Experiments on Fault-Tolerant Self-Reconfiguration and Emergent Self-Repair

Christensen, David Johan

Published in:
Proceedings of Symposium on Artificial Life part of the IEEE Symposium Series on Computational Intelligence

Publication date:
2007

Document Version
Publisher's PDF, also known as Version of record

Experiments on Fault-Tolerant Self-Reconfiguration and Emergent Self-Repair

David Johan Christensen
AdapTronics group, Maersk Mc-Kinney Moller Institute
University of Southern Denmark
Odense, Denmark
Email: david@mip.sdu.dk

***Abstract*— This paper presents a series of experiments on fault tolerant self-reconfiguration of the ATRON robotic system. For self-reconfiguration we use a previously described distributed control strategy based on meta-modules that emerge, move and stop. We perform experiments on three different types of failures: 1) *Action failure:* On the physical platform we demonstrate how roll-back of actions are used to achieve tolerance to collision with obstacles and other meta-modules. 2) *Module failure:* In simulation we show, for a 500 module robot, how different degrees of catastrophic module failure affect the robot's ability to shape-change to support an insecure roof. 3) *Robot failure:* In simulation we demonstrate how robot faults such as a broken robot bone can be emergent self-repaired by exploiting the redundancy of self-reconfigurable modules. We conclude that the use of emergent, distributed control, action roll-back, module redundancy, and self-reconfiguration can be used to achieve fault tolerant, self-repairing robots.**

I. Introduction

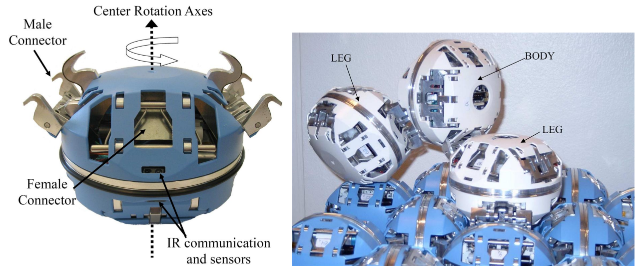
Biological organisms are highly robust. Bone and skin will heal itself if broken, the brain and body will adapt to the loss of an eye or an arm - staying alive, functioning as well as possible. Robots, on the contrary, are in general not very fault tolerant. The loss of a sensor, an actuator or a piece of mechanics will in most cases leave the robot completely helpless and unable to perform its function. One reason for the successfulness of biological organisms is the trillions of cells making up the body. A multi-cellular biological organism has no single-point-of-failure; this is ensured by the redundancy of cells - which may divide, migrate, differentiate, or die to assemble or repair the organism they compose.

Inspired by multi-cellular organisms, self-reconfigurable robots are made up by numerous interconnected (in a lattice or a chain) robotic modules [5]. Modules can sense the environment as well as communicate with and manipulate neighbor modules to change the structure of modules. Related approaches include systems for stochastic [6], [25] and mobile self-assembly [7]. In principle, a self-reconfigurable robot has the ability to tolerate failures of its individual modules and adapt its shape to accommodate a range of functionalities in a given environment. Thus, self-reconfigurable robots have the potential to become more versatile, robust and adaptive than traditional robots.

State-of-the-art self-reconfigurable robots consist of up to a few dozen modules and can perform tasks such as locomotion (e.g. [3], [11], [13], [17], [19], [22]). We are, however, concerned with scaling up the number of modules while scaling down their size. Two challenges related to scaling are centralized or distributed control of self-reconfiguration [1], [10], [15], and robustness in spite of inevitable module failures. In this paper we address the latter challenge. Related work on self-repair of self-reconfigurable robots generally involves the detection of module failure, decisions on how to remove a defect module, and how to replace it with a spare module [4], [21], [24]. Alternatively, as in this work, self-repair can emerge as a side effect of the self-reconfiguration, without having a specialized self-repairing part of the controller [20].

The ATRON self-reconfigurable robotic system is our experimental platform. ATRON is a simple, one degree of freedom, homogeneous, lattice-based module, which is able to self-reconfigure in 3D (described in Section II). To control self-reconfiguration between shapes we use a strategy which is based on distributed control of meta-modules. A meta-module consists of three modules and is able to move around quite freely, on top of other modules (see Figure 1(b)). The control strategy is described in [2] and summarized in Section III. As in preceding work [12], [16], [18], [23] meta-modules are used to reduce the motion constraints of the base modules, in order to simplify the process of self-reconfiguration at a higher hierarchical level.

