

Enhanced Directional Self-Assembly Based on Active Recruitment and Guidance

Nithin Mathews[†], Anders Lyhne Christensen[‡], Rehan O’Grady[†], Philippe Rétornaz[§],
Michael Bonani[§], Francesco Mondada[§], and Marco Dorigo[†]

Abstract—We introduce enhanced directional self-assembly (EDSA) – a novel mechanism for morphology growth through the creation of directed connections in a self-assembling multi-robot system. In our approach, a robot inviting a physical connection actively recruits the best located neighboring robot and guides the recruit to the location on its chassis where the connection is required. The proposed mechanism relies on local, high-speed communication between connection inviting robots and their recruits. Communication is based on a hybrid technology that combines radio and infrared to provide local relative positioning information when messages are transmitted between adjacent robots. Experiments with real robotic hardware show that EDSA is precise (misalignment of only 1.2° on average), robust (100% success rate for the experiments in this study) and fast (16.1 seconds on average from a distance of 80 cm). We show how the speed and precision of the new approach enable adaptive recruitment and connection in dynamic environments, a high degree of parallelism, and growth of a moving morphology.

I. INTRODUCTION

Self-assembling robots can connect to each other autonomously to form composite robotic entities [1]. Self-assembling systems can respond flexibly to their environment by forming appropriately shaped composite robotic entities. These composite entities are commonly “grown”, by allowing robots that are on a growing edge of a morphology (henceforth *extending* robots) to invite connections from robots that are not yet part of the morphology (henceforth *free* robots). See Fig. 1 for an example of a morphology that has been grown in response to its environment.

Many different algorithms have been proposed to control morphology growth [3]–[7]. Each morphology growth algorithm relies on some underlying behavior to form inter-robot connections. For all such existing behaviors, connection forming efficiency is dependent on the density of robots in the environment. Note that this is true for both “externally propelled” and “self-propelled” self-assembling platforms. In externally propelled systems, robots take kinetic energy from their environment to move and arrive at morphology extension points as a result of their stochastic movements [8],

[†]N. Mathews, R. O’Grady, and M. Dorigo are with IRIDIA-CoDE, Université Libre de Bruxelles, 50 Avenue F. Roosevelt, CP 194/6, 1050 Bruxelles, Belgium {nmathews, rogrady, mdorigo}@ulb.ac.be

[‡]A. L. Christensen is with Instituto de Telecomunicações, Instituto Universitário de Lisboa (ISCTE-IUL), Av. Rovisco Pais 1, Lisboa 1049-001, Portugal anders.christensen@iscte.pt

[§]P. Rétornaz, M. Bonani, and F. Mondada are with EPFL-LSRO, École Polytechnique Fédérale de Lausanne, Station 9, 1015 Lausanne, Switzerland {philippe.retornaz, michael.bonani, francesco.mondada}@epfl.ch

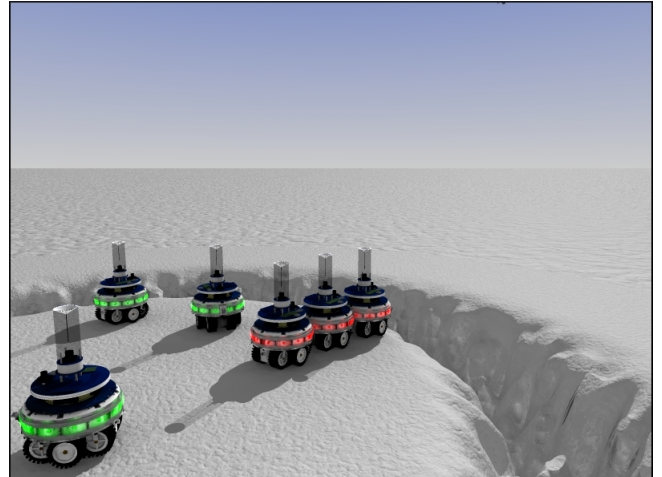


Fig. 1: A group of marXbots [2] self-assemble into a chain-like morphology to cross a gully.

[9]. In self-propelled systems, robots generate their own kinetic energy from internal power sources [10], [11]. Usually, however, free robots in self-propelled systems are attracted to extension points within their own sensory range, and must negotiate between each other (usually with stochastic mechanisms) to determine which robot attaches where.

In both externally propelled and self-propelled systems, low robot densities may result in free robots clustering around some extensions points, leaving others empty (thus slowing morphology growth). And at high densities, interference between robots can become a problem, again slowing morphology growth. In [6], a form of passive recruitment is used to mitigate interference issues — the extending robot stops attracting robots as soon as a single free robot is in range. However the system was incapable of parallel morphology growth, and was only demonstrated in simulation.

