sub-population j :

$$C_j = \frac{\gamma_c A_j s \sum_{i=1}^N \theta_{ij} T_i}{\gamma_d + \gamma_c \sum_{i=1}^N \theta_{ij} T_i}$$

The total number of conjugated effector and regulatory cells on the APC sub-population j is then calculated, proportional to the relative frequency of T_E and T_R cells, and weighted by their affinity to the APC sub-population j . For the total density of conjugated effector Ec_j and regulatory Rc_j cells at APC sub-population j , we have:

$$Ec_j = C_j \frac{\sum_{i=1}^N \theta_{ij} E_i}{\sum_{i=1}^N \theta_{ij} T_i} \quad \text{and} \quad Rc_j = C_j \frac{\sum_{i=1}^N \theta_{ij} R_i}{\sum_{i=1}^N \theta_{ij} T_i}$$

The total number of conjugated effector and regulatory cells on the APC sub-population j is now factored into the different T-cell clones i in the population. For the conjugated effector Ec_{ij} and regulatory Rc_{ij} cells of clone i at APC sub-population j , we have:

$$Ec_{ij} = Ec_j \frac{\theta_{ij} E_i}{\sum_{u=1}^N \theta_{uj} E_u} \quad \text{and} \quad Rc_{ij} = Rc_j \frac{\theta_{ij} R_i}{\sum_{u=1}^N \theta_{uj} E_u}$$

Finally, utilising eq. 4 and 5 to compute the density of activated effector E_i^* and regulatory R_i^* cells respectively, for each T-cell clone i , the population of effector E_i (eq. 1) and regulatory R_i (eq. 2) cells is updated.

In the implementation of the simplified CRM, the T-cell population is able to discriminate between antigens based only on their abundance, and not on

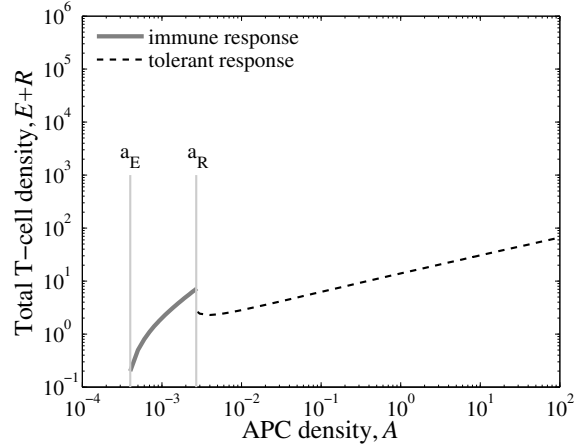


Figure 2: The simplified CRM’s equilibrium densities of E and R cell populations, reached by solving the system for a fixed seed T-cell population ($E_0 = 4$, $R_0 = 2$, remaining parameter values in Table 1).

their persistence in the environment (Figure 2). For the selected parameter values (Table 1), at low APC densities, the system evolves towards a single immune state composed only of effector cells. By contrast, at relatively high density of APCs, the system exhibits bistability, and evolves into the regulatory cell dominated tolerance state, provided that the seeding population has sufficient T_R cells.

The model’s capability to discriminate between the antigens in the environment based only on the density, allows for a reactive abnormality detection system. In the resulting system, the abnormally behaving agents are detected immediately upon their occurrence, and without any delay in allowing for the abnormality to establish persistence in the swarm.

4. The CRM in a Multiagent System

In this section, we demonstrate how the CRM can be implemented for a distributed embodied multiagent system in order to give the system the capacity to detect abnormally behaving agents, while maintaining a tolerance towards normal swarm behaviour. Behaviours that are abundant (performed by many agents in swarm) are to be tolerated. By contrast, rare behaviours (exhibited by a single or few agents)

are to be detected as abnormal behaviours that may be caused by a fault. We show that the multiagent system is able to detect abnormally behaving agents, and adapt online and tolerate different normal behaviours.

We assume that the agents can observe neighbouring robots over a certain period of time with sufficient accuracy to characterise their behaviour. On real robots, the sensory equipment necessary to facilitate observations is task-dependent. Movement-centric behaviours, such as the traditional swarm behaviours used in this study, could be characterised by robots equipped with relatively localisation hardware (e.g. an infrared range and bearing system [44]), while for other robots and tasks in which body posture is relevant, sensors such as depth cameras or laser scanners could be used.

We use a stochastic, spatial, discrete-time multiagent system simulator. The simulated environment is toroidal and has a size of $5 \times 5 \text{ m}^2$, and composed of 20 mobile agents of diameter 7.5 cm. Each agent has a maximum speed of 10 cm/s, with a control cycle of 0.1 s, and performs the behaviour assigned to it at the start of the simulation. During the simulation, each agent senses the behaviour of its ten nearest neighbours, and runs an internal and individual instance of a CRM in order to determine if the perceived behaviours of the neighbouring agents are to be tolerated or not. In the private CRM instance running on each agent, different APC subpopulations are associated with distinct perceived behaviours. The APCs stimulate the growth of virtual T_E and T_R cell populations of different clonal types (eq. 1 and 2) to immune or tolerance stable states, to determine if the behaviours observed are normal or abnormal, respectively.

Swarm behaviours: The normal swarm behaviours simulated are (a) dispersion, (b) aggregation, (c) flocking, and (d) homing towards a moving landmark. The behaviours are implemented using a subsumption architecture [45]. According to the designed architecture, the basic behaviour allows an agent with no neighbours to perform a random walk. In dispersion, the agents move in the opposite direction of the centre of mass of their neighbours (Figure 3a). By contrast, in aggregation, the agents move towards the centre of mass of surrounding agents, but disperse away if too close to their neighbours (Figure 3b). Similarly, homing agents move towards a single prespecified agent that serves as a moving landmark, and move away if too close to the landmark or to other agents (Figure 3c). The landmark agent for homing is selected at random at the start of the experiment. Finally, in flocking (Figure 3d), agents continually adjust their velocity to that of neighbouring agents. Furthermore,

Table 2: Parameters of an agent

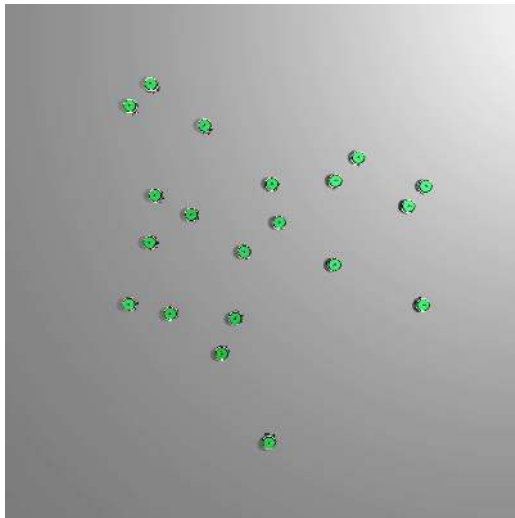
Param.	Description	Value
$ \vec{v}_{\max} $	Maximum linear speed of agent	10 cm/s
$ \vec{v} $	Linear speed of agent	—
ω_{\max}	Maximum change in direction of agent per control cycle	π radians
ω	Change in direction of agent per control cycle	—
\vec{u}	Average velocity of neighbours in 60 cm range	—
ϕ_{vu}	Direction of relative velocity to neighbours in 60 cm range	$\arccos\left(\frac{\vec{v}\cdot\vec{u}}{ \vec{v} \vec{u} }\right)$ radians
n_i	Number of neighbouring agents in the inner range of [0, 30] cm	—
n_o	Number of neighbouring agents in the outer range of (30, 60] cm	—
W	Length of the time window for feature computation	45 s
p	Distance traversed by the agent in the past W s	—

flocking agents, aggregate towards and disperse from neighbours, if they are too far away or close by, respectively.‡

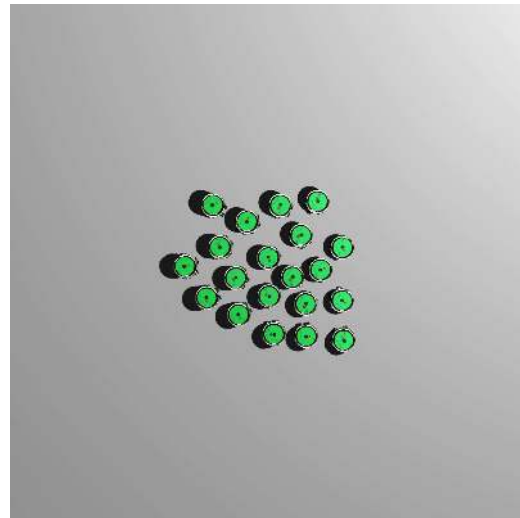
Faulty behaviours: An agent may behave abnormally so as to: (a) move continually in a straightly line (STRLN); (b) perform a random walk, with a 0.01 probability of changing to a new random direction each simulation control cycle (RNDWK); (c) circle with diameter 10 cm around a fixed point (CIRCLE); or (d) stop completely (STOP). These additional behaviours are introduced to mimic: (a) software bugs and sensor faults in the agent controller (STRLN and RNDWK); (b) motor malfunctions or blocked wheel (CIRCLE); and (c) a broken or dead battery (STOP).

