

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



# In situ self-reconfiguration of hexapod robot OSCAR using biologically inspired approaches

Bojan Jakimovski and Erik Maehle

*Institute of Computer Engineering, University Lübeck  
Germany*

## 1. Introduction

A highly desirable feature for next generation robots operating on terrestrial or extraterrestrial environment (Chien et al., 2006) is that they possess the property of sustainable autonomic systems enveloping the self-management and self-x characteristics such: self-reconfiguration, self-optimization, and self-healing. Responses taken automatically by a system without real-time human intervention are called autonomic responses (Sterritt et al., 2006; Lewandowski et al., 2006). The self-x properties will enable the robot to continue with its mission tasks even in the cases when the robot has some faults within the system. The robot shall be able to reconfigure itself and continue with its mission tasks.

The autonomic concept was introduced with the IBM Manifesto for Autonomic Computing (IBM, 2001). This proposed several key elements important for the autonomic systems: self-configuration, self-healing, self-optimization, and self-protection which were inspired by the human body's autonomic nervous system. Complementary to the IBM's initiative, the Organic Computing initiative (DFG, 2004) on the other hand, proposes the means of achieving such self-x properties of the next generation of self-organizing embedded systems, inspired by information processing seen within the biological systems.

Transferring such biologically inspired paradigms (Hinchey & Sterritt, 2007) into computing systems and robots will enable the systems to perform in a more robust, safe, and flexible way. In that context we have been researching towards practically applying biologically inspired methodologies and developing novel procedures for next generations of self-reconfiguring and joint leg walking robots.

The rest of the paper is organized as follows: In the second chapter we give an overview of our hexapod robot prototype OSCAR on which we have conducted the experiments. We also introduce our innovative and patent pending mechanism for in-situ walking robot leg amputation and for robot reconfiguration.

In the third chapter we explain the biologically inspired fault detection method used for fault/anomaly detection. In the fourth chapter we describe the swarm intelligence concept for robot reconfiguration, which is used to perform a stable spatial reconfiguration of the hexapod walking robot. We also present the results from real experiments on self-

reconfiguration performed on the experimental hexapod robot OSCAR using our robot leg amputation mechanism.

## 2. Organic Self-Configuring and Adapting Robot - OSCAR and innovative robot leg amputation mechanism

### 2.1 Self-Configuring and Adapting Robot - OSCAR and innovative robot leg amputation mechanism R-LEGAM

In order to perform real demonstrations on self-reconfiguration of a walking robot, we have built up a new prototype in our series of experimental hexapod robots - named OSCAR - Organic Self-Configuring and Adapting Robot (El Sayed Auf et al., 2006) (Jakimovski et al., 2006). The robot is constructed of a fiberglass framed body, six legs set up spatially in a circle with 60 degrees between each leg, three servos per leg, onboard control electronics, wireless camera, and additional sensors. The robot is shown in Fig. 1.



Fig. 1. OSCAR (Organic Self-Configuring and Adapting Robot)

### 2.2 Innovative robot leg amputation mechanism

In the newly built hexapod robot prototype, we have introduced an innovative mechanism called Robot Leg Amputation Mechanism (R-LEGAM), which is performing on demand robot leg amputation.

The mechanism is built out of electrical connectors which are used for providing the servos with power and also for the control signals for the bus on which the servos are connected. Additionally there are integrated springs in each amputation mechanism and those are used to eject the leg from the robot's body in the process of leg amputation. Every mechanism has an additional small servo that on demand moves small mechanical part that further releases the energy stored within the compressed springs and the robot's leg is ejected.

This mechanism is used in cases for example when the robot experiences some problems in some of its legs and in such circumstances can amputate the malfunctioned legs.

By this the robot does not have to carry on the malfunctioned legs during the rest of its mission. The advantages are that the weight is decreased, the energy from batteries is used effectively, and the operating range of the robot after performing the leg amputation is increased in comparison with having the robot still walking with the malfunctioned legs. On the other hand, the disadvantages are that the stability and the speed of the robot are decreasing with every amputated leg.

The robot's leg with the integrated leg ejection mechanism attached to the robot's body is presented in the Fig. 2.



Fig. 2. OSCAR's leg with integrated leg ejection mechanism attached to the robot's body

When the robot's control architecture finds an anomaly within some leg, and it decides that the leg should be amputated, a signal is sent to the amputation mechanism which then detaches the leg from the robot's body.

In Fig. 3 the robot's leg is represented - detached from the robot's body.



Fig. 3. OSCAR's leg detached from the robot's body after the amputation is performed.

### 3. Robot anomaly detection using artificial immune system based approach - RADE

Fault or anomaly detection is a scope of many projects, especially those related to reliable and safety-critical systems. For fault detection in general and for the domain of autonomous robots various approaches have been considered like: Time-Delay Neural Networks (TDNN) (Christensen et al., 2008), Recurrent Neural Networks (RNN) (Przystalka, 2006), particle filters (Verma & Simmons, 2006; Zhuo-hua et al., 2006). The mentioned approaches are mostly related to the procedure of synthesizing fault detection components based on the collected data in the training runs. After the fault detectors have been synthesized they are evaluated on fault scenarios in real time, or on the recorded data, and the fault positives or fault negatives are estimated.

However, building a full failure model for robotic systems and generating fault detectors for every possible fault scenario that may occur is often an exhaustive process (Haldar & Sarkar, 2006). To overcome this constraint the fault detection mechanism should possess self-adapting and learning properties, just like biological systems have.

In order to make our fault detection more adaptive to the environment and the current behavior of the robot, we have considered using the Artificial Immune System (AIS) paradigm as metaphoric implementation of the natural immune system.

Similar to the natural immune system, AIS also introduces a metric that allows the system to distinguish between self (correctly functioning system) and non-self (anomaly in the system) and also to memorize and detect specific patterns. Therefore main properties of AIS are: recognition, identification, adaptation, self-organization. It has been successfully applied for various domains: pattern recognition (Cao & Dasgupta, 2003), data mining (Nasraoui et al., 2005), network security (Pagnoni & Visconti, 2005), robotics (Michelan & Von Zuben, 2002), (Singh & Nair, 2005), (Sathyanath & Sahin, 2002), (Neal et al., 2006), (Canham et al., 2003), and others. There exist several different approaches (De Castro & Timmis, 2002) for AIS. However, the most commonly found in literature are: negative selection (Forrest et al., 1994), positive selection (Nino & Beltran, 2002), artificial immune networks (AIN) (Galeano et al., 2005) and clonal selection (De Castro & Von Zuben, 2002).

Negative selection mechanism is based on the ability of the immune system to learn to categorize between non-self and self by providing tolerance for the self. The negative selection consists of generating a set of detectors and evaluation of those detectors. Only the detectors that detect non-self, but do not react to self are considered for further detection. The positive selection, on the other hand, generates, evaluates and enables those detectors that can detect only the non-self. The clonal selection within AIS is based on the proliferation mechanism where self, upon recognizing non-self, starts to proliferate by cloning itself and also memorizing the pattern of the non-self (immune memory), so it has better responsiveness for the next encounter with such a particular non-self pattern.

For the detection of an anomaly within robot systems different immune system approaches have been considered, like inflammation (Sathyanath & Sahin, 2002), or usage of negative selection (Neal et al., 2006). For our RADE anomaly detection approach (Jakimovski & Maehle, 2008) we are using the clonal selection method in combination with fuzzy logic for representing the information within the AIS.

Within our robot's control architecture ORCA (Organic Robot Control Architecture) introduced in (Brockmann et al., 2005) and represented in Fig. 4. RADE is situated in monitoring and reasoning units called OCUs (Organic Control Units). OCUs are responsible

for permanently monitoring the health status of the system and for taking suitable counteractions like reconfigurations in case malfunctions are detected. BCUs (Basic Control Units) implement the fault-free behaviors of the robotic system. Depending on its implementation, within the robot’s control model several BCUs as behavioral units, several OCUs and a number of sensors and actuators can be present.

