Self-assembly of Mobile Robots: From Swarm-bot to Super-mechano Colony

Roderich Groß [a,b], Marco Dorigo [a], and Masaki Yamakita [b]
[a] *IRIDIA, Université Libre de Bruxelles, Belgium*
[b] *Yamakita Lab, Tokyo Institute of Technology, Japan*

Abstract.
Up to now, only a few collective or modular robot systems have proven capable of letting separate and autonomous units, or groups of units, self-assemble. In each case, ad hoc control algorithms have been developed. The aim of this paper is to show that a control algorithm for autonomous self-assembly can be ported from a source multi-robot platform (i.e., the swarm-bot system) to a different target multi-robot platform (i.e., a super-mechano colony system). Although there are substantial differences between the two robotic platforms, it is possible to qualitatively reproduce the functionality of the source platform on the target platform—the transfer neither requires modifications in the hardware nor an extensive redesign of the control.

The results of a set of experiments demonstrate that a controller that was developed for the source platform lets robots of the target platform self-assemble with high reliability. Finally, we investigate mechanisms that control the patterns formed by autonomous self-assembly.

Keywords. Self-assembly, self-reconfigurable robots, pattern formation, collective robotics, swarm robotics, super-mechano colony, swarm-bot, swarm intelligence

1. Introduction

Self-assembly is a reversible processes by which discrete entities bind to each other without being directed externally. It may involve components from molecular scale (e.g., DNA strands forming a double helix) to planetary scale (e.g., weather systems) [23]. The study of self-assembling systems attracted much interest as it may lead to self-constructing, self-repairing and self-replicating machines [21,26,11].

Self-assembly has been widely observed in social insects [1]. Insects physically connect to each other to form aggregate structures with capabilities exceeding those of an individual insect. *Oecophylla longinoda* worker ants, for example, form connected living bridges which other colony members traverse [15]. Understanding how insects use self-assembly to manipulate objects or to navigate all-terrain may have strong implications for robotic systems design.

Self-reconfigurable robots [24,20] hold the potential to self-assemble and thus to mimic the complex behavior of social insects. In current implementations [18,24,20, 14], however, single modules usually have highly limited autonomous capabilites (when compared to an insect). Typically, they are not equipped with sensors to perceive the environment. Nor, typically, are they capable of autonomous motion. These limitations, common to most self-reconfigurable robotic systems, make it difficult to let separate modules, or groups of modules, connect autonomously. In some systems, self-assembly was demonstrated with the modules being pre-arranged at known positions [25,26]. Rare instances of less constrained self-assembly have been reported which are detailed in the following:

- Fukuda *et al.* [9] demonstrated self-assembly among cells of the CEBOT system [10]. In an experiment, a cell approached and connected with another one. Communication among connected cells was studied to enable the group to approach and connect with additional cells [8].
- Bererton *et al.* [3] studied docking in the context of self-repair. One robot was equipped with a fork-lift mechanism to replace a component of its teammate. Successful docking was demonstrated for distances of up to $30\,\mathrm{cm}$ and angular displacements of up to $30°$. Image processing was accomplished on an external PC.
- Rubenstein *et al.* [19] demonstrated the ability of two modular robots to self-assemble. Each robot consisted of a chain of two linearly-linked CONRO modules [5]. The robot chains were set up at distances of $15\,\mathrm{cm}$, facing each other with an angular displacement not larger than $45°$.
- White *et al.* [22], Griffith *et al.* [11], and Bishop *et al.* [4] studied simple, programmable parts capable of self-assembling in the context of self-replication and pattern formation. The parts slid passively on an air table and bound to each other upon random collisions.
- Groß *et al.* [12] demonstrated self-assembly of 16 physical robots using the *swarm-bot* system [16,7]. The robots self-assembled from arbitrary initial positions and on different types of terrain. At present, swarm-bot is the state of the art in autonomous self-assembly.

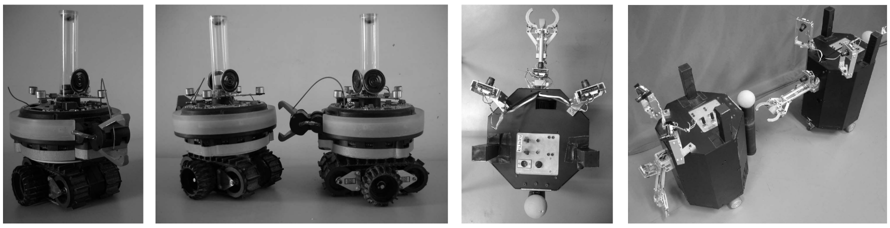
The aim of this paper is to show that a control algorithm for autonomous self-assembly can be ported from a source multi-robot platform to a different target multi-robot platform. The source platform is the swarm-bot system [16,7]. The target platform is a super-mechano colony (SMC) system [6]. Although there are substantial differences between the two robotic platforms, it is possible to qualitatively reproduce the functionality of the source platform on the target platform. This requires neither modifications in the hardware nor an extensive redesign of the control.

The paper is organized as follows. Section 2 overviews the hardware of the source and target platforms. Section 3 presents the control as implemented on the source platform, and details the transfer to the target platform. Section 4 presents the results of a set of experiments. Finally, Sections 5 and 6 discuss the results and conclude the paper.

2. Hardware

2.1. Source Platform (Swarm-bot)

Fig. 1a shows an *s-bot*, the basic robotic component of the swarm-bot system [16,7, see also `www.swarm-bots.org`]. The s-bot has a diameter of $12\,\mathrm{cm}$, a height of $19\,\mathrm{cm}$



(a) (b) (c) (d)

Figure 1. Swarm-bot: (a) the s-bot and (b) two s-bots connecting to each other. Super-mechano colony: (c) a child robot (top view) and (d) two child robots connecting to each other.

(including the transparent cylinder on top), and weighs approximately $700\,\mathrm{g}$.

The s-bot has nine degrees of freedom (DOF), all of which are rotational, including two DOF for the traction system, one DOF to rotate the s-bot's upper part with respect to the lower part, one DOF for the grasping mechanism of the gripper (located in what we define to be the s-bot's front), and one DOF for elevating the arm to which the gripper is attached. The robot's traction system consists of a combination of tracks and two external wheels, called *treels*©. An s-bot can connect with another by grasping the connection ring (see Fig. 1b). The s-bot can receive connections on more than two thirds of its perimeter. For the purpose of robot to robot communication, the s-bot has eight RGB LEDs.

The s-bot is equipped with a variety of sensors, including 15 proximity sensors distributed around the robot, two optical barriers integrated in the gripper, and a VGA omnidirectional vision system. The control is executed on an XScale board running a Linux operating system at $400\,\mathrm{MHz}$. A battery provides full autonomy.

2.2. Target Platform (SMC)

Super-mechano colony (SMC) [6,17] is a modular robotic concept composed of a single main body, called *mother ship*, and many *child robots* attached to it. Child robots are an integral part of the system's locomotion. They can disband to accomplish separate, autonomous missions, and reconnect once the missions are accomplished. Furthermore, child robots have the potential to connect to each other.

Fig. 1c shows the physical implementation of a child robot of an SMC system [6]. The robot has a diameter of $26\,\mathrm{cm}$, a total height of $51\,\mathrm{cm}$ and weighs $11\,\mathrm{kg}$.

The child robot has five DOF, including two DOF for the traction system, one DOF to rotate the robots' upper part with respect to the lower part, one DOF for elevating a manipulation arm (located in what we define to be the robot's front), and one DOF to open and close a gripper that is attached to the manipulation arm. The traction system consists of two active wheels on the left and the right side, and two passive wheels in the front and the back. Each child robot is equipped with a coupling cylinder in its back that allows for receiving connections from a teammate (see Fig. 1d).

A directional VGA stereo vision system is mounted on top of the robot. An additional camera is attached to the manipulation arm. The vision system can detect the relative position of the mark attached to the top of the coupling cylinder of another robot. The control is executed on an on-board Pentium MMX computer running a Microsoft Windows operating system at $233\,\mathrm{MHz}$. A battery provides full autonomy. In the experiments presented in this paper we used an external power supply instead.

Algorithm 1 Self-assembly module for the source platform

```
1:  set-color-ring(blue)
2:  repeat
3:    if risk-of-stagnation() then
4:      set-traction-system-recovery()
5:    end if
6:    (i_1, i_2) ← camera-feature-extraction()
7:    (i_3, i_4) ← proximity-sensor-readings()
8:    (o_1, o_2, o_3) ← neural-network(i_1, i_2, i_3, i_4)
9:
10:   if (o_3 > 0.5) ∧ grasping-requirements-fulfilled() then
11:     set-gripper(close)
12:     if successfully-connected() then
13:       set-color-ring(red)
14:       halt()
15:     else
16:       set-gripper(open)
17:     end if
18:   end if
19:   set-traction-system(o_1, o_2)
20: until timeout-reached()
```

3. Control Transfer

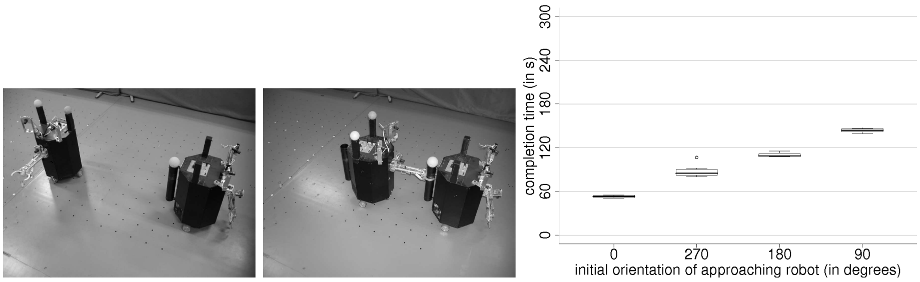
3.1. Control of the Source Platform (Swarm-bot)

Algorithm 1 details the control module for self-assembly. It is assumed that the process is seeded by the presence of a non-moving red object (e.g., the color ring of a robot). A reactive neural network (line 8) constitutes the principal control mechanism. The network takes as input the binary values i_1 and i_2 from the robot's vision system (line 6) and the values i_3 and i_4 from the left-front and right-front robot's proximity sensors (line 7). The network's output (o_1, o_2, o_3) is used to control the speed of the left and the right side of the traction system (line 19) and the connection mechanism (lines 10 to 18). Before being applied to the traction system, o_1 and o_2 are smoothed by a moving average function. By default, the tuple (i_1, i_2) is assigned $(0, 0)$. Any other assignment indicates the presence of a red object (in the front, or to the left or the right side). If an unconnected robot (i.e., a blue object) is perceived in between, i_1 and i_2 are set to zero. To avoid damage to the traction system and to improve the reliability of the control, a recovery move is launched if high torque is continuously present on the traction system (lines 3 to 5). During recovery, the s-bot moves backwards with a small lateral displacement.

The neural network had been shaped by artificial evolution in a computer simulation [13] and subsequently transferred to the physical swarm-bot system [12].

3.2. Transfer to the Target Platform (SMC)

Algorithm 1 describes the control algorithm as it was developed for the source multi-robot platform. In the following, we explain how the sensing and acting functions of



(a) (b) (c)

Figure 2. Self-assembly of two robots: influence of the initial orientation of the approaching robot. Examples of (a) initial and (b) final configurations. (c) Box-and-whisker plot [2] of the completion times (in s) grouped according to the initial orientation of the approaching robot (39 observations in total).

the source platform were realized on the target platform so that Algorithm 1 could be ported without any change. Some functions as, for instance, **neural-network()** remained identical (except for the time required for processing). Many other functions such as **set-traction-system()** could be transferred with minor modifications (e.g., by scaling the speed values to an appropriate range). In the following, we detail those functions which required a different implementation on the target platform to qualitatively reproduce the original function of the source platform.

- **risk-of-stagnation()**: to detect the risk of stagnation of an s-bot, the torque on each side of the traction system was monitored. For the SMC child robot, we use the camera vision system instead. If there is the risk that the left side of the manipulation arm collides with another robot, the recovery move is executed.[1]
- **proximity-sensor-readings()**: as the target platform is not equipped with proximity sensors, we mimic *virtual* proximity sensors heading in the front-left and front-right directions by making use of the vision system. The reading values of the virtual sensors are computed based on the relative position to other robots.
- **grasping-requirements-fulfilled()**, **successfully-connected()**: to test if the grasping requirements are fulfilled, the stereo vision system is used. The system allows for computing the relative position of the coupling cylinder. Consequently, no additional tests must be performed to validate the connection.
- **set-color-ring()**: as the current prototype of the SMC system is not equipped with communication mechanisms other than wireless network, the robots do not signal their connection state. Therefore, each robot can receive connections at any time.

4. Experiments on the Target Platform (SMC)

4.1. Influence of Initial Orientation

We examine the ability of a robot to approach and connect with a passive teammate (see Fig. 2). The two robots have identical hardware. The approaching robot is placed at a

[1]Note that the neural network lets the robot approach the object either straight, or by turning anti-clockwise. If the right side of the manipulation arm collides with the object, the neural network lets the robot retreat as a result of the high reading values from the front-right proximity sensor.

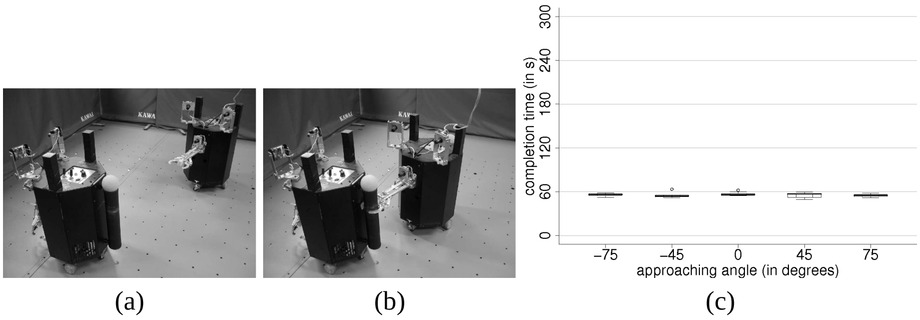


Figure 3. Self-assembly of two robots: influence of the approaching angle. Examples of (a) initial and (b) final configurations. (c) Box-and-whisker plot [2] of the completion times (in s) grouped according to the approaching angle (50 observations in total).

distance of $100\,\mathrm{cm}$ and orientation α with respect to its teammate. The latter is oriented so that its back with the coupling cylinder is heading towards the approaching robot. For each initial orientation $\alpha \in \{0°, 90°, 180°, 270°\}$, 10 repetitions are carried out, thus in total 40 trials are performed. If the robots have not established a connection within $300\,\mathrm{s}$, the trial is stopped.

In 39 out of 40 cases, the robots self-assembled successfully. Fig. 2c shows the observed completion times (in s). If no robot to approach is perceived, the neural network controller lets the robot turn anti-clockwise. This explains the differences in performance. Overall, it seems that the success rate does not depend on the initial orientation of the approaching robot.

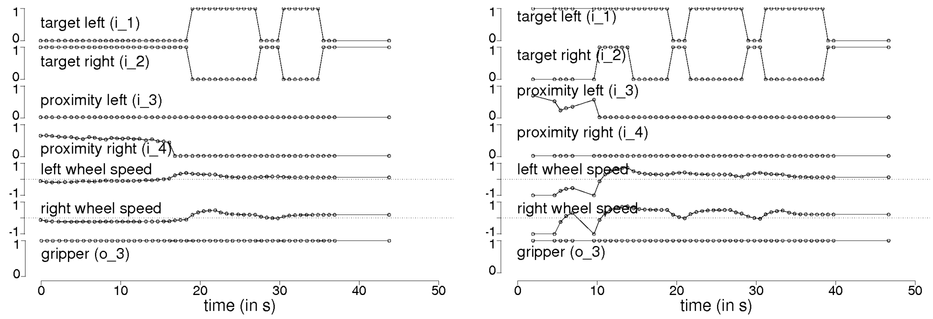
4.2. Influence of Approaching Angle

We examine the ability of a single robot to connect with a passive teammate when approaching it from different angles (see Fig. 3). Due to the mechanical design, the robot cannot connect with the coupling cylinder of the teammate from every angle. In fact, if the angular mismatch between the orientations of the two robots exceeds $85°$, it is impossible to establish a connection. Therefore, potential approaching angles for a successful grasp are limited to the range $[-85°, 85°]$. For approaching angles in the range $[-45°, 45°]$, there should be no difference in the performance as the jaws of the gripper element are not likely to collide with the body of the teammate. The bigger the angular deviation, the more difficult gets the task. We study the approaching angles $\alpha \in \{-75°, -45°, 0°, 45°, 75°\}$. Initially, the approaching robot is oriented towards the teammate. For each angle, ten repetitions are carried out, thus in total 50 trials are performed. If the robots have not established a connection within $300\,\mathrm{s}$, the trial is stopped.

In all 50 cases, the robots correctly self-assembled. Fig. 3c shows the observed completion times (in s). The fluctuations in performance are astonishingly low: all completion times are in the range $[50, 63]$.

4.3. Start Positions That Require to Turn Away or Retreat

We examine the ability of two robots to self-assemble when their starting position and orientation are such that self-assembly is particularly difficult. To create such a situation, we take two robots forming a linear, connected chain and we generate the start positions



(a) (b)

Figure 4. Sensor readings and actuator commands over time (in s) for the two difficult initial arrangements: translation of the approaching robot (a) to the left and (b) to the right.

from this situation via a translation of the grasping robot for $10\,\mathrm{cm}$ to either the left or the right side. These start positions oblige the grasping robots to turn away or retreat before approaching the target. In fact, aligning the robot on the spot in the direction of the target would result in a collision between one side of the manipulation arm and the coupling cylinder.

The robots correctly self-assembled in both situations. Figs. 4a and 4b show the corresponding sensor readings and actuator commands as monitored at the end of each control cycle for the whole duration of the trial. In the first case (see Fig. 4a), the entire situation was handled by the neural network that caused the robot to retreat. In the second case (see Fig. 4b), instead, a recovery move was launched during three control cycles (at time 0, 2, and $7\,\mathrm{s}$).

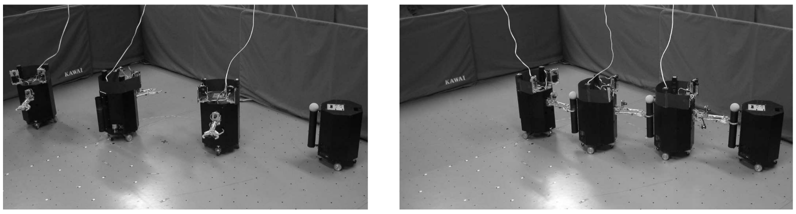
5. Discussion

We have shown that a control algorithm for autonomous self-assembly can be ported from a source multi-robot platform (i.e., the swarm-bot system) to a different target multi-robot platform (i.e., a super-mechano colony system).

A set of experiments demonstrated the validity of the approach. The robots self-assembled reliably—in 91 out of 92 trials the robots correctly established a connection. A robot can approach another and connect to it from a wide range of angles (at least $150°$).[2] The initial orientation of the approaching robot does not affect the reliability. In addition, the control proved capable of dealing with two particular challenging arrangements in which the approaching robot is required to retreat. All experiments have been recorded on video tape (see `http://iridia.ulb.ac.be/~rgross/ias2006/`).

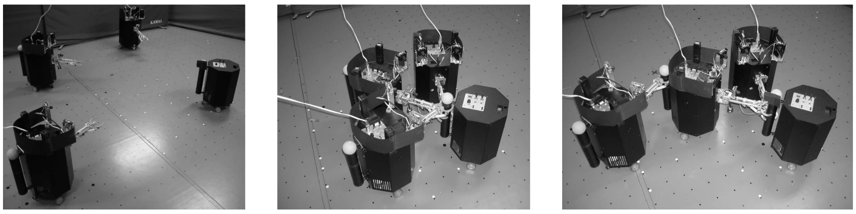
The success can be partially attributed to some design choices made when developing the control for the source platform. First, the use of sensory feedback was reduced to a small set of input variables that appeared to be indispensable to perform the task. For instance, the perceptual range of the robot vision was limited substantially. Consequently, no major difficulties arose when porting the control on the target platform which

[2]Note that also for the source platform the range of possible approaching angles is limited. When starting from outside this range, the approaching robot can search for another position or the target robot may change its orientation.



(a) (b)

Figure 5. A group of four robots self-assembling: (a) initial configuration and (b) final configuration reached after 475 s. Once a robot had completed its task, a visual mark was manually attached to the coupling cylinder.



(a) (b) (c)

Figure 6. Pattern formation: a group of four robots self-assembles starting from a specific initial arrangement, as shown in (a). Depending on the type of visual mark of the robot seeding the process (i.e., the robot on the right side in the figures), different patterns emerge. The final configurations shown in (b) and (c) were reached after 102 s and 207 s, respectively. During the experiments, there was no human intervention.

was equipped with directional vision only. Second, a simple neural network controller was chosen to compute the motor commands based on the sensory feedback. The neural network controller proved robust with respect to changes in the hardware as well as in the provided control functions.

An open issue to address is the problem of scalability. To enable tens or more SMC child robots to self-assemble, we believe that it is beneficial if each robot can signal whether it is connected or not (as it is the case on the swarm-bot platform). Although it is possible to mimic such a function using the existing actuators of the target platform, it might be more appropriate to equip the robot with a communication mechanism (e.g., a controllable LED). To illustrate the use of such a mechanism, we conducted a preliminary set of experiments. Fig. 5 shows a group of four robots self-assembling. In this experiment, the (visual) marks on the top of the coupling cylinder of each robot were attached manually as soon as the robot completed its task. In a second experiment, we adjusted the type of visual mark of the robot seeding the process prior to experimentation. It was shown that depending on the visual mark present, distinct patterns emerged (see Fig. 6). Video recordings are available at `http://iridia.ulb.ac.be/~rgross/ias2006/`.

6. Conclusions

The aim of this study was to show that a control algorithm for autonomous self-assembly can be ported from a source multi-robot platform (i.e., the swarm-bot system) to a different target multi-robot platform (i.e., a super-mechano colony system) if the control algorithm is designed adequately. Although there were substantial differences between the two robotic platforms, we showed that it is possible to qualitatively reproduce the basic functionality of the source platform on the target platform. In fact, the transfer did neither require modifications in the hardware nor an extensive redesign of the control.

The task to let two mobile robots of identical hardware self-assemble starting from random position had so far been demonstrated only with the CONRO system [19] and the swarm-bot [12]. The results presented in this paper demonstrate that the control ported from the swarm-bot to the super-mechano colony lets the robots accomplish the task, while preserving high reliability and low fluctuations in performance. This suggests that the control algorithm is based on some generic principles for the design of self-assembling systems.

Finally, we presented a first attempt to control the patterns that are formed by autonomous self-assembly. An interesting research direction will be the study of mechanisms that lead to patterns that have a functional value for the robotic swarm.

Acknowledgments

This work was accomplished during the visit of Roderich Groß to the Tokyo Institute of Technology in 2005, supported by the Japan Society for the Promotion of Science (JSPS). Marco Dorigo acknowledges support from the Belgian FNRS, of which he is a Research Director, and from the “ANTS” project, an “Action de Recherche Concertée” funded by the Scientific Research Directorate of the French Community of Belgium.

References

[1] C. Anderson, G. Theraulaz, and J.-L. Deneubourg. Self-assemblages in insect societies. *Insect. Soc.*, 49(2):99–110, 2002.

[2] R. A. Becker, J. M. Chambers, and A. R. Wilks. *The New S Language. A Programming Environment for Data Analysis and Graphics*. Chapman & Hall, London, UK, 1988.

[3] C. A. Bererton and P. K. Khosla. Towards a team of robots with repair capabilities: a visual docking system. In *Proc. of the* 7^{th} *Int. Symp. on Experimental Robotics*, volume 271 of *Lecture Notes in Control and Information Sciences*, pages 333–342. Springer Verlag, Berlin, Germany, 2000.

[4] J. Bishop, S. Burden, E. Klavins, R. Kreisberg, W. Malone, N. Napp, and T. Nguyen. Programmable parts: A demonstration of the grammatical approach to self-organization. In *Proc. of the 2005 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 2644–2651. IEEE Computer Society Press, Los Alamitos, CA, 2005.

[5] A. Castano, A. Behar, and P. M. Will. The CONRO modules for reconfigurable robots. *IEEE/ASME Trans. Mechatron.*, 7(4):403–409, 2002.

[6] R. Damoto, A. Kawakami, and S. Hirose. Study of super-mechano colony: concept and basic experimental set-up. *Adv. Robots*, 15(4):391–408, 2001.

[7] M. Dorigo, V. Trianni, E. Şahin, R. Groß, T. H. Labella, G. Baldassarre, S. Nolfi, J.-L. Deneubourg, F. Mondada, D. Floreano, and L. M. Gambardella. Evolving self-organizing behaviors for a *Swarm-Bot*. *Auton. Robots*, 17(2–3):223–245, 2004.

[8] T. Fukuda, M. Buss, H. Hosokai, and Y. Kawauchi. Cell structured robotic system CEBOT—control, planning and communication. In *Proc. of the 2nd Int. Conf. on Intelligent Autonomous Systems*, volume 2, pages 661–671. IOS Press, Amsterdam, The Netherlands, 1989.

[9] T. Fukuda, S. Nakagawa, Y. Kawauchi, and M. Buss. Self organizing robots based on cell structures - CEBOT. In *Proc. of the 1988 IEEE/RSJ Int. Workshop on Intelligent Robots and Systems*, pages 145–150. IEEE Computer Society Press, Los Alamitos, CA, 1988.

[10] T. Fukuda and T. Ueyama. *Cellular Robotics and Micro Robotic Systems*. World Scientific Publishing, London, UK, 1994.

[11] S. Griffith, D. Goldwater, and J. M. Jacobson. Self-replication from random parts. *Nature*, 437(7059):636, 2005.

[12] 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 Verlag, Berlin, Germany, 2006.

[13] R. Groß and M. Dorigo. Group transport of an object to a target that only some group members may sense. In *Proc. of the* 8^{th} *Int. Conf. on Parallel Problem Solving from Nature*, volume 3242 of *Lecture Notes in Computer Science*, pages 852–861. Springer Verlag, Berlin, Germany, 2004.

[14] M. W. Jørgensen, E. H. Østergaard, and H. H. Lund. Modular ATRON: Modules for a self-reconfigurable robot. In *Proc. of the 2004 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, volume 2, pages 2068–2073. IEEE Computer Society Press, Los Alamitos, CA, 2004.

[15] A. Lioni, C. Sauwens, G. Theraulaz, and J.-L. Deneubourg. Chain formation in *Oecophylla longinoda*. *J. Insect Behav.*, 14(5):679–696, 2001.

[16] F. Mondada, L. M. Gambardella, D. Floreano, S. Nolfi, J.-L. Deneubourg, and M. Dorigo. The cooperation of swarm-bots: Physical interactions in collective robotics. *IEEE Robot. Automat. Mag.*, 12(2):21–28, 2005.

[17] K. Motomura, A. Kawakami, and S. Hirose. Development of arm equipped single wheel rover: Effective arm-posture-based steering method. *Auton. Robots*, 18(2):215–229, 2005.

[18] S. Murata, E. Yoshida, A. Kamimura, H. Kurokawa, K. Tomita, and S. Kokaji. M-TRAN: Self-reconfigurable modular robotic system. *IEEE/ASME Trans. Mechatron.*, 7(4):431–441, 2002.

[19] M. Rubenstein, K. Payne, P. Will, and W.-M. Shen. Docking among independent and autonomous CONRO self-reconfigurable robots. In *Proc. of the 2004 IEEE Int. Conf. on Robotics and Automation*, volume 3, pages 2877–2882. IEEE Computer Society Press, Los Alamitos, CA, 2004.

[20] D. Rus, Z. Butler, K. Kotay, and M. Vona. Self-reconfiguring robots. *Commun. ACM*, 45(3):39–45, 2002.

[21] J. von Neumann. *Theory of Self-Reproducing Automata*. Univ. Illinois Press, Urbana, IL, 1966.

[22] P. J. White, K. Kopanski, and H. Lipson. Stochastic self-reconfigurable cellular robotics. In *Proc. of the 2004 IEEE Int. Conf. on Robotics and Automation*, volume 3, pages 2888–2893. IEEE Computer Society Press, Los Alamitos, CA, 2004.

[23] G. M. Whitesides and B. Grzybowski. Self-assembly at all scales. *Science*, 295(5564):2418–2421, 2002.

[24] M. Yim, Y. Zhang, and D. Duff. Modular robots. *IEEE Spectr.*, 39(2):30–34, 2002.

[25] M. Yim, Y. Zhang, K. Roufas, D. Duff, and C. Eldershaw. Connecting and disconnecting for chain self-reconfiguration with PolyBot. *IEEE/ASME Trans. Mechatron.*, 7(4):442–451, 2002.

[26] V. Zykov, E. Mytilinaios, B. Adams, and H. Lipson. Self-reproducing machines. *Nature*, 435(7039):163, 2005.