Autonomous Self-Assembly in Swarm-Bots

Roderich Groß, *Student Member, IEEE*, Michael Bonani, Francesco Mondada, *Associate Member, IEEE*, and
Marco Dorigo, *Fellow, IEEE*

Abstract—In this paper, 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 is comprised of 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—Autonomous, collective, distributed, modular, reconfigurable, self-assembly, swarm-bot, swarm robotics.

I. INTRODUCTION

MODULAR robotics is a research field that is still progressing very quickly [1], [2]. Modular robotic systems tend to be inherently robust and flexible, properties that are likely to become increasingly important in real-world robotics applications. Recently, special attention has been paid to *self-reconfigurable* robots, that is, modular robots whose components can autonomously organize into different connected configurations. In the majority of current implementations, modular robots, even those that can self-reconfigure, are initially manually assembled and, once assembled, they are incapable of assimilating additional modules without external direction. In contrast, in this paper, we are interested in modular systems whose components are capable of self-assembling to set up modular robots of arbitrary size autonomously.

Following Whitesides and Grzybowski [3], *self-assembly* can be defined as *a reversible process by which pre-existing discrete entities bind to each other without being directed externally*, and may involve components at scales from the molecular (e.g., DNA strands forming a double helix) to the planetary (e.g., weather systems).

In robotics, self-assembly is of particular interest because it can provide modular robots with additional capabilities and functions. An important example is that of a modular robot that could change the number and/or type of its modules in order to solve a problem that originally it could not solve. We talk in this case of *task-oriented self-assembly*. Other interesting examples are those of modular robots that through self-assembly could achieve *self-replication* by using building blocks provided by the environment, or *self-repair* by replacing defective components with new modules available in the environment.

Additionally, modular robots could also use self-assembly as a way to reproduce capabilities observed in non-self-assembling modular systems. For instance, a modular robot could, by self-assembling, display task-oriented reconfiguration, that is, change its morphology so that it can solve a problem it could not solve in its original configuration.

We believe that the capabilities mentioned above will become more and more important as increasingly complex missions place greater demands on robotic systems. This applies in particular to missions that require a high degree of autonomy, flexibility, and robustness, such as, for instance, planetary pre-colonization, search and rescue, deep sea exploration, and underground exploitation of energy resources. Requirements for such autonomous missions can hardly be satisfied by a pre-designed robot because of its inherent limits in size, strength, versatility, and robustness.

In this paper, we present a comprehensive study on autonomous self-assembly with a new collective and mobile reconfigurable robotic system called *swarm-bot* ([4], see also http://www.swarm-bots.org). The modules are fully autonomous mobile robots that, by establishing physical connections with each other, can organize into modular robots. We do not consider any particular application of such a system, as our focus in this paper is on the hardware and control mechanisms that allow the system to realize self-assembly.

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 system is applicable to larger group sizes. In Section VIII, we discuss the results. Finally, Sections IX and X summarize our ongoing work and conclude the paper.

II. RELATED WORK

In this section, we provide a brief survey of related work. The survey does not span the entire field of self-reconfigurable

robotics. Instead, it focuses on systems for which self-assembly has been demonstrated. Following on from Yim *et al.* [1], we identify four different reconfiguration mechanisms used by these systems: chain-based, lattice-based, mobile, and stochastic. Our survey is organized accordingly (see Sections II-A–D, respectively). In the systems described in the first three parts, modules or groups of them move autonomously. In stochastic self-reconfigurable robots, by contrast, motion is induced externally.

A. Chain-Based Self-Reconfigurable Robots

1) PolyBot: PolyBot [1], [5] is a modular chain robot that can configure its shape with no external mechanical assistance. Each module has one degree of freedom (DOF) involving rotation of two opposite connection plates through a $\pm 90°$ range. A shape memory alloy actuator integrated in each connection plate can rotate a latch to catch lateral grooves in the pins from the mating connection plate. Additional passive cuboid segments with six connection plates can be used to introduce branches to the structure and to connect with an (external) power supply. Active modules are equipped with infrared (IR) detectors and emitters integrated in the connection plates.

Yim *et al.* [6] demonstrated the ability of a modular robot arm composed of six PolyBot modules of generation G2 to approach and 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 a 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 detectors and emitters, and by the mechanical properties of the connection mechanism (pins sliding into chamfered holes). A similar experiment was accomplished using PolyBot generation G3 [5], [6]. A modular arm composed of seven modules approached and docked with another module [7]. The modular arm could operate in 3-D. In the experiment, the arm and the target module were set up approximately in a same vertical plane.

2) CONRO: CONRO is a homogeneous, modular chain robot [8], [9]. 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 in the pitch and yaw axes by means of two motorized joints. A shape memory alloy actuator integrated in the active connector can rotate a latch to catch lateral grooves in the pins from the plate of the mating passive connector. IR emitters and detectors are integrated in the connection plates to support the docking and to enable communication between connected modules.

Recently, Rubenstein *et al.* [10] 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 both 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. During the experiment, the two modular robots were tethered to an external power supply.

B. Lattice-Based Self-Reconfigurable Robots

1) Molecubes: Molecubes [11] is a homogeneous, lattice-based self-reconfigurable robot. Each module is a 10-cm cube, and one half of it can swivel relative to the other half. Each half can bind with one additional module by using electromagnets. Molecubes are powered through a baseplate and transfer data and power through their faces.

Zykov *et al.* [12] demonstrated the self-replication of a four-module entity provided with an ordered supply of additional units. The system executed a predetermined sequence of actions. To confirm a successful connection among modules, communication was employed.

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 [13], [14]. CEBOT is a heterogeneous system comprised of modules with different functions (e.g., to move, bend, rotate, and slide). A series of prototypes has been implemented. The first prototype, the CEBOT Mark I [15], is of cuboid shape with active and passive connectors on opposite sides. Similar to the connection mechanism in PolyBot or CONRO, a shape memory alloy actuator can cause a latch to catch a lateral groove in a pin from the mating module. It was shown that a module (equipped with two motorized wheels) could approach the back of another module [15]. However, such a "rough approach" was found ineffective for coupling the two modules, as the connection mechanism required a very precise alignment. In CEBOT Mark II [16], [17] and CEBOT Mark IV [18], [19], a mechanical hook is used instead for connecting. Additionally, a cone-shaped part fixed on the front of each module matches a counterpart on the back of each module to facilitate alignment during approach. In CEBOT Mark III [20], modules have a hexagonal shape. Three faces each are provided with active and passive connectors, respectively. The connection mechanism is similar to the one employed in CEBOT Mark I; however, the pins of the active connectors are made of elastic material. The module is equipped with six nozzles providing propulsion on flat terrain.

Fukuda *et al.* demonstrated the successful docking of a mobile module with a stationary module, using the CEBOT Mark II [16], Mark III [20], and Mark IV [19] platforms. In each case, coordination was achieved by making use of a set of IR detectors and emitters. Communication among the (connected) modules of a modular robot was studied to enable it to approach and connect with an additional module [17].

2) Gunryu: Hirose *et al.* proposed a distributed robotic concept called Gunryu (GR) [21]. 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 [21]. In GR-I,

no active connection mechanism was implemented. The robots were mechanically linked by means of a passive 6-DOF arm.