This paper explores fault tolerance to *action failure*, *module failure* and *robot failure*, especially applicable in the context of self-reconfigurable robots consisting of large number of modules (>50). First, Sections V-A to V-C present real world experiments with up to nine active and up to 24 passive modules both to illustrate the basic capabilities of a meta-module, and to verify tolerance to *action failures*. We demonstrate that meta-modules can roll-back their actions when colliding with obstacles or other meta-modules. As a side-effect, roll-back of failed actions enable the meta-modules to find their way around unknown obstacles in their environment, and allow mul-



(a) (b)

Fig. 1. (a) A single ATRON module: on the top hemisphere the two male connectors are extended, on the bottom hemisphere they are contracted. (b) A meta-module is composed of three modules, one center module and two legs.

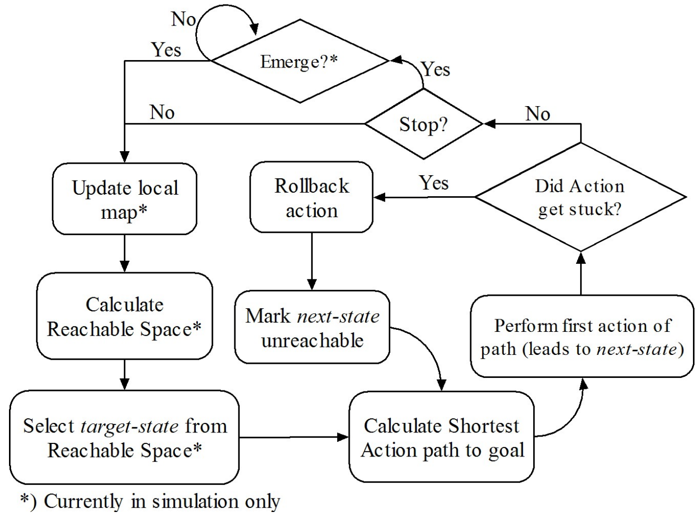


Fig. 2. Each module is controlled using the strategy illustrated above. Modules can be passive or active as part of a meta-module. A meta-module selects an action to perform based on the local configuration of modules and guided by attraction points. It then performs the action which can fail or succeed.

tiple meta-modules to co-exist (without the need of any explicit presentation/knowledge of environment or other meta-modules). Second, Section V-D presents simulated experiments with 500 active modules shape-changing to support an insecure roof. The experiment shows graceful degradation of functionality when modules fail, since the system can tolerate more than 10% catastrophic *module failures* and still perform its purpose. Third, Section V-E presents a simulated experiment on *robot failure* using emergent self-repair through self-reconfiguration: 3426 modules assembled as a bone is partly broken by removing 114 modules. The modules then self-reconfigure and recover the bone strength, by exploiting the redundancy of modules.

The strategies used to achieve the reported level of fault tolerance are simple and efficient, although they may not be sufficient in all cases.

II. The ATRON Module

The ATRON self-reconfigurable robotic system [9] is a homogeneous modular system. This means that all modules are identical both in hardware and software, functional differentiation is achieved using roles. Modules can be assembled into a variety of robots: Robots for locomotion (like snakes, cars, and walkers), robots for manipulation (like small robot arms) or robots that achieve some functionality from their physical shape, such as structural support. By self-reconfiguring, modules can change the shape of the robot, for example from a car to a snake and then to a walker.

An ATRON module has a spherical appearance composed of two hemispheres, which can actively be rotated relative to each other. On each hemisphere a module has two actuated male connectors and two passive female connectors. In the HYDRA project [14] we have manufactured 100 ATRON modules, a single module is shown in Figure 1(a).

Rotation around the center axes is, for self-reconfiguration, always done in 90 degree steps. This moves a module, connected to the rotating module, from one lattice position to another. One full 360 degree rotation takes about 6 seconds, without the load from other modules. Encoders are used to control the rotation of the center axes. Male connectors are actuated and shaped like three hooks, which grasp on to passive female connector bars. A connection or a disconnection takes about two seconds. In relation to each connector is an infrared transmitter and receiver, that allow modules to communicate with neighbor modules and sense distance to nearby objects. Connectors are positioned in such a way that the ATRON modules sits in a global surface-centered cubic lattice structure. Furthermore, each module is equipped with three tilt sensors that allow the module to know its orientation relative to the direction of gravity.

