



Towards the development of robot immune system: A combined approach involving innate immune cells and T-lymphocytes

Maria Akram*, Ali Raza

Department of Mechatronics and Control Engineering, University of Engineering and Technology, Lahore, Pakistan

ARTICLE INFO

Keywords:
Mobile robots
Artificial immune system
Homeostasis
Fault detection and health recovery

ABSTRACT

Mobile robots in uncertain and unstructured environments frequently encounter faults. Therefore, an effective fault detection and recovery mechanism is required. One can possibly investigate natural systems to seek inspiration to develop systems that can handle such faults. Authors, in this pursuit, have explored the possibility of designing an artificial immune system, called Robot Immune System (RIS), to maintain a robot's internal health-equilibrium. This contrasts with existing approaches in which specific robotic tasks are performed instead of developing a self-healing robot. In this respect, a fault detection and recovery methodology based on innate and adaptive immune functions has been successfully designed and developed. The immuno-inspired methodology is applied to a simulated robot using Robot Operating System and Virtual Robot Experimentation Platform. Through extensive simulations in increasingly difficult scenarios, the RIS has proven successful in autonomously detecting the abnormal behaviors, performing the recovery actions, and maintaining the homeostasis in the robot. In addition to being multi-tiered, the developed RIS is also a non-deterministic and population-based system.

1. Introduction

The performance of mobile robots in unstructured and real-world scenarios has shown a noticeable lack of reliability. Carlson and Murphy's work (Carlson and Murphy, 2003) on testing the reliability of different mobile robots over a period of two years concluded that faults in mobile robots are quite frequent. Some of the faults can be related to hardware including partial or complete failure of any physical component of a robotic system. Such faults influence the incoming data from robot-sensors as well as the execution of tasks. The faults may also arise from software-related issues involving faulty algorithms and or faulty implementations of correct algorithms. Such faults influence behavior of the robot in terms of perception, decision-making, and task execution. Another category can be the interaction-related faults which may arise due to unstructured nature of environment and exogenous events (Khalastchi and Kalech, 2018). For continuous and reliable operation of robotic systems, it is important that these faults are *detected* and *diagnosed* autonomously and continuously, like in humans. Moreover, it is equally important to have a *recovery* mechanism against such faults.

In an attempt to search a suitable method to handle such faults, one can possibly look into human's capability to recognize and heal abnormalities restoring the internal health equilibrium (homeostasis).

Biological immune system (BIS) (Abbas et al., 2014) can be a source of inspiration because it displays functions of feature extraction, distributed control, self-regulation, self-organization, learning, adaptability and dynamic memory (Van Parijs and Abbas, 1998; Medzhitov and Janeway, 1998; CAPRA et al., 1999). The immune system acts to help protect and heal the body's cells and tissues by differentiating between normal and abnormal functioning cells and tissues, as well as by handling invading pathogens (Medzhitov and Janeway, 1997). Therefore, practically valuable analogies can be made between fault-detection-and-recovery in systems and the functions of BIS (Timmis et al., 2010). Thus, artificial immune system (AIS) (d Castro and Timmis, 2002), inspired from BIS, is being used in many applications such as anomaly detection (Bayar et al., 2015), pattern recognition (Wu et al., 2016), data mining (Timmis et al., 2000), computer security (Kim et al., 2007), adaptive control and fault detection (Dasgupta, 2006). AIS has also been effectively used in robotic applications for navigation, fault detection, optimization, and robot trap escaping, as well as in multi-robot systems (MRS) e.g. to handle communications during team work (Raza and Fernandez, 2015).

However, the literature on immune-inspired robotic applications indicates that the immunological functions have been employed only to solve specific robotic tasks (Raza and Fernandez, 2015; Hart and Timmis, 2008), instead of developing *self-healing* robots. It has also been

* Corresponding author.

E-mail addresses: mariaakram86@gmail.com (M. Akram), ali.raza@ymail.com (A. Raza).

observed that the possibility of using BIS analogies to formulate a similar immune system for robots is not explored. In this context, authors propose an alternate approach. The proposed approach, assuming that if the robot's health is maintained then task-completion will be a natural outcome, focuses on the robot's health maintenance. This new approach may induce a paradigm shift.

The basic idea of our research is to look inwards, get inspiration from the biological immunity and develop an immune system for robots, called Robot Immune System (RIS). The RIS will help to produce robust and dependable robots that can maintain a healthy homeostatic-balance and manage self-sufficiency. This requires continuous monitoring of predefined health indicators (e.g., energy and temperature) to identify any deviation from normal behavior. The RIS uses the data from these health indicators, labelled as Pathogen Associated Molecular Patterns (PAMPs), to activate immune cells. Abnormalities in the health indicators, due to any fault in the robot, increase the inflammation levels. The rise in inflammation is handled through regulating the immune responses by stimulating initially the innate and later the adaptive immune functionalities of the robot. Cells of the presented innate immune component include reactive immune cells (RICs) and antigen presenting cells which later transform into dendritic cells. The rise in inflammation level triggers RICs to move towards the fault location and perform recovery actions, as illustrated in Fig. 2. In cases, where RICs fail to regulate the homeostatic balance, antigen-presenting cells transform into dendritic cells and stimulate the concerned T-cells to proliferate (cloning) and perform recovery actions to re-attain the homeostatic balance. This multilevel structure enables the robot to successfully recover from faults.

The immuno-inspired approach presented herein successfully ensures that the robot is able to handle various faults and maintains homeostasis. Although the approach does not directly aim at performing different tasks, it helps in effective and efficient execution of tasks by ensuring maintenance of the robot's health. Robot Operating System (ROS) (Koubaa, 2017) is used to implement the proposed RIS to ensure availability of our approach for transparency and benchmarking, whereas the simulation is done using Virtual Robot Experimentation Platform (V-REP) (Rohmer et al., 2013). The remaining paper is outlined as follows: section 2 presents the review of relevant literature on

fault detection and recovery in robots using AIS. In section 3, the methodology to develop the RIS is presented. Section 4 presents the simulation results and statistical analysis of the developed algorithm. A discussion on simulation results and performance of the developed RIS is provided in section 5. Finally, the conclusions and future work is presented in section 6.

2. Related Work

Fault detection is the first and necessary stage towards developing the proposed robot-immunity. Observing a system's behavior can be the starting point. Winfield and Nembrini (Bjerknes and Winfield, 2013) observed that robots exhibit distinguishable abnormal behavior, as compared to the expected one, when they undergo different faults. Thus, deviations from normal behavior can be inferred as signs of faults in the robot. Researchers have explored many fault diagnosis approaches, examples include approaches using neural network learning (Huang and Tan, 2008; Johnson et al., 2018; Ferrari et al., 2012), genetic algorithms (Trivun et al., 2017), reinforcement learning (Chatzilygeroudis et al., 2018), Kalman filters (Jing et al., 2017; Kamel et al., 2018), fuzzy logic systems (Bakdi et al., 2017; Jiao and Li, 2018) and artificial immune systems (Laurentys et al., 2010). The AIS have been effectively applied in robotic systems (Raza and Fernandez, 2015) and a significant number of researchers have tried to *identify faults* in robotic systems using AIS-based approaches, whereas only a few have attempted some *fault-recovery* actions (Ismail and Timmis, 2010; Khadidos et al., 2015). The following discussion, in this context, presents the related work in both the categories.

2.1. AIS-based fault detection systems

The literature review reveals that a few immuno-inspired approaches have been developed for fault detection in mobile robots. For example, (Abbas et al. (2014)) developed a system for error detection based on the concept of *self* and *non-self* discrimination using negative selection method. Their system identifies error on the basis of input-output mapping. The output data which fails to map with the corresponding input is described as non-self and is considered as an error.

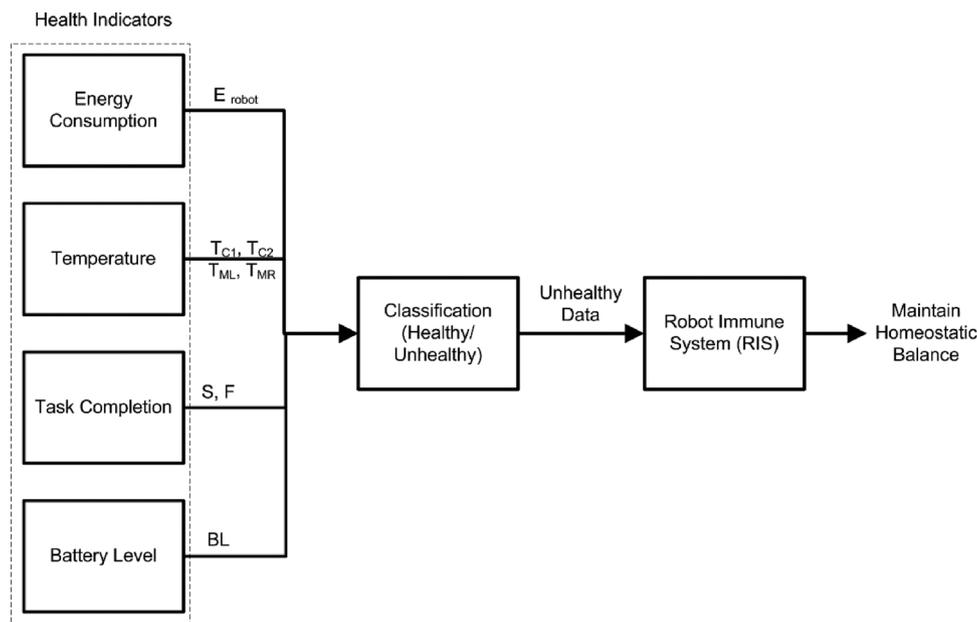


Fig. 1. The Proposed Immuno-Inspired Scheme for the Robot.