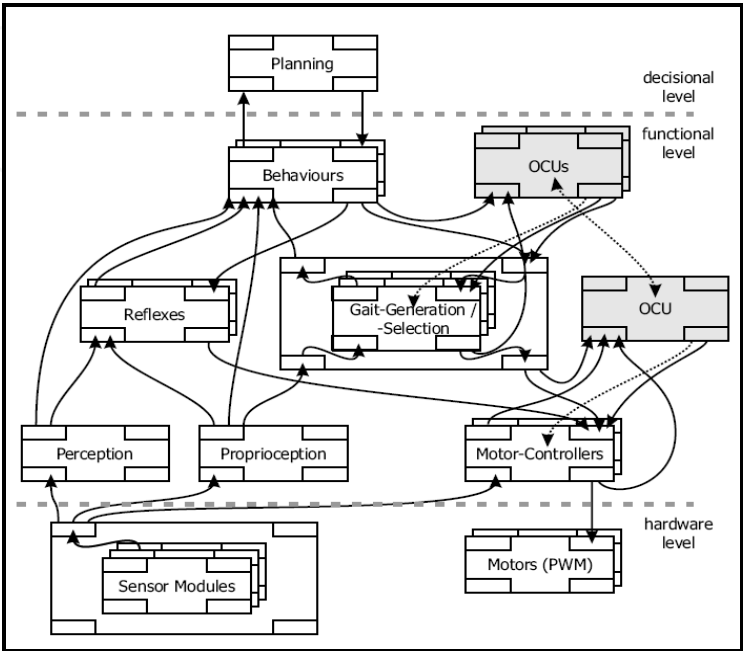


Fig. 4. ORCA (Organic Robot Control Architecture).

RADE uses fuzzy logic for representing information because an exact recognition is not necessary for triggering of an anomaly response. It is also similar to the way the association is made within the immune system, where given a similar (but not necessarily identical) stimulus, the response can be initiated. In a practical implementation, this would mean for example: if the behavior is walking then the acceleration level should not be low. Or in monitoring the servo’s motor status, this can be interpreted as: by normal walking, the servo’s current should not be high. In RADE such generalized self / non-self situations can be defined by fuzzy rules which are part of self / non-self sets. The rules in the non-self set detect when there is some anomaly present within the system, and the rules in the self set detect when the situation is not characterised as anomalous. The rules of both sets have the following structure:

IF X1&X2&...Xn THEN Anomaly is Y WITH weight\_factor Z

(1)

The “X1&X2&...Xn” in (1) represents the premise part which constitutes of monitored behaviours: walking, standing, etc.; and particular some characteristics like: current, acceleration, etc. The “Y” in (1) is the consequence part and can have two types of values: “anomaly is present” or “anomaly is absent”. The weight factor “Z” represents the clonal proliferation within AIS, and is in a range from 0.0 to 1.0. The “Z” value in (1) will increase for some constant value (for example: 0.1) if the rule has “fired”. In parallel to that the



weights will decrease in all the rules belonging to the opposite set, just as the concentration of self/non-self drops being influenced by an increased concentration of non-self/self within the immune system. The firing level of each rule is therefore always adjusted, depending on the value of "Z". The weight factor "Z" has also another positive characteristic for the anomaly detection engine. Namely we want to reduce the factor of hand coded elements in RADE, and let the system dynamically adjust itself to the current situation.

For example, in case we have coded fuzzy rules *without* using weights, depending on the manually pre-designed fuzzy membership sets for the premise parts of the rules, the rules can have an optimal response for some situation and perhaps a not satisfying response for other unforeseen situations. In case the fuzzy rules *have* weights, this would introduce two new features:

- The premise parts of the rules do not require any additional handcrafting and expert designing for their fuzzy membership sets. Therefore they can have some automatically generated "standard" triangular fuzzy membership sets, normally distributed within a valid range for the observed variable. For example such fuzzy membership sets for monitored variable "Current" having values in the range from min 0 to max 3 Amperes can be represented as:

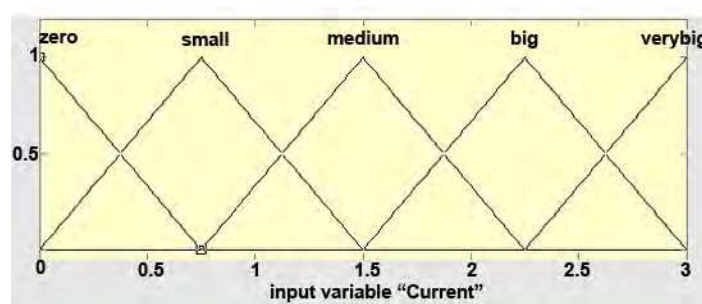


Fig. 5. Fuzzy membership set for monitored variable "Current".

The other membership sets for other monitored variables (for example: acceleration) are going to have the same "standard" triangular fuzzy membership normally distributed for their input range. The nice thing for having such "standard" generated fuzzy membership sets is that they can be part of the learning process, which we plan to introduce in the next step of development of our anomaly detection engine. Having the "standard" fuzzy membership sets for each of the observed parameters, we may build up a rule-based learning system which incorporates only new situations since the fuzzy membership sets for the observed parameters are not going to be changed, and so it will be possible to distinguish between what has been already learned and what can be learned.

- The weight factors for such rules having the "standard" generated fuzzy membership sets allow the rule to adapt to the situation even without the rule being optimally pre-designed at start, i.e. having its membership sets optimally

pre-designed. Therefore the changes of the weights depend on the particular situation and therefore contribute for the dynamics of overall system.

The previously discussed self and non-self rule sets and the dynamical change of their weights can be visually represented as in Figure 6.

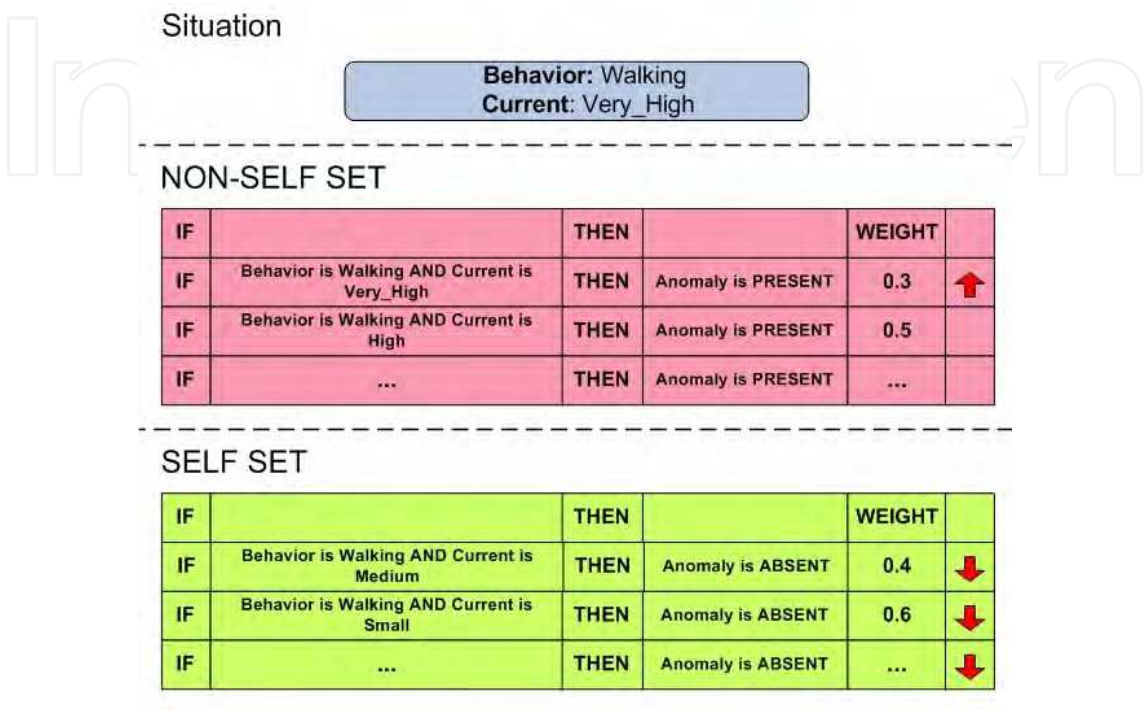


Fig. 6. Functioning principle of RADE and the dynamically changing weights within the self and non-self rule sets (for clearness, here only monitoring parameter “Current” is represented, although additional monitored parameter, like “Acceleration” can be considered as well).

As illustrated in the figure, when the premise within a rule belonging to the non-self set is satisfied, the rule “fires” and its weight is also increased by some constant value, e.g. by 0.1 from 0.3 to 0.4. At the same time, the weights of rules belonging to another set are decreased with the same constant value of 0.1. In such a way they lower their value from 0.4 on 0.3 or from 0.6 on 0.5 and so on.

In every computation step, a weighted output is calculated from such a fuzzy system which contains two membership functions: self and non-self. The value is in the range from 0.0 (no anomaly) to 1.0 (full anomaly) and represents the output of the RADE method. The output of RADE is computed in a defuzzification process as a centroid of fuzzy outputs of the “fired” rules. Therefore the output of RADE is influenced by the weight factor of each of the firing rules. The weight factor acts in the similar way as the secondary and subsequent responses within the immune system, i.e. the more the weight is associated to some rules, the more significant will be the response of those rules to the output of RADE in the next moment of their firing.

