Auton Robot
DOI 10.1007/s10514-010-9177-0

Self-assembly strategies in a group of autonomous mobile robots

Rehan O'Grady · Roderich Groß · Anders Lyhne Christensen · Marco Dorigo

Received: 6 July 2008 / Accepted: 11 January 2010

Abstract Robots are said to be capable of self-assembly when they can autonomously form physical connections with each other. By examining different ways in which a system can use self-assembly (i.e., different *strategies*), we demonstrate and quantify the performance costs and benefits of (i) acting as a physically larger self-assembled entity, (ii) letting the system choose when and if to self-assemble, (iii) coordinating the sensing and actuation of the connected robots so that they respond to the environment as a single collective entity. Our analysis is primarily based on real world experiments in a hill crossing task. The configuration of the hill is not known by the robots in advance—the hill can be present or absent, and can vary in steepness and orientation. In some configurations, the robots can overcome the hill more quickly by navigating individually, while other configurations require the robots to self-assemble to overcome the hill. We demonstrate the applicability of our self-assembly strategies to two other tasks—hole crossing and robot rescue—for which we present further proof-of-concept experiments with real robots.

Keywords Self-assembly · All-terrain navigation · Cooperation · Autonomous robots · Modular robots · Swarm robotics

Electronic supplementary material The online version of this article (http://dx.doi.org/10.1007/s10514-010-9177-0) contains supplementary material, which is available to authorized users.

R. O'Grady (✉) · M. Dorigo
IRIDIA, CoDE, Université Libre de Bruxelles, Brussels, Belgium
e-mail: rogrady@ulb.ac.be

M. Dorigo
e-mail: mdorigo@ulb.ac.be

R. Groß
ACSE, The University of Sheffield, Sheffield, UK
e-mail: r.gross@sheffield.ac.uk

A.L. Christensen
Instituto de Telecomunicações, Lisbon, Portugal
e-mail: anders.christensen@iscte.pt

1 Introduction

In the pursuit of scalability, flexibility and parallelism, many robotics researchers have proposed systems based on simple components that self-organise through local interactions (Bonabeau et al. 1999). Two parallel branches have emerged that pursue this philosophy. In distributed multi-robot systems, the base components of the system are individual robots that can either work in parallel or cooperate to solve more complex tasks (Cao et al. 1997; Shen et al. 2004). In modular robotic systems, the base components are robotic modules that can be combined in different ways to create robotic entities of different sizes and shapes (and that once assembled can often autonomously reconfigure) (Yim et al. 2003).

Self-assembling robotic systems have the potential to leverage the benefits of both of the above approaches (Groß and Dorigo 2008b). The components in self-assembling systems are independent robots that can form physical connections with one another. Future self-assembling systems could carry out some tasks in parallel by allowing the robots in the system to act on their environment individually. Other more physically demanding tasks could be carried out by composite robotic entities of the appropriate size and shape formed on the fly through self-assembly. Autonomous self-assembly does, nonetheless, have an associated cost—the robotic components need to be more sophisticated, and time, energy and computational resources are used during the self-assembly process. This cost makes self-assembling robots inappropriate in scenarios where the size, number and shape of required robotic entities are known in advance—in such cases it is just as effective to use simpler and cheaper pre-assembled robotic entities. Robotic entities that self-assemble are, however, justified in situations where a priori knowledge of the environment is limited, and self-assembly is used as a response mechanism based on environmental contingencies.

To the best of our knowledge, this is the first study to consider the problem of using self-assembly as a response mechanism in a real-world robotic system. Existing research in self-assembling systems has almost completely neglected the problem of autonomously controlling the timing and nature of the self-assembly process and the subsequent deployment of the assembled entity. Research to date has largely focused on the initial problem of autonomously creating physical connections between robots (with little or no consideration of subsequent functional use) (Groß and Dorigo 2008b). The few experiments that have attempted to solve real world tasks almost always take for granted that all participating agents must self-assemble as a first step in the task execution process.

In this paper, we apply different self-assembly *strategies* to a range of tasks. The different strategies represent varying degrees of autonomous control that the system can exercise over the self-assembly process and the deployment of the resulting collective robotic entity. We present a detailed quantitative analysis of experiments conducted on a real world hill crossing experiment that requires a group of up to three robots to navigate towards a target light source over a priori unknown terrain. In some of our experimental environments, the robots do not encounter a hill, or encounter a simple hill that can be navigated by a single robot individually. In other environments, the hill encountered is sufficiently steep that the robots must physically connect to one another (self-assemble) and navigate as a connected entity to overcome the hill successfully. We analyse both the costs (overheads) and benefits (improvements in efficiency) of the different strategies, and discuss the conditions under which autonomous self-assembly is an appropriate solution to a task. We also demonstrate the application of our strategies to two further tasks—hole crossing and robot rescue.

This paper is organised as follows. In Sect. 2, we discuss related work and the state of the art in self-assembling systems in more detail. In Sects. 3 and 4, we present the swarm-bot robotic platform that we use in our experimentation and the experimental set-up. In Sect. 5, we present the *basic self-assembly response* strategy. This strategy allows the system to trigger the self-assembly process when individual robots prove incapable of solving a task in parallel. In Sect. 6, we use the *basic self-assembly response* strategy to analyse the benefits of acting as a physically larger connected entity. In Sect. 7, we use the *basic self-assembly response* strategy to analyse the benefits of allowing the system to choose when and if to self-assemble. In Sect. 8, we present the *connected coordination* strategy. This strategy coordinates the sensing and actuation of the assembled robots to allow them to respond to the environment as a single collective entity. In Sect. 9, we analyse the costs and benefits of the *connected coordination* strategy. In Sect. 10, we consider the wider context of our work. We present further real world experimentation concerning the scalability of our system, and the applicability of the strategies we presented to two further tasks—hole crossing and robot rescue. We also consider future avenues of research to achieve a higher level of adaptivity. Finally, in Sect. 11, we present our conclusions. Additional material including videos of experiments and full details of distributed control can be found in the online supplementary material or on the support page (O'Grady et al. 2010).

2 Related work and the state of the art

Self-assembly processes are responsible for the generation of much of the order that we observe in nature. Such processes involve components at a variety of different scales, including molecules, cells, organisms, and weather systems (Caspar 1966; Sendova-Franks and Franks 1999; Whitesides and Grzybowski 2002; Anderson et al. 2002). In the past 50 years, many researchers have designed and studied modular systems whose components—ranging from passive mechanical parts to mobile robots—can self-assemble into physically connected structures (Groß and Dorigo 2008b).

Penrose and Penrose (1957), for example, built a system of modules made of wood that represents the first mechanical analogue for self-replication, and thereby showed "how reproduction can be demonstrated by an exceedingly simple mechanism". In the 1980's, Fukuda and Nakagawa (1988), inspired by (biological) cellular organisms, proposed the concept of "dynamically reconfigurable robotic system"—a pioneering work that laid the foundation for subsequent research on modular robotic systems and multi-robot systems. The authors anticipated potential applications of such modular systems "in many fields, e.g. maintenance robots, more advanced working robots, free-flying service robots in space, more evolved flexible automation, etc.". Hirose et al. (1996) investigated a modular robot concept for planetary exploration, and described potential benefits of such systems in the context of autonomous all-terrain locomotion. For example, "a single unit by itself will fall off into the crevice, but if it is a connected body, falling can be prevented". On the other hand, "the torso may be separated into several groups, and each of those groups can function as an autonomously distributed group robot". Yim et al. (2000) predicted that such systems would be particularly suited to applications in which versatility is critical. "Typically, these are situations in which some information about the environment is not known a priori. Thus, a system cannot be designed specifically for a task, since the task that is needed is not known".

A large body of research has already shown that the component modules of some modular robotic systems can be set up in a range of configurations to address a variety of tasks under a range of environmental circumstances—see for example Castano et al. (2002), Murata et al. (2002), Yim et al. (2003), Brown et al. (2002), Yamakita et al. (2003), Mumm et al. (2004), Kamimura et al. (2005), Mondada et al. (2005), Østergaard et al. (2006), Shen et al. (2006). However, very little research has been conducted into how the modules of a robotic system can autonomously organise themselves into configurations that are well suited to their task and environment.

Our previous research with physical and simulated modular robotic systems shows that self-assembly can provide a group of robotic modules with additional capabilities in a simple object manipulation task—the transport of an object (Groß and Dorigo 2004, 2008a; Groß et al. 2006b; Tuci et al. 2006). We also demonstrated that self-assembly can provide a group of modules with additional capabilities in all-terrain navigation (O'Grady et al. 2005; Groß et al. 2006a). However, none of these studies quantify the contribution of self-assembly to system performance. The problem of using self-assembly as an autonomous response mechanism has been investigated by Trianni et al. (2004) in a highly simplified simulated environment (see also Tuci et al. 2006). In their study, a group of three simulated robots had to perform phototaxis across a terrain composed of discrete low and high temperature zones. Offline evolution was used to generated neural network controllers using a fitness function that allocated the highest fitness to the robots when they navigated individually through the high temperature zones and collectively through the low temperature zones. Our current work differs in that it is conducted using a real-world robotic system and our notions of cost and benefit have a real world meaning.

3 The swarm-bot robotic platform

For our experiments, we use the SWARM-BOT robotic platform (Mondada et al. 2004). This platform is made up of multiple mobile autonomous robots called s-bots (see Fig. 1) that can form physical connections with each other. The entity formed by two or more connected s-bots is called a swarm-bot. The s-bot is 12 cm high without its camera turret, and has a diameter of about 12 cm without its connection mechanism. Thanks to its traction system that combines tracks and wheels, the s-bot has good mobility on uneven terrain whilst still retaining the ability to rotate on the spot efficiently. The main s-bot body houses most of its sensory and processing systems and can rotate with respect to the chassis by means of a motorised axis.

Physical connections between s-bots are established by a gripper-based connection mechanism. Each s-bot is surrounded by a transparent ring that can be grasped by other s-bots. An optical light barrier inside the s-bot gripper indicates when another s-bot's ring (or another object) is between the jaws of the gripper. S-bots advertise their location by means of eight sets of RGB coloured LEDs (Light Emitting Diodes) distributed around the inside of their transparent ring. These LEDs can also provide indications of the s-bot's internal state to other nearby s-bots.