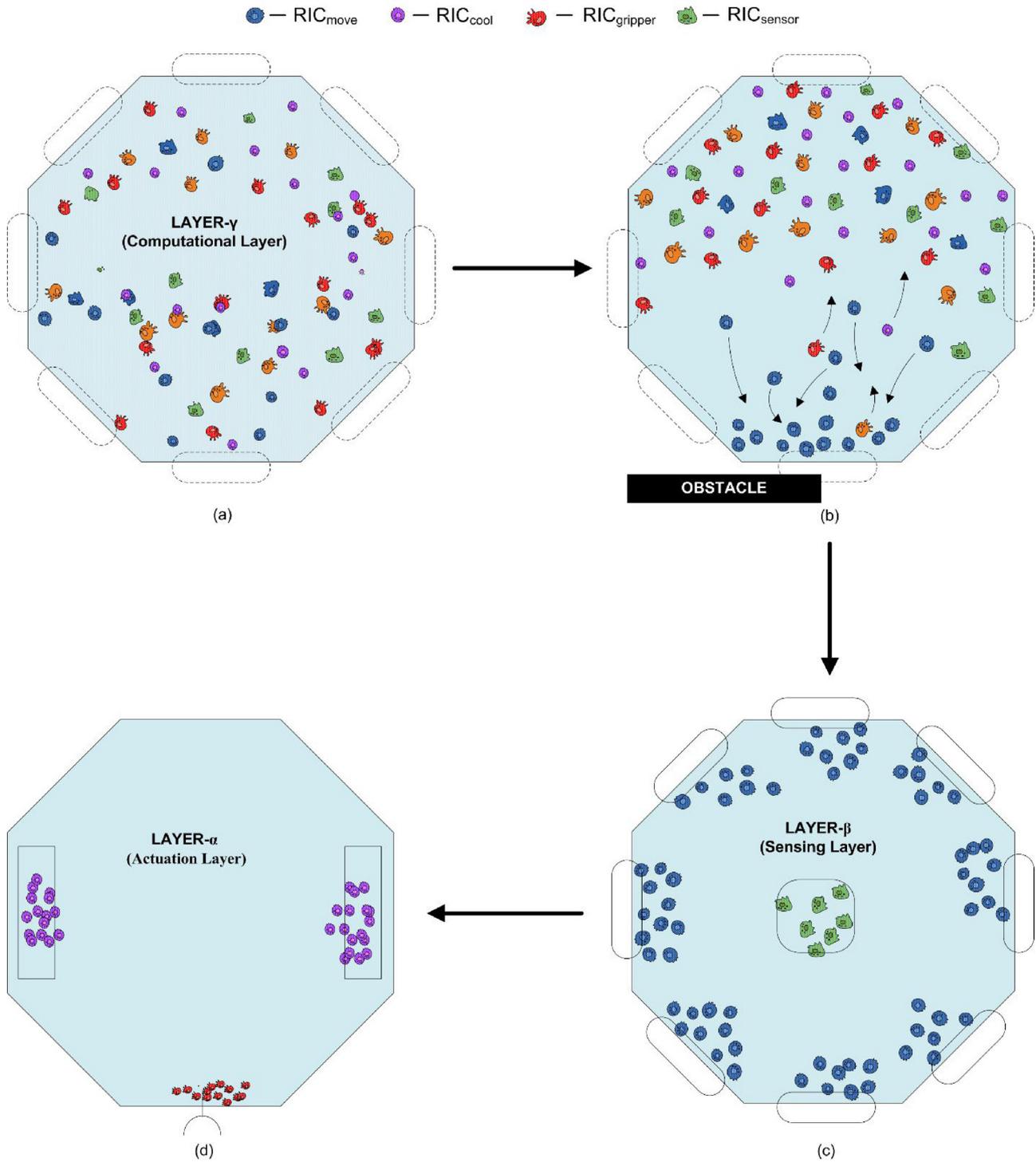


Fig. 2. The Behaviour of Reactive Immune Cells (RICs) of Innate Immunity against Faults. The presence of various RICs throughout the robot's body is shown in 2(a). Mobilization of RICs to the fault location can be seen in 2(b). Figs. 2(c) and 2(d) illustrate RICs surrounding different parts of the robot's body in case of faults at these parts.

However, the system is limited only to the error-detection and does not take any action to overcome the instability in the robot-behavior.

On similar lines, Canham et al. trained Khepera (Mondada et al., 1999) and Rascal (Behringer et al., 2005) robots during fault free

operations and then tested their capability to detect faults using different pre-built fault detectors (Canham et al., 2003). Other relevant work, described by Bjercknes (Bjercknes and Winfield, 2013), proposed the use of dendritic cell algorithm (DCA) to identify instantaneous

Table 1
List of Tasks and ways to measure task completion in the presence of Failures.

Nature of task	Measurement of Task Completion
Mapping	The completeness of map
Goal seeking	By calculating the Euclidean distance between the goal position and the destination position of the robot

health status of the robot to be anomalous or normal. DCA performs anomaly detection based on the indication of danger associated with different parts of the system. AIS has also been used effectively to determine faults in hardware components of robots like performance of DC motors (Crestani et al., 2015), high current in servo motors during robot walking, collisions with some obstacles, disconnection of some motor or isolation of some joint (Jakimovski and Maehle, 2008; Kernbach et al., 2009), etc. Other applications include sensor noise neutralization (Ishida and Adachi, 1996), sensor replacements in mobile robots (Watanabe and Jyo, 2014), path planning in dynamic environment (Eslami, 2018), behavior coordination in robots (Fernandez-Leon et al., 2011), and detection of abnormal robots having some hardware or software faults irrespective of temporal changes in swarm behavior (Tarapore et al., 2015). All of these approaches focus only on fault detection, not on important aspects of diagnosis and recovery.

AIS has also been used for fault detection in multi-robot systems. (Tarapore et al. (2017)) presented an immune-inspired algorithm for fault detection in robot-swarms based on cross regulation model (CRM). CRM mimics the dynamics of effector-T-cell and regulatory-T-cell populations during their interactions with antigen presenting cells. The algorithm of Tarapore et al. classifies the behavior of robots in a swarm to be normal and abnormal which is then tested with 20 e-puck robots in the multirobot simulator ARGoS. Performance of the algorithm is evaluated against the capability of the system to detect faulty robots and isolates them from the swarm. The algorithm does not consider the possibility of any recovery action in faulty robots to regain the normal behavior.

2.2. AIS-based fault detection-and-Recovery systems

On the other side, the self-healing feature is not explored much in mobile robotic applications. An example is the work by (Timmis et al. (2016)) proposing an immune-inspired approach for self-healing under certain failure modes to address the anchoring of robots in the swarm due to partial failures. The authors also compared the experimental results of their algorithm with the existing algorithms i.e. Single Nearest Charger Algorithm, ω -algorithm, and Shared Nearest Charger Algorithm. As a result, they found that their algorithm performed significantly better, when the number of faulty robots were more than two. However, the algorithm is applied only on MRS in supervisory settings and does not provide any provision for the robot to have its own recovery or healing system. The faulty robots depend on the healthy robots in the swarm to come forward and heal them.

Table 2
List of Different Types of Immune Cells at Innate and Adaptive Immune Level for Recovery Actions to Maintain the Homeostasis in the RIS.

Immune Cells	Types	Description
Reactive Immune Cells (RICs) (Innate Immune Level)	RIC _{move}	RICs seek to move the robot out of any stuck position.
	RIC _{cool}	RICs seek to cool down the temperature of various components by switching on the central fan.
T-Cells (Adaptive Immune Level)	TC _{left}	T-cells seek to move the mobile robot towards left.
	TC _{right}	T-cells seek to move the mobile robot towards right.
	TC _{forward}	T-cells seek to move the robot in forward direction.
	TC _{backward}	T-cells seek to move the robot in reverse direction.
	TC _{cool-motor-left}	T-cells seek to cool down the temperature of left motor by switching on left motor fan.
	TC _{cool-motor-right}	T-cells seek to cool down the temperature of right motor by switching on right motor fan.
	TC _{cool-controller-1}	T-cells seek to cool down the temperature of controller-1 by switching on controller-1 fan.
	TC _{cool-controller-2}	T-cells seek to cool down the temperature of controller-2 by switching on controller-2 fan.

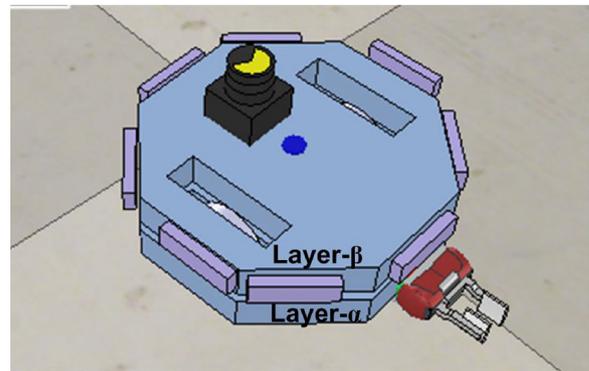


Fig. 3. 3D View of the Developed Robot Highlighting Actuation Layer (Layer-α) and Sensing Layer (Layer-β).

Adil et al. (Khadidos et al., 2015) presented their work proposing an approach to detect the faulty robot among its neighboring healthy robots. On the confirmation of unhealthy state of a robot by healthy robots, the controller shuts down the faulty robot to avoid any chaos among the rest of the robots in the swarm. Another example is the work of Tahir et al. (Khan and De Silva, 2009) which illustrates that the faulty robot among the swarm is isolated to avoid crippling of the system and to maintain the system's performance. The idea of just isolating and shutting down the faulty robots can help to decrease disturbance among the healthy robots of a swarm but is a weak representation of healing process.

In conclusion, the detection of faults has been explored in the literature but not on the notion of health awareness in the robots. Health awareness, in this context, refers to the ability of robots to determine the current environmental conditions as well as assess their internal health status. This awareness will help the robots to take necessary actions and regain health before the degradation of their performance. Therefore, in this paper, we have presented a robot immune system (RIS) which mimics innate and adaptive immune functions of BIS to regulate homeostasis by performing sustained maintenance of the robot's health. In case of unhealthy status, the system autonomously counter abnormalities by appropriate recovery actions like those in humans. The unhealthy status is determined using an objective function (Eq. (18)) that represents the system's current state. The developed RIS not only classifies the robots' current health status as either healthy or unhealthy but also performs actions at two different immunological levels to maintain homeostasis. The methodology of this RIS is presented in the following section.

3. Methodology

The primary objective of the developed methodology revolves around maintaining homeostasis in the robot. In biological context, homeostasis refers to an equilibrium that is maintained by biological feedback albeit the changing conditions. In the context of our approach,

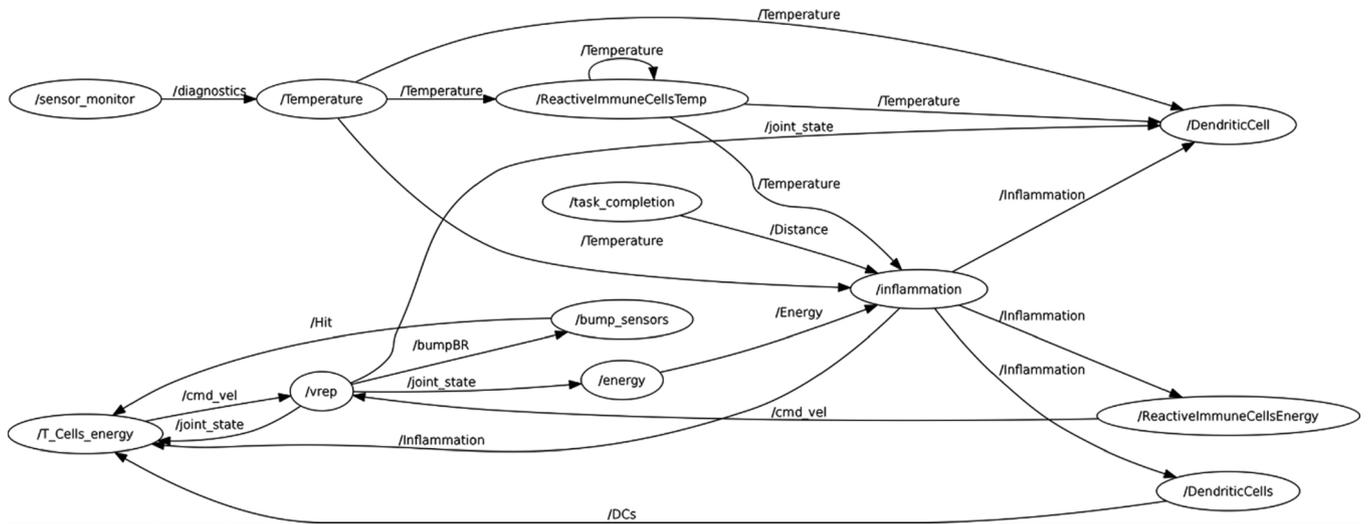


Fig. 4. ROS-Graph showcasing active nodes and topics of ris_v1 for the implementation of RIS. The eclipses represent nodes, where each node is an individual program; rectangles represent topics on which messages are published.