3) Repairable Robot Teams: Bererton and Khosla studied cooperative repair in a team of two autonomous, wheeled robots [22], [23]. Although the robots cannot establish a firm connection with each other, the difficulties encountered in this study are similar to those that we face in self-assembly experimentation. One robot (the *repair robot*) is equipped with a forklift mechanism that can be partially inserted into a receptacle of a defective component of its (stationary) teammate. A black-and-white camera is mounted on top of the approaching robot. It is connected to an external PC that processes the images and sends control commands to the approaching robot via an RF link.

A simple state machine proved capable of controlling the repair robot to replace a component of its teammate [22]. The robot could perform the docking for distances up to 30 cm, and for angular displacements up to $30°$.

4) Super Mechano Colony: Super Mechano Colony (SMC) [24], [25] is a modular robotic concept composed of a *parent* unit and several *child* units attached to it. Child units are an integral part of the system's locomotion. In addition, the child units can disband to accomplish separate, autonomous missions, and reconnect once the missions are accomplished. Hirose *et al.* [24], [25] introduced an early prototype of the SMC concept. Two motorized and two passive wheels provide mobility on flat terrain. Each unit is equipped with a manipulation arm that can be elevated, and a gripper attached to it. The upper body (including the gripper) can be rotated with respect to the chassis by means of a motorized vertical axis. For a similar prototype [26], a modular robot composed of a parent unit and three child units proved capable of task-oriented reconfiguration. The parent unit was supposed to move in a straight line. The tracking performance depended on both the speed and the connection structure. Initially, the three child units were manually arranged into a chain pulling the parent unit. The two child units at the back of the chain disconnected, followed a predefined path, and reconnected to the parent unit directly. This allowed for an optimal tracking performance at a different speed.[1]

5) SMC Rover: A more recent implementation of SMC is the SMC rover [25], [27]. It is a planetary rover composed of a parent unit with attachable child units (called *Uni-Rovers*). Uni-Rovers are composed of a single wheel and a manipulation arm. The latter is equipped with a gripper that can grasp another Uni-Rover unit, as well as the parent unit. The current prototype is not equipped with any sensors. Its primary purpose is to study and improve the basic capabilities of the underlying mechanics [27].

6) Millibot Trains: Similar to GR, the Millibot Train [28] is composed of multiple modules that can be linearly linked. Each module is equipped with caterpillar tracks. Similar to CONRO, PolyBot, and CEBOT Mark I, the connection mechanism of a Millibot Train module is based on protruding pins that register with a matching receptacle. In contrast to the other systems, however, the latching mechanism is integrated in one of the pins. The prototype is not equipped with any sensors; it is manually controlled [28].

D. Stochastic Self-Reconfigurable Robots

1) Self-Assembly on an Air Table: Recently, there has been growing interest in the design and study of a new type of reconfigurable system made of simple programmable modules provided with no full DOF. These modules move passively and bind to each other upon random collision. White *et al.* studied a system in which the modules float passively on an air table that was fixed to an orbital shaker [29]. The modules are cuboid and equipped with switchable electromagnets for binding to each other.[2] The modules have no locomotion abilities and are unpowered. Once they bind with a seed module that is connected to a power supply, they become active. Self-assembly and self-reconfiguration was demonstrated with three modules. Moreover, systematic experiments with two to three modules were carried out to determine the time required to form a modular entity comprising two modules by self-assembly.

Klavins' group has demonstrated self-assembly of modules that slide passively on an air table [30]. The modules are triangular, having a side length of 12 cm. Each side is equipped with a connection mechanism comprised of one fixed and two movable permanent magnets. Power is provided onboard. Once connected, modules execute a common graph grammar in a distributed fashion. N modules can assemble up to $\lfloor N/6 \rfloor$ hexagons autonomously, although the yield may not be optimal, given the stochastic nature of the system.

2) Self-Replication on an Air Table: Griffith *et al.* developed a system of simple *electromechanical assemblers* [31], [32]. The basic module has two active and two passive connection sides. Each active side is equipped with a physical latch that is activated by an electromagnet once a mating module is sufficiently close. The modules slide passively on an air table. The system demonstrated the self-replication of a five-module entity provided with an unordered supply of additional units, as well as the construction of entities comprising up to 26 connected modules by self-assembly [31].

3) Self-Assembly in Agitated Fluids: White *et al.* developed two modular robot systems and an apparatus containing an agitated fluid in which modules are subject to random motion [33]. In the first modular robot system, switchable electromagnets are employed for letting modules bind to each other. The ability of two modules to self-assemble was assessed in 50 trials. One module was manually attached to a magnetic plate, and thereby connected to an external power supply. The other module moved passively in the apparatus. In 24% of the trials, the modules could, by self-assembling, form and change their configuration. Communication among connected modules was employed to synchronize the actions required for disconnecting. In addition, passive aggregation, an irreversible process (and thus not self-assembly), was demonstrated with up to four free-moving, unpowered modules.

In the second modular robot system, the fluid of the apparatus flows through pipelines that are integrated in the modules. Each module is a cube with one opening on each face. Six pipelines, one for each face, join in the module's center. Each pipeline

---

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

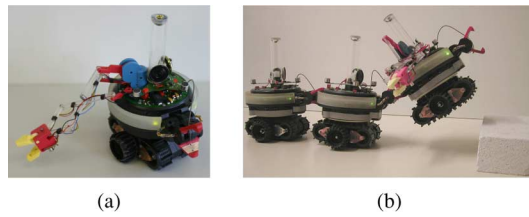[2]In a second implementation, modules are of triangular shape and equipped with permanent magnets instead.

Fig. 1. Swarm-bot concept. (a) The s-bot, a fully autonomous mobile robot. (b) Three connected s-bots forming a swarm-bot able to change its shape, in this case, to climb a step too difficult for a single s-bot.
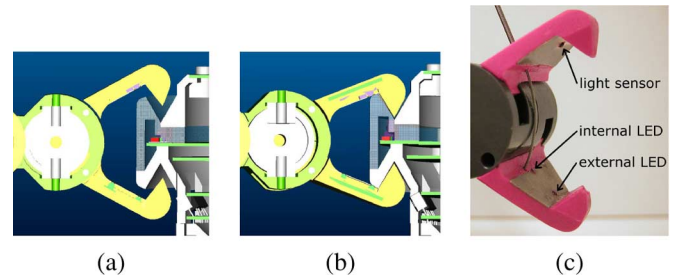

Fig. 2. Rigid gripper. (a) Loose and (b) tight connections of an s-bot with the connection ring of a teammate. (c) Optical barrier(s) to detect objects to grasp.

is equipped with a valve that can be opened or closed to control the flow. White *et al.* [33] demonstrated the ability of two modules to self-assemble in order to form and change configurations. One module was fixed to the apparatus and a pump was connected to the opening of one face. The force of the fluid was directed towards the module and let another module approach and bind with the previous one. There was no binding force other than the pressure caused by the flow.

III. HARDWARE DESIGN

The swarm-bot platform [34], [35] is a new distributed robotic concept lying in between collective and self-reconfigurable robotics. The concept is illustrated in Fig. 1. The basic components of the system, called *s-bots*, are fully autonomous mobile robots. On the other hand, multiple s-bots, by connecting to each other, can organize into a modular robot, called *swarm-bot*, that can self-reconfigure its 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 700 g.

The s-bot has nine DOFs, all of which are rotational:

- two DOFs 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 as 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);
- three DOFs for controlling the position of the flexible arm (not exploited in this experiment).