The s-bot has an omni-directional camera that, depending on light conditions, can detect other s-bots' LEDs up to

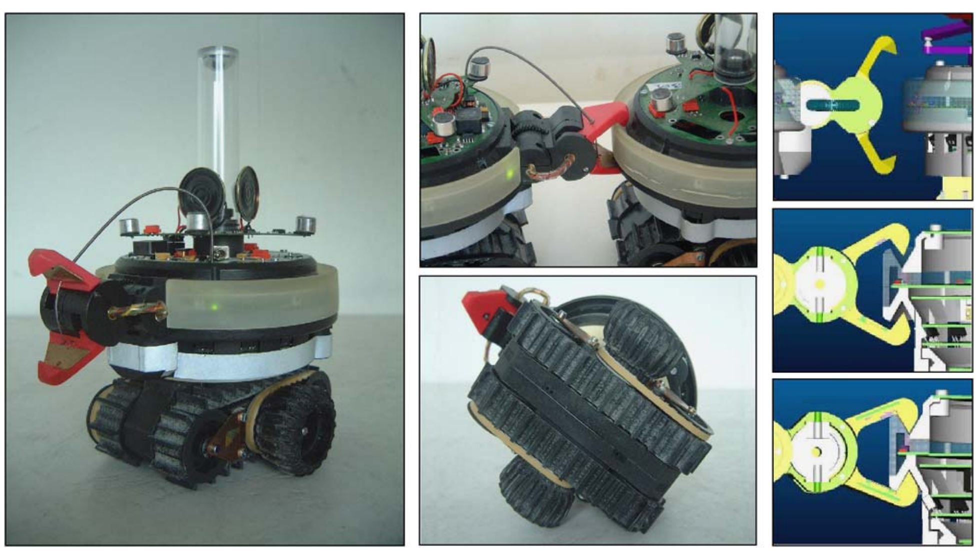


Fig. 1 *Left*: The s-bot. *Centre Top*: The s-bot connection mechanism. *Centre Bottom*: The s-bot traction system. *Right*: Cross section of the s-bot connection mechanism in action

about 40 cm away or an external light source up to about 400 cm away. The s-bot has 15 infra-red proximity sensors distributed around its body that allow for the detection of obstacles. Ground facing proximity sensors under the tracks allow the s-bot to detect whether or not it is over a hole and, to a limited extent, the width of the hole it is over. A 3-axis accelerometer provides information on the s-bot's inclination that can be used to detect if the s-bot is in danger of toppling over. Other sensors provide the s-bot with proprioceptive information about its internal motors. This includes positional information (e.g., of the rotating turret) and torque information (e.g., of forces acting on the traction system).

4 Experimental setup (hill crossing task)

The hill crossing task requires a group of between one and three s-bots to navigate over a priori unknown terrain to a *target area* containing a light source. The s-bots are considered to have completed the task if they reach the target area without toppling over.

We conduct experiments in seven different environments (see Fig. 2). Six of the seven environments contain a hill. We use two types of hill—a 'moderate' hill and a 'difficult' hill. The moderate hill can be overcome by a single s-bot. S-bots can only navigate the difficult hill as part of a larger self-assembled entity (the steepness of the hill would cause a single s-bot to topple). Both the moderate hill and the difficult hill can be oriented in three different ways.

The hill and the arena boundaries together demarcate the *starting area* and the target area. In the no-hill environment, the starting area and target area are considered to be the same as for environments with hill orientation 90° (see Fig. 2). In each experiment, the starting positions of the participating s-bots are assigned randomly by uniformly sampling without replacement from a set of 12 possible starting points in the starting area. The initial orientations are chosen randomly from a set of 4 possible directions.

The robotic system investigated in our study has the following features:

- each s-bot is autonomous in power, perception, control and action,
- each s-bot has no a priori knowledge of its environment or of its initial position and orientation,
- at the start of each experiment, an identical controller is copied onto each of the s-bots and executed on each of the s-bots independently,
- communication (when used) is visual and strictly local—the s-bots illuminate their LED rings with different colours to advertise their relative location and to provide indications of their internal state to other s-bots within visual range.

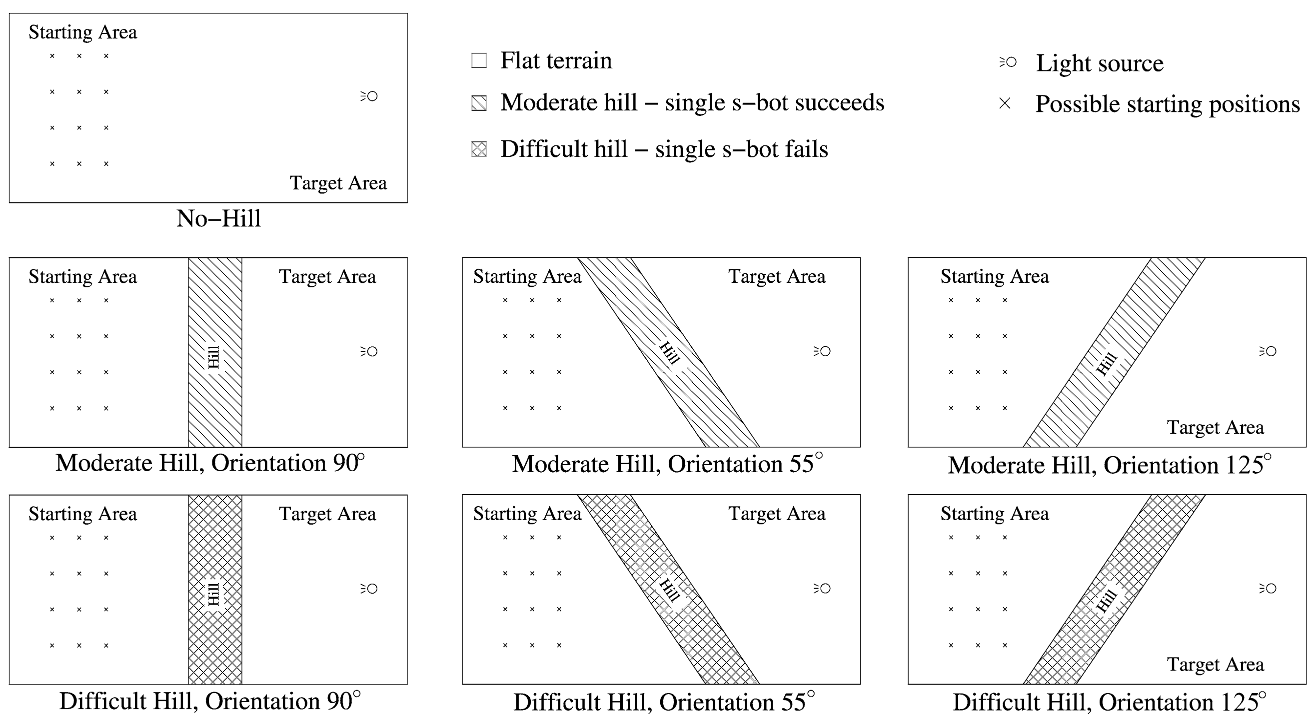


Fig. 2 Scale diagram of the seven environments used in this study. Each environment measures 210 cm × 105 cm. Three environments contain the 'moderate' hill (2.8 cm high, navigable by a single s-bot). Three environments contain the 'difficult' hill (6.5 cm high, not navigable by a single s-bot). One environment has no hill. Starting positions are marked by *crosses*

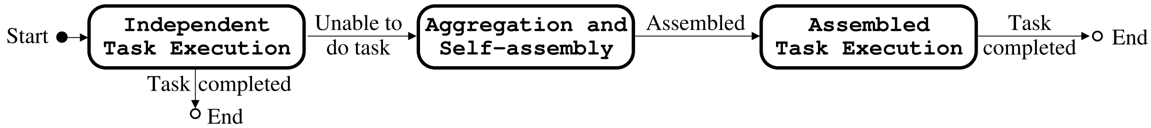


Fig. 3 *Basic self-assembly response* strategy: group-level behaviour

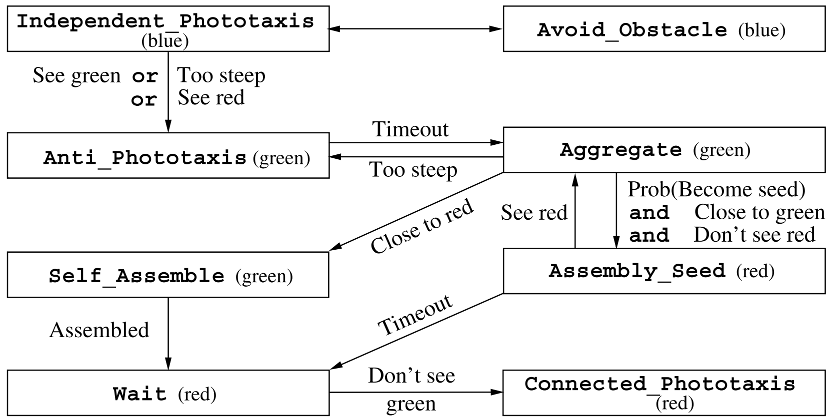


Fig. 4 Distributed control to implement the *basic self-assembly response* strategy for the hill crossing task. This finite state machine controller is executed independently on individual robots. The starting state is `Independent_Phototaxis`. Colours in parentheses refer to the LEDs that are illuminated in the corresponding state

Fig. 5 A group of s-bots using the *basic self-assembly response* strategy in an environment containing a hill that is not navigable individually. (**A**): Three s-bots start from random positions and orientations. Initially, they perform independent phototaxis, and have their blue LEDs illuminated. One s-bot detects a slope it cannot overcome alone and illuminates its green LEDs. (**B**): The other s-bots detect colour green (local communication). All of the s-bots retreat away from the light source for a fixed time period. One of the robots probabilistically becomes the seed for the self-assembly process. (**C**): The s-bots self-assemble. (**D**, **E**): The s-bots perform connected phototaxis towards the light source and successfully overcome the obstacle to arrive in the target area. For a video of this experiment, see file 3_Sbots_Cross_Hill in the online supplementary material

5 The *basic self-assembly response* strategy

In this section, we describe the *basic self-assembly response* strategy and present the distributed control that enables the corresponding group level behaviour for our hill crossing task. The *basic self-assembly response* strategy is illustrated in Fig. 3. The robots start by trying to execute a task independently. If they fail to complete the task independently, they self-assemble and attempt to solve the task as a larger composite robotic entity.

5.1 Strategy implementation for the hill crossing task

In our implementation, the s-bots start to navigate independently towards the target light source. In the absence of any steep hill, the robots complete the task independently. If, however, the robots detect a hill that is too steep to be navigated individually, they first aggregate, then self-assemble into a larger connected entity. The robots then navigate collectively to the target light source.