Table 3
List of ROS Packages Used to Assist the Developed ROS Package for Robot Immune System.

The platform	ROS Indigo Igloo on Ubuntu 14.04 LTS (Trusty)
Stack used for autonomous navigation	Navigation stack
Package used for manual navigation	teleop_twist_keyboard
Package used for temperature information	diagnostic_common_diagnostics
Package used for 3D visualization	Rviz
Package used for mapping environment	Gmapping
Package used for visualizing numeric values in multiple 2D plots	rqt_multiplot

the homeostasis concept in robots involves minimizing adverse health-effects, caused due to the existence of different faults defined earlier. This involves continuous monitoring of health parameters, classification of self and non-self-data, fault-detection, and recovery actions. Fig. 1 depicts the proposed RIS that detects and classifies abnormalities in the robot through its health indicators, and subsequently maintains the homeostatic balance using the innate and adaptive immunological functions.

3.1. Health indicators

The key to maintain internal health of a robot is to identify *intrinsic* and *extrinsic* parameters that can project the true representation of its internal health or homeostatic balance. We call these parameters health indicators, as they would represent robot’s health and abnormality beyond usual and would help us to determine any disturbances in the internal equilibrium of robot. These parameters are discussed here.

3.1.1. Energy consumption

One of the most important characteristic of a mobile robot is its mobility. Mobile robots have to move when mapping, exploring, goal-seeking, etc. All these tasks consume energy. Some of the energy will also be required for sensing, computing, or communication. Therefore, we selected the energy consumption as our first intrinsic health parameter. The rate of energy consumption is used as one of the ways to detect any abnormality in the functioning of the robot. For example, when a robot experiences a bump the motors draw more current and

resultantly more energy is consumed. Energy consumption is computed using power consumption over time i.e., area under the power curve by sensors and motors given in the following equation (Ahmed et al., 2015):

$$E_{robot} = \int_{t_1}^{t_2} P_{robot} dt ; \tag{1}$$

$$P_{robot} = \sum_{k=1}^n P_k(\omega) + P_s ; \tag{2}$$

$$P_k(\omega) = \tau_k * \omega_k ; \tag{3}$$

where E_{robot} = Energy consumed by the robot,
 P_{robot} = Power required by the robot,
 n = number of motors,
 τ_k = Torque delivered by k^{th} motor,
 ω_k = Angular speed of the k^{th} motor, and
 t_1 and t_2 = The time instances between which the energy consumed is calculated.

Sensing power P_s depends on the sensing frequency, f_s , which is 10 Hz in our case for Hokuyo laser scanner URG-04 LX-UG01 and can be determined by a simple linear function (Mei et al., 2005):

$$P_s = c_0 + c_s f_s \tag{4}$$

where c_0 and c_s are constants.

3.1.2. Battery level

In addition to energy consumption rate, battery level is another related health indicator (intrinsic). Fast battery depletion is considered as a fault and is regulated by innate and adaptive immune functions of the RIS. A low battery level also serves as an alarm to hold the robot’s current task and rush the robot to the docking station for recharging batteries before they are fully drained.

3.1.3. Temperature

Temperatures of different physical components of the robot (i.e., motors and controllers) are set of critical intrinsic parameters that point towards healthy functioning of the robot. We used the direct relationship of temperature and delivered motor-torque to determine temperature of motors and a specific ROS package (diagnostic-common-diagnostics¹), which provides the temperature and memory usage of different cores of the computer, to determine the temperature of robot

¹ Please see http://wiki.ros.org/diagnostic_common_diagnostics

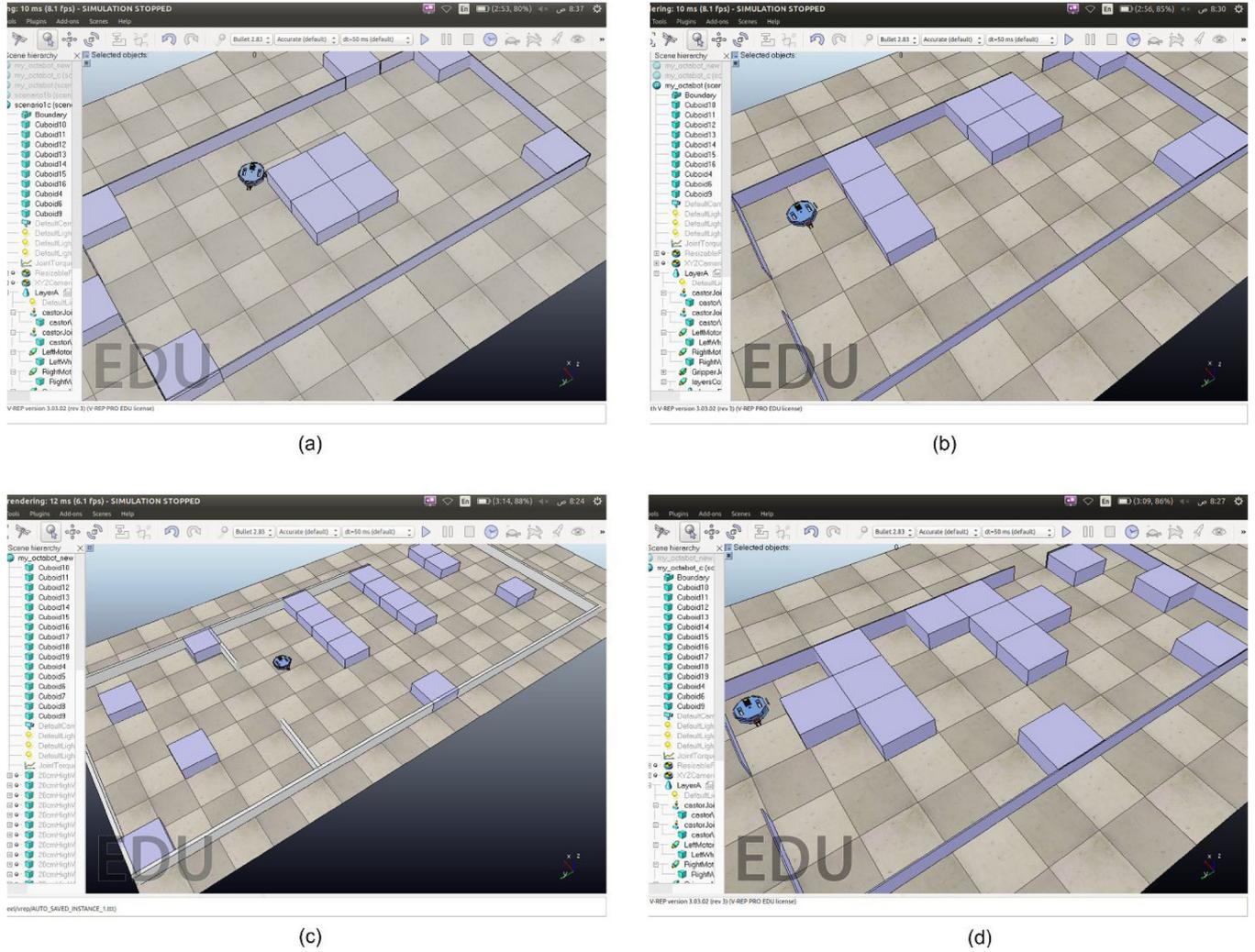


Fig. 5. Snapshots of the simulation environment scenes with different scenarios in V-REP. (a) and (b) represent simple environments. (c) represents the environment having two rooms sharing a common door with different obstacles. (d) represents the most complex environment in terms of obstacles placement.

controllers. During simulations, all the computational requirements of the robotic activity were assigned to the core-2 of the computer, whereas the RIS was executed solely by core-1. As such, temperature of each core represented temperature of the robot’s respective controller during simulation studies.

3.1.4. Task completion

The task-completion is used as the only extrinsic health parameter to evaluate how efficiently the robot has completed its tasks. This extrinsic parameter may be affected due to any fault inside the robot or unstructured nature of the environment in which it is working. During the experimentation, the robot had to perform three tasks: wandering, mapping, and goal seeking. The task completion parameter monitors accuracy and completeness of the task performed in the presence of induced failures as shown in Table 1.

For example, in the case of goal seeking it observes how close the robot has reached its goal by finding the Euclidean distance between the goal and its own instantaneous position.

3.2. Classification

The classification process involves making sense of the health-indicators data and later passing the health-state (healthy or unhealthy) to the RIS as shown in Fig. 1. The operations of the Classification Block are discussed in the following detail.

3.2.1. PAMP Signal (PS)

PS is used to represent a potentially problematic signal coming from the health indicators. The value of PS outside the usual range of values indicates abnormalities in the robot’s functioning. PS is expressed as:

$$PS(k) = \{E_{robot}(k), T_{C1}(k), T_{C2}(k), T_{M1}(k), T_{M2}(k), BL(k)\} \quad (5)$$

where E_{robot} = Energy consumed by the robot,
 $T_{C1}(k)$ = Temperature of controller-1 for RIS implementation,
 $T_{C2}(k)$ = Temperature of controller-2 for robot simulation,
 $T_{M1}(k)$ = Temperature of motor-1,
 $T_{M2}(k)$ = Temperature of motor-2, and
 $BL(k)$ = Battery Level.