Encoding of agent behaviour: We divide an agent’s behaviour into three different classes: (i) the agent’s immediate environment (sensors), (ii) the agent’s actions (actuators), and (iii) the agent’s response to events (sensorimotor interactions). Behavioural features from each class are then used to characterise an

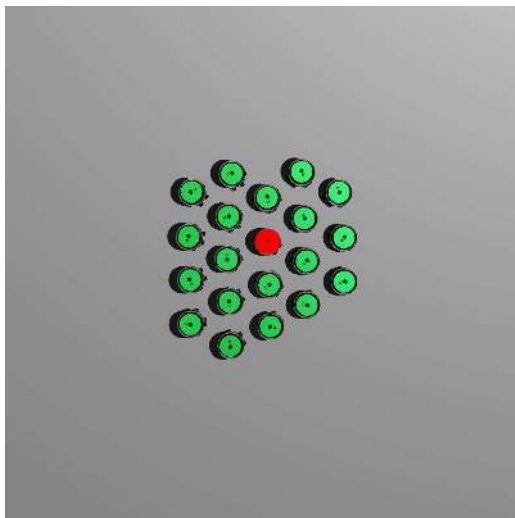
‡ Simulation source code can be downloaded from <http://www.isr.ist.utl.pt/~dtarapore/scalablefaultdetection>.



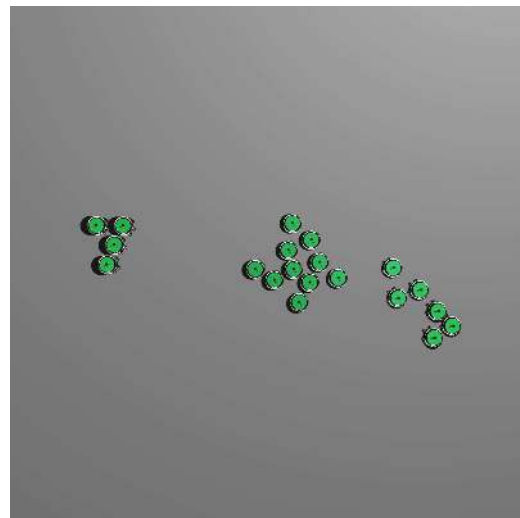
(a) Dispersion



(b) Aggregation



(c) Homing towards moving landmark



(d) Flocking

Figure 3: Examples of the normal behaviours exhibited by the swarm.

agent's behaviour. Individual features are encoded in Boolean form (present= 1, absent= 0), and then concatenated to form a binary string, the *feature vector* (FV). In our simulations, a FV comprises the concatenation of six features ($F_1, F_2 \dots F_6$), with two features from each behaviour class. The features are computed utilising the behavioural information collected over a moving time window of W s. Consequently, the computation of the FV is initiated after W s of the simulation.

The first two features, $F_1(\tau)$ and $F_2(\tau)$ at time τ , pertain to the agent's immediate environment, i.e., neighbours in proximity:

$$F_1(\tau) = 1 \text{ if } \frac{\sum_{t=\tau-W}^{\tau} U[n_i(t)]}{W} > 0.5, \text{ otherwise } F_1(\tau) = 0 \quad (9)$$

$$F_2(\tau) = 1 \text{ if } \frac{\sum_{t=\tau-W}^{\tau} U[n_o(t)]}{W} > 0.5, \text{ otherwise } F_2(\tau) = 0 \quad (10)$$

where n_i and n_o are the number of neighbours in the inner ($[0, 30]$ cm), and outer ($(30, 60]$ cm) range, respectively. Furthermore, $U[n]$ is the unit step function, defined as:

$$U[n(t)] = \begin{cases} 1, & \text{if } n(t) > 0 \\ 0, & \text{otherwise} \end{cases}$$

At time τ , the features $F_1(\tau)$ and $F_2(\tau)$ are set, if the agent has at least one neighbour in range $[0, 30]$ cm and $(30, 60]$ cm, respectively, for the majority of the past W s (see parameters in Table 2).

The next two features, $F_3(\tau)$ and $F_4(\tau)$, pertain to the agent's motors. For these two features, we have:

$$F_3(\tau) = 1 \text{ if } p(\tau) > 0.05W|\vec{v}_{\max}|, \text{ otherwise } F_3(\tau) = 0 \quad (11)$$

$$F_4(\tau) = 1 \text{ if } |\vec{v}(\tau)| > 0.05|\vec{v}_{\max}|, \text{ otherwise } F_4(\tau) = 0 \quad (12)$$

where, at time τ , $p(\tau)$ cm is the distance traversed by the agent in the past W s, speed $|\vec{v}(\tau)|$ cm/s, and maximum speed $|\vec{v}_{\max}|$ cm/s. The feature $F_3(\tau)$ is set, if $p(\tau)$, exceeds 5% of the maximum distance that may be traversed by the agent in W s. Similarly, $F_4(\tau)$ is set, if $|\vec{v}(\tau)|$ cm/s exceeds 5% of the maximum agent speed. For both features $F_3(\tau)$ and $F_4(\tau)$, the 5% threshold is set to compensate for stochastic variation in agent behaviour.

Finally, for the last two features, $F_5(\tau)$ and $F_6(\tau)$, pertain to the agent's sensorimotor interactions. For these interactions, we define two sensorimotor interaction events S_m and S_n , as follows:

$$\begin{aligned} S_m(\tau) &= U[n_i(\tau) + n_o(\tau)] \wedge U[|\omega'(\tau)| - 0.03\omega'_{\max}] \\ S_n(\tau) &= \neg U[n_i(\tau) + n_o(\tau)] \wedge U[|\omega'(\tau)| - 0.03\omega'_{\max}] \end{aligned}$$

The above sensorimotor interaction event $S_m(\tau)$ is set, if the agent's angular acceleration exceeds ± 0.1 radians (3% of maximum angular acceleration), in the presence of sensory input (one or more neighbours in range). Similarly, $S_n(\tau)$ is set, for the same motor response, in the absence of sensory input (no neighbours in range). The angular acceleration threshold in registering a motor response is set suprazero at 3% of maximum angular acceleration, to compensate for stochastic variation in agent behaviour.

Consequently, for the the features $F_5(\tau)$ and $F_6(\tau)$,

$$F_5(\tau) = 1 \text{ if } \sum_{t=\tau}^{\tau-W} S_m(t) > 0, \text{ otherwise } F_5(\tau) = 0 \quad (13)$$

$$F_6(\tau) = 1 \text{ if } \sum_{t=\tau}^{\tau-W} S_n(t) > 0, \text{ otherwise } F_6(\tau) = 0 \quad (14)$$

where, the features $F_5(\tau)$ and $F_6(\tau)$ are set if the sensorimotor interaction events S_m and S_n respectively, occur at least once in time window W .

Immunological response to agent behaviour: At the start of each time-step, an agent senses the behaviour of its ten nearest neighbours as 6-bit FVs according to eqs. 9-12, and computes the number of agents assigned to each 6-bit FV (FV_j). In the agent's internal CRM instance, APCs are then generated corresponding to each of the feature vectors perceived. Each APC presents an individual FV to the T-cells. The number of each type of the APCs generated $A_j = kFV_j$, for $j \in \{1, \dots, M\}$, where k is a scaling constant, and M is the maximum number of different feature vectors perceived by the agent ($M = 2^6$ for 6-bit FV).

The T-cell clones (T_1, T_2, \dots, T_N), each have a different receptor encoded as a binary string, which determines their affinity to the APC sub-population. The affinity between T-cell clonal i and APC sub-population j is denoted by θ_{ij} :

Table 3: Parameters of the stochastic simulator

Param.	Description	Value (a.u.)
l	Length of binary feature vector	6 bits
M	Maximum number of different feature vectors	2^l
N	Maximum number of T-cell clones	2^l
c	Cross-reactivity between T-cells and APCs	0.15
I_E	Density of new effector cells introduced at each simulation time-step	10
I_R	Density of new regulatory cells introduced at each simulation time-step	10
k	FVs to APCs scaling factor	0.002
S	Time CRM instance is executed, in a single simulation time-step	5×10^7
d	Proportion of T-cells diffused to neighbouring agents	0.5

$$\theta_{ij} = \exp\left(-\frac{H[i, j]}{cl}\right) \quad (15)$$

where H is the Hamming distance between the receptor of T_i and the FV presented by A_j , l is the length of the FV, and c is the cross-reactivity between T-cells and APCs. A high value of c would result in all T-cell clones having a high affinity to all APC sub-populations. By contrast, at low c , each T-cell clone would have a high affinity to only one distinct APC sub-population.