“Directional Self-Assembly” was introduced in [10] as a connection forming behavior for the self-propelled *swarm-bot* platform. Its novel feature was the use of camera and LEDs to enable direction specific connections between robots. These direction specific connections allowed a single extending robot to have multiple *extension points* on its body, to each of which a free robot could attach. However, the connection of multiple free robots to a single extending robot was strictly sequential (the morphology could be extended in parallel by multiple extending robots). The system also relied on stochastic movement of free robots, thus rendering

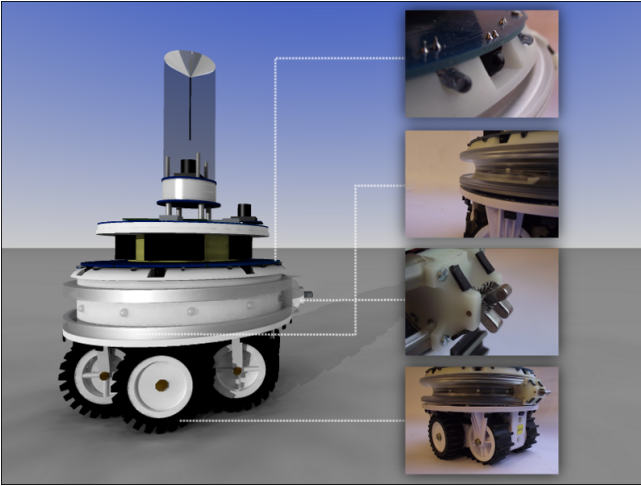


Fig. 2: The configuration of the marXbot used in this study. The insets show (from top to bottom): i) two infrared emitters and one sensor of the mxRAB device, ii) the docking ring, iii) the docking unit with three fingers, and iv) the treels.

it susceptible to density related inefficiencies.

In this paper, we propose an enhanced directional self-assembly mechanism (EDSA) that avoids the inefficiencies of stochastic free robot attraction and allows for parallel morphology growth from a single extending robot. EDSA is based on the self-propelled marXbot platform (see Sect. II), and relies on the marXbot’s mxRAB device (see Sect. III) — a novel situated communication device that uses infrared and radio in combination. In EDSA, extending robots actively recruit and guide free robots (see Sect. IV). We conduct a series of benchmark experiments using the same setup and measures defined in [10] to show how our new system goes beyond the state-of-the-art (see Sect. V). We go on to show how the speed and precision of the new approach allow us to achieve previously unattainable types of morphology growth (see Sect. VI). For example, we show how morphologies can be grown while the morphology as a whole is moving.

II. THE MARXBOT PLATFORM

The marXbot [2] is a modular robot that has a circular chassis with a diameter of 17 cm (see Fig. 2). A combination of tracks and wheels provides the marXbots with differential drive motion capabilities. The marXbot configuration used in this study includes a rotatable docking module that enables marXbots to physically connect to one another. The docking module is composed of an active docking unit with three fingers and a passive docking ring. A marXbot can form a physical connection with another marXbot by inserting the docking unit into its docking ring and then opening the three fingers.

The marXbot is equipped with 12 RGB-colored LEDs distributed around its docking ring. The LEDs allow a marXbot to visually display its internal state to nearby robots. A force sensor is placed between the chassis and the rotatable docking module. The sensor gives the marXbot the capacity to register the forces applied to the unit in the XY-plane.

This allows physically connected marXbots to coordinate their motion.

A marXbot uses its mxRAB device (**marXbot Range And Bearing**) to communicate both with other marXbots and with other robotic platforms [12], [13]. The mxRAB device is a combination of 16 infrared transmitters, 8 infrared receivers, and one radio chip. The mxRAB device provides the marXbot with high-speed communication (1 Mbps) and relative range and bearing estimates to other robots in line of sight (up to a distance of 5 m). Refer to Sect. III for details about the mxRAB device. Further features of the marXbot include a 3-axis gyroscope, a 2D distance scanner, 24 IR proximity sensors, 8 IR ground sensors, and two 3 megapixel cameras. The marXbot is fully autonomous and is equipped with an ARM 11 processor (i.MX31 clocked at 533 MHz and with 128 MB RAM) running a Linux-based operating system.

III. THE MXRAB DEVICE

The mxRAB device enables *situated communication* between marXbots. Situated communication refers to the ability to receive messages whilst having a spatial sense of where the message originated from [14]. The design of the mxRAB device is based on the work presented in [15]. The mxRAB device achieves situated communication at 10Hz with a combination of infrared and radio technologies. The use of hybrid infrared and radio technology allows higher bandwidth communication than would be possible in an equivalent system that relied only on infrared. The message sending robot emits infrared in all directions that recipient robots in line of sight can use to infer the relative position (range and bearing) of the sender. At the same time, the message sending robot broadcasts the contents of the message over radio. Time division multiplexing is used to ensure that only one robot at a time in the system is transmitting infrared and radio signals. This multiplexing allows receiving robots to associate data received over radio with positioning information inferred from the infrared signal. Although the mxRAB device provides only broadcast communication, one-to-one communication is possible with the implementation of higher level logic (see Sect. IV-A).

The mxRAB device can estimate the range and bearing of robots at a distance of up to 5 m. The infrared receivers are used to triangulate the range and bearing of an incoming infrared signal. The triangulation calculation is based on sensor readings from the receiver that registers the strongest signal and the two adjacent receivers. The mxRAB device has the following shortcomings when estimating relative positions at short distances: due to infrared saturation, the range cannot be estimated for distances shorter than 30 cm. At distances shorter than 15 cm, the bearing estimate becomes noisy and unreliable. These two issues related to the estimation of range and bearing at close distances have to be considered in the design of the robot behavioral control and communication algorithms (see Sect. IV-B).