The change of weights therefore acts as some sort of short memory for events that occurred some moments ago. The anomaly output level of RADE depends on its short history and



also on actual system's state. Thus, the anomaly detection method RADE autonomously adapts to the situation and generates an appropriate health signal.

Within our experiments RADE method is used to detect a potential anomaly situation within the robot's system and robot's legs and to initiate a leg amputation and robot self-reconfiguration.

## **4 Self Reconfiguration of a joint-leg hexapod robot**

### **4.1 Swarm Intelligence for Robot Reconfiguration (S.I.R.R.) method**

Different approaches have been tried for overcoming the robot's malfunctioning. One of them is swarm robotics. Swarm robotics is a synonym for decentralized autonomous systems built out of many robots which communicate and cooperate with one another in order to accomplish some mission tasks. Due to the high number of entities in such swarm system, swarm robots have a high potential for fault tolerance. Namely, the failure within one or several robots may not compromise the overall system's functionality. (Winfield, 2006) (Hyun Yool et al., 2006).

However, we can also shift the macro view and consider that the robots themselves can be built out of several autonomous entities, where the failure of one or several such entities will not make the robot completely malfunctioned. Modular robots (Xuan et al., 2006) (Hancher & Hornby, 2006) are such an approach, where the robot is built out of many identical functional parts which when combined together and coordinated in the proper way can function like a robust robot. However, it is a still ongoing research on how those separately functional elements can be combined together and coordinated autonomously to function like one robot.

In our project, we have taken yet another direction with different types of granularity of the system. Namely for our walking hexapod robot, we have assumed that the robot legs themselves act like own entities. So we have considered that one robot leg is a functional entity built of several servos which cannot be considered as independent functional units, but only when combined together in one robot leg segment.

Therefore, when taking into account this type of granularity view, we have six legs acting like six independent entities which can be considered further for reconfiguration purposes.

Another part of the research is based on an idea on how to perform the reconfiguration on such functional entities within one hexapod robot with changeable morphology. The approach which we developed is based on the natural phenomena of flocks of birds and schools of fish and is named S.I.R.R. (Swarm Intelligence for Robot Reconfiguration) (Jakimovski et al., 2008). The flocking phenomenon was researched by Craig Reynolds who introduced boids as a distributed behavior model to simulate coordinated movements of the flocking of birds (Reynolds, 1987). Each member of a boid implements three rules and interacts only with its neighboring boid members.

These rules are defined as:

1. Separation rule: the member tries to avoid the crowding with other local boids;
2. Alignment rule: the member moves towards the average heading of the local boids;
3. Cohesion rule: the member moves towards the local average position of the other local boids.

In our experiments we have defined that each robot's leg represents one member of boid and since the robot has intrinsic symmetry with three legs on one side and three on another, the legs can be considered as members of two groups of boids - see Fig. 7 (a). Additionally for performing proper spatial robot reconfiguration we have considered that each boid besides the three above mentioned rules implements an additional two boid rules:

4. Boids are allowed to have  $n$ ,  $n+1$ , or  $n-1$  number of participants in their boid in comparison to the other boid, where  $n$  is the number of entities (legs) in the boid. In case there are  $n+2$  more members in one boid in comparison to another one, the legs which belong to the "overcrowded" boid and that are situated on the edge of the boid, i.e. they have just one other neighbor member in the boid, will be moved to another boid until the "overcrowded" boid's size becomes  $n+1$ .
5. Members (legs) of one boid that join the other boid change their swinging and stance end-position parameters with respect to the parameters characteristic for that boid.

The fifth rule is used for example when boid (leg) changes from one to the other side of the robot's symmetry line.

In Fig. 7 we represent the usage of the above five boid rules used for a potential robot reconfiguration.

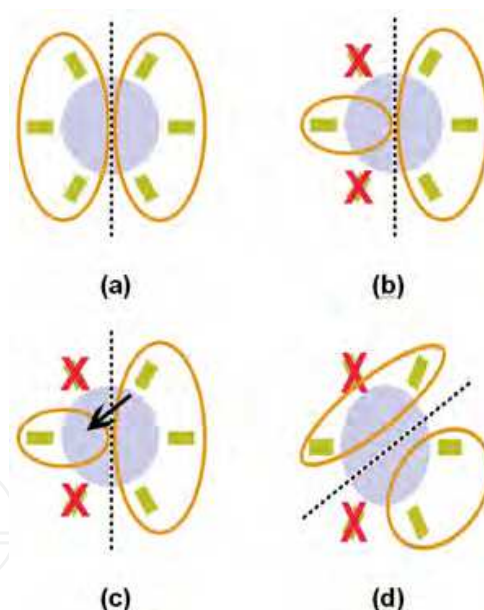


Fig. 7. Illustration of reconfiguration of a robot using the five boid rules within S.I.R.R. (Swarm Intelligence for Robot Reconfiguration) algorithm; (a) legs of the robot forming two boids; (b) two legs of the robot have malfunctioned; (c) member (leg) from "crowded" boid joins another less "crowded" boid; (d) situation after reconfiguration. The dotted line represents the robot's line of symmetry.

Fig. 7 (a) is the initial robot configuration with legs belonging to two boids. Fig. 7 (b) shows the situation when two legs have malfunctioned. Fig. 7 (c) shows that after the two legs have malfunctioned the number of members in the right boid does not equal the number of

members in the left boid. This matches the fourth rule used in our SIRR algorithm and therefore a member from the right boid transfers to the left boid and therefore reconfiguration is performed. Fig. 7 (d) shows the situation after the reconfiguration.

In Fig. 7 (d), we can also see that the robot's line of symmetry changes. This change of robot's symmetry line occurs in emergent way due to the reconfiguration of members in boids that takes place.

After the reconfiguration is performed using the SIRR approach, the robot has a more stable spatial configuration of its legs, since the angles and distance between the legs are more equally distributed. Such configurations enable the robot to have better static and dynamic weight distribution during walking.

In (Jakimovski et al., 2008) we have introduced and explained the Swarm Intelligence for Robot Reconfiguration approach in more detail.

#### 4.2 Simulation - demonstration of self-reconfiguration of hexapod robot OSCAR

In order to validate the practical applicability of the S.I.R.R. approach, we have conducted an initial simulation test scenario. The test scenario considers several robot spatial leg posture reconfigurations, after some legs of the robot have been declared as malfunctioned. For each test scenario, we start with a fully functional configuration of the robot.

The leg numbering of the robot OSCAR is as following (Fig. 8):

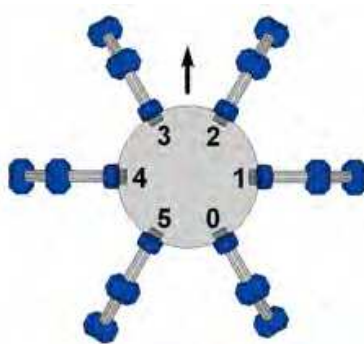


Fig. 8. Model of hexapod robot OSCAR with leg numbering. The arrow is showing the front direction of the robot.

We then assume that some randomly chosen legs have malfunctioned, and the reconfiguration of the robot is performed.

The scenario can be described as follows:

1. leg 0 has malfunctioned; reconfiguration;
2. leg 2 has malfunctioned; reconfiguration;
3. leg 3 has malfunctioned; reconfiguration;
4. leg 4 has malfunctioned; reconfiguration.

When a leg has malfunctioned, we assume that the leg is centered and it is lifted up. Then the reconfiguration takes place.