III. ATRON Meta-Module

Generally speaking, a meta-module is composed of a number of modules collaborating to achieve some common task. For the purpose of self-reconfiguration the meta-module's task is to move based on some strategy. A meta-module can from the outside be seen as a single acting entity or agent. The specific ATRON meta-module consists of three modules: a center module is connected to two other modules, one on each of its hemispheres. We use meta-modules to relax the hard motion constraints on the individual modules. Factors contributing to the motion constraints include: a module only has a single degree of freedom, it can never move itself but must be moved by other modules, and must take into account its limited actuator strength while it avoids collisions and disconnections that will break apart the structure of modules.

Self-reconfiguration of large module structures (>50), can be realized using a control strategy presented in [2] (simulation only). Basically, meta-modules emerge from

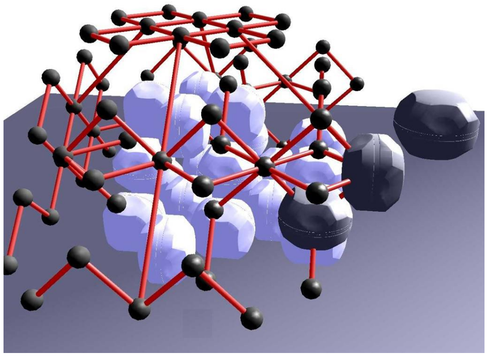
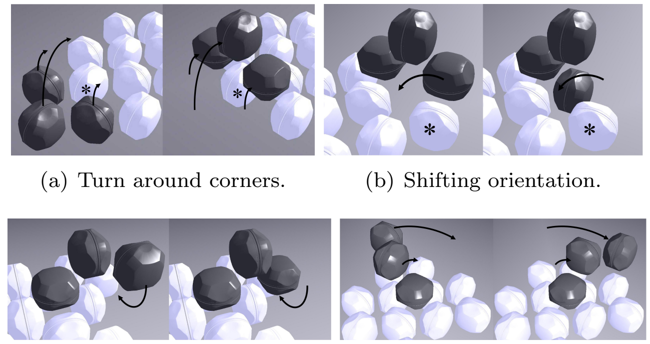


Fig. 3. Illustration of reachable-space. A meta-module performs actions that will bring it closer to an attraction point. The meta-module selects the actions by performing a shortest path search on a subset of its reachable-space.



(a) Turn around corners. (b) Shifting orientation.

(c) Rotation of body-module. (d) Rotation of leg-module.

Fig. 4. The meta-actions an ATRON meta-module is able to perform. Dark modules comprise the meta-module. The ∗-marked modules in (a) and (b) are required to participate in the corresponding meta-actions.

unstructured groups of modules, move on the surface of the other modules (guided by attraction points), and stop when reaching a local minimum in the distance to an attraction point. Below we summarize this control strategy (refer to Figure 2).

a) Passive or active: Modules are either passive or part of a meta-module. Passive modules can respond to simple request from meta-modules such as *connect-to-me*, but can also decide to form and emerge a meta-module from the structure of modules. A meta-module can only emerge if three passive modules are connected in a legal meta-module configuration (no constraint on orientation). A meta-module is then able to move, but may also decide to stop its movement. If the meta-module stops its modules become passive once again and the process can be repeated later. This approach is inspired by the division, migration, and death of biological cells. The decisions to emerge or stop a meta-module are taken by two artificial neural networks, which are optimized by evolution to quickly shape-change a large structure of modules (see [2]).

b) Action selection: At any given time many meta-modules will be moving in the system. To guide the flow of meta-modules we use attraction points, which define the shape of the desired global configuration. Meta-modules move towards attraction-points by performing a sequence of meta-actions. A module will inhibit an attraction-point if placed at the same location. Inhibited attraction-points are ignored by meta-modules. Meta-modules perform some local planning to select which meta-action to perform: First, the meta-module constructs a map of the local configuration of modules (6 hops). Second, the meta-module calculates a local subset of its *reachable-space* (see Figure 3). The reachable-space is a graph where vertices are legal states (position and orientation) of the meta-module and edges are legal meta-actions which brings the meta-module from one legal state to another. Third, the meta-module calculates (using an A* algorithm [8] on the reachable-space) a shortest path sequence of meta-actions towards a goal-state. The goal-state is selected by a neural network (also evolved) based on characteristics such as proximity to attraction points. Finally, the meta-module will perform the first meta-action from the found sequence, and the action selection process can be repeated.

c) Action execution: A meta-module performs a sequence of meta-actions to move. A meta-action is composed of a sequence of basic module actions (rotation, connection and disconnection), which are performed by the modules part of, or neighbor to, the meta-module. Meta-modules can perform four different types of meta-actions (see Figure 4). Each type represents 2 or 4 different meta-actions, so in total a meta-module can perform 12 different meta-actions. However, in a given situation only a subset of these 12 meta-actions will be legal. The meta-actions allow the meta-module to move quite freely on the surface of a structure of modules. Robustness is increased by handling meta-actions that fail. Usually this means that a rotation results in a collision, or that a failed module is connected to and therefore locks the module. Collisions are detected by the rotating modules using its encoders. The roll-back strategy is to reverse the rotation, mark the state (in the reachable space) as unreachable and select another action to perform (recalculate shortest path).