The mxRAB device uses radio to send 14 byte data packets (see Fig. 3). Of the 14 bytes of data sent, the first four bytes

are the *header* and are used internally by the mxRAB device. The first byte is the unique ID number of the mxRAB device and the other three bytes are used to ensure compatibility with flying robots presented in [12]. The remaining 10 bytes can be used by the control program. We call the content of these 10 bytes the *message*. In this study, we use the first byte to specify the *message type* and the remaining 9 bytes for the *message payload*.

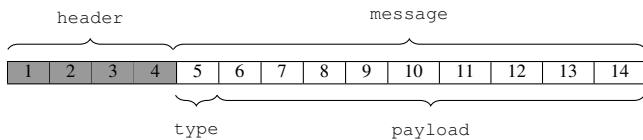


Fig. 3: A depiction of the data packet (14 bytes) transmitted by the mxRAB device. The first 4 bytes (shown in shades) are used internally by the mxRAB device (see text). The remaining 10 bytes make up the actual message. A message has a type (5th byte) and a payload (bytes 6 to 14).

Synchronisation of the time division multiplexing between mxRAB devices is handled by the same 2.4 GHz radio transceiver that is used to transmit data. The time division multiplexing is configured in advance using a predefined sequence that assigns a timeslot to each mxRAB device. The current implementation of the mxRAB device allows the assignation of up to 25 timeslots, giving an upper limit of 25 robots in any experiment involving the mxRAB device. This upper limit is a function of the 10 Hz refresh rate and the minimum time (roughly 4 ms) that a single infrared signal needs to stabilize, be read and then dissipate. The 25 robot limit could thus be raised by reducing the refresh rate. For true scalability, however, the mxRAB system would need to be extended to allow the dynamic formation of local groups of robots that perform time division multiplexing independently. Each local group would then still have a maximum size 25 robots, but the number of groups, and therefore the total number of robots in the system, could be potentially unlimited.

The mxRAB device enables a control step of 100 ms during which each robot can receive data packets from up to 24 other robots. In this paper, we use the set $R = \{r_1, \dots, r_n\}$ to denote the output of the mxRAB device, where r_i is a neighboring robot and $n \leq 24$ is the number of robots that are within the range of 5 m. Each $r \in R$ is a tuple of the form (*mxRABID*, *range*, *bearing*, *message*).

IV. ENHANCED DIRECTIONAL SELF-ASSEMBLY

EDSA is based on the situated communication capabilities provided by the mxRAB device. Free robots broadcast their availability to be recruited. An extending robot recruits and guides the best situated free robot to the point on its body at which the free robot should connect. Once the recruit has successfully formed the connection, it sends an acknowledgement to the extending robot. Fig. 4 illustrates how a connection is formed between an extending marXbot and the best situated free marXbot.

We implemented two behavioral controllers to demonstrate EDSA on real robotic hardware: one for the extending robot and one for the free robots. In our experiments, each robot executes either one of these controllers. No robot has access to any global information. However, in every control step, each robot has access to R , the messages received by the mxRAB device. Each robot also sends one of the messages listed in Tab. I. A free robot sends the message `FREE` to broadcast its availability to be recruited, whereas the message `RECRUITED` is sent by an already recruited robot while maneuvering to a connection slot. While the former has an empty message content, the latter sends the unique mxRAB ID of the extending robot which recruited it. We describe the remaining two messages and the semantics of their contents when detailing the two behavioral controllers in Sect. IV-A and Sect. IV-B, respectively.

TABLE I: Messages exchanged using the mxRAB device.

Message type	Message payload
<code>FREE</code>	-
<code>RECRUITED</code>	ID
<code>GUIDANCE</code>	α, σ_i, ID
<code>CONNECTED</code>	σ_i, ID

A. Behavioral control for the extending robot

A marXbot can invite a connection on the whole perimeter of its chassis, except where the docking unit is mounted (between 340° and 20°). This corresponds to an arc of 320° within which a connection can occur. Up to 6 marXbots can connect around the chassis of another marXbot — any two extension points must be at least 53° apart from one another (due to physical constraints). Given a set $\Sigma = \{\sigma_1, \dots, \sigma_n\}$ of extension points, where σ_i is a bearing in the interval $[20^\circ, 340^\circ]$ and $n \leq 6$, the objective of an extending robot is twofold. First, recruit the best situated free robot r for each σ_i ; that is, create a mapping M of each extension point σ_i to a free robot r_i . Second, guide each recruit r_i by providing it with information required to navigate and connect to the extending robot's docking ring at σ_i .

The extending robot executes its recruitment and guidance logic at every control step. The mapping of free robots to extension points is thus continuously revised — if the extending robot detects a free robot which is more favorably located for a given extension point than the robot it has currently recruited, it releases the prior recruit, and recruits the more appropriate free robot instead (see Alg. 1). Tailored guidance instructions are then sent to each recruit, telling them how to navigate to their respective extension points (see Alg. 2).

