Performance Benefits of Self-Assembly in a *Swarm-Bot*

Rehan O'Grady[1], Roderich Groß[1,2], Anders Lyhne Christensen[1],
Francesco Mondada[3], Michael Bonani[3] and Marco Dorigo[1]

[1] IRIDIA-CoDE, Université Libre de Bruxelles, Brussels, Belgium
[2] Unilever R&D Port Sunlight, Bebington, United Kingdom
[3] LSRO, Ecole Polytechnique Fédérale de Lausanne, Switzerland
Email: {rogrady,rgross,alyhne,mdorigo}@ulb.ac.be, {francesco.mondada,michael.bonani}@epfl.ch

***Abstract*— Mobile robots are said to be capable of self-assembly when they can autonomously form physical connections with each other. Despite the recent proliferation of self-assembling systems, little work has been done on using self-assembly to add functional value to a robotic system, and even less on quantifying the contribution of self-assembly to system performance.**

In this study we demonstrate and quantify the performance benefits of i) acting as a physically larger self-assembled entity, ii) using self-assembly adaptively and iii) making the robots morphologically aware (the self-assembled robots leverage their new connected morphology in a task specific way).

In our experiments, two real robots must navigate to a target over a-priori unknown terrain. In some cases the terrain can only be overcome by a self-assembled connected entity. In other cases, the robots can reach the target faster by navigating individually.

I. Introduction

Self-assembly is a widely observed naturally occurring phenomenon [16]. Self-assembly adds functional value to biological systems and has evolved both at the cellular level and at the level of distinct organisms [1], [2]. Members of the ant species *Œcophylla longinoda*, for example, link to one another to form bridges that other ants can then traverse [9].

Self-assembly in biological systems has been of inspiration to the robotics community [5]. This study concerns the use of self-assembly with autonomous robots [7]. Existing systems of particular relevance, therefore, are those in which the components that assemble are self-propelled. A variety of such systems have been implemented and studied over the past fifty years [4], [6], [8], [12], [14], [17], [18]. However, there has been little research on how self-assembly can add functional value to a system of autonomous robots, and even less on quantifying the contribution of self-assembly to system performance.

In this study we consider a task that requires two real robots to navigate towards a target light source over a-priori unknown terrain. In some of our experimental environments the terrain that the robots encounter can be navigated by a single robot individually. In other environments, the robots must self-assemble and navigate as a connected entity to successfully reach the target.

We demonstrate and quantify the contribution to system performance of the self-assembly mechanism. Initially, we consider the simple benefits of scale conferred by self-assembly: some tasks can be carried out more effectively if the robots act as a larger connected entity [10]. For example, a larger robotic entity might have increased stability for rough terrain navigation [6] or increased strength for object transportation [15].

We go on to show that the system is more efficient if the robots choose autonomously when and if to assemble based on the environment they encounter. Such adaptivity is innovative, as in most existing studies on self-assembly in collective (and modular) robotics the robots are statically programmed in advance either to act individually or to self-assemble. Typically, the experiment is set up to provide a context in which the pre-programmed behaviour appears meaningful. In this study, we quantify the benefits of adaptive self-assembly over and above static pre-programmed behaviour, extending our previous work in which we implemented an adaptive self-assembly mechanism [13].

Finally, we argue that self-assembling systems can improve their performance by making the connecting robots 'morphology aware'. We present a mechanism through which the self-assembled robots leverage their new connected morphology in a task oriented way. For our task this means that the self-assembled robotic entity rotates its body with respect to the terrain it encounters so as to maximise its stability.

II. Experimental Setup

A. The S-Bot

We use the SWARM-BOT robotic platform [11]. This platform is made up of independent 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 $12\,\mathrm{cm}$ high without its perspex camera turret, and has a diameter of $12\,\mathrm{cm}$ without its gripper. Thanks to its traction system that combines tracks and wheels, the *s-bot* is mobile 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