3.2.2. Danger signal (DS) and safe signal (SS)

Samples of the health indicators collected at consecutive time instances (k and $k-1$) provide the information to determine whether the robot is in a danger or safe-state. If the value of a health indicator is below the predefined threshold, then the health-state of the robot is tagged as *safe*. In contrast, if the values are greater than the predefined thresholds then the robot is in *danger*. Greater the difference, greater will be the instantaneous danger and vice versa. The danger signal helps to represent the stress the robot might be experiencing due to any fault. The equations to determine the DS and SS signals are given as:

$$DS(k, 1) = E_{robot}(k) - E_{robot}(k-1); \quad (6)$$

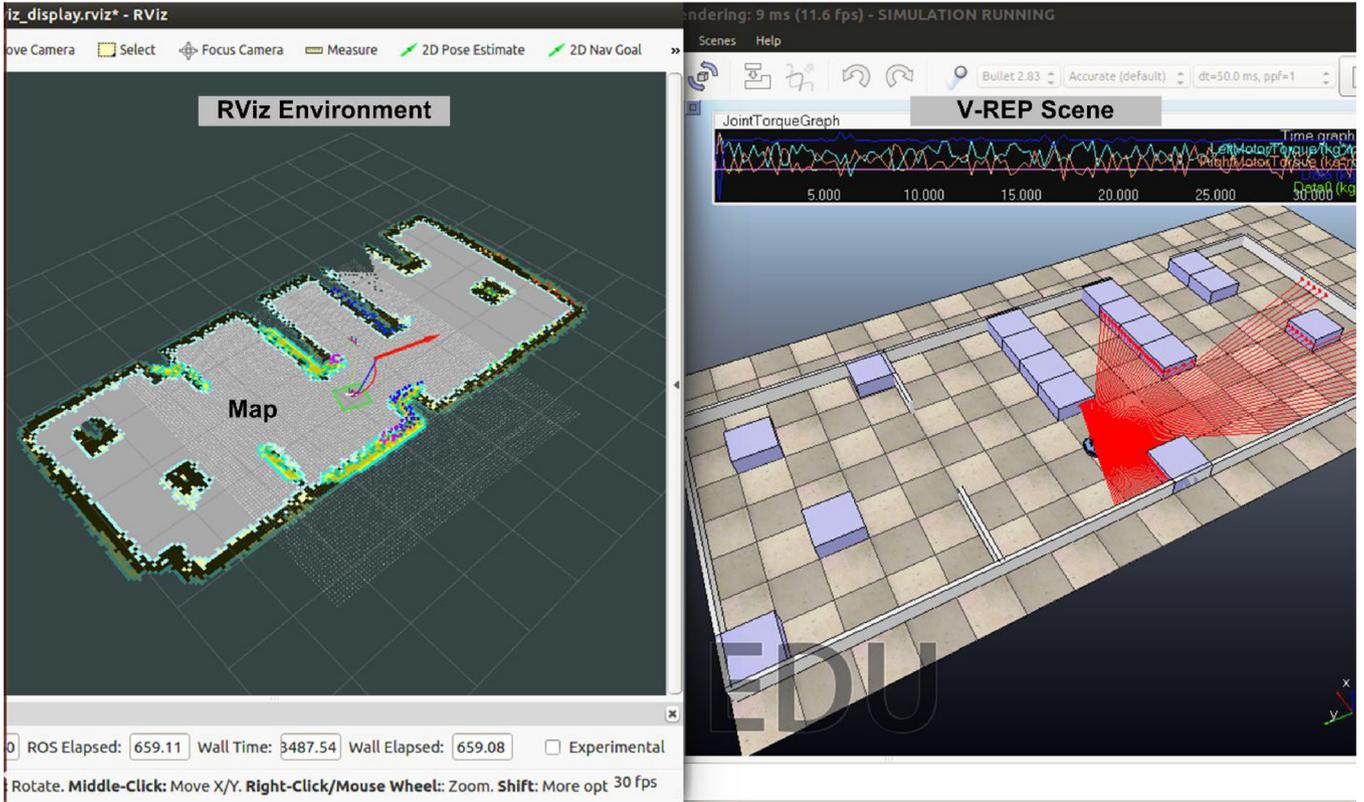


Fig. 6. Visualization of Mapping and Goal seeking in RViz tool for the environment shown in the V-REP scene.

Table 4
Fixed Parameters of the Robot for all Simulations.

Parameter	Description
Robot radius	25cm
Robot normal speed	10 cm/s
Battery capacity	5000J
Minimum energy required to move	500J
Possible faults	Sensor Failure (F1) Battery Drain (F2) Heating up of Components (F3)
Robot tasks	Wandering Mapping Goal Seeking

$$DS(k, 2) = T_{C1}(k) - T_{C1}(k-1); \quad (7)$$

$$DS(k, 3) = T_{C2}(k) - T_{C2}(k-1); \quad (8)$$

$$DS(k, 4) = T_{M1}(k) - T_{M1}(k-1); \quad (9)$$

$$DS(k, 5) = T_{M2}(k) - T_{M2}(k-1) \quad (10)$$

$$DS(k, 6) = BL(k) - BL(k-1) \quad (11)$$

$$SS(k, 1) = E_{robot}(k) - E_{robot}(k-1); \quad (12)$$

$$SS(k, 2) = T_{C1}(k) - T_{C1}(k-1); \quad (13)$$

$$SS(k, 3) = T_{C2}(k) - T_{C2}(k-1); \quad (14)$$

$$SS(k, 4) = T_{M1}(k) - T_{M1}(k-1); \quad (15)$$

$$SS(k, 5) = T_{M2}(k) - T_{M2}(k-1) \quad (16)$$

$$SS(k, 6) = BL(k) - BL(k-1) \quad (17)$$

3.2.3. Inflammatory signal (IS)

IS is used to amplify the effects of PS, DS, and SS. Moreover, the rise in inflammation level beyond the usual triggers the appropriate functions of the RIS, as detailed in the forthcoming sections. The objective function for the inflammation signal to determine the internal equilibrium of the robot is given by:

$$F(k) = \varepsilon_1 * (w_e f_e(k) + w_t f_t(k) + w_b f_b(k) + w_p f_p) \quad (18)$$

$$f_e(k) = \alpha_1 E_{robot} + \alpha_2 P_s \quad (19)$$

$$f_t(k) = \beta_1 T_{C1} + \beta_2 T_{C2} + \beta_3 T_{M1} + \beta_4 T_{M2} \quad (20)$$

$$f_b(k) = \theta_1 BL \quad (21)$$

$$f_p(k) = \gamma_1 D_A + \gamma_2 D_L \quad (22)$$

where $F(k)$ is the objective function to find the inflammation signal, $f_e(k)$ is the function for energy consumption and w_e is the associated weight,

$f_t(k)$ is the function for the temperature of the robot components and w_t is the associated weight,

$f_b(k)$ is the function for battery level and w_b is the associated weight, $f_p(k)$ is the function for the evaluation of task completion and w_p is the associated weight,

D_A is the angular and D_L is the linear difference in goal position and robot position when reached its target in terms of Euclidean distance and

$\alpha_1, \alpha_2, \beta_1, \beta_2, \beta_3, \beta_4, \gamma_1, \gamma_2, \theta_1$ and ε_1 are all scaling constants.

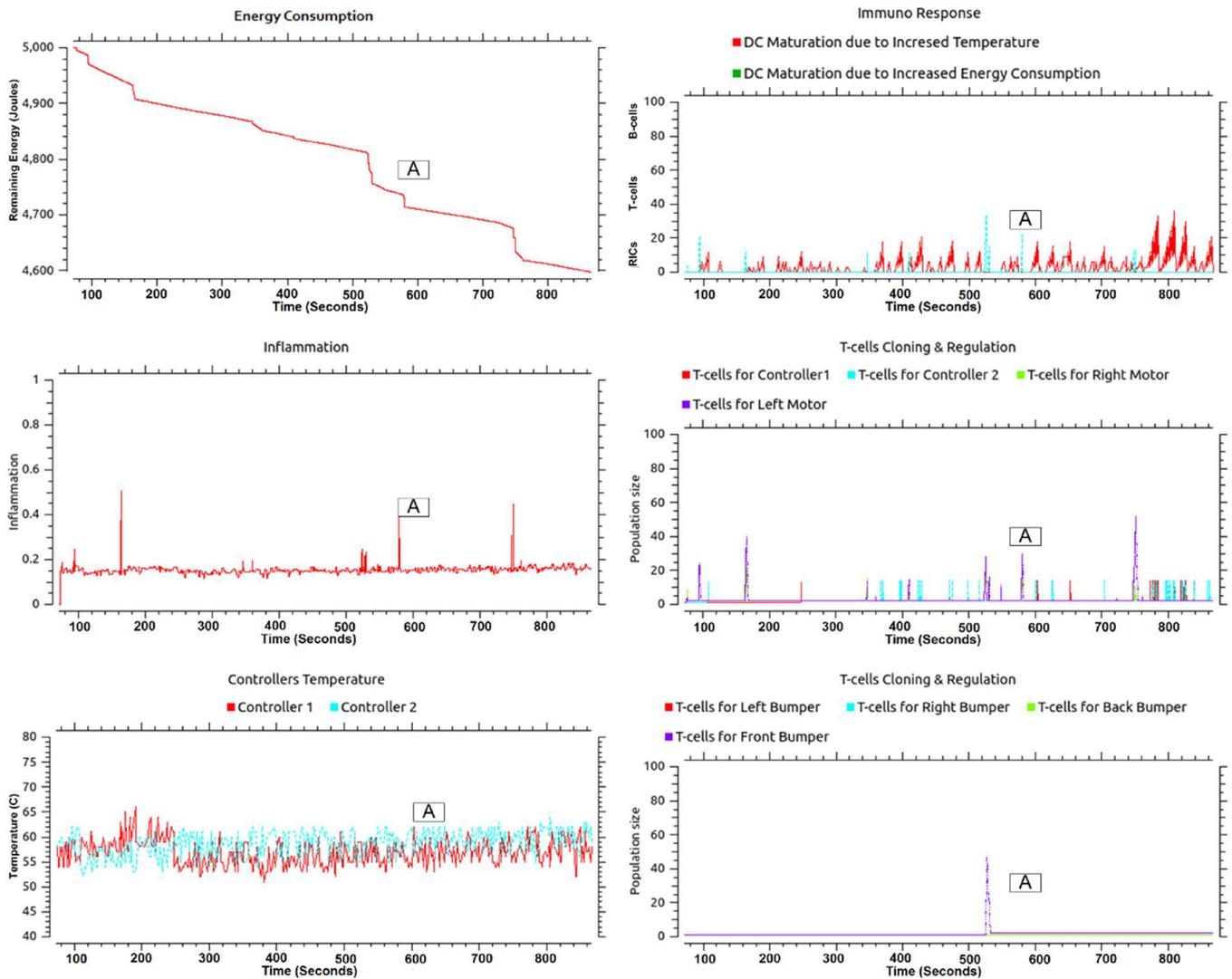


Fig. 7. Results illustrating the immuno-response against abnormalities in health parameters to bring back the Homeostatic balance.

3.3. Fault detection and recovery

Abnormality in the health indicators due to a fault results in the inflammation-rise. If the inflammation level is greater than the predefined threshold, a random population of reactive immune cells (RIC) of innate immunity is directed to move towards the fault location. A list of different types of RICs as well as the T-cells of adaptive immunity are presented in Table 2. RICs are guided towards the fault location using the concept of cytokines. The function among f_e, f_i, f_b, f_p (from Eq. 18) that contributes to the rise in inflammation behaves as cytokines for RICs. These cells interact to fight against the disturbance to bring back the homeostatic balance in the robot. The movement of immune cells towards the fault is visualized in Fig. 2. At the same time, the immature dendritic cells mature to semi-mature dendritic cells. The innate immunity may sometimes fail to regain internal equilibrium due to the fact that RICs do not provide specific solutions but are generic similar to their biological counterparts i.e. monocytes, neutrophils, macrophages, etc.