The continuous recruitment process takes the output of the mxRAB device R and a set Σ of extension points to map each $\sigma_i \in \Sigma$ to one $r_i \in R$. It should be noted that an extending robot either recruits a free robot that is sending the message `FREE`, or a robot that is sending the message `RECRUITED` with the unique ID of extending robot's mxRAB device. The 360° perception of the mxRAB device is divided into 8 sectors of 45° each. The sector

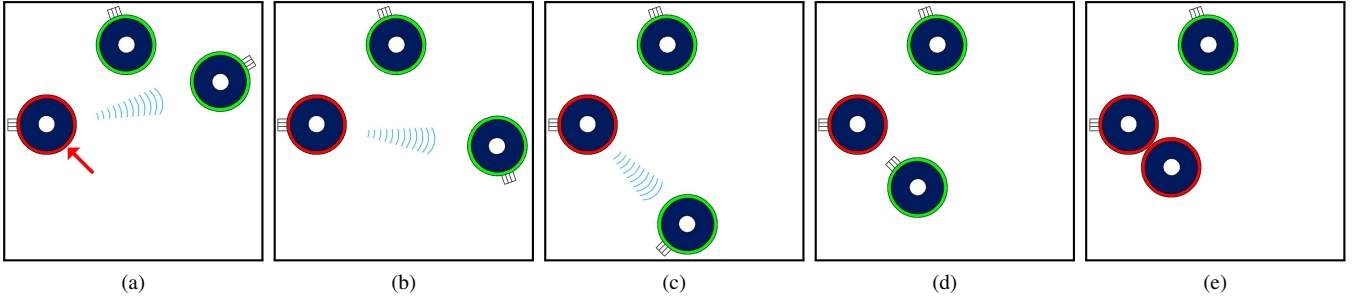


Fig. 4: In EDSA, a connection can be formed between (a) a connection inviting marXbot (shown on center-left with the arrow indicating an expected extension point) and the best situated free marXbot with respect to an extension point. The extending robot recruits the best situated free robot and (b) guides this recruit around its chassis until (c) the recruit is aligned with the extension point. (d) The recruit rotates to face the extending robot and approaches the extension point. (e) A connection is formed between the extending robot and the recruit at the expected extension point.

in which an extending robot will perceive any given free robot depends on the relative bearing of the extending robot and the free robot. In order to find the best situated free robot for an extension point σ_i , the extending robot first orders the sectors according to their proximity to σ_i in descending order. For example, for an extension point at 90° the sectors are ordered as follows: $[45^\circ, 90^\circ]$, $[90^\circ, 135^\circ]$, $[0^\circ, 45^\circ]$, $[135^\circ, 180^\circ]$, $[315^\circ, 0^\circ]$, $[180^\circ, 225^\circ]$, $[270^\circ, 315^\circ]$, and $[225^\circ, 270^\circ]$. The extending robot then goes through this list and finds the closest robot in each sector. The first found robot r_i is mapped as the recruit for extension point σ_i and saved in mapping M . Pseudocode for the extending robot's recruitment logic is shown in Alg. 1.

The extending robot sends tailored instructions to each recruit for the recruit to be able to maneuver to its designated extension point. The extending robot uses the message GUIDANCE to communicate these instructions to its recruits. The message is composed of the inverse bearing α (i.e., the bearing at which the extending robot perceives the recruit), the target bearing σ_i (i.e., the bearing at which the connection is anticipated by the extending robot), and the unique ID of the recruit's mxRAB device. This unique ID needs to be included in the message to enable one-to-one communication (the mxRAB device only provides broadcast communication). Each robot that receives a GUIDANCE message determines whether or not it is the intended recipient by comparing the ID of its own mxRAB device with the ID in the message.

The mxRAB hardware allows every robot to send one message in each control step. Thus, if the extending robot has recruited multiple robots, it must take turns guiding its recruits. In the extreme case, where an extending robot has recruited six robots for six open extension points, each recruit can only be guided every sixth control step – that is every 600 ms. Although this frequency is sufficient to let recruits drive at maximum speed in the system considered in this study, it could be a potential issue in a system that can support many more extension points. The process of guiding a recruit is shown in Alg. 2.

An extending robot executes its behavioral control until

Algorithm 1: The extending robot's recruitment logic.

Input: Output of the mxRAB device R , open extension points Σ

Output: Mapping M of each $\sigma_i \in \Sigma$ to one $r_i \in R$

```

1 foreach Open extension point  $\sigma_i$  in  $\Sigma$  do
2    $S_\sigma \leftarrow \text{Order\_Sectors}(\sigma_i)$ ;
3   while no recruit found and not at the end of  $S_\sigma$  do
4      $s_i \leftarrow \text{Get\_Next\_Sector}(S_\sigma)$ ;
5      $r_i \leftarrow \text{Get\_Closest\_Robot\_In\_Sector}(s_i)$ ;
6     if  $r_i$  is free || was recruited by myself then
7       if  $r_i$  is not recruited for another extension
         point then
8          $M \leftarrow \text{Recruit}(\sigma_i, r_i)$ ;
9       end
10    end
11  end
12 end

```

Algorithm 2: The extending robot's guidance logic.

NB: The GUIDANCE message is comprised of the recruit's inverse bearing, the target bearing, and the unique ID of the recruit's mxRAB device.