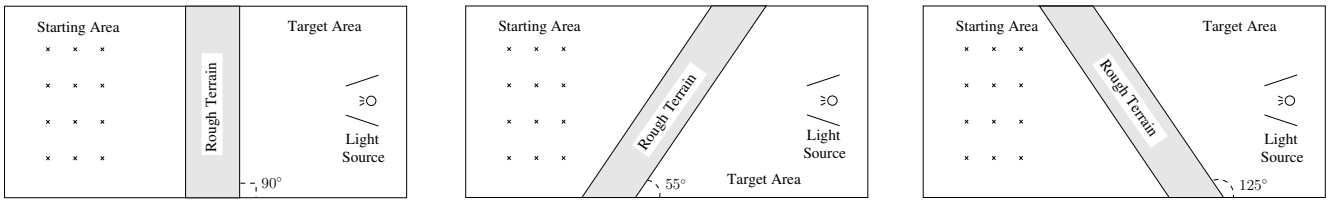

Fig. 2. Scale diagram of the three possible hill orientations. The arena for each of the seven environments used in this study measures 210 cm x 105 cm. Three environments contain the 'moderate' hill (2.8 cm high, navigable by a single *s-bot*). Three environments contain the 'difficult' hill (6.5 cm high, not navigable by a single *s-bot*). One environment has no hill. The starting area and target area are demarcated by the hill and the arena boundaries (in the no-hill environment the target area is considered to be the same as for environments with hill orientation 1). Starting positions are marked by crosses.

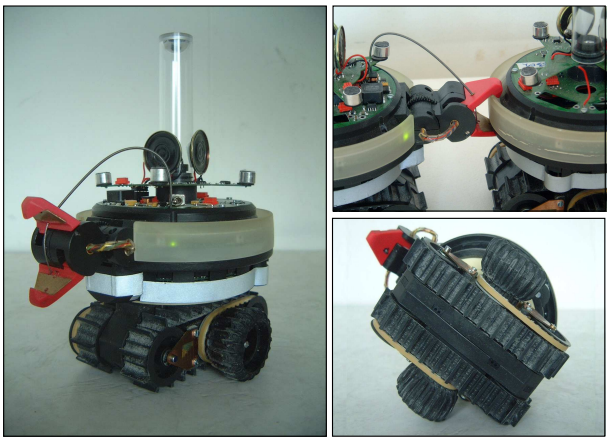

Fig. 1. Left: The *s-bot*. Above Right: The *s-bot* connection mechanism. Below Right: The *s-bot* traction system.

s-bots. An optical light barrier inside the *s-bot* gripper indicates when another *s-bot*'s ring (or another graspable object) is between the jaws of the gripper. *S-bots* advertise their location by means of 8 sets of RGB coloured LEDs 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 40 cm away or an external light source up to 200 cm away. The *s-bot* has 15 proximity sensors distributed around its body that allow for the detection of obstacles. A 3-axes accelerometer provides information on the *s-bot*'s inclination that can be used to detect if the *s-bot* is in danger of falling.

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).

B. The Task

We conduct experiments in seven different environments (see Fig. 2). In each environment we designate a 'starting area' and a 'target area'. Six of the seven environments contain a hill. In these environments the target area is the region containing the light source demarcated by the hill and the walls of the arena. In the hill-less environment we designate the 'target' area as a specific rectangular region containing the light source.

We use two types of hill — a 'moderate' hill and a 'difficult' hill. The moderate hill is 2.8 cm high and can be overcome by a single *s-bot*. The difficult hill is 6.5 cm high and is only navigable if two or more *s-bots* assemble into a bigger connected entity (the steepness of the hill would cause a single *s-bot* to topple).

The initial position of each *s-bot* in the starting area is assigned randomly by uniformly sampling without replacement from a set of 12 possible starting points. The *s-bot*'s initial orientation is chosen randomly from a set of 4 possible directions. To complete the task the *s-bots* must reach the target area without toppling over.

III. Simple Benefits of Scale

In this section we consider the simple benefits of scale derived by acting as a larger connected entity. We conducted experiments with two different controllers in the difficult hill environments. When executing the *adaptive self-assembly* controller, the *s-bots* make use of self-assembly in order to overcome the hill. When executing the *phototaxis only* controller, the *s-bots* attempt to navigate to the target individually. We first present the two controllers, then discuss their relative performance.

A. Control

All of the controllers presented in this study (not just in this section) have the following features in common: each *s-bot* is completely autonomous, and has no a-priori knowledge of the environment it is in or of its initial position and orientation. A single 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 location and to provide indications of their internal state to other *s-bots* within visual range.

