Abnormality Detection in Multiagent Systems Inspired by the Adaptive Immune System

Danesh Tarapore[a,b], Anders Lyhne Christensen[c], Pedro U. Lima[a], Jorge Carneiro[b]
[a]Institute for Systems and Robotics (ISR), Instituto Superior Técnico (IST), Lisbon, Portugal
[b]Instituto Gulbenkian de Ciência, Oeiras, Portugal
[c]Instituto de Telecomunicacões, Instituto Universitário de Lisboa (ISCTE-IUL), Lisbon, Portugal
daneshtarapore@gmail.com, alcen@iscte.pt, pal@isr.ist.utl.pt, jcarneir@igc.gulbenkian.pt

ABSTRACT

Fault tolerance is one of the most prominent challenges in the field of multirobot systems. The efficient and long term operation of a robot collective requires an accurate detection and accommodation of abnormally behaving robots. Most of the existing fault tolerant systems prescribe a characterization of normal behavior, and train a model to recognize them. Behaviors not recognized by the model are labelled abnormal. However, these models require a priori knowledge of the normal behavior. Furthermore, multirobot systems employing these models do not transition well to scenarios involving temporal changes to normal behavior. We propose to address this challenging problem by taking inspiration from the regulation of tolerance and (auto)immunity in the adaptive immune system. We adopt the Crossregulation model, used to explain the robust immunological maintenance of tolerance, and deploy it within a multiagent system. Results of extensive simulation-based experiments demonstrate that a distributed multiagent system can detect abnormalities under varying conditions of normal behaviors. The collective dynamics gives rise to a meaningful normal-abnormal classification of the behavior by individual agents, even if these categories were not prescribed a priori in the agents.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Algorithms, Experimentation

Keywords

Fault detection; Crossregulation model; decentralized control; multirobot systems; swarm robotics

1. INTRODUCTION

Multiagent systems (MAS) comprise a large number of research domains, ranging from software agents to multirobot systems (MRS), and play an important role in several applications, such as supply chain management [8] and transportation logistics [3]. Individual agents of a MAS are vulnerable to failures. In a MAS, an agent's behavior depends not only on interactions with their immediate environment but also on the behavior of other agents [9]. Consequent to the wide variety of intricate inter-agent interactions affecting agent behavior, the prediction and modeling of potential faults to an individual agent is a major challenge. In addition, the faults plaguing an individual may not only be consequent to bugs in the software controlling the agent, but more deliberative in nature and a result of adversarial agents [13].

MAS has expanded rapidly into the domain of physical agents since the 1980s, with groups of robots coordinating to perform a wide variety of tasks ranging from exploration [11], to warehouse-management [16]. Individual robots of a MRS, like their virtual counterparts, are susceptible to failure. In contrast to the virtual agents, faults in robots may also be electro-mechanical in nature, and manifest themselves in the robot's sensor and actuation devices. Additionally, the simple and small sized robots typically used in large scale collectives, do not have the hardware capabilities to detect some of the common faults (e.g., rotary encoders to detect actuator faults). Considerable work has focused on engineering fault detection in robot collectives (e.g., see endogenous [6], and exogenous [7] fault detectors). The large majority of these models are built on the assumption that the normal operating behavior is known, and can be characterized beforehand. Consequently, the models are trained to recognize prescribed normal behavior (e.g., see [15]), and behaviors not recognized by the model are labelled abnormal. However, while this approach does provide some interesting results of robust fault detection and tolerance, the fault detection in the collective may not easily transition to different and varying characterization of normality. This may be particularly relevant in scenarios wherein agents change their behavior through learning, or in response to perturbations in their environment. In addition, the prior information on the characterization of normal operating behavior may not always be available to the system designer. In summary, the fault detection capabilities of the MRS tend to be designed rather specifically to the particular behavior demonstrated by the target system.

An interesting analogy can be made between fault tolerant systems and the adaptive immune system. The immune system too has to allow the body's cells and tissues to function normally, while mounting an immune response or attack-

Appears in: *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013), Ito, Jonker, Gini, and Shehory (eds.), May, 6–10, 2013, Saint Paul, Minnesota, USA.*

ing what may be abnormal cells or tissues (e.g. unrelated grafts, infected cells, cancerous cells). The characteristics of these abnormalities are in principle open-ended. 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., [14]). The decentralized nature of these interactions imparts a high degree of robustness for tolerance and fault detection, without the need of a genetically hardwired record of what normal tissues should look like, making the immune system an appealing model to designers of fault detection systems.

The Crossregulation model (CRM) [5], 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 characterized as being neither persistent nor abundant. The model has been used successfully in classification tasks involving spam detection (e.g., [1]). Additionally, recent work using the CRM for a decentralized environment classification in a MAS [10] revealed that the agents successfully mounted distinct responses to different environment objects based on their persistence and abundance, even if the object categories were not prescribed a priori to the agents. The successful deployment of the model in these studies, led us to propose the use of the CRM for a more generic fault detection system.