IV. Experimental Setup

A partial implementation of the meta-module-based controller has been transferred to the physical ATRON platform. The implementation builds on abilities of the modules, such as rotate, connect and disconnect. It corresponds to the lower level control of individual meta-modules (refer to Figure 2). Each physical meta-module can:

- On-line find a shortest-path of meta-actions from a reachable-space, which is known at compile-time.
- Perform the meta-action types of Figure 4(c) or 4(d).
- Detect and perform roll-back of failed actions.

Module-to-module communication is currently unstable (due to reflections of IR-communication), this limitation

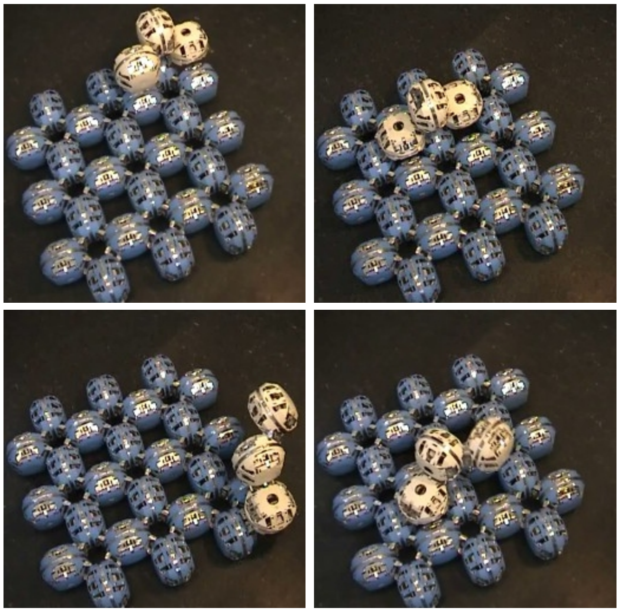


Fig. 5. The meta-module repeatedly calculates and then moves shortest path from one randomly selected state in its reachable-space to another.

Exp.	#Meta-Actions	Exp. time (seconds)	Second pr. meta-action
1	8	52	6.5
2	48	330	6.9
3	38	228	6.0
4	93	570	6.1
Total	187	1180	6.3

TABLE I

Single meta-module following online-planned sequences of meta-actions.

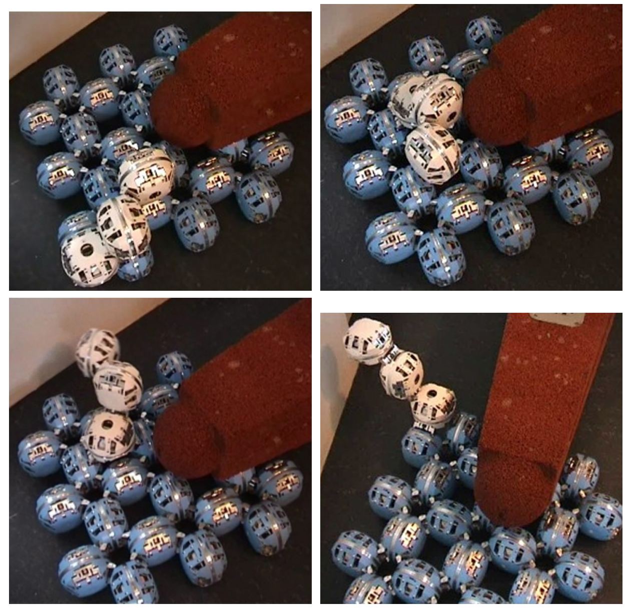


Fig. 6. In this trial the meta-module must move from one corner to the opposite corner. Movement is blocked by an obstacle unknown to the meta-module. The meta-module detects the obstacle by collision and finds its way around it by trying alternative routes.

