DOI: http://dx.doi.org/10.7551/978-0-262-33027-5-ch072

Abnormality Detection in Robots Exhibiting Composite Swarm Behaviours

Danesh Tarapore[1], Anders Lyhne Christensen[2,3] and Jon Timmis[1]

[1]York Robotics Laboratory, Department of Electronics, University of York, York, UK
[2]Bio-inspired Computation and Intelligent Machines Lab, Lisbon, Portugal
[3]Instituto de Telecomunicações, Instituto Universitário de Lisboa (ISCTE-IUL), Lisbon, Portugal
danesh.tarapore@york.ac.uk

Abstract

Fault detection is one of the most prominent challenges in the field of multirobot systems (MRS). Most existing fault-tolerant systems prescribe a characterisation of normal behaviours (fault-free behaviours), and train a model to recognise them. Behaviours not recognised by the model are labelled abnormal. MRS employing these models do not transition well to scenarios involving gradual changes in normal behaviour. In such scenarios, existing fault-detection systems may not be applicable, or may incur potentially costly false positive detections. We propose to address this challenging problem by taking inspiration from the regulation of tolerance and (auto)immunity in the adaptive immune system. We deploy an immune system-based fault-detection approach to detect abnormalities in *heterogeneously* behaving robots. Results of extensive simulation-based experiments demonstrate that a distributed MRS can correctly tolerate delayed propagation of different normal behaviours in the collective, at low false-positive rates. Furthermore, the fault-detection system is able to reliably detect robots performing different fault-simulating behaviours.

Introduction

The field of multirobot systems (MRS) has progressed rapidly in recent years, with groups of robots performing increasingly complex behaviours, ranging from self-assembly (Christensen et al., 2008a), to warehouse-management (Wurman et al., 2007). Despite the availability of several low-cost robot platforms (IFR, 2014), and coordination algorithms for task allocation and division of labour (e.g., Berman et al. (2009)), platform reliability and endurance still inhibit the wide-spread usage of MRS outside of the laboratory (Dunbabin and Marques, 2012). The individual robots of a MRS are susceptible to failures, prominently resulting from electro-mechanical faults in the robot's sensor and actuation devices, and bugs in the software controlling the robot (Winfield and Nembrini, 2006). Consequent to the wide variety of intricate inter-robot interactions affecting robot behaviour, the prediction, detection and/or diagnosis of potential faults for an individual robot represent major challenges. While the large number of individual robots in self-organised robot collectives may produce a robust and resilient system, even in such systems, robots that exhibit partial failure have the potential to disrupt the entire collective (Bjerknes and Winfield, 2013). Explicit fault detection is therefore crucial to enhance the autonomy and operating capacity of MRS.

In most engineered fault-detection systems, robots are trained (e.g., using supervised learning) to detect anomalies in their own sensory and actuator data (e.g., Christensen et al. (2008b)). While these approaches can provide robust fault detection when trained on the normal behaviour (no faults present) of the target system, they do not transcend well to changes in this behaviour. Other fault-detection systems, specifically designed for MRS, allow the robots to detect faults in each other (e.g., Lau et al. (2011)). However, many of these approaches are centralised and/or rely on detailed knowledge of the MRS tasks. Furthermore, the more task-generic and decentralised versions of such approaches are limited to detecting robots experiencing complete failure (details in the Related Work section).

An interesting analogy can be made between task-generic fault-detection systems and the adaptive immune system in vertebrates (e.g., Timmis et al. (2010); Tarapore et al. (2012)). The immune system acts to help defend and repair the body's cells and tissues in response to pathogenic insults, has the need to differentiate between what might be normal, that is, normally functioning cells and tissues, and abnormalities such as invading pathogen (Janeway et al., 1997). The characteristics of these abnormalities are in principle open-ended, and therefore differ from the limited set of faults the current approaches to robot fault-detection are designed to detect. 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., Sakaguchi (2004)). The decentralised nature of these intercellular interactions imparts a high degree of robustness within the system, and a flexibility to respond to a very broad range of possible pathogenic attacks. Such properties provide rich inspiration for the engineering of robust, fault-tolerant systems.

In previous work, we developed a generic fault-detection

Danesh Tarapore, Anders Lyhne Christensen, Jon Timmis (2015) Abnormality Detection in Robots Exhibiting Composite Swarm Behaviours. Proceedings of the European Conference on Artificial Life 2015, pp. 406-413

approach, based on the adaptive immune system, for exogenous fault detection in large-scale MRS (Tarapore et al., 2015). An agent-based simulator was used to model scenarios where individual robots have to tolerate certain behaviours, while mounting an immune response against others. The salient feature of the developed fault-detection system was that the characterisation of behaviour (normal or faulty) were not prescribed apriori, but rather learned online based on their abundance in the MRS. Behaviours exhibited by many of the robots in the MRS were considered normal. By contrast, rare behaviours exhibited by a single or few robots were detected as abnormal behaviours that could be caused by a fault. The resultant fault-detection system was successfully utilised in MRS comprising of *homogeneously* behaving robots. However, in many MRS, the individual robots of the collective often exhibit distinct behaviours at any given time. For example, even in simple foraging scenarios, the different behaviours exhibited may include, searching for resources, signalling the presence of new resources, and returning them to the nest. The scenarios are often more complex, and include composite exploration and exploitation like behaviours. In such tasks, existing adaptive fault-detection systems may result in a large number of false-positive classifications of normal behaviours.