At the start of the simulation, the number of effector and regulatory cells on each agent are initialised to E_0 and R_0 respectively. Following this, Algorithm 1 (parameters in Table 3) is executed by the agents in each simulation time-step, to simulate their internal CRM. The agents begin by sensing their neighbours, and computing the distribution of feature vectors. The CRM is then numerically integrated for time S , allowing the system to respond to the different APCs.

After computing the number of effector and regulatory cells at time S , the cells diffuse among the agents. In this communication phase, each agent selects at random another agent, from one of its ten nearest neighbours. Following the

selection, each agent sends and receives d of its effector and regulatory cells. In our previous work [38], we found that communication of cells yields a higher performance when agents move between regions with different feature vector distributions. Finally, the agent decides the nature of each FV FV_j sensed by first computing the following quantities:

$$E = \sum_{i=1}^N \theta_{ij} E_i \quad R = \sum_{i=1}^N \theta_{ij} R_i$$

and tolerating the FV if $R > E$. By contrast, if $E > R$, the FV is classified as faulty by the agent.

The parameters of T-cell proliferation and death rates are set so that an agent tolerates a FV if it is expressed by more than a single neighbouring agent. Additionally, the cross-reactivity parameter c is set so that the FV expressed by a single agent is tolerated if it differs by less than $l/3$ features from any of the perceived FVs being expressed by two or more neighbours. Otherwise, the perceived FV is classified as faulty by the agent.

In the simulation of the CRM instance for an agent, the number of different APC sub-populations and T-cell clones are limited to the number of observed neighbouring agents. Consequently, in the implementation of the of Algorithm 1, a linked-list data-structure is used to explicitly represent the existing APC sub-populations, T-cell clonal types and their conjugates.

5. Experiments

The main advantage of a CRM-based approach is that classification of normal vs. abnormal behaviour is continually and autonomously learnt online while the agents operate in the task environment. Prominent advantages of the online learning of the classification include that normal behaviour need not be prescribed a priori, and that the performance is unaffected by changes to normal behaviour as long as most or all agents change their behaviour simultaneously. In the following, we assess the performance of our CRM-based approach in different scenarios. We first evaluate the capacity of a multirobot system to detect abnormal behaviour for different combinations of normal and fault-simulating behaviours (Section 5.1). We then evaluate how frequently normally behaving agents are misclassified as behaving abnormally (Section 5.2), and the influence of transitions in normal behaviour on

Algorithm 1 An agent’s control loop (simulation of a CRM instance)

- 1: {Perceive neighbouring agents}
 - 2: Compute distribution of feature vectors (FV_j) of neighbouring agents
 - 3: Assign feature vectors to APCs i.e., $\forall j, A_j = kFV_j$
 - 4: {Influx of new T-cells}
 - 5: $\forall j \in \{1, 2 \dots M\}$, if $A_j > 0$, increment E_j and R_j by I_E and I_R , respectively
 - 6: {Run instance of CRM}
 - 7: $time \leftarrow 0$
 - 8: **while** $time \leq S$ **do**
 - 9: $\forall i \in \{1, 2 \dots N\}$ and $T_i > 0$, and $\forall j \in \{1, 2 \dots M\}$ where $A_j > 0$, compute the number of conjugated cells C_{ij} in quasi-steady state.
 - 10: Using the number of conjugated cells, compute the updated number of effector and regulatory cells with the Euler-Heun adaptive step method [41]. The adaptive step size is stored in h
 - 11: $time \leftarrow time + h$
 - 12: **end while**
 - 13: {Diffuse cells across neighbouring agents}
 - 14: Randomly select one of the agents in the communication range following a linear distribution and weighted by the total number of cells on the respective neighbouring agents
 - 15: Exchange cells with agent
 - 16: {Decide if feature vectors are to be tolerated or not}
 - 17: For each feature vector, compute the sum of effector and regulatory cells, weighted by their affinity.
 - 18: Tolerate the feature vector if total regulatory cells exceeds effectors, else interpret it as faulty. Log the outcome of the classification.
-

performance (Section 5.3). The scalability of our CRM-based approach is then evaluated, first in terms of the dimensionality of the FV space (Section 5.4), and then in terms of swarm size (Section 5.5). Finally, we compare the performance of our CRM-based approach with the performance of a threshold-based approach to behaviour classification (Section 5.6).

5.1. Detection of abnormalities

We ran experiments with 20 randomly placed agents in a 5×5 m² toroidal environment. In the swarm, 19 of the 20 agents performed one of the normal behaviours, that is, aggregation, dispersion, flocking or homing, while one agent performed one of the fault-simulating behaviours, STRLN, RNDWK, CIRCLE and STOP. The FVs used by the CRM-based abnormality detector were composed

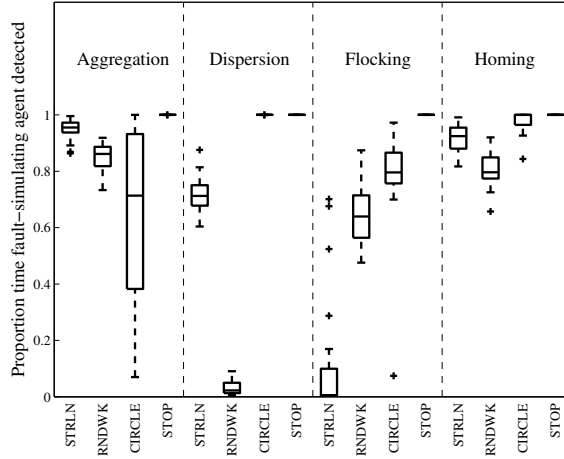


Figure 4: **Detection of abnormalities.** Proportion of time fault-simulating agent is detected as abnormal across 20 replicates, in each of the 16 distinct combinations of normal (aggregation, dispersion, flocking and homing) and abnormal (STRLN, RNDWK, CIRCLE and STOP) behaviours.§

of $(F_1, F_2 \dots F_6)$ (see Section 4). We ran 20 replicates for each combination of normal behaviour and fault-simulating. Each replicate lasted 15,000 cycles (corresponding to 1,500 s), and we recorded the proportion of time during which the fault-simulating agent was correctly detected as behaving abnormally by neighbouring agents. The results are summarised in the box-plot in Figure 4 with one box for each combination of normal and fault-simulating behaviour.

The results of experiments with three combinations of normal and fault-simulating behaviour are prominent: (i) in aggregation/CIRCLE, the distribution of proportion of time the fault-simulating agent is detected is wide; (ii) in dispersion/RNDWK, the proportion of time the fault-simulating agent is detected is very low (median: 0.023); and (iii) in flocking/STRLN, the proportion of time the fault-simulating agent is detected is low in the majority of the replicates. In all three cases, the relatively low proportion of time that the fault-simulating agent

§ On each box, the mid-line marks the median, and the box extends from the lower to upper quartile below and above the median. Whisker outside the box indicate the maximum and minimum values, except in case of outliers, which are shown as crosses. Outliers are data points outside of 1.5 times the interquartile range.

is detected is caused by similarities between the fault-simulating behaviour and the normal behaviours in certain situations: in the aggregation/CIRCLE experiments, some or all normally behaving agents may aggregate around the circling agent causing it to become a member of an aggregate, and detected as such. In the dispersion/RNDWK experiments, all agents disperse away from one another and from the fault-simulating agent performing random walk. As soon as the fault-simulating agent approaches a normally behaving agent, the normally behaving agent will disperse away from the random walking agent. All the robots in the MRS will therefore be dispersed and the fault-simulating agent is thus difficult to detect. Finally, in several flocking/STRLN replicates, normally behaving robots end up forming a flock around the fault-simulating agent moving in a straight line. The relatively low proportion of time that the fault-simulating is detected in many of the aggregation/CIRCLE, dispersion/RNDWK, and flocking/STRLN experiments are thus due to the self-organised behaviour of the swarm that makes the fault-simulating behaviour of a single agent indistinguishable from normal behaviour.

5.2. Tolerance to normal behaviour

The effectiveness of a fault detection system depends as much on its capacity to avoid false positives and correctly classify normal behaviour as its capacity to detect faults. Consequently, we evaluated our CRM-based approach to abnormality detection in a series of experiments in which all 20 agents behaved normally, and we measured the proportion of time the agents were correctly classified as behaving normally (tolerated) by their neighbours. In Figure 5, we have plotted the mean proportion of time that agents are tolerated in each of 20 replicates of the four normal swarm behaviours (horizontal axis), and the variation between the 20 agents observed in each replicate calculated as the difference between the maximum and minimum time tolerated (vertical axis).

The mean proportion of time that normally behaving agents were correctly classified was high across all experiments, at 0.9974 ± 0.002 for aggregation, 0.9819 ± 0.002 for dispersion, 0.9966 ± 0.002 for flocking, and 0.9997 ± 0.001 for homing (mean \pm SD). The variation in time tolerated between the agents of the swarm, in individual replicates, is low (less than 0.1) for all four swarm behaviours (see Figure 5).