Most of these DOFs are actuated by DC motors equipped with an incremental encoder and controlled in torque, position, or speed by a proportional-integral-derivative (PID) controller. Only two DOFs (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:

- four proximity sensors fixed underneath (ground sensors);
- 15 proximity sensors distributed around the turret;
- four optical barriers integrated in the two grippers;
- one force sensor between the turret and the chassis (2-D traction sensor);
- one torque sensor on the elevation arm of the rigid gripper;
- two humidity and temperature sensors;
- three-axis inclinometer;
- eight light sensors distributed around the module;
- four omnidirectional microphones;
- one video graphics array (VGA) omnidirectional camera.

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

When connected in a swarm-bot, the chassis of each s-bot can be rotated in any (horizontal) direction, which allows for coordinated motion. The s-bot's actuators and (internal as well as external) sensors allow the swarm-bot 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], [35], and [36].

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 DOFs 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 omnidirectional 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 PIC® processors communicating with the main XScale 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 2 h of autonomy.

IV. CONTROL DESIGN

We aim to develop a controller that allows a group of s-bots to self-assemble. The process is seeded by a dedicated object. The s-bots must locate, approach, and connect to the seed, either directly or by becoming part of a modular robot that is connected to the seed. The seed can be either a specifically designed object or an s-bot.

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.

   - **Decentralized Control**: The s-bots are controlled in a fully autonomous and distributed manner.
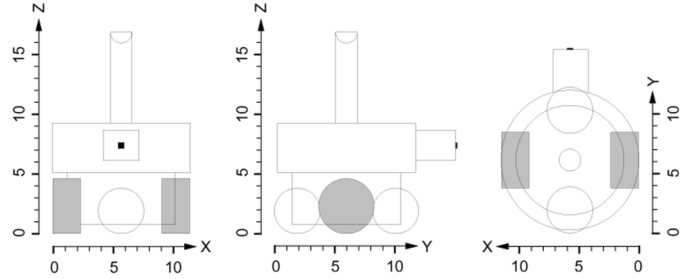

Fig. 3. Simulation model of the s-bot: front, side, and top views (sizes in centimeters).

   - **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.

   Due to these properties, the controller is applicable to robotic swarms of any (finite) size. In Section VII, we examine the relationship between the performance and the number of modules. 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 3-D. 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 is 2 rad/s.
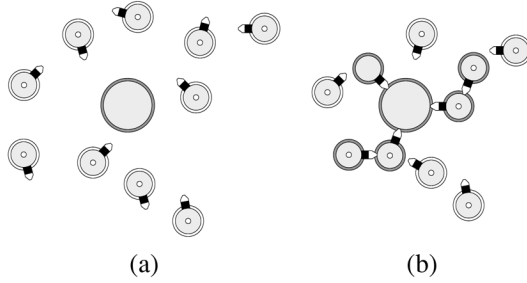
Fig. 4. Group of s-bots self-assembling and connecting to a prey, which in this case acts as a seed for the process of self-assembling.

The connection mechanism is represented by the cuboid heading forward with a small contact plate in the front. Once the plate is in contact with a grippable object, the gripper can establish a rigid connection. 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 blue or red light. 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) or not gripped (blue).

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 $D_{\max} = 60$ 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 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:     $(i_1, i_2) \leftarrow$ feature extraction (camera)

4:     $(i_3, i_4) \leftarrow$ sensor readings (proximity)

5:     $(o_1, o_2, o_3) \leftarrow f(i_1, i_2, i_3, i_4)$

6:

7:     **if** $(o_3 > 0.5) \wedge$ (grasping requirements fulfilled) **then**

8:         close gripper

9:         **if** successfully connected **then**

10:             activate color ring in red

11:             halt until timeout reached

12:         **else**

13:             open gripper

14:         **endif**

15:     **endif**

16:     apply (o_1, o_2) 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 right sides of the traction system (line 16) and the connection mechanism (lines 7–15). 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 course orientation.

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 right sides 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 $D_{\text{coll}} = 20$ 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 right

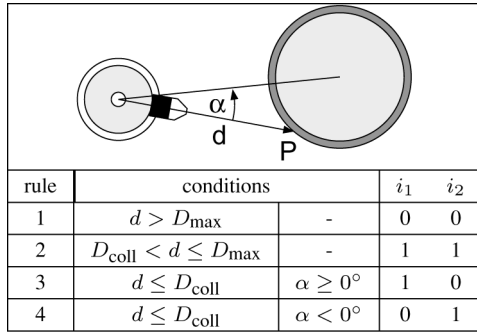| rule	conditions		i_1	i_2
1	$d > D_{\max}$	-	0	0
2	$D_{\text{coll}} < d \leq D_{\max}$	-	1	1
3	$d \leq D_{\text{coll}}$	$\alpha \geq 0°$	1	0
4	$d \leq D_{\text{coll}}$	$\alpha < 0°$	0	1

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. $D_{\text{coll}} = 20$ cm is the distance between the s-bot and another object under which there is high risk of collision. $D_{\max} = 60$ cm defines the sensing range.

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

(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).

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 (NN). As illustrated in Fig. 6, the NN 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 NN 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 NN are listed in Table II. They were shaped by artificial evolution in the context of a cooperative transport task [38]. The evolutionary algorithm used was a self-adaptive $(\mu + \lambda)$ evolution strategy [39], [40].
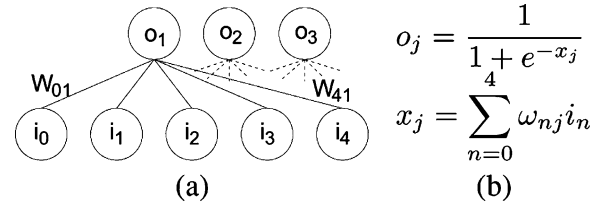

Fig. 6. (a) Graphical representation of the feedforward two-layer artificial NN (i.e., a perceptron [37]) of the assembly module. i_1, i_2, i_3, and i_4 are the nodes which take input from the s-bot's sensors. i_0 is the bias term. o_1, o_2, and o_3 are the output nodes. (b) Equations used to compute the network output values.

TABLE II
WEIGHTS OF THE NN 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

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.

1) Recovery Move: 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 $K = 6$ 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 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.

2) Vision: 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. Figs. 7 and 8 detail the rules to determine the values of the arguments i_1 and i_2 of the mapping function f. Thereby, 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., $D_{\max} = \infty$). The software we use to detect colored objects makes it possible to recognize red (blue) objects up to a distance of 70–90 cm (35–50 cm), depending on which s-bot is used.

3) Connection Mechanism: The connection mechanism is controlled in lines 7–15 of **Algorithm 1**. The gripper is closed if the following set of requirements is fulfilled (see also Fig. 7):
- $o_3 > 0.5$;
- $((i_1 \neq 0) \vee (i_2 \neq 0)) \wedge (d_1 \leq D_{\text{grasp}}) \wedge (|\alpha| \leq 30°)$;[3]
- no connection attempt failed within the last 18 control steps (i.e., approximately 3 s).