Exp.	#Collisions	#Meta-Actions performed (optimal)	Percent longer than optimal
1	5	32 (20)	38%
2	4	24 (17)	29%
3	10	38 (20)	47%
Total	19	94 (57)	39%

TABLE II

Meta-module moving from a starting to a goal position around an unknown obstacle.

is the main reason for only demonstrating a partial transference, and the small number of experiments performed on the physical modules. We are working towards resolving this issue.

In the physical experiments 24 passive modules are initially assembled as a horizontal sheet, on which the meta-modules can easily move. Meta-modules (with white shells), move on top of these modules. Meta-modules are place at a predefined position (usually the corner) on the sheet of modules. By sending a special message to the meta-module, using another module as a remote control, the meta-module is started.

Simulation experiments are performed in a transition-based simulation (no physics except collisions), which contains a full implementation of the meta-module-based control strategy described above.

V. Experiments

A. Basic Meta-Module Behavior

In this experiment (see Figure 5) the meta-module selects a random state, then moves (following shortest path) to that state, and then selects a new random state and so forth. After a meta-action is performed the meta-module always recalculates the shortest path. This allows the meta-module to adapt to possible changes in the environment or configuration of modules. The details of four experiments (all with the same initial setup) are shown in Table I. On average a meta-action takes 6.3 seconds to perform, including the calculation of shortest path and the coordination between modules comprising the meta-module.

B. Tolerance of Action Failure - Unknown Obstacle

In this experiment we demonstrate how a meta-module handles collisions with obstacles in its environment. By using its encoder, a meta-module detects an obstacle when colliding with it. If a collision is detected while performing a meta-action, the meta-module performs a roll-back rotation to the lattice-position it came from. The corresponding state in the meta-module's reachable-space is then assumed to be filled with an obstacle and is hereafter ignored when doing shortest-path search. The meta-

Fig. 7. In this trial three meta-modules move on the same surface of modules. They do not communicate so they collide with one another, but the control system is able to tolerate this so that they can co-exist.

#Meta-Modules	#Meta-Actions (#Collisions)	Collisions pr. meta-action
2	56 (7)	0.125
3	52 (10)	0.19

TABLE III

Two and three meta-modules coexisting - no coordination.

module then finds an alternative shortest-path and follows that until it perhaps again collides with an obstacle. By repeating this pattern the meta-module is able to find its way around unknown obstacles (Figure 6). Table II summarizes the results of three different experiments with unknown obstacles. The position of the attraction-point and obstacle are varied for each experiment. The number of meta-actions performed by the meta-modules using this trial-and-error approach ranges from being 29% to 47% higher than what could optimally be achieved using global knowledge.

C. Tolerance of Action Failure - Colliding Meta-Modules

There is a tradeoff between the amount of coordination and the rate of collisions between moving meta-modules. In this experiment there is no coordination between the meta-modules, so they will collide with each other from time to time. To handle this we apply the same roll-back rotation strategy as is used to handle collision with unknown obstacles.

In the trial, shown on Figure 7, three independent meta-modules moves following shortest path of meta-actions,

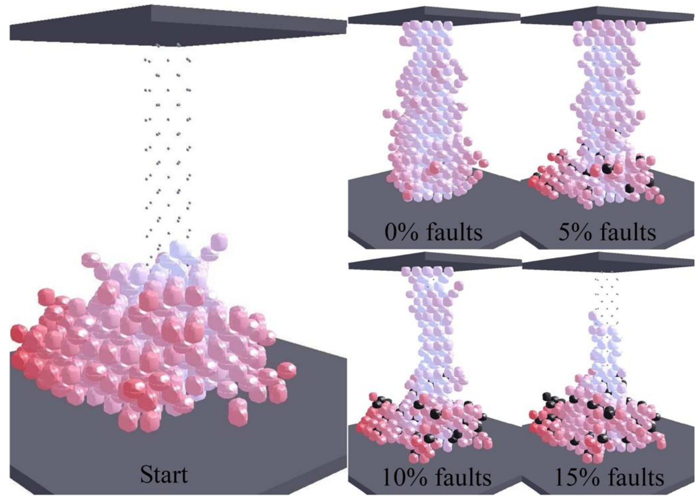


Fig. 8. The meta-module based control is fairly tolerant to module failures. The initial (random) configuration of 500 ATRON modules is shown on the left. The robot shape-change guided by attraction-points shown as small dots. On the right the result is shown for different degrees of module failures. The failed modules are black and are failed from the start of the simulation.

from one randomly selected state to another. The meta-modules collide but roll-back resolves this conflict and the meta-modules can find their way around each other. Table III summarizes two experiments with two and three meta-modules respectively moving on a surface of modules. During the experiment 12.5% and 19%, for two and three meta-modules respectively, of the performed meta-actions are rolled back due to collisions.