In this study, we use an agent-based simulator to model a situation where individuals have to tolerate certain behaviors, while mounting an immune response against others. In these simulations, normal agents perform a typical swarm behavior (aggregation, flocking, dispersion or homing). Furthermore, on one of the agents, a predefined abnormal behavior is introduced (e.g., agent circling to mimic motor malfunction). The different agent behaviors, and their characterization (normal or abnormal) are not known by the agents beforehand. We demonstrate the capacity of the system to tolerate normal swarm behaviors that may be characterized as persistent and abundant, while mounting an immune response against abnormal behaving agents. In addition, the system response is resilient to temporal variations in normal behavior.

The rest of the paper is organized as follows: In the following section, we describe the CRM. We then present the application of the CRM in a MAS (Section 3). We go on to report the results of our experiments for different swarm behaviors and under different introduced faults (Section 4). Finally, we discuss our approach to fault detection (Section 5) and highlight the conclusions of this study (Section 6).

2. THE CROSSREGULATION MODEL

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 conjugation sites (s) 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 subpopulations of equivalent APCs, with each APC in a subpopulation presenting the same antigen on its surface. Similarly $\mathtt{T_E}$ and $\mathtt{T_R}$ cells are also classified into different clones according to their specificity.

A mathematical formulation using ordinary differential equations, of the dynamics of interactions between $\mathtt{T_E}$ and $\mathtt{T_R}$ cells, with APCs, is detailed in [10]. Below, we describe these interactions, introduce the important parameters, and highlight the interesting properties of the CRM that are later incorporated in a MAS (detailed description of model [5]).

2.1 Functioning of the model

The CRM implemented on each agent of the MAS is as follows. The CRM provides a system of differential equations governing the density of each of the clonal types (i) of $\mathtt{T_E}$ (E_i), and $\mathtt{T_R}$ (R_i) T-cells. The subpopulations of each of these clonal types is subject to the following: (a) growth by proliferation (division of parent cells into two daughter cells) of their individual activated cells; and (b) shrinkage consequent to death of T-cells (see Table 1 for proliferation and death rates of $\mathtt{T_E}$ and $\mathtt{T_R}$ cells). In this study, there is no continual influx rate of new $\mathtt{T_E}$ and $\mathtt{T_R}$ cells. 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 $\mathtt{E_i}$ and $\mathtt{R_i}$ cells of each clonal type i, is dependent on their interactions with APCs A_j of each subpopulations j. For example, let us consider the interactions between the i-th T-cell clone and the j-th APC population. The resulting conjugates C_{ij} is subject to the following: (a) Formation of new conjugates by the free T-cells of clone i with available conjugation sites on APCs of subpopulation j. This conjugation rate is also controlled by the affinity between the T-cells clone i and APCs subpopulation j; and (b) Dissociation of existing conjugated T-cells from APCs (see Table 1 for the conjugation and deconjugation rates, and the affinity between T-cells and APCs). The conjugation and deconjugation of T-cells from APCs are fast processes with respect to the overall T-cell clone dynamics. Consequently, we solve at each time step, the steady state values of the conjugates utilizing the Euler-Heun adaptive step method [4]. Finally, the density of activated $\mathtt{T_E}$ and $\mathtt{T_R}$ cells is computed from the quasi-steady state densities of the conjugates. The conjugated $\mathtt{T_E}$ cells are activated in the absence of $\mathtt{T_R}$ cells on the same APC. In contrast, conjugated $\mathtt{T_R}$ cells can only be activated if at least one $\mathtt{T_E}$ cell is simultaneously conjugated to the same APC.

2.2 Behavior of T-cell population

Considerable work has focused on analyzing the CRM, and the underlying dynamics of $\mathtt{T_E}$, $\mathtt{T_R}$ and APC populations [12]. An important property of the CRM is the ability to distinguish between antigens based on their density. At low concentrations of APCs, the system always reaches a globally stable state consisting solely of $\mathtt{T_E}$ cells (immune response). By contrast, at higher concentrations of APCs, the system exhibits bistable behavior, i.e., the system can evolve either into an equilibrium state composed predominantly of $\mathtt{T_E}$ cells (immune response), or into a state consisting mostly of $\mathtt{T_R}$ cells (tolerant response). The system evolves into the $\mathtt{T_R}$ cell dominated state, provided that the seeding population has sufficient $\mathtt{T_R}$ cells. The seed T-cell population density (E_0 and R_0) is chosen to ensure this. However, if $\mathtt{T_R}$ cells

Table 1: Parameters of the crossregulation model.*

Param.	Description	Value (a.u.)
A_j	Density of APCs of population j	–
s	Maximum number of T-cells that can bind to an APC	3
E_0	Seed density of $\mathtt{T_E}$ cells	10
R_0	Seed density of $\mathtt{T_R}$ cells	10
E_i	Density of $\mathtt{T_E}$ cells of clone i	–
R_i	Density of $\mathtt{T_R}$ 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 $\mathtt{T_E}$ cells	10^{-3}
π_R	Proliferation rate of $\mathtt{T_R}$ cells	0.7×10^{-3}
δ	Death rate of $\mathtt{T_E}$ and $\mathtt{T_R}$ cells	10^{-6}

*In this study the continual influx rate of new $\mathtt{T_E}$ and new $\mathtt{T_R}$ cells (σ_E and σ_R respectively, from [10]) is 0.