Input: Output of the mxRAB device R , mapping M of each $\sigma_i \in \Sigma$ to one $r_i \in R$

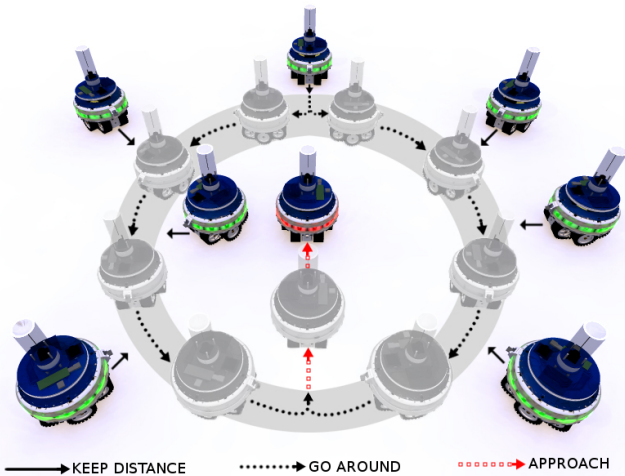
Output: Send tailored instructions to r_σ

```

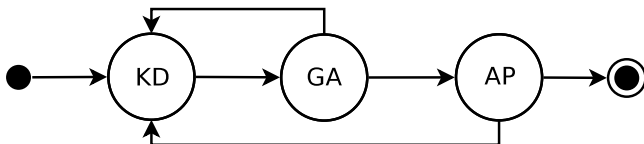
1 foreach Open extension point  $\sigma_i \in M$  do
2    $r_\sigma \leftarrow \text{Get\_Recruit}(\sigma_i, M)$ ;
3   if  $r_\sigma$  was guided less often than another recruit ||
     all recruits were guided equally often then
4      $\alpha \leftarrow \text{Get\_Bearing\_Recruit}(r_\sigma, R)$ ;
5      $ID \leftarrow \text{Get\_Recruit\_ID}(r_\sigma)$ ;
6     Send_Message(GUIDANCE,  $\alpha$ ,  $\sigma_i$ ,  $ID$ );
7     break;
8   end
9 end

```

all extension points are filled. An extension point σ_i is filled when a recruit acknowledges a successful docking with



(a) The trajectories taken by potential recruits to maneuver to a connection slot at the rear of the extending robot (in the center). The three arrow types represent the three segments of a recruit's trajectory. The shaded, ring shaped area represents the go around zone.



(b) The finite state machine representation the recruit's behavior. KD = keep distance, GA = go around, and AP = approach.

Fig. 5: An illustration of the recruit's behavior: (a) the trajectories taken by potential recruits, (b) the finite state machine that segments a trajectory into three.

the message `CONNECTED` with σ_i and the unique ID of extending robot's mxRAB device in its message payload.

B. Behavioral control for the free robots

The message `FREE` is sent by all free robots to broadcast their availability for recruitment. A free robot accepts the recruitment of the first extending robot that sends the message `GUIDANCE` with the unique ID of its mxRAB device in the message payload. The recruit acknowledges its recruitment by sending the message `RECRUITED` with the unique ID of the extending robot's mxRAB device in the message payload. Using the instructions sent by the extending robot, a recruit maneuvers to an extending robot's target bearing and connects to the extension point. A recruit can start from any position within a radius of 5 m around the extending robot. A recruit's trajectory to an extension point takes the imprecision of the mxRAB device at close distances into account (i.e., unavailable range estimations for distances shorter than 30 cm and unreliable bearing estimation for distances shorter than 15 cm). Fig. 5a shows an illustration of possible trajectories taken by recruits starting from different positions around the extending robot.

There are three states that make up the recruit's approach behavior. These three states are shown as a finite state machine in Fig. 5b. They split a recruit's trajectory to a extension point into three different segments that allow a

recruit to i) keep a certain distance to the extending robot, ii) go around the extending robot until the recruit is aligned to the extension point, and iii) to approach and connect to the extension point. Fig. 5a shows how different segments of the recruit's trajectory relates to particular behavioral states.

A recruit's initial state is "keep distance". The recruit moves away or towards the extending robot until range estimation is between 40 cm and 60 cm. This state permits the recruit to reposition itself to a distance where the mxRAB device can accurately estimate the range to the extending robot. Once the recruit is within the distance interval, the recruit enters the "go around" state.

In the "go around" state, a recruit maneuvers around the extending robot until the recruit is aligned to the extension point it is recruited for. A recruit goes around an extending robot either in clockwise or in counter-clockwise direction. The chosen direction is the one that has the shortest angular difference between the inverse bearing α and the target bearing σ_i sent by the extending robot. The recruit goes around the extending robot while remaining within the go around zone, which is a ring shaped area around the extending robot with an inner radius of 40 cm and an outer radius of 60 cm (see Fig. 5a). If a recruit drives out of the go around zone, it switches back to the "keep distance" state. The transition to state "approach" is triggered if the absolute difference between α and σ_i is less than 2° .

In the "approach" state, the recruit rotates until it faces the extending robot – and as a consequence also the extension point σ_i – with its docking unit. The recruit first approaches the extending robot while minimizing the absolute difference between α and σ_i . Then, if the range estimation to the extending robot is lower than 32 cm, the recruit drives towards the extension point, while updating its bearing estimation β to the extending robot using the following filtering method:

$$\beta_t = (1 - c) \cdot \beta_{t-1} + c \cdot \beta_t,$$

where t is the control step and c is $0 \leq c \leq 1$ and is the confidence level of a bearing estimation at close distance. This filtering method makes the recruit less reactive and smoothens the noisy bearing estimations at close distances while still being able to take actual environmental changes into account (such as a extending robot that has moved in the mean time). We experimentally determined $c = 0.7$ to be appropriate. The recruit uses β to drive towards the extension point until the force sensor in the docking module registers contact with the extending robot's docking ring, at which point the recruit stops and opens the three fingers (two to the bottom and one to the top) to dock onto the extending robot's docking ring. The success of a connection is tested by the recruit by trying to rotate its docking module. If the rotation can be carried out, the connection has failed and the recruit goes back to the "keep distance" state. If the docking module cannot be rotated, the connection is firm. The message `CONNECTED` is sent to acknowledge a successful connection to the extending robot. The message payload includes σ_i and the unique ID of the extending