[3] D_{grasp} is an estimate of the maximum distance to an object that can still be grasped.
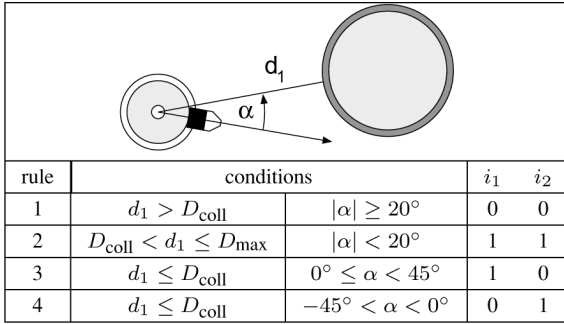
Fig. 7. On the physical s-bot, the perceptual range for detecting red objects to approach is 45° to the left and right sides of the s-bot's front. If no red block resides in this range, or if an obstacle (a blue block; for details, see next figure) is present, i_1 and i_2 are set to zero. Otherwise, (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.
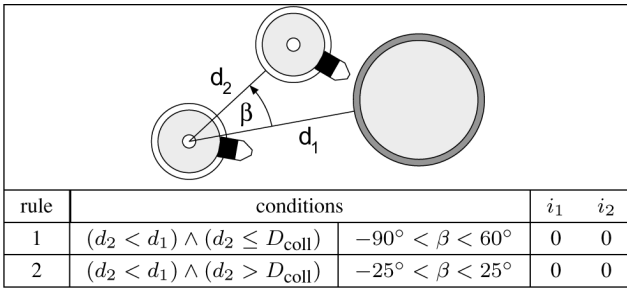

Fig. 8. Rule set defining whether an obstacle is present. If, in addition to the red block at distance d_1, there exists a blue block at distance d_2 and with angular displacement β, and if rules 1 or 2 are satisfied, then an obstacle is present. In this case, i_1 and i_2 are set to zero. The range of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

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.

4) Traction System: 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 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 cm/s of the s-bot. 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.

(a)   (b)

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

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 [38] 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).

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 cm and is positioned 0.5 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 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. 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 cm (50 cm) is 22.6 s (34.9 s).

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).[4]

S-bot 6 performed significantly worse than the other s-bots given a starting distance of 50 cm (see Fig. 10). We observed that the camera images of s-bot 6 were of bad quality when compared with the other s-bots. Therefore, s-bot 6 sporadically could not detect the prey at a distance of 50 cm. Nevertheless, s-bot 6 succeeded in all 20 trials to connect starting from this
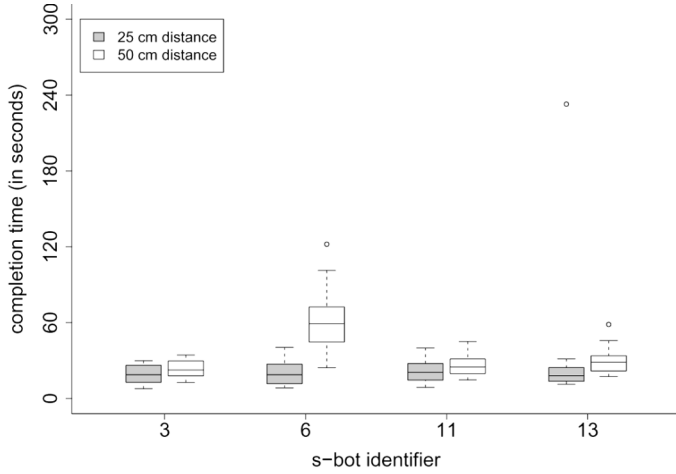
---

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

Fig. 10. Self-assembly of a single s-bot with a prey. Box-and-whisker plot [41] of the completion times (20 observations per box) grouped according to the s-bot involved and its initial distance from the prey.
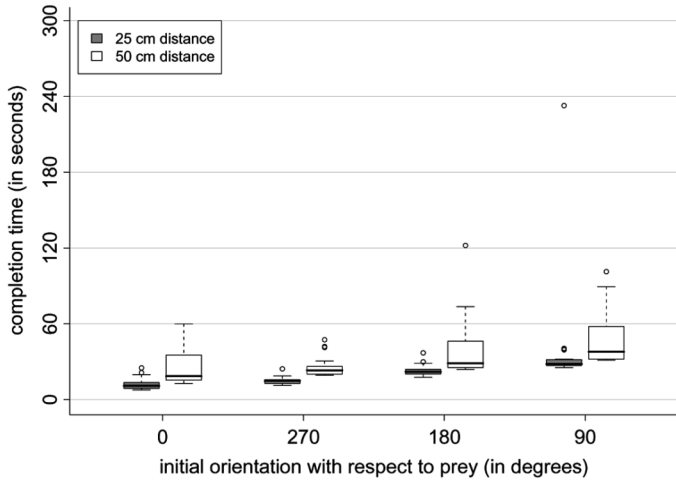

Fig. 11. Self-assembly of a single s-bot with a prey. Box-and-whisker plot [41] of the completion times (20 observations per box) grouped according to the s-bot's initial orientation and distance with respect to the prey.

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 NN causes the s-bot to turn anticlockwise 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 [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 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 s, the trial is stopped.
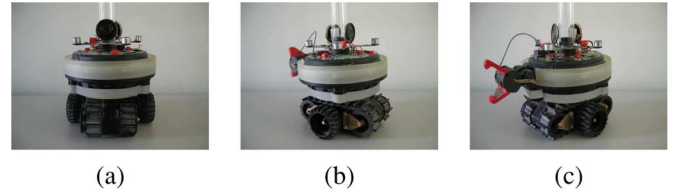

Fig. 12. Illustration of angles in which the static teammate is approached in the two s-bot experiments. (a) $0°$. (b) $60°$. (c) $120°$.
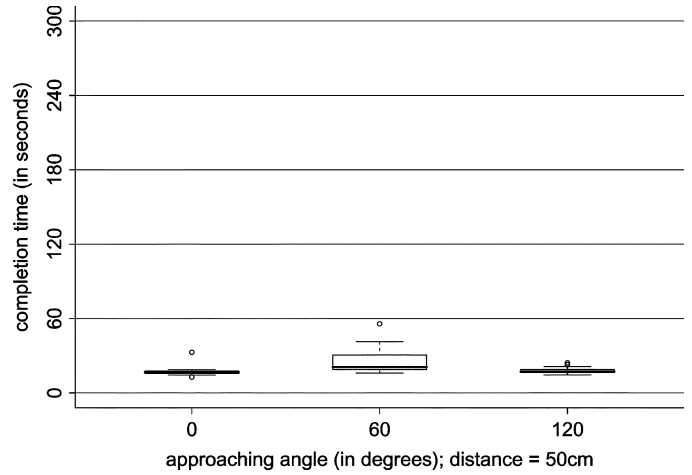

Fig. 13. Self-assembly of an s-bot with a teammate. Box-and-whisker plot [41] of the completion times (20 observations per box).

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 the object is approached for another time. 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
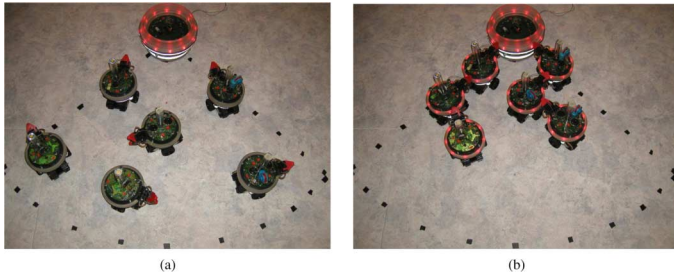
Fig. 14. Self-assembly of six s-bots with the prey. (a) Initial configuration. (b) Final configuration in a typical trial.
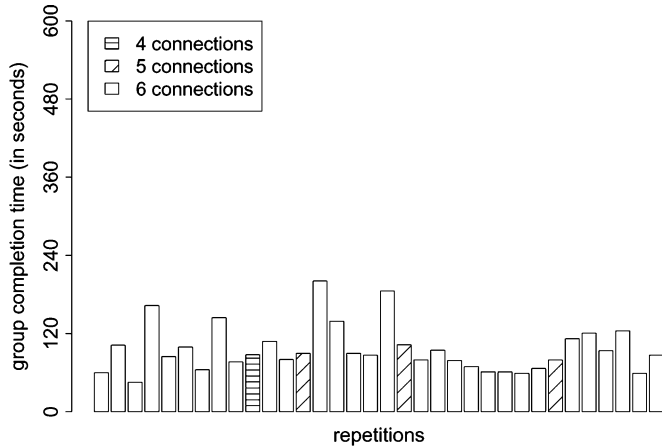

Fig. 16. Types of rough terrain. (a) Moderately rough terrain. (b) Very rough terrain.