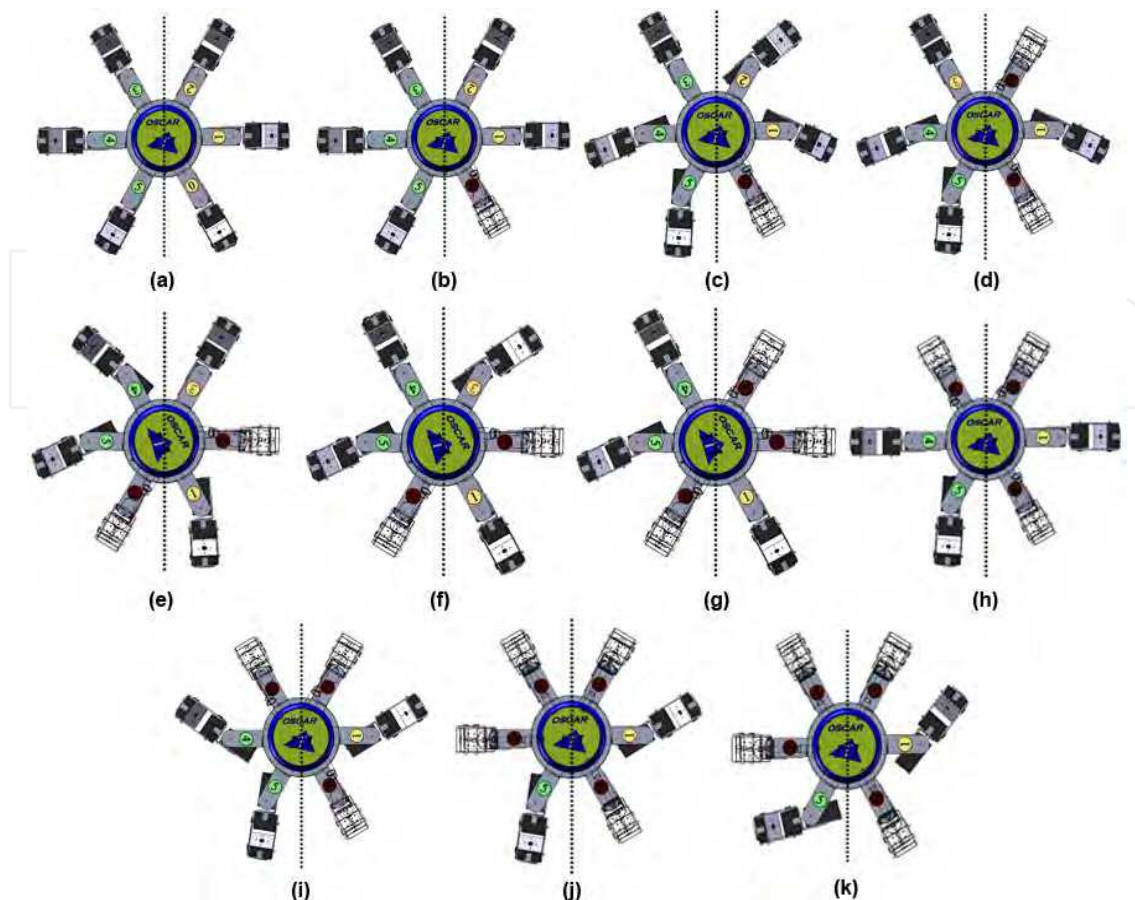


Fig. 8. Test scenario - reconfiguration of a robot using the S.I.R.R. approach; (a) Top model view of fully functional hexapod robot; (b) leg number 0 has malfunctioned; (c) after S.I.R.R. reconfiguration; (d) leg number 2 has malfunctioned; (e) S.I.R.R. reconfiguration with change of symmetry axis; (f) after S.I.R.R. reconfiguration; (g) leg number 3 has malfunctioned; (h) S.I.R.R. reconfiguration with a change of symmetry axis; (i) after S.I.R.R. reconfiguration; (j) leg number 4 has malfunctioned; (k) after S.I.R.R. reconfiguration; the dotted line represents the robot's line of symmetry.

The results of the simulated reconfiguration including the intermediate steps are represented in Fig. 8. The test scenario starts with a fully functional hexapod robot, Fig. 8(a). Then the leg 0 of the robot malfunctions, Fig. 8(b). After this, the S.I.R.R. reconfiguration goes into action and the spatial reconfiguration of the robot's legs is performed. Fig. 8(c) represents the robot's spatial configuration of the legs after the S.I.R.R. reconfiguration was done. After this, leg number 2 malfunctions, Fig. 8(d) S.I.R.R. reconfiguration is performed, and the direction of the axis of symmetry of the robot is adjusted, Fig. 8(e). The result of S.I.R.R. reconfiguration and new spatial arrangement of the legs is represented in Fig. 8(f). In Fig. 8(g), leg number 3 malfunctions; S.I.R.R. reconfiguration is performed, and the direction of the axis of symmetry of the robot is adjusted, Fig. 8(h). Reconfigured robot using S.I.R.R. is shown in Fig. 8(i). In Fig. 8(j), the robot's leg number 4 malfunctions. The final result after robot reconfiguration using S.I.R.R. is shown in Fig. 8(k). Although the algorithm has correctly distributed the legs, however the robot's stability is drastically impaired due to the only two legs configuration.

In comparison with the previous reconfiguration experiment of the robot, the results from the reconfiguration experiment using the S.I.R.R. approach show a better spatial reconfiguration of the robot's legs, in the sense of acquiring stability for the robot when a leg has malfunctioned, and in that way, enabling the robot to continue with its mission tasks, even in cases when it has mechanical failures in its legs.

#### 4.3 Real case demonstration of self-reconfiguration of hexapod robot OSCAR

Initial positive simulation experimental results done with the SIRR method have motivated us to proceed with additional real robot experiments in which the goal is to perform in-situ real time hexapod robot reconfiguration with leg amputations and enable the hexapod robot to continue with its mission despite the malfunctioned legs. For achieving this requirement we have used the already introduced innovative robot leg amputation mechanism which enables the robot on demand to amputate the malfunctioned leg. When the monitoring unit in the robot's architecture detects that there is an anomaly present within the leg, it sends a control signal to ejection mechanism located on the robot's leg to initiate a leg ejection, i.e. to amputate the malfunctioning leg and then after to reconfigure the spatial positioning of the robots legs to

We have conducted the following demonstration scenario and simulation of leg defects:

1. First leg numbered 3 becomes malfunctioned and the robot performs SIRR reconfiguration;
2. Second leg number 1 becomes malfunctioned and the robot performs SIRR reconfiguration;
3. Third leg numbered 5 becomes malfunctioned and the robot performs SIRR reconfiguration;

This is represented in Fig. 9 (a) - (l).

As can be seen in the Fig. 9 (a), the robot starts with the initial six leg configuration. In the first fault case, leg number 3 becomes malfunctioned and the robot control architecture sends a signal to the leg amputation mechanism to amputate the leg number 3. This is shown in Fig. 9 (b)

After that the robot performs self-reconfiguration using the SIRR approach - Fig. 9 (c) and continues with its mission.

In the second fault case, leg number 1 becomes malfunctioned - Fig. 9 (d) and gets amputated - Fig. 9 (e) After that the robot performs self-reconfiguration - Fig. 9 (f) and continues with walking. In Fig. 9 (g) the third leg, number 5 becomes malfunctioned and gets amputated - Fig. 9 (h). After that the robot performs self-reconfiguration using the SIRR approach - Fig. 9 (i) and continues with walking - Fig. 9 (j) - (l).