The distributed control we use to implement the strategy is shown in Fig. 4. Figure 5 shows a three s-bot system using the strategy. An s-bot starts by illuminating its blue LEDs and navigating independently towards the light source in the target area (state `Independent_Phototaxis`). The light source is detected with the camera. While navigating towards the light source, the s-bot uses its proximity sensors to move away from arena walls and other robots that are too close (state `Avoid_Obstacle`). If the s-bot finds itself on a hill too difficult for it to pass alone (detected using its 3D accelerometers), or if it sees (i.e., detects with its camera) a green s-bot or a red s-bot, it illuminates its green LEDs and retreats away from the hill for a given length of time (state `Anti_Phototaxis`). It then switches into state `Aggregate`, and tries to get close to a red s-bot (the seed or an s-bot already assembled to the seed), or if no red s-bot is perceived, to search for and get close to another green (aggregating) s-bot. In the latter case, if the s-bot is sufficiently close to another green s-bot and can still see no other red s-bots, it can become, with a given probability, the seed that triggers the process of self-assembly (state `Assembly_Seed`). A seed s-bot lights up its red LEDs, and waits until a timeout has expired. If it sees another red s-bot within the timeout period, it reverts to state `Aggregate`. Otherwise, after the timeout it switches to state `Wait`. If an aggregating s-bot gets sufficiently close to a red (assembled) s-bot, then it starts self-assembling (state `Self_Assemble`). Assembled s-bots switch to state `Wait`. S-bots in state `Wait` illuminate their red LEDs. When they no longer see any green (aggregating or assembling) s-bots, they switch to state `Connected_Phototaxis` and navigate collectively to the light source. During collective navigation, each connected s-bot performs greedy phototaxis (i.e., each s-bot heads in a direct line towards the light source). After self-assembly, however, the orientations of the constituent s-bots' turrets are fixed. Each s-bot, therefore, continually rotates its turret to compensate for the track movements that move the s-bot towards the light source (Groß et al. 2006b). For a detailed description of the individual states in this finite state machine see O'Grady et al. (2009b, 2010).

6 Benefits of scale

In this section, we describe experiments we conducted with the *basic self-assembly response* strategy to analyse the simple scale benefits of acting as a physically larger self-assembled entity. To provide a basis for comparison, we also conducted control experiments with a simple strategy that causes the s-bots to carry out the task independently, irrespective of environmental conditions.

6.1 The *independent execution only* strategy

This strategy is illustrated in Fig. 6. The distributed control to implement this strategy for the hill crossing task is a modified version of the distributed control for the *basic self-assembly response* strategy in which the transition to state `Anti_Phototaxis` is disabled (see Fig. 4). Thus, only the states `Independent_Phototaxis` and `Avoid_Obstacle` are executed. The result is that each s-bot moves independently towards the light at a constant speed irrespective of the type of terrain it encounters.

Fig. 6 *Independent execution only* strategy: group-level behaviour

6.2 Results

We conducted 60 trials with a single s-bot using the *independent execution only* strategy in the difficult hill environments (20 trials per hill orientation). The s-bot failed to overcome the hill in all 60 trials. In each trial, the s-bot reached the hill and then toppled backwards due to the steepness of the slope. To confirm that the s-bot was failing due to the intrinsic properties of the slope, we repeated this experiment at a number of different constant speeds and observed the same result.

We conducted 60 trials with 2 s-bots using the *basic self-assembly response* strategy in the difficult hill environments (20 trials per hill orientation). Both s-bots successfully self-assembled into a two s-bot swarm-bot in every trial. In 21 trials (35%), the swarm-bot succeeded in overcoming the hill, and thus completed the task. In the other 39 trials (65%), the assembled swarm-bot failed to overcome the hill. These failures happened when the assembled swarm-bot moved towards the light source with an orientation overly parallel to the orientation of the hill and thus toppled over.

We conducted 60 trials with 3 s-bots using the *basic self-assembly response* strategy in the difficult hill environments (20 trials per hill orientation). In 47 trials (78.3%), all 3 s-bots succeeded in self-assembling and overcoming the hill. A number of different types of failure occurred. In 9 trials, all three s-bots toppled over on the hill because they self-assembled into a roughly linear swarm-bot and also approached the hill with an orientation nearly parallel to that of the hill (the same type of failure that we witnessed in the 2 s-bot trials). In some trials, the self-assembly process itself failed (the addition of an extra s-bot makes the self-assembly process more complex): in 2 trials, a single s-bot failed to complete the task when only 2 out of the 3 s-bots succeeded in self-assembling into a connected entity (the 2 s-bot swarm-bot went on to complete the task). In 1 further trial, all three s-bots failed when again only 2 out of the 3 s-bots successfully self-assembled, but the 2 s-bot swarm-bot toppled on the hill. In a single trial, an s-bot that was part of the three s-bot swarm-bot detached on the hill due to a faulty grip, and thus toppled over. In total, 147 of the 180 participating robots successfully completed the task (81.7%).

The results show that navigating as a two s-bot swarm-bot instead of individually caused the task completion rate to increase from 0% to 35% (a significant increase according to the two-tailed Fisher's exact test, $p < 0.001$).[1] The addi-

[1] To minimise damage to the robots caused by toppling over, we used just a single s-bot for the trials of the *independent execution only* strategy. We make the assumption that the task completion rate of an s-bot in a one robot trial is identical to the task completion rate of an s-bot in a two robot trial if the two s-bots are navigating independently. Note that this assumption does not take into account accidental interference

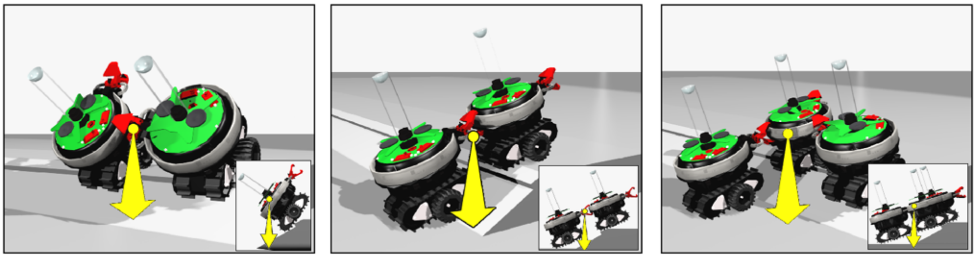
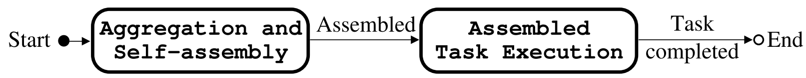


Fig. 7 Different swarm-bot configurations and orientations on the difficult hill (to scale). *Left*: a linear two s-bot swarm-bot topples on the difficult hill as its centre of gravity (see *arrow*) escapes its vertical footprint. For a video of an experiment in which 2 s-bots topple, see file 2_Sbots_Fail_To_Cross_Hill in the online supplementary material. *Centre*: a linear two s-bot swarm-bot with an appropriate orientation does not topple. *Right*: a non-linear three s-bot swarm-bot can approach with any orientation and will not topple

Fig. 8 *Preemptive self-assembly* strategy: group-level behaviour



tion of a third s-bot to the system produced even more stable connected swarm-bot entities, further increasing the task completion rate to 81.7% (a significant increase according to the two-tailed Fisher's exact test, $p < 0.001$).

The overall task completion rate was still far from optimal—35% and 81.7% for two s-bot and three s-bot experiments respectively (percentages of individual s-bots that succeed in overcoming the hill). Inappropriate orientation of the swarm-bot caused 100% and 81.8% of failures with two s-bots and three s-bots respectively. Inappropriate orientation occurs when the stochastic self-assembly process produces a swarm-bot with an approximately linear morphology, and when the resulting random orientation of such a linear swarm-bot is overly parallel to the orientation of the hill (see Fig. 7). Using the *basic self-assembly response* strategy, a swarm-bot attempts to overcome the hill with its initial random orientation. For a strategy which resolves this problem see Sect. 8.

7 Benefits of self-assembly as an autonomous response mechanism

In this section, we describe experiments we conducted with the *basic self-assembly response* strategy to analyse the performance benefits of the autonomous self-assembly response mechanism, that is, the benefits of letting the system choose when and if to self-assemble. To provide a basis for comparison, we also conducted control experiments with a simple strategy that causes the s-bots to aggregate and self-assemble irrespective of their environment.

between independently navigating s-bots—it is for example possible (although unlikely) that one s-bot could prevent another from toppling over by coincidentally giving it a push at the right moment.

7.1 *Preemptive self-assembly* strategy

The strategy is illustrated in Fig. 8. The distributed control to implement this strategy for the hill crossing task is a modified version of the distributed control for the *basic self-assembly response* strategy in which the starting state is `Aggregate` instead of `Independent_Phototaxis` (see Fig. 4). The result is that the s-bots aggregate and self-assemble irrespective of the environment. The connected swarm-bot entity then performs connected phototaxis to the light source in the target area.

7.2 Results: validation of the response mechanism

We conducted a number of trials using the *basic self-assembly response* strategy. In the difficult hill environments, we conducted 60 trials with two s-bots and 60 trials with three s-bots (these experiments have already been discussed in Sect. 6.2). In every trial, all of the s-bots successfully detected the presence of the slope, thus triggering the self-assembly process.

In the moderate hill environments, we again conducted 60 trials with two s-bots (20 trials per hill orientation) and 60 trials with three s-bots (20 trials per hill orientation). Finally, in the no-hill environment, we conducted 20 trials with two s-bots. In the $60 + 60 + 20 = 140$ trials in the moderate hill and no-hill environments, all of the s-bots always correctly 'chose' not to self-assemble and to navigate to the target area individually. Thus, in all 260 trials, the group-level response mechanism correctly classified the environment and provoked the appropriate system response (self-assembly or no self-assembly).