This failure stimulates maturation of dendritic cells into mature dendritic cells. These mature DCs will then stimulate cytotoxic T-cells present at the adaptive immunity level of the RIS and invoke the proliferation (cloning) of specific T-cells (as tabulated in Table 2). The function among f_e, f_i, f_b, f_p (from Eq. 18), similar to its role in choosing RICs, helps to decide which T-cells are needed to be cloned. The cloned

army of T-cells provides specific solutions for the corresponding faults to bring back the homeostatic balance in the robot. For instance, if the robot hits the obstacle from front due to its sensor failure then $TC_{backward}$ is stimulated and proliferate to move the robot in backward direction. Similarly, if the left motor overheats then $TC_{cool-motor-left}$ is stimulated to switch on the locally placed cooling fan.

This recovery leads to normalization of PS thus inflammation level decreases to normal and regulation of T-cells starts. This regulation is accomplished by reducing the population of cloned T-cells to a minimum level, as the army is no more needed. This regulation ensures internal stability and maintenance of the homeostatic balance in the robot. Some of the T-cells of the cloned army add to the T-cells population in the form of memory T-cells. The memory cells help the RIS manage the same problem if it re-occurs by making the system increasingly adaptive over time. The pseudocode of the RIS, presented earlier, implements this methodology. The developed RIS, in this manner, enables the robot to be cognizant of and make efforts to maintain internal health when facing a potential fault before the unavoidable degradation of its performance.

In short, the RIS detects faults in the robot and then, contingent on the nature of the fault, suggests different solutions to maintain the internal equilibrium of the robot at two immunological levels. The recovery concept of robots' healing system and regaining of normal performance level is derived from the biological system which maintains



Fig. 8. Results showing the comparison of the robot’s health without (a) and with (b) RIS in the case of abnormal health parameters. (a) illustrates the continuous increase in inflammation due to abnormal health parameters (high energy consumption/fast batteries depletion and overheating of components). (b) illustrates the immuno-response against abnormalities in health parameters (high energy consumption/fast batteries depletion and overheating of components) to bring back the homeostatic balance.

the body function in case of any viral attack e.g., flu. Properties like multilayered protection, distributed detections, matching strategies, selection, etc. of the BIS are remarkable and highly compelling, and at the same time may be employed to improve the designs of self-healing systems, such as the one presented herein.

4. Simulation and results

The RIS was implemented on the robot in order to test its ability to detect faults and recover while maintaining the homeostatic balance. The faults were artificially induced and 120 simulations were

performed in different scenarios and conditions to test the effectiveness of the RIS.

4.1. Robot description

A multilayered differential drive wheeled robot in a robot simulator was developed. We have designed the architecture of our robot to have three layers: Layer- α (actuation layer), Layer- β (sensing layer), and Layer- γ (computational layer) as partially illustrated in Fig. 3. The simulated robot is 50 cm x 50 cm in size, octagonal in shape, equipped with 8 bump sensors (one on each facing edge of the sensory layer of

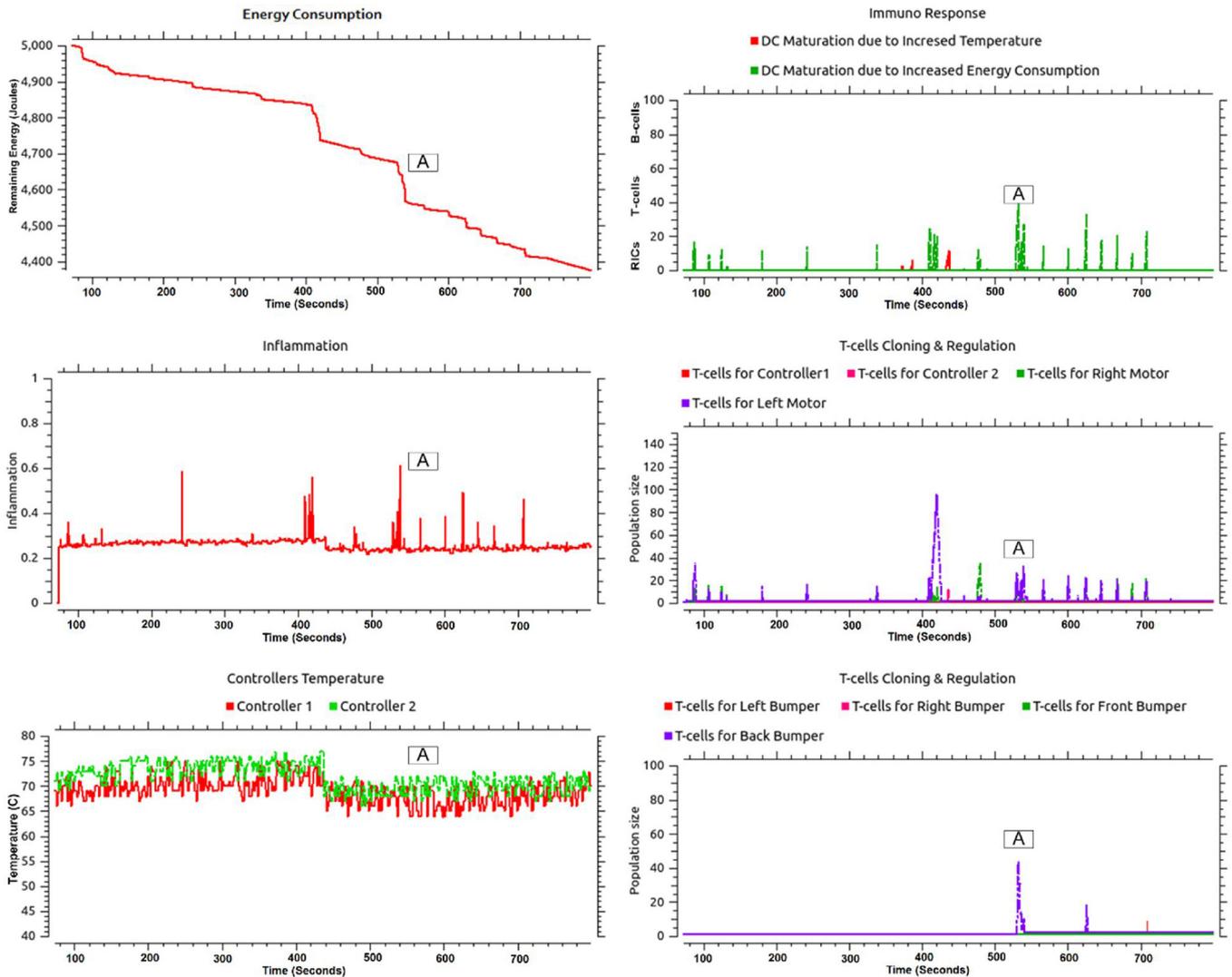


Fig. 9. Results illustrating the immuno-response against abnormalities in health parameters to bring back the homeostatic balance in scenario 2.

the robot), a laser scanner, and a gripper. The laser scanner is used to map the environment. The robot can localize itself in the saved map of the environment as well as perform simultaneous localization and mapping (SLAM). All the three layers are connected with each other as the computational layer receives information from the sensory layer and sends information to control the actuation layer.

4.2. Working environment

4.2.1. Virtual robot experimentation platform (V-REP)

V-REP (Rohmer et al., 2013) was used to develop the robot. V-REP has grown to be a robust and broadly used robot simulator and controller. Its capabilities range from system verification, algorithm optimization, and simulation of complex assembly chains in factory automation applications to robot task planning and control. V-REP supports five programming techniques i.e. embedded script, add on, plug in, remote API client, and ROS nodes.

4.2.2. Robot operating system (ROS)

ROS (Koubaa, 2017) is used for the control and execution of the developed RIS. It provides services to operate robots and has about 2000 ROS packages, which have different functionalities, in the global ROS package repository. This helps the researchers to quickly develop robotic system collaboratively. The collaborative approach avoids the

need to repeat the work of others while creating newer research applications. ROS uses independent nodes to execute specific tasks through message carriers called topics. These nodes and topics have standardized protocols allowing easy incorporation of nodes and or topics of other researchers. The information regarding ROS distribution and pre-existing ROS packages used for the implementation of the RIS is given in Table 3.

4.2.3. ROS Package ris_v1

In the research presented herein, authors have created a new ROS package ris_v1 (the package has been made publicly available²) to design and implement the RIS. It also invokes some existing packages, as listed in Table 3, for assistance. The package contains multiple subscribing and publishing ROS nodes. The ROS-graph illustrating the nodes of ris_v1 running simultaneously, publishing and subscribing different topics at one instance, is shown in Fig. 4. The ellipses represent nodes and rectangles represent topics. Publishing nodes publish messages on different topics, listened by subscribing nodes using call-back functions, for the execution of necessary actions. An example is the value of inflammation published by the inflammation node and listened by the reactive immune cells, dendritic cells and T-cells nodes.