Table 2: Parameters of an agent

Param.	Description	Value (a.u.)
v_m	Maximum linear speed of agent	0.1
v	Linear speed of agent	–
ω	Change in direction of agent per time-step	–
n_i	Number of neighboring agents in inner range $(0, 3]$ units	–
n_o	Number of neighboring agents in outer range $(3, 6]$ units	–
W	Length of time window for feature computation	450
p	Distance traversed by agent in past W time-steps	–

are initially underrepresented, $\mathtt{T_E}$ cells will competitively exclude the former from the system. Consequent to the APC density dependent response, $\mathtt{T_R}$ cells suppress the immune response of $\mathtt{T_E}$ cells, and thereby ensure tolerance to antigens that are persistent and abundant. Additionally, $\mathtt{T_E}$ cells are free to mount immune responses to antigens that are not persistent or not abundant.

3. CRM IN A MULTIAGENT SYSTEM

In this section, we demonstrate how the CRM can be implemented on a distributed embodied MAS in order to give the system the capacity to detect abnormally behaving agents, while maintaining a tolerance towards normal swarm behavior. Behaviors that are persistent and abundant (performed by most of agents) are to be tolerated. By contrast, rare behaviors (exhibited by fewer agents) are to be detected as abnormal. We show that the MAS is able to detect abnormally behaving agents, and adapt online and tolerate different normal behaviors.

We use a stochastic, spatial, discrete-time MAS simulator. The simulated environment is toroidal and has a size of 50×50 units, and composed of 20 mobile point-sized agents. Each agent has a maximum speed of 0.1 units/time-step and performs the behavior assigned to it at the start of the simulation. During the simulation, each agent senses the behavior of its 10 nearest neighbors, and runs an internal and individual instance of a CRM in order to determine if the perceived behaviors should be tolerated or not.

Swarm behaviors: The normal swarm behaviors simulated are (a) dispersion, (b) aggregation, (c) flocking, and (d) homing towards a moving landmark. The behaviors are implemented using a subsumption architecture [2], wherein an agent with no neighbors performs a random walk. Additionally, in dispersion, agents move in the opposite direction of center of mass of neighboring agents. In aggregation, agents move towards the center of mass of surrounding agents, but disperse away if too close to their neighbors. Similarly, homing agents move towards a single selected agent that serves as a moving landmark, and move away if too close to the landmark or other agents. In flocking, agents continually adjust their velocity to that of neighboring agents. Furthermore, flocking agents, aggregate towards and disperse from neighbors, if they are too far away or close by, respectively (simulation source code can be downloaded from `www.isr.ist.utl.pt/~dtarapore/AAMAS2013/simulator`).

Faulty behaviors: 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 time-step (RNDWK); (c) circle with diameter 1 unit around a fixed point (CIRCLE); or (d) stop completely (STOP). These additional behaviors are introduced to mimic: (a) software bugs and sensor faults in the agent controller (STRLN and RNDWK); (b) motor malfunctions (CIRCLE); and (c) a broken or dead battery (STOP).

Binary encoding of agent behavior: The agents detect behavioral features within their sensory range (6 units), and use the perceived information to encode their individual behavior. Individual features of an agent's behavior are encoded in Boolean form (present= 1, absent= 0), and then concatenated to form a binary string, the *feature-vector*. In our simulations, a feature-vector comprises 6 features ($\mathtt{F_1}, \mathtt{F_2} \ldots \mathtt{F_6}$) and is based on behavioral information over a moving time window of W time-steps. Consequently, the computation of the feature-vector is initiated after time-step W of the simulation.

The first two features, $F_1(\tau)$ and $F_2(\tau)$ at time-step τ, pertain to the agent's sensory input, i.e., neighbors in proximity:

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

$$F_2(\tau) = 1 \text{ if } \frac{\sum_{t=\tau}^{\tau-W} U[n_o(t)]}{W} > 0.5 \text{ else } F_2(\tau) = 0 \quad (2)$$

where n_i and n_o are the number of neighbors in the inner ($(0, 3]$ units), and outer ($(3, 6]$ units) range, respectively. Furthermore, $U[n]$ is the unit step function, defined as:

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

At time-step τ, the features $F_1(\tau)$ and $F_2(\tau)$ are set, if the agent has at least one neighbor in range $(0, 3]$ units and

$(3, 6]$ units, respectively, for the majority of the past W time-steps (see parameters in Table 2).

The next two features, $F_3(\tau)$ and $F_4(\tau)$, pertain to the agent's sensory-motor interactions. For these interactions, an agents motor response is characterized as follows:

$$O[\omega(\tau)] = \begin{cases} 1, & \text{if } |\omega(\tau) - \omega(\tau - 1)| > 0.1 \text{ radians} \\ 0, & \text{otherwise} \end{cases}$$

where, a motor response is registered ($O[\omega(\tau)]$), if the agent's angular acceleration exceeds 0.1 radians (3% of maximum angular acceleration).