D. Tolerance of Module Failure

To investigate the meta-module-based controller's ability to tolerate catastrophic module failures, a series of experiments have been performed in simulation. The task is to support an insecure roof. The initial structure consists of 500 modules and 117 attraction points. If no modules fail the robot will reach the roof and thereby support it (see Figure 8). The experiment is repeated with failure rates of 5%, 10% and 15%. A failure rate of e.g. 10% means that initially and during the experiment 50 randomly selected modules out of the 500 ATRON modules are non-functional from the start. This means that the modules are unable to communicate, rotate, connect, or disconnect. The initial state of a connector is connected, so a failed module will generally lock other functional modules in place with its male connectors. The functional modules can not use the failed module to move on since they have no way of detecting it as a module, in fact failed modules will be treated as obstacles. As can be seen from Figure 8, the robot is able to support the roof up to a failure rate of 10%, but the strength of the robot tends to become lower as the failure rate increase. Also, the speed of changing shape declines as the failure rate increases.

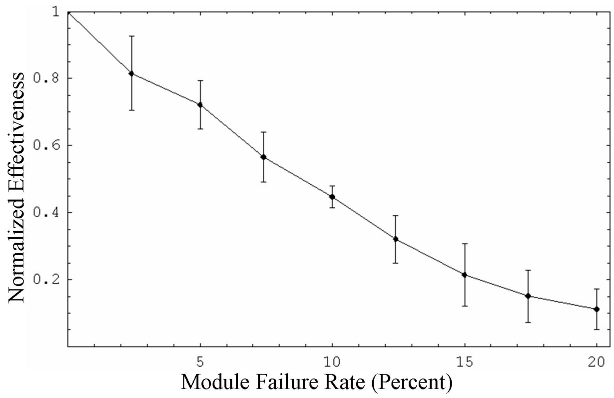


Fig. 9. The graph shows the performance of a 500-module robot as a function of module failure rate. The robot shape changes from an initial random configuration to support an insecure roof. Each point is the average of 10 experiments with varying starting configurations. The effectiveness of each experiment is normalized with respect to the effectiveness on the same initial configuration in the case of 0% module failure rate. Error-bars are 95% confidence intervals.

The performance of the robot declines when the module failure rate increases (Figure 9). We measure performance as *effectiveness* $= (D_{start} - D_{end})/D_{start}$, which is the relative decrease in sum of Euclidian distances between the modules and the attraction-points. This may not a good performance measurement for this particular task, it is however a good general performance measurement for the ability to shape-change.

The observed degree of tolerance to module failures emerges from the redundancy of modules and the use of distributed control of meta-modules. This tolerance could be improved further by removing failed modules from the system (e.g. let them fall off), however, the current connector system would require up to four functional modules to be "sacrificed" per failed module.

E. Tolerance of Robot Failure

A future miniaturization of modules would open up for new possible applications, e.g. smart material which could self-repair. Such applications are feasible since smaller modules can be expected to be stronger and faster (due to physical scale effects).

This experiment demonstrates the use of miniature ATRON modules in an emergent self-repair scenario. Initially, using a CAD model, 3426 ATRON modules are assembled in the form of a bone (Figure 10). A total of 1663 inhibiting attraction-points are placed at the same positions as modules which are connected to eight neighbors. This leaves the surface of the bone free of attraction-points. At timestep 20, 114 modules are removed which damages the strength of the bone. Since the removed modules no longer inhibit the attraction points, this triggers the emergence of meta-modules. After 1000 timesteps (equivalent of 150 seconds on the physical system) the modules have rearranged themselves, the bone is self-repaired, and its strength largely recovered.