In this paper, we propose a solution to the above issue, specifically: extending our generic exogenous fault-detection system to operate successfully on *heterogeneously* behaving robots of an MRS. In our extended model, a history of past behaviour observations is taken into account in order to classify robot behaviour, not only as normal/abnormal, but also as *suspicious*. Using a history of past T-cell populations embodied on the robot, observed behaviours misclassified as abnormal in our original model are now simply treated as suspicious, if the behaviour was considered normal in the past. We demonstrate the capacity of the system to tolerate normal behaviours, despite temporal variations – ranging from an almost simultaneous to slow changes in the behaviour across the MRS. Furthermore, our extended fault-detection system continues to reliably detect abnormally behaving robots.

The rest of the paper is organised as follows: in the following section, we describe the different approaches to fault detection in autonomous robots, followed by our fault-detection system based on the adaptive immune system. We then present extensions to our fault-detection system for heterogeneously behaving robots. We go on to report the results of our experiments in different scenarios, and under varying behaviour transition rates. Finally, we discuss our approach to fault detection and highlight the conclusions of this study.

Related work

The engineering of fault-detection systems for robots is a well-studied problem, and can be broadly classified into *endogenous* and *exogenous* approaches. The endogenous fault detection approaches have robots proprioceptively detecting faults in their individual behaviour (e.g., Christensen et al. (2008b); Skoundrianos and Tzafestas (2004)). The models assume that hardware faults affect the flow of sensory information, and actuation of the robot. Consequently, statistical learning algorithms (e.g., artificial neural networks) are trained to detect anomalies based on the input-output relationship of focal components on the robot. These approaches have been successfully used to detect faults in components such as, wheels (Skoundrianos and Tzafestas, 2004), and tracks with wheels (Christensen et al., 2008b) of a mobile robot. Most endogenous fault-detection models are built on the assumption that the normal operating behaviour of the robot is known, and can be characterised predeployment. Consequently, the models are trained to recognise prescribed normal behaviour, and behaviours not recognised by the model are labelled abnormal. However, while such approaches have resulted in reliable fault detection in specific scenarios, they may not easily transition to different and varying characterisations of normality in MRS (e.g., online adaptation of existing behaviours or even learning of new behaviours). In summary, endogenous fault-detection systems are finely tuned to the particular behaviour of the target system, under a specific set of task parameters.

Exogenous fault detection systems leverage the multitude of robots constituting a MRS, to provide individual robots the capability to detect faults in one another (e.g., Parker (1998); Christensen et al. (2009); Millard et al. (2014); Lau et al. (2011)). Such an approach is particularly advantageous to detect faults that are difficult to detect endogenously by the robot (e.g., mechanical failures consequent to an unstable connection to a power source), or that disable the robots communication and capability to alert other robots or a human operator. However, while exogenous approaches do provide some interesting results of robust fault detection and tolerance (e.g., see ALLIANCE software architecture Parker (1998)), successful fault detection require prior knowledge of the various tasks to be performed, and their corresponding measures of performance. Furthermore, many of these fault-detection approaches (e.g., Gerkey and Matarić (2002); Parker (1998)) are designed for MRS consisting of a limited number of tightly coupled, and relatively complex robots.

Adaptive immune system based fault-detection

Our previously developed fault-detection system for homogeneously behaving robots is based on the *crossregulation model* (CRM) (Leon et al., 2000), 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. According to Leon et al. (2000), the immune 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

Danesh Tarapore, Anders Lyhne Christensen, Jon Timmis (2015) Abnormality Detection in Robots Exhibiting Composite Swarm Behaviours. Proceedings of the European Conference on Artificial Life 2015, pp. 406-413

pathogens, that are characterised as being neither persistent nor abundant.

The CRM describes the population dynamics of cells of the adaptive immune system, consisting of three mutually interacting cell types: (a) Antigen presenting cells (APCs) that present the antigen on their surface. Individual APCs have a fixed number of binding sites on which effector and regulatory cells can form conjugates; (b) effector cells $\mathtt{T_E}$ that can potentially mount immune responses which, depending on receptor specificity, may be directed to foreign pathogens or to body-antigens; and (c) regulatory cells $\mathtt{T_R}$ that suppress proliferation of $\mathtt{T_E}$ cells with similar specificities. Furthermore, 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 clones according to their specificity.

A mathematical formulation comprising ordinary differential equations, of the dynamics of interactions between effector cells and regulatory cells, with APCs, is detailed in Tarapore et al. (2015). In the next subsection, we provide an overview of these interactions, introduce the important parameters, and highlight the interesting properties of the CRM (detailed description of model at Carneiro et al. (2007)). We then describe the implementation of the CRM in the MRS, and the resultant fault detection achieved by the model.

Functioning of the CRM

The CRM provides a differential equation governing each of the clonal types (i) of effector (E_i), and regulatory (R_i) T-cells. The sub-populations of each of these clonal types is subject to the following: (a) growth by proliferation (division of parent cells to two daughter cells) of their individual activated cells; and (b) shrinkage consequent to death of T-cells.