² Please see <https://github.com/mariaakram/robot-immune-system-v1>

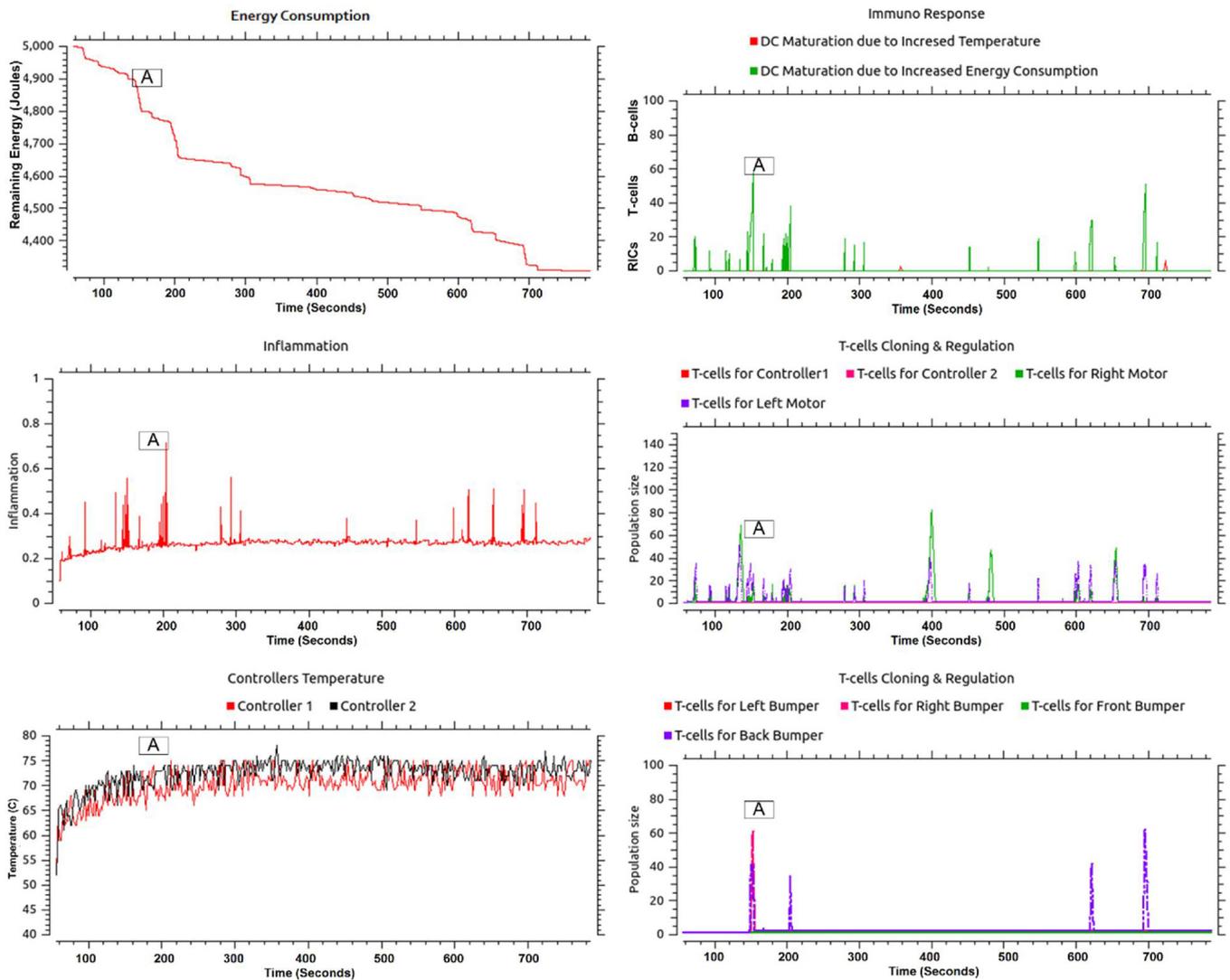


Fig. 10. Results illustrating the immuno-response against abnormalities in health parameters to bring back the Homeostatic balance in Scenario 3.

In case of rise in inflammation, each of these nodes will check the underlying reason. The corresponding node will then publish some actions that help other subscribing nodes in bringing back the homeostatic balance.

4.3. Tasks

The robot developed herein, separately or collectively, executes three major tasks: wandering, mapping, and goal seeking. For each of these tasks, three different and increasingly-complex scenarios are used. The complexity is added in terms of number of obstacles and free space available for the robot to move. Fig. 5 illustrates the simulation environment in V-REP and different scenarios used to test the RIS. A total of 120 simulations were performed to test the RIS wherein 40 simulations were performed in each of the three different scenarios. For each scenario, we selected four different initial positions, and then the robot was simulated 10 times from each of the four initial positions. Fig. 6 showcases the robot seeking the goal in V-REP (right) and RViz (left) environments. The highlighted area in RViz tool is the map (of the robot’s environment) shown in V-REP created for localization and navigation (Zaman et al., 2011). Once the mapping and localization are successfully done then the robot starts navigating, using move_base node, in the environment for goal seeking. The red arrow in the Fig. 6 points towards the goal location. The move_base package maintains the

global map, using A* search and the local map using the dynamic window approach, to seek goal while avoiding obstacles.

4.4. Testing

The robot was initially tasked to wander in a 2.5 m x 5 m arena with different obstacles as shown in Fig. 5, in one of the three scenarios. Each simulation was then repeated ten times for four different initial position in each of the three scenarios thus generating 120 simulations in total as elaborated earlier. This design of simulations allows a thorough analysis of the algorithm, which should be independent of the robot starting location. During testing, the key objective was to test the RIS against various faults, as illustrated in Table 4, and not the task completion. The task completion is assumed to be the natural outcome of the effective implementation of RIS. Some faults (e.g., scanner failure) were induced at the start of the simulation. However, other faults were incurred due to the instantaneous experience of the robot in the arena and may get invoked at any time when the condition is fulfilled e.g. Overheating of Components, Fast Battery Depletion. The fixed parameters for the simulation are displayed in Table 4.

4.5. Results

In this section, the performance of the developed RIS in different

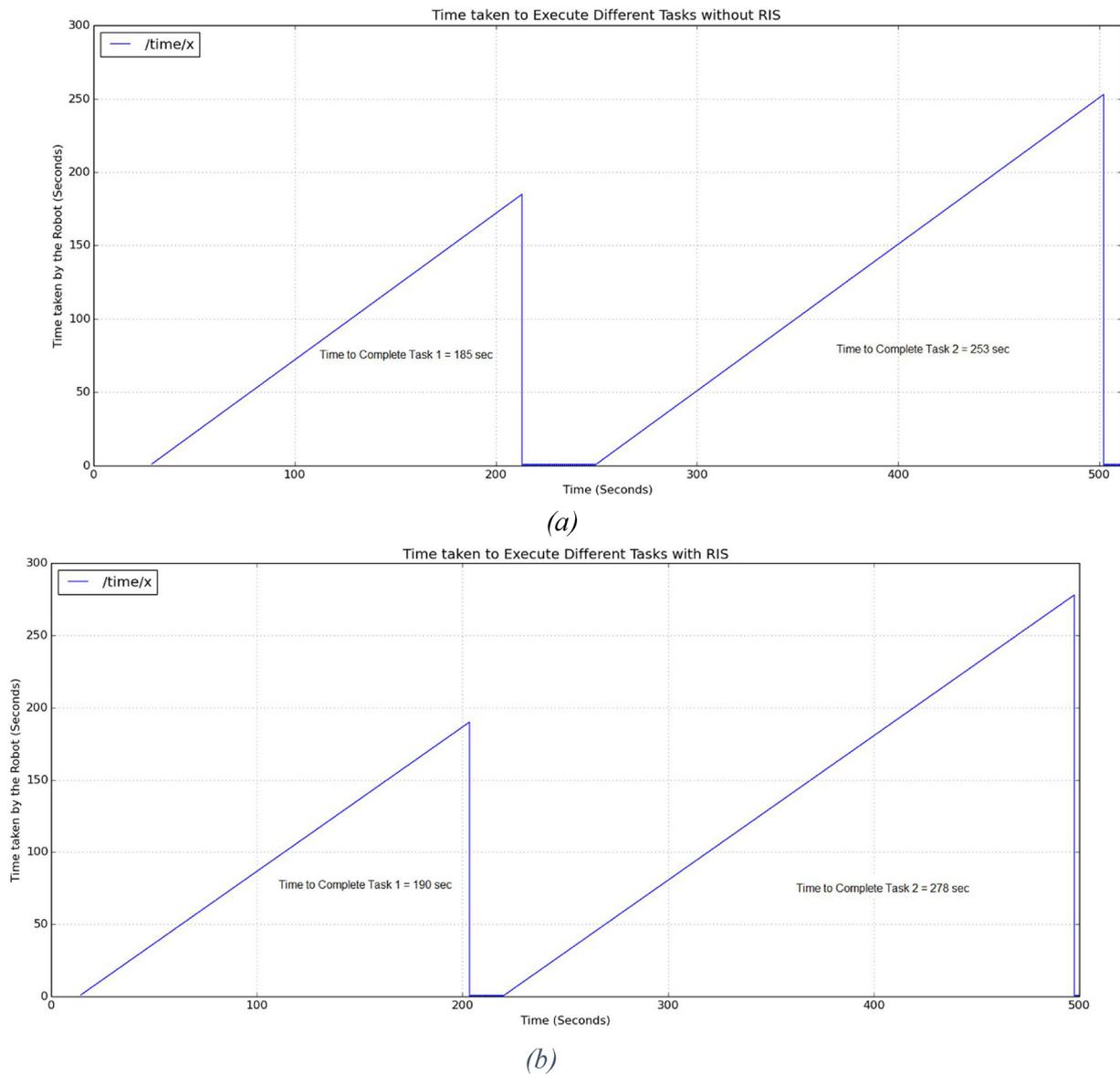


Fig. 11. Results illustrating the minimal time difference in completing individual tasks with and without RIS. (a) showcase the time taken to complete Task 1 and Task 2 without RIS and (b) showcases the time taken to complete Task 1 and Task 2 with RIS.

scenarios is presented and evaluated. The authors first evaluated the RIS’ ability to detect abnormalities in the robot’s behavior using the health indicators. The second evaluation parameter was the RIS’ success to regain the homeostatic balance using recovery actions proposed by the innate and adaptive immune cells. Overall, there are three faults that are considered in this research: sensor failure (F1), battery drain (F2), and heating up of components (F3). The results of the experiments are discussed here.

4.5.1. Scenario S1

This is the simplest of the three scenarios (as shown in Fig. 5 (b)). The corresponding simulation results are presented in Fig. 7. The induced-faults were scanner-failure and overheating of the components. The first column presents the data of health indicators (i.e. energy consumption rate and the temperature of motors and controllers) and the inflammation level. The second column illustrates the immuno-response of the RIS indicating involvement of innate and adaptive immune functionalities that maintain the homeostatic balance. The cloning and regulation of T-cells is also presented under the immuno-response graph for temperature and energy respectively.

The results exhibit the ability of the RIS to recover from faults F1, F2 and F3 by observing the abnormal health parameters and subsequent maintenance of the homeostatic balance. It can also be observed from immuno-response graph that the innate-immunity component works most of the times. For example, it can be observed in Fig. 7 that the adaptive immune functions were called only once out of six instances where the energy depletion rate was faster. For the remaining five instances the innate immune functions were sufficient to handle the problem. The RIS also successfully maintained the temperature using RICs and T-cells. A comparison of the robot’s internal health between RIS and without-RIS approaches is presented in Fig. 8. It can be observed in column (a) representing ‘without RIS’ that abnormal values of the energy-consumption and temperature keep on increasing the inflammation level representing the unhealthy status of the robot, whereas the column (b) representing ‘with RIS’ clearly exhibits the successful maintenance of inflammation levels.

4.5.2. Scenario S2

This is a two-room scenario (as shown in Fig. 5(c)) where the rooms share a common door and have multiple obstacles. Fig. 9 presents the

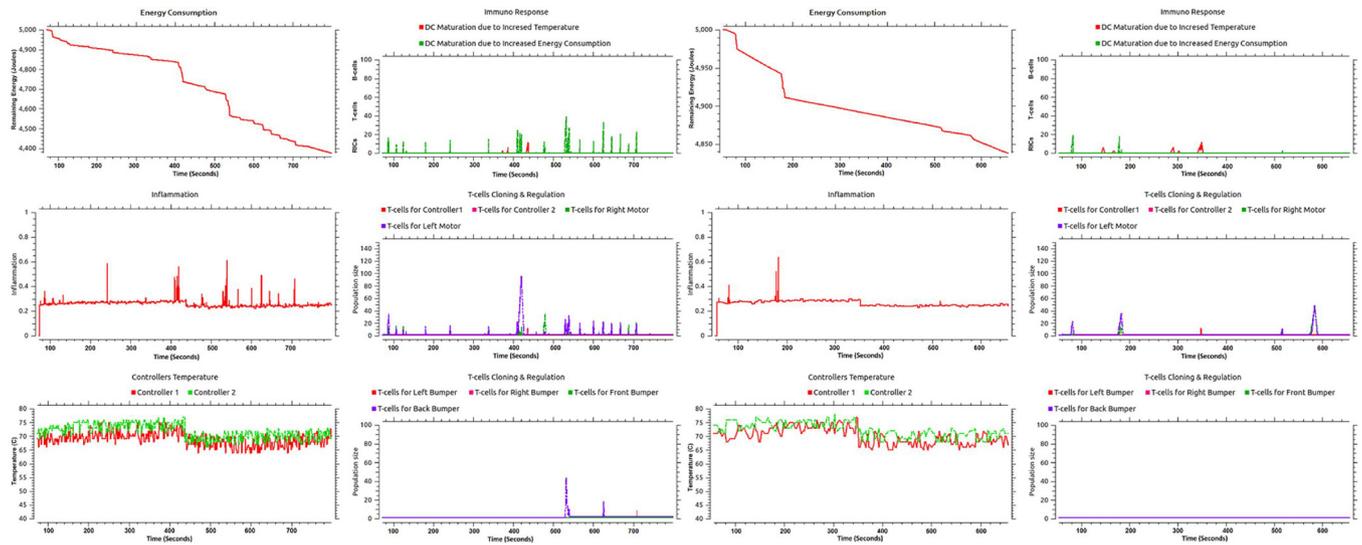


Fig. 12. Results illustrating the Non-deterministic Behaviour of RIS approach. (a) and (b) illustrates the Results of RIS by Repeating the Same Experiment in Scenario 2.