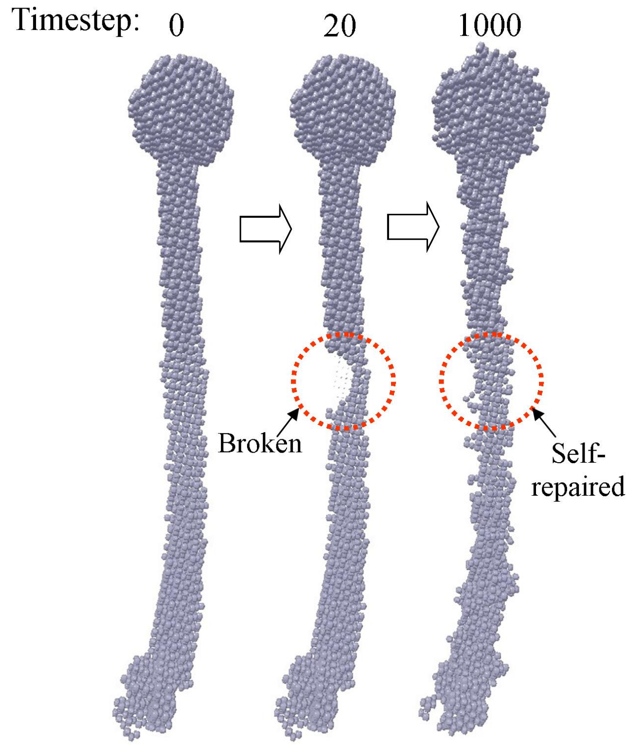


Fig. 10. Self-repair of a bone build from 3426 ATRON modules: At timestep 0, there is no activity. At timestep 20, the bone breaks. At timestep 1000, modules have rearranged themselves to self-repair the bone.

VI. Discussion

We envision self-reconfigurable robots to ultimately consist of billions of micron-size modules. In such robots, at any given time, modules will fail and a considerable portion of the modules can be expected to be non-functional. In this scenario, distributed methods that handle failures locally and in a non-explicit fashion are desirable. Based on our experiments we observe that the impact of action, module, and robot failures can be reduced, as follows:

Action Failures: Control can be simplified, and robustness increase, by using short action that can be rolled back locally. Because this control approach limits the need for collaboration between modules and assumptions about the environment.

Module Failures: The impact of module failures is limited by using meta-modules that emerge from unstructured groups of modules and move somewhat independently on other modules. However, we also observe that morphological adaptation (such as two-way disconnect) of the modules could increase the system's robustness. A key factor is limiting the dependence between modules.

Robot Failures: The system can self-organize its modules to achieve some level of self-repair, without any part of the system ever having to be "aware" of any faults. This is due to the use of emergent, distributed control where modules reacts locally to the removal of modules. Key factors are redundancy and self-reconfiguration, which limits the robot's dependence on its modules.

VII. Conclusion and Future Work

This paper has presented some experiments on emergent self-repair and fault-tolerant self-reconfiguration of the ATRON robot. We have in part transferred a meta-module-based control strategy to the physical modules and verified the basic characteristics of meta-modules and the use of roll-back of failed actions. This allows the meta-modules to co-exist with other meta-modules and find their way around unknown obstacles in their environment. Simulated experiments shows that the emergence of meta-modules from unstructured groups of modules helps tolerate up to 10% failed modules. We have also demonstrated how the redundancy of modules allows self-repair of a bone to emerge. Future work includes a complete transference from simulation to the physical world of the results obtained with meta-module-based control of the ATRON system.

VIII. Acknowledgements

This work was partly funded by the EU FET project HYDRA, and the Self-assembling Robotic Artefacts project sponsored by the Danish Technical Science Council. Thanks to C. Ryberg, D. Kyrping, D. Brandt, E. Østergaard, H. Lund, J. Hallam, K. Støy, K. Kassow, L. Dalgaard, L. Paramonov, R. Beck, and U. Schultz who have all contributed to the ATRON.

References

[1] Z. Butler, K. Kotay, D. Rus, and K. Tomita. Generic decentralized control for a class of self-reconfigurable robots. In *Proceedings, IEEE Int. Conf. on Robotics and Automation (ICRA)*, volume 1, pages 809–815, Washington, DC, USA, 2002.

[2] D. J. Christensen. Evolution of shape-changing and self-repairing control for the ATRON self-reconfigurable robot. In *Proceedings of the IEEE Int. Conference on Robotics and Automation(ICRA)*, May 2006.

[3] D. Duff, M. Yim, and K. Roufas. Evolution of polybot: A modular reconfigurable robot. In *Proceedings, Harmonic Drive International Symposium*, Nagano, Japan, 2001.

[4] R. Fitch, D. Rus, and M. Vona. A basis for self-repair using crystalline modules. In *Proceedings, Intelligent Autonomous Systems (IAS-6)*, Venice, Italy, 2000.

[5] T. Fukuda and S. Nakagawa. Dynamically reconfigurable robotic system. In *Proceedings, 1988 the IEEE Int. Conf. on Robotics & Automation*, 1988.