The density of proliferating T-cells of each clonal type i, is dependent on their interactions with APCs of each sub-populations j. Consider the interactions between the i-th T-cell clone and the j-th APC population. The resulting conjugates C_{ij} are subject to the following: (a) Formation of new conjugates by the free T-cells of clone i with available binding sites on APCs of sub-population j. This conjugation rate is also controlled by the affinity between the T-cells clone i and APCs sub-population j; and (b) Dissociation of existing conjugated T-cells from APCs. The density of activated effector and regulatory cells is computed from the quasi-steady state densities of the conjugates. The conjugated effector cells proliferate in the absence of regulatory cells on the same APC. In contrast, conjugated regulatory cells can only proliferate if at least one effector cell is simultaneously conjugated to the same APC.

Table 1: Parameters of the CRM implementation.

Param.	Description	Value
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 a.u.
I_R	Density of new regulatory cells introduced at each simulation time-step	10 a.u.
k	Feature vectors to APCs scaling factor	0.002
S	Length of time CRM instance is numerically integrated, in a single robot control cycle	5×10^7 a.u.
d	Proportion of T-cells diffused to neighbouring robots	0.5
s_j	Suspicion value associated with feature vector FV_j	–
Δ_s	Increment to suspicion value for newly observed feature vector	0.95
γ	Threshold below which feature-vector is interpreted as suspicious and not abnormal	0.95

Execution of the CRM on a robot

Our original CRM-based model implemented on a distributed embodied MRS, in simulation, affords the system the capacity to detect abnormally behaving robots, whilst maintaining a tolerance towards normal robot behaviour (see Tarapore et al. (2015)). Within the CRM framework, behaviours exhibited by an abundant proportion of the robots (normal behaviour) are interpreted as body antigens (so normal). By contrast, faulty or abnormal behaviours are considered foreign antigens (abnormal). Each robot executes an independent instance of the CRM.

The CRM-based fault-detection model was tested for four normal swarm behaviours (aggregation, flocking, dispersion, and homing) and four fault-simulating behaviours. The fault-simulating behaviours performed by one of the 20 robots (selected at random) was, (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 time-step (RNDWK); (c) circle with diameter 1 unit around a fixed point (CIRCLE); or (d) stop completely (STOP). These additional behaviors were introduced to mimic: (a) software bugs and sensor faults in the robot controller (STRLN and RNDWK); (b) motor malfunctions (CIRCLE); and (c) a broken or dead battery (STOP).

In both the original, and extended CRM-based models, a robot computes a 6 bit binary feature vector (concatenation of 6 simple Boolean features) encoding its behaviour (Table 2), at the start of each control cycle. The robot then senses the feature vectors of its 10 nearest neighbours (tested with up to 100 simulated robots in MRS, see Tarapore et al. (2015)), and counts the number of robots assigned to each of the 2^6 feature vectors (FV_j, $j \in \{1 \dots 2^6\}$). In an individ-

Table 2: Boolean features encoding robot behaviour (parameters in Table 3).

Notation	Value at time τ*
$F_1(\tau)$	$\frac{\sum_{t=\tau}^{\tau-W} U[n_i(t)]}{W} > 0.5$
$F_2(\tau)$	$\frac{\sum_{t=\tau}^{\tau-W} U[n_o(t)]}{W} > 0.5$
$F_3(\tau)$	$p(\tau) > 0.05W\lvert\vec{v}_{\max}\rvert$
$F_4(\tau)$	$\lvert\vec{v}(\tau)\rvert > 0.05\lvert\vec{v}_{\max}\rvert$
$F_5(\tau)$	$\sum_{t=\tau}^{\tau-W} U[n_i(\tau) + n_o(\tau)] \wedge U[\omega'(\tau) - 0.03\omega'_{\max}] > 0$
$F_6(\tau)$	$\sum_{t=\tau}^{\tau-W} \neg U[n_i(\tau) + n_o(\tau)] \wedge U[\omega'(\tau) - 0.03\omega'_{\max}] > 0$

*The Boolean feature is set if the condition is satisfied, else 0. Function $U[x]$ is 1 if $x > 0$, and 0 otherwise.

ual robot's internal CRM instance, APCs are then generated corresponding to each of the feature vectors perceived. Each APC presents an individual feature vector to the T-cells. The number of each type of the APCs generated $A_j = kFV_j$, for $j \in \{1, \ldots, M\}$, where k is a scaling constant, and M is the number of different feature vectors perceived by the robot. The T-cell clones $(T_1, T_2, \ldots, T_N)$, each have a different receptor encoded as a binary string, which determines their affinity to the APC population. The affinity between T-cell clonal i and APC population j is denoted by θ_{ij}:

$$\theta_{ij} = \exp\left(-\frac{H[i,j]}{cl}\right) \tag{1}$$

where H is the Hamming distance between the receptor of T_i and the feature vector presented by A_j, l is the length of the presented feature vector, and c is the cross-reactivity between T-cells and APCs.