7.3 Results: benefits of responsiveness

We conducted 20 trials with two s-bots using the *preemptive self-assembly* strategy in the no-hill environment. In

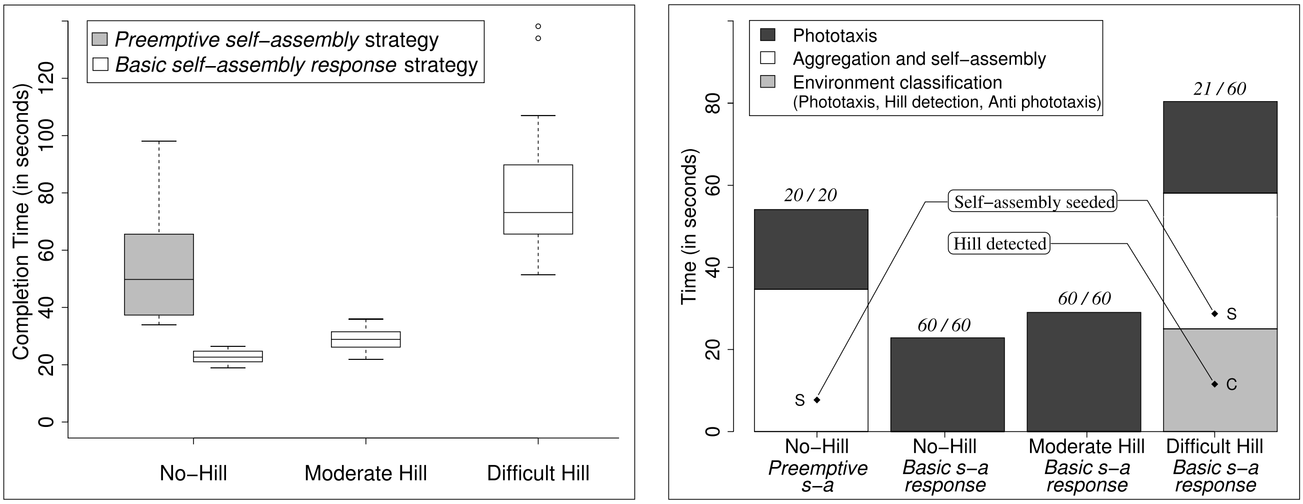


Fig. 9 *Left*: Box-and-whisker plot of completion times for two s-bots using the *preemptive self-assembly* strategy (*grey box*) and the *basic self-assembly response* strategy (*white boxes*). Each box comprises observations ranging from the first to the third quartile. The median is indicated by a horizontal bar, dividing the box into the upper and lower part. The whiskers extend to the farthest data points that are within 1.5 times the interquartile range. Outlier data points are represented by circles. *Right*: Break-down of mean completion times for two s-bots using the *preemptive self-assembly* strategy (*left bar*) and the *basic self-assembly response* strategy (*right three bars*) in no-hill, moderate hill and difficult hill environments. Only data from completed trials are presented (number of completed trials and number of trials in total are indicated above each bar)

all 20 trials, both s-bots successfully completed the task. This provides a baseline against which we compare the *basic self-assembly response* strategy. Throughout this section, we assume that the mean completion time of the *preemptive self-assembly* strategy in the no-hill environment is a lower bound for the mean completion time of the same strategy in moderate hill or difficult hill environments.

Figure 9(left) shows a box-and-whisker plot of completion times for the *preemptive self-assembly* strategy in the no-hill environment and the *basic self-assembly response* strategy in the no-hill, moderate hill and difficult hill environments (successful trials only). Figure 9(right) shows the mean completion times from the same experiments broken down into the different phases of task completion.

In the no-hill environment, the *basic self-assembly response* strategy performed significantly better than the *preemptive self-assembly* strategy (two-tailed Mann-Whitney, $p < 0.001$). The mean task completion times for the two s-bots were respectively 22.9 s and 54.1 s. The *basic self-assembly response* strategy took on average 57.7% less time to complete the task than the *preemptive self-assembly* strategy. Looking at the break-down of the mean completion times in Fig. 9(right), we can see that s-bots using the *preemptive self-assembly* strategy spent over half of their time on actions that were not necessary to complete the task (i.e., aggregation and self-assembly).

In the moderate hill environments, the mean completion time for the two s-bots using the *basic self-assembly response* strategy was 29.0 s, which is 27.1% more than the mean completion time for the same strategy in the no-hill environment. This increase is due to the extra overhead of environment classification, which takes place during phototaxis. S-bots using the *basic self-assembly response* strategy slow down on the slope to test its navigability. Nevertheless, even using the lower bound mean completion time for the *preemptive self-assembly* strategy, the *basic self-assembly response* strategy still significantly outperforms the *preemptive self-assembly* strategy (two-tailed Mann-Whitney, $p < 0.001$). In the moderate hill environments, the mean completion time for the *basic self-assembly response* strategy was 46.4% less than the lower bound mean completion time for the *preemptive self-assembly* strategy.

In the difficult hill environments, the mean completion time for the two s-bots using the *basic self-assembly response* strategy was 80.4 s. In environments where self-assembly is necessary (difficult hill environments), it is intuitively clear that the *preemptive self-assembly* strategy is more efficient than the *basic self-assembly response* strategy. S-bots using the *basic self-assembly response* strategy have an extra overhead of environment classification consisting of initial independent phototaxis, hill detection and anti-phototaxis—see Fig. 9(right).

The relative frequency with which the s-bots encounter the different environments determines which of the two strategies is more efficient. If we consider a distribution of environments containing only no-hill and difficult hill environments, then we can use the mean completion times to calculate the upper bound ratio of no-hill environments en-

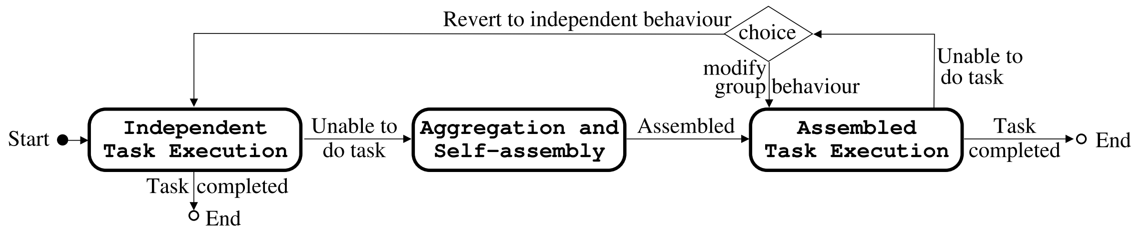


Fig. 10 *Connected coordination* strategy: group-level behaviour. In this study, the choice between modifying group behaviour and reverting to independent execution is task specific and is therefore decided in advance when implementing the strategy for a particular task

countered, α, for which the efficiency of the two strategies is identical:[2]

$$22.9\alpha + 80.4(1-\alpha) = 54.1 \quad \Rightarrow \quad \alpha \approx 0.457$$

We conclude that for two s-bots executing the hill crossing task, the *basic self-assembly response* strategy is more efficient than the *preemptive self-assembly* strategy if more than 45.7% (see footnote 2) of encountered environments are no-hill environments. If, instead, we consider a distribution of environments containing only moderate hill environments and difficult hill environments, a similar analysis reveals that the *basic self-assembly response* strategy will be more efficient if at least 51.2% of the environments are moderate hill environments.

In the future, it will be essential for the designers of distributed robotic systems to consider the costs as well as the benefits of autonomous self-assembly. Calculating α for the hill crossing task tells us when the autonomous self-assembly response mechanism actually reduces efficiency. Of course, future tasks will be more complex, and future versions of the α metric will need to be correspondingly more sophisticated. However, the calculation of such metrics might allow future system designers to determine whether an autonomously self-assembling system is an appropriate choice for the task they need to solve.

8 The *connected coordination* strategy

In this section, we present the more sophisticated *connected coordination* strategy that allows the assembled robots to coordinate their sensing and actuation so that they respond to their environment as a single collective robotic entity.[3] The *connected coordination* strategy is shown in Fig. 10. As before, the robots self-assemble (if necessary) in response to environmental contingencies. However, the assembled robotic entity is sensitive to its collective success or failure. Depending on the task, if a potential group failure is detected, the assembled entity can either modify its group level behaviour or its constituent robots can revert to independent task execution.

8.1 Strategy implementation for the hill crossing task

Our implementation allows the self-assembled swarm-bot to detect when it is inappropriately oriented, and then to retreat and rotate itself to try and achieve a more appropriate orientation. The s-bots self-assemble, as before, if they encounter a difficult hill. When the self-assembled swarm-bot encounters the hill, it analyses the orientation of the hill and compares it to its own orientation. If its own orientation is insufficiently perpendicular to the orientation of the hill, the swarm-bot retreats to flat ground by performing anti-phototaxis. It then rotates until its orientation is appropriate with respect to the remembered orientation of the hill. The swarm-bot then performs phototaxis again, and upon encountering the hill again compares its orientation to that of the hill. If its orientation is appropriate, it continues to perform phototaxis and thus navigates over the hill. Otherwise, the cycle of retreating and rotating repeats until the swarm-bot has an appropriate orientation. Note that the 'choice' in this implementation of the *connected coordination* strategy is to modify the group behaviour rather than reverting to individual behaviour. (For an example of an implementation of this strategy for another task, where the choice to revert to individual behaviour is more appropriate, see Sect. 10.2.)

The distributed control uses a leader-follower architecture (see Fig. 11). The s-bot that seeds the self-assembly process becomes the *lead s-bot* and is responsible for determining whether or not the swarm-bot is appropriately rotated. Using its LEDs, the lead s-bot issues instructions to advance, retreat or rotate to all other s-bots (*follower s-bots*) in the swarm-bot. Follower s-bots illuminate their own LEDs to mimic the LEDs of the s-bot they are gripping

[2] Upper bound ratio, as it is calculated using a lower bound for the mean completion time of the *preemptive self-assembly* strategy.

[3] The *basic self-assembly response* strategy did require some dedicated control to allow the individual s-bots to move while connected (see Sect. 5.1). However, the group level behaviour was simply what emerged from the combination of the 'greedy' behaviours of the constituent s-bots.