The CRM-based abnormality detector classifies the behaviour of neighbouring

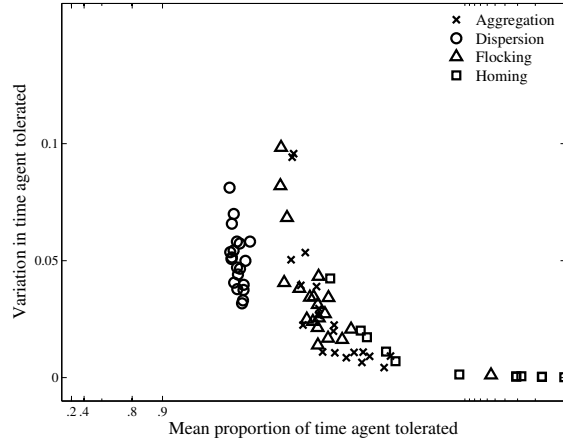


Figure 5: **Tolerance to normal behaviour.** Mean and variation in proportion of time agent tolerated, across the 20 agents of the MAS, in each of 20 replicates of four swarm behaviours. Variation measured in each replicate as the difference between the maximum and minimum time tolerated, across the 20 agents.

agents at every control cycle. The results presented in Figures 4 and 5 for abnormality detection and for tolerance of normal behaviour, respectively, correspond to the direct, uninterpreted output of the CRM-based abnormality detector. A more reliable classification of behaviour can be obtained by accumulating the output over a series of consecutive control cycles instead of classifying the behaviour of neighbouring agents based on the output in a single control cycle. Depending on the relative cost of tolerating an abnormally behaving agent (false negatives) and of incorrectly classifying an agent as behaving abnormally (false positives), the output may be interpreted in one of several ways. A simple scheme involves storing the past n outputs of the CRM-based abnormality detector and only classifying an agent as behaving abnormally if it has been detected as such for the majority or all past n control cycles. If fault accommodation is expensive, while the presence of abnormally behaving agents has a relatively small impact on performance, n could be set to a relatively high value. Conversely, in critical tasks where abnormal behaviour can be catastrophic, n could be set to a relatively low value. With $n = 100$ (corresponding to 10 s), wherein only agents detected abnormal in all n control cycles were interpreted as behaving abnormal, we observed a significant reduction in the number of false

positive incidents compared to when the output of the CRM-based classifier is used directly (corresponding to $n = 1$): with $n = 1$, the median number of false positive incidents per replicate was 15 (max. 38) across all experiments, while for $n = 100$, the median number of false positive incidents was 0.5 (max. 1). A false positive incident corresponds to a series of consecutive simulation cycles in which an agent is incorrectly being classified as behaving abnormally, thus, to the number of times that a swarm would incorrectly have taken steps to accommodate a non-existing fault. The trade-off between false positives and false negatives can thus be fine-tuned based on the task and its parameters.

5.3. Response to changes in agent behaviour

One of the main advantages of a CRM-based approach is that classification of normal vs. abnormal behaviour is continually learnt online while the agents operate in the task environment. Continual learning of the classifier potentially allows for agents to change their behaviour without being classified as abnormal as long as a sufficient number of agents follow suit. We setup a series of experiments to assess the CRM-based approach’s capacity to tolerate changes to agent behaviour. We first evaluated the performance of our CRM-based abnormality detector in three different setups in which all agents transitioned from one behaviour to another and then back again to the original behaviour. Each phase of an experiment lasted 5,000 control cycles (corresponding to 500 s). We used three different pairs of behaviours: (a) aggregation and dispersion, (b) aggregation and flocking, and (c) aggregation and STOP. It should be noted that the STOP behaviour in our experiments in the previous section (Section 5.1) was considered one of the abnormal behaviours because it was only performed by a single agent, but that when all agents perform the behaviour, it should be considered normal and therefore tolerated. We conducted 20 replicates in each setup and the results are shown in Figure 6 where we have plotted the mean time that agents are tolerated in each experiment (horizontal axis) and the variation in each replicate (vertical axis).

5.3.1. Change in behaviour of all agents The proportion of time that the normal behaviour is tolerated in the experiments in which the normal behaviour switches from aggregation to either dispersion, flocking or STOP (see Figure 6) is comparable to the proportion of time that normal behaviour is tolerated when the normal

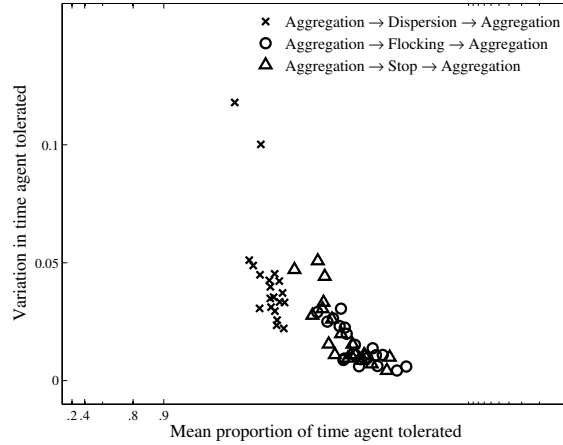


Figure 6: **Tolerance to changes in normal behaviour.** Mean and variation in proportion of time agent tolerated, across the 20 agents of the MAS, in each of 20 replicates, and two transitions in normal behaviour: (a) aggregation to dispersion to aggregation (crosses), (b) aggregation to flocking to aggregation (circles), and (c) aggregation to STOP to aggregation (triangles).

behaviour remains constant throughout an experiment (see Figure 5). Only in the experiments in which behaviour was changed from aggregation to dispersion and back again did we observe a slight drop in (uninterpreted) classification performance immediately after the behaviour had been switched. Around 50 s after the agents switch from dispersion back to aggregation, some agents will have aggregated while others are still dispersed and moving toward an aggregate. Therefore, the differences in observable behaviour can briefly lead the dispersed agents to be classified as abnormal.

We analysed the distribution of FVs during the experiments in which all agents changed behaviour, see Figure 7. As it can be seen in the figure, the distribution of FVs changes as the behaviour changes, and several different FVs are present in the population of robots at any given time.

5.3.2. Change in behaviour of single agent We ran a set of experiments in order to determine if an agent that starts to behave abnormally halfway through an experiment can be detected reliably. For the first half of the experiment, all agents

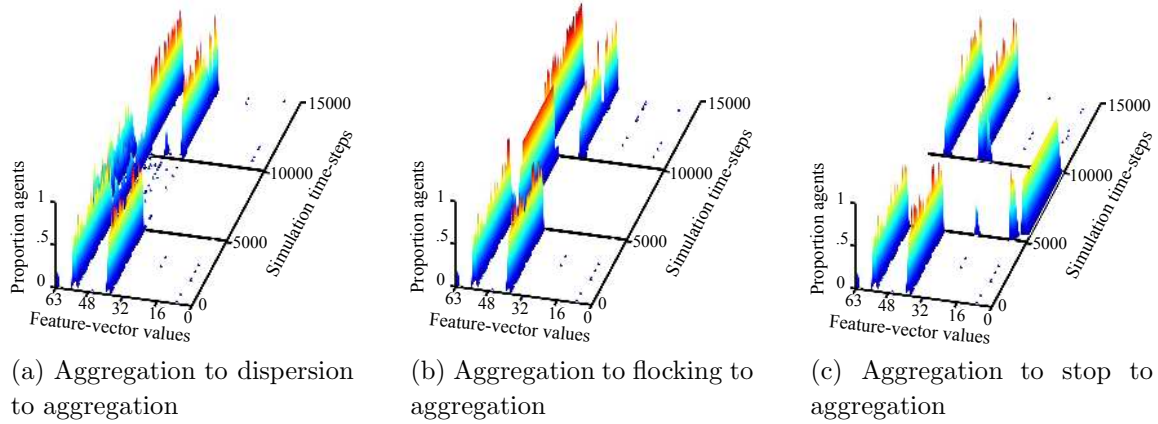


Figure 7: **Behaviour FV distribution.** Distribution of 6-bit FVs across the 20 agents of the swarm during 15, 000 simulation control cycles, and with transitions in swarm behaviour occurring at the 5, 000th and 10, 000th control cycle.

performed aggregation, while at the midpoint, a single agent switched its behaviour to dispersion. Across 20 replicates, the mean proportion of time a normal aggregating agent was tolerated during the first half of the simulation was 0.99 ± 0.001 , while the mean proportion of time that the same agent was detected as behaving abnormally in the second half of the experiments was 0.70 ± 0.386 . However, in 5 of the 20 replicates conducted, the focal agent was surrounded by other aggregating agents when it switched from performing aggregation to dispersion, which explains the low performance compared to experiments in which an abnormal behaviour is performed from the start of an experiment (see Figure 4).