1) Adaptive Self-Assembly Controller: The controller is a finite state machine (see Fig. 3). Fig. 4 illustrates a two *s-bot* system executing the controller. An *s-bot* starts by illuminating its blue LEDs and navigating independently towards the target light source (state `Solo_Phototaxis`). If the *s-bot* finds itself on a hill too difficult for it to pass alone (i.e., if its 3D accelerometers detect an inclination angle that exceeds a preprogrammed threshold), or if it sees a green *s-bot* or a

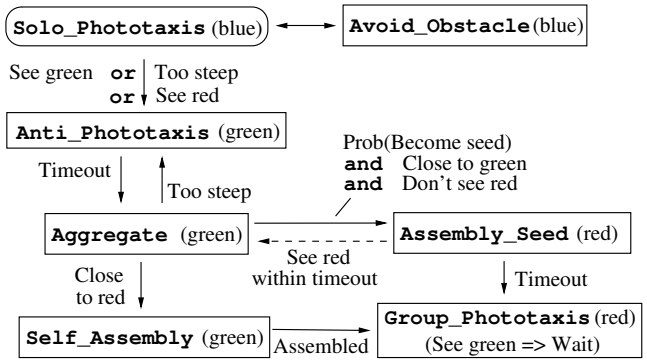

Fig. 3. Finite State Machine representing the Adaptive Self-Assembly Controller. The starting state is `Solo_Phototaxis`. Colours in parentheses refer to the LEDs that are illuminated in the corresponding state.

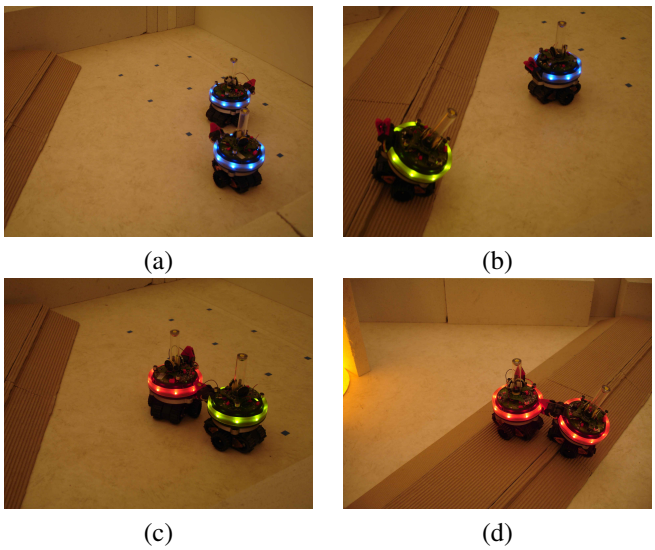

Fig. 4. (a): Two *s-bots* start from random positions and orientations. Initially, they perform individual phototaxis, and have their blue LEDs illuminated. (b): One *s-bot* detects a slope it cannot overcome alone and illuminates green LEDs. The other *s-bot* detects colour green (local communication). (c): The group aggregates and self-assembles. (d): The *s-bots* collectively overcome the rough terrain and reach the target area

red *s-bot*, it retreats away from the hill for a given length of time (state `Anti_Phototaxis`). It then switches into state `Aggregate`, illuminates its green LEDs and tries to get close to a red (assembled) *s-bot*, 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 trigger self-assembly with a given probability by becoming a stationary seed (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 `Group_Phototaxis`. If an aggregating *s-bot* gets sufficiently close to a red (assembled) *s-bot*, then it starts self-assembling (state `Self_Assembly`). Assembled *s-bots* switch to state `Group_Phototaxis`. An *s-bot* in state `Group_Phototaxis` illuminates its red LEDs, and navigates (collectively) to the target light source once it can no longer detect any green (aggregating) *s-bots*.[1]

2) Phototaxis Only Controller: This controller is a modified version of the *adaptive self-assembly* controller in which the transition to state `Anti_Phototaxis` is disabled. Thus, only the states `Solo_Phototaxis` and `Avoid_Obstacle` are executed. Each *s-bot* moves independently towards the light at a constant speed irrespective of the type of terrain it encounters.

B. Comparative Performance

We conducted 60 trials with each controller in the difficult hill environments (20 trials per hill orientation). To minimise damage to the robots, we used just a single robot for the trials of the *phototaxis-only* controller.[2]

When executing the *phototaxis-only* controller, 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.

When executing the *adaptive self-assembly* controller, 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.

Overall, the results show that the additional stability provided by navigating as a larger self-assembled entity (i.e., 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, $p < 0.001$).

