Swarm Intell (2013) 7:83–113
DOI 10.1007/s11721-013-0082-y

Modular self-assembling and self-reconfiguring e-pucks

Lachlan Murray · Jon Timmis · Andy Tyrrell

Received: 30 October 2012 / Accepted: 16 April 2013 / Published online: 9 May 2013

Abstract In this paper, we present the design of a new structural extension for the e-puck mobile robot. The extension may be used to transform what is traditionally a swarm robotics platform into a self-reconfigurable modular robotic system. We introduce a modified version of a previously developed collective locomotion algorithm and present new experimental results across three different themes. We begin by investigating how the performance of the collective locomotion algorithm is affected by the size and shape of the robotic structures involved, examining structures containing up to nine modules. Without alteration to the underlying algorithm, we then analyse the implicit self-assembling and self-reconfiguring capabilities of the system and show that the novel use of 'virtual sensors' can significantly improve performance. Finally, by examining a form of environment driven self-reconfiguration, we observe the behaviour of the system in a more complex environment. We conclude that the modular e-puck extension represents a viable platform for investigating collective locomotion, self-assembly and self-reconfiguration.

Keywords Self-reconfigurable modular robotics · Swarm robotics · e-puck · Collective locomotion · Self-reconfiguration · Self-assembly

1 Introduction

The fields of swarm robotics and self-reconfigurable modular robotics are closely related. Both are situated within the larger field of autonomous mobile robotics and both concern the

Electronic supplementary material The online version of this article (doi:10.1007/s11721-013-0082-y) contains supplementary material, which is available to authorised users.

L. Murray · J. Timmis (✉) · A. Tyrrell
Intelligent Systems Group, Department of Electronics, The University of York, York, UK
e-mail: jon.timmis@york.ac.uk

L. Murray
e-mail: lachlan.murray@york.ac.uk

A. Tyrrell
e-mail: andy.tyrrell@york.ac.uk

problem of coordinating the behaviour of a group of relatively simple embodied agents. In the case of swarm robotics, this coordination is achieved in a distributed and self-organising manner, reliant entirely upon local sensing and communication (Şahin 2005). Though not exclusively, the control of modular robotic systems is also often performed in a distributed manner. The main difference between the two types of system lies in the fact that, unlike robotic swarms, the individual units of a modular robotic system may physically connect with one another to form larger robotic structures (Yim et al. 2007, 2009).

As noted by Beni (2005), whilst recalling the history of swarm robotics, the publication of early discussions on both swarm (Beni 1988) and self-reconfigurable robotics (Fukuda and Nakagawa 1988a) were coincident. Despite their similar origins and focus, the two fields differ significantly in one important aspect, namely, the accessibility of the robotic hardware that they require. Several swarm robotic platforms are available to buy, or have been released as open hardware projects (Mondada et al. 2009; Rubenstein et al. 2011; K-Team Corporation 2012; GCtronic 2012; Kernbach 2012; English et al. 2012; Pololu Corporation 2012). Whereas in contrast, the authors are aware of only two modular robotic platforms that are available commercially (Ryland and Cheng 2010; Schweikardt 2011) and only a single open source project (Zykov et al. 2007).

This disparity between the availability of swarm and modular robotic hardware may, at least partially, be attributed to the differing complexity of the required hardware. Swarm robots are purposefully simple units, whereas modular robots, although simple in comparison to the structures that they may form, require complex electrical and mechanical hardware to facilitate the processes of docking, reconfiguration and inter-robot communication.

In Murray et al. (2012), to help redress the balance between the accessibility of hardware in swarm and modular robotics research, we presented a low-cost open source extension for the e-puck robot (Mondada et al. 2009). The e-puck is a small mobile robot, equipped with basic sensors and actuators, including a colour camera, a ring of LEDs and an array of infrared proximity sensors. Our extension, hereafter referred to as the *modular e-puck extension*, consists of a plastic frame and four passive magnetic docking interfaces. The extension allows a group of e-pucks to physically join with one another, transforming what is traditionally a swarm robotics platform into a self-reconfigurable modular robotics system.

Murray et al. (2012) also introduced an algorithm for controlling the collective locomotion of a group of robots equipped with the modular e-puck extension. The algorithm, tested with groups of up to four e-pucks, was shown not only to be amenable to the task of collective locomotion, but, without alteration to the control algorithm, also demonstrated a novel form of implicit *self-assembly* that allows a group of robots to reform when broken apart.

In this paper, we extend our previous work with an alteration to the underlying control algorithm and significantly more experimental results. Specifically, we analyse in more detail the properties of the algorithm which lead to the exhibition of self-assembly. We also investigate how using structures of different shapes and sizes affects the performance of the system. Finally, by combining self-assembly, collective locomotion and self-disassembly, we investigate the task of self-reconfiguration under changing environmental conditions.

The remainder of this paper is structured as follows. In Sect. 2, we review similar existing modular robotic hardware and related approaches to the control of such systems. In Sect. 3, we introduce the modular e-puck extension. In Sect. 4, we describe the collective locomotion algorithm. In Sect. 5, we detail the common setup used during our experiments. In Sect. 6, we present the results of our collective locomotion experiments. In Sect. 7, we analyse the self-assembling properties of our system. In Sect. 8, we investigate a form of environment driven self-reconfiguration. Finally, in Sect. 9, we provide a summary and discuss potential areas of future work.

2 Related work

In 2007, Yim et al. (2007) produced a comprehensive review of the field of self-reconfigurable modular robotics. The review includes a 'taxonomy of architectures' which classifies platforms as either: *chain*, *lattice*, *mobile*, or if they combine elements of more than one of the previous three, *hybrid*. In chain-based architectures, modules are connected to one another in series but may branch to form tree-like structures or fold and reconnect to form loops. Lattice architectures are more restrictive than chain-based systems, with modules only able to occupy discrete positions within a conceptual grid. In a mobile architecture, as well as being able to form collective robotic structures, modules are able to move freely around their environment as individuals.

The modular e-puck extension can be described as a mobile-lattice platform. Each module is independently mobile but may connect with up to four other neighbours within a two-dimensional (2D) grid. Over the last 25 years, a large number of self-reconfigurable robots have been developed. A number of different connection mechanisms have been designed for physically joining modules, and a variety of different methods have been proposed for controlling reconfiguration. In this section, we review several of the most relevant mobile and lattice platforms from the field of self-reconfigurable modular robotics, highlighting the similarities and differences between these systems and the modular e-puck extension. We also review some of the previous approaches to controlling the locomotion, assembly, and reconfiguration of such systems.

2.1 Platforms

Most early work into lattice-based modular robotics focused on 2D systems. The *Fracta* modules of Murata et al. (1994) and the *Metamorphic* robots described by Pamecha et al. (1996) are two such examples. In both systems, the individual modules occupy a hexagonal lattice. Whilst the robots themselves are not independently mobile, by altering their connections with neighbouring units, they are still capable of reconfiguration. The *Programmable Parts* of Bishop et al. (2005) are another example of a 2D-lattice system. Like the modular e-puck extension, the Programmable Parts do not need to be physically connected to their neighbours in order to reconfigure. However, since their motion is controlled stochastically by underlying jets of air, it cannot be said that they are independently mobile.

The first truly mobile self-reconfigurable modular robotic system was the *CEBOT* (cellular structured robot) platform of Fukuda and Nakagawa (1988b), Fukuda et al. (1988). Despite Fukuda and Nakagawa's pioneering work, the concept of a mobile self-reconfigurable platform was largely ignored in early modular robotics research, with most researchers tending to focus on chain and lattice based systems. In recent years, however, a number of different mobile platforms have emerged. This resurgence was led by the *s-bot* platform (Mondada et al. 2004) from the Swarm-bot project[1] (Dorigo et al. 2006), and the *foot-*, *hand-* and *eye-* bots (Dorigo et al. 2013) of the subsequent Swarmanoid project.[2] Each s-bot possesses: a combined track and wheel ('treel') differential drive system, an array of sensors, an omnidirectional camera and grippers for physically connecting with neighbouring modules. As an evolution of the s-bot, the foot-bot possesses a similar set of sensors and actuators, but housed in a more robust and slightly larger platform. The hand- and eye-bots are designed

[1] http://www.swarm-bots.org/.

[2] http://www.swarmanoid.org/.

to supplement the foot-bot with additional sensors and actuators, resulting in a highly flexible heterogeneous swarm. Although unable to create arrangements as complex as those produced by other modular robotic systems, the hand- and foot-bots may still physically connect to form simple 2D structures.

The robots being developed as part of the SYMBRION[3] and REPLICATOR[4] projects (Kernbach et al. 2010) represent another good example of a heterogeneous self-reconfigurable modular robotic system. Like the modules from the Swarm-bot and Swarmanoid projects, the individual robots are independently mobile. However, unlike the s-bot and its derivatives, the modules are also designed to be capable of forming complex 3D structures.

The *Distributed Flight Array* (DFA) of (Oung and D'Andrea 2011) represents a very different type of self-reconfigurable system. The DFA modules are hexagonal and measure approximately 25 cm in length. At the centre of each module is a single 3-blade propeller and surrounding it are three omnidirectional wheels. The modules may move along the floor as individuals, but when connected, form a multi-rotor vehicle that is capable of autonomous flight.