In interpreting the CRM classification over a series of control cycles, our results revealed a more reliable classification, despite the variations in behaviour. In classifying a behaviour as abnormal if it was detected abnormal in all of the past n control cycles ($n = 100$), no more than one false-positive incident was recorded for the normally behaving aggregating agents. By contrast, with $n = 1$, the maximum number of such incidents was 430. The introduction of this temporal filter has an impact on the latency in detection of the abnormal agent. In the absence of the filter, the abnormal dispersing agent was detected with a median latency time of 11.6 s (max. 481.2 s). With the filter ($n = 100$), the latency time was 26.7 s (max. 491.1 s), registering an increase by 44.0% when discounting the length of the time

Table 4: Binary features encoding agent behaviour (parameters in Table 2).

Notation	Value at time τ^*
$F_1(\tau)$	$\frac{\sum_{t=\tau}^{\tau-W} U[n_i(t)]}{W} > 0.5$
$F_2(\tau)$	$\frac{\sum_{t=\tau}^{\tau-W} U[n_o(t)]}{W} > 0.5$
$F_3(\tau)$	$p(\tau) > 0.05W \vec{v}_{\max} $
$F_4(\tau)$	$ \vec{v}(\tau) > 0.05 \vec{v}_{\max} $
$F_5(\tau)$	$\sum_{t=\tau}^{\tau-W} U[n_i(\tau) + n_o(\tau)] \wedge U[\omega'(\tau) - 0.03\omega'_{\max}] > 0$
$F_6(\tau)$	$\sum_{t=\tau}^{\tau-W} \neg U[n_i(\tau) + n_o(\tau)] \wedge U[\omega'(\tau) - 0.03\omega'_{\max}] > 0$
$F_7(\tau)$	$n_i(\tau) > 0$
$F_8(\tau)$	$n_o(\tau) > 0$
$F_9(\tau)$	$\sum_{t=\tau}^{\tau-W} U[n_i(\tau) + n_o(\tau)] \wedge U[\omega(\tau) - 0.03\omega_{\max}] > 0$
$F_{10}(\tau)$	$\sum_{t=\tau}^{\tau-W} \neg U[n_i(\tau) + n_o(\tau)] \wedge U[\omega(\tau) - 0.03\omega_{\max}] > 0$
$F_{11}(\tau)$	$U[\vec{v}(\tau) - 0.05 \vec{v}_{\max}] > 0$
$F_{12}(\tau)$	$U[\omega(\tau) - 0.03\omega_{\max}] > 0$
$F_{13}(\tau)$	$U[\omega'(\tau) - 0.03\omega'_{\max}] > 0$
$F_{14}(\tau)$	$U[\vec{v}(\tau) - \vec{a}(\tau) - 0.05 v_{\max}] > 0$
$F_{15}(\tau)$	$U[\phi_{vu} - 0.03\omega_{\max}] > 0$

*The binary feature is set if the condition is satisfied, else 0.

series, and the three replicates where the abnormal agent was never detected.

5.4. Scalability of CRM in the FV space

In this section, we study the scalability of our CRM-based approach with respect to number of behavioural features considered by the abnormality detector. The main questions are: (i) if and how the inclusion of additional features affects the accuracy of abnormality detection, and (ii) what is the impact of the inclusion of additional features on the computational cost of running the CRM-based abnormality detector? We ran a series of experiments to assess performance and resources, for FV spaces of different dimensionalities: 6, 9, 12, and 15 features. We considered FVs of, 6 features ($F_1, F_2 \dots F_6$), 9 features ($F_1, F_2 \dots F_9$), 12 features ($F_1, F_2 \dots F_{12}$), and 15 features ($F_1, F_2 \dots F_{15}$). All features are computed based on behavioural information over a moving time window of W s. The nine new features ($F_7, F_8 \dots F_{15}$), pertain to the agent's immediate environment (F_7, F_8), sensorimotor interactions (F_9, F_{10}), and absolute (F_{11}, F_{12}, F_{13}) and relative (F_{14}, F_{15}) motor outputs as detailed in Table 4.

Results of experiments in which FVs of different lengths were used are

Table 5: Experimental setups for results shown in Figures 8, 11, and 12.

Setup	Description
(A)	Detection: a single agent in the swarm simulates a fault. The fault-simulating behaviours considered are STRLN, RNDWK, CIRCLE and STOP, while the normal behaviours for the rest of the agents in the swarm are: aggregation, dispersion, flocking, and homing, resulting in a total of 16 combinations. Each box in the plot summarises the proportion of time the fault-simulating agent is detected in 20 replicates in each of the 16 experimental setups (a total of 320 experiments per box).
(B)	Tolerance: all agents in the swarm behave normally. The normal behaviours considered are: aggregation, dispersion, flocking, and homing. Each box in the plot summarises the proportion of time that the normally behaving agents are tolerated in 20 replicates for each of the four normal behaviours (a total of 80 experiments per box).
(C)	Changes in normal behaviour: all agent synchronously transition from aggregation to an alternative behaviour (dispersion, flocking, or STOP) and back to aggregation during an experiment. The first transition in behaviour occurs one-third into the experiment, and the second transition occurs two-thirds into the experiment. Each box in the plot summarises the proportion of time the agents are tolerated in 20 replicates for each alternative behaviour (dispersion, flocking, or STOP, thus, a total of 60 experiments per box). Note that when most, or all of the agents in the swarm perform the STOP behaviour, it is considered the normal behaviour.
(D)	Changes in the behaviour of one agent: a single agent transitions from the normal behaviour performed by the rest of the swarm (aggregation) to an abnormal behaviour (dispersion) halfway through the experiment. Each box in the plot summarises the proportion of time that the behaviour of the focal agent is correctly classified by its neighbours, that is, the proportion of time that it is tolerated while performing the same behaviour as the rest of the swarm, and then the proportion of time that it is detected as behaving abnormally. We ran 20 experimental replicates in the experiment setup.

summarised in Figure 8, where box-plots in (A) summarise abnormality detection performance, (B) summarise tolerance to normal behaviour, (C) summarise tolerance when all agents change behaviour simultaneously, and (D) summarise tolerance and detection of a single agent that changes its behaviour from the normal behaviour to an abnormal behaviour halfway through an experiment. Table 5 lists the details on the experimental setups used.

Across all experiments, we only observed relatively small differences in performance when FV spaces of distinct dimensionalities were used, with a maximum difference in performance (proportion of time that behaviour is correctly classified) of 7.9%. Some of the differences were found to be statistically significant using the non-parametric Kruskal-Wallis significance test ($d.f. = 4$, $p < 0.001$), followed by the Bonferroni multicomparison test: in the experiments summarised in Figure 8(A), FVs with a length of 9 bits yielded a significantly higher performance than FVs of length 6, 12, and 15 bits; in the experiments summarised in Figure 8(B) and 8(C),

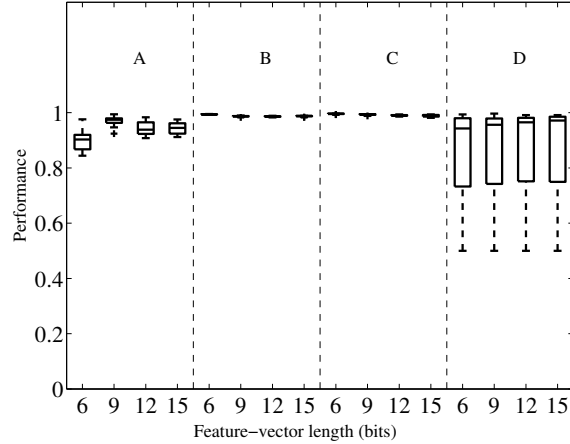


Figure 8: **Scalability of CRM performance in terms of FV classification space.** Abnormality detection performance of the CRM across 20 replicates, with 6, 9, 12 and 15-bit FV length: (A) detection of fault-simulating behaviours, (B) tolerance to normal behaviours, (C) tolerance to transitions in normal behaviour of entire swarm, and (D) response (immune and tolerance) to transitions in single agent behaviour. See Table 5 for details on the experimental setups and quantification of performance.

the FV of length 6 bits yielded a significantly higher performance than all of the longer FVs; while no statistically significant difference was found in performance in the experiments summarised in Figure 8(D). The high variance in performance in detection of a single agent transitioning to an abnormal behaviour (Figure 8(D)), was because of low performance in 5 of the 20 replicates (irrespective of FV size) where the transitioning agent was surrounded by other aggregating agents when it switched from performing aggregation to dispersion, and consecutively could not be detected immediately as behaving abnormal. Overall, the results indicate that a CRM-based abnormality detector is able to improve performance with the addition of new behaviour features (F_7 , F_8 and F_9), upto a point, and that the further addition of redundant features do not compromise its classification accuracy.