IV. Efficiency Gains through Adaptivity

We demonstrate the feasibility of adaptive self-assembly by analysing the behaviour of the *adaptive self-assembly* controller (see section III-A.1) in different environments, some of which require self-assembly (difficult hill environments), and some of which are navigable individually (moderate hill and no-hill environments). In section IV-A we demonstrate that the system is able to 'choose' appropriate behaviour adaptively based on the environment encountered.

In section IV-B we quantify the benefits of adaptivity by comparing task completion speed of the *adaptive self-assembly* controller against that of a non-adaptive controller — the *preemptive self-assembly* controller. The *preemptive self-assembly* controller is a modified version of the *adaptive self-assembly* controller in which the start state is `Aggregate` instead of `Solo_Phototaxis`. Using this

[1]For a more detailed description of this controller refer to [13].

[2]We make the assumption that the task completion rate of a single robot equals the task completion rate of two independently navigating *s-bots*.

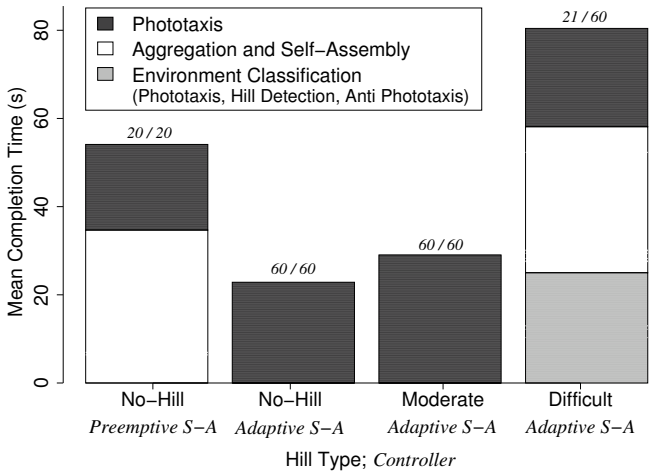

Fig. 5. Break-down of mean completion times for *s-bots* using the *preemptive self-assembly* controller (left bar) and the *adaptive self-assembly* controller (right three bars) in no-hill, moderate hill and difficult hill environments. Only data from completed trials are presented (number of completed trials and number of trials in total are indicated above each bar).

controller, the *s-bots* thus aggregate and self-assemble irrespective of the environment. The connected *swarm-bot* entity then performs group phototaxis to the light source.

A. Validation of the Adaptive Mechanism

With the adaptive self-assembly controller, we conducted 60 trials in the difficult hill environments (already discussed in section III-B), 60 trials in the moderate hill environments and 20 trials in the no-hill environment.

In the 60 trials in the difficult hill environments, both *s-bots* successfully detected the slope in every trial and 'chose' to assemble into a two *s-bot swarm-bot*. In the 80 trials in the moderate hill and no-hill environments, both *s-bots* correctly 'chose' not to self-assemble and to navigate to the target individually. Thus the adaptive mechanism correctly classified the environment in all 140 trials.

B. Comparative Performance

We conducted 20 trials with the *preemptive self-assembly* controller in the no-hill environment. This provides a baseline against which we compare the *adaptive self-assembly* controller. Throughout this section we assume that the mean completion time of the *preemptive self-assembly* controller in the no-hill environment is a lower bound for the mean completion time of the same controller in moderate hill or difficult hill environments.

Fig. 5 shows a break-down of mean completion times for the *preemptive self-assembly* controller and the *adaptive self-assembly* controllers in no-hill, moderate hill and difficult hill environments.

In the no-hill environment, the *adaptive self-assembly* controller performed significantly better than the *preemptive self-assembly* controller (two-tailed Mann-Whitney, $p < 0.001$). The mean task completion times were respectively $22.9\,\mathrm{s}$ and $54.1\,\mathrm{s}$. The *adaptive self-assembly* controller took on average 57.7% less time to complete the task than the *preemptive self-assembly* controller. Looking at the break-down of the mean completion times in Fig. 5, we can see that *s-bots* using the *preemptive self-assembly* controller 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 *s-bots* using the *adaptive self-assembly* controller was $29.0\,\mathrm{s}$, which is 27.1% more than the mean completion time for the same controller 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 *adaptive self-assembly* controller slow down on the slope to test its navigability. Nevertheless, even using the lower bound mean completion time for the *preemptive self-assembly* controller, the *adaptive self-assembly* controller still significantly outperforms the *preemptive self-assembly* controller (two-tailed Mann-Whitney, $p < 0.001$). In the moderate hill environments, the mean completion time for the *adaptive self-assembly* controller was 46.4% less than the lower bound mean completion time for the *preemptive self-assembly* controller.