Two other examples of mobile systems are Wei et al.'s (2011) *Sambot* and Hong et al.'s (2011) *X-Cell*. Both systems approximate cubes with 8 cm sides, and both are made mobile by a two wheeled differential drive system. The X-Cell modules consist of two parts, a main unit which houses the core electronics and battery, and a square faced shell, upon which sensors and docking connectors are mounted. A servo motor allows the shell to be rotated independently of the base, in the range of 0–180°. The Sambot modules possess four passive docking sides and one active side, mounted on a rotating arm. In contrast to X-Cell, which may only form structures within a 2D grid, the rotating arm of the Sambots allows them to form chain-like structures.

One of the most important factors in the design of a self-reconfigurable modular robotic system is the method by which the individual robots connect with one another. Over the years, several different methods have been proposed for connecting mobile and lattice modular robots. Systems such as the SYMBRION and REPLICATOR, Swarm-bot and Swarmanoid, Metamorphic, ATRON (Jørgensen et al. 2004) and Sambot robots rely on purely mechanical connection mechanisms. Many other systems favour a magnetic approach. The Programmable Parts, DFA, Miche (Gilpin et al. 2007, 2008) and Telecube (Suh et al. 2002) systems all use permanent magnets. The X-Cell, Catom (Goldstein et al. 2005) and Smart Pebble systems (Gilpin et al. 2010; Gilpin and Rus 2010), on the other hand, all favour electromagnets, whilst the Fracta system utilises a combination of permanent magnets and electromagnets. The modular e-puck extension uses only permanent magnets. Mirroring the approach of the DFA platform, the strength and positioning of the magnets on the modular e-puck extension were chosen such that the connection between two modules is stable when it needs to be, but not so strong that the modules are prevented from disconnecting when it is desired.

In terms of their structure, robots equipped with the modular e-puck extension closely resemble the X-Cell modules. Both systems possess a separate square frame that rests on top of their main body, allowing them to form stable structures within a 2D lattice, and both systems utilise a two-wheeled differential drive system. However, we can identify two important differences between the X-Cell modules and robots equipped with the modular e-puck extension. Firstly, the positioning of the electromagnets on the X-Cell shell, and the

[3] http://www.symbrion.eu/.

[4] http://www.replicators.eu/.

provision that no two electromagnets may connect with one another, limits the number of possible configurations that the robots may form. With the modular e-puck extension, there are no limitations upon the number of configurations that the robots can be arranged in—other than those imposed by the lattice grid. Secondly, the rotation of the X-Cell shell is controlled by a servo motor and may only move in the range of 0–180°. The frame of the modular e-puck extension, on the other hand, is completely independent of the e-puck robot, allowing for infinite rotation about the robot's body. This leads to a far more flexible system, albeit at the expense of controllability.

2.2 Control strategies

An important requirement of locomotion in collective robotic systems is that the individual robots reach a consensus regarding the goal of the collective. This may be achieved in a centralised fashion, whereby a designated leader propagates commands to the other modules, or in a decentralised manner, in which no single robot is responsible. Owing to the perceived lack of a 'single point of failure', decentralised methods of control are often favoured.

Gutiérrez et al. (2009a) describe a decentralised method for synchronising the headings of a group of stationary e-pucks. Gutiérrez et al.'s approach relies on exchanging relative bearings using infrared (IR) communication. In related work, Baldassarre et al. (2007) evolve controllers for the s-bot platform which use information from the robots' traction sensors to synchronise the headings of a group of mobile robots. Using a similar approach, Trianni et al. (2006) incorporate information from ground sensors and demonstrate coordinated motion with hole avoidance. Baldassarre et al.'s controllers were verified on real robots, under various conditions, and although it was not their main focus, the authors also demonstrate the robot's ability to complete the related task of collective transport.

In an another example that uses the s-bot's traction sensor, Groß et al. (2006b) describe a solution to the collective transport task that allows a group of robots to efficiently transport an object to a goal location, even if one or more of the robots cannot perceive the goal. Groß and Dorigo (2009) evolve controllers for a similar task but start with the robots in a disconnected state, requiring that the robots first self-assemble before performing collective transport. Ferrante et al. (2013) and Campo et al. (2006) describe two hand-coded solutions to the collective transport task, both of which include explicit negotiation between robots. Ferrante et al.'s approach relies on infrared communication, whilst Campo et al. utilise the robots' LEDs and vision systems.

Our approach to collective locomotion builds upon that of Gutiérrez et al. (2009a). Like Gutiérrez et al., we use IR communication, and like both Campo et al. (2006) and Ferrante et al. (2013), we use explicit negotiation. Unlike Groß et al. (2006b), Trianni et al. (2006) and Baldassarre et al. (2007), in the absence of a traction sensor, the robots must determine their heading based solely on their perception of the environment and communication with their neighbours.

In Groß and Dorigo (2008), the authors review several different types of self-assembling systems and classify them as either self-propelled or externally-propelled, depending upon the manner in which the individual modules move. Due to the independent mobility of the e-puck, in Groß and Dorigo's taxonomy, the modular e-puck extension may be viewed as a self-propelled system.

Many of the systems reviewed in Sect. 2.1 are capable of self-propelled assembly. The CEBOT, X-Cell and Sambot platforms, as well as the robots from the SYMBRION and REPLICATOR projects, have all demonstrated alignment and docking approaches based upon the use of infrared sensors (Fukuda et al. 1988; Wei et al. 2011; Liu and Winfield 2010;

Hong et al. 2011). The s-bot platform, on the other hand, has been used to develop several distributed methods of self-assembly (Groß et al. 2006a; O'Grady et al. 2009) and self-reconfiguration (O'Grady et al. 2008) that make use of the robots' LEDs and vision systems. O'Grady et al. (2009), for example, describe a distributed approach to self-assembly in which structures are grown by recruiting new modules to specific locations, based upon local interaction rules. A common property of these approaches is that they rely on the presence of a seed module to initiate self-assembly, and in most cases (O'Grady et al. 2009 excepted), require a target shape to guide the assembly process. In contrast, the approach used by the DFA system (Oung and D'Andrea 2011) requires neither a seed nor a target. The DFA modules simply move around the arena until they collide with other robots, at which point permanent magnets and protruding features of the robot's chassis cause them to join together. Since it negates the need for complex alignment procedures, this form of random passive docking—which is more commonly found in externally-propelled forms of self-assembly—can be beneficial.

The approach to self-assembly described in this paper resembles that of the DFA platform. Self-assembly is not directed and does not require a seed, however, unlike the DFA approach, communication between nearby robots does produce a tendency for modules to move towards one another.

In other related work, Groß et al. (2011) describe a system of heterogeneous modules that are capable of responding to changes in environmental conditions, whilst demonstrating self-replication and externally-propelled self-assembly. Two similar tasks are described by Trianni et al. (2004) and Weel et al. (2012), in which controllers are evolved to perform self-assembly in response to environmental triggers. Inspired by this work, we use a similar task to investigate self-reconfiguration with the modular e-puck extension.

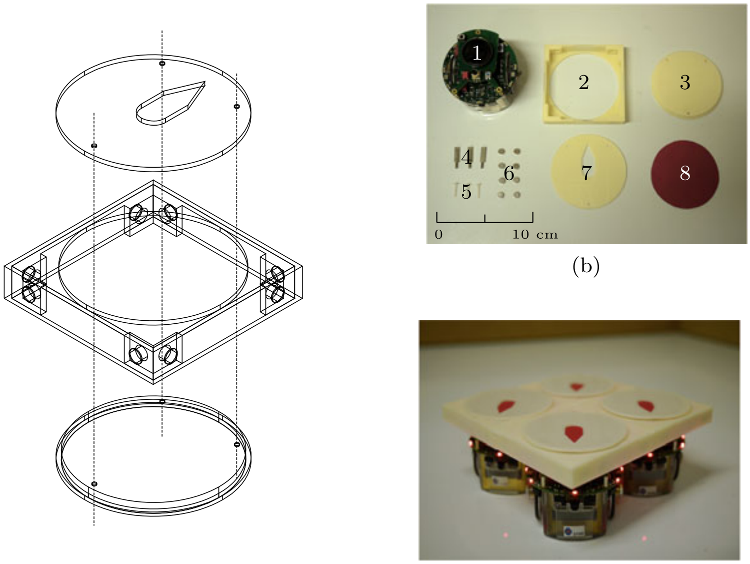
2.3 Summary

According to the classification of Yim et al. (2007), the modular e-puck extension may be described as a mobile-lattice modular robotic system. Although, structurally, robots equipped with the extension resemble the X-Cell modules of Hong et al. (2011), they are far more flexible in terms of the different configurations they may form. Like the Programmable Parts, DFA, Miche and Telecube systems, the extension uses a connection mechanism based upon permanent magnets, and like the DFA system, the strength of the magnets was carefully chosen to allow for both self-assembly and self-reconfiguration.

The approach to collective locomotion described in this paper builds upon that of Gutiérrez et al. (2009a), and uses explicit negotiation in a manner similar to that of Campo et al. (2006) and Ferrante et al. (2013). The approach to self-assembly places the system within the 'self-propelled' category of Groß and Dorigo's (2008) taxonomy, but unlike many other such systems, the behaviour is not directed, and does not require a seed module. Neither, on the other hand, is it fully stochastic like the approach employed by the DFA platform.

3 Modular e-puck extension

As an open hardware project, the e-puck robot is very well suited to modification. Over recent years, a number of extensions have been developed, including an omnidirectional vision turret (Mondada et al. 2009), a range-bearing board (Gutiérrez et al. 2009a), colour LEDs, a ZigBee radio module (Cianci et al. 2007), and even an embedded Linux implementation (Liu and Winfield 2011).