The small decrease in performance with the FVs of length 12 and 15 bits (Figure 8(A)), follows from the equation of affinity between T-cell clonal types and APC sub-populations (eq. 15). In order to detect as abnormal an APC sub-

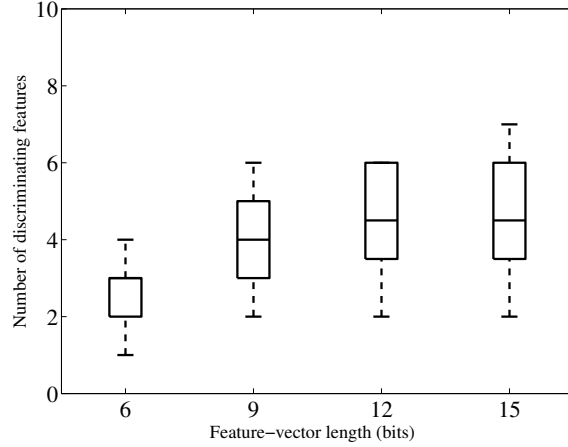


Figure 9: **Number of discriminating features.** The number of fault-simulating/normal behaviour discriminating features in 6, 9, 12 and 15-bit FV length, across 20 replicates. Each box corresponds to the average number of discriminating features for all 16 fault-simulating behaviour detection experiments (experiment details in Table 5, setup (A)).

population presenting the FV of a fault-simulating behaviour, as the FV length l is incremented, more features of the FV are required to be different (longer Hamming distance) between this APC sub-population and the receptors of regulatory cells representing normal behaviour. This number of different features for our CRM parameters is 2, 3, 4 and 5 for FV length of 6, 9, 12 and 15, respectively. An analysis of the number of fault-simulating/normal behaviour discriminating features in FVs of length 6, 9, 12 and 15 bits (Figure 9) shows an increase in number of discriminating features with the addition of behaviour features F_7 , F_8 and F_9 . However, there is no further increase in such features with FVs of length 12 and 15 bits (computational details in Appendix A).

5.4.1. Number of floating-point operations for FVs of lengths 6, 9, 12 and 15 bits

To assess the computational resources necessary to run the CRM with FV spaces of different sizes, we recorded the number of floating-point operations executed by the CRM-based abnormality detector over the course of a set of experiments with 20 agents. Figure 10 shows the number of floating-point operations (FLO)

in experiments involving the transition of normal behaviour from aggregation to dispersion to aggregation. This experiment was selected for illustration as it revealed the maximum total number of FLOs, for the largest FV space (15-bits FVs). The box-plot indicates the number of FLOs, averaged across 20 agents and 15,000 control cycles (corresponding to 1500 s), for 20 independent replicates. Fluctuations in number of FLO necessary in different control cycles go up to five times the median number. The results show that while the size of the FV space increases exponentially with the number of features considered, the computational resources in terms of FLOs required to execute the CRM-based abnormality detection increases in accordance with a power function with an exponent of 0.3 ($R^2 = 0.97$), with decreasing growth-rates of 176.31 ± 20.2 , 27.52 ± 4.7 , and 7.88 ± 1.7 FLOs/FV, for 6 to 9, 9 to 12, and 12 to 15-bit FVs, respectively. Furthermore, a similar relationship between the number of FLOs and the size of the FV space, was revealed in all the other experimental setups.

Our simplified CRM-based abnormality detector is computationally efficient in terms of FV classification space. However, does this efficiency come at a cost to performance, because of the simplifications introduced in the model implementation (Section 3)? To address this question, we compared the performance and required computational resources for the simplified and the complete CRM implementations. Results revealed that while the simplifications had no effect on the performance of the abnormality detector (see Supplementary Section S1.1 and Figures S1, S2 and S3), they improved its computationally efficiency by almost an order of magnitude (see Supplementary Section S1.2 and Figure S4).

5.5. Scalability of CRM in the Swarm Size

A distributed abnormality detection approach is particularly relevant for multiagent systems with a large number of units. We ran a series of experiments to assess the performance of our CRM-based abnormality detection approach in swarms of different sizes. Results of experiments with swarms of 20, 40, 60, 80 and 100 agents are shown in Figure 11.||

The results in Figure 11 show that, aside from the experiments in (A), the CRM-based abnormality approach is unaffected by swarm size. The decrease in the

|| The environment area was increased proportional to the number of agents, keeping the agent density constant in all our experiments.

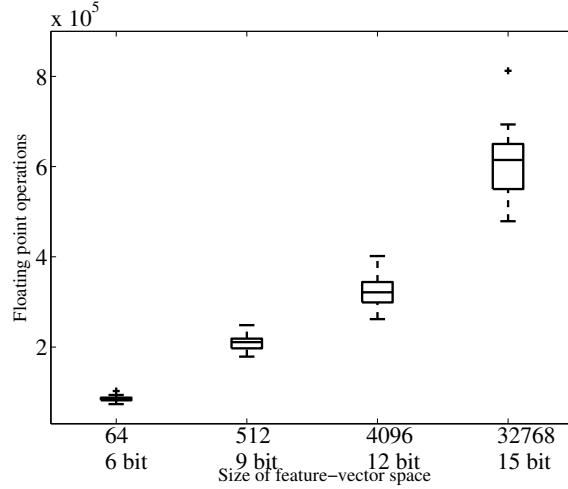


Figure 10: **Scalability of CRM resources in terms of FV classification space.** Computational costs of abnormality detection using the CRM with FV space for 6, 9, 12 and 15-bits FV length (in log scale), across 20 replicates. Each box corresponds to the average number of FLOs per agent, per control cycle in experiments involving the transition of normal behaviour from aggregation to dispersion to aggregation (details of experimental setups in Table 5).

proportion of time a fault-simulating agent is detected as the number of units in the swarm is increased observed (Figure 11(A)) is the result of situations that mainly occur with the combinations of the normal behaviours aggregation and homing, and the fault-simulating behaviour CIRCLE: large moving aggregates are formed of normal behaving agents, thereby increasing the possibility that the stationary circling abnormal agent is intermingled in this aggregate and thus not detected as behaving abnormally. In the experiments with other combinations of normal and fault-simulating behaviour, we only observed stochastic differences in performance and thus no relationship between swarm size and proportion of time that an abnormally behaving agent is detected. Furthermore, the high variance in performance in Figure 11(D) was consequent to low performance in three replicates (swarm of 80 agents), four replicates (swarm of 40 agents), and five replicates (swarm of 20, 60 and 100 agents), where the focal agent transitioning from aggregation to dispersion was surrounded by other aggregating agents, and could not disperse from within the

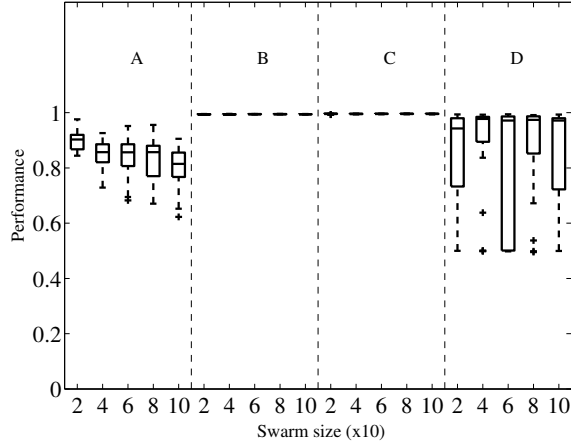


Figure 11: **Scalability of CRM performance in terms of swarm size.** Abnormality detection performance of the CRM across 20 replicates, with 6-bit FV length, and swarm sizes of 20, 40, 60, 80 and 100 agents: (A) detection of fault-simulating behaviours, (B) tolerance to normal behaviours, (C) tolerance to transitions in normal behaviour of entire swarm, and (D) response (immune and tolerance) to transitions in single agent behaviour. See Table 5 for details on the experimental setups and quantification of performance.

aggregate.

5.6. Resilience to stochastic variations in behaviour: A comparison with threshold-based model

We conducted a series of experiments to compare the performance of our CRM-based approach to abnormality detection to a simpler threshold-based model. In a threshold-based model, each agent computes the distribution of perceived FVs, and FVs displayed by less than t agents are considered abnormal. The threshold t is set at the bifurcation point of the CRM (a_R in Figure 2), separating the immune and tolerant responses of the model, and remains fixed throughout each replicate. Furthermore, as each agent performs comparisons of all perceived FVs with the threshold at most n times, where n is the number of perceived neighbouring agents, our threshold-based model implementation takes time $\mathcal{O}(n)$. Details on the number of FLOs required in actual experiments can be found in Supplementary Section S2.2

and Figure S5.

The performance of abnormality detection based on the threshold-based model for FVs of lengths 6, 9, 12, and 15 bits are shown in Figure 12: boxes in (A) summarise fault-detection performance, (B) summarise tolerance to normal behaviour, (C) summarise tolerance when all agents change behaviour simultaneously, and (D) summarise tolerance and detection of a single agent that changes its behaviour from the normal behaviour to an abnormal behaviour halfway through an experiment (see Table 5 for details).