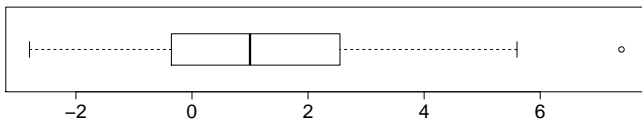


Fig. 6: Boxplot showing the misalignment (in degrees) between the orientation of the recruit and the orientation of the extending robot after they have connected to each other.

robot’s mxRAB device.

While a recruit is navigating to an extension point, another better located free robot may become available and therefore be recruited by the extending robot. Therefore, if a recruit does not receive any `GUIDANCE` messages from the extending robot for 10 consecutive control steps, it assumes that it is no longer recruited and becomes a free robot again.

V. EXPERIMENTAL SETUP AND RESULTS

To enable comparison of our results with the state-of-the-art, we replicated the experimental setup presented in [10].¹ We placed an extending robot with an open extension slot to its rear (i.e., at 180°) in the center of a circle of 80 cm radius. We placed a free robot at 12 equally separated starting positions on the circle. For each starting position, we considered 8 starting orientations of the free robot. We let the robots execute EDSA for each combination of starting position and starting orientation, which resulted in 96 independent trials. Using an overhead camera, we took images of the robots after a connection had occurred and recorded both the angular precision² and recorded the elapsed time for each trial.

The angular precision is plotted in Fig. 6. It shows the misalignment in the orientation of the recruit when compared to that of the extending robot. Note that a perfectly aligned connection has a misalignment of 0° . The median of the misalignment is 1° (the mean is 1.2°) and the standard deviation is 0.77° . Note that while this median is comparable to the results reported in [10], the distribution range of the acquired data is within 10° (interval $[-3^\circ, 7^\circ]$) approximately 6.5x smaller than the results obtained in [10] (distribution range 65° , interval $[-25^\circ, 40^\circ]$). The most extreme misalignment measured in our experiments was 7° — approximately 5.7x better than the 40° reported in [10]. The data indicate that the angular precision achieved using EDSA is more accurate than that achieved using the mechanism reported in [10].

On average, a recruit spent 16.1 s maneuvering to an extension point. The average time spent by a free robot in [10] was 54.3 s. Hence, EDSA is 3.5x faster than the state-of-the-art. This result was achieved despite the fact that in our experiments, the free robot was placed at 80 cm distance from the connection inviting robot rather at 35 cm distance as in [10]. There are three contributing factors to this faster self-assembly time. First, the mxRAB device gives the

robots more precise estimates of relative positions to neighboring robots than the camera and LEDs-based mechanism presented in [10]. Thus, in EDSA the marXbots can drive at higher speeds without sacrificing precision. Second, a recruit in EDSA always takes the shortest direction around the extending robot as it is actively guided. In contrast, the robots in [10] chose the direction randomly. Third, the marXbot has a mechanical advantage over the robotic platform used in [10]. The marXbot’s innovative docking module (that includes the docking unit with the three fingers and the docking ring) forms inter-robot connections faster than the gripper based mechanism of the robots in [10].

All our trials resulted in successful connections between the extending robot and the recruit. Indeed, no failed attempt of EDSA was observed after the development of the controllers had reached a stable stage. However, in 5 trials we observed that the mxRAB device of either one of the robots stopped working because of low battery voltage. These 5 trials were repeated.

VI. FEATURES OF EDSA

In this section, we discuss three features of EDSA to show its potential when used as part of a self-assembling morphology growth system. Complete video footage of all experiments discussed in this section can be seen online at <http://iridia.ulb.ac.be/supp/IridiaSupp2011-001/>.

Adaptive Recruitment. The active recruitment algorithm in EDSA is able to adapt to changing conditions such as the introduction of new free robots, or the malfunction of a recruited robot. Because the mapping from recruited robots to extension points is updated every time step (see Alg. 1), EDSA guarantees that each extending robot will find the free robots most appropriately located with respect to its own extension points.³ Figure 7 shows snapshots of an experiment in which a free robot was introduced into an already on-going self-assembly process, where the extending robot is already guiding a recruit to an extension point. The free robot is introduced close to the extension point. The extending robot adapts to the new situation by freeing up the current recruit, and recruiting the newly introduced free robot.

Enhanced Parallelism. Morphologies branch when multiple free robots connect to a single extending robot. In EDSA, these connections can happen in parallel. In [10], by contrast, multiple free robots could only connect sequentially to a single extending robot. EDSA’s parallelism enhances morphology growth speeds, especially for more sophisticated morphologies with a high degree of branching. Figure 8 shows snapshots of an experiment in which a single extending robot has four extension points. The extending robot recruits and guides four recruits in parallel, one to each extension point.

EDSA also allows different extending robots to work in parallel either when they are part of the same morphology, or

¹Our setup differs only in that the initial distance between the two robots was set to 80 cm (and not to 35 cm as in [10]).

²We used an open source tool named ImageJ: <http://rsbweb.nih.gov/ij/>.