(a) (b) (c)

Fig. 1 A schematic of the main structural components of the modular e-puck extension (**a**) and photographs of unassembled and assembled prototypes (**b**)–(**c**). The designs for the structural components are made freely available online (http://www.elec.york.ac.uk/research/projects/Modular_e_puck_Extension.html)

In this section, we describe an extension that may be used to transform the existing e-puck platform into what can be described as a mobile-lattice modular robotic system. Robots equipped with the extension remain independently mobile, but through passive magnetic docking interfaces may physically connect with other modules within a 2D grid. Whilst not possessing the same functionality as some of the more sophisticated modular robotic systems, the platform may serve as an entry point, or stepping stone, to more advanced work. Providing a low-cost alternative for investigating the interesting properties of self-reconfigurable modular robotic systems, at an abstract level.

As shown in Fig. 1a, the extension consists of three parts: a circular base plate which sits directly on top of the e-puck, a central frame which rests on top of the base plate, and a second circular cover which sits on top of the frame to help secure the extension together.

The base plate is positioned on top of the default extension board using three 15 mm hexagonal spacers. To remain compatible with other extension boards, larger spacers may be used. A small overhang on the base plate allows the inner ring of the central frame to rest on the base plate, without being permanently attached. This lip allows the frame to rotate unhindered around the central axis of the e-puck. To enable separate modules to connect with one another, two magnets are fitted on each internal edge of the central frame, with opposing poles facing outwards. The strength and positioning of the magnets were chosen such that, if connected modules coordinate their motion, they will remain attached, but if they do not, they will break apart. This approach ensures that the extension provides a suitable platform for investigating both collective behaviour and self-reconfiguration. Screws which pass through the two circular plates secure the extension to the e-puck and an arrow shaped window in the top cover allows the current heading of the robot to be easily recognised. To aid computer tracking, a piece of coloured card may be positioned between the base plate and the top cover.

Table 1 The list of parts required to construct a single e-puck extension

Part	Label	Description	Quantity	Cost (€)
e-puck	1	The e-puck mobile robot	1	–
Structure	2, 3, 7	3D-printed ABS plastic, ∼25 g	1	0.90
Spacers	4	Brass/nickel stand-off pillars, M2	3	0.50
Screws	5	Nylon set screw, slotted, M2, 12 mm	3	0.20
Magnets	6	Neodymium N38 disc magnets, 6×2 mm	8	4.00
Card	8	Coloured card for tracking (optional)	1	–
			Total	5.60

The parts required to construct a single extension are listed in Table 1, and labelled in Fig. 1b. Figure 1c shows a potential arrangement of four e-pucks equipped with the fully assembled extension. The three main structural parts were fabricated using a MakerBot thing-o-matic 3D printer.[5] We estimate the total expense to be around €5 per unit, with the magnets taking up the majority of the cost. The total weight of the extension is <50 g.

4 Algorithm description

In this section, we describe an algorithm for controlling the collective locomotion of a group of e-pucks equipped with the modular e-puck extension. Through a behaviour-based approach, every robot in the group is motivated to move forward, to align with its neighbours and to avoid obstacles. The summation of these three objectives determines the speed of the robot's motors. Regardless of their position within the larger structure, each robot runs the same controller and exchanges information only via local communication.

Figure 2 shows a high level overview of the controller. The parts of the controller corresponding to the alignment, obstacle avoidance and forward motion of the robots are labelled. The inputs $h_0, h_1, \ldots, h_N$ represent the relative headings of a robot's neighbours and the positions of nearby obstacles. The outputs M_L and M_R correspond to the speed of the robot's left and right motors. The content of boxes (1), (2) and (3) corresponds to equations of the same name and will be introduced in the forthcoming sections.

As a collective, the robots are able to exhibit continuous coordinated motion within an enclosed arena, whilst at the same time demonstrating robustness to perturbations in their overall structure. Following the removal of one or more modules from a group, whether deliberate or accidental, the system is able to reconfigure and re-form either the original structure, or an entirely new one. This process of self-assembly is not pre-programmed but emerges due to a combination of factors including: the design of the structural extension, the design of the locomotion controller, and the nature of the robots' environment.

The controller makes extensive use of the e-puck's infrared (IR) sensors. The arrangement of the eight sensors on a single e-puck is shown in Fig. 3a, the points marked 'i', 'j' and 'k' correspond to blind spots at which no sensors are present. The obstacle avoidance behaviour uses the IR sensors for proximity detection whilst, with the help of the *LibIrcom* library (Gutiérrez et al. 2009b), the alignment behaviour uses them for short-range communication.

[5] http://www.makerbot.com/support/thingomatic/.

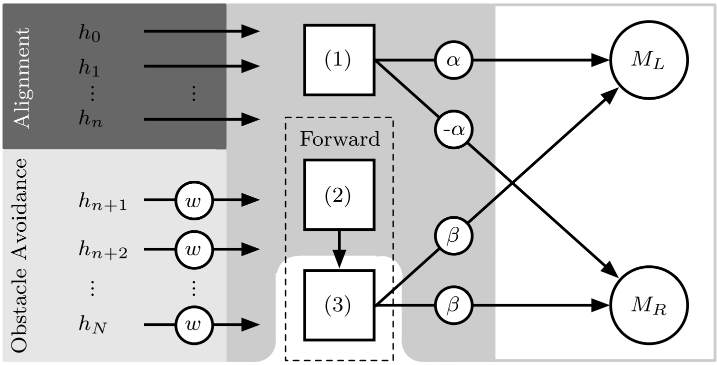


Fig. 2 A high-level overview of the collective locomotion controller. For clarity, details of the inter-robot communication components are omitted. The alignment behaviour is introduced in Sects. 4.1–4.2, the obstacle avoidance behaviour is introduced in Sect. 4.3 and the forward behaviour is introduced in Sect. 4.4

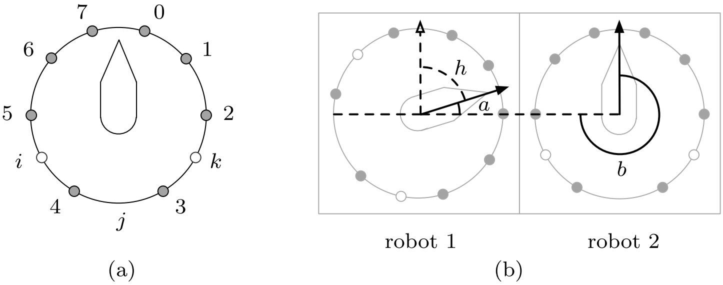
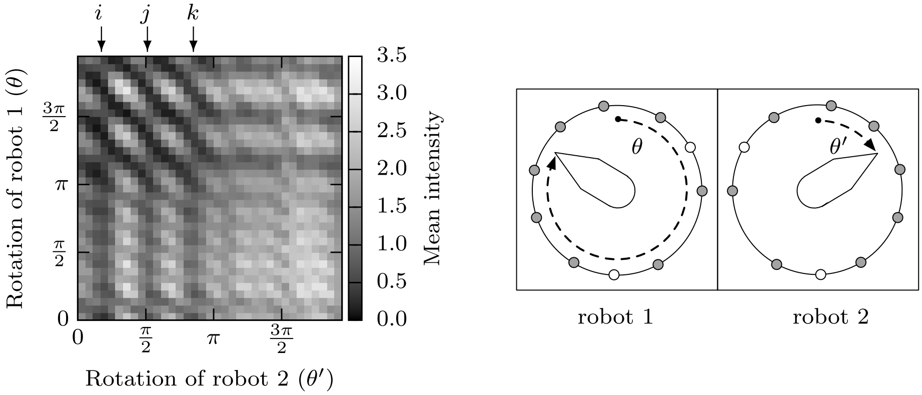


Fig. 3 The positioning of the infrared sensors on board an e-puck robot (**a**) and the mechanism for exchanging relative bearings between two modules (**b**)

The alignment behaviour is based upon the same principle of exchanging relative bearings as both the *LibIrcom* library's 'synchronise' example, and the alignment technique described by Gutiérrez et al. (2009a). We describe this method of alignment shortly, hereafter referred to as *static synchronisation*, due to the fact that the robots remain stationary when executing this behaviour. We then identify some problems with this approach when considering non-stationary alignment and whilst introducing a new alignment behaviour, propose some solutions. Following which, we introduce the obstacle avoidance behaviour and describe how the two parts are combined with a forward bias, as shown in Fig. 2, to produce the desired overall locomotion.

4.1 Static synchronisation

By exchanging relative bearings with one another, the static synchronisation approach allows a group of stationary robots to converge to and maintain a common heading. Every robot broadcasts a unique ID and listens for the IDs of others. Based upon the sensor at which a message is received, robots are able to estimate the position of their neighbours as an angle relative to their own heading. For every ID that a robot receives, a message is sent to the corresponding neighbour, notifying it of the angle at which it was detected. As shown in Fig. 3b, using the angle at which robot 2 was detected 'a', and the angle at which robot 2



(a) (b)

Fig. 4 An intensity heatmap of messages sent between two modules at various orientations (**a**) and a diagram of the setup used to gather the data (**b**)

detected robot 1 ‘b’, robot 1 may calculate the relative heading of robot 2 as $h = a + \pi - b$. The relative heading of each of a robot’s neighbours is used to incrementally update the robot’s own desired heading, which consequently determines whether a robot should turn left, turn right, or remain stationary at each control cycle.

The approach is effective at synchronising the alignment of stationary modules, but in preliminary experiments with mobile robots, we observed that if the robots were not able to converge to the same heading quickly enough, they had a tendency to break apart. We identify two potential causes of this problem, both of which relate to the positioning of the sensors on the e-puck.