The results show that while the abnormality detection performance improves as more features are considered (Figure 12(A)), the tolerance of normal behaviour (Figure 12(B)) decreases as increasingly longer FVs are considered. The decrease in performance when all the agents change their behaviour simultaneously during the experiment (Figure 12(C)) and when a single agent transitions from the normal behaviour to an abnormal behaviour halfway through the experiment (Figure 12(D)), are due to the decrease in tolerance of normally behaving agents when longer FVs are considered. Both the increase in abnormality detection performance and the decrease in tolerance of normal behaviour is due to the stochastic differences in FVs of agents performing the same behaviour (see the example in Figure 7). The more features considered, the larger the opportunity for stochastic variations in the distribution of the FVs displayed by normally behaving agents. The tolerance to normal behaviour does, in fact, degrade to the point where the median proportion of time that normally behaving agents are tolerated (Figure 12(B) and Figure 12(C)) is respectively 0.64 when no transition in normal behaviour occurs, and 0.59 when all agents transition to a different normal behaviours during experiment. The increase in the proportion of time that an abnormally behaving agent is detected when longer FVs are considered thus comes at the expense of an increasing number of false positives.

The CRM-based approach, on the other hand, achieved a higher median performance than the threshold-based model in all the experiments (all experimental setups in Table 5), ranging from a 3.1% increase in performance in experiments with 6-bit FVs, and up to a 52.8% increase in performance in experiments with 15-bit FVs (detailed analysis in Supplementary Section S2.1 and Table S1). The improvement in performance with the CRM, particularly for large FVs, is because of the resilience the model exhibits to stochastic variations in FVs displayed by normally behaving agents (see Figure 8). This resilience to stochastic variations in the perceived FVs of

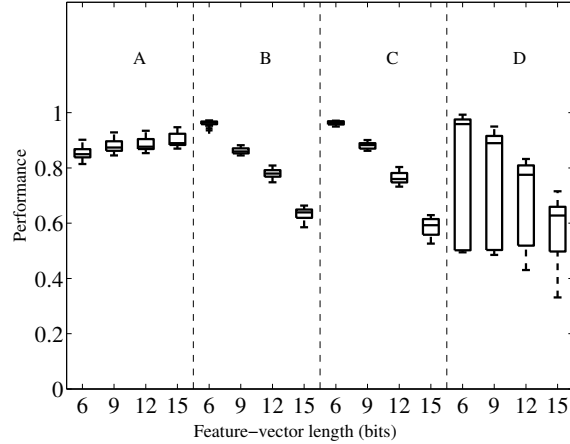


Figure 12: **Performance of threshold-based model.** Fault-detection performance of the threshold-based model across 20 replicates, with 6, 9, 12 and 15-bit FV length: (A) detection of fault-simulating behaviours, (B) tolerance to normal behaviours, (C) tolerance to transitions in normal behaviour of entire swarm, and (D) response (immune and tolerance) to transitions in single agent behaviour. See Table 5 for details on the experimental setups and quantification of performance.

agents executing the same normal behavior, results from the cross-reactivity between T-cells and APCs (c , see eqn. 15). The cross-reactivity causes the regulatory cells to react with and suppress effector cells whose specificities are close to (low Hamming distance apart) the specificities of the regulatory cells. Consequently, the CRM-based approach not only tolerates as normal, the behaviours exhibited by a majority of the swarm, but also the stochastically variant behaviours of a minority of agents with similar FVs to the majority.

6. Discussion and Conclusions

In this study, we investigated the use of the CRM [35, 36, 37] for online abnormality detection in a MRS. In our experiments, all robots executed the same behaviour, except those simulating faults. To detect abnormal behaviour, each robot encoded the behaviour of the ten nearest neighbours as binary feature vectors. The agents behaving normally were therefore always sensed in abundance by their neighbours. Our abnormality detection system was thus not required to look for persistence in

normal behaviour, but only the abundance, to operate successfully. Experimental results revealed that our CRM-based approach, (i) displayed robust maintenance of tolerance to individuals exhibiting a wide variety of normal behaviours, characterised as being exhibited by a large proportion of the agents in the swarm, (ii) was able to reliably detect abnormal behaviours, irrespective of changes in the normal behaviour of the rest of the swarm, (iii) was scalable both in terms of the number of agents in the MRS, and in terms of the number of features used for normal-abnormal behaviour classification, and (iv) was resilient to stochastic variations in observed behaviour, in comparison to the threshold-based model for behaviour classification.

The abnormality detection model developed in this study highlights an important difference between fault detection and abnormality detection. Behaviours that are not similar to behaviours exhibited by a sufficiently large proportion of proximal agents in the swarm may be considered as abnormal, although they may not necessarily be caused by faults. In the detection of specific faulty agents, the false negatives occurred when the behaviour of a fault-simulating agent was indistinguishable from normal behaviour. For example, in some of the flocking/STRLN experiments, the normally behaving agents aggregated around the fault-simulating agent moving in a straight line causing the whole system, including the fault-simulating agent, to move in flocks. Similarly, in dispersion/RNDWK experiments, the dispersion behaviour of the normally behaving agents sufficed to keep the whole swarm dispersed in the environment, even when a fault-simulating agent performed a random walk. However, faults which lead to behaviour that is either indistinguishable from normal behaviour and/or does not disrupt the behaviour of the system are often tolerable.

Current endogenous approaches to fault detection (see [21, 22, 19, 23, 24] for examples) assume that the normal behaviour of the robot is known and can be characterised beforehand. Consequently, such approaches may not easily operate in scenarios involving variations in the characterisation of normality, or wherein the characterisation of normal behaviour is not available beforehand. By contrast, in our CRM-based approach, abnormality detection is determined online, without an initial training step. Furthermore, a priori knowledge of normal and abnormal behaviour classification is not required, as demonstrated in our results, when, for instance, the dispersion behaviour is detected abnormal when exhibited by a single agent, and the STOP behaviour is tolerated as normal when performed by all the agents of the swarm. Consequently, our approach may compliment the existing endogenous

models in scenarios involving temporal variations in normal behaviour (e.g., online learning, perturbations in the environment). In such scenarios, our fault-detection algorithm does not need to be retrained, consequently saving costs and increasing the autonomy of the resultant system.

In exogenous fault detection, Parker *et al.* [25] proposed the ALLIANCE software architecture, wherein the mathematically modelled motivation of different robots to perform different tasks, resulted in a high degree of fault tolerance for cooperating teams of robots. Similarly, in other exogenous fault-detection models [26, 27, 28], an explicit and intricate inter-robot communication process was used to facilitate task allocation. However, these approaches require relatively complex and tightly-coupled robot platforms to operate successfully. By contrast, our proposed fault-detection model is distributed in design and consequently scalable as demonstrated with up to a fivefold increase in swarm size. Furthermore, in contrast to other exogenous and distributed fault detection models, our approach is not limited to the detection of specific faults involving complete robot failure [30], and does not require detailed task-performance metrics known apriori [16, 32].

The experiments presented in this study involved only relatively simple swarm behaviours, and in the experiments with transitions in swarm behaviour, all the robots transitioned simultaneously. In the current version of our approach, the normal/abnormal classification is based exclusively on feature vectors observed in the present, and may therefore not be applicable to scenarios in which robots perform more complex (or composite) behaviours, or in which behaviour transitions propagate gradually. In order to extend CRM-based abnormality detection to scenarios in which robots independently can perform different behaviours at different times, classification must be based on observations made over a period of time. One approach would be for robots to store a recent record of observed feature vectors. The robots would use the record when executing their internal CRM-instance. A record of past observations could also allow our approach to be applied to scenarios in which robots cannot always observe a certain minimum number of neighbouring robots. In our ongoing work, we are evaluating the use of a history of recent observations in CRM-based abnormality detection.

Real sensors and actuators are typically noisy resulting in some degree of stochasticity in observation and action. Such stochasticity does, however, only have an impact on the CRM-based abnormality detector if it can cause changes in one or more of entries of the feature vector. One way to compensate for noisy sensors

and actuators is by taking the characteristics of the specific hardware platform into account in the design of the individual features. In our experiments, features F_3, F_4, F_5 , and F_6 had associated thresholds to compensate for stochastic variations in agent behaviour (see Section 4). If we had subjected the movement of agents to noise in our experiments, those thresholds could, for instance, have been set accordingly. The value of individual features was furthermore calculated based on several observations made over a period of time (W seconds), and the decision on whether to classify an agent as behaving abnormally or not, was based on votes from several agents. There are thus several mechanisms in place (feature design, use of multiple observations, and voting) to avoid that a single noisy observation leads to the misclassification of agents. Still, in case an agent miscalculates the one or more features due to noise, it has previously been shown that CRM-based classification is robust to moderate degrees of perturbations in feature vectors [38].

The scalability of our abnormality detection algorithm in terms of the amount of information used for behaviour classification was evaluated with 6 to 15 different features. Our results revealed that the CRM-based fault-detection algorithm performed well with relatively short feature vectors (i.e., of 6 and 9 features). Further increase in length of the feature vector did not have a strong effect on performance in fault detection. Additionally, the exponential increase in the feature-vector space only lead to a small increase (following a power law) in the number of floating-point operations, due to our computationally efficient implementation of the crossregulation model.