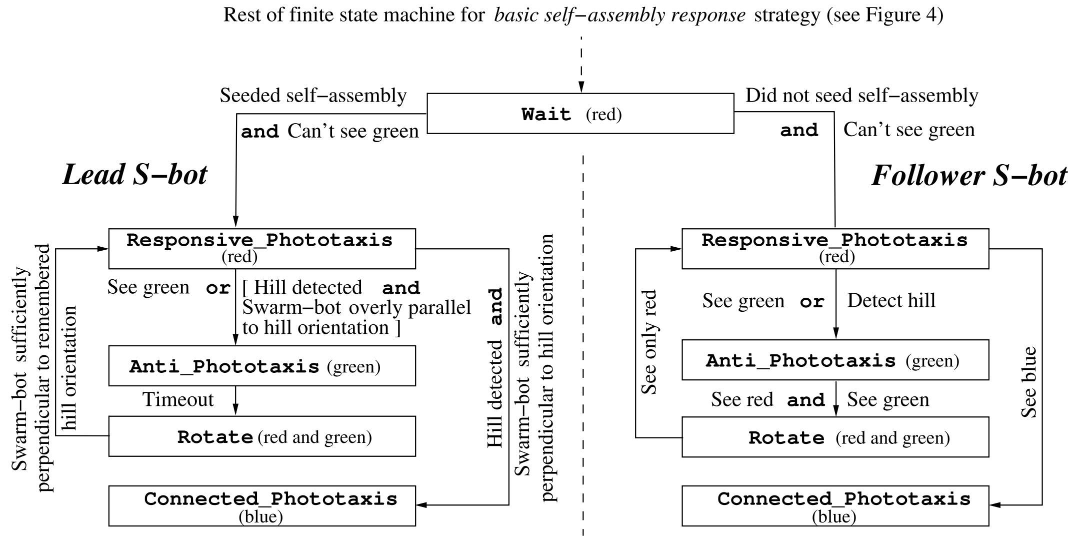


Fig. 11 Distributed control to implement the *connected coordination* strategy for the hill crossing task. This distributed control consists of finite state machine extensions to the distributed control for the *basic self-assembly response* strategy. The s-bot first executes the distributed control for the *basic self-assembly response* strategy (see Fig. 4). However, instead of executing the `Connected_Phototaxis` state the s-bot switches into state `Responsive_Phototaxis`, either as the lead s-bot, if it seeded self-assembly, or, otherwise, as a follower s-bot

Fig. 12 Execution of the controller for the *connected coordination* strategy. (**A**): The s-bots have already aggregated and self-assembled as a response to having encountered the hill. The connected swarm-bot performs phototaxis and approaches the hill with an inappropriate (random) orientation. Having detected the hill, the swarm-bot performs antiphototaxis, with the result that it moves away from the hill. (**B**): The swarm-bot rotates until it has a more appropriate orientation (based on memory of the hill orientation). (**C**): The swarm-bot approaches the hill with its new orientation. (**D**): The swarm-bot recognises that it has an appropriate orientation and attempts to overcome the hill. (**E**): The swarm-bot arrives in the target area. For a video of this experiment, see file 2_Coordinated_Sbots_Cross_Hill in the online supplementary material

(which is guaranteed to be closer to the lead s-bot in a linear morphology). In this way, instructions propagate along the swarm-bot from the lead s-bot to all the follower s-bots. The only exception is if a follower s-bot detects the hill before the lead s-bot, in which case the instruction to retreat propagates in the opposite direction.

The distributed control for the *connected coordination* strategy is an extension of the distributed control for the *basic self-assembly response* strategy. Control is branched into one of two possible finite state machine extensions (one for the lead s-bot, one for the follower s-bots) after the system has self-assembled. The two finite state machine extensions are shown in Fig. 11. All of the s-bots start by executing the controller for the *basic self-assembly response* strategy, as illustrated in Fig. 4. However, once they are in state `Wait`, instead of switching to state `Connected_Phototaxis`, they trigger the relevant *connected coordination* strategy finite state machine extension. In other words, the controller for the *connected coordination* strategy is constructed by substituting the state `Connected_Phototaxis` in Fig. 4 with the state `Responsive_Phototaxis` in Fig. 11 from either the lead s-bot finite state machine, if the s-bot seeded self-assembly, or, otherwise, the follower s-bot finite state machine.

Figure 12 illustrates a two s-bot system executing the controller for the *connected coordination* strategy. In state `Responsive_Phototaxis`, the s-bots perform collective phototaxis while constantly checking the orientation of any hill they encounter with respect to the orientation of the swarm-bot (each s-bot uses its 3D accelerometers to determine the orientation of the hill and its camera to determine the orientation of the swarm-bot). If a hill is encountered and the orientation of the swarm-bot is appropriate with respect to the orientation of the hill (perpendicular with a tolerance

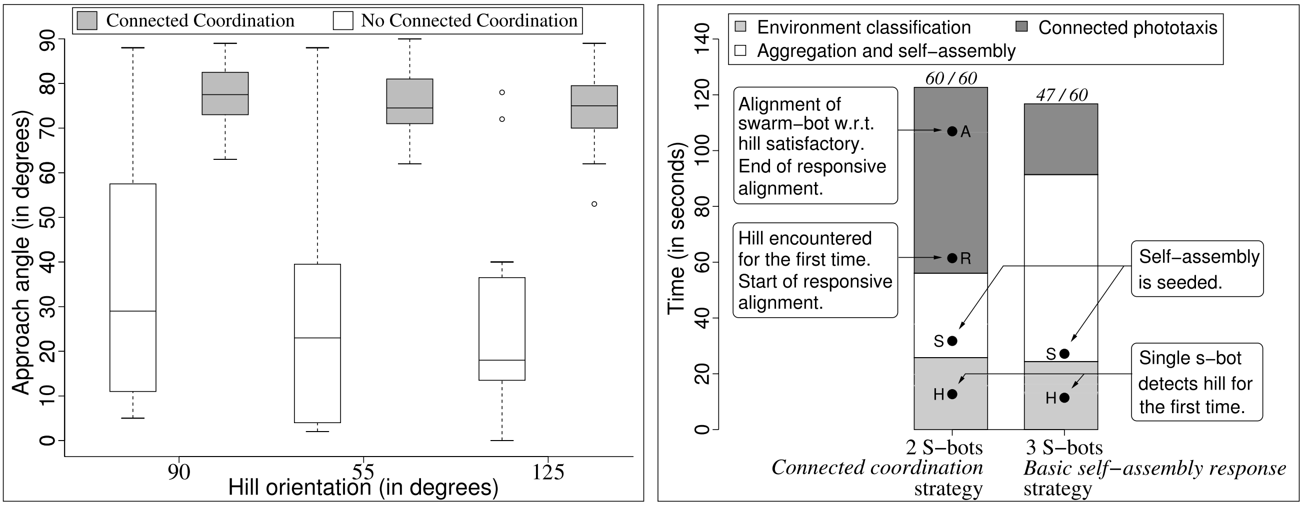


Fig. 13 *Left*: Box-and-whisker plot showing acute approach angles of two s-bot swarm-bots (orientation of the swarm-bot with respect to the orientation of the hill). The orientation of the swarm-bot is measured at the moment that the swarm-bot first makes contact with the hill on its final approach (if the swarm-bot approached the hill more than once, the earlier approaches are ignored). Each box represents 20 trials. *White boxes*: swarm-bots without connected coordination. *Grey boxes*: swarm-bots with connected coordination. *Right*: Break-down of mean completion times for 2 s-bots using the *connected coordination* strategy (*left bar*) and for 3 s-bots using the *basic self-assembly response* strategy (*right bar*) in the difficult hill environments. Only data from trials that were completed by all s-bots are presented (number of completed trials and number of trials in total are indicated above each bar)

of 20°), the s-bots continue performing phototaxis to the light source, but no longer check the orientation of any encountered hills (state `Connected_Phototaxis`). If the orientation of the swarm-bot is not appropriate, the s-bots remember the orientation of the hill and retreat away from the hill for a given length of time (state `Anti_Phototaxis`). They rotate until the orientation of the swarm-bot is appropriate with respect to the remembered hill orientation (state `Rotate`), and then start performing collective phototaxis again (state `Responsive_Phototaxis`). For a detailed description of the individual states in this finite state machine see O'Grady et al. (2009b, 2010).

9 Benefits of connected coordination

In this section, we present experiments that we conducted with the *connected coordination* strategy to analyse the benefits of coordinating the sensing and actuation of the assembled robots to allow them to act as a single collective entity. We restrict our experimentation to the two s-bot case, as a two s-bot swarm-bot is linear by definition. We thus avoid the additional challenge of either checking whether the swarm-bot that has formed is linear or of controlling the self-assembly process to explicitly generate a linear morphology.[4]

Table 1 Experimental results for environments with the difficult hill. All strategies were evaluated in 60 independent trials. The last two columns show the percentage of s-bots that correctly assembled and the percentage of s-bots that completed the task

# S-bots	Strategy	Assembled	Completed
1	*Independent execution only*	–	0.0%
2	*Basic self-assembly response*	100.0%	35.0%
3	*Basic self-assembly response*	98.3%	81.7%
2	***Connected coordination***	**100.0%**	**100.0%**

9.1 Results

Table 1 shows how the *connected coordination* strategy compares against the other strategies in the difficult hill environments. With the *connected coordination* strategy, over 60 trials, two s-bots achieved the optimal task completion rate of 100% (see last row of the table). This increase in task completion rate can be attributed to connected coordination, that ensures the swarm-bot is appropriately rotated with respect to the hill. The effectiveness of the rotation mechanism can been seen in Fig. 13(left). Swarm-bots with connected coordination orient themselves against each of the three hill orientations significantly better than swarm-bots

[4] To let more than two s-bots self-assemble into a linear formation, a more elaborate control of the self-assembly process would be required. This is a subject of ongoing research (Christensen et al. 2008; O'Grady et al. 2009a).

without connected coordination (two-tailed Mann-Whitney, $p < 0.001$).

We have seen in Sect. 6.2 that a two s-bot swarm-bot relying only on benefits of scale has a success rate of 35%. To improve the success rate, we investigated two options.

- The first option was to further rely on the simple benefits of scale by increasing the number of s-bots to three whilst still using the *basic self-assembly response* strategy. This resulted in a task completion rate of 81.7%. However, the self-assembly and aggregation process becomes more complex with three s-bots and therefore takes longer and is more prone to failure (see also Table 1).
- The second option was to use a different strategy altogether. Using the *connected coordination* strategy, the assembled s-bots coordinated their sensing and actuation to leverage their collective morphology. This strategy achieved an optimal task completion rate. However, the responsive rotation mechanism is an extra time overhead and on average took up 37.1% of the total task completion time.

A breakdown of the mean completion times for these two options in the difficult hill environments is given in Fig. 13(right). Overall, there is no significant difference in the mean completion times of the *basic self-assembly response* strategy with three robots and the *connected coordination* strategy with two robots (two-tailed Mann-Whitney, $p < 0.001$).

10 Discussion

In this section, we discuss the relevance of this study in a wider context. We first consider the scalability potential of our system. We then discuss the applicability of the strategies we propose to other problem domains. Finally, we discuss the future creation of a genuinely adaptive self-assembling system. Where possible we illustrate our discussion with dedicated proof-of-concept experimentation.