We further define two sensor-motor interaction events $\mathtt{S_m}$ and $\mathtt{S_n}$:

$$S_m(\tau) = U[n_i(\tau) + n_o(\tau)] \wedge O[\omega(\tau)]$$
$$S_n(\tau) = \neg U[n_i(\tau) + n_o(\tau)] \wedge O[\omega(\tau)]$$

The above sensor-motor interaction event $S_m(\tau)$ is set, if the agent's motors respond in the presence of sensory input (one or more neighbors in range). Similarly, $S_n(\tau)$ is set if the agent's motors respond in the absence of sensory input (no neighbors in range).

Consequently, for the the features $F_3(\tau)$ and $F_4(\tau)$,

$$F_3(\tau) = 1 \text{ if } \sum_{t=\tau}^{\tau-W} S_m(t) > 0 \text{ else } F_3(\tau) = 0 \quad (3)$$

$$F_4(\tau) = 1 \text{ if } \sum_{t=\tau}^{\tau-W} S_n(t) > 0 \text{ else } F_4(\tau) = 0 \quad (4)$$

where, the features $F_3(\tau)$ and $F_4(\tau)$ are set if the sensor-motor interaction events $\mathtt{S_m}$ and $\mathtt{S_n}$ respectively, occur at least once in time window W.

Finally, for the last two features, $F_5(\tau)$ and $F_6(\tau)$, pertaining to the agent's motors, we have:

$$F_5(\tau) = 1 \text{ if } p(\tau) > 0.05 W v_m \text{ else } F_5(\tau) = 0 \quad (5)$$

$$F_6(\tau) = 1 \text{ if } v(\tau) > 0.05 v_m \text{ else } F_6(\tau) = 0 \quad (6)$$

where, at time-step τ, $p(\tau)$ is the distance traversed by the agent in the past W time-steps, speed $v(\tau)$, and maximum speed v_m. The feature $F_5(\tau)$ is set, if $p(\tau)$, exceeds 5% of the maximum distance that may be traversed by the agent in W time-steps. Similarly, $F_6(\tau)$ is set, if $v(\tau)$ exceeds 5% of the maximum agent speed.

Immunological response to agent behavior: At the start of each time-step, an agent computes the 6 bit feature-vector encoding its behavior (eqs 1-6). The agent then senses the feature-vectors of its 10 nearest neighbors, and computes the number of agents assigned to each feature-vector (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 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 agent.

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}:

Table 3: Parameters of the stochastic simulator

Param.	Description	Value (a.u.)
N	Number of T-cell clones	64
M	Number of different feature-vectors	64
c	Cross-reactivity between T-cells and APCs	0.15
I_E	Density of new $\mathtt{T_E}$ cells introduced at each simulation time-step	10
I_R	Density of new $\mathtt{T_R}$ 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 neighboring agents	0.5

$$\theta_{ij} = \exp\left(-\frac{H[i,j]}{c}\right) \quad (7)$$

where H is the Hamming distance between the receptor of T_i and the feature-vector presented by A_j, 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 populations. By contrast, at low c, each T-cell clone would have a high affinity to only one distinct APC population.

At the start of the simulation, the number of $\mathtt{T_E}$ and $\mathtt{T_R}$ cells on each agent is initialized to E_0 and R_0, respectively. Following this, Algorithm 1 (parameters in Table 3) is performed by the agents in each simulation time-step, allowing the agents to execute their internal CRM. The agents begin by sensing their neighbors, 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 $\mathtt{T_E}$ and $\mathtt{T_R}$ cells at time S, the cells diffuse among agents. In this communication phase, each agent selects at random another agent from its 10 nearest neighbors. Following the selection, each agent sends and receives d of its $\mathtt{T_E}$ and $\mathtt{T_R}$ cells. Finally, the agent decides the nature of each feature-vector FV_j sensed by first computing the following quantities:

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

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

4. EXPERIMENTS

We set up a series of experiments to evaluate the behavior classification capabilities of a MAS operating according to the model described above. Within the CRM conceptual framework, the behaviors exhibited by an abundant proportion of the agents (normal behavior) are interpreted as normal body antigens. By contrast, faulty or abnormal behaviors are considered extraneous antigens.

The CRM deployed in the MAS is passive and does not alter agent behavior. Rather, individual agents 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, an agent's behavior is considered

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

1: {Perceive neighboring agents}
2: Compute distribution of feature-vectors (FV_j) of neighboring 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 $\forall j \in \{1, 2 \dots M\}$, compute the number of conjugated cells C_{ij} in quasi-steady state, integrating using the Euler-Heun adaptive step method
10: Using the number of conjugated cells, compute the updated number of $\mathtt{T_E}$ and $\mathtt{T_R}$ cells with the Euler-Heun adaptive step method. The adaptive step size is stored in h
11: $time \leftarrow time + h$
12: **end while**
13: {Diffuse T-cells across neighboring agents}
14: Randomly select one of the agents in the communication range following a linear distribution and weighted by the total number of T-cells on the respective neighboring agents
15: Exchange T-cells with agent
16: {Decide if feature-vectors are to be tolerated or not}
17: For each feature-vector, compute the sum of $\mathtt{T_E}$ and $\mathtt{T_R}$ cells, weighted by their affinity.
18: Tolerate the FV if total $\mathtt{T_R}$ cells exceeds $\mathtt{T_E}$ cells, else interpret it as faulty. Log the outcome of the classification.