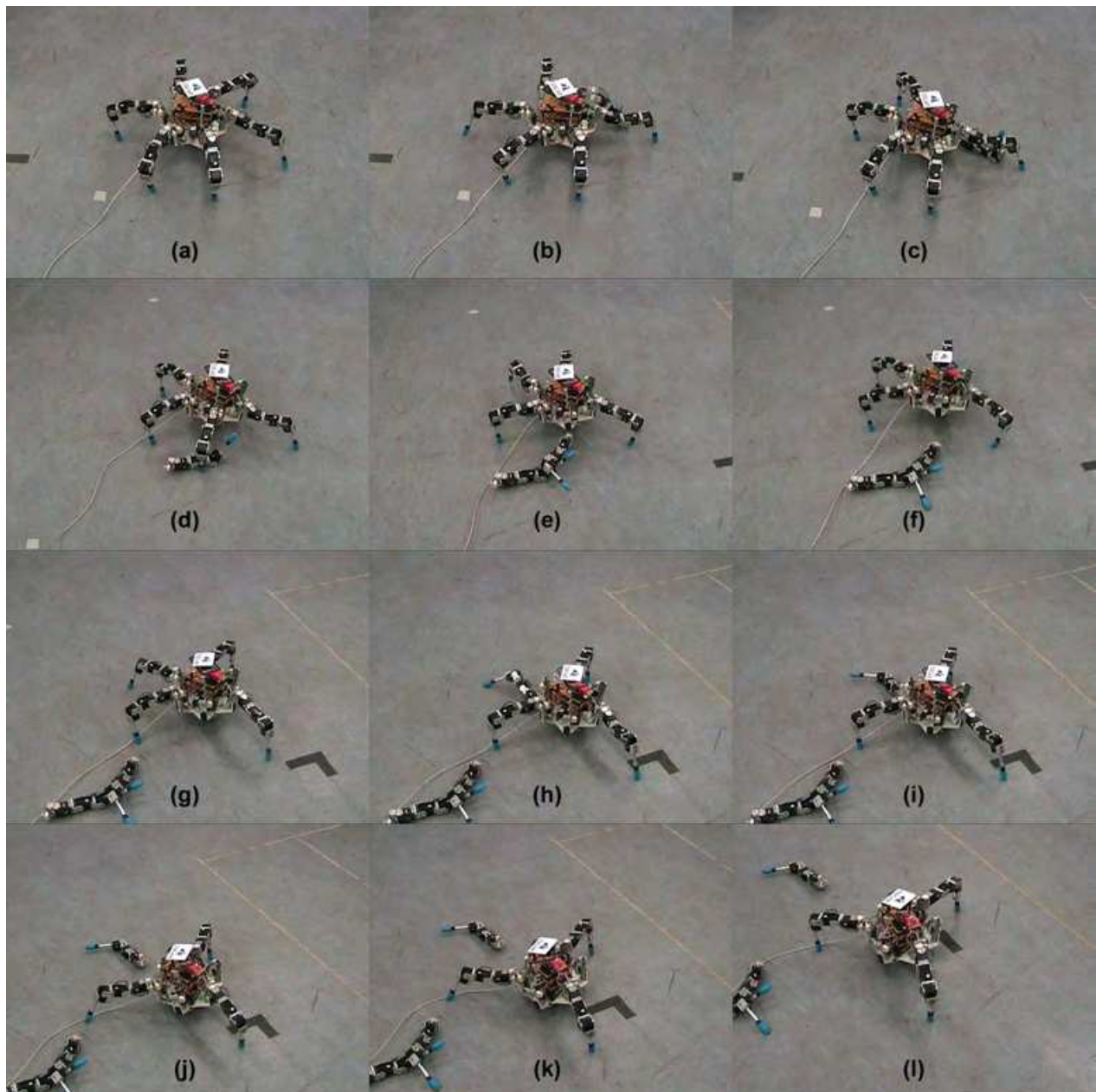


Fig. 9. Runtime reconfiguration of a hexapod robot OSCAR from 6 to 3 legs: (a) normal six legged configuration; (b) leg number 3 is malfunctioned and gets amputated; (c) robot performs reconfiguration using the SIRR approach and continues with walking; (d) leg number 1 becomes malfunctioned; (e) leg number 1 gets amputated; (f) robot performs reconfiguration using the SIRR approach and continues with walking; (g) leg number 5 becomes malfunctioned; (h) leg number 5 gets amputated; (i) robot performs reconfiguration using the SIRR approach and continues with walking; (j)-(l) robot OSCAR continues with its mission despite the loss of 3 legs.

We have made an analysis chart representing the ground contacts of legs by normal walking and by walking with leg amputations and robot self-reconfiguration. The results of these analyses can be seen in Fig. 10, Fig. 11.

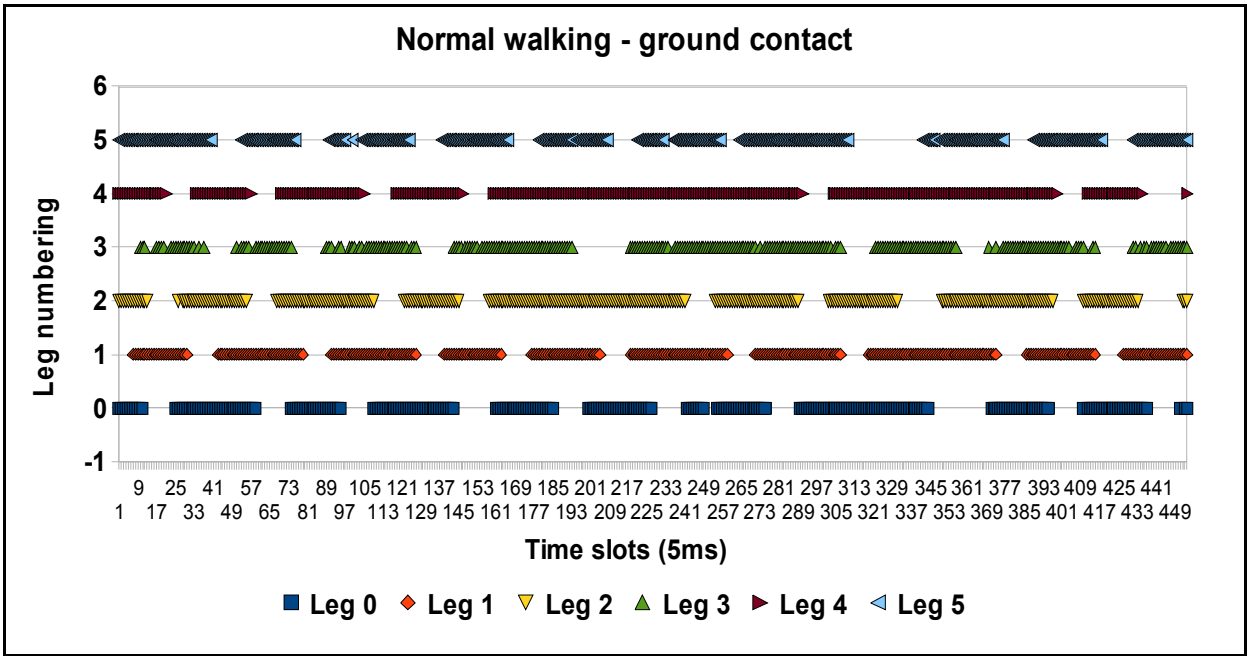


Fig. 10. Ground contacts of the robot’s feet during normal walking of the hexapod robot

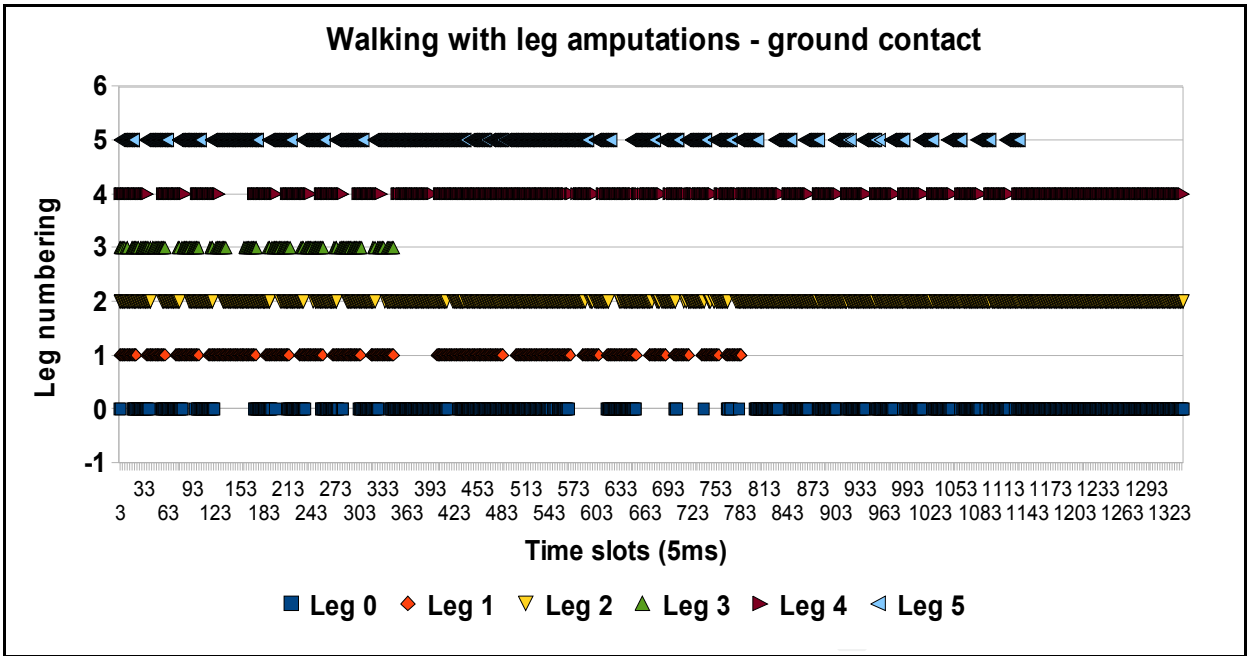


Fig. 11. Ground contacts of the robot’s feet by walking of hexapod robot with leg amputations and self-reconfiguration

The robot in these experiments is walking with a biologically inspired emergent gait, which means that the gait is not “hard-wired” or by any means predefined. A simple rule is used which allows a leg to swing only if its two neighboring legs are on the ground (El Sayed Auf et al., 2006). By this, the gait pattern emerges from the local swing and stance phases of the robot’s legs “joining” the “legs boid” at the particular robot’s side after the reconfiguration has been performed. In Fig. 10 the chart represents the leg ground contacts for normal

walking - fully functional robot. In Fig. 11 the chart represents the leg ground contacts of the robot walking with leg amputations where we can see how the legs get amputated during the experiment, the leg ground contacts are lost and the robot still continues with walking. Leg number 3 gets amputated at time slot 335; Leg number 1 gets amputated at time slot 785; Leg number 5 gets amputated at time slot 1140. The swing phases are drastically shortened with each reconfiguration and after the time slot 1140 the robot still continues to walk although with very shortened swing phases comparing to relatively longer stance phases.