10.1 Scalability

The control that we used to implement our strategies was strictly distributed and relied only on local communication. Thus, although the hill climbing task we presented was solved optimally with two robots executing the *connected coordination* strategy, we would nonetheless expect our distributed control to scale to larger numbers of robots. To test this hypothesis we conducted several proof-of-concept experiments repeating our detailed experimentation but with larger numbers of robots. Snapshots from some of the trials are shown in Fig. 14. The top two rows of Fig. 14 show trials using the *basic self-assembly response* strategy in the difficult hill environment with six real robots. The bottom row of the figure shows the responsive rotation mechanism of the distributed control for the *connected coordination* controller working with three (manually pre-assembled) s-bots.

The middle row of Fig. 14 is interesting, as it shows a rudimentary form of group size regulation that we had not

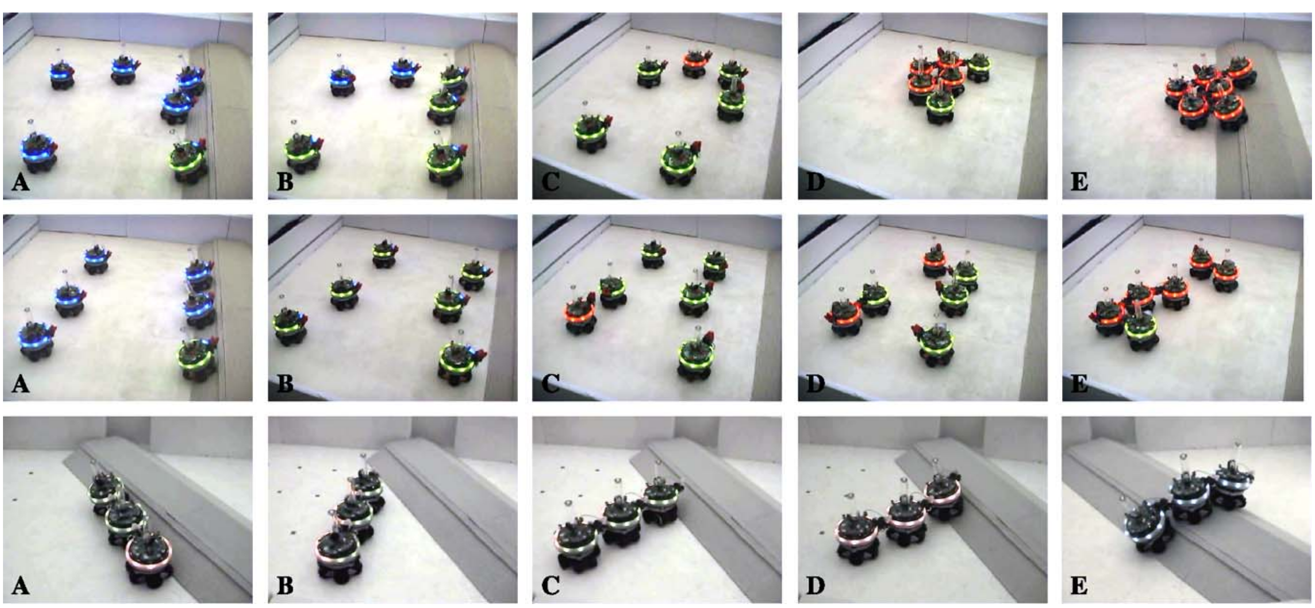


Fig. 14 Scalability experiments in the difficult hill environment. *Top*: 6 s-bots using the *basic self-assembly response* strategy self-assemble to cross the hill. *Middle*: 6 s-bots using the *basic self-assembly response* strategy self-assemble into two groups, both of which successfully cross the hill. *Bottom*: Three manually pre-assembled s-bots using the *connected coordination* strategy rotate to cross the hill. For videos of these experiments, see files 6_Sbots_Cross_Hill, 6_Sbots_Cross_Hill_2_Groups, 3_Coordinated_Sbots_Cross_Hill in the online supplementary material

explicitly encoded into the system, but that emerged as a result of the distributed, stochastic nature of the control. In this trial, the robots aggregated into two separately seeded groups of four and two s-bots respectively. This double seeding was possible because in this particular trial the two s-bots that became seeds were out of visual range of each other. Both groups successfully self-assembled and navigated over the hill. The dynamics of this group size regulation mechanism are stochastic and depend on the density of robots and the visual range of the camera. For our system to become practically useful in the future, more sophisticated group size regulation will be necessary. It is neither practical nor efficient for the size of the compound robotic entity to grow indefinitely. In our experiments with 6 real robots, we already noticed that interference effects were common—the larger numbers of constituent robots meant that it was more probable for the tracks of one robot to impede the tracks of another robot during collective motion. In addition, the characteristics of the hardware platform place physical limits on the optimal (and maximum) size of compound entities (Mondada et al. 2005). An interesting avenue for future research would, therefore, be to create a more explicit group size regulation mechanism without sacrificing the principles of distributed control.

10.2 Wider applicability of strategies and analysis

In this section, we consider the application of the self-assembly strategies to other problem domains. We conducted experiments with two other tasks—hole crossing and robot rescue.

In the hole crossing task, the robots are required to cross a hole of a priori unknown width as they navigate to a light source. We parametrise the task with two possible hole widths—a 3 cm hole that a single s-bot can cross, and a 10 cm hole that a single s-bot would topple into (Mondada et al. 2005). The robots detect holes with their infrared ground sensors. To implement the *basic self-assembly response* strategy for hole crossing, we only needed to make a single minor modification to the distributed control for the hill crossing task. We replaced the ‘too steep’ condition with a ‘wide hole’ condition that is triggered when more than one ground sensor fires at the same time (see Fig. 4). Results of a single trial for each width of hole are shown in Fig. 15. The success of the hole crossing experiment shows that the strategies we developed are reusable in other problem domains. Furthermore, a similar analysis that we have conducted for the hill crossing experiment could be conducted for the hole crossing experiment, including the calculation of a dedicated α metric for this task determining when *preemptive self-assembly* would be more appropriate.

The second task we considered was robot rescue, in which a group of rescuing s-bots are required to transport one or more broken robots to a repair zone (see Fig. 16). The task is parameterised by the number and size of broken robots. Single broken s-bots can be transported by a single rescuing s-bot, while to transport larger composite broken swarm-bots the rescuing s-bots must self-assemble and physically co-operate.

We used the *connected coordination* strategy. In our implementation of the strategy for this task, the success/failure detection mechanism is based on the s-bot track motor torque sensors. Each rescue s-bot tries to move any broken robots that it finds, and uses its torque sensors to determine whether or not it is moving the object—if the object is moving, the s-bot’s tracks will be rotating and the motor torque will be low, while if the object is not moving its tracks will be blocked and the torque will be high. Based on this determination, the rescuing s-bot uses local communication to either attract (red LEDs illuminated) or repel (green LEDs illuminated) other rescue s-bots. Thus, if a rescue s-bot attaches to a broken robot and is unable to move it, it will illuminate its red LEDs to attract more rescue s-bots. As long as the broken robot remains immobile, all of the attached rescue s-bots will illuminate their red LEDs, because they will each independently determine that the transport is unsuccessful. As soon as enough rescue s-bots are attached to start transporting the object, each of the attached rescue s-bots independently registers that the object is successfully being moved, and all of them start repelling other rescue s-bots by illuminating their green LEDs. The *connected coordination* strategy contains a choice—in the face of group failure, the system can either modify the group behaviour or revert to individual task execution (see

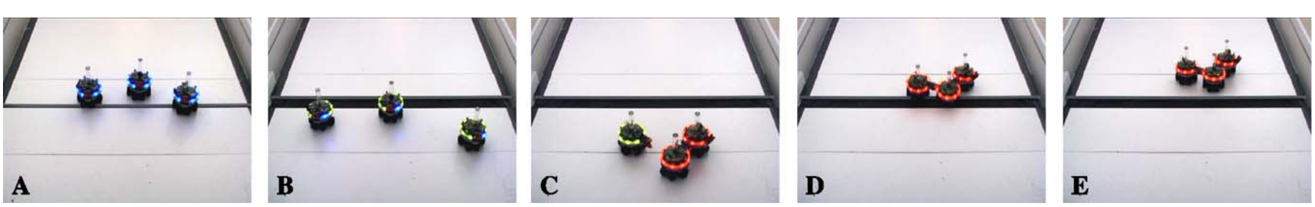


Fig. 15 The *basic self-assembly response* strategy in the hole crossing task. (**A**): s-bots cross a 3 cm hole individually. (**B**–**E**): s-bots respond to a 10 cm hole by self-assembling and crossing collectively. For videos of these experiments, see files 3_Sbots_Cross_2cm_Hole, 3_Sbots_Cross_10cm_Hole in the online supplementary material

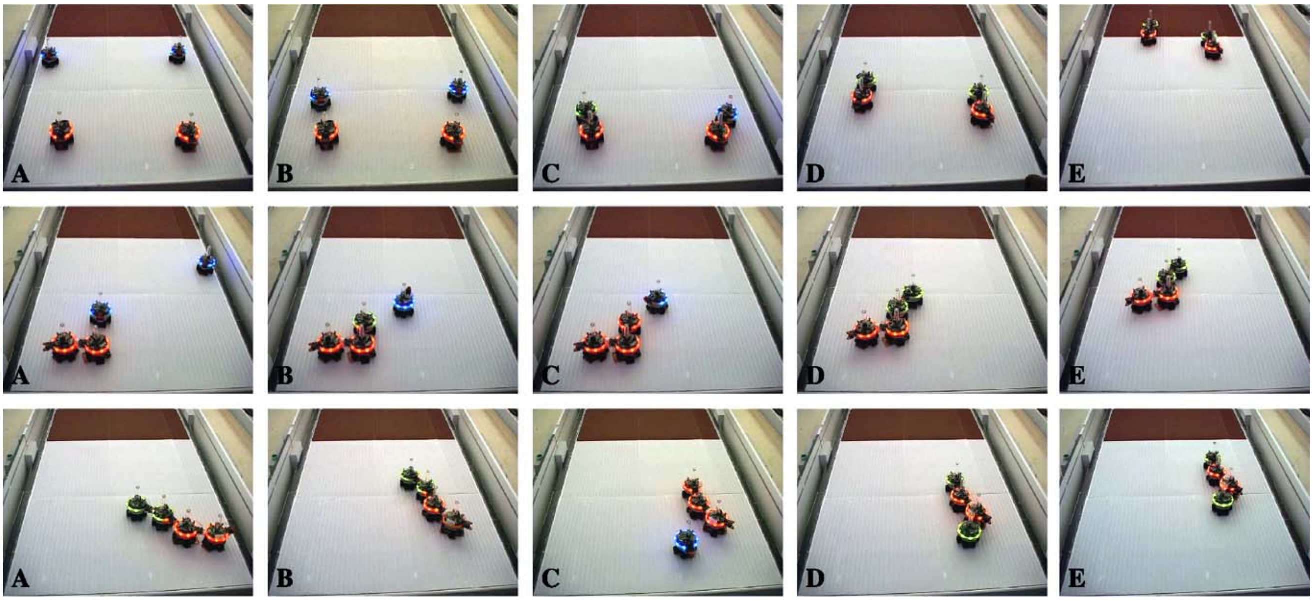


Fig. 16 The *connected coordination* strategy in the robot rescue task. *Top*: 2 rescue s-bots transport two broken s-bots in parallel to the repair zone. *Middle*: 2 rescue s-bots respond to finding a broken swarm-bot that is too heavy for a single s-bot to move by self-assembling and collectively transporting the swarm-bot to the repair zone. *Bottom*: 2 rescue s-bots transport a broken swarm-bot made up of two connected s-bots. The two rescue robots attach to the broken swarm-bot (**A**). The rescue s-bots get stuck (**B**), but stochastically reconfigure by reverting to independent task execution (**C**, **D**) until the system finds a configuration in which they succeed in transporting the swarm-bot (**E**). For videos of these experiments, see files 2_Sbots_Rescue_2_Sbots, 2_Sbots_Rescue_Swarmbot, 2_Reconfiguring_Sbots_Rescue_Swarmbot in the online supplementary material