At the start of the simulation, the number of effector and regulator cells on each robot is initialised to I_E and I_R, respectively. Following this, Algorithm 1 (parameters in Table 1) is performed by every robot in each control cycle, allowing the robots to execute their internal CRM. The robots begin by sensing their neighbours, and then compute 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 robots. In this communication phase, each robot selects at random (linear distribution weighted on total T-cell density in the CRM instance) another robot, from one of its 10 nearest neighbours. Following the selection, each robot sends and receives d of its effector and regulatory cells. Finally, the robot decides the nature of each feature vector FV_j sensed by first computing the following quantities:

$$E = \sum_{i=1}^{N} \theta_{ij} E_i \qquad R = \sum_{i=1}^{N} \theta_{ij} R_i \tag{2}$$

and tolerating the feature vector if $R > E$. By contrast, if $E > R$, the feature vector is classified as faulty by the robot.

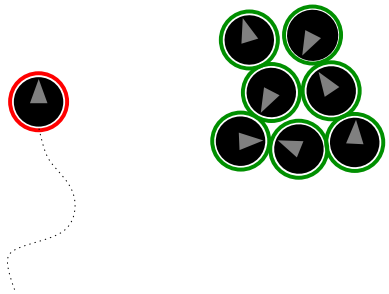

Figure 1: Example of a robot (left) unable to join an aggregate (right) due to faulty sensors. This robot is detected as behaving abnormally (coloured red) by its neighbours, whereas the rest of the swarm is behaving normally and tolerated (coloured green).

The CRM deployed in the MRS is passive and does not alter the behaviour of the robots. Rather, the individual robots merely report the outcome of the classification for the different behaviors observed in their vicinity, at each simulation time step. At the end of each time step, a robot's behaviour is considered normal, if a simple majority of the robot's 10 nearest neighbours tolerate it. Similarly, the behaviour is treated abnormal, if a simple majority of these neighbours interpret it as faulty.

Extending fault-detection to heterogeneously behaving robots

The CRM-based abnormality detection system classifies behaviours solely based on their abundance in the MRS. Behaviours exhibited by many, or the majority of the robots of the MRS are considered as normal. By contrast, rare behaviours exhibited by one of few robots are considered as abnormal, and assumed to be resulting from a fault on the robot. While such an approach is capable of robustly tolerating normal behaviours, and reliably detecting faults in homogeneously behaving robots, it is not designed to detect faults in robots executing complex (or composite) behaviours, or in which behaviour transitions propagate gradually across the MRS. In such scenarios, normal behaviour exhibited by a minority of robots of the MRS trigger false positive incidents.

In order to extend our CRM-based model to scenarios in which robots independently perform different behaviours at different times, behaviour classification must be based on observations made over a period of time. In accounting for past observations, we may ask if a behaviour detected as abnormal in the current time step, has always been abnormal? Considering the behaviour was also abnormal in the previous contexts that it was observed (context is the behaviours of the rest of the MRS), than it may indeed be abnormal. However, if the behaviour was treated as normal in the past, or if a new behaviour has just emerged in the

Danesh Tarapore, Anders Lyhne Christensen, Jon Timmis (2015) Abnormality Detection in Robots Exhibiting Composite Swarm Behaviours. Proceedings of the European Conference on Artificial Life 2015, pp. 406-413

MRS and was therefore not encountered in the past, we may not want to classify it as abnormal (and take appropriate action), but merely treat such behaviour as *suspicious*. This suspicion associated with an observed behaviour quantifies the proportion of past contexts, with respect to which the presently observed behaviour would have been considered as abnormal.

Algorithm 1 A robot's control loop (simulation of a CRM instance).

1: Compute distribution of feature vectors (FV_j) of neighbouring robots
2: Assign feature vectors to APCs i.e., $\forall j$, $A_j = kFV_j$
3: $\forall j \in \{1, 2 \dots M\}$, if $A_j > 0$, increment E_j and R_j by I_E and I_R, respectively
4: **while** $time \leq S$ **do**
5: $\quad \forall i \in \{1, 2 \dots N\}$ 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.
6: $\quad$ Using the number of conjugated cells, compute the updated number of effector and regulatory cells with the Euler-Heun adaptive step method (Butcher, 2003).
7: $\quad$ Increment $time$
8: **end while**
9: Randomly select one of the robots in the communication range following a linear distribution and weighted by the total number of cells on the respective neighbouring robots
10: Exchange cells with robot
11: For each feature vector, compute the sum of effector and regulatory cells, weighted by their affinity.
12: Tolerate the feature vector if total regulatory cells exceeds effectors, else interpret it as faulty.

The suspicion is computed by the robot, when its instance of the CRM classifies an observed behaviour as abnormal. Algorithm 2 details the procedure to compute the suspicion (s_j) associated with a feature vector FV_j, already classified as abnormal. It involves an increment to the suspicion value for every simulation time-step t over the recorded history D, when FV_j would have been considered abnormal (parameters in Table 1).

$$E^t = \sum_{i=1}^{N} \theta_{ij} E_i^t \qquad R^t = \sum_{i=1}^{N} \theta_{ij} R_i^t \tag{3}$$

where E_i^t and R_i^t are the recorded density of effector and regulatory cells at simulation time-step t.