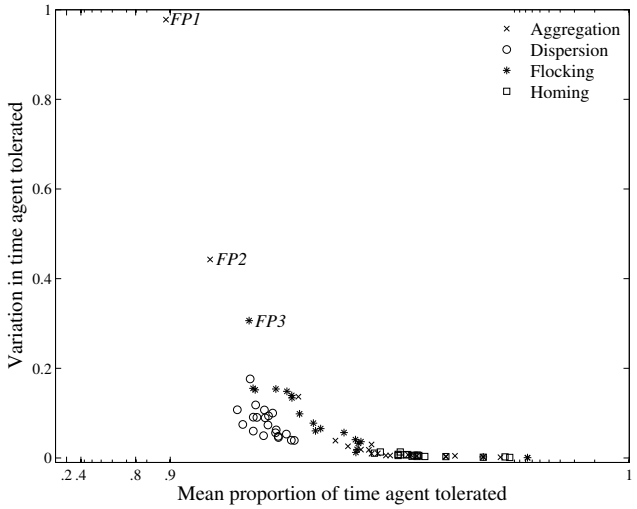

Figure 1: Mean and variation in proportion of time agent tolerated, across the 20 agents of the MAS, in each of 20 replicates of 4 swarm behaviors. Variation measured in each replicate as the absolute difference between the maximum and minimum time tolerated, of the 20 agents.

normal, if a simple majority of the agent's 10 nearest neighbors tolerate its feature-vector. Similarly, the behavior is treated abnormal, if a simple majority of these neighbors interpret its feature-vector as faulty.

4.1 Tolerance to normal behavior

In a first set of four distinct experiments, all 20 agents of the MAS performed the same swarm behavior i.e. aggregation, dispersion, flocking or homing. At the start of each replication of these experiments, the agents were placed at random positions. We recorded the proportion of time each agent in the swarm was tolerated by its neighbors. We conducted a total of 20 replications with each swarm behavior (see Fig. 1). In the figure, each point summarizes a single replicate of the experiment for a particular swarm behavior. A point's location is determined by the mean proportion of the time each agent in the experiment was tolerated (x-coordinate) and variation of the time tolerated within the swarm calculated as the absolute different between the maximum and minimum time tolerated (y-coordinate).

Results of the first set of experiments indicate that the swarm largely is able to tolerate the collective behavior with a mean proportion of time that agents are tolerated above 0.97 in all replications except for one (point labeled *FP1* (false positive) in Fig. 1). In replication *FP1*, the agents executed the aggregation behavior and a particular situation arose in which all the agents except two had formed a large aggregate. The remaining two agents had formed a small stable aggregate at a different location in the environment. The agents in the large aggregate classified the behavior of the two agents in the small aggregate as abnormal due to the differences in the number of neighbors between the agents in the large aggregate and the agent in the small aggregate. As such, the collective correctly classified the behavior of the two agents in the small aggregate as being abnormal with respect to the rest of the collective, but the abnormal behavior is not a result of faulty agents in this case. A similar situation occurred in the experiment labeled *FP2* in Fig. 1: three large aggregates formed (five or more agents in each aggregate) and a single aggregate of only two agents.

In the experiment labeled *FP3*, agents executed the flocking behavior, but one agent took longer than the others to join a flock (the particular agent random walked for 1900 time-step before it encountered the flock, while all other agents were flocking by 500 time steps).

4.2 Detection of faulty behavior

We set up a second series of experiments to evaluate the capabilities of a MAS to detect an agent behaving abnormally due to a fault. In each of these experiments, 19 of the 20 agents performed the same swarm behavior (one of aggregation, dispersion, flocking and homing), and the remaining agent carried out a fault-simulating behavior (one of STRLN, RNDWK, CIRCLE and STOP). Furthermore, the identity of this abnormally behaving agent was unknown to the MAS. All combinations of normal and abnormal behaviors were simulated, in 16 (4 normal behaviors $\times$ 4 abnormal behaviors) distinct experiments. We have summarized the results of the experiments in Fig. 2. The results show that the median proportion of the time that the abnormally behaving agent was detected is above 0.50 in 14 out of the 16 experimental setups. The differences between the results obtained in experimental setups in terms of the proportion of

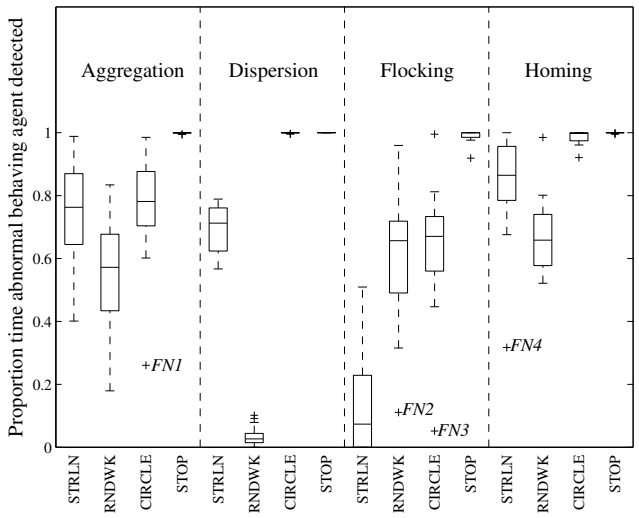

Figure 2: Proportion of time abnormal behaving agent is detected as faulty across 20 replicates, in each of the 16 distinct combinations of normal (aggregation, dispersion, flocking and homing) and abnormal (STRLN, RNDWK, CIRCLE and STOP) behaviors.