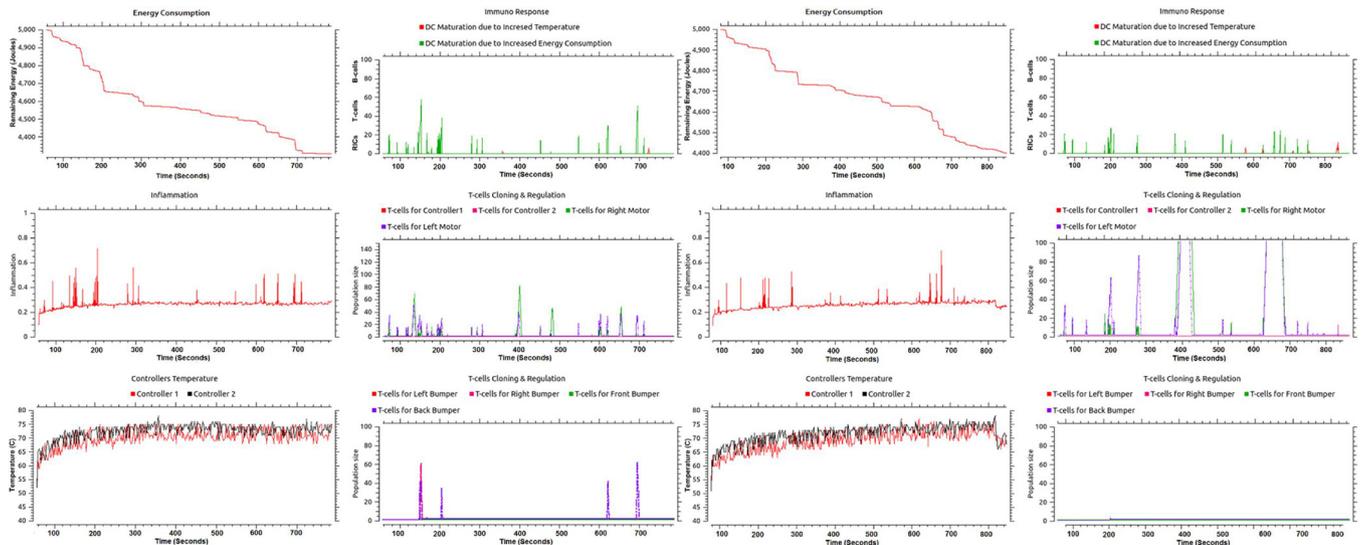


Fig. 13. Results illustrating the Non-deterministic Behaviour of RIS approach. (a) and (b) illustrates the Results of RIS by Repeating the Same Experiment in Scenario 3.

simulation results against the same faults (i.e., F1, F2, and F3) and proliferation (cloning) of T-cells to regulate the homeostatic balance. It can be seen from the graphs of inflammation level and health indicators that the RIS successfully regains the homeostatic balance after experiencing these faults thus indicating the ability of RIS to recover. In addition, the invocation of adaptive immune system is relatively more than the previous scenario. Finally, it can be seen from the energy-consumption graph that the RIS succeeds in arresting the energy drop to avoid fast depletion of batteries. This may enable the robot to operate for longer hours.

4.5.3. Scenario S3

This is the most complex scenario (as illustrated in Fig. 5(d)) in terms of number of obstacles and available space for the robot to move around. Regardless of the complexity, the RIS successfully maintained the homeostatic balance as shown in Fig. 10. Although the robot frequently experiences problem during its activity, as observed in the immuno-response graph, the recovery actions of the RIS enable the robot to recover from them and regain the homeostatic balance. It is evident from all the three scenarios and repeated simulations that the

RIS is successful in terms of handling the faults emanating from scanner failure, high depletion rate of batteries and elevated temperatures.

5. Discussion

The findings indicate that the developed RIS has successfully performed in all the scenarios and faults. For example, when we introduced the sensor-failure, to test the effectiveness of the algorithm, the robot experienced some bumps into the walls. This increased the inflammation level due to the unusual increase in the energy-consumption. Since the motors delivered more torque, this also raised the motors' temperatures as labelled A in Figs. 7, 9 and 10. The instances where the robot hits any obstacle, due to scanner failure, can be observed as steep slopes in the energy-consumption graph. So, the scanner failure is detected by its symptoms i.e., high energy consumption due to the occurrence of robot bumps. In immunological terms this is non-self. The peaks in the inflammation graph indicate detection of abnormality in the robot. The inflammation-rise then mobilizes the RICs of the innate immune system towards the fault location for recovery actions. These effects can be seen in the Figs. 7, 9 and 10 at this particular instance.

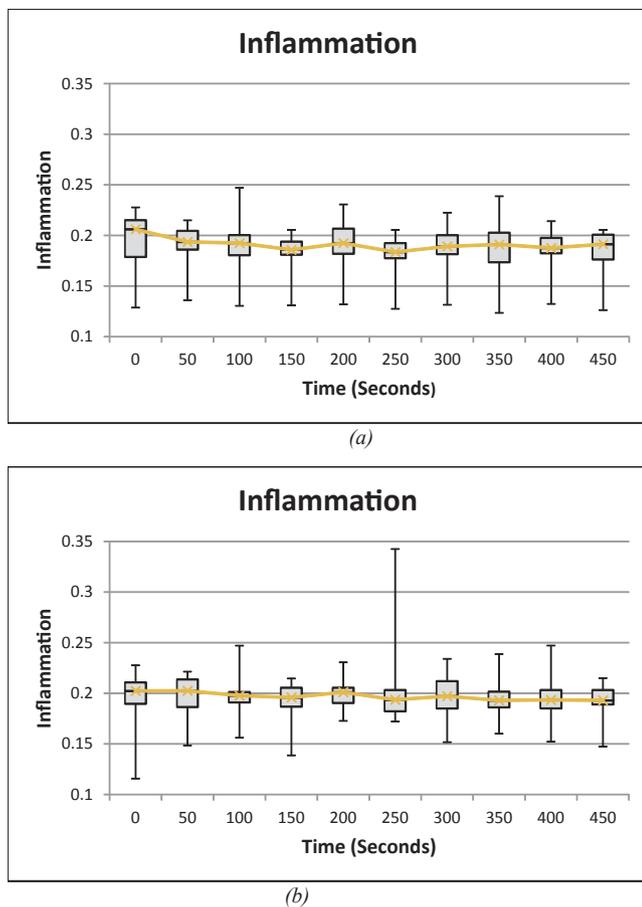


Fig. 14. Boxplots showing the comparison between the inflammation variations as a function of time for 10 experiments and 25 experiments in a sensor failure mode to show the consistency analysis. The line joining the centers of boxes represent the mean inflammation level, the top line represents the highest inflammation level and the bottom line represents the lowest inflammation level during these set of experiments.

Moreover, it can be seen from the graphs that the innate part was able to solve the problem most of the times. However, sometimes the abnormal health indicators result in the maturation of dendritic cells which in turn activate the cytotoxic T-cells of the adaptive immune system. This happens when the recovery actions of RICs fail to maintain the homeostatic balance within a pre-defined time-period. The recovery actions, performed by the cloned army of the specific T-cells (among the existing T-cells tabulated in Table 2), then bring the robot back to the homeostatic level. This multi-tier structure emerging from BIS, also suggested by Raza and Fernández (2018), provides another level of protection. Once the system has recovered, the population of T-cells is regulated back to the minimum required level as shown in the graph of T-cells cloning and regulation in Figs. 7, 9 and 10.

Table 5
Results for Evaluating the Performance Indicators of the Proposed Approach against Faults Detection, Recovery and Task Completion.

Faults	Fault Detection (Out of 120 Simulations)		Recovery (Maintaining Homeostatic Balance) (Out of 120 Simulations)			Task Completion (Out of 120 Simulations)
	Detection Rate	False alarm rate	Temperature (within range)	Energy Consumption rate (slow depletion)	Inflammation level (Maintenance)	Wandering, Mapping, Goal Seeking
Scanner failure (F1)	96.07%	0.127%	100%	100%	100%	N/A
Heating up of Components (F2)	99.9%	0.01%	98%	100%	100%	100%
Fast Battery drain (F3)	99.9%	0.01%	100%	100%	100%	100%

It is noteworthy that although in the current scenarios the RIS is exhibiting two tiers of immunity to successfully maintain the robot's health. A more complex scenario may demand third tier of immunity level i.e., the B-cells of adaptive immunity. At third tier, the immune cells of RIS will produce antibodies to adapt to the failures to maintain the robot's health. This is the most sophisticated tier of immunity where we can benefit most out of the adaptive capabilities of natural immune system to recover from failures. Moreover, the simulation results also seem to suggest that the difference between processing times for RIS-based robot and non-RIS-based robot is minimal. This can be seen in Fig. 11 which compares the two times. The time-addition in completing different tasks with RIS, in the presence of Fault F3, ranges from 2%–10% as compared to the time taken without RIS. Thus, it is plausible to suggest that implementation of the RIS in mobile robots involved in time-critical tasks does not induce a significant lag.

Since the immunological approach devised herein is population-based and non-deterministic, repeated experiments produce slightly different results. This is illustrated in Fig. 12, where an experiment was repeated with same initial conditions in scenario S2. The RIS successfully accomplished the task both the times but with slight differences. Similarly, another experiment was repeated in scenario S3 with same initial conditions and the corresponding comparison is presented in Fig. 13. This non-deterministic behavior is further illustrated in Fig. 14 which presents the variation in inflammation when an experiment was repeated multiple times in the scenario S2. The figure illustrates the maintenance of inflammation level - and thus the homeostatic levels - in these repeated experiments. The figure also illustrates a consistency analysis by comparing the results for inflammation values of an experiment consisting of 10 simulations with an experiment consisting of 25 simulations with the same initial conditions. The results are consistent with a slight improvement in the case of 25 simulations thus lending more endorsement to the RIS.