When $E^t > R^t$, the FV_j is considered abnormal with respect to the context at simulation time-step t, and the suspicion s_j associated with FV_j is incremented by 1.

The appearance of a new behaviour, i.e., a behaviour not observed by the robot at time-step t, also results in an increment to s_j, but by a smaller value of Δ_s ($\Delta_s < 1$).

Finally, the feature vector FV_j is considered as *abnormal* if its normalised suspicion value (s_j/D) exceeds a threshold γ. Otherwise, FV_j is merely considered as *suspicious*.

Latency in fault detection: The minimisation of the time required to detect a behaviour as abnormal since it began

Algorithm 2 Subroutine to determine suspicion, called when feature vector FV_j is interpreted as abnormal.

1: {Initialise suspicion value associated to FV_j}
2: $s_j \leftarrow 0$
3: {Iterate over past D time-steps}
4: **for** $t = current$ to $current - D$ **do**
5: $\quad$ {If FV_j was not observed at time t, increment s_j by a lower value}
6: $\quad$ **if** FV_j not observed at time t **then**
7: $\quad\quad s_j \leftarrow s_j + \Delta_s$
8: $\quad$ **else**
9: $\quad\quad$ {Analyse FV_j with T-cell population of time t}
10: $\quad\quad E(t) \leftarrow \sum_{i=1}^{N} \theta_{ij} E_i(t)$
11: $\quad\quad R(t) \leftarrow \sum_{i=1}^{N} \theta_{ij} R_i(t)$
12: $\quad\quad$ {If FV_j would have been interpreted as abnormal at time t, increment s_j by a higher value}
13: $\quad\quad$ **if** $E(t) > R(t)$ **then**
14: $\quad\quad\quad s_j \leftarrow s_j + 1$
15: $\quad\quad$ **end if**
16: $\quad$ **end if**
17: **end for**
18: {Compare normalised suspicion value to threshold}
19: **if** $s_j/D < \gamma$ **then**
20: $\quad FV_j$ is *suspicious*
21: **else**
22: $\quad FV_j$ is *abnormal*
23: **end if**

to be executed is an important requirement of any fault-detection system. At the offset of a new behaviour, our suspicion towards the behaviour is incremented by a small value (Δ_s), since we may not want to immediately classify it as abnormal. Obviously, the suspicion parameter Δ_s influences detection latency, and can be analysed. For a behaviour to be detected as abnormal in our extended model, the following condition has to be satisfied:

$$\frac{y}{D} + \Delta_s \left(\frac{D - y}{D} \right) \geq \gamma \tag{4}$$

where y is the amount of time in the past that the behaviour would have been classified as abnormal (see eq. 3).

Solving eq. 4 for y with parameters in Table 1, the latency is at least 295 s, 128.3 s, and 45 s, for Δ_s at 0.90, 0.94, and 0.95, respectively (including time window W to formulate the feature vector). The latency may be longer if the discriminatory features in the feature vector are not present in time window W.

Experiments

Experimental setup: We conducted a series of experiments to assess the performance of our abnormality detection approach. In all experiments, we simulated a swarm of 20 e-puck like robots located in a 5×5 m^2 toroidal environment. Each robot had a diameter of 7.5 cm and a maximum speed

Danesh Tarapore, Anders Lyhne Christensen, Jon Timmis (2015) Abnormality Detection in Robots Exhibiting Composite Swarm Behaviours. Proceedings of the European Conference on Artificial Life 2015, pp. 406-413

Table 3: Parameters of simulated robot

Param.	Description	Value
$\lvert\vec{v}_{\max}\rvert$	Maximum linear speed of robot	10 cm/s
$\lvert\vec{v}\rvert$	Linear speed of robot	–
$\omega_{\max}$	Maximum change in direction of robot per control cycle	π radians
ω	Change in direction of robot per control cycle	–
n_i	Number of neighbouring robots in the inner range of $[0, 30]$ cm	–
n_o	Number of neighbouring robots in the outer range of $(30, 60]$ cm	–
W	Length of the time window for feature computation	45 s
p	Distance traversed by the robot in the past W s	–

of 10 cm/s (see Table 3).[1] At the start of each experiment, the robots were placed at a random location and assigned a random orientation drawn from uniform distributions.

Latency in fault detection: In a first set of experiments, we assessed the capacity of our CRM-based detector to correctly detect abnormally behaving robots. All robots initially performed Dispersion in which they tried to maximise the distance to all neighbours. Halfway through each experiment, one of the robots in the swarm simulated a fault by switching to the STOP behaviour, which caused the robot to remain immobile. We conducted 20 replicates of each of three experimental setups with different values of the suspicious increment parameter, Δ_s: 0.90, 0.94, and 0.95. Each experimental replicate had a duration of $1,500$ seconds of simulated time. In all 20 out of 20 replicates, for all values of Δ_s, we observed that the fault-simulating robot was correctly detected as behaving abnormally by other members of the swarm. However, the latency, that is the time between a fault-simulating robot begins to execute STOP behaviour, and until it is detected, varied significantly across the three setups (Mann-Whitney test, $df = 18$, all $p < 0.001$, see Fig. 2). For $\Delta_s = 0.90$, the median fault detection latency was 288.2 ± 23.6 s (Median $\pm$ IQR), for $\Delta_s = 0.94$ the median latency was 108.8 ± 20.7 s, while for $\Delta_s = 0.95$ the median latency was 33.2 ± 21 s. As expected (see eq. 4), with suspicion parameter Δ_s at 0.95, the MRS achieved the lowest latency in detecting simulated faults.