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

Fig. 17. Self-assembly of one s-bot with a prey. Box-and-whisker plot [41] 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).

angle $0°$, $60°$, and $120°$ (and initial distance 50 cm) are 17.9, 26.4, and 17.9 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 as used in the one s-bot experiments.

1) Experimental Setup: At the beginning of each trial, the s-bots are placed at arbitrary positions[5] and orientations inside a circle of radius 70 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 s, the trial is stopped.

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 an s-bot failed only five times. At the end of 30

[5]As in simulation, the s-bots are positioned in such a way that there is a minimum distance of 20 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.
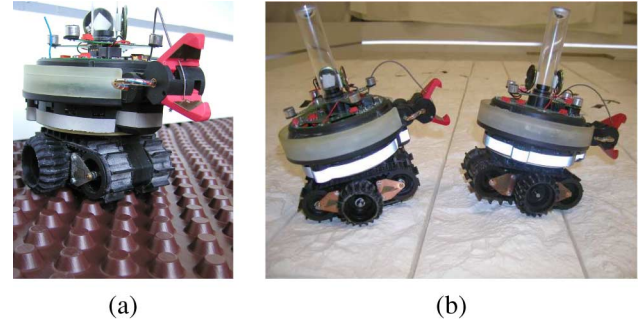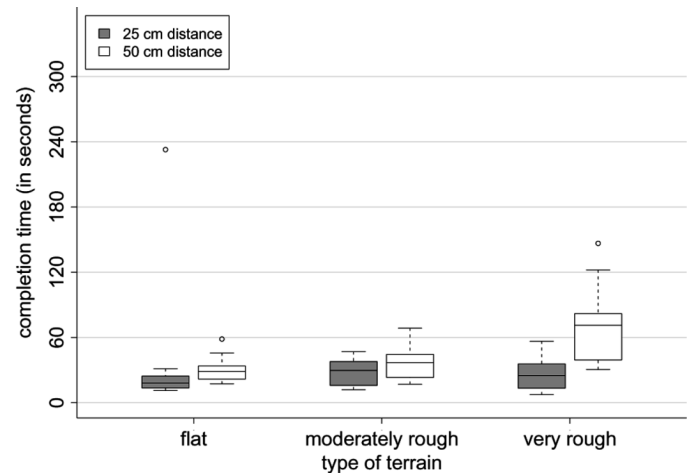
out of 34 trials, all seven objects were physically connected; on average, this took 96.4 s.

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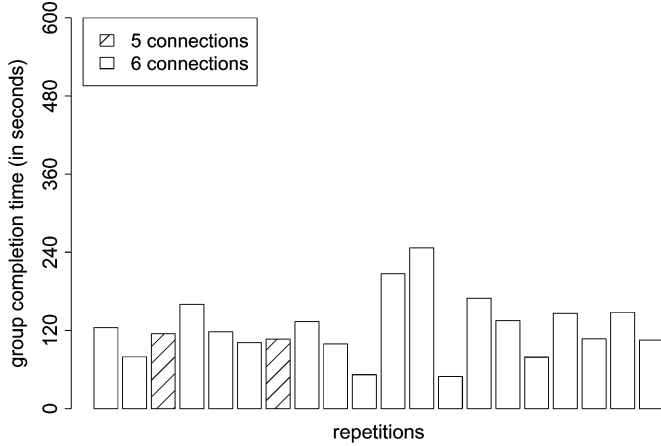
Fig. 18. Self-assembly of six s-bots with a prey on the moderately rough terrain (20 repetitions).

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

the very rough terrain, the s-bot failed only once for both initial distances (25 cm and 50 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.

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 sometimes even 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 K of our control (see Section IV-C), so that the recovery move is executed only if the torque remains high for 12 subsequent control steps (i.e., approximately 2 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, an s-bot was controlled in this experiment 120 times. 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.
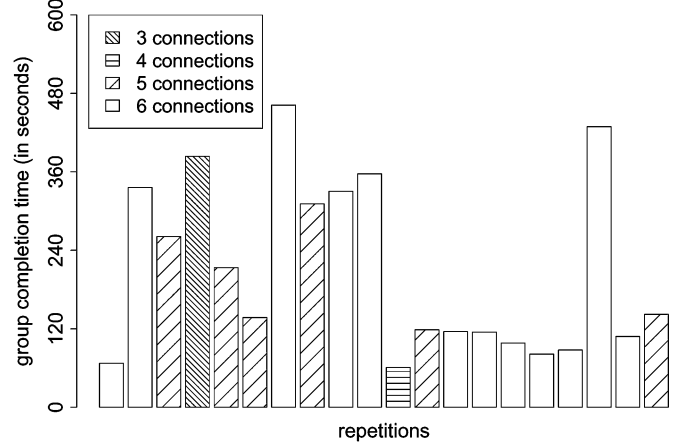
N	D	flat terrain		mod. rough terrain		very rough terrain	
		C	T	C	T	C	T
1	25	100.00	18.0	100.00	29.7	95.00	24.9
1	50	100.00	28.7	100.00	36.9	95.00	71.3
6	< 70	97.55	86.7	98.33	121.2	90.83*	115.4*

Table III summarizes the results obtained for the experiments with s-bot 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 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 six, a single s-bot, controlled by the modified controller, still connected 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 used 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 4 s it stops

Fig. 20.   Self-assembly of 16 physical s-bots put in a circle of radius 50 cm. Trial 12: (a) 19 s and (b) 104 s after activation of the seed.
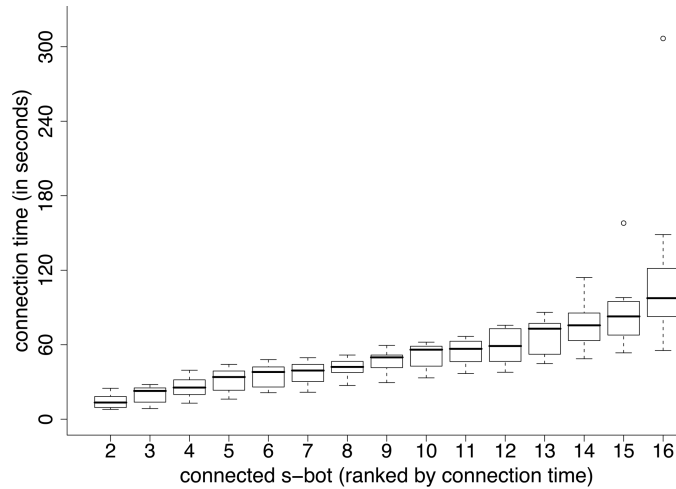

Fig. 21.   Self-assembly of 16 physical s-bots. Box-and-whisker plot [41] showing the time (counted from seed activation) after which the ith s-bot connected (observations from the 11 out of 12 trials in which all 16 s-bots successfully self-assembled).