Fig. 10). In this case, when the system is not able to recruit enough rescue s-bots to move a particular object (group failure), the system reverts to independent action (individual attached s-bots probabilistically detach and revert to individual action). Thus rescue s-bots that might potentially become deadlocked while attached to a broken robot that is not transportable are freed either to reconfigure into a more successful configuration, or to transport other broken robots.

The top row of Fig. 16 shows two rescue s-bots transporting two broken s-bots in parallel. The first rescue s-bot to attach to one of the broken robots succeeds in transporting it alone, and thus illuminates its green LEDs to repel other rescue s-bots. The repelling green LEDs ensure that the second rescue s-bot attaches to the other broken robot instead of unnecessarily attaching to the broken robot that is already successfully being rescued. The middle row of Fig. 16 shows the attraction dynamic—the first rescue s-bot fails to transport a heavier broken swarm-bot made up of two connected s-bots and illuminates its red LEDs to attract help. The second rescue s-bot attaches and together they successfully transport the broken swarm-bot (while illuminating their green LEDs). The bottom row of the figure shows a trial in which the initial assembled configuration fails to transport the broken swarm-bot all of the way to the repair zone. The connected entity detects its inability to move the object (all rescue s-bots keep their red LEDs illuminated), and one of the constituent s-bots reverts probabilistically to independent task execution. The result is that the system stochastically reconfigures until it finds an assembled morphology that succeeds in transporting the broken swarm-bot. For more details of these robot rescue experiments see O'Grady et al. (2009c).

The trials in Fig. 16 demonstrate that the system allocates the appropriate number of rescue s-bots to each broken swarm-bot. An interesting research direction would be to try and add this type of distributed resource allocation mechanism to other types of task, in particular the hill crossing and hole crossing tasks presented in this study. If we wanted to replicate the dynamics of the rescue control strategy, one of the key challenges would be to enable the robots to determine success or failure without causing catastrophic failure (e.g., without toppling on a hill or falling into a hole), and without compromising the distributed nature of the control.

10.3 Towards genuine adaptivity

The long term vision that we outlined in the introduction described future self-assembling systems that would respond to their task and environment by allocating resources and creating appropriate robotic entities when needed. In this subsection, we discuss how future work might build on the work presented in this paper and start achieving this kind of adaptivity. We consider adaptivity to be firstly the ability to detect the salient features of a task and environment and secondly the ability to respond to these features appropriately.

In this study, the detection of environmental features performed by our system was restricted by the limits of the s-bot's sensing apparatus—our system could only detect simple environmental cues such as presence of a hill, steepness of a hill, orientation of a hill. For future systems to overcome such limitations, one possibility would be to give each agent in the system better sensing hardware. However, the simplicity of individual agents is often cited as a key strength of distributed self-assembling systems. An alternative might be to introduce heterogeneity into the system, and have some robots specialised in sensing that could then communicate relevant information to the other robots. This is the approach being taken by the Swarmanoid project (the successor to the swarm-bots project), in which flying robots with a wider vision can direct the wheeled robots on the ground (Dorigo 2009). Another interesting research direction might be to leverage group sensing capabilities, whereby the sensory data of individual distributed agents are somehow combined to build up a more sophisticated composite picture of the environment and of the agents' own spatial configuration (Funiak et al. 2009).

In this study, we have not leveraged one important ability intrinsic to self-assembling systems: the ability to form different morphologies by controlling the self-assembly process. This ability gives such systems another potential way in which to vary their response to their environment. In the robot rescue experiment presented in Sect. 10.2, our system did display a stochastic form of morphological variation—by reverting to independent behaviour when the group failed to transport a broken robot, the system could try different configurations until it found one that worked (see Fig. 16 bottom). This type of morphology control is a topic of research that we are also investigating. At present, we are limited to forming pre-determined rigid morphologies (Christensen et al. 2008; O'Grady et al. 2009a). More elaborate morphology control mechanisms (Hirose 1993; Ishiguro et al. 2004; Brown et al. 2002) and a better understanding of the dependencies between form and function (Yim 1994; Yim et al. 2001; Mondada et al. 2005; Campbell and Pillai 2008) could yield self-assembling systems that are much better adapted to their environment and/or user demands. Such systems would also be of general interest, for example, in the creation of "synthetic reality" (Goldstein et al. 2005).

Factors limiting the range of appropriate responses include hardware limitations, the ingenuity of the human system designers, and the ability of the system to learn appropriate responses from trial and error. Learning in particular is an important avenue for future research in self-assembling systems. Learning, whether off-line or on-line, depends on the ability to detect success and failure. Although, in this study, we have not considered learning, some of the success/failure detection mechanism we designed, especially those at the group level, could be used as a starting point for future learning based systems. For example, in the robot rescue experiment, the robots can already stochastically try different patterns until they find one that succeeds. The robots use their torque sensors to implement a distributed success/fail criterion that applies to the whole group with minimal communication. The next step might be to make this stochastic process more explicit—to find a way for the system to remember configurations that have succeeded and to try and recreate them in the future.

11 Conclusion

Groups of autonomous robots can use self-assembly to work together and overcome the physical limitations of individual robots. In this study, we presented different self-assembly strategies, and showed how they could be applied to three different tasks—hill crossing, hole crossing and robot rescue. For each task, we implemented the strategies using distributed control that was homogeneous and that used only local (visual) communication. In each case, we verified the distributed control with real-world experimentation.

For the hill crossing task, we performed a quantitative analysis of our experiments. Our analysis showed that simply by carrying out the task as a physically larger self-assembled entity the system could leverage benefits of scale to improve the task completion rate from 0% to 35% (when two robots self-assembled) or to 81.7% (when three robots self-assembled). We went on to show that significant benefits in system efficiency could be derived by making self-assembly responsive—allowing the robots to choose when and if to self-assemble based on the nature of the environments they encounter. Finally, we demonstrated that significant improvements in the task completion rate (up to 100%) could be achieved if the assembled robots coordinated their sensing and actuation so that they could respond to environmental contingencies as a single collective entity.

Acknowledgements This work would not have been possible without the innovative robotic hardware developed by Francesco Mondada's group at the Laboratoire de Systèmes Robotiques of the École Polytechnique Fédérale de Lausanne. This work was supported by the Sixth Framework Programme of the European Commission in the form of the IST FET project SWARMANOID (contract IST-022888), the COMP2SYS project, a Marie Curie Early Stage Research Training Site (contract MEST-CT-2004-505079), and a Marie Curie Intra-European Fellowship (contract MEIF-CT-2006-040312), and by the Scientific Research Directorate of the French Community of Belgium in the form of the "ANTS" project, an "Action de Recherche Concertée". The work of M. Dorigo was additionally supported by the Belgian F.R.S.–FNRS, of which he is a Research Director. The information provided is the sole responsibility of the authors and does not reflect the European Commission's opinion. The European Commission is not responsible for any use that might be made of data appearing in this publication.

References

Anderson, C., Theraulaz, G., & Deneubourg, J.-L. (2002). Self-assemblages in insect societies. *Insectes Sociaux*, *49*(2), 99–110.

Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm intelligence: from natural to artificial systems*. New York: Oxford University Press.

Brown Jr., H. B., Weghe, J. M. V., Weghe, E., Bererton, C. A., & Khosla, P. K. (2002). Millibot trains for enhanced mobility. *IEEE/ASME Transactions on Mechatronics*, *7*(4), 452–461.

Campbell, J., & Pillai, P. (2008). Collective actuation. *International Journal of Robotics Research*, *27*(3–4), 299–314.

Cao, Y. U., Fukunaga, A. S., & Kahng, A. B. (1997). Cooperative mobile robotics: antecedents and directions. *Autonomous Robots*, *4*(1), 7–27.

Caspar, D. L. D. (1966). Design principles in organized biological structures. In Wolstenholme, G. E. W., & O'Connor, M. (Eds.), *Principles of biomolecular organization* (pp. 7–34). London: Churchill.

Castano, A., Behar, A., & Will, P. M. (2002). The Conro modules for reconfigurable robots. *IEEE/ASME Transactions on Mechatronics*, *7*(4), 403–409.

Christensen, A. L., O'Grady, R., & Dorigo, M. (2008). SWARMORPH-script: a language for arbitrary morphology generation in self-assembling robots. *Swarm Intelligence*, *2*(2–4), 143–165.

Dorigo, M. (2009). The swarmanoid project. http://www.swarmanoid.org.

Fukuda, T., & Nakagawa, S. (1988). Approach to the dynamically reconfigurable robotic system. *Journal of Intelligent and Robotic Systems*, *1*(1), 55–72.

Funiak, S., Pillai, P., Ashley-Rollman, M. P., Campbell, J. D., & Goldstein, S. C. (2009). Distributed localization of modular robot ensembles. *International Journal of Robotics Research*, *28*(8), 946–961.

Goldstein, S. C., Campbell, J. D., & Mowry, T. C. (2005). Programmable matter. *Computer*, *38*(6), 99–101.

Groß, R., & Dorigo, M. (2004). Group transport of an object to a target that only some group members may sense. In *Lecture notes in computer science: Vol. 3242. Parallel problem solving from nature—8th international conference (PPSN VIII)* (pp. 852–861). Berlin: Springer.