³In fact, there may be a more efficient global allocation of free robots to extension points on multiple extending robots that EDSA in its current form would not find.

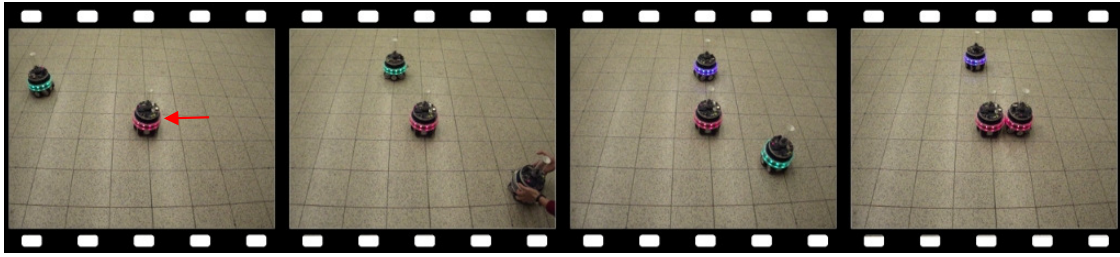


Fig. 7: The extending robot (in the center) opens an extension point at 90° (indicated by the arrow). Initially, the only available free robot is recruited and guided. When a second free robot is introduced at an angle that is smaller with respect to the extension point, the extending robot adapts to the situation and recruits the closer free robot and ignores the initially recruited robot. The initially recruited robot leaves the self-assembly process and becomes available for other tasks.

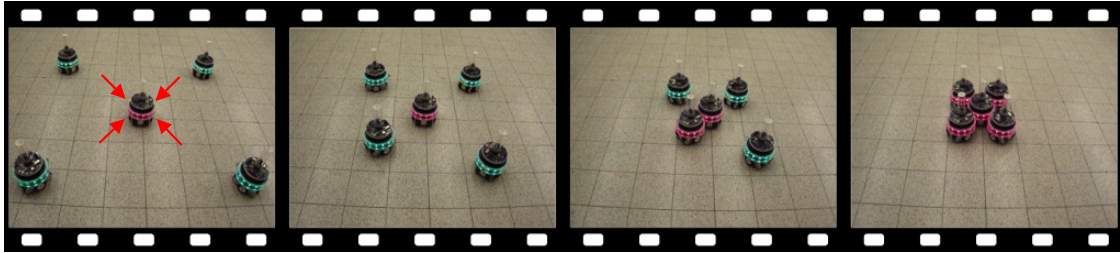


Fig. 8: The extending robot (in the center) opens four extension points at 45° , 135° , 225° , and at 315° . Four free robots are placed around the extending robot. The extending robot recruits the best situated free robot for each extension point and guides each of them to their respective extension point. All four robots are connected within 16 s to form a star morphology.



Fig. 9: A chain (top-right) and a star morphology (bottom-left) are under formation in parallel. The extending robot in the chain morphology seeks for a connection at its 180° . A further extending robot is in the center of the star morphology and seeks for a connection at its 45° . The extension points are indicated with arrows. The extending robots share the available resources (i.e., the two free robots) without hindering each other and complete both morphologies in parallel.

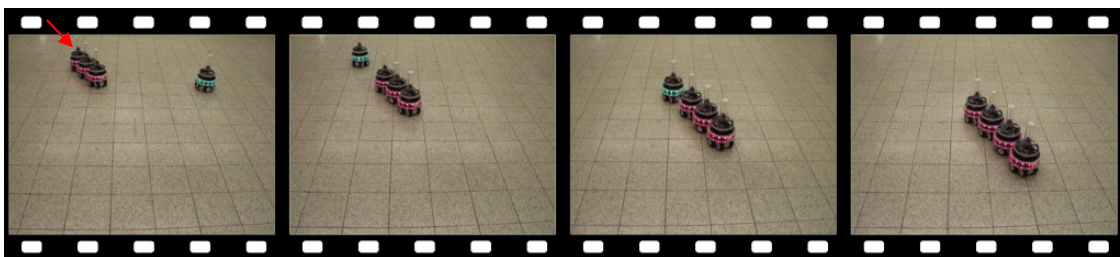


Fig. 10: The morphology extending robot is the third link of a chain morphology that is already in motion towards the camera. The extension point is at the rear (i.e., at 180°) of the morphology extending robot. The recruit first aligns to the extension slot, and then approaches the morphology by driving at a velocity that is higher than that of the morphology.

when they are part of two separate morphologies. In EDSA, neighboring robots (i.e., within communication range) can extend their morphologies in parallel without interfering with each other. Figure 9 shows snapshots of an experiment in

which two morphologies are grown next to each other in parallel.

Morphology Growth in Motion. Using EDSA, we demonstrate controlled growth of a moving morphology for the first

time. Morphology growth in motion could save crucial time for real world morphology growth platforms, by allowing either transport or task execution to commence while growth is ongoing. All previous morphology growth systems have, however, assumed a stationary morphology as a prerequisite for growth. Figure 10 shows snapshots from an experiment in which a chain morphology is extended while the morphology is at all times in motion.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we introduced enhanced directional self-assembly – a mechanism that enables self-assembling, self-propelled robots to connect to each other. Our mechanism differs from previously proposed self-assembly mechanisms in that it is based on i) the mxRAB situated communication device, ii) active recruitment and guidance logic, iii) the marXbot docking module that enables inter-robot connections. Using mxRAB, a robot wishing to receive a connection actively recruits a free robot and guides it to the location where the connection is anticipated.