In contrast, in environments where self-assembly is necessary (difficult hill environments), it is intuitively clear that the *preemptive self-assembly* controller is more efficient than the *adaptive self-assembly* controller. *S-bots* using the *adaptive self-assembly* controller have a major extra overhead of environment classification consisting of initial solo-phototaxis, hill detection and anti-phototaxis (see Fig. 5). The mean completion time for the *adaptive self-assembly* controller in the difficult hill environments was $80.4\,\mathrm{s}$.

Thus, the relative frequency with which the *s-bots* encounter the different environments determines which of the two controllers 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 percentage of no-hill environments encountered, α, for which the efficiency of the two controllers is identical[†]:

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

$$\Rightarrow \alpha = 0.457 \quad (2)$$

We conclude that the *adaptive self-assembly* controller is more efficient than the *preemptive self-assembly* controller if more than $45.7\%^{\dagger}$ of encountered environments are no-hill environments. If we consider a distribution of environments containing only moderate hill environments and difficult hill environments, a similar analysis reveals that the *adaptive self-assembly* controller will be more efficient if at least $51.3\%^{\dagger}$ of the environments are moderate hill environments.

V. Benefits of Morphology Awareness

Using the *adaptive self-assembly* controller, the *s-bots* always correctly responded to the difficult hill environments by self-assembling. However, the overall task completion

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

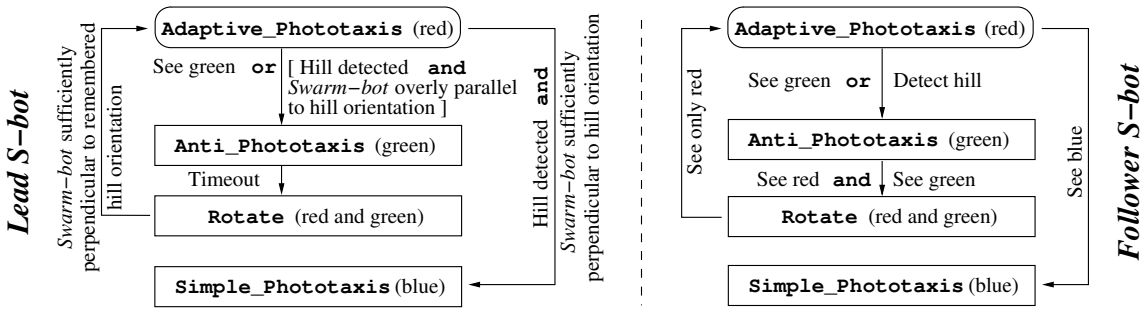

Fig. 6. The *morphology aware* controller — finite state machine extension of the *adaptive self-assembly* controller. The *s-bot* first executes the *adaptive self-assembly* controller (see Fig. 3). However, instead of executing the `Group_Phototaxis` state the *s-bot* switches into state `Adaptive_Phototaxis`, either as a *lead s-bot* (the *s-bot* that seeded self-assembly) or as a *follower s-bot* (all other *s-bots*).

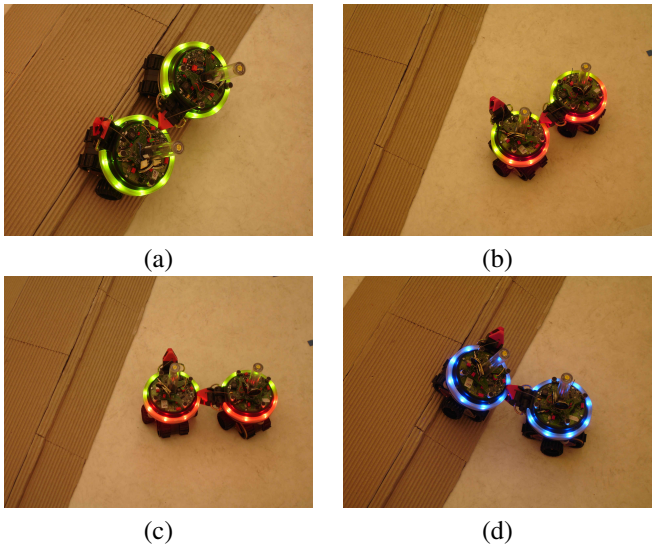

Fig. 7. Execution of the *morphology aware* controller. (a): The *s-bots* have already adaptively aggregated and self-assembled. The connected *swarm-bot* approaches the hill with an inappropriate orientation. (b): The *swarm-bot* performs antiphototaxis away from the hill and starts to rotate. (c): The *swarm-bot* rotates until it has a more appropriate orientation (based on memory of the hill orientation). (d): The *swarm-bot* recognises that it has an appropriate orientation and overcomes the hill.