Firstly, because the angle between neighbouring sensors ranges from around 30° to 60°, unless two sensors are perfectly aligned, the estimate of angles a and b is often inaccurate. Although the static synchronisation approach incorporates mechanisms for reducing this uncertainty, it is still present.

The second problem arises due to the large gaps between sensors 2, 3, 4 and 5. When two robots are connected, the close proximity of the modules and the large gaps between the sensors can create blind spots in some orientations (marked i, j and k in Fig. 3a). As a result of these blind spots, in certain configurations, the time taken to converge to a common heading is increased.

These two problems are further highlighted in Fig. 4. When sending messages via infrared, it is possible to estimate the distance between the sending and receiving sensors by measuring the intensity of the light received. Figure 4a maps the intensity of the IR signal for messages sent between two robots, arranged at various orientations. The setup used to gather this data is shown in Fig. 4b, where robot 1 is the receiving module and robot 2 is the sender. The intensity of the signal associated with five received messages was recorded at 10° intervals for every 1296 (36 × 36) possible configurations of the two robots. Where no message was received within a certain time limit, an intensity of 0 was assigned. The mean value of the five measurements is plotted.

It can be noted from Fig. 4a that, due to the distribution of the sensors, when the two robots are facing each other (bottom right) the intensity of the received signals is high, but when two robots are facing away from each other (top left) the intensity is often low. A high intensity value indicates that the sending and receiving sensors are closely aligned, so when

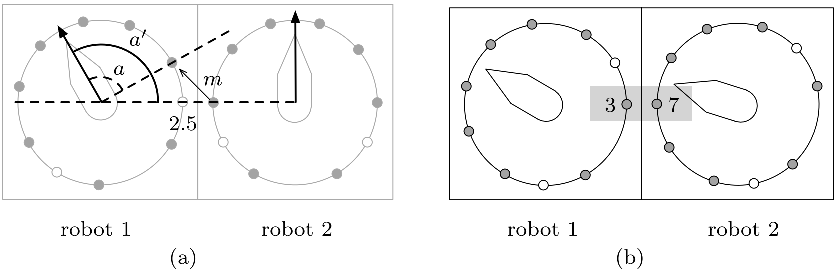


Fig. 5 Diagrams showing the strategy for correcting the misalignment of two connected robots using both ‘virtual’ sensors (**a**) and paired sensors (**b**)

two robots are facing each other the measurement of angles a and b is likely to be more accurate than when they are facing away.

4.2 Sensor-aware alignment

Our new approach to alignment aims to tackle the problems identified in the previous section, through greater awareness of the positioning and behaviour of the e-puck’s sensors. Like the static synchronisation approach, robots still communicate with their neighbours and track the relative orientation of other modules. However, as well as making use of the content and direction of the messages they receive, the signal intensity also influences their behaviour.

In Fig. 4a, the lines at $x = i$, $x = j$ and $x = k$ correspond respectively to the configurations at which the blind spots i, j and k of robot 2 are directly aligned with robot 1. As shown in Fig. 4a, the intensity values of the messages received along and adjacent to the lines i, j and k are low. It is possible to make use of this fact to infer when the blind spot of a robot is aligned with its neighbour, and hence to determine the position of the neighbour more accurately. Specifically, as shown in Fig. 5a, when the message m received at sensor 2 reports a low intensity, we may infer that blind spot k of robot 1 is facing robot 2. Whilst it is true that sensor 2 will also report low intensity values when the point between sensors 1 and 2 is aligned with robot 2, because this gap is smaller, these values will never drop as far they do in blind spot k.

A similar inference can be applied to blind spot i and its relation to sensor 5. Notionally we may define two *virtual* sensors ‘2.5’ and ‘4.5’ which lie between sensors 2–3 and 4–5, respectively. As shown in Fig. 5a, if a message is detected at sensor 2.5, rather than assume it to have originated from a point at an angle a, we may more accurately assume that it originated from an angle a', halfway between sensors 2 and 3. Note that it is not possible to define a virtual sensor ‘3.5’ which lies between sensors 3 and 4, because from the perspective of these sensors, the blind spots i, j and k are indistinguishable. In Fig. 3a, the virtual sensors are represented by the white circles at points i and k.

It should be noted that, using intensity values alone, it is difficult for a robot to differentiate between scenarios in which its own blind spot is facing its neighbour, its neighbour’s blind spot is facing it, or both blind spots are facing each other. This is not a major concern, however, since in either scenario the reaction is the same, the robots will turn towards each other. Furthermore, we may note that messages received from neighbours that are not directly connected, i.e. neighbours positioned at a diagonal, will always have lower intensity values. Since the *LibIrcom* library preferentially processes high intensity messages, the proportion of messages received from indirect neighbours, and thus the influence they exert, will be lower than that of direct neighbours. In the worst case scenario, robots will over eagerly turn towards each other, but as we suggest in Sect. 7, this is not always detrimental.

To further improve the time taken for the robots to converge upon a common heading, we also implement a new method for translating the relative headings of neighbours into motor commands. We begin by calculating the average heading of all of the robot's most recently detected neighbours. This value (η) is calculated using Eq. (1) and fits into box (1) of Fig. 2, where $(h_i, h_{i+1}, \ldots, h_n)$ are the relative headings of the robots neighbours, $k = n$ is the number of neighbours and γ_i is a weight, which in this case is set to 1:

$$\eta = \arctan 2\left(\sum_{i=0}^{i=k} \sin(h_i) \cdot \gamma_i, \sum_{i=0}^{i=k} \cos(h_i) \cdot \gamma_i\right). \tag{1}$$

The average heading η, which will always belong to the range $-\pi < \eta \leq \pi$, is used to determine the speed of the robot's motors. As shown in Fig. 2, before being applied to the robot's motors, η is multiplied by a synchronisation parameter α, the larger this value, the more aggressively the robots will attempt to turn. For the motor on the right (M_R), the α parameter is inverted, this means that for values of $\eta < 0$ the robot will turn left and for values of $\eta > 0$ will turn right. For values of $\eta = 0$ and for control cycles in which no messages are received, the turning speed of the robot's motors is set to zero.

In communicating the relative angle at which a neighbour was detected, robots transmit the number of the sensor, rather than the angle itself. Furthermore, if $|\eta| > \frac{\pi}{2}$, to pre-empt the fact that the robot is about to make a fast turn, the sensor number that is transmitted is incremented or decremented by one —depending upon whether the robot is turning left or right.

Based upon the knowledge that a high intensity signal is indicative of a close alignment between two sensors, we also define sensor pairings which, when the intensity of the signal is high, should not influence the movement of the robots. For example, in Fig. 5b, if robot 1 receives a high intensity message on sensor 3, that was sent from sensor 7 of robot 2, the relative heading of robot 2 will be set to 0. Note that although the alignment between robots 1 and 2 in this scenario is not perfect, it is considered 'good enough' for the task at hand, and preferential to the robots continuously changing direction.

4.3 Obstacle avoidance

Every sensor that has not received a message within the last few seconds, and does not neighbour with a sensor that has received a message, contributes to obstacle avoidance. Each contributing sensor which detects an obstacle closer than the avoidance threshold τ creates a new desired heading in the opposing direction to the obstacle. The distance to the detected object is used to assign a weight w_i in the range (0, 1) to each of these new headings. The closer the obstacle is, the larger the weight. These headings are added to the relative headings of the robots neighbour's using Eq. (1) with γ_i set to w_i and k set to N, where N is the total number of headings from both the obstacle avoidance and alignment behaviours. As described in Sect. 4.2, the average heading η is used to determine the speed of the robot's motors.

This approach is equivalent to assuming that there is a neighbouring robot facing every sensor that has detected an obstacle. The robots attempt to align with these imaginary neighbours in the same way that they align with real robots. For example, if an obstacle was detected on the left hand side of a robot, the robot would react as if there was another module directly facing this side, it would attempt to align with it, and in doing so would turn right, therefore avoiding the obstacle. The only difference between the robots reaction to obstacles and neighbouring robots, is that headings derived from the presence of obstacles are weighted according to how far away they are, with closer obstacles exerting a greater influence over alignment.

4.4 Forward bias

In preliminary experiments, to ensure that the robots always moved forward, we added a small positive bias to the speed of their motors. For structures containing only a few robots, this bias, combined with the obstacle avoidance behaviour described above, was sufficient to ensure that the robots remained synchronised whilst still obtaining good coverage of the arena. However, when the number of robots was increased, a more adaptive approach was required.

When we first increased the number of modules in our experiments, we observed that the forward motion of the robots could have a negative effect on the robot's alignment. In an unsynchronised structure, it is not desirable for all of the individual robots to attempt to move forwards. The forward motion of the robots reduces their turning rates, which results in an increase in synchronisation time. Furthermore, in a structure containing many modules with different headings, the forward motion of the individuals increases the risk of the group breaking apart.

To handle this problem, we modified the algorithm so that the robots' forward bias is scaled in proportion to how well they believe themselves to be aligned —based upon the known relative headings of their neighbours. Using Eq. (2), we calculate a *synchronisation score* s, by taking the average magnitude of the N headings $(h_i, h_{i+1}, \dots, h_N)$ provided by the obstacle avoidance and alignment behaviours, this value will be low for well aligned robots and high otherwise. The score is then scaled using function $B(s)$ (see Eq. (3)) and multiplied by the forward speed parameter β to determine how much bias to add to the speed of the robots' motors.

$$s = \frac{1}{N} \sum_{i=0}^{i=N} |h_i|, \tag{2}$$

$$B(s) = \begin{cases} 0 & \text{if } s \geq \pi, \\ \frac{\pi - s}{\pi} & \text{if } s < \pi. \end{cases} \tag{3}$$

Robots with a low score, i.e. those which are well aligned, are consequently assigned a larger forward bias than those with a high score. In a poorly synchronised structure, the scores will be high, meaning the forward motion will be less disruptive and the robot's rotary motion will dominate, leading to faster alignment and reducing the chances of the structure breaking apart.