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.[6] 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 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 cm between the centers of any two s-bots).

2) Results: We repeated the experiment 12 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

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

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.

B. Experiments With Up to 100 S-Bots in Simulation

1) Experimental Setup: We examine the problem of letting groups of 10–100 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 2-D area covered by the modules, divided by the size of the available 2-D area. The area size covered by a module (in simulation) is $A = 116$ 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.[7]

2) Results: We assess the performance of both the rule-based controller and the NN-based controller for implementing function f of our controller (see **Algorithm 1**). The performance of both controllers was assessed previously with groups of 4–16 simulated s-bots in the context of a cooperative transport task [38].

Fig. 22(a) and (b) present the percentage of the group that could successfully connect within a time period of 300 s for all group sizes and densities in 200 trials, using the rule-based controller and the NN-based controller, respectively.

In case of the two lowest densities (0.050 and 0.075), the performance for both controllers reduces drastically with group

[7]To ensure a minimum gap of about 1 cm, the s-bots are positioned so that a minimum distance of 20 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 11 congruent disks without overlapping in a unit circle, the disk radius may not exceed $r = 0.2548485$ (for a proof, see [42]). This packing would result in a module density of $(10\pi r^2)/(\pi - \pi r^2)(A)/(\pi 10^2) = 0.256$. 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 to or lower than 0.256.
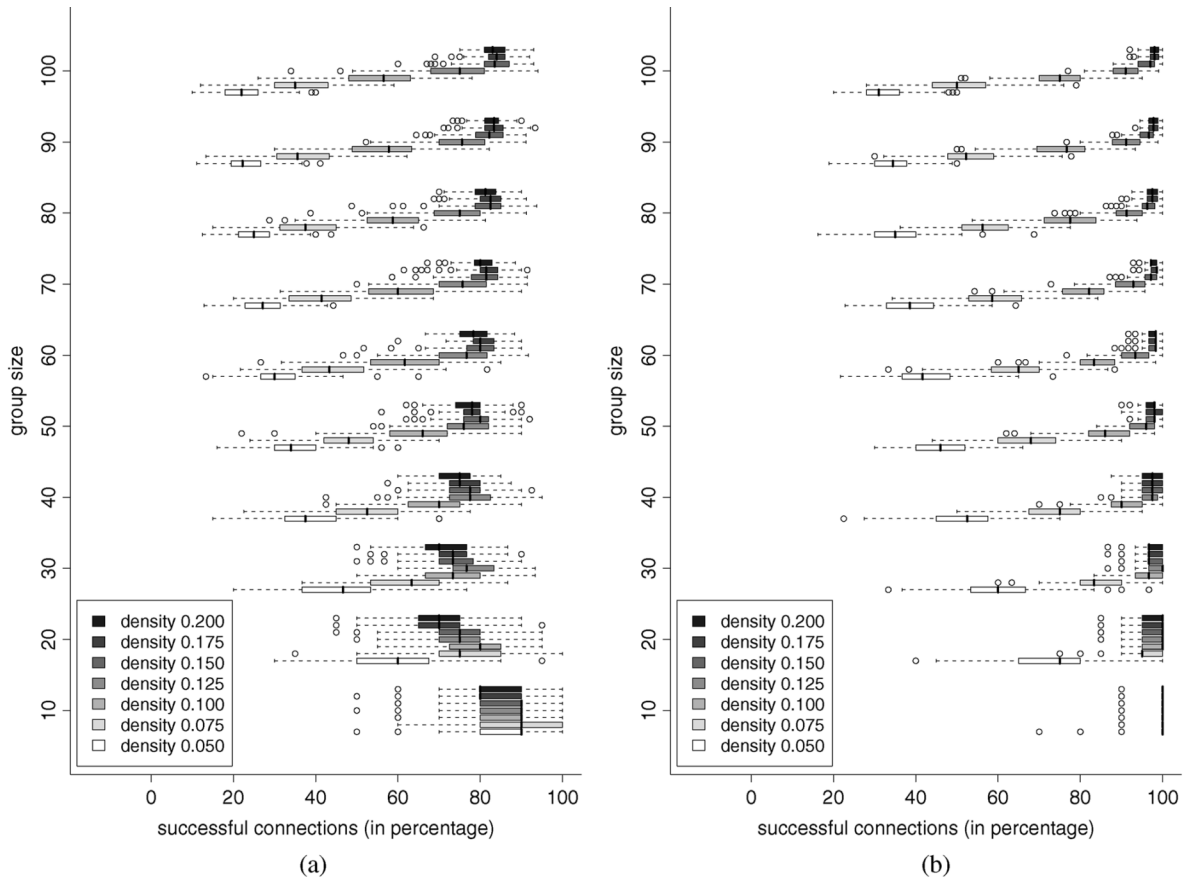
Fig. 22.   Box-and-whisker plot [41] showing the percentage of successful connections during self-assembly in a group of 10–100 s-bots, for different initial densities (200 observations per box). (a) Rule-based controller. (b) NN-based controller.
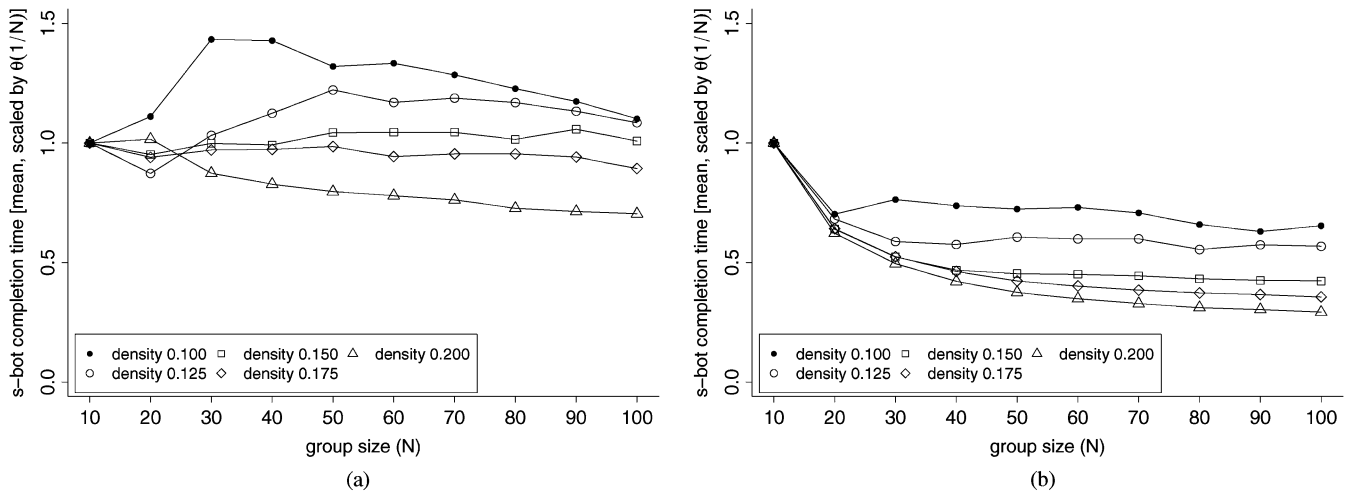

Fig. 23.   Time complexity for groups of 10–100 s-bots and different initial densities. (a) Rule-based controller. (b) NN-based controller.