Our approach to the design of the feature vector was to make it generalisable across various behaviour tasks. Indeed, in our selection, the same features were used for eight different normal and abnormal behaviours. However, some level of domain knowledge was required to select the features, that may not always be available to the system designer. These selected features are independent of the controller architecture employed to execute the robot behaviours, evident by the absence of any of the subsumption architecture controller parameters in our feature space. Furthermore, as our model scales well with the size of behaviour classification space, features suspected to be of use for fault detection may be introduced without seriously affecting the computational cost of the algorithm, allowing the potential use of our generalised abnormality detection algorithm to a broader range of multirobot system applications. In ongoing work, we are studying the inclusion of non-binary features (“distance to neighbour” instead of “close to neighbour”, for instance) to

allow individual features to characterise a whole set or a continuum of behavioural aspects.

Acknowledgement: This work was supported by Fundação para a Ciência e a Tecnologia (FCT) project grants PTDC/EEACRO/104658/2008, PEst-OE/EEI/LA0009/2013 and EXPL/EEI-AUT/0329/2013.

- [1] Fukuda T and Nakagawa S 1988 Dynamically reconfigurable robotic system *Proceedings of the IEEE International Conference on Robotics and Automation* (IEEE Press, Piscataway, NJ) pp 1581–1586
- [2] Beni G 1988 The concept of cellular robotic system *Proceedings of the IEEE International Symposium on Intelligent Control* (IEEE Press, Piscataway, NJ) pp 57–62
- [3] Hauert S, Zufferey J and Floreano D 2009 *Autonomous Robots* **26** 21–32
- [4] Waibel M, Keller L and Floreano D 2009 *IEEE Transactions on Evolutionary Computation* **13** 648–660
- [5] Parker C, Zhang H and Kube C 2003 Blind bulldozing: multiple robot nest construction *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE Press, Piscataway, NJ) pp 2010–2015
- [6] Wurman P R, D’Andrea R and Mountz M 2007 Coordinating hundreds of cooperative, autonomous vehicles in warehouses *Proceedings of the National Conference on Innovative Applications of Artificial Intelligence* (AAAI Press) pp 1752–1759
- [7] Werfel J, Petersen K and Nagpal R 2014 *Science* **343** 754–758
- [8] Crespi V, Galstyan A and Lerman K 2008 *Autonomous Robots* **24** 303–313
- [9] Mermoud G, Upadhyay U, Evans W and Martinoli A 2010 Top-down vs bottom-up model-based methodologies for distributed control: a comparative experimental study *Proceedings of the International Symposium on Experimental Robotics* (Springer-Verlag, Berlin, Germany)
- [10] Parker L 2000 Current state of the art in distributed autonomous mobile robotics *Proceedings of the International Symposium on Distributed Autonomous Robotic Systems* (Springer, Berlin, Germany) pp 3–12
- [11] Yamins D and Nagpal R 2008 Automated global-to-local programming in 1-D spatial multi-agent systems *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems* (IFAAMAS, Red Hook, NY) pp 615–622
- [12] Hamann H 2010 *Space-Time Continuous Models of Swarm Robotic Systems: Supporting Global-to-Local Programming* Cognitive Systems Monographs (Springer-Verlag, Berlin, Germany)
- [13] Payton D, Estkowski R and Howard M 2003 *Robotics and Autonomous Systems* **44** 229–240
- [14] Şahin E 2005 Swarm robotics: From sources of inspiration to domains of application *Swarm Robotics* (Springer, Berlin, Germany) pp 10–20
- [15] O’Grady R, Groß R, Christensen A L and Dorigo M 2010 *Autonomous Robots* **28**(4) 439–455
- [16] Lau H, Bate I, Cairns P and Timmis J 2011 *Robotics and Autonomous Systems* **59** 1021–1035
- [17] Timmis J, Andrews P and Hart E 2010 *Swarm Intelligence* **4** 247–273
- [18] Tarapore D, Christensen A L, Lima P U and Carneiro J 2013 Abnormality detection in

- multiagent systems inspired by the adaptive immune system *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems* (IFAAMAS, Red Hook, NY) pp 23–30
- [19] Terra M H and Tinos R 2001 *Journal of Robotic Systems* **18** 357–374
- [20] Bjercknes J D and Winfield A F T 2013 On fault tolerance and scalability of swarm robotic systems *Distributed Autonomous Robotic Systems* (Springer Berlin Heidelberg) pp 431–444
- [21] Christensen A L, O’Grady R, Birattari M and Dorigo M 2008 *Autonomous Robots* **24** 49–67
- [22] Skoundrianos E N and Tzafestas S G 2004 *IEEE Robotics and Automation Magazine* **11** 83–90
- [23] Verma V and Simmons R 2006 *Robotics and Autonomous Systems* **54** 184–191
- [24] Goel P, Dedeoglu G, Roumeliotis S I and Sukhatme G S 2000 Fault detection and identification in a mobile robot using multiple model estimation and neural network *Proceedings of the IEEE International Conference on Robotics and Automation* (IEEE Press, Piscataway, NJ) pp 2302–2309
- [25] Parker L 1998 *IEEE Transactions on Robotics and Automation* **14** 220–240
- [26] Gerkey B P and Matarić M J 2002 *IEEE Transactions on Robotics and Automation* **18** 758–768
- [27] Gerkey B P and Matarić M J 2002 Pusher-watcher: An approach to fault-tolerant tightly-coupled robot coordination *Proceedings of the IEEE International Conference on Robotics and Automation* (IEEE Press, Piscataway, NJ) pp 464–469
- [28] Bernardine Dias M, Zinck M, Zlot R and Stentz A 2004 Robust multirobot coordination in dynamic environments *Proceedings of the IEEE International Conference on Robotics and Automation* vol 4 (IEEE Press, Piscataway, NJ) pp 3435–3442
- [29] Christensen A L, O’Grady R, Birattari M and Dorigo M 2007 Exogenous fault detection in a collective robotic task *Proceedings of the European Conference on Artificial Life* (Springer, Berlin, Germany) pp 555–564
- [30] Christensen A L, O’Grady R and Dorigo M 2009 *IEEE Transactions on Evolutionary Computation* **13** 1–12
- [31] Winfield A F T and Nembrini J 2006 *International Journal of Modelling, Identification and Control* **1** 30–37
- [32] Lau H, Timmis J and Bate I 2011 Collective self-detection scheme for adaptive error detection in a foraging swarm of robots *Artificial Immune Systems* (Springer Berlin Heidelberg) pp 254–267
- [33] Janeway C, Travers P, Walport M and Shlomchik M 1997 *Immunobiology: The Immune System in Health and Disease* (New York: Garland Science)
- [34] Sakaguchi S 2004 *Annual Review of Immunology* **22** 531–562
- [35] Carneiro J, Leon K, Caramalho I, Van Den Dool C, Gardner R, Oliveira V, Bergman M, Sepúlveda N, Paixão T, Faro J and Demengeot J 2007 *Immunological Reviews* **216** 48–68
- [36] Leon K, Perez P, Lage A, Farob J and Carneiro J 2000 *Journal of Theoretical Biology* **207** 231–254
- [37] Leon K, Lage A and Carneiro J 2003 *Journal of Theoretical Biology* **225** 107–126
- [38] Tarapore D, Christensen A L, Lima P U and Carneiro J 2012 Environment classification in multiagent systems inspired by the adaptive immune system *Proceedings of the International Conference on the Simulation and Synthesis of Living Systems* (MIT Press, Cambridge) pp

275–282

- [39] Alberts B, Bray D, Lewis J, Raff M, Roberts K and Watson J D 1994 *Molecular biology of the cell* (New York: Garland Science)
- [40] Tarapore D, Christensen A L, Lima P U and Carneiro J 2012 Clonal expansion without self-replicating entities *Proceedings of the International Conference on Artificial Immune Systems* (Springer, Berlin, Germany) pp 191–204
- [41] Butcher J 2003 *Numerical methods for ordinary differential equations* (John Wiley & Sons, West Sussex, England) chap 23 2nd ed
- [42] Evans M, Hastings N and Peacock B 2000 *Statistical Distributions* (John Wiley & Sons, New York, NY) chap 27, pp 134–136 3rd ed
- [43] De Boer R J and Perelson A S 1995 *Journal of Theoretical Biology* **175** 567–576
- [44] Gutiérrez A, Campo A, Dorigo M, Amor D, Magdalena L and Monasterio-Huelin F 2008 *Sensors* **8** 7545–7563
- [45] Brooks R 1986 *IEEE Journal of Robotics and Automation* **2** 14–23

Appendix A. Behaviour discriminating features

The section details the computation of the number of behaviour discriminating features, in FVs of length 6, 9, 12 and 15 bits. In each of the 16 fault-simulating behaviour detection experiments (see Table 5, experiment setup (A)), we compute the number of control cycles when the value of each of l features of the fault-simulating agent differs from the majority of the rest of the swarm. A feature in the FV is counted discriminative if it differentiates between the normal and fault-simulating behaviours in most of the 15,000 control cycles of the experiment.