5 Experimental setup

The experiments detailed in the upcoming sections were all conducted using the same setup. Every experiment was performed within a small table-top arena, measuring approximately 86 cm × 56 cm. The arena, shown in Fig. 6a, is surrounded by a ring of white LEDs; however, in this work, the LEDs were only used during the final experiment of Sect. 8. The base of

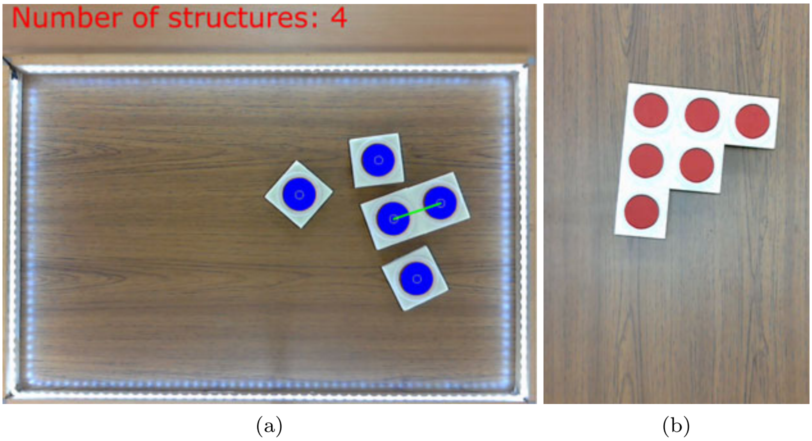


Fig. 6 *Top down* views of the arena used for the experiments, both with (**a**) and without (**b**) information from the computer tracking software superimposed

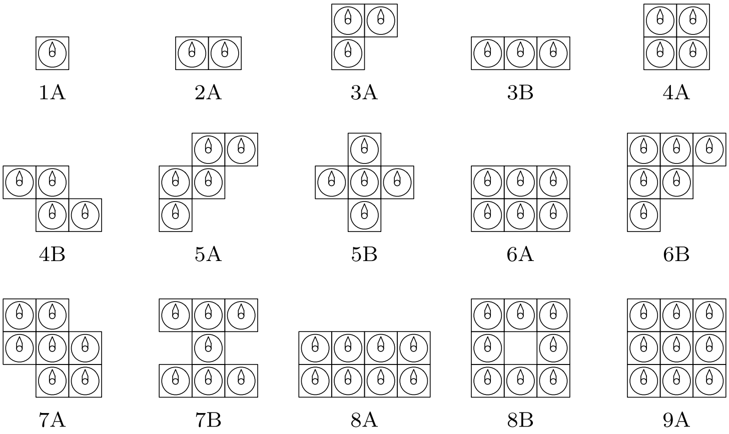


Fig. 7 The various different module configurations used in the experiments

the arena is wooden, which provides good traction for the e-puck's rubber coated wheels, but is not so resistive that it will completely prevent robots from 'skidding' if lateral forces are applied. An overhead camera was used to record the experiments and tracking software was used to extract positional and connectivity information from the scene. Some of the output from the tracking software is shown in Fig. 6a, the blue circles highlight the robots currently being tracked and the green links signify which robots are connected. The number of independent structures detected is shown in the top left corner of the view.

The experiments reported in this paper may be split into two classes, those in which the robots start in a fully connected state and those in which they start disconnected. Section 6, and the first part of Sect. 8, cover those of the former class, whilst Sect. 7, and the remainder of Sect. 8, handle the latter. The 15 starting configurations that were used for the first class of experiments are pictured and labelled in Fig. 7. In the experiments where the robots began disconnected, the modules were evenly distributed within the arena and their headings

Table 2 The two parameter sets used in the experiments

Parameter	Description	Value	
		A	B
α	Synchronisation weight	100	125
β	Forward speed weight	20	50
τ	Avoidance threshold	30	30

randomised at the start of each run. In the experiments where the robots began connected, the structure was always placed in the centre of the arena at the same orientation, but the underlying robots were aligned and rotated at 10 different starting angles, each offset by 36°. In total, 280 independent runs were performed and ~45 hours worth of data was collected.

The parameters used in the experiments are displayed in Table 2. Parameter set ‘A’ was used only for the stationary alignment experiments reported in Sect. 6.1. Following these experiments, the parameters were tweaked to produce set ‘B’, which was then used for the remainder of the experiments. The values were chosen by trial and error. Once satisfactory behaviour was observed, no further attempt was made to optimise the parameters.

6 Collective locomotion

In this section, we investigate the task of collective locomotion. In order to assess the effectiveness of the alignment behaviour introduced in Sect. 4.2, we begin by examining the sub-task of stationary alignment. We then move on to investigate the full collective locomotion task.

6.1 Stationary alignment results

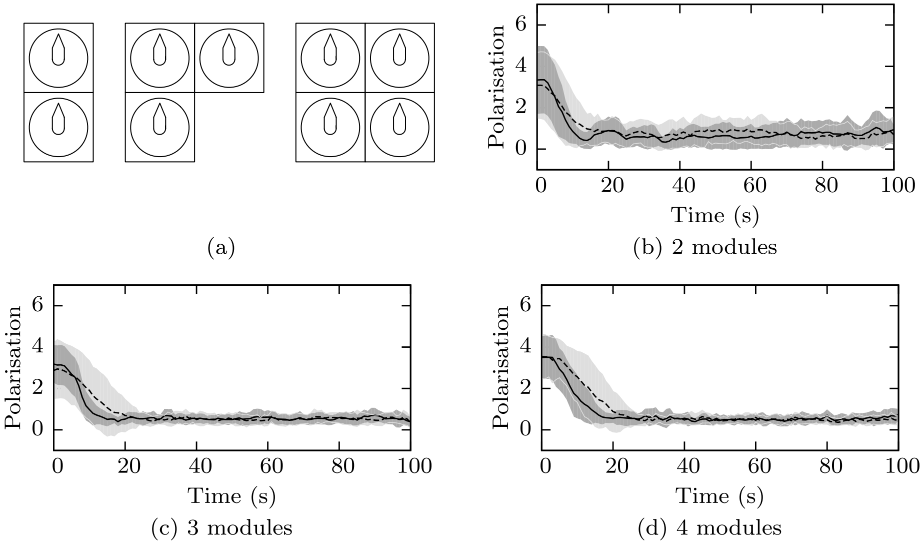
In this set of experiments, we compare our modified alignment behaviour with that of the original static synchronisation strategy from the *LibIrcom* library. Experiments were conducted using groups of two, three and four stationary robots, arranged as shown in Fig. 8a.

For each controller, and each of the three arrangements, 20 individual runs were conducted. The orientation of the robots was randomised at the start of each run and the absolute heading of each robot was recorded at one second intervals over a period of 100 seconds.

To assess the robots’ ability to converge towards a common heading, we use the same polarisation metric as (Gutiérrez et al. 2009a). As shown by (4), the polarisation P of a group of robots G is defined as the sum of the distance between the heading of every robot and its angular nearest neighbour θ_{ann}:

$$P(G) = \sum_{i \in G} \theta_{\mathrm{ann}}(i). \qquad (4)$$

Figures 8b–d plot the mean polarisation of the two approaches, for each of the three module configurations. As is evident by the eventual low polarisation values in all of the figures, in every experimental run, the modules were observed to converge to and maintain a common heading. In comparing the two approaches, using the Wilcoxon rank-sum test with a significance level of 0.05, we can say that there is no significant difference in polarisation over the final 20 seconds of each experiment. However, in every configuration, we can observe that convergence is faster for the experiments utilising the new approach to alignment. Furthermore, during the convergence phase, the variance in the polarisation of the static synchronisation approach is greater.



(a)

(b) 2 modules

(c) 3 modules

(d) 4 modules

Fig. 8 Graphs (**b**)–(**d**) plot the mean polarisation ± one standard deviation, for each of the three configurations in (**a**). The static synchronisation approach is represented by the *dashed line* and the *lighter grey region*, and our new approach is represented by the *solid line* and the *darker region*

6.2 Collective locomotion results

With the integration of the obstacle avoidance and forward bias behaviours, we now investigate the task of controlling the collective locomotion of a group of mobile robots. For each of the 15 starting configurations shown in Fig. 7, we performed 10 independent runs, each lasting 10 minutes. If one or more modules became disconnected from a structure, the run was terminated. As reported in the following section, 29 out of 150 runs ended in this manner. In every run, the positions of the individual modules, the number of structures, and the number of modules in each structure was recorded. We use this information to calculate the *success rate*, *coverage*, *cohesion* and *speed* for each of the different configurations. The results of these experiments summarised in Table 3, and analysed in more detail in the remainder of this section.

Following on from these experiments, we examine the *scalability* of our approach by considering linear structures of between one and six modules.

6.2.1 Success rate

We define 'success' as the ability of a structure to navigate the arena for the full 10 minute duration of a run, whilst all of the modules remain connected to one another in the same configuration in which they started. To account for any inaccuracies in the tracking software, and to allow for minor breaks that are quickly repaired, we only acknowledge a break in the link between two modules if it exists for more than one second. Our results show that our approach is successful in the vast majority of scenarios, even with structures containing up to nine modules. Out of 150 runs, across all 15 configurations, we found that the robots were successful on 121 occasions and that 10 of the 15 configurations were successful 100 % of the time.