Using Δ_s at 0.95, we also tried the following normal/abnormal behaviour combinations in the same setup: (i) Aggregation/Dispersion; (ii) Flocking/RNDWK; and (iii) Homing/Dispersion (see Table 4). In the three behaviour combinations, the abnormally behaving robot was detected in less than 3.5 minutes (median). However, for the Aggregation/Dispersion and Homing/Dispersion behaviours, the abnormal robot exhibiting Dispersion was not detected in

[1]Simulation source code can be downloaded from `https://github.com/daneshtarapore/robotsim_ecal2015.git`

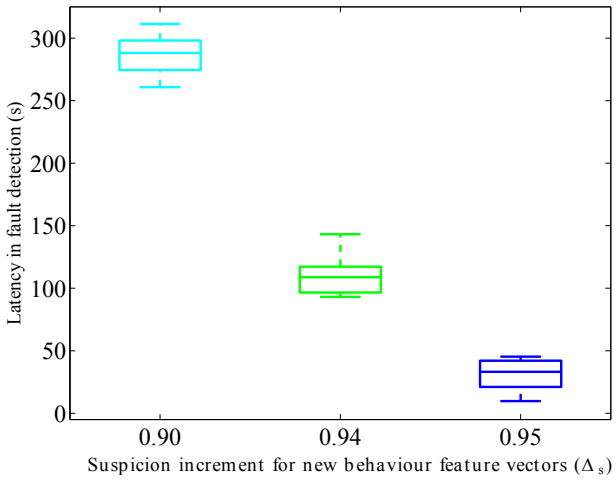

Figure 2: **Latency in the detection of fault simulating behaviour.** Time required to detect an robot performing fault simulating behaviour across 20 replicates for different suspicion increment parameters. The detected robot switches its behaviour from dispersion (normal behaviour of the MRS) to stop halfway through the simulation.

8 of 20, and 7 of 20 replicates, respectively. In these replicates, the abnormal robot was trapped by other aggregating or homing robots when it switched to performing Dispersion, and therefore could not be differentiated from these robots.

False-positive incidents for heterogeneously behaving robots: The minimisation of false positives is important in fault-detection systems, as subsequent fault accommodation may be costly and could lead to the exclusion of capable robots. In a series of experiments, we assessed the system's capacity to avoid false positives when transitions in behaviour occur over time. We conducted 20 replicates in which all robots switched behaviour. In each replicate, the robots started out by performing Dispersion, then gradually switched to the STOP behaviour over a period of time, and then reverted back to Dispersion over a period of time. Each experimental replicate had a duration of $1,500$ seconds. During the first 500 seconds, all the robots of the MRS performed Dispersion. Then, robots selected at random (following a uniform distribution), began switching to the STOP behaviour. The time between robots switching behaviour was 3.75 s, 6.25 s, 12.5 s, and 25.0 s (in four separate and independent experimental setups). After all robot had switched behaviour, robots (selected at random) began to switch back to Dispersion with the same delay used in the initial switch. It should be noted that while in the previous set of experiments (Fig. 2), the STOP behaviour was used to simulate a fault, in this new set of experiments, the STOP behaviour should no longer be considered abnormal since all the robots in the swarm eventually transition to this

Danesh Tarapore, Anders Lyhne Christensen, Jon Timmis (2015) Abnormality Detection in Robots Exhibiting Composite Swarm Behaviours. Proceedings of the European Conference on Artificial Life 2015, pp. 406-413

Table 4: Number of false-positive incidents per robot, and the latency, for different behaviours propagated in the MRS. Suspicion parameter Δ_s is 0.95.

Behaviours	**Number of false-positives incidents at behaviour switching time**				**Latency (s)**
	3.75 s	6.25 s	12.5 s	25 s	
Dispersion, STOP	8.1 ± 12.6	14.8 ± 22.6	38.0 ± 30.7	83.0 ± 22.4	33.2 ± 21.0
Aggregation, Dispersion	2.3 ± 7.0	3.5 ± 9.7	5.1 ± 13.1	21.5 ± 25.9	49.9 ± 28.0†
Flocking, RNDWK	9.1 ± 15.7	17.8 ± 17.6	19.9 ± 22.1	18.8 ± 27.6	194.7 ± 219.4
Homing, Dispersion	0.4 ± 5.5	0.2 ± 7.2	7.7 ± 9.3	4.7 ± 10.6	88.2 ± 67.7‡

†In Aggregation/Dispersion, the dispersing robot was not detected as abnormal in 8 of 20 replicates.

‡In Homing/Dispersion, the dispersing robot was not detected as abnormal in 7 of 20 replicates.