We presented experimental results that demonstrated the performance and some of the novel features of recruitment-based self-assembly. Our approach was found to be robust (100% success rate), precise (average misalignment of 1.2° degrees), and more than 3.5 times faster than the state-of-the-art. We demonstrated how multiple free robots could be recruited in parallel and connect to the same robot, how multiple free robots could be recruited and connect to different robots, and how connections could be formed while the morphology under formation is in motion.

In ongoing work, we are studying how enhanced directional self-assembly can be applied in the formation of larger, task-specific robot morphologies. We are also studying how larger morphologies can be formed by first self-assembling smaller segments in parallel and then have the segments assemble into the target morphology.

VIII. ACKNOWLEDGMENTS

This work was partially supported by the European Commission via the Future and Emerging Technologies programme (SWARMANOID project, grant IST-022888), and the ERC Advance Grant “E-SWARM: Engineering Swarm Intelligence Systems” (grant 246939). It was also partially supported by the FCT grant PTDC/EEACRO/104658/2008. The information provided is the sole responsibility of the authors and does not reflect the European Commission’s or the European Union’s opinion. The European Commission or the European Union are not responsible for any use that might be made of data appearing in this publication. Marco Dorigo acknowledges support from the Belgian F.R.S.-FNRS, of which he is a research director, through the Swarm-Morph project.

REFERENCES

- [1] R. Groß and M. Dorigo. Self-assembly at the macroscopic scale. *Proceedings of the IEEE*, 96(9):1490–1508, 2008.
- [2] M. Bonani, V. Longchamp, S. Magnenat, P. Rétonnaz, D. Burnier, G. Roulet, F. Vaussard, H. Bleuler, and F. Mondada. The MarXbot, a Miniature Mobile Robot Opening new Perspectives for the Collective-robotic Research. In *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010)*, pages 4187–4193. IEEE Press, Piscataway, NJ, USA, 2010.
- [3] A. L. Christensen, R. O’Grady, and M. Dorigo. SWARMORPH-script: a language for arbitrary morphology generation in self-assembling robots. *Swarm Intelligence*, 2(2–4):143–165, 2008.
- [4] C. Jones and M.J. Matarić. From local to global behavior in intelligent self-assembly. volume 1, pages 721–726. IEEE Computer Society Press, Los Alamitos, CA, 2003.
- [5] E. Klavins, R. Ghrist, and D. Lipsky. A grammatical approach to self-organizing robotic systems. *IEEE Transactions on Automatic Control*, 51(6):949–962, 2006.
- [6] W. Liu and A. Winfield. Autonomous morphogenesis in self-assembling robots using ir-based sensing and local communications. In *Proceedings of the 7th International Conference on Swarm Intelligence (ANTS 2010)*, pages 107–118. Springer, Berlin, Germany, 2010.
- [7] K. Støy and R. Nagpal. Self-reconfiguration using directed growth. In *Proceedings of the International Conference on Distributed Autonomous Robot Systems (DARS-04)*, pages 1–10. Springer, Berlin, Germany, 2004.
- [8] J. Bishop, S. Burden, E. Klavins, R. Kreisberg, W. Malone, N. Napp, and T. Nguyen. Programmable parts: A demonstration of the grammatical approach to self-organization. In *Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005)*, pages 2644–2651. IEEE Press, Piscataway, NJ, USA, 2005.
- [9] P. J. White, K. Kopanski, and H. Lipson. Stochastic self-reconfigurable cellular robotics. volume 3, pages 2888–2893. IEEE Computer Society Press, Los Alamitos, CA, 2004.
- [10] R. O’Grady, A. L. Christensen, and M. Dorigo. SWARMORPH: Multi-robot morphogenesis using directional self-assembly. *IEEE Transactions on Robotics*, 25(3):738–743, 2009.
- [11] H. Wei, D. Li, J. Tan, and T. Wang. The distributed control and experiments of directional self-assembly for modular swarm robots. In *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010)*, pages 4169–4174. IEEE Press, Piscataway, NJ, USA, 2010.
- [12] J.F. Roberts, T.S. Stirling, J-C. Zufferey, and D. Floreano. Quadrotor using minimal sensing for autonomous indoor flight. In *European Micro Air Vehicle Conference and Flight Competition, (EMAV 2007)*, 2007.
- [13] M. Bonani, S. Magnenat, P. Rétonnaz, and F. Mondada. The Handbot, a Robot Design for Simultaneous Climbing and Manipulation. In *Proceedings of the Second International Conference on Intelligent Robotics and Applications (ICIRA 2009)*, pages 11–22. Springer, Berlin, Germany, 2009.
- [14] K. Støy. Using situated communication in distributed autonomous mobile robotics. In *SCAI ’01: Proceedings of the Seventh Scandinavian Conference on Artificial Intelligence*, pages 44–52. IOS Press, Amsterdam, The Netherlands, 2001.
- [15] J.F. Roberts, T.S. Stirling, J-C. Zufferey, and D. Floreano. 2.5D infrared range and bearing system for collective robotics. In *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2009)*, pages 3659–3664. IEEE Press, Piscataway, NJ, USA, 2009.