Table 3 A summary of the results from the collective locomotion experiments. From left to right, the columns display: the name of shape; the average number of Moore and von Neuman neighbours; the speed, coverage and cohesion scores (described in the text); and the total number of runs successfully completed

Shape	Moore	von Neumann	Speed (cm/s)	Coverage	Cohesion	Runs
1A	0.00	0.00	0.57 [3.0e–5]	0.56	1.00	10/10
2A	1.00	1.00	0.47 [4.9e–4]	0.60	1.00	10/10
3A	2.00	1.33	0.39 [1.1e–4]	0.64	0.91	7/10
3B	1.33	1.33	0.35 [2.9e–4]	0.63	1.00	10/10
4A	3.00	2.00	0.34 [1.4e–4]	0.67	1.00	10/10
4B	2.50	1.50	0.33 [3.2e–4]	0.68	1.00	10/10
5A	2.80	1.60	0.26 [3.1e–3]	0.62	0.51	1/10
5B	3.20	1.60	0.25 [3.5e–4]	0.59	0.66	5/10
6A	3.67	2.33	0.23 [3.5e–4]	0.65	1.00	10/10
6B	3.33	2.00	0.23 [7.2e–4]	0.64	0.62	2/10
7A	4.00	2.29	0.20 [1.2e–4]	0.70	1.00	10/10
7B	2.86	1.71	0.22 [7.3e–4]	0.69	0.85	6/10
8A	4.00	2.50	0.20 [3.0e–4]	0.68	1.00	10/10
8B	3.00	2.00	0.19 [2.3e–4]	0.67	1.00	10/10
9A	4.44	2.67	0.16 [4.3e–4]	0.70	1.00	10/10
					Total	121/150

6.2.2 Coverage

Figure 9 shows the area covered by the centre of mass of each structure across all 10 runs. We observe that, in the single module configuration (1A), the robot is able to obtain an even coverage of the entire arena.[6] As the number of robots is increased, it naturally becomes harder for the centre of mass of the structures to reach the edges of the arena. Despite not being able to reach the arena perimeter, and although coverage is increasingly sparse, we still observe a relatively even coverage with configurations of up to nine modules.

To allow us to analyse coverage more quantitatively, we split the arena into 672 equal squares,[7] each measuring approximately 7.18 cm^2. The coverage value in Table 3 is calculated as the percentage of these squares that the centre of mass of any module in the structure has passed through. We would expect the coverage to increase inline with an increase in the number of modules, by virtue of the increase in surface area. However, whilst there is a general trend of increasing coverage between 1A and 9A, it is not as striking as one might expect. This is highlighted by the fact that the two 4 module structures obtain the same coverage score as the two 8 module structures, despite having half as many modules. In the following sub-section, we suggest that the reason coverage does not scale directly with size, is related to the speed at which structures of various sizes travel.

[6] For 1A, there is a bias towards the left-hand side of the arena. This is due to the fact that, in this experiment, the heading of the robot was kept the same across all runs.

[7] It would be easy to achieve 'better' coverage by using fewer squares; however, the purpose of this experiment was to compare the coverage of different configurations, not to measure the overall performance. We therefore deliberately chose a value that was large enough to ensure that no configuration was able to achieve a score of 1.0.

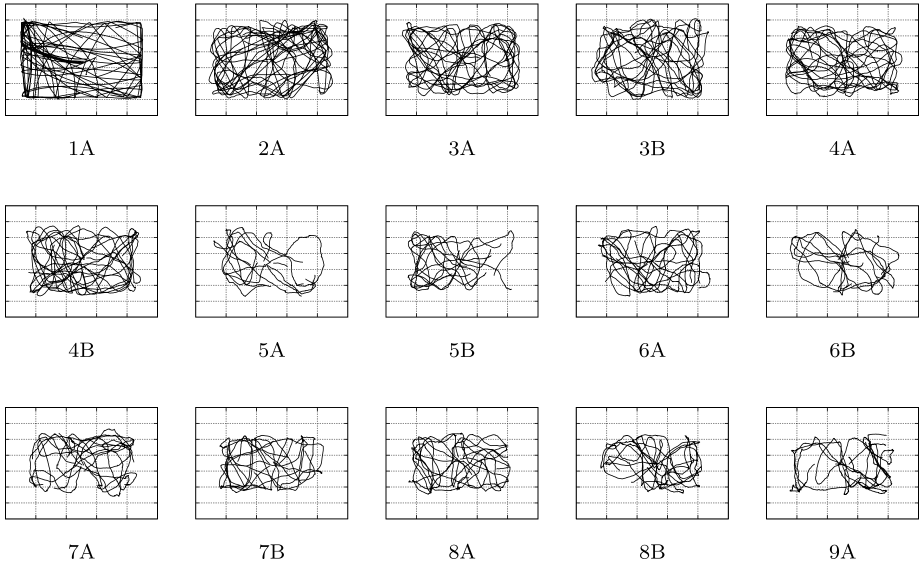


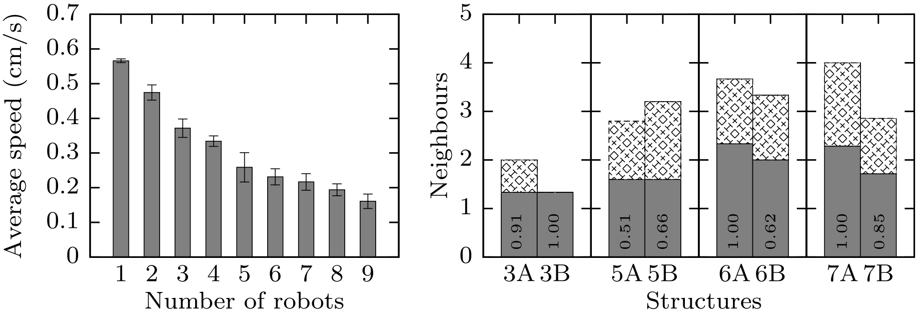
Fig. 9 Plots tracing the route of the centre of mass of each structure in the collective locomotion experiments. Each plot combines data from 10 runs

6.2.3 Speed

Speed is calculated by measuring the average distance travelled by each module in 0.5 seconds, summed over the course of a run and divided by the duration of the run. The value in Table 3 is averaged over all runs and the variance is shown alongside. In general, we observe that as the number of modules in a structure is increased, the average speed at which the structure travels decreases. This may help explain why the coverage of larger structures is lower than expected.

Figure 10a plots the mean average speed for structures of different sizes. The results from configurations which contain the same number of modules are combined into the same sample. Whiskers are used to show the extent of the mean values plus or minus one standard deviation. Using the Wilcoxon rank-sum test with a significance level of 0.05, we can say that there is a statistically significant difference between all pairs of samples in Fig. 10a.

We suggest that the reason why larger structures travel slower than smaller ones is because, in a larger structure, it takes the robots longer to synchronise their headings. The problem is compounded by the fact that a larger structure will need to turn to avoid obstacles more often than a smaller one, further increasing the time in which the robots will be out of synchronisation with one another. Furthermore, because the robots can only ever determine the approximate heading of their neighbours, there will always be noise in their measurements. Since the approach relies heavily on infrared communication it is also highly susceptible to errors caused by reflections or interference. With greater numbers of robots, the probability of such errors will increase, further increasing the level of noise and reducing the accuracy in the estimation of a neighbours heading. With less accurate measurements, the time taken to reach a consensus can be expected to increase further.



(a)

(b)

Fig. 10 Graphs showing the mean average speed (± one standard deviation) for structures of different sizes (**a**) and the average number of von Neumann (*dark grey box*) and Moore neighbours (*dark grey box plus hatched box*) in pairs of structures containing 3, 5, 6 and 7 robots (**b**)

6.2.4 Cohesion

In order to analyse the ability of robots to remain connected with one another in more detail, we define a ‘cohesion’ value. The cohesion value is calculated by dividing the total time in which the modules remained connected, over the maximum available time.

Out of the 15 different structures, 10 were able to remain connected for the full duration of every run, and are therefore assigned a cohesion value of 1.0. From the remaining configurations, structures 5A and 6B were observed to be the least stable, obtaining cohesion scores of 0.51 and 0.62 whilst remaining connected for the full duration in only 1 and 2 of the 10 runs, respectively. Structures 5B and 7B were the next least stable, with scores of 0.66 and 0.85. Of all the structures with a success rate of less than 100 %, structure 3A was the most stable, remaining connected 91 % of the time.

We observe structural similarities between all five configurations with a cohesion score of less than 1.0. Noting that the 3A pattern is repeated in all of these shapes. However, the 3A pattern is also present in structures 4B, 7A and 8B, which each obtained cohesion scores of 1.0. This implies that the 3A pattern alone does not represent a universal motif of instability, and other factors such as the density and symmetry of a shape may also have a role.

We were unable to find a single metric which accurately predicts the stability of a configuration purely from its geometric shape. Having attempted to use measures including the length of the perimeter, the number of loosely connected modules (those with only one neighbour) and the elongation of the shape, we eventually found that, in comparing two structures of the same size, the average number of neighbours provided the best estimate of stability.