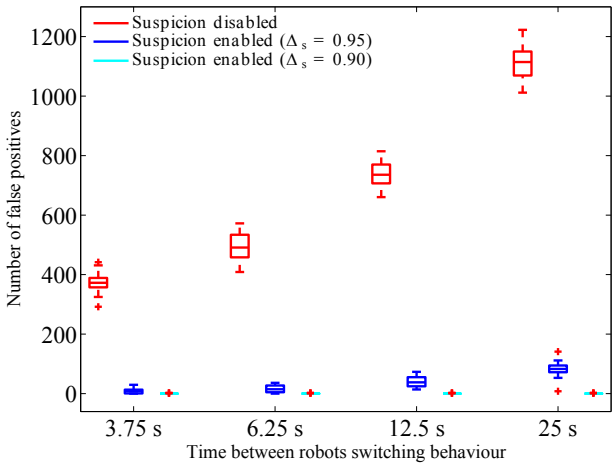

Figure 3: **False-positive incidents of fault-detection for heterogeneously behaving robots.** Number of false-positive incidents per robot across 20 replicates for four different behaviour switching times (for Dispersion–STOP–Dispersion), in each of the following three fault-detection experiments: (i) suspicion disabled (our original fault-detection model); (ii) suspicion enabled, and incremented by 0.90 ($\Delta_s = 0.90$) for new behaviour feature vectors; and (iii) suspicion enabled, and incremented by 0.95 ($\Delta_s = 0.95$) for new behaviour feature vectors.

behaviour. The results, in terms of number of false-positive incidents, for the original (suspicion disabled) and the extended CRM-based fault detector (suspicion enabled), and for different behaviour transition periods are shown in Fig. 3.

Our extended CRM-based model achieved at least an order of magnitude improvement in the number of false-positive incidents over the original model, for all four behaviour switching times. The difference in performance was higher for the more gradual transitions in behaviour. For the extended CRM-based model, at $\Delta_s = 0.90$, the number of false-positive incidents incurred by the MRS was not affected by the behaviour switching delay. However, for $\Delta_s = 0.95$, the behaviour switching delay significantly affected the number of false-positive incidents (Kruskal-Wallis test: $p < 0.001$). Experiments where the MRS was subjected to a higher behaviour switching delay between robots of 25 s incurred more false-positive classifications than MRS experiencing behaviour-switching delays of 3.75 s, 6.25 s and 12.5 s (Mann-Whitney test, $df = 18$, all $p < 0.001$), but with the difference not exceeding 75 incidents in a total of 15,000 control cycles.

In our extended CRM-based model, a trade-off exists between latency and the number of false positives, regulated by the suspicion parameter Δ_s. For Δ_s at 0.90, the model registered a low number of false positives, but a high latency in detecting fault-simulating robots (Fig. 2 and 3). Incrementing Δ_s to 0.95 resulted in a much lower latency, and only a slight increase in the number of false positives. Using this suspicion parameter at 0.95, we also tried the following behaviour propagations across the MRS, each replicated 20 times: (i) Aggregation–Dispersion–Aggregation; (ii) Flocking–RNDWK–Flocking; and (iii) Homing–Dispersion–Homing (see Table 4). In all three behaviour combinations, the number of false-positive incidents per robot was no higher than 22 (median across 20 replicates) in 15,000 control cycles, and irrespective of the tested behaviour switching times.

Conclusions and future work

The results demonstrate the suitability of our generic fault-detection system for MRS exhibiting composite swam behaviours. In our approach, the individual robots of the MRS utilise a record of past behaviour observations to make more accurate normal/abnormal classifications of the behaviours of neighbouring robots. While our approach relies on an immune system-based model to operate in scenarios involving a gradual transition in MRS behaviour, other more traditional methods, such as statistical hypothesis testing could also be applied to detect changes in distribution of past sampled observations (Ladi et al., 2014). In future work, we plan to investigate how these more centralised classification approaches could be integrated into our distributed fault-detection system.

Noise evident in real sensor and actuator readings only has an impact on our fault-detection system if it can cause changes in one or more features of the feature vector. One approach to compensate for perturbations in feature vectors is to account for the specific hardware platform in the design of individual features. Additionally, the value of individual features is calculated based on several observations made over a period of time (W s), and the decision on whether to classify a faulty robot is based on votes from several robots.

Danesh Tarapore, Anders Lyhne Christensen, Jon Timmis (2015) Abnormality Detection in Robots Exhibiting Composite Swarm Behaviours. Proceedings of the European Conference on Artificial Life 2015, pp. 406-413

There are thus several mechanisms in place to avoid misclassification, when noise is added to our simulation as part of future work.

Our fault-detection system is designed to be generic with respect to the behaviours of the underlying MRS. Behaviours are classified (normal/abnormal) solely based on their persistence and abundance in the MRS. However, a number of behaviours may be known in advance to be abnormal, or normal, in particular contexts. "Vaccinating" our immune system-based model with this information, may expedite the process of fault detection and improve the performance of the suspicion module. The long-term goal is to assess our fault-detection system in more complex behaviours for realistic task scenarios.