rate for these environments was not high — only 35% of the *s-bots* completed the task. This is because a two *s-bot swarm-bot* topples over whenever it approaches the difficult hill with an orientation that is insufficiently perpendicular to the orientation of the hill. Such inappropriate *swarm-bot* orientations occurred frequently in the difficult hill experiments, as the approach orientation of the *swarm-bot* was purely random. Immediately after self-assembling into a *swarm-bot*, the constituent *s-bots* would both just perform phototaxis. Thus the *swarm-bot* would approach the hill with whatever orientation it happened to assume as it was formed.

In this section, we present a mechanism which allows the *s-bots* to leverage the morphology of the connected *swarm-bot* in a task oriented way. The *s-bots* ensure that the *swarm-bot* is appropriately rotated with respect to any rough terrain encountered so as to keep the vertical projection of the *swarm-bot*'s center of gravity inside its footprint.

A. Morphology Aware Controller

We extended the *adaptive self-assembly* controller (see Fig. 3) to allow a *swarm-bot* that assembles with a linear morphology to detect the orientation of a hill and rotate in order to approach the hill at an appropriate angle.[3]

Fig. 6 shows the finite state machine extension for the *morphology aware* controller. Fig. 7 illustrates a two *s-bot* system executing the *morphology aware* controller. The *s-bots* start by executing the *adaptive self-assembly* controller. However, instead of executing the final `Group_Phototaxis` state of the *adaptive self-assembly* controller, the *s-bots* start execution of the *morphology aware* controller in state `Adaptive_Phototaxis`.

In state `Adaptive_Phototaxis` the *s-bots* perform collective phototaxis while constantly checking the orientation of any hills 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 *swarm-bot* is appropriately rotated with respect to the hill (perpendicular to the orientation of the hill with a tolerance of 20°) the *s-bots* continue performing phototaxis to the light source, but no longer check the orientation of any encountered hills (state `Simple_Phototaxis`). If the *swarm-bot* is not appropriately rotated, 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 *swarm-bot* is appropriately rotated with respect to the remembered hill orientation (state `Rotate`), and then start performing collective phototaxis again (state `Adaptive_Phototaxis`).

We use a leader-follower architecture. The *s-bot* that seeded 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

[3]Note that in our experiments, since there are only two *s-bots* self-assembling, the assembled *swarm-bot* must be linear, and there will always be a single *lead s-bot* and a single *follower s-bot*. The controller has, however, been written to be applicable to a linear *swarm-bot* of arbitrary length — hence the instruction propagation mechanism. 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 [3].

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 (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 other direction.

Below, we detail the different states of the *morphology aware* controller and the state transition conditions.

State `Adaptive_Phototaxis`

Both the *lead s-bot* and the *follower s-bot* illuminate their red LEDs.

1) Lead S-Bot: If the *lead s-bot* detects a hill, it monitors whether or not the *swarm-bot* is appropriately rotated with respect to the orientation of the hill. Orientation of the hill is measured using the accelerometers. Orientation of the *swarm-bot* is calculated based on the positions of nearby LEDs detected with the camera. If the *swarm-bot* is appropriately rotated, the *lead s-bot* switches into state `Simple_Phototaxis`. Otherwise, the *lead s-bot* switches into state `Anti_Phototaxis`. When the *lead s-bot* switches into state `Anti_Phototaxis` it stores the orientation of the hill with respect to the target light source for later use in state `Rotate`.

If the *lead s-bot* perceives green, it assumes that a *follower s-bot* of the linear *swarm-bot* has detected the presence of the hill, and therefore switches into state `Anti_Phototaxis`. In this case, it uses the direction towards the nearest *follower s-bot* (instead of its accelerometers) to estimate the orientation of the hill.

2) Follower S-Bot: If the *follower s-bot* detects a hill or perceives green, it switches into state `Anti_Phototaxis`. The *follower s-bot* does not check the alignment of the *swarm-bot*.