In Fig. 10b, we plot the average number of neighbours in a von Neumann neighbourhood (dark grey box) and the average number in a Moore neighbourhood (dark grey box plus hatched box) for every configuration containing 3, 5, 6 or 7 modules. Within each box we print the cohesion score for the corresponding shape. In comparing structures of the same size, we find that in every case, the structure with the larger cohesion value has at least as many von Neumann neighbours as the structure with the lower cohesion value. When considering a Moore neighbourhood, we find that in every case, apart from structures containing 3 modules, the configuration with the largest average number of neighbours has the largest cohesion score.

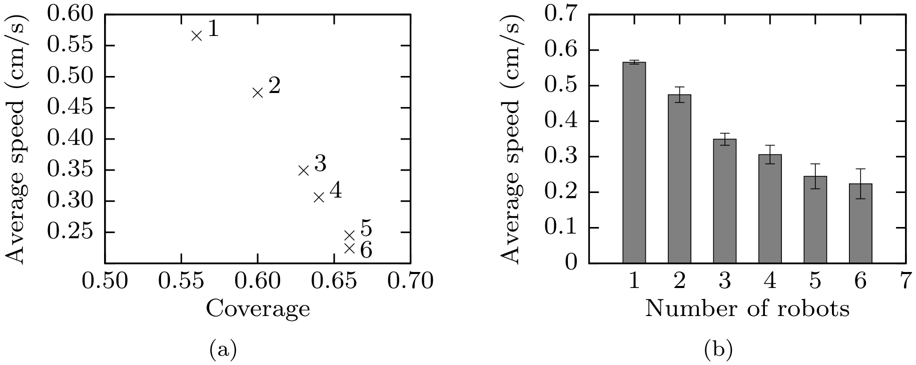


Fig. 11 Graphs (**a**) shows the relationship between the average speed and coverage scores for linear chains of between one and six robots, whilst (**b**) shows the mean average speed (± one standard deviation)

6.3 Scalability

To examine the scalability of our approach, we performed three further experiments involving linear chains of four, five and six modules. By combining these results with those of structures 1A, 2A and 3B, it is possible to observe how the system scales for linear configurations of between one and six modules.

As shown in Table 3, for chains containing one, two and three modules, the robots remained connected for the full duration of every run. This was also true in the four module case. For the five and six module cases, in one and two runs, respectively, the structures were not able to remain connected, resulting in cohesion scores of 0.95 and 0.92. We observe that, as the number of modules is increased, the chances of them remaining connected decreases.

Figure 11a plots the average speed against the coverage scores of each of the linear structures. The graph shows that as the number of modules is increased, whilst the average speed decreases, the coverage continues to increase.

Figure 11b plots the average speed of linear chains containing up to six modules. As the number of modules is increased from one to three, we observe a sharp drop in speed, however, in chains containing between four and six modules, the decrease in speed begins to level off. Based upon a Wilcoxon rank-sum test with a significance level of 0.05, we may state that the drop in performance between the four and six module chains is not statistically significant. That is to say, the performance of the system is not significantly affected by the addition of extra modules, indicating that the system scales well with respect to speed. It is noted, however, that due to the small size of the arena, whilst avoiding obstacles, larger structures spend a lot of their time stationary. Different insights into the scalability of the system may be gained by experimenting within a larger arena, where less time is spent avoiding obstacles.

7 Self-assembly

In Murray et al. (2012), we described an experiment using a previous version of the collective locomotion algorithm introduced in Sect. 4. In this earlier version of the algorithm, the link between boxes (2) and (3) in Fig. 2 was not present and the forward bias always took the same value as the forward speed parameter, equivalent to assigning the robots a synchronisation score of 0. In this experiment, three robots, each running the collective locomotion algorithm with parameter set ‘A’, were placed in different corners of a small arena and left to operate for 10 minutes. We observed that the robots gradually converged to a close proximity and after about five minutes, two of the modules joined together. Soon after, the third module joined the other two to form a single structure, and the modules remained connected until the end of the run.

This self-assembling behaviour was not pre-programmed, it emerged purely due to the interaction of the robots and their environment. Specifically, we propose that it results from a combination of four factors. Firstly, the enclosed arena ensures that robots never stray too far away from one another. Secondly, the alignment behaviour ensures that the robots all head in a similar direction. Thirdly, the design of the e-puck extension ensures that if two robots come into close proximity, their magnetic docking interfaces will cause them to ‘snap’ together. Finally, although there is no explicit cohesion behaviour, the implementation of virtual sensors described in Sect. 4.2 may cause robots to move towards one another when they mistakenly believe themselves to be aligned with the blind spot of another robot.

When robots are near to one another, we state that the virtual sensors have the effect of producing a more accurate and stable form of alignment, but that when the robots are far apart, the virtual sensors cause them to turn towards one another, resulting in a cohesive behaviour. In this section, we describe a new experiment to test whether virtual sensors really do benefit self-assembly.

7.1 Experimental setup

Beginning with a group of separate individuals, we measure the time taken for the robots to assemble into a single structure. We test the performance of three different systems, one in which virtual sensors are used (A), one in which virtual sensors are not used but the rest of the alignment behaviour is present (B) and a control experiment in which no form of alignment is implemented, and the robots simply perform wandering (C).

For each system, we performed 20 experimental runs, half of which involved five robots, and half of which involved ten. Data was recorded for ten minutes of every run, regardless of whether the robots successfully assembled into a single structure. The number of times in which the system was able to successfully assemble, and the time taken to do so was recorded. Unsuccessful runs were assigned the maximum completion time of 600 s. To test whether there is a significant difference between the time taken to self-assemble in systems which use the virtual sensor approach (A) and systems which do not (B and C), we present the following four null hypotheses:

$H1_0$: There is no difference in the time taken by *five* robots using system A and five robots using B to self-assemble into a single structure

$H2_0$: There is no difference in the time taken by *five* robots using system A and five robots using C to self-assemble into a single structure

$H3_0$: There is no difference in the time taken by *ten* robots using system A and ten robots using B to self-assemble into a single structure

$H4_0$: There is no difference in the time taken by *ten* robots using system A and ten robots using C to self-assemble into a single structure

7.2 Analysis

The number of times in which assembly was successfully completed, and the average time taken to do so, is reported in Table 4. We observe that, with both five and ten robots, the

Table 4 A summary of results from the self-assembly experiments. From left to right, the columns show: the number of robots, the system used, the average assembly time, the average number of structures assembled during the final 300 s of each run, and the number of runs in which assembly was successful

#	System	Time (s)			# Structures (final 300 s)			Runs
		Mean	Median	Std.	Mean	Median	Std.	
5	A	245.40	255.00	64.93	1.01	1.00	0.03	10/10
5	B	398.60	399.00	133.30	1.44	1.33	0.42	8/10
5	C	573.15	600.00	84.91	2.13	2.03	0.47	1/10
10	A	381.15	397.00	178.47	1.50	1.37	0.42	7/10
10	B	411.30	544.00	225.49	2.19	2.05	0.54	5/10
10	C	600.00	600.00	0.00	3.57	3.68	0.58	0/10

system in which virtual sensors are used always outperforms the other two, both in terms of the mean assembly time and the number of successful runs. In the five robot scenario, system A was able to assemble into a single structure during every run and did so on average 2.5 minutes quicker than system B. System B was successful in completing assembly on seven occasions, whilst system C, purely by chance, was able to fully assemble on one occasion. With ten robots, system A was able to assemble into a single structure on seven occasions, and did so on average 30 seconds quicker than system B, which completed assembly on five occasions. In the ten robot case, system C was never able to successfully complete assembly.

Figure 12 plots the assembly time for systems A and B (top), and the mean number of structures over time for all three systems, both for the five robot (a) and ten robot (b) scenarios. In general, we observe that the mean number of structures present during the experiments of system A is lower than for systems B and C; however, the difference is much less in the ten robot setting than in the five robot setting, specifically during the first 200 seconds.

Behaviourally, in systems A and B, we first observe the formation of small structures. These structures combine with other robots to form larger groups, eventually resulting in the formation of a single coherent structure. With system C we observe chance collisions that lead to the formation of static structures. Without an alignment behaviour the robots are not able to perform collective locomotion, but do not move with sufficient force to break apart. Therefore, after forming structures, the robots in system C simply remain stationary for the remainder of the run.

For the five robot scenario, using the Wilcoxon rank-sum test with a significance level of 0.05 we may reject both $H1_0$ and $H2_0$, and accept the alternatives. That is to say, with five robots, there is a significant difference between the time taken by system A and systems B and C to self-assemble.

For the ten robot scenario, in comparing systems A and B, it is clear from the large overlapping of the boxplots in Fig. 12b that we cannot reject $H3_0$. This was confirmed using the Wilcoxon rank-sum test with a significance level of 0.05. Although the average (both mean and median) assembly time of system A is shown to be less than that of system B in Table 4, the difference between the medians is not statistically significant. We can, however, reject $H4_0$, and state that there is a significant difference in the time taken by systems A and C to self-assemble into a single structure.