Additional measurements have been done on tracking the robot's heading while performing leg amputations and robot reconfigurations. With these measurements we wanted to test the straight walking and heading of the robot while it is performing leg amputations in different order of leg ejections and its influence on robot's walking.

The solid line in figures: Fig. 12. (a, b); Fig. 13 (a, b); Fig. 14 (a, b) represents the track of the robot during its walking. The arrow lines represent the heading of the robot. The initial heading angle is  $270^\circ$ .

#### Experiment 1:

- Fig. 12. (a) OSCAR performing leg amputations during its walking in the following order: 0, 1, 2 (in Fig. 12. a, from left to right and from up to down).
- Fig. 12. (b) Tracking of the robot's heading while the robot is amputating the legs during its walking in the following order: 0, 1, 2. The solid line represents the track of the robot during its walking. The arrow lines represent the heading of the robot during its walking.

#### Experiment 2:

- Fig. 13. (a) OSCAR performing leg amputations during its walking in the following order: 0, 2, 4 (in Fig. 13. a, from left to right and from up to down).
- Fig. 13. (b) Tracking of the robot's heading while the robot is amputating the legs during its walking in the following order: 0, 2, 4. The solid line represents the track of the robot during its walking. The arrow lines represent the heading of the robot during its walking.

#### Experiment 3:

- Fig. 14. (a) OSCAR performing leg amputations during its walking in the following order: 5, 1, 2 (in Fig. 14. a, from left to right and from up to down).
- Fig. 14. (b) Tracking of the robot's heading while the robot is amputating the legs during its walking in the following order: 5, 1, 2. The solid line represents the track of the robot during its walking. The arrow lines represent the heading of the robot during its walking.

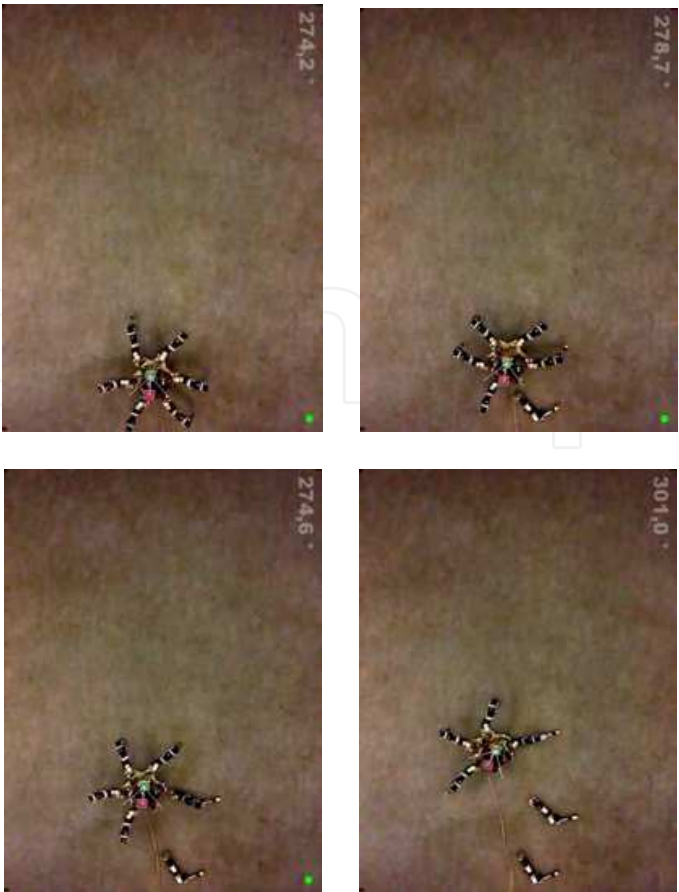


Fig. 12. (a). OSCAR performing leg amputations in the following order: fully functional, leg 0 amputated, leg 1 amputated, leg 2 amputated - from left to right and from up to down.

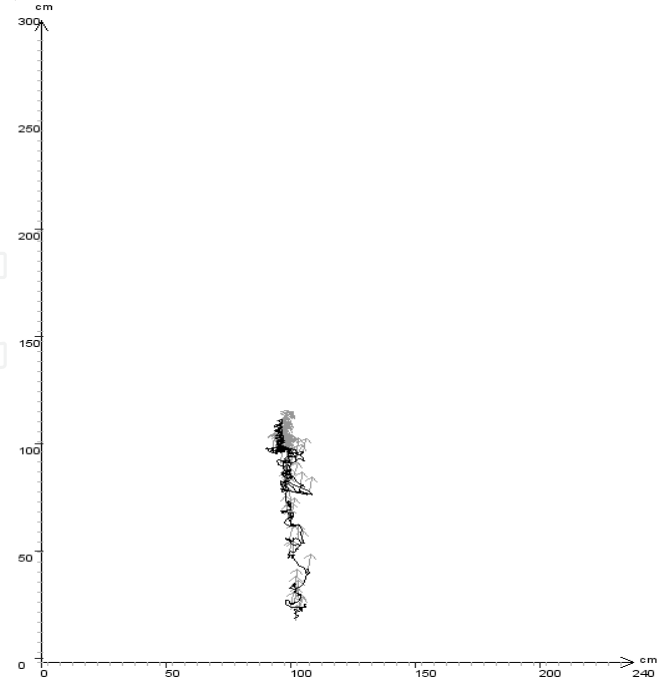


Fig. 12. (b) Tracking of robot's heading while the robot is ejecting the legs during its walking in the following order: 0, 1, 2.

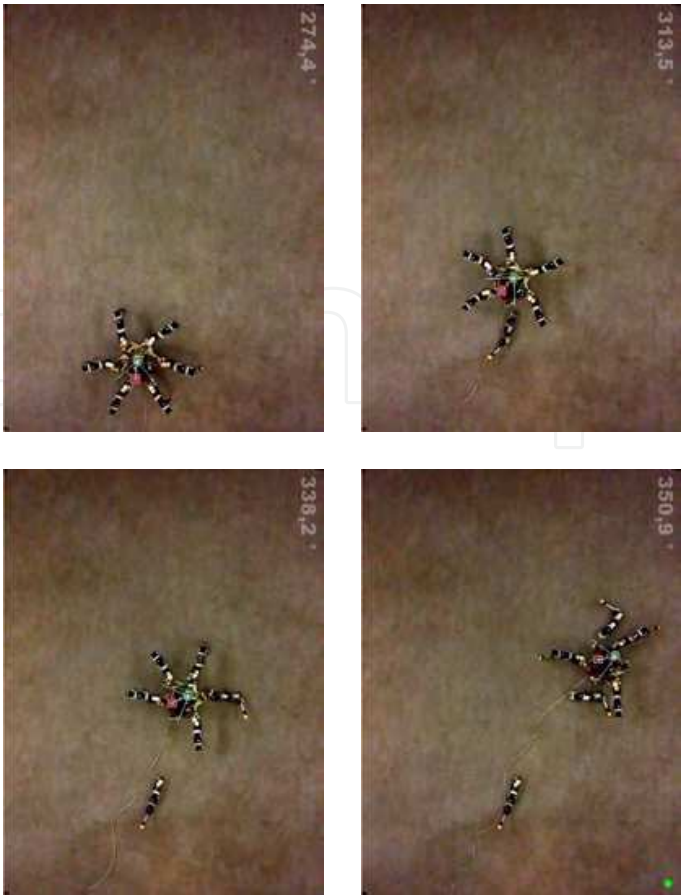


Fig. 13. (a) OSCAR performing leg amputations in the following order: fully functional, leg 0 amputated, leg 2 amputated, leg 4 amputated - from left to right and from up to down.