Groß, R., & Dorigo, M. (2008a). Evolution of solitary and group transport behaviors for autonomous robots capable of self-assembling. *Adaptive Behavior*, *16*(5), 285–305.

Groß, R., & Dorigo, M. (2008b). Self-assembly at the macroscopic scale. *Proceedings of the IEEE*, *96*(9), 1490–1508.

Groß, R., Bonani, M., Mondada, F., & Dorigo, M. (2006a). Autonomous self-assembly in swarm-bots. *IEEE Transactions on Robotics*, *22*(6), 1115–1130.

Groß, R., Tuci, E., Dorigo, M., Bonani, M., & Mondada, F. (2006b). Object transport by modular robots that self-assemble. In *Proceedings of the 2006 IEEE international conference on robotics and automation* (pp. 2558–2564). Los Alamitos: IEEE Computer Society Press.

Hirose, S. (1993). *Biologically inspired robots: snake-like locomotors and manipulators*. New York: Oxford University Press.

Hirose, S., Shirasu, T., & Fukushima, E. F. (1996). Proposal for cooperative robot "Gunryu" composed of autonomous segments. *Robots and Autonomous Systems*, *17*(1–2), 107–118.

Ishiguro, A., Shimizu, M., & Kawakatsu, T. (2004). Don't try to control everything!: an emergent morphology control of a modular robot. In *Proceedings of the 2004 IEEE/RSJ international conference on intelligent robots and systems* (Vol. 1, pp. 981–985). Los Alamitos: IEEE Computer Society Press.

Kamimura, A., Kurokawa, H., Yoshida, E., Murata, S., Tomita, K., & Kokaji, S. (2005). Automatic locomotion design and experiments for a modular robotic system. *IEEE/ASME Transactions on Mechatronics*, *10*(3), 314–325.

Mondada, F., Pettinaro, G. C., Guignard, A., Kwee, I. V., Floreano, D., Deneubourg, J.-L., Nolfi, S., Gambardella, L. M., & Dorigo, M. (2004). SWARM-BOT: a new distributed robotic concept. *Autonomous Robots*, *17*(2–3), 193–221.

Mondada, F., Bonani, M., Guignard, A., Magnenat, S., Studer, C., & Floreano, D. (2005). Superlinear physical performances in a SWARM-BOT. In *Lecture notes in artificial intelligence: Vol. 3630. 8th European conference on artificial life, ECAL 2005* (pp. 282–291). Berlin: Springer.

Mumm, E., Farritor, S., Pirjanian, P., Leger, C., & Schenker, P. (2004). Planetary cliff descent using cooperative robots. *Autonomous Robots*, *16*(3), 259–272.

Murata, S., Yoshida, E., Kamimura, A., Kurokawa, H., Tomita, K., & Kokaji, S. (2002). M-TRAN: self-reconfigurable modular robotic system. *IEEE/ASME Transactions on Mechatronics*, *7*(4), 431–441.

O'Grady, R., Groß, R., Mondada, F., Bonani, M., & Dorigo, M. (2005). Self-assembly on demand in a group of physical autonomous mobile robots navigating rough terrain. In *Lecture notes in artificial intelligence: Vol. 3630. 8th European conference on artificial life, ECAL 2005* (pp. 272–281). Berlin: Springer.

O'Grady, R., Christensen, A. L., & Dorigo, M. (2009a). SWARMORPH: multi-robot morphogenesis using directional self-assembly. *IEEE Transactions on Robotics*, *25*, 738–743.

O'Grady, R., Groß, R., Christensen, A. L., & Dorigo, M. (2009b). *Distributed control to implement self-assembly strategies for the hill crossing task* (Technical Report TR/IRIDIA/2009-022). IRIDIA, Faculté des Sciences Appliquées, Université Libre de Bruxelles.

O'Grady, R., Pinciroli, C., Groß, R., Christensen, A. L., & Dorigo, M. (2009c, in press). Swarm-bots to the rescue. In *Proceedings of the 10th European conference on artificial life, ECAL 2009*. Berlin: Springer.

O'Grady, R., Groß, R., Christensen, A. L., & Dorigo, M. (2010). Self assembly strategies in a group of autonomous mobile robots—support page. http://iridia.ulb.ac.be/supp/IridiaSupp2008-016/.

Østergaard, E. H., Kassow, K., Beck, R., & Lund, H. H. (2006). Design of the ATRON lattice-based self-reconfigurable robot. *Autonomous Robots*, *21*(2), 165–183.

Penrose, L. S., & Penrose, R. (1957). A self-reproducing analogue. *Nature*, *179*(4571), 1183.

Sendova-Franks, A. B., & Franks, N. R. (1999). Self-assembly, self-organization and division of labour. *Philosophical Transactions Royal Society B*, *354*(1388), 1395–1405.

Shen, W.-M., Will, P., Galstyanm, A., & Chuong, C.-M. (2004). Hormone-inspired self-organization and distributed control of robotic swarms. *Autonomous Robots*, *17*(1), 93–105.

Shen, W.-M., Krivokon, M., Chiu, H., Everist, J., Rubenstein, M., & Venkatesh, J. (2006). Multimode locomotion for reconfigurable robots. *Autonomous Robots*, *20*(2), 165–177.

Trianni, V., Tuci, E., & Dorigo, M. (2004). Evolving functional self-assembling in a swarm of autonomous robots. In *Proceedings of the 8th international conference on the simulation of adaptive behavior* (pp. 405–414). Cambridge: MIT Press.

Tuci, E., Groß, R., Trianni, V., Mondada, F., Bonani, M., & Dorigo, M. (2006). Cooperation through self-assembly in multi-robot systems. *ACM Transactions on Autonomous and Adaptive Systems*, *1*(2), 115–150.

Whitesides, G. M., & Grzybowski, B. (2002). Self-assembly at all scales. *Science*, *295*(5564), 2418–2421.

Yamakita, M., Taniguchi, Y., & Shukuya, Y. (2003). Analysis of formation control of cooperative transportation of mother ship by SMC.

In *Proceedings of the 2003 IEEE international conference on robotics and automation* (Vol. 1, pp. 951–956). Los Alamitos: IEEE Computer Society Press.

Yim, M. (1994). *Locomotion with a unit-modular reconfigurable robot*. PhD thesis, Department of Mechanical Engineering, Stanford University, Stanford, CA.

Yim, M., Duff, D. G., & Roufas, K. D. (2000). PolyBot: A modular reconfigurable robot. In *Proceedings of the 2000 IEEE international conference on robotics and automation* (Vol. 1, pp. 514–520). Los Alamitos: IEEE Computer Society Press.

Yim, M., Duff, D., & Zhang, Y. (2001). Closed-chain motion with large mechanical advantage. In *Proceedings of the 2001 IEEE/RSJ international conference on intelligent robots and systems* (Vol. 1, pp. 318–323). Los Alamitos: IEEE Computer Society Press.

Yim, M., Roufas, K., Duff, D., Zhang, Y., Eldershaw, C., & Homans, S. B. (2003). Modular reconfigurable robots in space applications. *Autonomous Robots*, *14*(2–3), 225–237.

Rehan O'Grady is a researcher at the IRIDIA-CoDE laboratory of the Université Libre de Bruxelles, Belgium. He graduated in 1999 from Edinburgh University with First Class Honours in Mathematics and Computer Science. He received the Kevin Clark memorial prize, awarded each year to the top Mathematics and Computer Science graduate. He subsequently worked for Micromuse Ltd. (now part of IBM), researching and implementing new network management paradigms. His design for detecting usage outages in IP networks was awarded a US Patent. In 2005, he received the Diplôme d'Études Approfondies (master's equivalent) from the Université Libre de Bruxelles. His research focus is on swarm robotics, specifically self-assembly, morphogenesis and swarm control.

Roderich Groß received the Diploma degree (*summa cum laude*) in computer science in 2001 from Universität Dortmund, Dortmund, Germany and the doctoral degree in engineering sciences in 2007 from Université Libre de Bruxelles, Brussels, Belgium. In 2005 he held a short-term JSPS postdoctoral research fellowship at the Department of Control and System Engineering, Tokyo Institute of Technology, Tokyo, Japan. From 2006 to 2007 he was a research assistant at the School of Biological Sciences, University of Bristol, Bristol, UK. In 2007, he was a Marie Curie Fellow at Unilever R&D Port Sunlight, Bebington, UK. From 2008 to 2009 he was a Marie Curie Fellow at the Robotic Systems Laboratory (LSRO), Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland. He is currently a lecturer in the Automatic Control and Systems Engineering (ACSE) department of the University of Sheffield, Sheffield, UK. His research interests include artificial life, computational biology, robotics, and swarm intelligence. He is member of the program committee of numerous international conferences.

Anders Lyhne Christensen is currently a researcher at Instituto de Telecomunicações and an assistant professor at Lisbon University Institute, Portugal. He has spent several years in the private sector and worked on software projects ranging from 3D acoustics to high performance computing. In 2002, he obtained a Master's degree in computer science and bio-informatics from Aalborg University, Denmark. Later that year, he received a Marie Curie Fellowship hosted by the Dependable Systems Group at the Universidade de Coimbra and Critical Software, Portugal. He completed his Diplôme d'Études Approfondies (Master's equivalent) at the Université Libre de Bruxelles in 2005. In 2008, he obtained the doctoral degree from IRIDIA-CoDE, Université Libre de Bruxelles, Belgium. He has published work in bioinformatics, high performance computing, and autonomous robotics. His research interests include swarm robotics, bio-inspired robotics, learning and self-assembly.

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 from Politecnico di Milano, Milan, Italy, and the title of Agrégé de l'Enseignement Supérieur, from the Université Libre de Bruxelles, Belgium, in 1995. From 1992 to 1993 he was a research fellow at the International Computer Science Institute of 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-CoDE, 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 swarm intelligence, swarm robotics, and metaheuristics for discrete optimization. Dr. Dorigo is the Editor-in-Chief of the Swarm Intelligence journal. He is an Associate Editor for the IEEE Transactions on Evolutionary Computation, the IEEE Transactions on Systems, Man, and Cybernetics, and the ACM Transactions on Autonomous and Adaptive Systems. He is a member of the Editorial Board of numerous international journals, including: Adaptive Behavior, AI Communications, Artificial Life, Cognitive Systems Research, Evolutionary Computation, Information Sciences, Journal of Heuristics 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 2003 the Marie Curie Excellence Award, in 2005 the Dr A. De Leeuw-Damry-Bourlart award in Applied Sciences, and in 2007 the Cajastur International Prize for Soft Computing.