[6] S. Griffith, D. G., and J. Jacobson. Self-replication from random parts. *Nature*, 437:636, 29 September 2005.

[7] R. Groß, M. Bonani, F. Mondada, and M. Dorigo. Autonomous self-assembly in a swarm-bot. In *Proc. of the 3rd Int. Symp. on Autonomous Minirobots for Research and Edutainment (AMiRE 2005)*, pages 314–322. Springer, Berlin, Germany, 2006.

[8] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, SSC-4(2):100–107, 1968.

[9] M. W. Jørgensen, E. H. Østergaard, and H. H. Lund. Modular ATRON: Modules for a self-reconfigurable robot. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2068–2073, 2004.

[10] J. Kubica, A. Casal, and T. Hogg. Complex behaviors from local rules in modular self-reconfigurable robots. In *Proceedings, IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 360–367, Seoul, Korea, May 2001.

[11] H. Kurokawa, A. Kamimura, E. Yoshida, K. Tomita, S. Kokaji, and S. Murata. M-TRAN II: Metamorphosis from a four-legged walker to a caterpillar. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2454–2459, 2003.

[12] C. McGray and D. Rus. Self-reconfigurable molecule robots as 3d metamorphic robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'98)*, volume 2, pages 837–842, Victoria, B.C., Canada, 1998.

[13] S. Murata, H. Kurokawa, E. Yoshida, K. Tomita, and S. Kokaji. A 3-d self-reconfigurable structure. In *Proceedings, IEEE Int. Conf. on Robotics & Automation (ICRA'98)*, pages 432–439, Leuven, Belgium, 1998.

[14] E. H. Østergaard, D. J. Christensen, P. E. Hotz, T. Taylor, P. Ottery, and H. H. Lund. Hydra: From cellular biology to shape-changing artefacts. In *Proceesings of International Conference on Artificial Neural Networks (ICANN)*, pages 275–281, 2005.

[15] E. H. Østergaard and H. H. Lund. Distributed cluster walk for the ATRON self-reconfigurable robot. In *Proceedings of the The 8th Conference on Intelligent Autonomous Systems (IAS-8)*, pages 291–298, Amsterdam, Mar. 2004. IOS Press.

[16] K. C. Prevas, C. Ünsal, M. Ö. Efe, and P. K. Khosla. A hierarchical motion planning strategy for a uniform self-reconfigurable modular robotic system. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA2002)*, pages 787–792, 2002.

[17] D. Rus and M. Vona. A physical implementation of the self-reconfiguring crystalline robot. In *Proceedings, IEEE International Conference on Robotics & Automation*, pages 1726–1733, San Francisco, USA, 2000.

[18] D. Rus and M. Vona. Crystalline robots: Self-reconfiguration with compressible unit modules. *Autonomous Robots*, 10(1):107–124, 2001.

[19] Behnam Salemi, Mark Moll, and Wei-Min Shen. SUPERBOT: A deployable, multi-functional, and modular self-reconfigurable robotic system. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, Beijing, China, October 2006.

[20] K. Stoy and R. Nagpal. Self-repair through scale independent self-reconfiguration. In *Proceedings of IEEE/RSJ International Conference on Robots and Systems, (IROS)*, pages 2062–2067, Sendai, Japan, 2004.

[21] K. Tomita, S. Murata, H. Kurokawa, E. Yoshida, and S. Kokaji. A self-assembly and self-repair method for a distributed mechanical system. *IEEE Transactions on Robotics and Automation*, 15(6):1035–1045, Dec 1999.

[22] C. Ünsal and P.K. Khosla. Mechatronic design of a modular self-reconfiguring robotic system. In *Proceedings, IEEE International Conference on Robotics & Automation (ICRA)*, pages 1742–1747, San Francisco, USA, 2000.

[23] S. Vassilvitskii, J. Kubica, E. Rieffel, J. Suh, and M. Yim. On the general reconfiguration problem for expanding cube style modular robots. In *Proceedings of the 2002 IEEE Int. Conference on Robotics and Automation (ICRA)*, pages 801–808, 11-15 May 2002.

[24] Eiichi Yoshida, Satoshi Murata, Kohji Tomita, Haruhisa Kurokawa, and Shigeru Kokaji. An experimental study on a self-repairing modular machine. *Robotics and Autonomous Systems*, 29(1):79–89, 1999.

[25] V. Zykov, E. Mytilinaios, B. Adams, and H. Lipson. Self-reproducing machines. *Nature*, 435:163–164, May 12 2005.