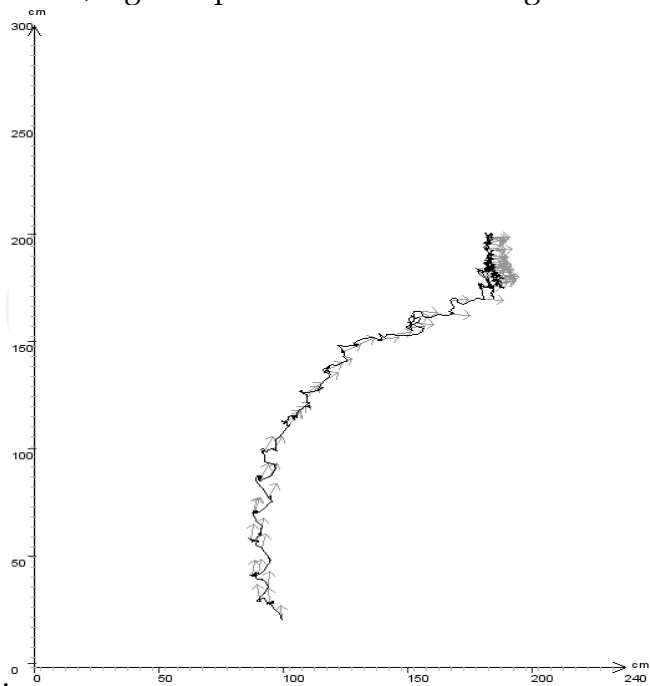


Fig. 13. (b) Tracking of robot’s heading while the robot is ejecting the legs during its walking in the following order: 0, 2, 4.



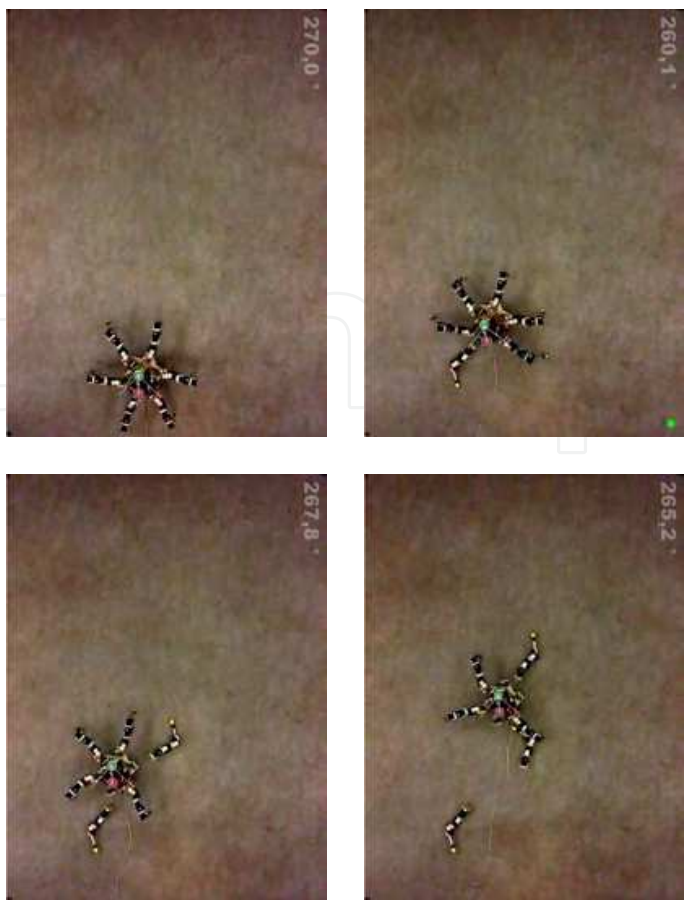


Fig. 14. (a) OSCAR performing leg amputations in the following order: fully functional, leg 5 amputated, leg 1 amputated, leg 2 amputated - from left to right and from up to down.

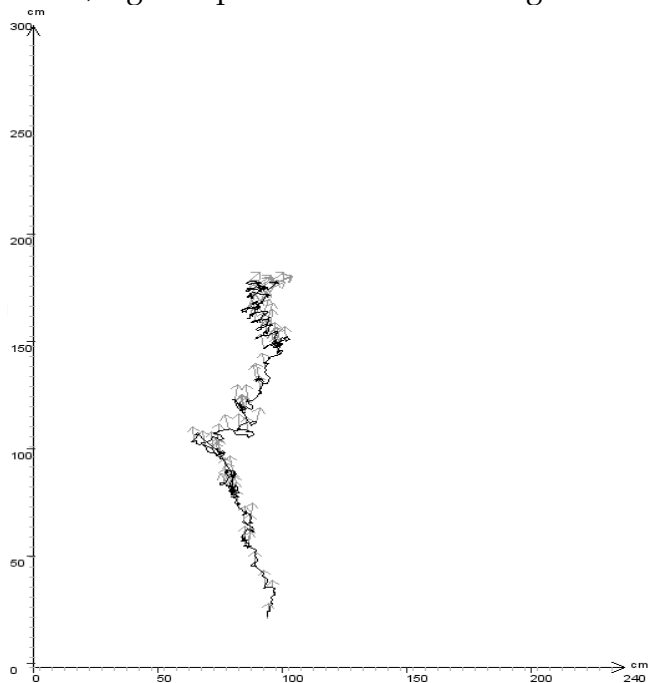


Fig. 14. (b) Tracking of robot's heading while the robot is ejecting the legs during its walking in the following order: 5, 1, 2.

On one hand it is “nice to have” a robotic system that exhibits emergent walking. However on the other hand, this kind of pure emergent walking has perhaps negative influence on how the robot is walking straight and its keeping the heading.

Despite this fact, we still wanted to measure how the robot deviates from the straight path (keeping the course to  $270^\circ$ ) while performing the leg amputations and walking with emergent gait. The results show that even when the robot has malfunctions within its legs and performs legs amputations, it is still more or less capable to walk straight with slight turning in some cases (Fig.13). Although this deviation from course is present, we must take in account also that the robot has amputated legs and that the deviation is perhaps still not that radical - like for example: robot walking in circles immediately, or similar.

One additional idea that might be used to avoid or minimize such deviation from the main course is to couple the emergent behavior with some other behaviors like going right or left, which in that case will somehow intervene with the emergent walking gait in order to keep the robot on its course. This will be as extension to the research done on curve walking with robot OSCAR (El Sayed Auf et al., 2007). This idea will be analyzed further in future experiments done on self-reconfiguring walking robots.

## 5. Conclusion

In this section we have elaborated on biologically inspired methods and experiments done for real case hexapod robot self-reconfiguration. We have introduced a patent pending mechanism used for leg amputation by joint-leg walking robots which is practically used for reconfiguration cases by our hexapod robot OSCAR. Further, we have explained the artificial immune system based approach - RADE, used for monitoring the robot's health status and leg anomaly detection in joint-leg walking robot.

We have also introduced and explained the biologically inspired Swarm Intelligence for Robot Reconfiguration (S.I.R.R.) method which is used for performing in-situ robot self-reconfiguration. The S.I.R.R method is used for spatial distributing of the robot's legs when a reconfiguration is performed. So, the robot achieves a stable spatial configuration even when one or more legs are malfunctioned and get amputated from the robot's body.

Through experimental cases we have demonstrated how the hexapod robot OSCAR - despite the anomalies that occur within its legs - manages to amputate the malfunctioned legs, self-reconfigures and continues with walking. In these experiments also tracking measurements were done on tracking the robot's heading while it is performing leg amputations and self-reconfigurations.

The presented results from experiments on self-reconfiguration look promising, and therefore future work will consider an additional research on integrating self-reconfiguration with the walking robot's high-level behaviors aiming to improve the robot's heading after some reconfiguration is performed. Additional work will be also done on improving the robustness and generic usefulness of the presented self-reconfiguration approach and its potential application for other types of robots.

## 6. Acknowledgment

This work is partly supported by German Research Foundation - DFG (associated to SPP 1183, MA 1412/8-1).