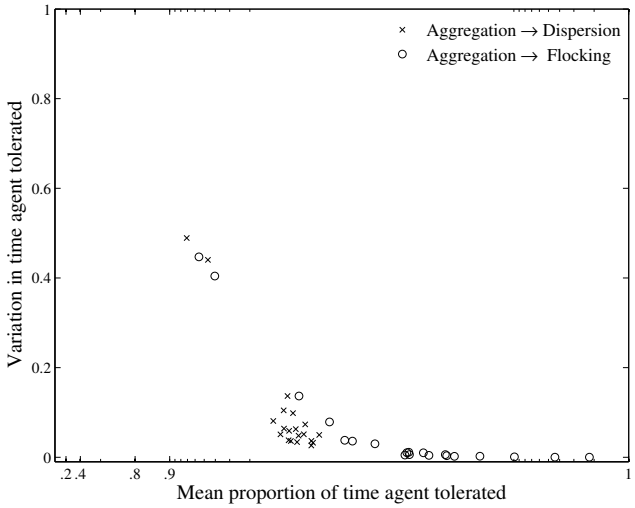

Figure 3: Mean and variation in proportion of time agent tolerated, across the 20 agents of the MAS, in each of 20 replicates, and 2 transitions in normal behavior: (a) aggregation to dispersion (crosses) and, (b) aggregation to flocking (circles).

time that the abnormal agent is detected can be ascribed to initial random placement and orientations of the agents and to the subsequent interactions between normally behaving agents and the faulty agent. In particular, we observed low median proportions of time that the abnormally behaving agent was detected in two experimental setups (see Fig. 2): dispersion/RNDWK and flocking/STRLN.

In the dispersion/RNDWK setup, the dispersing agents perform random walk when they are further than a certain distance from their closest neighbor, and their behavior is therefore identical to that of the faulty agent. It is only when the faulty agent gets close to another agent and does not respond by dispersing away from that agent, that its behavior becomes distinct from the behavior of the rest of the swarm.

The relatively low median proportion of time that the faulty agent is detected in the flocking/STRLN setup (see Fig. 2) is due to the fact that one or more normally behaving agents often end up flocking with the faulty agent moving in a straight line. The behavior of the faulty agent therefore becomes indistinguishable from the behavior of the normal, flocking agents. Below, we explain the situations that caused the outliers *FN1–FN4* (false negatives) in Fig. 2:

FN1: In one aggregation/CIRCLE experiment, an aggregate formed around the faulty circling agent.

FN2: In one flocking/RNDWK experiment, the random walking faulty agent was stochastically followed by different groups of flocking agents for most of the experimental trial.

FN3: In one flocking/CIRCLE experiment, one to two agents "flocked" with the circling faulty agent.

FN4: In one homing/STRLN experiment, the agent moving in a straight line was stochastically assigned a direction of motion aligned with the direction of motion of the moving homing target. Although the abnormally behaving agent moved faster than the homing agent and the target, it was often part of the cluster of agents moving with the homing target.

4.3 Robust classification to online changes in agent behavior

In a third series of experiments, we investigated the capabilities of a MAS to successfully maintain tolerance to normal behavior and to detect abnormal behavior, under variations in agent behavior. At the start of these experiments, all the 20 agents performed an aggregation behavior. After half the simulation time had elapsed, in independent experiments: (a) the behavior of all the agents was simultaneously switched to dispersion, (b) the behavior of all the agents was simultaneously switched to flocking. The results are shown in Fig. 3, where we have plotted the mean proportion of time that agents are tolerated along with the variance in terms of the difference between the maximum and minimum time within the swarm for each replication of the experiment. For both transitions, from aggregation to dispersion and from aggregation to flocking, the agents manage to maintain tolerance to one another even as the swarm undergoes the change in behavior.

The four outliers in Fig. 3, two for replications from aggregation-dispersion, and two from aggregation-flocking experiments, are the result of a small aggregate of only two agents forming in the first part of the experiment in which all agents perform aggregation. As discussed in Section 4.1, the formation of a small aggregate of two agents causes the rest of the agents (all in larger aggregates) to classify the behavior of the two agents in the small aggregate as abnormal.