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 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 NN-based controller has a particularly high success rate. In contrast, the rule-based controller's success rate drops considerably when moving from group size 10 to 20. For increasing group sizes, however, the performance tends to improve.

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 10–100 s-bots for the different densities (200 trials per situation). S-bots that have not established a connection within the predefined timeout of 300 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. 23(a) (rule-based controller) and (b) (NN-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 NN-based controller, the time grows sublinearly with the group size. This might be due to the fact that the bigger the structure, the more it provides surface for potential connections.

VIII. Discussion

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

- **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.

- **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, is a crucial factor for the design of robotic systems capable of self-assembling on rough terrain.

- **Complex Individuals Expressing Simple Collective Rules**: our s-bots might, given the current state of the art in mechatronics, be considered complex artifacts. In fact, the s-bot is equipped with a considerable amount of computational resources and is equipped with a variety of complex sensors that guide it during: 1) the approach of red objects; 2) the avoidance of blue objects; and 3) the connection phase. To some extent, the sensory system indicates also the presence of failures (e.g., in the connection). Despite the complexity of the s-bots, however, 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 NN with 15 connection weights.

- **Scalability**: as detailed in Section IV, the control is decentralized (s-bots are fully autonomous) and homogeneous (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 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 properties of the formed structures. In a test, modeling a real-world rescue scenario with 19 s-bots of approximately 700 g each, pulling a nine-year-old child of 20 kg towards a light source, it happened that the connection mechanism of an s-bot broke.[8]

IX. Ongoing Work

One of the issues we intend 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 capability of task-oriented self-assembly with the swarm-bot system to achieve the following tasks autonomously (based on the control described in Section IV).

- **Crossing a Hole**: we demonstrated the ability of up to seven s-bots to self-assemble into a single entity that crosses a hole. The hole we used is a ditch of width 10 cm from edge to edge. The ditch cannot be crossed by a swarm-bot consisting of less than three s-bots.[9]

- **Object Transport**: in a systematic experiment, six s-bots were required to pull or push a heavy but small object to a target zone.[10] By using the object as a seed for self-assembly, the s-bots organized into modular robots of up to four s-bots each, that pulled the object to the target zone [44], [45].

- **Navigation Over a Hill**: in a systematic experiment, three s-bots were required to navigate over unknown terrain towards a light source. If possible, the s-bots navigated to the target independently. If, however, the terrain proved too difficult for a single s-bot, the group self-assembled into a larger entity and collectively navigated to the light source [46].

In these studies, we observed that the performance of the system may depend on the morphology of the swarm-bot. For example, of all formations observed during object transport, s-bots organized in a linear chain formation seemed to exhibit the highest pulling force. For navigating an unstructured rough terrain, however, other formations might be preferable. It is, therefore, particularly interesting to study mechanisms for task-oriented reconfiguration to let a swarm-bot autonomously change its morphology in response to the demands of the environment.

X. Conclusion

This paper presented a comprehensive study of the self-assembling capabilities of the swarm-bot, a distributed robotic system that lies at the intersection between collective and

---

[8]Video recordings are available at http://swarm-bots.org/selfassembly/.

[9]Theoretical lower bounds for the size of swarm-bots crossing holes have been derived in [43].

[10]Due to the limited surface of the transported object (20-cm diameter), pushing behaviors with more than two s-bots were ineffective. Due to frictional forces, the object required the cooperative effort of four or more s-bots to be moved.

self-reconfigurable robotics. The s-bot, the basic module of the swarm-bot, is a fully self-contained, mobile robot integrating an effective connection mechanism, adequate computing resources, and advanced sensing and communication devices.

The contributions that make our work unique are:

- implementation of a modular robot, the modules of which (35 in total) can receive connections on more than two-thirds of their perimeter;
- integration of more than 50 sensors in each module, including proximity sensors, optical barriers, a force sensor, a torque sensor, humidity and temperature sensors, an inclinometer, light sensors, omnidirectional microphones, motor position and torque sensors, and a VGA omnidirectional camera;
- demonstration of self-assembly with component modules that are fully autonomous in perception, control, action, and power;
- systematic performance evaluation of self-assembly of more than two modules (up to 16 modules);[11]
- demonstration (and systematic performance evaluation) of self-assembly on rough terrain (with up to six modules);
- implementation of a self-assembly mechanism that scales well with group size: on average, a module assembled in 98%–100% of the trials (with up to 16 modules), and with a sublinear time complexity (with up to 100 modules in simulation).

Given the breadth and quality of the results obtained, we can say that the swarm-bot qualifies as the current state of the art in autonomous self-assembly.

ACKNOWLEDGMENT

REFERENCES

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

[7] M. Yim, "Modular reconfigurable robotics: What's next?," presented at the US-Korea Conf. Sci., Technol., Entrepreneurship, Durham, NC, 2004 [Online]. Available: http://www.ksea.org.

[8] 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.

[9] 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, Jul. 2002.

[10] M. Rubenstein, K. Payne, P. Will, and W.-M. Shen, "Docking among independent and autonomous CONRO self-reconfigurable robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, Los Alamitos, CA, 2004, vol. 3, pp. 2877–2882.

[11] E. Mytilinaios, M. Desnoyer, D. Marcus, and H. Lipson, "Designed and evolved blueprints for physical self-replicating machines," in *Proc. 9th Int. Conf. Simul. Synthesis Living Syst. (Artif. Life IX)*, Cambridge, MA, 2004, pp. 15–20.

[12] V. Zykov, E. Mytilinaios, B. Adams, and H. Lipson, "Self-reproducing machines," *Nature*, vol. 435, no. 7039, pp. 163–164, 2005.

[13] T. Fukuda and S. Nakagawa, "A dynamically reconfigurable robotic system (concept of a system and optimal configurations)," in *Proc. IEEE Int. Conf. Ind. Electron., Control, Instrum.*, Los Alamitos, CA, 1987, pp. 588–595.

[14] T. Fukuda and T. Ueyama, *Cellular Robotics and Micro Robotic Systems*. London, U.K.: World Scientific, 1994.

[15] T. Fukuda and S. Nakagawa, "Approach to the dynamically reconfigurable robotic system," *J. Intell. Robot. Syst.*, vol. 1, no. 1, pp. 55–72, 1988.

[16] T. Fukuda, S. Nakagawa, Y. Kawauchi, and M. Buss, "Self organizing robots based on cell structures—CEBOT," in *Proc. IEEE Int. Workshop Intell. Robots*, Los Alamitos, CA, 1988, pp. 145–150.

[17] T. Fukuda, M. Buss, H. Hosokai, and Y. Kawauchi, "Cell structured robotic system CEBOT: Control, planning and communication," *Robot. Auton. Syst.*, vol. 7, no. 2-3, pp. 239–248, 1991.

[18] T. Fukuda, K. Sekiyama, T. Ueyama, and F. Arai, "Efficient communication method in the cellular robotic system," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Los Alamitos, CA, 1993, vol. 2, pp. 1091–1096.

[19] T. Fukuda, T. Ueyama, and K. Sekiyama, *Artificial Intelligence in Industrial Decision Making, Control and Automation*. Dordrecht, The Netherlands: Kluwer, 1995, ch. 8.

[20] T. Fukuda, T. Ueyama, and Y. Kawauchi, "Self-organization in cellular robotic system (CEBOT) for space application with knowledge allocation method," in *Proc. Int. Symp. Artif. Intell., Robot. Autom. Space*, Kobe, Japan, 1990, pp. 101–104.