Acknowledgements

D.T. is supported by a Marie Curie Intra-European Fellowship (Project: GiFteD-MrS, EC Grant No. 623620). A.C. is supported by Fundação para a Ciência e a Tecnologia (FCT) under the grants EXPL/EEI-AUT/0329/2013 and UID/EEA/50008/2013. J.T. is supported by The Royal Society and The Royal Academy of Engineering.

References

Berman, S., Halasz, A., Hsieh, M. A., and Kumar, V. (2009). Optimized stochastic policies for task allocation in swarms of robots. *Robotics, IEEE Transactions on*, 25(4):927–937.

Bjerknes, J. D. and Winfield, A. F. T. (2013). On fault tolerance and scalability of swarm robotic systems. In *Distributed Autonomous Robotic Systems*, pages 431–444. Springer, Berlin Heidelberg, Germany.

Butcher, J. (2003). *Numerical methods for ordinary differential equations*, chapter 23. John Wiley & Sons, West Sussex, England, second edition.

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). When three is not a crowd: a crossregulation model of the dynamics and repertoire selection of regulatory $CD4^+$ T cells. *Immunological Reviews*, 216(1):48–68.

Christensen, A., O'Grady, R., and Dorigo, M. (2008a). SWARMORPH-script: a language for arbitrary morphology generation in self-assembling robots. *Swarm Intelligence*, 2(2-4):143–165.

Christensen, A. L., O'Grady, R., Birattari, M., and Dorigo, M. (2008b). Fault detection in autonomous robots based on fault injection and learning. *Autonomous Robots*, 24(1):49–67.

Christensen, A. L., O'Grady, R., and Dorigo, M. (2009). From fireflies to fault tolerant swarms of robots. *IEEE Transactions on Evolutionary Computation*, 13(4):754–766.

Dunbabin, M. and Marques, L. (2012). Robots for environmental monitoring: Significant advancements and applications. *IEEE Robotics & Automation Magazine*, 19(1):24–39.

Gerkey, B. P. and Matarić, M. J. (2002). Sold!: Auction methods for multirobot coordination. *IEEE Transactions on Robotics and Automation*, 18(5):758–768.

IFR (2014). Industrial robot statistics. `http://www.ifr.org/industrial-robots/statistics`.

Janeway, C., Travers, P., Walport, M., and Shlomchik, M. (1997). *Immunobiology: The Immune System in Health and Disease*. Garland Science, New York, NY.

Ladi, A., Timmis, J., Tyrrell, A. M., and Hickey, P. J. (2014). Statistical hypothesis testing for chemical detection in changing environments. In *2014 IEEE Symposium on Computational Intelligence in Dynamic and Uncertain Environments (CIDUE)*, pages 77–84. IEEE Press, Piscataway, NJ.

Lau, H., Bate, I., Cairns, P., and Timmis, J. (2011). Adaptive data-driven error detection in swarm robotics with statistical classifiers. *Robotics and Autonomous Systems*, 59(12):1021–1035.

Leon, K., Perez, P., Lage, A., Farob, J., and Carneiro, J. (2000). Modelling T-cell-mediated suppression dependent on interactions in multicellular conjugates. *Journal of Theoretical Biology*, 207(2):231–254.

Millard, A., Timmis, J., and Winfield, A. (2014). Run-time detection of faults in autonomous mobile robots based on the comparison of simulated and real robot behaviour. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)*, pages 3720–3725. IEEE Press, Piscataway, NJ.

Parker, L. (1998). Alliance: An architecture for fault tolerant multirobot cooperation. *IEEE Transactions on Robotics and Automation*, 14(2):220–240.

Sakaguchi, S. (2004). Naturally arising $CD4^+$ regulatory T cells for immunologic self-tolerance and negative control of immune responses. *Annual Review of Immunology*, 22:531–562.

Skoundrianos, E. N. and Tzafestas, S. G. (2004). Finding fault-fault diagnosis on the wheels of a mobile robot using local model neural networks. *IEEE Robotics & Automation Magazine*, 11(3):83–90.

Tarapore, D., Christensen, A. L., Lima, P. U., and Carneiro, J. (2012). Clonal expansion without self-replicating entities. In *Proceedings of the International Conference on Artificial Immune Systems*, pages 191–204. Springer, Berlin, Germany.

Tarapore, D., Lima, P., Carneiro, J., and Christensen, A. L. (2015). To err is robotic, to tolerate immunological: fault detection in multirobot systems. *Bioinspiration & Biomemetics*, 10(1):016014.

Timmis, J., Andrews, P., and Hart, E. (2010). On artificial immune systems and swarm intelligence. *Swarm Intelligence*, 4(4):247–273.

Winfield, A. F. and Nembrini, J. (2006). Safety in numbers: fault-tolerance in robot swarms. *International Journal of Modelling, Identification and Control*, 1(1):30–37.

Wurman, P. R., D'Andrea, R., and Mountz, M. (2007). Coordinating hundreds of cooperative, autonomous vehicles in warehouses. In *Proceedings of the National Conference on Innovative Applications of Artificial Intelligence*, pages 1752–1759. AAAI Press.

Danesh Tarapore, Anders Lyhne Christensen, Jon Timmis (2015) Abnormality Detection in Robots Exhibiting Composite Swarm Behaviours. Proceedings of the European Conference on Artificial Life 2015, pp. 406-413