State `Anti_Phototaxis`

Both the *lead s-bot* and the *follower s-bot* illuminate their green LEDs. The *lead s-bot* switches to state `Rotate` after a timeout. *Follower s-bots* switch into state `Rotate` when they see red and green LEDs in front of them.

State `Rotate`

3) Lead S-Bot: The *lead s-bot* continually compares the current orientation of the *swarm-bot* against the remembered hill orientation. If the *swarm-bot* is appropriately oriented and the *lead s-bot* is at the end of the *swarm-bot* closest to the hill, the *lead s-bot* switches into state `Adaptive_Phototaxis`. Otherwise, the *swarm-bot* needs to rotate—the *lead s-bot* moves in a direction perpendicular to the orientation of the *swarm-bot* so as to rotate the *swarm-bot* either clockwise or anti-clockwise as appropriate.

While rotating, the *lead s-bot* communicates rotation instructions to the *follower s-bots* by illuminating its green LEDs in the direction in which it is moving and its red LEDs in the opposite direction.

4) Follower S-Bot: If the *follower s-bot* can see red to its left and green to its right, it interprets this as an instruction to rotate the *swarm-bot* clockwise. It therefore moves to its left (in Fig. 7 the *s-bot* furthest from the hill is a *follower s-bot* carrying out a 'rotate clockwise' instruction). If the *follower s-bot* can see red to its right and green to its left, it instead rotates the *swarm-bot* anti-clockwise by moving to its right. If the *follower s-bot* only detects red (without any green), it switches into state `Adaptive_Phototaxis`.

State `Simple_Phototaxis`

Both the *lead s-bot* and the *follower s-bot* illuminate their blue LEDs and perform collective phototaxis.

TABLE I

EXPERIMENTAL RESULTS FOR ENVIRONMENTS WITH THE DIFFICULT HILL. EACH CONTROLLER WAS EVALUATED IN 60 INDEPENDENT TRIALS (20 TRIALS PER HILL ORIENTATION).

Controller	Assembled	Completed
Phototaxis-Only	0%	0%
Adaptive Self-Assembly	100%	35%
Morphology Aware	100%	100%

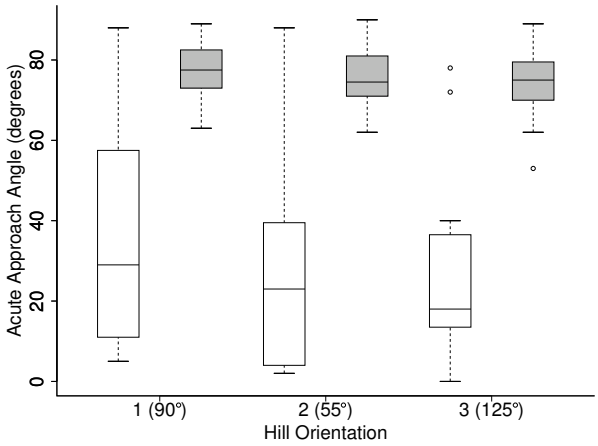

Fig. 8. Box-and-whisker plot showing acute approach angles (orientation of the *swarm-bot* with respect to the orientation of the hill when the *swarm-bot* first makes contact with the hill). If the *swarm-bot* approached the hill more than once, only data from the final approach is shown. Each box represents 20 trials. White boxes: *swarm-bots* without morphology awareness. Gray boxes: *swarm-bots* with morphology awareness.

B. Results

Table I summarises the results for the *morphology aware* controller in the difficult hill environments. Over 60 trials, the robots achieved the optimal task completion rate of 100% (see last row of the table). This increase in task completion rate can be attributed to morphology awareness, that ensures the *swarm-bot* is appropriately rotated with respect to the hill. The effectiveness of the rotation mechanism can been seen in Fig. 8. Morphology aware *swarm-bots* orient themselves

against each of the three hill orientations significantly better than *swarm-bots* that are not morphology aware (two-tailed Mann-Whitney, $p < 0.001$).

When the *s-bots* leveraged the morphology of the connected *swarm-bot* in a task specific way, they achieved an optimal completion rate. Note, however, that the *morphology aware* controller required additional time for task completion, as the *swarm-bot* has to retreat back from the hill, rotate, and approach the hill with a different orientation. The rotation mechanism on average took up 37.1% of the total task completion time.