The performance measures used to evaluate the performance of our RIS are: (1) detection rate which depends on the correct and incorrect rejection of faults, (2) false alarm rate which depends on the incorrect identification of fault, (3) homeostatic balance which depends on limiting the energy-consumption rate, keeping the temperatures of controllers and motors low, and maintaining healthy inflammation level, (4) task completion which depends on how efficiently the robot performs its task in the presence of some faults.

For all the experiments in this research, the performance of the RIS was recorded. It was then translated in terms of above mentioned performance metrics as tabulated in Table 5. The table clearly illustrates that the RIS has performed efficiently to detect the induced faults and recover from them while maintaining a healthy homeostatic balance. It can also be observed that the false alarm rate is low. The false alarm instances were only due to unusual increase in robot speed resulting in fast batteries depletion. These instances were falsely considered as fault and the RIS was able to recover by bringing the speed within the normal range. But generally, the RIS was able to successfully detect the faults.

Table 5 also indicates the 100% success in recovering from faults

except fault F2 where the recovery rate is at 98% level. The corresponding instances which prevented the RIS to bring the temperature of motors within range were when the robot was in contact with the wall at very low speeds. But other than that, the RIS was able to bring the temperature within the range, manage the low energy-consumption rate, and maintain healthy levels of inflammation. Moreover, the RIS was able to complete the task in the cases of fault F2 and F3. Task completion in the case of fault F1 is not applicable because the scanner is necessarily required for localization/mapping. Successful recovery in case of repeated experiments, in a multitude of different scenarios, points towards the reliability of the system. Thus, the RIS designed and tested herein has a potential to serve as a foundation of a complete RIS encompassing a variety of robot faults.

6. Conclusions

In this paper, an immuno-inspired fault detection and recovery system, namely the robot immune system, is successfully developed in ROS and applied to a simulated robot in V-REP. The authors established that the system is able to detect the faults as well as to recover from them which is in contrast with the existing approaches limited only to fault-detection. The results generated from extensive experimentation conclude that the robot immune system successfully detects 98% of the faults across all the increasingly difficult test-scenarios with only 0.128% false alarm rate. Moreover, the robot successfully recovers from multiple faults as indicated by its 98% ability to keep the temperature in acceptable range, 100% ability to limit the energy-drop and 100% maintenance of the inflammation level. Moreover, it is also highlighted that the RIS is a population-based, non-deterministic, and multi-tiered system.

The performance of RIS against the faults in increasingly complex scenarios provides evidence that the system has the potential to be successfully applied to a real robot in future research. Additionally, future research may incorporate more faults (e.g., sensor/actuator failure) and add another layer of B-cell-functionalities in the RIS.

References

- Abbas, A.K., Lichtman, A.H., Pillai, S., 2014. *Basic Immunology: Functions and Disorders of the Immune System*. Elsevier Health Sciences.
- Ahmed, I., Marhaban, M.H., Ishak, A.J., 2015. Energy consumption analysis procedure for robotic applications in different task motion. IOP Conference Series: Materials Science and Engineering. IOP Publishing.
- Bakdi, A., et al., 2017. Optimal path planning and execution for mobile robots using genetic algorithm and adaptive fuzzy-logic control. *Rob. Auton. Syst.* 89, 95–109.
- Bayar, N., et al., 2015. Fault detection, diagnosis and recovery using artificial immune systems: a review. *Eng. Appl. Artif. Intell.* 46, 43–57.
- Behringer, R., et al., 2005. Rascal-an autonomous ground vehicle for desert driving in the Darpa grand challenge 2005. *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE*.
- Bjerknes, J.D., Winfield, A.F., 2013. *On fault tolerance and scalability of swarm robotic systems. Distributed Autonomous Robotic Systems*. Springer, pp. 431–444.
- Canham, R., Jackson, A.H., Tyrrell, A., 2003. Robot error detection using an artificial immune system. *Evolvable Hardware, 2003. Proceedings. NASA/DoD Conference on. CAPRA, J.D., et al., 1999. Immunobiology: the Immune System in Health and Disease*. Garland Publishing.
- Carlson, J., Murphy, R.R., 2003. Reliability analysis of mobile robots. *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*.
- Chatzilygeroudis, K., Vassiliades, V., Mouret, J.-B., 2018. Reset-free trial-and-error learning for robot damage recovery. *Rob. Auton. Syst.* 100, 236–250.
- Crestani, D., Godary-Dejean, K., Lapiere, L., 2015. Enhancing fault tolerance of autonomous mobile robots. *Rob. Auton. Syst.* 68, 140–155.
- d Castro, L.R., Timmis, J., 2002. *Artificial Immune Systems: A New Computational Intelligence Paradigm*.
- Dasgupta, D., 2006. Advances in artificial immune systems. *IEEE Comput. Intell. Mag.* 1 (4), 40–49.
- Eslami, A., 2018. A real-time global optimal path planning for mobile robot in dynamic environment based on artificial immune approach. *GSTF J. Comput. (JoC)* 2 (1).
- Fernandez-Leon, J.A., Acosta, G.G., Mayosky, M.A., 2011. From network-to-antibody robustness in a bio-inspired immune system. *Biosystems* 104 (2-3), 109–117.
- Ferrari, R.M., Parisini, T., Polycarpou, M.M., 2012. Distributed fault detection and isolation of large-scale discrete-time nonlinear systems: an adaptive approximation approach. *IEEE Trans. Automat. Control* 57 (2), 275–290.
- Hart, E., Timmis, J., 2008. Application areas of AIS: the past, the present and the future. *Appl. Soft Comput.* 8 (1), 191–201.
- Huang, S.N., Tan, K.K., 2008. Fault detection, isolation, and accommodation control in robotic systems. *IEEE Trans. Autom. Sci. Eng.* 5 (3), 480–489.
- Ishida, Y., Adachi, N., 1996. An immune algorithm for multiagent: application to adaptive noise neutralization. *Intelligent Robots and Systems' 96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on*.
- Ismail, A.R., Timmis, J., 2010. Towards self-healing swarm robotic systems inspired by granuloma formation. *Engineering of Complex Computer Systems (ICECCS), 2010 15th IEEE International Conference on*.
- Jakimovski, B., Maehle, E., 2008. Artificial immune system based robot anomaly detection engine for fault tolerant robots. *International Conference on Autonomic and Trusted Computing*. Springer.
- Jiao, W., Li, Q., 2018. Anomaly detection based on fuzzy rules. *Int. J. Perform. Eng.* 14 (2) p. 376.
- Jing, C.S., et al., 2017. Fault detection and isolation for complex system. *AIP Conference Proceedings*. AIP Publishing.
- Johnson, A.P., et al., 2018. Homeostatic fault tolerance in spiking neural networks: a dynamic hardware perspective. *IEEE Trans. Circuits Syst. I Regul. Pap.* 65 (2), 687–699.
- Kamel, M.A., Yu, X., Zhang, Y., 2018. Fault-tolerant cooperative control design of multiple wheeled mobile robots. *IEEE Trans. Control. Syst. Technol.* 26 (2), 756–764.
- Kernbach, S., et al., 2009. Towards self-adaptation of robot organisms with a high developmental plasticity. *Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns, 2009. COMPUTATIONWORLD'09. Computation World*.
- Khadidos, A., Crowder, R.M., Chappell, P.H., 2015. Exogenous fault detection and recovery for swarm robotics. *IFAC Papers OnLine* 48 (3), 2405–2410.
- Khalastchi, E., Kalech, M., 2018. On fault detection and diagnosis in robotic systems. *ACM Comput. Surv. (CSUR)* 51 (1) p. 9.
- Khan, M.T., De Silva, C., 2009. Immune system-inspired dynamic multi-robot coordination. *ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*.
- Kim, J., et al., 2007. Immune system approaches to intrusion detection—a review. *Nat. Comput.* 6 (4), 413–466.
- Koubaa, A., 2017. *Robot Operating System (ROS)*. Springer.
- Laurentys, C., Palhares, R.M., Caminhas, W.M., 2010. Design of an artificial immune system based on danger model for fault detection. *Expert Syst. Appl.* 37 (7), 5145–5152.
- Medzhitov, R., Janeway, C.A., 1997. Innate immunity: the virtues of a nonclonal system of recognition. *Cell* 91 (3), 295–298.
- Medzhitov, R., Janeway Jr, C.A., 1998. Innate immune recognition and control of adaptive immune responses. *Seminars in Immunology*. Elsevier.
- Mei, Y., et al., 2005. A case study of mobile robot's energy consumption and conservation techniques. *Advanced Robotics, 2005. ICAR'05. Proceedings., 12th International Conference on*.
- Mondada, F., Franzi, E., Guignard, A., 1999. The development of khepera. *Experiments With the Mini-Robot Khepera, Proceedings of the First International Khepera Workshop*.
- Raza, A., Fernandez, B.R., 2015. Immuno-inspired robotic applications: a review. *Appl. Soft Comput.* 37, 490–505.
- Raza, Ali, Fernández, Benito R., 2018. A multi-tier immuno-inspired framework for heterogeneous mobile robotic systems. *Appl. Soft Comput.* 71, 333–352.
- Rohmer, E., Singh, S.P., Freese, M., 2013. V-REP: a versatile and scalable robot simulation framework. *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*.
- Tarapore, D., et al., 2015. To err is robotic, to tolerate immunological: fault detection in multirobot systems. *Bioinspir. Biomim.* 10 (1) p. 016014.
- Tarapore, D., Christensen, A.L., Timmis, J., 2017. Generic, scalable and decentralized fault detection for robot swarms. *PLoS One* 12 (8) p. e0182058.
- Timmis, J., Neal, M., Hunt, J., 2000. An artificial immune system for data analysis. *Biosystems* 55 (1-3), 143–150.
- Timmis, J., Andrews, P., Hart, E., 2010. On artificial immune systems and swarm intelligence. *Swarm Intell.* 4 (4), 247–273.
- Timmis, J., et al., 2016. An immune-inspired swarm aggregation algorithm for self-healing swarm robotic systems. *Biosystems* 146, 60–76.
- Trivun, D., Dindo, H., Lacevic, B., 2017. Resilient hexapod robot. *2017 XXVI International Conference on Information, Communication and Automation Technologies (ICAT)*.
- Van Parijs, L., Abbas, A.K., 1998. Homeostasis and self-tolerance in the immune system: turning lymphocytes off. *Science* 280 (5361), 243–248.
- Watanabe, Y., Jyo, E., 2014. Analogy between immune system and sensor replacement using mobile robots on wireless sensor networks. *Procedia Comput. Sci.* 35, 1352–1359.
- Wu, J., et al., 2016. Sode: self-adaptive one-dependence estimators for classification. *Pattern Recognit.* 51, 358–377.
- Zaman, S., Slany, W., Steinbauer, G., 2011. ROS-based mapping, localization and autonomous navigation using a Pioneer 3-DX robot and their relevant issues. *Electronics, Communications and Photonics Conference (SIECP), 2011 Saudi International*.