The outliers listed above are not consequent to the transitions in normal behavior, but because of abnormalities in the aggregation, before the transitions occurs. In order to analyze if the tolerance to transitions in the normal behavior is trivial, i.e., the feature-vectors remain the same, we further analyze their distribution across the 20 agents of the

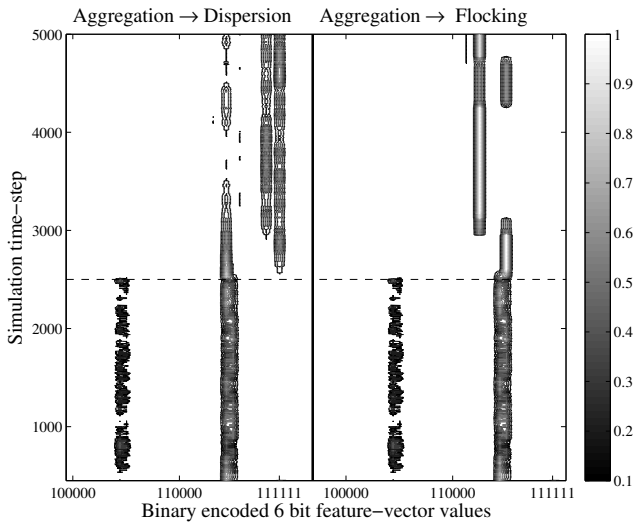

Figure 4: Distribution of 6 bit feature-vectors across the 20 agents of the swarm for the transitions in normal behavior for 2 experiment replicates: (a) aggregation to dispersion (left) and, (b) aggregation to flocking (right). Note, the distance between contours does not reflect Hamming distance.

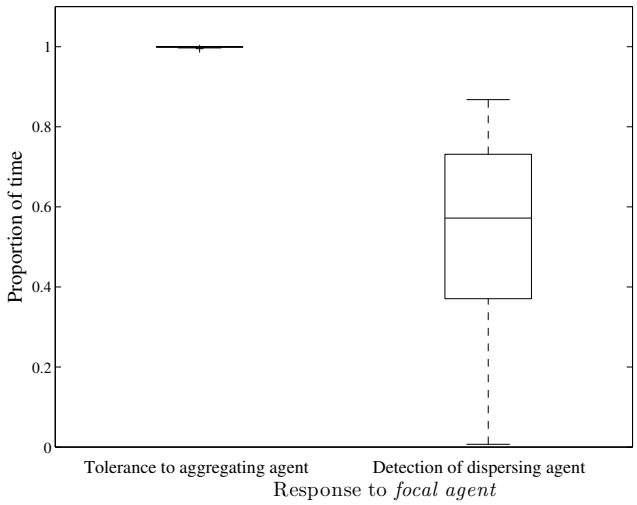

Figure 5: Proportion of time a normal aggregating agent is tolerated during the first half of the simulation (left box), and the same dispersing agent detected as abnormal during the second half of the simulation (right box).

swarm, before and after the transition. Results are shown in Fig. 4 for two replicates, one from the aggregation-dispersion experiment, and one from the aggregation-flocking, where we have plotted the proportion of agents with each of the feature-vectors through simulation time. In both the transitions, the results indicate differences in the feature-vectors of normal behavior, before and after the transition. In the indicated experiment with aggregation-dispersion, the dominant feature-vector shifted by a single bit (54 in aggregation, to 62 in dispersion). However, 25% of the dispersing agents retained the feature- vector 54. In the transition from aggregation-flocking, the difference in feature-vectors were relatively higher, with a dominant proportion of flocking agents shifting their feature-vector by 2 bits, to 51. Furthermore, only 1% of the flocking agents retained the dominant aggregating feature-vector.

In a variation to the above experiments, we allowed only a single *focal agent* to switch behavior from aggregation to dispersion, halfway through the simulation, whereas the remaining agent continued aggregating. The results over 20 replicates (Fig. 5) indicate that the *focal agent* was successfully tolerated during the first half of the simulations. Furthermore, upon switching its behavior to dispersion, this agent was detected as behaving abnormally (median proportion of time detected above 0.5). We also observed that in replicates that did not perform well, the focal agent was in the middle of and surrounded by an aggregate of six of more agents, and consequently could not conduct its dispersion.

5. DISCUSSION

Our results revealed a robust maintenance of tolerance to normal swarm behaviors, understood as being exhibited by a large proportion of the agents, irrespective of the type of swarm behaviors simulated. In our study, agents behaving normally are always sensed in abundance (agents sense feature-vectors of 10 nearest neighbors, irrespective of their distance). Additionally, agents perform an identical swarm behavior in each experiment. Consequently, our abnormality detection system does not require to look for persistence in normal behavior, only abundance. In order to improve response-time to transitions in normal behavior, the integration time of the CRM performed at each simulation time-step was set at a high value (5×10^7 a.u.). At lower values of this parameter, we may observe the need for persistence in characterization of normal behavior. In such a parameter regime, a newly transitioned normal behavior would provoke an immune response for some time before being tolerated. Interpreting this, our current implementation of the model may be considered to follow a majority rule with some margin of error in tolerating normal feature-vectors. Importantly, the resilience to error is essential, since the normal behavior is not just represented by a single feature-vector, but distributed over a set of them (Fig. 4), all of which may not be abundant. Hence, abnormality detection using a simple sub-threshold on the count of agents with the same feature-vector would not work.