VI. 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 a quantitative analysis of the performance benefits of self-assembly in a robotic system. We showed that simple advantages of scale derived from acting as a physically larger self-assembled entity allowed for a significant improvement in task completion (from 0% to 35%). We went on to show that significant benefits in system efficiency could be derived by making self-assembly adaptive—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 task completion (from 35% to 100%) could be achieved if the robots leveraged their connected morphology in a task specific way.

The controllers used in this study were designed to be scalable. In particular, each robot executed the same finite state machine independently and only local (visual) communication was used. The scalability of the core self-assembly component has been verified previously with up to 16 real robots [6]. It remains, however, to test the scalability of the adaptive self-assembly mechanism and of the morphology aware controller. This will require the design of a new robotic task, since the current robotic task has already been solved optimally with two robots.

Our system was designed to leverage a particular connected morphology, but had no way of selecting what morphology was formed. We believe that robotic systems could benefit by using 'morphology selection' in addition to the 'morphology awareness' we demonstrated in this paper. To this end, we are currently investigating ways of generating specific connected morphologies using self-assembling robots [3].

VII. Acknowledgements

This work was supported by the ANTS project, an *Action de Recherche Concertée* funded by the Scientific Research Directorate of the French Community of Belgium; by the SWARMANOID project, funded by the Future and Emerging Technologies programme of the European Commission (grant IST-022888); and by COMP2SYS, a Marie Curie Early Stage Research Training Site funded by the European Community's Sixth Framework Programme (grant MEST-CT-2004-505079). The information provided is the sole responsibility of the authors and does not reflect the Community's opinion. The Community is not responsible for any use that might be made of data in this publication. Roderich Groß acknowledges support by the European Commission through the Marie Curie Transfer of Knowledge project BRIDGET (MKTD-CD 2005029961). Marco Dorigo acknowledges support from the Belgian FNRS, of which he is a Research Director.

References

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

[2] D. L. D. Caspar. Design principles in organized biological structures. In G. E. W. Wolstenholme and M. O'Connor, editors, *Principles of Biomolecular Organization*, pages 7–34. J. & A. Churchill Ltd., London, 1966.

[3] A. L. Christensen, R. O'Grady, and M. Dorigo. Morphology control in a self-assembling multi-robot system. *IEEE Robot. Automat. Mag.*, (in press), 2007. Special Issue: Robotic Self-Diagnosis, Self-Repair, Self-Replication and Self-Assembly.

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

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

[6] R. Groß, M. Bonani, F. Mondada, and M. Dorigo. Autonomous self-assembly in swarm-bots. *IEEE Trans. Robot.*, 22(6):1115–1130, 2006.

[7] S. Hirose, T. Shirasu, and E. F. Fukushima. Proposal for cooperative robot "Gunryu" composed of autonomous segments. *Robot. Auton. Syst.*, 17(1–2):107–118, 1996.

[8] H. Jacobson. On models of reproduction. *Am. Sci.*, 46:255–284, September 1958.

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

[10] F. Mondada, M. Bonani, A. Guignard, S. Magnenat, C. Studer, and D. Floreano. Superlinear physical performances in a SWARM-BOT. In *Advances in Artificial Life: 8th European Conf., ECAL 2005*, volume 3630 of *Lecture Notes in Artificial Intelligence*, pages 282–291. Springer-Verlag, Berlin, Germany, 2005.

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

[12] S. Murata, K. Kakomura, and H. Kurokawa. Docking experiments of a modular robot by visual feedback. In *Proc. of the 2006 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 625–630. IEEE Computer Society Press, Los Alamitos, CA, 2006.

[13] R. O'Grady, R. Groß, F. Mondada, M. Bonani, and M. Dorigo. Self-assembly on demand in a group of physical autonomous mobile robots navigating rough terrain. In *Advances in Artificial Life: 8th European Conf., ECAL 2005*, volume 3630 of *Lecture Notes in Artificial Intelligence*, pages 272–281. Springer-Verlag, Berlin, Germany, 2005.

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

[15] E. Tuci, R. Groß, V. Trianni, F. Mondada, M. Bonani, and M. Dorigo. Cooperation through self-assembly in multi-robot systems. *ACM Trans. Auton. Adapt. Syst.*, 1(2):115–150, 2006.

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

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

[18] V. Zykov, E. Mytilinaios, B. Adams, and H. Lipson. Self-reproducing machines. *Nature*, 435(7039):163–164, 2005.