[21] 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.

[22] C. Bererton and P. K. Khosla, "Towards a team of robots with repair capabilities: A visual docking system," in *Proc. 7th Int. Symp. Exp. Robot.*, Berlin, Germany, 2001, vol. 271 of LNCIS, pp. 333–342.

[23] ——, "Towards a team of robots with reconfiguration and repair capabilities," in *Proc. IEEE Int. Conf. Robot. Autom.*, Los Alamitos, CA, 2001, vol. 3, pp. 2923–2928.

[24] S. Hirose, R. Damoto, and A. Kawakami, "Study of super-mechano-colony (concept and basic experimental setup)," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Los Alamitos, CA, 2000, vol. 3, pp. 1664–1669.

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

[26] M. Yamakita, Y. Taniguchi, and Y. Shukuya, "Analysis of formation control of cooperative transportation of mother ship by SMC," in *Proc. IEEE Int. Conf. Robot. Autom.*, Los Alamitos, CA, 2003, vol. 1, pp. 951–956.

[27] 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.

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

[29] P. J. White, K. Kopanski, and H. Lipson, "Stochastic self-reconfigurable cellular robotics," in *Proc. IEEE Int. Conf. Robot. Autom.*, Los Alamitos, CA, 2004, vol. 3, pp. 2888–2893.

[30] 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. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Los Alamitos, CA, 2005, pp. 2644–2651.

[31] S. T. Griffith, "Growing machines," Ph.D. dissertation, MIT, Cambridge, MA, 2004.
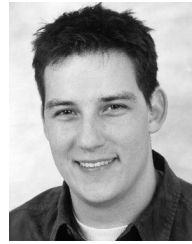
---

[11]Two supplementary multimedia movie clips (playable with Quicktime Player 7), which show, respectively, self-assembly of up to 16 physical s-bots on flat and rough terrain, are available at http://ieeexplore.ieee.org.

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

[33] P. White, V. Zykov, J. Bongard, and H. Lipson, "Three dimensional stochastic reconfiguration of modular robots," in *Proc. Robot.: Sci. Syst. Conf.*, Cambridge, MA, 2006, pp. 161–168.

[34] 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.

[35] 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.

[36] F. Mondada, M. Bonani, S. Magnenat, A. Guignard, and D. Floreano, "Physical connections and cooperation in swarm robotics," in *Proc. 8th Int. Conf. Intell. Auton. Syst.*, Amsterdam, The Netherlands, 2004, pp. 53–60.

[37] 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.

[38] R. Groß and M. Dorigo, "Group transport of an object to a target that only some group members may sense," in *Proc. 8th Int. Conf. Parallel Problem Solving from Nature*, Berlin, Germany, 2004, vol. 3242 of LNCS, pp. 852–861.

[39] H.-P. Schwefel, "Evolutionsstrategie und numerische Optimierung," Ph.D. dissertation, Technische Univ., Berlin, Germany, 1975.

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

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

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

[43] F. Mondada, M. Bonani, A. Guignard, S. Magnenat, C. Studer, and D. Floreano, "Superlinear physical performances in a SWARM-BOT," in *Proc. 8th Eur. Conf. Artif. Life*, Berlin, Germany, 2005, vol. 3630 of LNAI, pp. 282–291.

[44] R. Groß, E. Tuci, M. Dorigo, M. Bonani, and F. Mondada, "Object transport by modular robots that self-assemble," in *Proc. IEEE Int. Conf. Robot. Autom.*, Los Alamitos, CA, 2006, pp. 2558–2564.

[45] E. Tuci, R. Groß, V. Trianni, M. Bonani, F. Mondada, and M. Dorigo, "Cooperation through self-assembling in multi-robot systems," *ACM Trans. Auton. Adapt. Syst.*, vol. 1, no. 2, 2006.

[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 *Proc. 8th Eur. Conf. Artif. Life*, Berlin, Germany, 2005, vol. 3630 of LNAI, pp. 272–281.

Michael Bonani received the M.Sc. degree in microengineering in 2003 from Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, where he is currently working toward the Ph.D. degree in microengineering.

In 2003 and 2004, he was a Research Associate with the SWARM-BOTS project. Since 2005, he has been developing an educational robot called e-puck at EPFL. His research interests include swarm robotics and educational robotics.
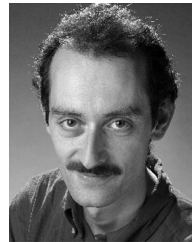
Francesco Mondada (S'93–A'95) received the M.Sc. degree in microengineering and the Ph.D. degree from the Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland.

During his thesis, he cofounded the company K-Team, being both CEO and President of the company for about 5 years. He is one of the three main developers of the Khepera robot, considered as a standard in bio-inspired robotics and used by more than 1000 universities and research centers worldwide. Fully back in research in 2000 and after a short period at the California Institute of Technology, Pasadena, he has been the main developer of the s-bot platform within the SWARM-BOTS project. He published more than 50 papers in the field of bio-inspired robotics and system-level robot design. He is coeditor of several international conference proceedings. He is currently a senior researcher at the EPFL. His interests include the development of innovative mechatronic solutions for mobile and modular robots, the creation of know-how for future embedded applications, and making robot platforms more accessible for education, research, and industrial development.

In 2005, Dr. Mondada was awarded the Swiss Latsis University prize for his contributions to bio-inspired robotics.

Marco Dorigo (S'92–M'93–SM'96–F'06) received the Laurea (Master of Technology) degree in industrial technologies engineering in 1986, and the doctoral degree in information and systems electronic engineering in 1992, both from the Politecnico di Milano, Milan, Italy, and the title of Agrégé de l'Enseignement Supérieur, from the Université Libre de Bruxelles, Brussels, Belgium, in 1995.

From 1992 to 1993, he was a Research Fellow at the International Computer Science Institute, Berkeley, CA. In 1993, he was a NATO-CNR Fellow, and from 1994 to 1996, a Marie Curie Fellow. Since 1996, he has been a tenured researcher of the FNRS, the Belgian National Fund for Scientific Research, and a research director of IRIDIA, the artificial intelligence laboratory of the Université Libre de Bruxelles. He is the inventor of the ant colony optimization metaheuristic. His current research interests include metaheuristics for discrete optimization, swarm intelligence and swarm robotics.

Dr. Dorigo is an Associate Editor for the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, and for the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS. He is a member of the Editorial Board of numerous international journals, including *ACM Transactions on Autonomous and Adaptive Systems, Adaptive Behavior, AI Communications, Artificial Life, Computers & Industrial Engineering, Evolutionary Computation, Information Sciences, Journal of Heuristics, Cognitive Systems Research,* and *Journal of Genetic Programming and Evolvable Machines*. In 1996, he was awarded the Italian Prize for Artificial Intelligence, in 2003, the Marie Curie Excellence Award, and in 2005, the Dr. A. De Leeuw-Damry-Bourlart award in applied sciences.


Roderich Groß (S'05) received the Diploma degree (*summa cum laude*) in computer science in 2001 from Universität Dortmund, Dortmund, Germany. Since 2002, he has been working toward the Ph.D. degree at the Université Libre de Bruxelles, Brussels, Belgium.

In 2005, he held a short-term JSPS postdoctoral research fellowship at the Tokyo Institute of Technology, Tokyo, Japan. His research interests include robotics and artificial life.