The abnormality detection model introduced in this study highlights an important difference between fault detection and abnormality detection. The behaviors that are not exhibited by most of the agents in the swarm may be considered as abnormal, although they may not necessarily be faults. For example, in experiments involving tolerance to aggregation behavior, the false positives (*FP1* and *FP2* in Fig. 1) observed were consequent to the two agents performing the aggregation behavior differently from the rest of the swarm. In order to detect such abnormalities, another layer may be introduced on top of our abnormality detection algorithm, forcing the detected agents to undergo certain application-specific test cases. Similarly, in the detection of specific faulty agents, the false negatives (*FNs* in

Fig. 1) observed were because faulty behavior became indistinguishable from normal ones (e.g., flocking agents moving in a straight line). It may be argued that the faults may not be a hindrance, if they do not disrupt the behavior of the swarm. However, this needs to be explored further with a quantitative analysis of swarm behaviors.

In our study, we assume the CRM of all the agents function normally. This is a critical consideration, since abnormalities may not only be restricted to the sensor, motor devices, and control software of the agent, but may also affect the proper execution of the CRM. Consequently, it is worth exploring the impact of perturbations on the CRM itself, as they could drive anomalous behaviors akin to "autoimmunity". It may be expected that communication of virtual T-cells from "healthy" neighboring agents, and a consensus amongst neighboring agents in the decision to mount an immune response, may dampen the effects of such perturbations. However, this needs further investigation.

6. CONCLUSIONS AND FUTURE WORK

In this study, we proposed an approach inspired by the capability of the adaptive immune system, to detect agents behaving abnormally due to faults, in MAS. We examined the validity of our approach using a collection of typical swarm behaviors, while introducing behaviors simulating common faults encountered by robots due to malfunctions in their sensors, motors and control software. Our approach to fault detection utilizes relatively few agents, and may therefore compliment the existing centralized approaches used in traditional MRS. Moreover, because of the inherent distributed nature of fault detection, our approach may also be applicable to swarm robotic systems, where centralized approaches may not always be feasible.

The results of our study encourage us to explore more realistic scenarios involving MRS exhibiting different behaviors depending on environmental contingencies, wherein the capability of our immune system model to maintain a history of robot behaviors may be used to characterize normality in more complex tasks. Furthermore, a detailed comparative investigation of the results of our immune system model with other fault detection algorithms used in MRS, is also underway.

Supplemental Data: Movies of MAS simulations are available online at `www.isr.ist.utl.pt/~dtarapore/AAMAS2013/videos`.

Acknowledgment: This study was supported by the FCT grants PTDC/EEACRO/104658/2008 and PEst-OE/EEI/LA0009/2011.

7. REFERENCES

[1] A. Abi-Haidar and L. Rocha. Adaptive spam detection inspired by the immune system. In *Proceedings of the 11th International Conference on the Simulation and Synthesis of Living Systems, Artificial Life XI*, pages 1–8. MIT Press, Cambridge, 2008.

[2] R. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14–23, 1986.

[3] B. Bullnheimer, R. F. Hartl, and C. Strauss. An improved ant system algorithm for the vehicle routing problem. *Annals of Operations Research*, 89:319–328, 1997.

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

[5] J. Carneiro, K. Leon, I. Caramalho, C. Van Den Dool, R. Gardner, V. Oliveira, M. Bergman, N. Sepúlveda, T. Paixão, J. Faro, and J. Demengeot. 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, 2007.

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

[7] A. L. Christensen, R. O'Grady, and M. Dorigo. From fireflies to fault tolerant swarms of robots. *IEEE Transactions on Evolutionary Computation*, 13(4):1–12, 2009.

[8] V. A. Cicirello and S. F. Smith. Wasp-like agents for distributed factory coordination. *Autonomous Agents and Multi-Agent Systems*, 8(3):237–266, 2004.

[9] V. Crespi, A. Galstyan, and K. Lerman. Top-down vs bottom-up methodologies in multi-agent system design. *Autonomous Robots*, 24(3):303–313, 2008.

[10] D.Tarapore, A. L. Christensen, P. U. Lima, and J. Carneiro. Environment classification in multiagent systems inspired by the adaptive immune system. In *Proceedings of the 13th International Conference on the Simulation and Synthesis of Living Systems, Artificial Life XIII*, pages 275–282. MIT Press, Cambridge, 2012.

[11] S. Hauert, J. Zufferey, and D. Floreano. Evolved swarming without positioning information: an application in aerial communication relay. *Autonomous Robots*, 26(1):21–32, 2009.

[12] K. Leon, A. Lage, and J. Carneiro. Tolerance and immunity in a mathematical model of T-cell mediated suppression. *Journal of Theoretical Biology*, 225:107–126, 2003.

[13] M. Rehák, M. Pěchouček, and J. Tožička. Adversarial behavior in multi-agent systems. In M. Pěchouček, P. Petta, and L. Varga, editors, *Multi-Agent Systems and Applications IV*, volume 3690 of *Lecture Notes in Computer Science*, pages 470–479. Springer, Berlin, Germany, 2005.

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

[15] M. H. Terra and R. Tinos. Fault detection and isolation in robotic manipulators via neural networks: A comparison among three architectures for residual analysis. *Journal of Robotic Systems*, 18(7):357–374, 2001.

[16] P. R. Wurman, R. D'Andrea, and M. Mountz. Coordinating hundreds of cooperative, autonomous vehicles in warehouses. In *Proceedings of the 19th National Conference on Innovative applications of artificial intelligence - Volume 2*, IAAI'07, pages 1752–1759. AAAI Press, 2007.