## 7. References

- Brockmann W.; Maehle E. & Mösch F. (2005). Organic Fault-Tolerant Control Architecture for Robotic Applications. *4th IARP/IEEE-RAS/EURON Workshop on Dependable Robots in Human Environments*, Nagoya, Japan.
- Canham R.; Jackson A. H. & Tyrrell A. (2003) Robot Error Detection Using an Artificial Immune System, *Proceedings of the 2003 NASA/DoD Conference on Evolvable Hardware*, 2003.
- Cao Y. & Dasgupta D. (2003). An Immunogenetic Approach in Chemical Spectrum Recognition. *Edited volume Advances in Evolutionary Computing* (Ghosh & Tsutsui, eds.), Springer-Verlag.
- Chien, S.; Doyle, R.; Davies, A.; Jonsson, A. & Lorenz, R. (2006), The Future of AI in Space, *IEEE Intelligent Systems*, pp. 64-69.
- Christensen, A.L.; O'Grady, R.; Birattari, M. & Dorigo, M. (2008). Fault detection in autonomous robots based on fault injection and learning, *Journal Autonomous Robots*, Vol. 24, No. 1 (January, 2008), pp. 49-67.
- De Castro L. N. & Timmis J. (2002). Artificial Immune Systems: A New Computational Intelligence Approach. Springer-Verlag, pp. 36-46.
- De Castro, L.N. & Von Zuben, F.J. (2002). Learning and Optimization using the clonal selection principle, *IEEE Transaction on Evolutionary Computation*, pp. 239-251.
- El Sayed Auf A.; Larionova S.; Litza M.; Mösch F.; Jakimovski B. & Maehle E. (2007). Ein Organic Computing Ansatz zur Steuerung einer sechsbeinigen Laufmaschine. *Autonome Mobile Systeme 2007*, pp. 233-239.
- El Sayed Auf, A.; Mösch, F. & Litza, M. (2006). How the six-legged walking machine OSCAR handles leg amputations, *In Workshop on Bio-inspired Cooperative and Adaptive Behaviours in Robots*, pp. 115-124, Rome, Italy.
- Forrest S.; Perelson A. S.; Allen L. & Cherukuri R. (1994). Self-Nonself Discrimination in a Computer, *In Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy*, Los Alamitos, CA.
- Galeano J.C.; Veloza-Suan A. & Gonzalez F.A. (2005) A comparative analysis of artificial immune network models, *Proceedings of the Conference on Genetic and Evolutionary Computation*, pp. 361-368, Washington DC, USA.
- German Science Foundation (DFG) Priority Program SPP 1183 "Organic Computing", (<http://www.organic-computing.de/spp>), 2004.
- Haldar, B. & Sarkar, N. (2006). Robust fault detection and isolation in mobile robot, *Proceedings of IFAC*, Beijing, China.
- Hancher, M. D. & Hornby, G. S. (2006). A Modular Robotic System with Applications to Space Exploration, *2nd IEEE International Conference on Space Mission Challenges for Information Technology (SMC-IT'06)*, pp. 125-132.
- Hinchey, M. G. & Sterritt, R. (2007). 99% (Biological) Inspiration..., *Fourth IEEE International Workshop on Engineering of Autonomic and Autonomous Systems (EASE'07)*, pp. 187-190.
- Hyun Yool, K.; Lee, Y.; Choi, H.; Hyun Yool, B. & Hwan Kim, D. (2006). Swarm Robotics: Self Assembly, Physical Configuration, and Its Control, *SICE-ICASE International Joint Conference*, pp. 4276-4279.
- IBM Research: Autonomic Computing, (<http://www.ibm.com/research/autonomic>), pp. 21- 30.

- Jakimovski, B.; Litza, M.; Mösch, F. & El Sayed Auf, A. (2006). Development of an organic computing architecture for robot control, *In Informatik 2006 Workshop on Organic Computing - Status and Outlook*, pp. 145-152, Dresden.
- Jakimovski, B.; Meyer B. & Maehle, E. (2008). Swarm Intelligence for Self-Reconfiguring Walking Robot, *IEEE Swarm Intelligence Symposium*, St. Louis, Missouri, USA, Sep. 21-23, 2008.
- Jakimovski, B. & Maehle, E. (2008). Artificial Immune System Based Robot Anomaly Detection Engine for Fault Tolerant Robots, *5th International Conference on Autonomic and Trusted Computing (ATC-08)*, pp. 177-190, Oslo, Norway.
- Lewandowski, S.M.; Van Hook, D.J.; O'Leary, G.C.; Haines, J.W. & Rossey, L.M. (2001), SARA: Survivable Autonomic Response Architecture, *In Proc. DARPA Information Survivability Conference and Exposition II*, Vol. 1, pp. 77-88.
- Michelan R. & Von Zuben F.J. (2002). Decentralized control system for autonomous navigation based on an evolved artificial immune network, *Proceedings of the 2002 Congress on Evolutionary Computation*, Vol. 2., pp. 1021-1026.
- Nasraoui O.; Cardona C.; & Rojas C. (2005). Using retrieval measures to asses similarity in mining dynamic web clickstreams, *Proceeding of the eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining KDD*, 2005.
- Neal M.; Feyereisl J.; Rascuna R. & Wang X. (2006). Touch Me, I'm Fine: Robot Autonomy Using an Artificial Innate Immune System, *5th International Conference on Artificial Immune Systems*, Oeiras, Portugal.
- Nino F. & Beltran O. (2002). A change detection software agent based on immune mixed selection, *Evolutionary Computation, CEC '02. Proceedings of the 2002 Congress on*, Vol. 1, pp. 693-698.
- Pagnoni A. & Visconti A. (2005). An innate immune system for the protection of computer networks. *Proceedings of the 4th international symposium on information and communication technologies*, Vol.92, pp. 63-68.
- Przystalka, P. (2006). Model-Based Fault Detection and Isolation Using Locally Recurrent Neural Networks, *Lecture Notes In Artificial Intelligence*, Vol. 5097, pp. 123 - 134.
- Reynolds, C. (1987) Flocks, herds, and schools: A distributed behavioral model, *Comp. Graph*, Vol. 21, No. 4, 1987, pp. 25-34.
- Sathyanath S. & Sahin F.(2002). AISIMAM - An Artificial Immune System Based Intelligent Multi Agent Model and its Application to a Mine Detection Problem, *1st International Conference on Artificial Immune Systems*, Canterbury, UK.
- Singh C.T. & Nair S.B. (2005). An Artificial Immune System for a MultiAgent Robotics System, *Transactions of Engineering, Computing and Technology*, Vol.6, pp. 308-311.
- Sterritt, R.; Hinchey, M.; Rouff, C.; Rash, J. & Truszkowski, W. (2006), Sustainable and Autonomic Space Exploration Missions, *2nd IEEE International Conference on Space Mission Challenges for Information Technology (SMC-IT'06)*, pp. 59-66.
- Verma, V. & Simmons, R. (2006). Scalable robot fault detection and identification, *Robotics and Autonomous Systems*, Vol. 54, No. 2, pp. 184-191.
- Winfield, A.F.T. (2006). Safety in numbers: fault-tolerance in robot swarms, *International Journal of Modelling, Identification and Control*, Vol. 1, No. 1, pp. 30-37.
- Xuan, L.; Minglu, Z. & Wei, L. (2008). Methods to Modular Robot Design, *Second International Symposium on Intelligent Information Technology Application*, pp. 663-668.

Zhuo-hua, D.; Ming, F.; Zi-xing, C. & Jin-xia, Y. (2006). An adaptive particle filter for mobile robot fault diagnosis, *Journal of Central South University of Technology*, Vol. 13, No. 6 (December, 2006), pp. 689-693.

IntechOpen

IntechOpen





## **Climbing and Walking Robots**

Edited by Behnam Miripour

ISBN 978-953-307-030-8

Hard cover, 508 pages

**Publisher** InTech

**Published online** 01, March, 2010

**Published in print edition** March, 2010

Nowadays robotics is one of the most dynamic fields of scientific researches. The shift of robotics researches from manufacturing to services applications is clear. During the last decades interest in studying climbing and walking robots has been increased. This increasing interest has been in many areas that most important ones of them are: mechanics, electronics, medical engineering, cybernetics, controls, and computers. Today's climbing and walking robots are a combination of manipulative, perceptive, communicative, and cognitive abilities and they are capable of performing many tasks in industrial and non- industrial environments. Surveillance, planetary exploration, emergence rescue operations, reconnaissance, petrochemical applications, construction, entertainment, personal services, intervention in severe environments, transportation, medical and etc are some applications from a very diverse application fields of climbing and walking robots. By great progress in this area of robotics it is anticipated that next generation climbing and walking robots will enhance lives and will change the way the human works, thinks and makes decisions. This book presents the state of the art achievements, recent developments, applications and future challenges of climbing and walking robots. These are presented in 24 chapters by authors throughout the world. The book serves as a reference especially for the researchers who are interested in mobile robots. It also is useful for industrial engineers and graduate students in advanced study.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Bojan Jakimovski and Erik Maehle (2010). In Situ Self-Reconfiguration of Hexapod Robot OSCAR Using Biologically Inspired Approaches, Climbing and Walking Robots, Behnam Miripour (Ed.), ISBN: 978-953-307-030-8, InTech, Available from: <http://www.intechopen.com/books/climbing-and-walking-robots/in-situ-self-reconfiguration-of-hexapod-robot-oscar-using-biologically-inspired-approaches>

**INTECH**  
open science | open minds

### **InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821

[www.intechopen.com](http://www.intechopen.com)

IntechOpen

IntechOpen

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen