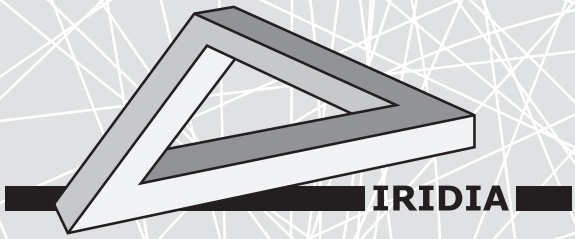


Université Libre de Bruxelles

Institut de Recherches Interdisciplinaires
et de Développements en Intelligence Artificielle

Autonomous Self-assembly in Mobile Robotics

Roderich Gross, Michael Bonani, Francesco Mondada,
and Marco Dorigo

IRIDIA – Technical Report Series

Technical Report No.
TR/IRIDIA/2005-002

March 2005

IRIDIA – Technical Report Series
ISSN 1781-3794

Published by:

IRIDIA, *Institut de Recherches Interdisciplinaires et de Développements en Intelligence Artificielle*
Université Libre de Bruxelles
Av F. D. Roosevelt 50, CP 194/6
1050 Bruxelles, Belgium

Technical report number TR/IRIDIA/2005-002

Revision history:

TR/IRIDIA/2005-002.001 March 2005
TR/IRIDIA/2005-002.002 November 2005

Autonomous Self-assembly in Mobile Robotics

Roderich Groß, Michael Bonani, Francesco Mondada, and Marco Dorigo

***Abstract*— In this paper, we present a comprehensive study on autonomous self-assembly. In particular, we discuss the self-assembling capabilities of the *swarm-bot*, a distributed robotics concept that lies at the intersection between collective and self-reconfigurable robotics. A swarm-bot comprises autonomous mobile robots called *s-bots*. S-bots can either act independently or self-assemble into a swarm-bot by using their grippers.**

We report on experiments in which we study the process that leads a group of s-bots to self-assemble. In particular, we present results of experiments in which we vary the number of s-bots (up to 16 physical robots), their starting configurations and the properties of the terrain on which self-assembly takes place. In view of the very successful experimental results, *swarm-bot* qualifies as the current state-of-the-art in autonomous self-assembly.

***Index Terms*— Self-assembly, collective robotics, self-reconfigurable robotics, swarm robotics, swarm intelligence**

I. Introduction

MODULAR robotics is still progressing very quickly and its systems hold the promise of being flexible and robust [1], [2]. Recently, special attention has been paid to *self-reconfigurable* robots, that is, modular robots whose modules can autonomously organize themselves into different connected configurations. In the majority of current implementations, however, modules have to be pre-assembled by the experimenter before self-reconfiguration can take place. Instead, we are interested in modular systems whose modules are capable of assembling autonomously.

We define the term *self-assembly* as a reversible process by which discrete entities bind to each other without being directed externally. Self-assembly may involve components from molecular scale (e.g., DNA strands forming a double helix) to planetary scale (e.g., weather systems) [3]. Self-assembling robotic systems offer a wide range of uses in the robotics domain, including:

1) **Self-construction and Self-repair**: a self-assembling modular robot may construct and maintain itself exploiting an unstructured source of building blocks;
2) **Self-replication**: a self-assembling modular robot may construct a copy of itself;
3) **Mobility**: a self-assembling modular robot can grow in size by assimilating additional modules, and thereby increase its mobility (e.g., to cross a gap too wide for a single module);
4) **Parallelism**: the modules of a self-assembling modular robot may autonomously detach from each other to accomplish different tasks in parallel;
5) **Force**: a self-assembling modular robot can grow in size by assimilating additional modules, and thereby increase its strength (e.g., a pulling chain that grows until it overcomes the object's resistance to motion).

R. Groß and M. Dorigo are with the *Institut de Recherches Interdisciplinaires et de Développements en Intelligence Artificielle* (IRIDIA) of the Université Libre de Bruxelles, Ave. F. Roosevelt 50, CP 194/6, 1050 Brussels, Belgium (e-mail: {rgross, mdorigo}@ulb.ac.be).

M. Bonani and F. Mondada are with the *Autonomous System Lab* (ASL) of the École Polytechnique Fédérale de Lausanne (EPFL), Office MEB330, building ME, Station 9, CH-1015 Lausanne, Switzerland (e-mail: {michael.bonani,francesco.mondada}@epfl.ch).

In this paper, we present a comprehensive study on autonomous self-assembly with a new mobile self-reconfigurable robotic system called *swarm-bot* [4, see also `http://www.swarm-bots.org`]. The modules comprising a swarm-bot, called *s-bots*, are fully autonomous mobile robots, and have the ability to establish physical connections with each other.

This paper is organized as follows. Section II surveys related work. Sections III and IV contain a description of the robotic hardware and control. Sections V and VI present experimental results obtained on flat and rough terrain. In Section VII we examine to what extent the solutions are applicable to larger group sizes. In Section VIII, we evaluate the results, identify the decisive design factors in making the system successful, and discuss open issues.

II. Literature Review

In self-reconfigurable modular robot systems (e.g., PolyBot [5], CONRO [6], Crystalline [7], M-TRAN [8], and ATRON [9]), modules are equipped with a mechanism that enables them to physically connect to and disconnect from each other. Four different types of such mechanism have been identified [10]: a) pin-hole with a latch, b) matching shape with a sliding latch, c) 3D shape matching by grasping, and d) magnetic.

To date, only a few studies have addressed the problem of letting separate modules, or groups of modules, self-assemble, often with serious constraints. We review studies on self-assembly in chain-based, lattice-based, and mobile self-reconfigurable robots [1] (see Sections II-A, II-B, and II-C, respectively). In these works modules move autonomously. In stochastic self-reconfigurable robots (see Section II-D), by contrast, motion is induced externally. Section II-E identifies the main limitations of the current approaches.

A. Chain-based Self-reconfigurable Robots

1) PolyBot: PolyBot [5], [11] is a modular chain robot that can configure its form with no external mechanical assistance. The modules of the second generation (G2) measure $6.0 \times 7.0 \times 11.0$ (cm) and weigh 416 g. Each module has one degree of freedom involving rotation of two opposite connection plates through a +/- 90 degree range. The connection mechanism falls into the category *pin-hole with a latch*. Additional passive cuboid segments with six connection plates are necessary to

introduce branches to the structures and to connect with an (external) power supply. Active modules are equipped with IR sensors and emitters integrated in the connection plates.

Yim *et al.* [12] demonstrated the ability of a modular robot arm composed of six PolyBot modules to grasp another module on flat terrain. One end of this arm was attached to a wall of the arena. To let the other end reach the (predetermined) position and orientation, the joint angles for each segment were calculated by an inverse kinematics routine. Further alignment and approach was supported by making use of the IR sensors and emitters, and by the mechanical properties of the connection mechanism (pins sliding into chamfered holes).

2) CONRO: CONRO is a homogeneous, modular chain robot [6], [13]. The modules measure $10.8 \times 4.5 \times 4.5$ (cm) and weigh 114 g. Each module comprises a processor, power supply, sensors, and actuators. The basic implementation consists of three segments connected in a chain: a passive connector, a body, and an active connector. The connectors can be rotated with respect to the body. The connection mechanism falls into the category *pin-hole with a latch*. IR emitters and receivers are integrated in the connectors.

Recently, Rubenstein *et al.* [14] demonstrated the ability of two CONRO robots to self-assemble. Each robot consisted of a chain of two linearly-linked CONRO modules. To ensure that the chains perceive each other, they were set up at distances of not more than 15 cm, facing each other with an angular displacement not larger than $45°$. The control was heterogeneous, both at the level of individual modules within each robot and at the level of the modular makeup of both robots.

B. Lattice-based Self-reconfigurable Robots

1) 3D Molecubes: 3D molecubes [15] is a homogeneous, lattice-based self-reconfigurable robot. Each module is a 10-cm cube, one half of it can swivel relative to the other half. The connection mechanism falls into the category *magnetic*. Modules are powered externally.

Zykov *et al.* [16] demonstrated self-replication of a 4-module robot. The system required an ordered supply of additional modules, and it could not adapt to situations in which they were supplied in places that were not predefined.

C. Mobile Self-reconfigurable Robots

1) CEBOT: Fukuda *et al.* proposed the concept of dynamically reconfigurable robotic systems and realized an implementation with CEBOT, the first cellular robotic system [17], [18]. CEBOT is a heterogenous system comprised of cells with different functions (e.g., to move, bend, rotate, and slide). A series of prototypes has been implemented. In CEBOT Mark II, cells measure $17.6 \times 12.6 \times 9.0$ (cm). The weight of a moving cell is 2700 g. The connection mechanism falls into the category *3D shape matching by grasping*. A cone-shaped part fixed on the front of each cell matches a counterpart on the back of each cell to facilitate alignment during approach. The moving cell is equipped with two motorized wheels. Cells can perceive and communicate with other cells by making use of LEDs and photodiodes (one of which can be rotated) [19].

Fukuda *et al.* [20], [21] reported about the successful docking of a cell with another one. Communication among a group of connected cells was studied to enable the group to approach and connect with another cell [22]. To the best of our knowledge there are no quantitative results provided to assess the performance and the reliability of autonomous self-assembly in a group of CEBOT cells.

2) Gunryu: Hirose *et al.* proposed a distributed robotic concept called Gunryu (GR) [23]. Each robot is equipped with a versatile manipulation device and is capable of fully autonomous locomotion. In addition, the manipulator can be employed to establish a physical link with another robot unit. A prototype (GR-I) of two units proved capable of locomotion on rough terrain under conditions in which single units failed [23]. Each unit measures $52.0 \times 40.0 \times 17.5$ (cm). The robots were mechanically linked by means of a passive arm. As a result, the robots were incapable of self-assembly.

3) Repairable Robot Teams: Bererton and Khosla studied self-repair in a team of two autonomous, wheeled robots [24], [25]. The robots measure $10.0 \times 6.0 \times 8.0$ (cm). One robot (the *repair robot*) is equipped with a fork-lift mechanism to install and detach a component of its (stationary) teammate. The connection mechanism is of type *pin-hole*, but without a latch. A black and white camera is mounted on top of the approaching robot.

A simple state machine proved capable of controlling the repair robot to dock with its teammate [24]. The robot could perform the docking for distances up to 30 cm, and for angular displacements up to $30°$. Image processing was performed externally on a PC.

4) Super Mechano Colony (SMC): Super Mechano Colony (SMC) [26], [27] is a modular robotic concept composed of a single main body (called the mother-ship) and several child units attached to it. Child robots are an integral part of the system's locomotion. In addition, the child robots can disband to accomplish separate, autonomous missions, and reconnect once the missions are accomplished. Damoto *et al.* [26] introduced the first prototype of an SMC system. The child robot has a size of 34.5 cm and a weight of 9800 g. Two motorized and two passive wheels provide mobility on flat terrain. Each robot is equipped with a manipulation arm and a connection device of the category *3D shape matching by grasping*. A similar prototype [28] proved capable of changing the connection pattern of three child robots and the mother-ship by letting child robots disconnect, follow a predefined path, and reconnect at a different place.[1] The most recent development is the SMC rover [29]. It is a planetary rover with attachable child robots (called *Uni-Rovers*), each one composed of a single wheel and a manipulation arm (also used as connection mechanism). The wheel is of diameter 19 cm, height 14 cm, and has a weight of 3800 g. The current prototype is not equipped with any sensors.

5) Millibot Trains: Similar to Gunryu, the Millibot Train [30] is composed of multiple, linearly linked, modules. The modules measure $10.9 \times 6.4 \times 4.1$ (cm) and weigh

[1] A video recording is available at `http://www.ac.ctrl.titech.ac.jp/~yamakita/coe/smc.html`.

266 g. Each module is equipped with caterpillar tracks. The connection mechanism falls into the category *pin-hole with a latch*. The prototype is not equipped with any sensors. Therefore, it is not capable of autonomous coupling [30].

D. Stochastic Self-reconfigurable Robots

1) Pattern Forming Parts: Recently, there has been growing attention to the design and study of programmable parts that move passively and bound to each other upon random collision. White *et al.* studied systems in which the parts float passively on an air table that was fixed to an orbital shaker [31]. Squared and triangular modules with side length 6 cm were implemented. Their connection mechanisms fall into the category *magnetic*. The modules have no locomotion abilities and are un-powered. Once they bind to a main structure, they become active. Self-assembly and self-reconfiguration was demonstrated with up to three modules. In another system, the random motion of parts is induced by a surrounding fluid [32]. Self-assembly and self-reconfiguration of two squared modules were demonstrated.

Bishop *et al.* demonstrated self-assembly of modules that slid passively on an air table [33]. The modules are triangular with side length 12 cm. Power is provided on-board. The connection mechanism falls in the category *magnetic*. Once attached, they executed a common graph grammar in a distributed fashion. Doing so, a collection of six programmable parts could form a hexagon.

2) Self-replicating Parts: Griffith *et al.* developed a system capable of self-assembly to study self-replication of strings of programmable parts [34], [35]. They constructed a set of electromechanical components measuring $5 \times 5 \times 1.5$ (cm) and weighting 26 g. The modules slide passively on an air table. The connection mechanism is of type *magnetic* and integrates a latch. The system was capable of the autonomous replication of a 5-bit string provided with an unordered supply of additional units.

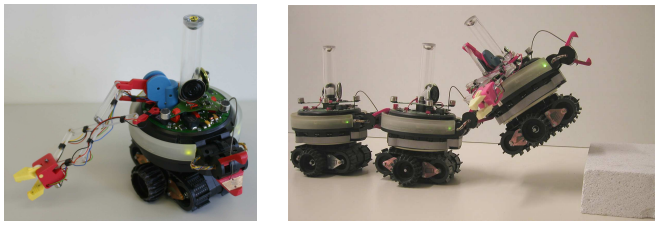
E. Limitations

In all current systems modules require relatively accurate positioning to establish a connection.

In chain-based self-reconfigurable robot systems, single modules have either no or highly limited autonomous mobility [5], [6]. Thus, self-assembly requires a pre-connected chain. The main limitations that make it difficult to let chains self-assemble are a) the complexity of autonomous locomotion, b) imprecision in the joints that results in positional errors which increase with the length of the chain, and c) the lack of complex sensors to support the approaching phase.

In lattice-based self-reconfigurable robot systems, single modules have no autonomous mobility [8], [9]. Once attached together, modules move within a lattice structure. Self-assembly requires a pre-connected seed of modules and a well-ordered supply of additional modules [16]. Typically, the lack of complex sensors makes it difficult to connect to modules in other than predefined places.

Mobile self-reconfigurable robots take advantage of the inherent mobility of individual modules. However, most of the current implementations are incapable of self-assembly as they lack some acting ability [23], sensory abilities [29], [30], or computational resources [24].



(a) (b)

Fig. 1. The swarm-bot concept: (a) a single s-bot, (b) three connected s-bots forming a swarm-bot able to self-reconfigure its shape, in this case, to climb a step too difficult for a single s-bot.

In stochastic self-reconfigurable robots, the modules require a medium in which they move passively. Thus, the modules are not capable of autonomous self-assembly. However, requirements for actuators, sensors, and computational resources are typically low. Power needs not to be provided on-board. Due to these minimalist demands, such systems have a high potential to be produced both in large numbers and at small scales.

III. Hardware Design

Swarm-bot is a new distributed robotic concept lying in between collective and self-reconfigurable robotics [36], [37]. The modules comprising a swarm-bot, called *s-bots*, are fully autonomous and mobile. However, they can also connect to each other to form versatile structures that can self-reconfigure their shape.

Fig. 1(a) shows the physical implementation of the s-bot. The total height is 19 cm. If the two manipulation arms and the transparent pillar on top of the s-bot are unmounted, the s-bot fits into a cylinder of diameter 12 cm and of height 12 cm. The weight of an s-bot is approximately of 700 g.

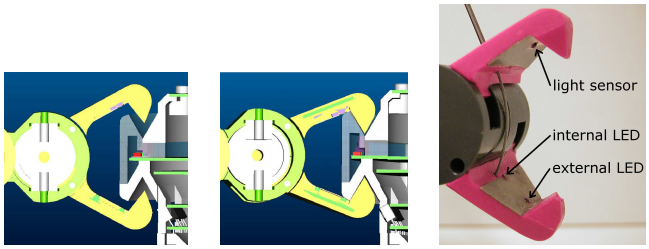
The s-bot has nine degrees of freedom (DOF) all of which are rotational:

- two DOF for the differential *treels*© system—a combination of tracks and two external wheels (see Fig. 1(a)),
- one DOF to rotate the s-bot's upper part (called the *turret*) with respect to the lower part (called the *chassis*),
- one DOF for the grasping mechanism of the rigid gripper (in what we define to be the s-bot's front),
- one DOF for the grasping mechanism of the gripper which is fixed on the flexible arm,
- one DOF for elevating the arm to which the rigid gripper is attached (e.g., to lift another s-bot), and
- three DOF for controlling the position of the flexible arm (not exploited in this experiment).

Most of these DOF are actuated by DC motors equipped with an incremental encoder and controlled in torque, position or speed by a PID controller. Only two DOF (of the flexible arm) are actuated by servo motors. For the purpose of communication, the s-bot is equipped with eight RGB LEDs distributed around the module, and two loudspeakers.

The s-bot is equipped with a variety of sensors:

- 4 proximity sensors fixed underneath (ground sensors),



(a) (b) (c)

Fig. 2. Rigid gripper: (a) loose and (b) tight connection of an s-bot with the connection ring of a teammate. (c) Optical barrier(s) to detect objects to grasp.

- 15 proximity sensors distributed around the turret,
- 4 optical barriers integrated in the two grippers,
- 1 force sensor between the turret and the chassis (2-D traction sensor),
- 1 torque sensor on the elevation arm of the rigid gripper,
- 2 humidity and temperature sensors,
- 3 axis inclinometer,
- 8 light sensors distributed around the module,
- 4 omni-directional microphones, and
- 1 VGA omni-directional camera.

Furthermore, proprioceptive sensors provide internal motor information such as the aperture of the grasping mechanism of the rigid gripper.

When connected in a group, the chassis of an s-bot can be rotated in any (horizontal) direction which allows for coordinated group navigation. The s-bot's actuators and (internal as well as external) sensors allow the group to self-reconfigure its shape in response to the demands of the environment. In the following, we focus on aspects of the hardware which we consider the most relevant to achieve self-assembly. For a more comprehensive description of the s-bot see [4], [10], [37].

A. Morphology and Mechanics

1) Mobility: The s-bot's traction system consists of a combination of tracks and two external wheels, called *treels*©. The tracks allow the s-bot to navigate on rough terrain. The diameter of the external wheels is slightly bigger than the one of the tracks, thus providing the s-bot with good steering abilities. To ensure a stable posture while enabling teammates to approach and connect from many different angles, the geometry of the treels© has been chosen to be roughly cylindrical and of a size comparable to that of the turret.

2) Connection Mechanism: The s-bot is equipped with a surrounding ring matching the shape of the gripper (see Fig. 2). This makes it possible for the s-bot to receive connections on more than two thirds of its perimeter. The design of the connection mechanism allows for some misalignment in all six DOF during the approach phase. A further fine-grained alignment occurs during the grasping phase, favored by the shape of the two teeth at the end of the gripper's jaws as well as the relatively high force by which the gripper is closed (15 N). If the jaws are not completely closed (see Fig. 2(a)), the s-bots maintain some mobility with respect to each other. If the grasp is firm (see Fig. 2(b)), the connection is rigid and can sustain the lifting of another s-bot (see Fig. 1(b)).

B. Sensory Systems

The proximity sensors around the turret can perceive other objects up to a distance of 15 cm. The omni-directional camera can detect s-bots that have activated their LEDs in different colors.

The rigid gripper is equipped with an internal and an external LED as well as a light sensor (see Fig. 2(c)). To test whether an object for grasping is present, two measurements are taken. One with only the external LED being active, and one with no LED being active (ambient light). The difference between the reading values indicates whether an object to grasp is present or not.

Once the s-bot has closed the rigid gripper, it can validate the existence of a connection by monitoring the gripper's aperture and the optical barriers. In this way, potential failures in the connection (e.g., no object grasped) can be detected.

By monitoring the torque of the internal motors (e.g., of the treels©), the s-bot gets additional feedback which can be exploited in the control design.

C. Computational Resources and Handling

The motors and sensors are controlled by 13 microchip PIC processors communicating with the main X-scale board via an I2C bus. This board runs a Linux operating system at 400 MHz. The s-bot can be accessed wirelessly to launch programs, and for the purpose of monitoring. The s-bot is equipped with a 10 Wh Lithium-Ion battery which provides more than two hours of autonomy.

IV. Control Design

We aim at controlling a group of s-bots in fully autonomous manner in such a way that they locate, approach, and connect directly with an object that acts as a seed, or with other s-bots already connected to the seed. To design, implement, and evaluate controllers for the s-bots, we have chosen the following methodology:

1) **Simulator Design**: in a first step, a simulation model of the s-bot and its environment is designed. We restrict the model to include only those elements that we consider relevant for solving the task at hand. We define an interface specifying the s-bot's basic sensing and actuating abilities at an abstract level. For instance, the interface includes a binary function that can be called to detect if the s-bot is in a position from which it may grasp an object without any displacement. Once the interface is specified, the functions are implemented in simulation. The simulator design is detailed in Section IV-A.
2) **Control Design in Simulation**: in a second step, controllers are designed in simulation. They use the functions provided by the interface to the simulation model.
   To design controllers that let swarms of s-bots (i.e., ten or more s-bots) cooperatively accomplish complex tasks we make use of *natural computation* techniques such as swarm intelligence and evolutionary computation. In particular, we emphasize the following properties of our control.

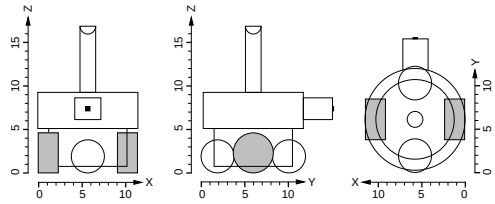
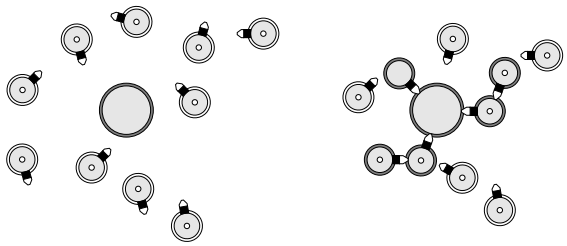


Fig. 3. The simulation model of the s-bot: front, side, and top view (sizes in cm).



(a) (b)

Fig. 4. Group of s-bots self-assembling and connecting to a prey, which in this case also acts as a seed for the process of self-assembling.

- **Decentralized Control**: the s-bots are controlled in a fully autonomous and distributed manner.
- **Homogeneous Control**: each s-bot is equipped with identical control.
- **Locality of Sensing/Action**: each s-bot makes use only of local sensing and acting abilities. No explicit communication or synchronization is present. The environment is the only resource that is shared by the s-bots (e.g., no global communication channels of limited bandwidth).

Due to these properties, the controller is applicable to robotic swarms of any (finite) size. In Section VII we examine how the performance scales with the group size. The control design in simulation is detailed in Section IV-B.

3) **Transfer to Reality**: the functions of the abstract interface are implemented on the physical s-bot. For instance, the binary function that can be called to detect if the s-bot resides in a position in which it may grasp an object was implemented using the camera and the optical barrier sensors of the connection mechanism. During the transfer, adjustments can become necessary to account for issues that have not been properly modeled in simulation.
The transfer to reality is detailed in Section IV-C.

A. Simulator Design

The simulator models the dynamics and collisions of rigid, partially linked, bodies in 3D. The simulation model of the s-bot is illustrated in Fig. 3. It is composed of six bodies: two spherical wheels in the front and the back, two cylindrical wheels on the left and the right, a cylindrical chassis and a turret. The turret is composed of several parts that are rigidly linked: a cylindrical body, a protruding cuboid with a small contact plate in its front (the connection mechanism), and a pillar fixed on top (representing the camera system). The spherical wheels are linked to the chassis via ball-and-socket joints. The cylindrical wheels and the turret are linked to the chassis via hinge joints.

In the following, the interface specifying the acting and sensing abilities is detailed.

1) Actuators: The simulated s-bot is equipped with several actuators. The cylindrical wheels are motorized; the angular speed (in rad/s) can be set to any value within the range $[-M, M]$, where $M = 8$. The turret of the s-bot can rotate with respect to the chassis by means of a motorized axis. Fig. 3 shows the turret's default orientation (i.e., no angular offset is present). The angular offset (in rad) can be set to any value in $[-\pi, \pi]$. The angular speed (in rad/s) is 2.

The connection mechanism is represented by the cuboid heading forward with a small contact plate in the front. In *gripping* mode, a rigid connection is established as soon as the contact plate touches a grippable object. There are two types of grippable objects: the turret of another s-bot, and a cylindrical passive object, called the *prey*. Both are equipped with a surrounding color ring that can emit light of blue or red color. In this study, the prey's ring light is always set to red, and each s-bot uses its ring color to signal whether an object is gripped (red color) or not gripped (blue color).

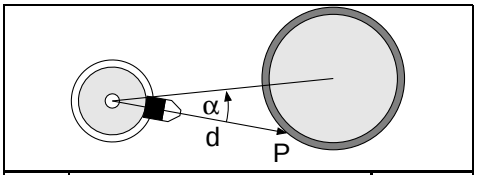
To account for the imprecise and unpredictable behavior of real hardware, the wheels and turret rotation actuators are affected by random noise. In addition, the speed is different for each wheel since a different random bias is present for each wheel during each trial.

2) Sensors: In simulation, the s-bot is provided with the following sensing abilities:

- **Connection Sensors**: the s-bot can detect whether it is in a position from which it may grasp an object without any displacement (i.e., the grasping requirements are fulfilled). Moreover, the s-bot can detect whether it is connected or not.
- **Proximity Sensors**: the turret of the s-bot is provided with 15 proximity sensors that are positioned as on the physical s-bot.
- **Camera**: the camera can detect the presence of colored objects (i.e., other s-bots or the prey) up to a distance of $D_{\max} = 60\,\mathrm{cm}$.

B. Control Design in Simulation

The process of self-assembling is governed by the attraction and the repulsion among s-bots, and between s-bots and the seed (see Fig. 4). The color ring of the seed is permanently activated in red (illustrated in the figure by a gray ring), the color ring of each s-bot is activated either in red or in blue (illustrated in the figure by a gray and a white ring, respectively). Initially, all s-bots set the ring color to blue. The controller lets the s-bots avoid blue objects, and approach/connect with red objects. Thus, the process is triggered by the presence of the seed. Once an s-bot has established a connection with a red object, the color of its ring is set to red, attracting unconnected s-bots to connect with it. The basic principle of signaling the state (of being connected or unconnected) allows



rule	conditions		i_1	i_2
1	$d > D_{\text{max}}$	-	0	0
2	$D_{\text{coll}} < d \leq D_{\text{max}}$	-	1	1
3	$d \leq D_{\text{coll}}$	$\alpha \geq 0°$	1	0
4	$d \leq D_{\text{coll}}$	$\alpha < 0°$	0	1

Fig. 5. In simulation, the camera scans for objects on a virtual ray directly ahead of the s-bot. The scan stops at the first (i.e., the closest) intersection point between the ray and another object, if any. If the first detected object is red, then P, d, and α refer to the intersection point, the distance (in cm) to it, and the horizontal angle (in degrees) to the center of the object. In this case, (i_1, i_2) is determined by the rule set above. In all other cases, i_1 and i_2 are set to zero. $D_{\text{coll}} = 20$ is the distance (in cm) between the s-bot and another object under which there is high risk of collision. $D_{\text{max}} = 60$ defines the sensing range (in cm).

the emergence of (global) connection patterns of dimensions far beyond the modules' (local) sensing range.

Algorithm 1 The assembly module

```
1:  activate color ring in blue
2:  repeat
3:      (i1, i2) ← featureExtraction(camera)
4:      (i3, i4) ← sensorReadings(proximity)
5:      (o1, o2, o3) ← f(i1, i2, i3, i4)
6:
7:      if (o3 > 0.5) ∧ (grasping requirements fulfilled) then
8:          grasp
9:          if successfully connected then
10:             activate color ring in red
11:             halt until timeout reached
12:         else
13:             open gripper
14:         end if
15:     end if
16:     apply (o1, o2) to traction system
17: until timeout reached
```

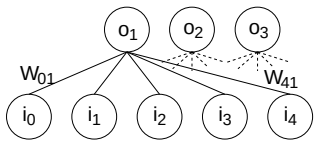
Algorithm 1 describes the control module for self-assembly. Function f (line 5) constitutes the principal control mechanism. This function maps sensory inputs to motor commands. The function takes as input the values i_1 and i_2 from the s-bot's vision system (line 3) and the values i_3 and i_4 from the left-front and right-front s-bot's proximity sensors (line 4). The function'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 16) and the connection mechanism (lines 7 to 15). Sections IV-B.1 and IV-B.2 propose two alternative implementations of function f.

Fig. 5 details the rules to determine the values of the first two function arguments $i_1 \in \{0, 1\}$ and $i_2 \in \{0, 1\}$. By default, the tuple (i_1, i_2) is set to $(0, 0)$. As illustrated in Fig. 5, the camera scans for the first colored object in front of the s-bot. If a red object is detected, (i_1, i_2) indicates its presence and coarse orientation.

TABLE I

RULE-BASED IMPLEMENTATION OF FUNCTION f.

rule	i_1	i_2	i_3	i_4	o_1	o_2	o_3
1	0	0	*	*	s_1	$1 - s_1$	1
2	1	1	*	*	1	1	1
3	1	0	*	*	s_2	s_3	1
4	0	1	*	*	s_3	s_2	1



(a)

$$o_j = \frac{1}{1 + e^{-x_j}}$$

$$x_j = \sum_{n=0}^{4} \omega_{nj} i_n$$

(b)

Fig. 6. (a) A graphical representation of the feed-forward two-layer artificial neural network (i.e., a perceptron [38]) of the assembly module. i_1, i_2, i_3, and i_4 are the nodes which take input from the s-bot's sensors. i_0 is the bias term. o_1, o_2, and o_3 are the output nodes. (b) The equations used to compute the network output values.

1) Rule-based Solution: Table I specifies a parameterized set of rules that defines the function f, mapping sensory inputs from the vision system (i_1 and i_2) and the proximity sensors (i_3 and i_4) to motor commands to control the speed of the left and the right side of the traction system (o_1 and o_2, respectively) as well as the connection mechanism (o_3). A speed value of 1 (0) corresponds to the maximum speed forward (backward) M. The parameter $s_1 \in (0.5, 1]$ specifies the speed with which the s-bot turns on the spot, if no red object is perceived (rule 1). If a red object is perceived but it is more than $D_{\text{coll}} = 20\,\text{cm}$ away, the s-bot moves forward with maximum speed (rule 2). If the red object is close and more to the left (rule 3) or to the right (rule 4), the parameters $s_2 \in [0.5, 1)$ and $s_3 \in [s_2, 1]$ specify to what extent the s-bot turns in the appropriate direction during approach. In any case, o_3 is set to 1, that is, the s-bot tries to establish a connection as soon as the grasping requirements are fulfilled.

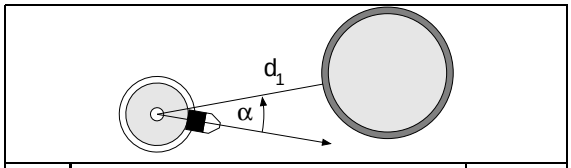
The rule-based controller does not take the inputs from the proximity sensors (i_3 and i_4) into account. Nevertheless, unconnected s-bots that reside between the s-bot itself and the object to approach are perceived as blue objects and thus shadow the presence of the red object (see caption of Fig. 5).

We assessed the quality of different parameter assignments by performing 200 simulation trials in which 2, 4, 6, or 8 s-bots had to self-assemble with a prey. 1000 different assignments to the parameter set (s_1, s_2, s_3) were assessed, and the one exhibiting the highest average performance was selected $(0.85, 0.60, 0.85)$.

TABLE II

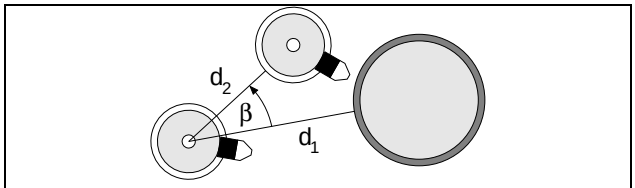
WEIGHTS OF THE NEURAL NETWORK IMPLEMENTING FUNCTION f.

ω_{nj}	i_0	i_1	i_2	i_3	i_4
o_1	-0.7274	2.7010	4.1814	-18.3508	-6.0088
o_2	1.7611	6.5784	-2.0708	-1.2650	-1.1433
o_3	7.4983	-0.7223	1.7492	0.7708	1.9560



rule	conditions		i_1	i_2
1	$d_1 > D_{\text{coll}}$	$\lvert\alpha\rvert \geq 20°$	0	0
2	$D_{\text{coll}} < d_1 \leq D_{\text{max}}$	$\lvert\alpha\rvert < 20°$	1	1
3	$d_1 \leq D_{\text{coll}}$	$0° \leq \alpha < 45°$	1	0
4	$d_1 \leq D_{\text{coll}}$	$-45° < \alpha < 0°$	0	1

Fig. 7. On the physical s-bot, the perceptual range for detecting red objects to approach has been extended to $45°$ to the left and right side of the s-bot's front. If no red block resides in this range, or if an obstacle (a blue block; for details see next figure) is present, i_1 and i_2 are set to zero. Otherwise, (i_1, i_2) is determined by the rule set above. d_1 and α (in degrees) correspond to the distance of, and the direction to, the closest red block within the perceptual range.



rule	conditions		i_1	i_2
1	$(d_2 < d_1) \wedge (d_2 \leq D_{\text{coll}})$	$-90° < \beta < 60°$	0	0
2	$(d_2 < d_1) \wedge (d_2 > D_{\text{coll}})$	$-25° < \beta < 25°$	0	0

Fig. 8. Rule set defining whether an obstacle is present. If in addition to the red block at distance d_1 there exists a blue block at distance d_2 and with angular displacement β, and if rules 1 or 2 are satisfied, then an obstacle is present. In this case, i_1 and i_2 are set to zero. The range of angles satisfying rule 1 was chosen to be asymmetrical in order to avoid potential deadlocks between two s-bots approaching the same object simultaneously.

2) Evolved Solution: As an alternative to the rule-based solution for mapping the sensory inputs to motor commands, we designed an artificial neural network. As illustrated in Fig. 6, the neural network has a bias node i_0, four input nodes i_1, i_2, i_3, and i_4, three output nodes o_1, o_2, and o_3, and 15 connection weights $\omega_{nj}, n \in \{0, 1, 2, 3, 4\}, j \in \{1, 2, 3\}$. i_0 is set to 1 by default. i_1, i_2, i_3, and i_4 take input from the s-bot's sensory system. The neural network computes the outputs o_j for the motor commands based on the weights w_{nj} and the inputs i_n as detailed in Fig. 6. The weights of the neural network are listed in Table II. They were shaped by artificial evolution in the context of a cooperative transport task [39]. The evolutionary algorithm used was a self-adaptive $(\mu + \lambda)$ evolution strategy [40], [41].

C. Transfer to Reality

We have ported the interface providing the sensing and acting abilities as well as the controller from simulation to the physical s-bot. In the following, we detail the implementation aspects involved.

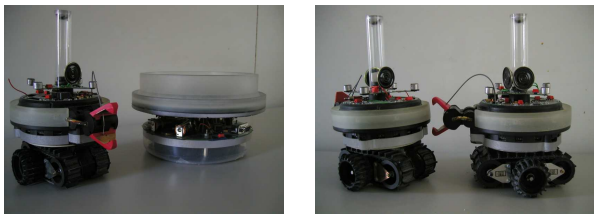
- To prevent the s-bot's traction system from being damaged, the internal motor torque values are monitored. If high torque is continuously present for a sequence of $P = 6$ control steps, a recovery move is executed. This may happen if an object collides with the s-bot's gripper and prevents the s-bot from moving forward or turning to a side. During recovery, the s-bot moves about $5\,\text{cm}$ backwards with a small lateral displacement. Each time a recovery move is executed the side of the lateral displacement (i.e., to the left or to the right) is changed.
- The camera image is partitioned into small rectangular blocks. For each block, it is determined if the color red or blue is prevalent. Colored blocks of the image correspond to different parts of the color ring of an s-bot or of the prey. Given the s-bot camera characteristics and the limited computational power available, it is difficult, if not impossible, to determine whether two blocks of equal color belong to the same object (e.g., an s-bot). Therefore, the procedure for feature extraction (line 3 of Algorithm 1) is implemented differently than in simulation. Figs. 7 and 8 detail the rules to determine the values of the arguments i_1 and i_2 of the mapping function f. The distance measure is based on the camera image frame. Due to imprecision in, and differences between, the hardware of different s-bots, it is difficult to estimate the corresponding distances in the real world. There is no explicit limit for the sensing range (i.e., $D_{\text{max}} = \infty$). The software we use to detect colored objects makes it possible to recognize red (blue) objects up to a distance of $70 - 90\,\text{cm}$ ($35 - 60\,\text{cm}$), depending on which s-bot is used.
- The connection mechanism is controlled in lines 7 to 15 of Algorithm 1. The gripper is closed if a set of requirements is fulfilled (see also Fig. 7):
  - $o_3 > 0.5$,
  - $(i_1 \neq 0) \vee (i_2 \neq 0)$,
  - $d_1 \leq D_{\text{grasp}}$,[2]
  - $|\alpha| \leq 30°$,
  - no connection attempt failed within the last 18 control steps (i.e., approximately $3\,\text{s}$).

  If these requirements are fulfilled, the gripper optical barrier is used to detect whether an object is present between the two jaws of the gripper (see Section III-B). If this is the case, the procedure closes the gripper. While closing, the gripper is slightly moved up and down several times to facilitate a tight connection. By monitoring the gripper aperture (line 9 of Algorithm 1), failures of the connection procedure can be detected. In this case the gripper is opened again.
- The speed vector for the traction system is applied in line 16 of Algorithm 1. To do so, the values o_1 and o_2 are scaled in the range $[-M, M]$. The maximum speed M is set according to the following rule:

$$M = \begin{cases} M_1 & \text{if } (i_1 = 0) \wedge (i_2 = 0); \\ M_2 & \text{if } d_1 \leq D_{\text{grasp}}; \\ M_3 & \text{if } D_{\text{grasp}} < d_1 \leq D_{\text{coll}}; \\ M_4 & \text{if } d_1 > D_{\text{coll}}. \end{cases} \quad (1)$$

After some preliminary experimentation, we have chosen

[2] D_{grasp} is an estimate of the maximum distance to an object that can still be grasped.



(a) (b)

Fig. 9. A single s-bot self-assembling with (a) an object and (b) a teammate.

the values $M_1 = 8$, $M_2 = 5$, $M_3 = 10$, and $M_4 = 20$. A value of 20 corresponds approximately to a speed of $6.5\,\mathrm{cm/s}$.

Once the speed vector has been scaled accordingly, a moving average function smoothes the speed values over time in order to avoid hardware damage by potentially oscillating speed settings.

V. Experiments on Flat Terrain

In the following, we examine the ability of the physical s-bot to self-assemble when moving on flat terrain. We employ the evolved solution for the mapping function f (see Section IV). This solution was experimentally shown, see [39], to be superior in performance to the rule-based solution especially if applied to the control of large groups of s-bots (see also Section VII-B). Details on the policy used in the following experimentation can be found in the appendix.

A. One S-bot and a Static Object

1) Experimental Setup: We examine the ability of a single s-bot to approach and connect with the prey (see Fig. 9(a)). The prey is equipped with a color ring of the same shape as the grippable ring of the s-bots. The ring has a diameter of $20\,\mathrm{cm}$ and is positioned $0.5\,\mathrm{cm}$ higher than the ring of the s-bots. Its color is set to red. Initially, the s-bot is put at a distance $d \in \{25, 50\}$ (in cm) with orientation $\alpha \in \{0°, 90°, 180°, 270°\}$ with respect to the prey. The distance is computed between the centers of the two objects. For each combination of d and α, five repetitions are carried out, thus in total 40 trials are performed. If the s-bot does not succeed in establishing a physical connection within $300\,\mathrm{s}$, the trial is stopped.

2) Results: We repeated the experiment with four different s-bots. In all 160 trials, the s-bots succeeded in approaching and connecting with the prey. This high reliability is partly due to the recovery move (see Section IV-C): in 14 cases during this experiment an s-bot monitored high torque reading values for its traction system, and launched the recovery move to prevent the traction system from potential damage. This usually occurred if the protruding rigid gripper collided with the prey and prevented the s-bot from further alignment. Every time this happened, the s-bot was able to detect this stagnation situation and the simple recovery move allowed the s-bot to approach again the object from a different direction.

Fig. 10 plots the observed completion times (in seconds), that is, the total time elapsed until the s-bot was successfully connected. The average completion time for the 80 trials with distance $25\,\mathrm{cm}$ ($50\,\mathrm{cm}$) is $22.6\,\mathrm{s}$ ($34.9\,\mathrm{s}$).

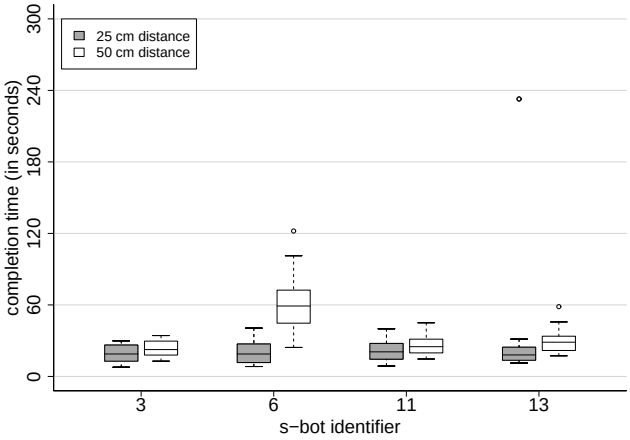


Fig. 10. Self-assembly of a single s-bot with a prey. Box-and-whisker plot [42] of the completion times (20 observations per box) grouped according to the s-bot involved and its initial distance from the prey.

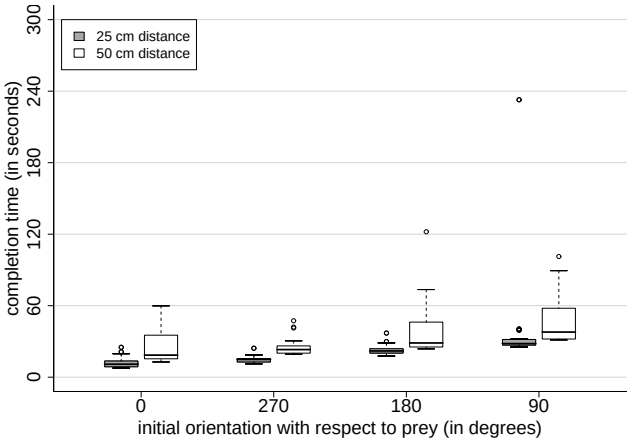


Fig. 11. Self-assembly of a single s-bot with a prey. Box-and-whisker plot [42] of the completion times (20 observations per box) grouped according to the s-bot's initial orientation and distance with respect to the prey.

Note that there were substantial differences in the hardware among the s-bots (e.g., s-bot 3, 6, and 11 were equipped with a camera different from the one used by s-bot 13).[3]

S-bot 6 performed significantly worse than the other s-bots given a starting distance of $50\,\mathrm{cm}$ (see Fig. 10). We observed that the camera images of s-bot 6 were of bad quality when compared to the other s-bots. Therefore, s-bot 6 sporadically could not detect the prey at a distance of $50\,\mathrm{cm}$. Nevertheless, s-bot 6 succeeded in all 20 trials to connect starting from this distance. Except for this single case, the four s-bots exhibit similar performances.

Fig. 11 shows the same observations grouped according to the s-bot's initial orientation and distance with respect to the prey. The neural network causes the s-bot to turn anti-clockwise if it does not get any input about objects to approach. This explains the differences in performance for different initial orientations with respect to the prey.

B. One S-bot and a Static Teammate

1) Experimental Setup: In this section we examine the ability of an s-bot to approach and connect to a teammate

[3] S-bots are labeled from 1 to 35.

(a) (b) (c)

Fig. 12. Illustration of angles in which the static teammate is approached in the two s-bot experiments: (a) $0°$, (b) $60°$, and (c) $120°$.

(see Fig. 9(b)). The teammate does not move and it activates its color ring in red. Initially, the s-bot is put at a distance of $50\,\mathrm{cm}$ heading in the direction of its teammate. The distance is computed between the centers of the two s-bots. If the s-bot does not succeed in establishing a physical connection within $300\,\mathrm{s}$, the trial is stopped.

Unlike the problem of approaching and connecting with the cylindrical prey, the performance in approaching and connecting with a teammate depends on the relative angle of approach. We do not consider approaching angles for which the two s-bots are heading directly towards each other (with their connection mechanisms to the front). Such situation was not present in the (evolutionary) design phase in which controllers were assessed for approaching and grasping the prey or already connected s-bots. One attempt to handle the new situation could be to modify the recovery move (see Section IV-C) so that it ensures a big, irregular lateral displacement before approaching again the object. Another possibility is to prevent other s-bots from approaching a red s-bot within the critical range of angles (for more details see Section VII-A).

We focus on the approaching angles $\alpha \in \{0°, 60°, 120°\}$, where $0°$ corresponds to the target s-bot's tail (see Fig. 12). The approaching angle $60°$ is of special interest, since at this angle a vertical pillar is mounted on the s-bot, which makes it impossible to grasp the ring.

2) Results: For each approaching angle, 20 trials were performed with s-bot 3. In all 60 trials, the s-bot successfully connected. A recovery move was launched six times; in each case the approaching angle was $60°$ and the s-bot's gripper collided with the pillar of the target s-bot. Due to the cylindrical shape of the pillar, the gripper often slid to the left or the right side and could eventually grasp the ring.

Fig. 13 plots the observed completion times (in seconds). The average completion times for the 20 trials with approaching angle $0°$, $60°$, and $120°$ (and initial distance $50\,\mathrm{cm}$) are 17.9, 26.4, and $17.9\,\mathrm{s}$, respectively.

C. A Group of Six S-bots and a Static Object

So far, we have studied situations in which a single s-bot is approaching a single object for grasping. In this section we assess the performance of a group of six s-bots accomplishing self-assembly with the prey as an initial seed. Each s-bot is driven by an identical controller. This is the same controller used in the one s-bot experiments.

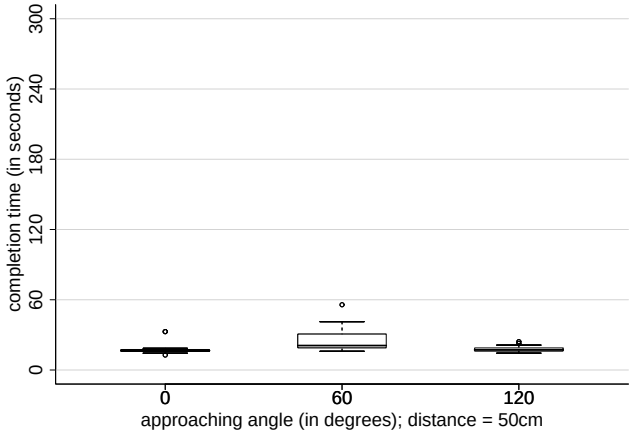
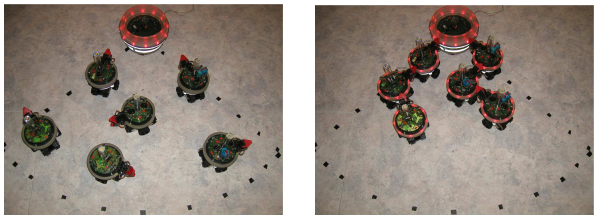


Fig. 13. Self-assembly of an s-bot with a teammate. Box-and-whisker plot [42] of the completion times (20 observations per box).



(a) (b)

Fig. 14. Self-assembly of six s-bots with the prey: (a) initial configuration, and (b) final configuration in a typical trial.

1) Experimental Setup: At the beginning of each trial, the s-bots are placed at arbitrary positions[4] and orientations inside a circle of radius $70\,\mathrm{cm}$ around the prey. To favor interactions among the s-bots, we limited their initial positions to a $90°$ segment of the circle. The same density could be obtained by putting a swarm of 24 s-bots inside a full circle of the same radius. Fig. 14 shows the initial and the final configurations in one typical trial. If the s-bots do not succeed within $600\,\mathrm{s}$, the trial is stopped.

[4]As in simulation, the s-bots are positioned in such a way that there is a minimum distance of $20\,\mathrm{cm}$ between the centers of any two objects. This allows all s-bots to turn on the spot with no collision of their gripper elements.

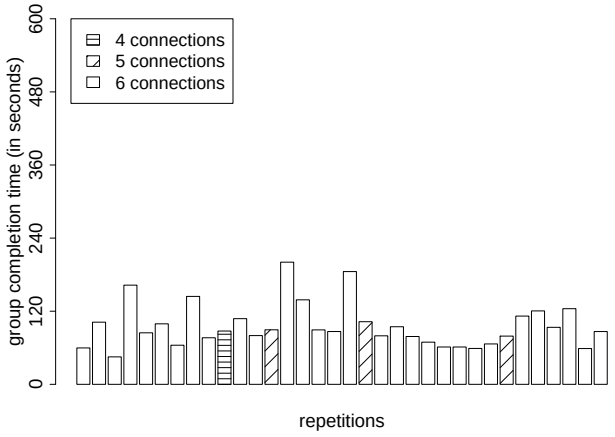
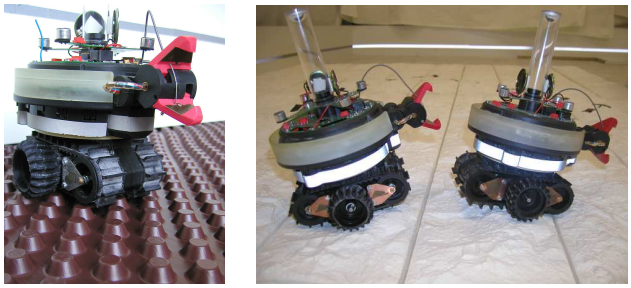


Fig. 15. Self-assembly of six s-bots with a prey (34 repetitions).



(a) (b)

Fig. 16. Types of rough terrain: (a) moderately rough terrain and (b) very rough terrain.

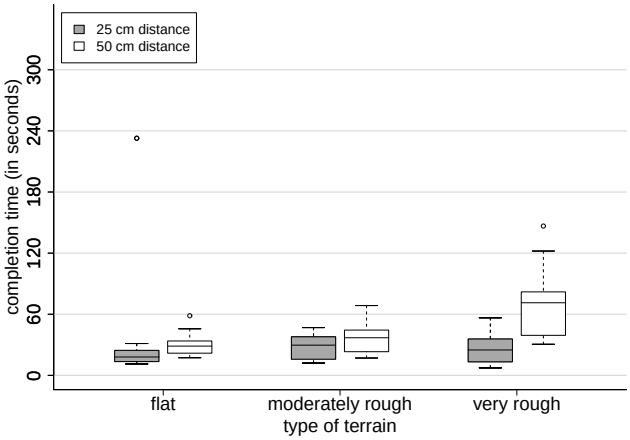


Fig. 17. Self-assembly of one s-bot with a prey. Box-and-whisker plot [42] of the completion times on flat terrain (20 observations per box), *moderately rough terrain* (20 observations per box), and *very rough terrain* (19 observations per box).

2) Results: Fig. 15 shows a bar plot of the 34 trials performed. The pattern of each bar indicates the number of s-bots that could successfully connect within the time frame. The height of the bar represents the number of elapsed seconds until the last s-bot completed connection.

In total, 199 times an s-bot succeeded in establishing a connection, while only five times an s-bot failed. At the end of 30 out of 34 trials, all seven objects were physically connected; on average this took $96.4\,\mathrm{s}$.

VI. Experiments on Rough Terrain

In the previous section we have shown that we can let an s-bot, or a group of six s-bots, self-assemble when moving on flat terrain. The s-bot was designed to perform tasks also under rough terrain conditions. However, the neural network, which is the main part of our controller, was evolved controlling s-bots on flat terrain. In this section, we study to what extent the behavior is disrupted when the roughness of the terrain is increased.

We consider two types of rough terrain (see Fig. 16). Both terrain types are unnavigable for most standard wheeled robots of a similar size. The first terrain type (here referred to as *moderately rough terrain*) has a surface with a regular structure. The second terrain type (here referred to as *very rough terrain*) consists of white plaster bricks providing a very rough, non-uniform surface.

A. One S-bot and a Static Object

1) Experimental Setup: Except for the difference in the terrain, the experimental setup and the control are kept unchanged (see Section V-A).

2) Results: Fig. 17 shows the performance of s-bot 13 for the different types of terrain. For each terrain, 40 trials were performed. In the 80 trials on the flat terrain and the *moderately rough terrain* the s-bot successfully connected to the prey. On the *very rough terrain*, the s-bot failed only once for both initial distances ($25\,\mathrm{cm}$ and $50\,\mathrm{cm}$). In the other 38 trials, the s-bot successfully connected with the prey.

We observed that on the *very rough terrain* the s-bots often launched the recovery move during the approach phase. The roughness of the terrain caused a high torque on the traction system during navigation. Thus, the mechanism to detect stagnation was erroneously activated. During the recovery move, the s-bot moves backwards without recognizing obstacles. In the two cases in which the s-bot failed to complete the task, it got stuck with its back colliding with the prey. A refined version of the controller, which takes obstacles into account during recovery, is introduced in the following section.

B. A Group of Six S-bots and a Static Object

1) Experimental Setup: Except for the difference in the terrain (see Fig. 16), the experimental setup is identical to the one described in Section V-C. In case of the *moderately rough terrain* the controller is kept unchanged. For the *very rough terrain* the original control induced disruptive behavior in the s-bots. The s-bots often collided and toppled down. As discussed in the previous section, we observed that the mechanism to detect stagnation and to launch the recovery move was too sensitive. In addition, during recovery s-bots risked collision with other objects. Therefore, we doubled the threshold P of our control (see Section IV-C) so that the recovery move is executed only if the torque remains high for twelve subsequent control steps (i.e., approximately $2\,\mathrm{s}$). In addition, the four rear facing proximity sensors are monitored during the recovery move, and if a certain threshold is exceeded, the s-bot stops moving backwards. Last but not least we changed the speed parameters (M_1, M_2, M_3, M_4) from $(8, 5, 10, 20)$ to $(10, 8, 10, 20)$ (see Section IV-C).

2) Results: Fig. 18 shows the results obtained in 20 trials on the *moderately rough terrain*. In total, 120 times an s-bot was controlled in this experiment. In 118 cases the s-bot successfully connected.

Fig. 19 shows the results obtained in 20 trials on the *very rough terrain*. In 12 out of 20 trials, all six s-bots connected with the prey. In total, 120 times an s-bot was controlled in order to establish a connection, and in 109 cases it succeeded.

Table III summarizes the results obtained for the experiments with one s-bot (number 13) and a prey, and those with six s-bots and a prey, for the three different types of terrain. Overall, the reliability of the algorithm which was designed to control s-bots on flat terrain is not affected by the roughness of the *moderately rough terrain*. However, 40% additional

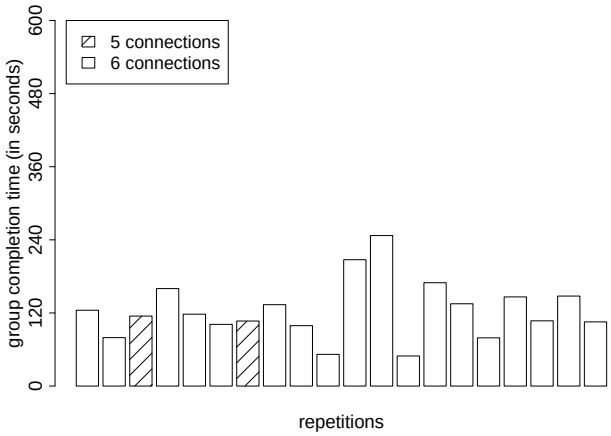


Fig. 18. Self-assembly of six s-bots with a prey on the *moderately rough terrain* (20 repetitions).

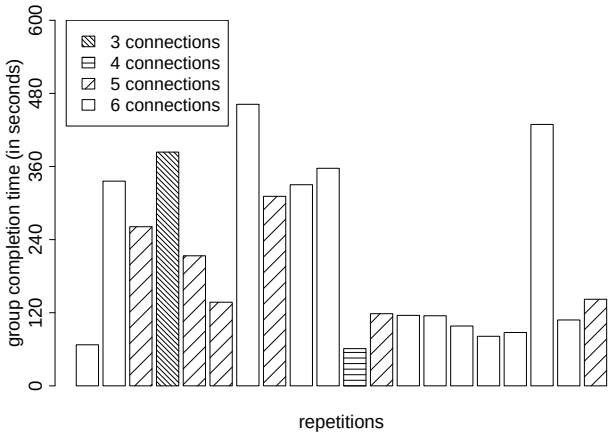
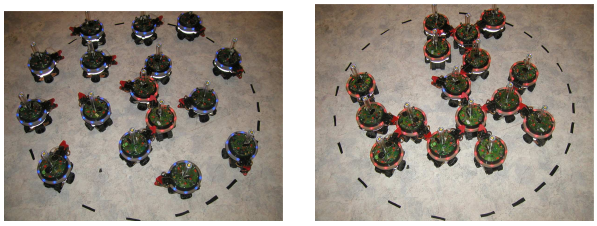


Fig. 19. Self-assembly of six s-bots with a prey on the *very rough terrain* (20 repetitions).

TABLE III

SUMMARY OF RESULTS ON SELF-ASSEMBLY OBTAINED FOR THE EXPERIMENTS WITH ONE S-BOT (NUMBER 13) AND A PREY, AND THOSE WITH SIX S-BOTS AND A PREY. NOTATION: N (GROUP SIZE), D (INITIAL DISTANCE IN CM), C (PERCENTAGE OF CONNECTIONS), T (MEDIAN GROUP COMPLETION TIME IN S; ONLY TRIALS WITH N CONNECTIONS). EACH CONFIGURATION WAS TESTED AT LEAST 20 TIMES (SEE TEXT FOR DETAILS). VALUES MARKED WITH THE *-SYMBOL WERE OBTAINED WITH THE MODIFIED CONTROLLER.

N	D	flat terrain		mod. rough terrain		very rough terrain	
		C	T	C	T	C	T
1	25	100.00	18.0	100.00	29.7	95.00	24.9
1	50	100.00	28.7	100.00	36.9	95.00	71.3
6	< 70	97.55	86.7	98.33	121.2	90.83*	115.4*



(a) (b)

Fig. 20. Self-assembly of 16 physical s-bots put in a circle of radius $50\,\mathrm{cm}$. Trial 12: (a) after $23\,\mathrm{s}$ and (b) after $108\,\mathrm{s}$.

time is required (comparing the median values) to connect all seven objects. Even on the *very rough terrain*, a single s-bot connected in 95% of the cases. Being part of a group of size six, a single s-bot, controlled by the modified controller, connected still in more than 90% of the cases.

The main cause of failure was due to visual misperceptions of the presence and/or angular positions of other objects. On the very rough terrain, s-bots also failed to align with their teammates and therefore could not connect.

VII. SCALABILITY

In this section, we study to what extent our controller allows large swarms of s-bots to self-assemble. First, we present the outcome of an experiment in which we utilized all physical s-bots available at the time of experimentation (in total 16). Then, we present additional results obtained in simulation with swarms of sizes beyond the number of s-bots that have been constructed.

A. Experiments with 16 Physical S-bots

1) Experimental Setup: We study self-assembly with a swarm of 16 s-bots. One s-bot acts as a seed, as after five seconds it stops moving and activates a pattern on its LED ring: the two LEDs in the front are set to blue, while the remaining six LEDs are set to red. In this way, it attracts teammates to approach from any direction other than the front.[5] The teammates are controlled by the (refined) version of the controller that has been detailed in Section VI-B.

The s-bot acting as a seed is put in the center of a circle of radius $50\,\mathrm{cm}$. 15 teammates are placed at arbitrary positions and orientations within the same circle. The s-bots are positioned so that each s-bot can rotate on the spot without colliding with a teammate (i.e., we ensure a minimum distance of $20\,\mathrm{cm}$ between the centers of any two s-bots).

2) Results: We repeated the experiment twelve times. Fig. 20 shows a typical trial. In all but one case, all 16 s-bots successfully assembled to each other (see Fig. 21 for the connection times). In one case a single s-bot entered the connection state without being connected, and another s-bot connected with it; the other 14 s-bots connected with each other. Thus, in total, 190 out of 192 times an s-bot succeeded in task completion.

[5]In fact, the front of the s-bot is unable to passively receive connections from other s-bots due to the location of its own gripper mechanism.

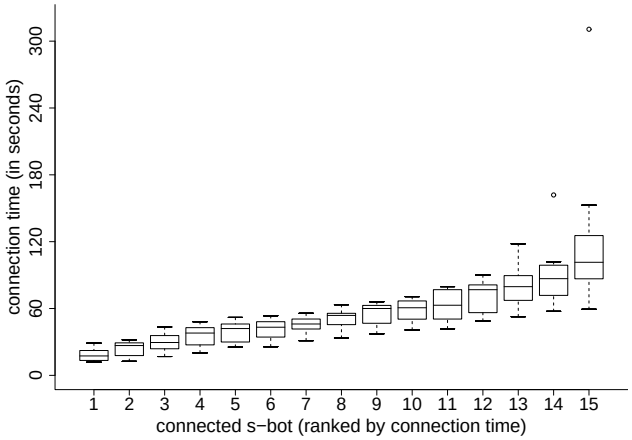


Fig. 21. Self-assembly of 16 physical s-bots. Box-and-whisker plot [42] showing the time at which the ith s-bot connected (observations from the 11 out of 12 trials in which all 16 s-bots successfully self-assembled).

B. Experiments with up to 100 S-bots in Simulation

1) Experimental Setup: We examine the problem of letting groups of 10 to 100 s-bots self-assemble with a static prey. The s-bots are initially placed at random positions and orientations within a circular area around the prey. We vary the radius of the initial area to study to what extent the behavior is affected by the density of s-bots. We define the density of modules as the size of the 2D area covered by the modules divided by the size of the available 2D area. The area size covered by a module (in simulation) is $A = 116\,\mathrm{cm}^2$. For each group size we studied densities of 0.050, 0.075, 0.100, 0.125, 0.150, 0.175, and 0.200. We could not study densities much higher than this as it is impossible to find an initial placement in which the s-bots may turn on the spot without collision.[6]

2) Results: We assess the performance of both the rule-based solution and the neural network based solution for implementing function f of our controller (see Algorithm 1). The performance of both solutions was assessed previously with groups of 4 to 16 simulated s-bots in the context of a cooperative transport task [39].

Figs. 22 and 23 present the percentage of the group that could successfully connect within a time period of 300 seconds for all group sizes and densities in 200 trials; the controller utilizes the rule-based mapping, and the neural network based mapping, respectively.

In case of the two lowest densities (0.050 and 0.075) the performance for both mappings reduces drastically with group size. We observed that, at such low density, some s-bots did not have visual contact with any teammates or with the prey. In addition, many s-bots lost visual contact, since all the teammates left their neighborhood when approaching red objects. For a swarm of s-bots to self-assemble in a situation

[6]To ensure a minimum gap of about $1\,\mathrm{cm}$, the s-bots are put so that a minimum distance of $20\,\mathrm{cm}$ is present between the centers of any two objects. Let us consider the s-bots and the prey as disks of radius r. To pack eleven congruent disks without over-lapping in a unit circle, the disk radius may not exceed $r = 0.2548485$ (for a proof see [43]). This packing would result in a module density of $\frac{10\pi r^2}{\pi - \pi r^2}\frac{A}{\pi 10^2} = 0.256$. If we consider our additional constraint that one disk (i.e., the prey) has to be positioned in the center of the unit circle, the highest possible module density is equal or lower than 0.256.

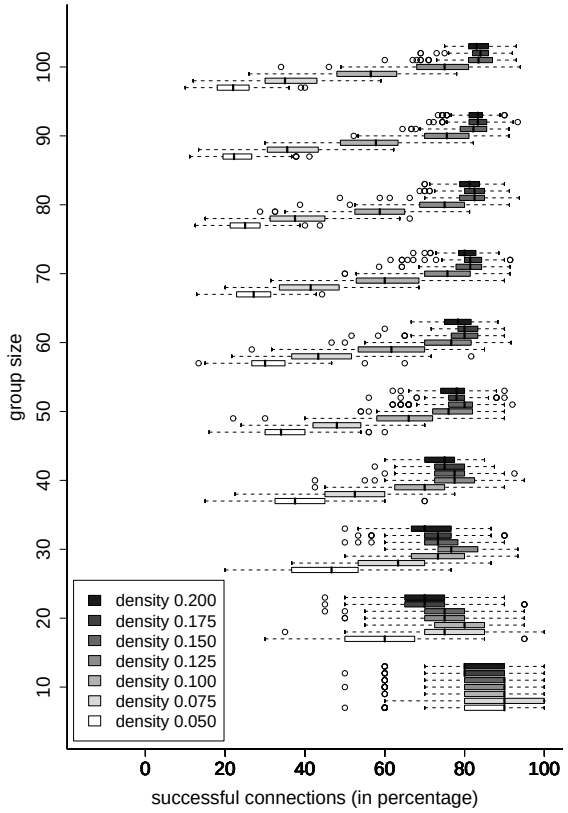


Fig. 22. Box-and-whisker plot [42] showing the percentage of successful connections during self-assembly in a group of 10 to 100 s-bots, for different initial densities (200 observations per box). The s-bots are controlled by the rule-based controller.

in which the module density is particularly low, it could be of advantage to propagate the presence of the prey using a third color (in addition to blue and red), and to use a rule set to let the modules form a cluster. However, in case the s-bots start from positions in which visual contact might not be present, the problem of exploration/aggregation has to be addressed.

For all other densities, the neural network based controller has a particularly high success rate. In contrast, in the case of the rule-based controller, the success rate drops considerably when moving from group size 10 to 20. For increasing group sizes, however, the performance tends to increase.

We now analyze the relationship between the time needed for an s-bot to connect and the group size. We measure the average time for an s-bot to self-assemble in a group of 10 to 100 s-bots for the different densities (200 trials per situation). S-bots that have not established a connection within the predefined timeout of $300\,\mathrm{s}$ are not taken into account. We do not consider the densities 0.050 and 0.075, as the percentage of connected s-bots is particularly low.

Fig. 24 (rule-based controller) and Fig. 25 (neural network based controller) present the average time (over all trials) it took an s-bot to connect, divided by the group size and scaled so that the performance for group size 10 equals 1. For the neural network based controller, the time grows sub-linearly with the group size. This might be due to the fact that the bigger the structure, the more it provides surface for potential connections.

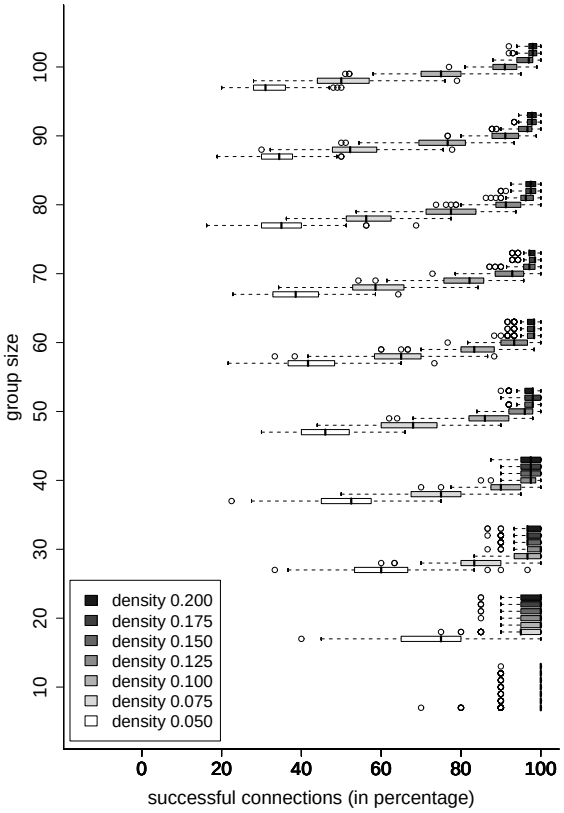


Fig. 23. Box-and-whisker plot [42] showing the percentage of successful connections during self-assembly in a group of 10 to 100 s-bots, for different initial densities (200 observations per box). The s-bots are controlled by the neural network based controller.

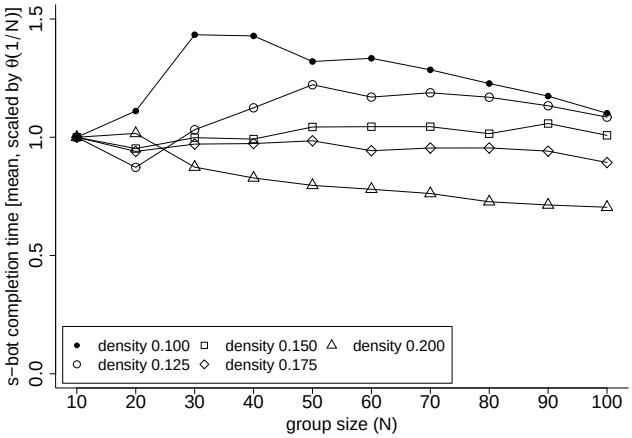


Fig. 24. Time complexity (see text for details) for groups of 10 to 100 s-bots and different initial densities. The s-bots are controlled by the rule-based controller.

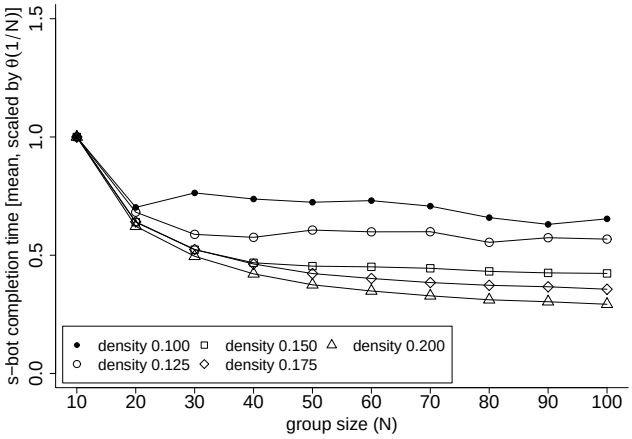


Fig. 25. Time complexity (see text for details) for groups of 10 to 100 s-bots and different initial densities. The s-bots are controlled by the neural network based controller.

VIII. Discussion

A. Evaluation of Results

We have demonstrated the ability of the modules of the swarm-bot platform to self-assemble under a variety of conditions. The reliability and the performance in each experiment can be judged by quantitative results. Additionally, all experiments were recorded on video (see `http://iridia.ulb.ac.be/~rgross`).

In Section VI, we examined self-assembly of up to six modules on two different types of rough terrain. Both terrain types are unnavigable for most standard wheeled robots of a similar size. The first terrain type has a surface with a regular structure. Experiments on this terrain type showed no loss in reliability. The second terrain type has an irregular surface. More than 90% of the s-bots used in the experiments on this terrain type still successfully established a connection.

In Section VII, we have shown that the system is scalable, that is, our controller is capable of letting large swarms of s-bots self-assemble. Quantitative results are presented with groups of up to 16 physical s-bots and up to 100 s-bots in simulation. It is shown that, when using the evolved neural network controller, the time it takes for an s-bot to connect grows sub-linearly with the group size.

B. Decisive Design Choices

We believe that our success can be attributed to the following critical choices made during the system design:

1) **Mobility**: the traction system was designed so that the s-bot is equipped with very good steering abilities (due to the external wheels). At the same time it allows for good all-terrain navigation (due to the tracks). This facilitates approaching a teammate to establish a connection on flat and rough terrain.
2) **Connection Mechanism**: the s-bot can receive connections on more than two thirds of its perimeter. Moreover, the connection mechanism is designed so that it does not require a specific and accurate alignment of the two s-bots during approach. This property, together with the mobility of the s-bot, are crucial factors for the design of robotic systems capable of self-assembling on rough terrain.
3) **Complex Individuals Expressing Simple Collective Rules**: the s-bot is equipped with a variety of complex sensors that guide it during a) the approach of red objects, b) the avoidance of blue objects, and c) the connection phase. To some extent, the sensory system indicates also the presence of failures (e.g., in the connection). To preprocess data provided by the sensors, the s-bot is equipped with a considerable amount of computational resources.

Our s-bots might currently be considered complex artefacts. We believe that the use of relatively complex modules and robots is unavoidable in order to achieve tasks of increasing complexity in the domain of self-reconfigurable and collective robotics.
Despite the complexity of modules, their behavior and the interactions among them can often be modeled by simple rules. In this study, the main part of the control is given by a simple, reactive neural network with 15 connection weights.

4) **Scalability**: as detailed in Section IV, the control is decentralized (s-bots are fully autonomous) and homogenous (group members have identical control). The s-bots make use only of local sensing and acting abilities (no global communication channels). Due to these properties, the controller can, in principle, be applied to robotic swarms of any (finite) size.
However, these properties by themselves do not ensure that the performance will scale well with group size. To improve scalability for our particular task, we introduced a simple binary communication mechanism which allowed s-bots to signal whether or not they were connected. This simple mechanism governs the process of attraction and repulsion, and allows for the progressive construction of (global) connection patterns of dimensions far beyond the s-bot's (local) sensing range.
The authors admit that the practical use of the system is limited by the physical constraints of the formed structures. In a test modeling a real world rescue scenario with 19 s-bots of approximately $700\,\mathrm{g}$ each, pulling a nine-year old child of $20\,\mathrm{kg}$ towards a light source, it happened that the connection mechanism of an s-bot broke.[7]

C. Ongoing Work and Open Issues

Future research will address the design of self-assembling robotic systems that operate in different types of environments, such as on the surface of (or under) water or other fluids, or in space [44].

One of the next issues we want to address with the *swarm-bot* is to identify the potential and the limitations of the structures formed. At the time of writing, we have already succeeded in demonstrating the ability of a group of physical s-bots to achieve the following tasks using self-assembly (based on the control described in Section IV):

- cooperative transport of very heavy, but small objects that do not allow for direct manipulation by a sufficient number of s-bots [45],
- crossing a hole of a size too big for a single s-bot to pass,
- navigation over a hill impossible for a single s-bot to overcome [46].

In all of these three problems we observed that the performance of the system may depend on the spatial arrangement of the connected s-bots. During cooperative transport, s-bots in a chain formation exhibited higher pulling forces than when organized in other formations. For overcoming a hill and for crossing a hole, it was beneficial, or even necessary, for the structure of the swarm-bot to be longer than the obstacle it encountered. Although it is possible to increase size and strength of a swarm-bot by making use of more s-bots, it would be of special interest to study mechanisms that let a swarm-bot self-reconfigure its shape in response to the demands of the environment.

Another promising direction in research is the study of *functional self-assembly* [47], that is, the ability to self-assemble and disband as an adaptive response to the environment. A first instance of functional self-assembly in a team of three physical s-bots has already been successfully demonstrated. The task requires the s-bots to navigate over unknown terrain towards a target light source. If possible, the s-bots should navigate to the target independently. If, however, the terrain proves too difficult for a single s-bot, the group must self-assemble into a larger entity and collectively navigate to the target [46].

D. Conclusions

In this article, we have presented a comprehensive study of the problem of autonomous self-assembly. At present, swarm-bot is the state of the art in autonomous self-assembly for what concerns group size, reliability, and speed. Our approach proved robust with respect to different initial conditions and different types of terrain. In addition, the system scales well with group size, as validated with 16 physical modules and up to 100 modules in simulation.

We have identified decisive factors in the system design that might have implications for the design of collective and self-reconfigurable robot systems in the future.

Ongoing work indicates that we can address more complex robotic tasks, at the cutting edge of the current research in autonomous, mobile robotics.

Acting in between the two research fields of collective and self-reconfigurable robotics, we believe that the study of autonomous self-assembly is a very promising avenue for future research.

APPENDIX

We used the following policy during experimentation.

- In case an s-bot gets in a situation in which its hardware can potentially be damaged (e.g., if it topples down), we remove it manually during the experimentation and we count this as a failure to achieve self-assembly. In case of an experiment with multiple s-bots, the trial is not stopped. In total, this situation has occurred 5 times, once on flat terrain and 4 times on rough terrain.
- We carefully check whether the connection the s-bot has established is tight. In total, it happened twice that a (solitary) s-bot ended up in the connection state without being connected (directly or indirectly) with the seed. This, we counted as failure to connect.

[7] A video recording is available at `http://iridia.ulb.ac.be/~rgross`.

REFERENCES

[1] M. Yim, Y. Zhang, and D. Duff, "Modular robots," *IEEE Spectr.*, vol. 39, no. 2, pp. 30–34, 2002.

[2] D. Rus, Z. Butler, K. Kotay, and M. Vona, "Self-reconfiguring robots," *Commun. ACM*, vol. 45, no. 3, pp. 39–45, 2002.

[3] G. M. Whitesides and B. Grzybowski, "Self-assembly at all scales," *Science*, vol. 295, no. 5564, pp. 2418–2421, 2002.

[4] F. Mondada, L. M. Gambardella, D. Floreano, S. Nolfi, J.-L. Deneubourg, and M. Dorigo, "SWARM-BOTS: Physical interactions in collective robotics," *IEEE Robot. Autom. Mag.*, vol. 12, no. 2, pp. 21–28, 2005.

[5] M. Yim, K. Roufas, D. Duff, Y. Zhang, C. Eldershaw, and S. B. Homans, "Modular reconfigurable robots in space applications," *Auton. Robots*, vol. 14, no. 2-3, pp. 225–237, 2003.

[6] A. Castano, W.-M. Shen, and P. M. Will, "CONRO: Towards deployable robots with inter-robots metamorphic capabilities," *Auton. Robots*, vol. 8, no. 3, pp. 309–324, 2000.

[7] D. Rus and M. Vona, "Crystalline robots: Self-reconfiguration with compressible unit modules," *Autonomous Robots*, vol. 10, no. 1, pp. 107–124, 2001.

[8] S. Murata, E. Yoshida, A. Kamimura, H. Kurokawa, K. Tomita, and S. Kokaji, "M-TRAN: Self-reconfigurable modular robotic system," *IEEE/ASME Trans. Mechatron.*, vol. 7, no. 4, pp. 431–441, 2002.

[9] 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*, vol. 2. IEEE Computer Society Press, Los Alamitos, CA, 2004, pp. 2068–2073.

[10] F. Mondada, M. Bonani, S. Magnenat, A. Guignard, and D. Floreano, "Physical connections and cooperation in swarm robotics," in *Proc. of the 8^{th} Conf. on Intelligent Autonomous Systems*. IOS Press, Amsterdam, The Netherlands, 2004, pp. 53–60.

[11] M. Yim, D. G. Duff, and K. D. Roufas, "PolyBot: a modular reconfigurable robot," in *Proc. of the 2000 IEEE Int. Conf. on Robotics and Automation*, vol. 1. IEEE Computer Society Press, Los Alamitos, CA, 2000, pp. 514–520.

[12] M. Yim, Y. Zhang, K. Roufas, D. Duff, and C. Eldershaw, "Connecting and disconnecting for chain self-reconfiguration with PolyBot," *IEEE/ASME Trans. Mechatron.*, vol. 7, no. 4, pp. 442–451, 2002.

[13] A. Castano, A. Behar, and P. M. Will, "The CONRO modules for reconfigurable robots," *IEEE/ASME Trans. Mechatron.*, vol. 7, no. 4, pp. 403–409, 2002.

[14] 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*, vol. 3. IEEE Computer Society Press, Los Alamitos, CA, 2004, pp. 2877–2882.

[15] E. Mytilinaios, M. Desnoyer, D. Marcus, and H. Lipson, "Designed and evolved blueprints for physical self-replicating machines," in *Proc. of the 9^{th} Int. Conf. on the Simulation and Synthesis of Living Systems (Artificial Life IX)*. MIT, Cambridge, MA, 2004, pp. 15–20.

[16] V. Zykov, E. Mytilinaios, B. Adams, and H. Lipson, "Self-reproducing machines," *Nature*, vol. 435, no. 7039, p. 163, 2005.

[17] T. Fukuda and S. Nakagawa, "A dynamically reconfigurable robotic system (concept of a system and optimal configurations)," in *Proc. of the 1987 IEEE Int. Conf. on Industrial Electronics, Control and Instrumentation*. IEEE Computer Society Press, Los Alamitos, CA, 1987, pp. 588–595.

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

[19] T. Fukuda, Y. Kawauchi, M. Buss, and H. Asama, "A study on dynamically reconfigurable robotic systems (recognition and communication system of cell-structured robot CEBOT)," *JSME Int. J. III-VIB. C.*, vol. 34, no. 2, pp. 295–302, 1991.

[20] 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*. IEEE Computer Society Press, Los Alamitos, CA, 1988, pp. 145–150.

[21] T. Fukuda and S. Nakagawa, "Method of autonomous approach, docking and detaching between cells for dynamically reconfigurable robotic system CEBOT," *JSME Int. J. III-VIB. C.*, vol. 33, no. 2, pp. 263–268, 1990.

[22] 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*, vol. 2. IOS Press, Amsterdam, The Netherlands, 1989, pp. 661–671.

[23] S. Hirose, T. Shirasu, and E. F. Fukushima, "Proposal for cooperative robot "Gunryu" composed of autonomous segments," *Robot. Auton. Syst.*, vol. 17, pp. 107–118, 1996.

[24] 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*, ser. Lecture Notes in Control and Information Sciences, vol. 271. Springer, Berlin, Germany, 2000, pp. 333–342.

[25] ——, "Towards a team of robots with reconfiguration and repair capabilities," in *Proc. of the 2001 IEEE Int. Conf. on Robotics and Automation*, vol. 3. IEEE Computer Society Press, Los Alamitos, CA, 2001, pp. 2923–2928.

[26] R. Damoto, A. Kawakami, and S. Hirose, "Study of super-mechano colony: concept and basic experimental set-up," *Adv. Robots*, vol. 15, no. 4, pp. 391–408, 2001.

[27] S. Hirose, "Super mechano-system: New perspective for versatile robotic system," in *Proc. of the 7^{th} Int. Symp. on Experimental Robotics*, ser. Lecture Notes in Control and Information Sciences, vol. 271. Springer, Berlin, Germany, 2001, pp. 249–258.

[28] M. Yamakita, Y. Taniguchi, and Y. Shukuya, "Analysis of formation control of cooperative transportation of mother ship by SMC," in *Proc. of the 2003 IEEE Int. Conf. on Robotics and Automation*, vol. 1. IEEE Computer Society Press, Los Alamitos, CA, 2003, pp. 951–956.

[29] K. Motomura, A. Kawakami, and S. Hirose, "Development of arm equipped single wheel rover: Effective arm-posture-based steering method," *Auton. Robots*, vol. 18, no. 2, pp. 215–229, 2005.

[30] H. B. Brown, M. Weghe, C. Bererton, and P. K. Khosla, "Millibot trains for enhanced mobility," *IEEE/ASME Trans. Mechatron.*, vol. 7, no. 4, pp. 452–461, 2002.

[31] 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 (ICRA 2004)*, vol. 3. IEEE Computer Society Press, Los Alamitos, CA, 2004, pp. 2888–2893.

[32] P. J. White, V. Zykov, J. Bongard, and H. Lipson, "Three dimensional stochastic reconfiguration of modular robots," in *Proc. of Robotics Science and Systems*. MIT, Cambridge, MA, 2005.

[33] 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*. IEEE Computer Society Press, Los Alamitos, CA, 2005, pp. 2644–2651.

[34] S. T. Griffith, "Growing machines," Ph.D. dissertation, MIT, MA, USA, September 2004.

[35] S. Griffith, D. Goldwater, and J. M. Jacobson, "Self-replication from random parts," *Nature*, vol. 437, no. 7059, p. 636, 2005.

[36] 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*, vol. 17, no. 2–3, pp. 223–245, 2004.

[37] F. Mondada, G. C. Pettinaro, A. Guignard, I. W. Kwee, D. Floreano, J.-L. Deneubourg, S. Nolfi, L. M. Gambardella, and M. Dorigo, "Swarm-Bot: A new distributed robotic concept," *Auton. Robots*, vol. 17, no. 2–3, pp. 193–221, 2004.

[38] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychol. Rev.*, vol. 65, no. 6, pp. 386–408, 1958.

[39] 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*, ser. Lecture Notes in Computer Science, vol. 3242. Springer Verlag, Berlin, Germany, 2004, pp. 852–861.

[40] H.-P. Schwefel, "Evolutionsstrategie und numerische Optimierung," Ph.D. dissertation, Technische Universität Berlin, Fachbereich Verfahrenstechnik, Germany, 1975.

[41] H.-G. Beyer, *The Theory of Evolution Strategies*. Springer, Berlin, Germany, 2001.

[42] 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.

[43] H. Melissen, "Densest packings of eleven congruent circles in a circle," *Geom. Dedicata*, vol. 50, no. 1, pp. 15–25, 1994.

[44] W.-M. Shen, P. Will, and B. Khoshnevis, "Self-assembly in space via self-reconfigurable robots," in *Proc. of the 2003 IEEE Int. Conf. on Robotics and Automation*, vol. 2. IEEE Computer Society Press, Los Alamitos, CA, 2003, pp. 2516–2521.

[45] E. Tuci, R. Groß, V. Trianni, M. Bonani, F. Mondada, and M. Dorigo, "Cooperation through self-assembling in multi-robot systems," IRIDIA

- Université Libre de Bruxelles, Tech. Rep. IRIDIA-TR-2005-3, 2005, to appear in *ACM Transactions on Autonomous and Adaptive Systems*.

[46] R. O'Grady, R. Groß, M. Bonani, F. Mondada, and M. Dorigo, "Self-assembly on demand in a group of physical autonomous mobile robots navigating rough terrain," in *Advances in Artificial Life - Proc. of the 8th European Conf. on Artificial Life (ECAL 2005)*, ser. Lecture Notes in Artificial Intelligence, vol. 3630. Springer Verlag, Heidelberg, Germany, 2005, pp. 272–281.

[47] V. Trianni, E. Tuci, and M. Dorigo, "Evolving functional self-assembling in a swarm of autonomous robots," in *Proc. of the* 8^{th} *Int. Conf. on Simulation of Adaptive Behavior*. MIT Press, Cambridge, MA, 2004, pp. 405–414.