In comparing systems A and B, we suggest that the reason why we only observe a significant improvement in performance in the five robot scenario, may be due to the population

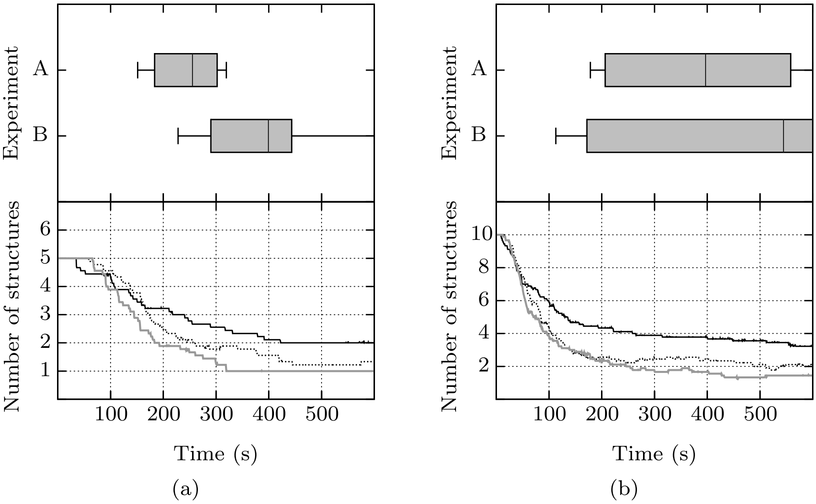


Fig. 12 Boxplots showing the assembly time for systems A and B (*top*), and graphs showing the mean number of structures over time (*bottom*), both for experiments containing five (**a**) and 10 robots (**b**). In the lower charts, the *grey line* corresponds to system A, the *dotted line* to B and the *black line* to C

density of robots within the arena. In the case of ten robots, with the modules packed twice as densely, the probability of two robots colliding purely by chance is greatly increased, potentially masking any advantage that the virtual sensors provide. This is supported by the graph in Fig. 12b in which, over the first 50 seconds, the mean number of structures follows the same trend for all three experiments. As the robots begin to cluster, however, the probability of chance collisions is reduced and as observed during the period from 300 seconds onwards, system A begins to outperform system B, which in turn outperforms system C.

Another reason that we do not observe a significant difference between the assembly time of systems A and B when using ten robots, may be due to the fact that uncompleted runs are assigned a completion time of 600 s. Assigning a value of 600 is a very conservative estimate. Allowing the runs to continue for longer would reveal a more accurate representation of the average time taken to assemble a structure. Based upon the fact that after 600 s, system A was successful on more occasions than B, and that the mean number of structures in Fig. 12b is consistently lower for A, if all runs were allowed to continue until completion we would expect A to perform significantly better than B.

To avoid the problems mentioned above, we devise a new metric to analyse the performance of groups of ten robots. Firstly, to prevent the problem that a high density of robots may simplify the task to the point of masking the benefits of the virtual sensors, we analyse the system only during the final 300 s of each run. Secondly, to allow us to assess the performance of uncompleted runs more fairly, instead of measuring the time taken until completion, we measure the average number of structures assembled at any point during the final 300 s of a run. For this new metric, we propose a fifth hypothesis:

$H5_0$: There is no difference in the mean number of structures assembled by ten robots during the final 300 s of runs involving systems A and B

As shown in Table 4, in comparing groups of ten robots using systems A and B, we observe that the groups using system A have both lower mean and median values for the number of structures assembled during the final 300 s of each run. Using the Wilcoxon rank-sum test with a significance level of 0.05 we can reject $H5_0$ and confirm that there is a significant difference in the average number of structures assembled by groups of robots that use virtual sensors (A) and groups that do not (B).

8 Environment driven self-reconfiguration

In order to test the behaviour of the robots in a more complex setting, we introduce a dynamic component to the environment. As shown in Fig. 6a, this is realised as a ring of LEDs surrounding the arena. The LEDs may be viewed as a proxy for more complex real-world perturbations, such as changes in terrain or weather conditions. In order to cope with such changes, robots must be able to adapt their behaviour. In our simplified setting, we examine the switch between self-assembling and self-disassembling behaviours in response to a change in the state of the surrounding LEDs. We begin by describing a simple self-disassembly behaviour, before going on to describe experiments, inspired by the work of Weel et al. (2012), which combine self-disassembly with the previously introduced self-assembly behaviour, to produce a simple form of environment driven self-reconfiguration.

8.1 Self-disassembly

As with the self-assembly behaviour, in our self-disassembly strategy, the robots continuously broadcast their IDs and listen for the IDs of their neighbours. Robots can detect whether they are connected to another module based upon the intensity of the messages that they receive. If a robot determines that it is connected to at least one other module, it will, with equal probabilities, either set both motors to the maximum forward value, both to the maximum reverse value, or both to alternate maximum values. The resulting ‘shaking’ motion is normally sufficient to break the connection between two modules. If a robot determines that it is not connected to any other modules, it will simply wander in the arena, avoiding obstacles and other robots.

To test this behaviour we ran a series of experiments, using a variety of different structures. Specifically, we investigated all of the configurations from Fig. 7 that contain either four, five or six modules. In each experiment, we measured the number of structures, the average number of robots in each structure and the time taken by the robots to completely disassemble.

Figure 13 plots the mean number of structures and the mean size of each structure over the first 150 seconds of each five minute run. In every case, we observe the number of structures increase from one to N, where N is the number of robots in the initial structure. Naturally, we also observe that as the number of structures increases, the average size of the structures decreases.

As may be expected, with more robots, we observe that the time taken to fully disassemble is greater. This is both because more connections must be broken between neighbouring robots and because, in a more densely packed environment, there is more chance of robots rejoining with each other during disassembly. In comparing structures of the same size, we also observe that the structure with the greater number of von Neumann neighbours (corresponding to the number of physical connections) takes longer to disassemble. This is most evident in the case of structures 4A and 4B (Fig. 13, top).

Unfortunately, fabrication defects, leading to variations in the positioning of the magnets, meant that certain sides of certain robots appear to be more ‘sticky’ than others. On two occasions, in the experiments involving six robots, this led to runs in which the robots were

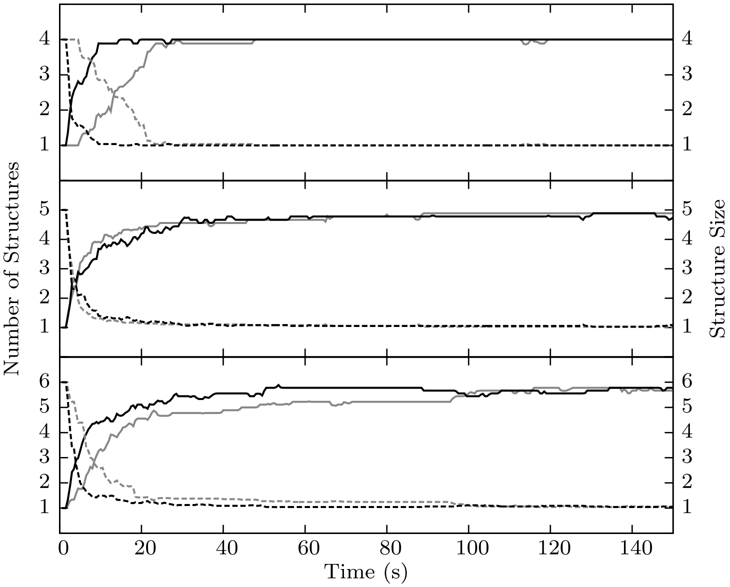


Fig. 13 The average number of structures (*solid line*) and the average structure size (*dashed line*) over time. Results are presented for six different structures, two of each containing four, five and six robots. In each plot, the *dark lines* correspond to 'A' shapes, whilst the *lighter lines* correspond to 'B' shapes

not able to completely disassemble, as one pair of robots could not be separated. This introduced large outliers into the data. These outliers, many times larger than the mean, make statistical analysis difficult with such small samples, however we still observe from Fig. 13 that the general desired behaviour is present.

8.2 Self-reconfiguration

To investigate environment driven self-reconfiguration, we conducted 10 independent runs, each lasting 11 minutes and involving six modules. At the start of each experiment, the robots were spread evenly throughout the arena and began by executing the self-disassembly behaviour. After 50 seconds, the robots were triggered to switch from disassembly to assembly and remained in this state for a further five minutes. They then switched back to disassembly until the end of the run. The number of structures present was recorded over the duration of each run, the mean values are plotted in Fig. 14 ± one standard deviation. For simplicity, repeatability, and to allow the experiments to be fully automated we use the robots' internal clocks to trigger the switch between assembling and disassembling. However, as shown in Fig. 15, we have also demonstrated this behaviour by using the LED ring as a trigger.

The light grey area in Fig. 14 marks the five minute period during which the self-assembly behaviour was active. During this period, the average number of structures steadily decreases, flattening out at just below two. In 7 out of 10 runs, this five minute period was sufficient for all of the robots to assemble into a single structure, in the other three cases, the robots assembled into two separate structures. After the robots switched back to the disassembly state, in all but one of the runs, they were able to completely disassemble into six independent structures. After disassembling, the robots did not always remain disassociated

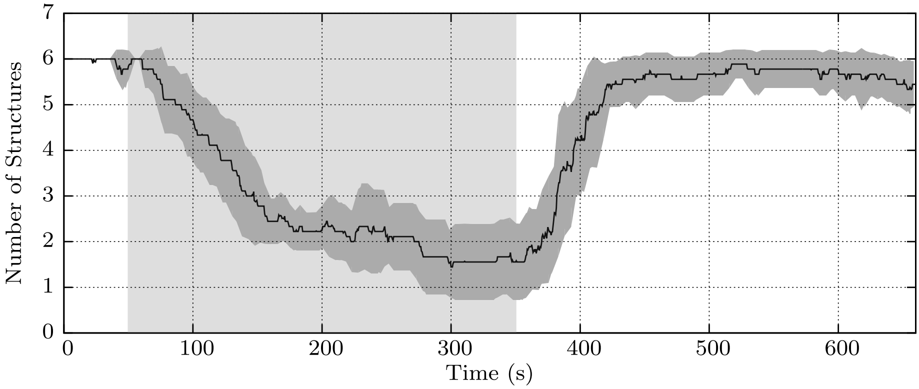
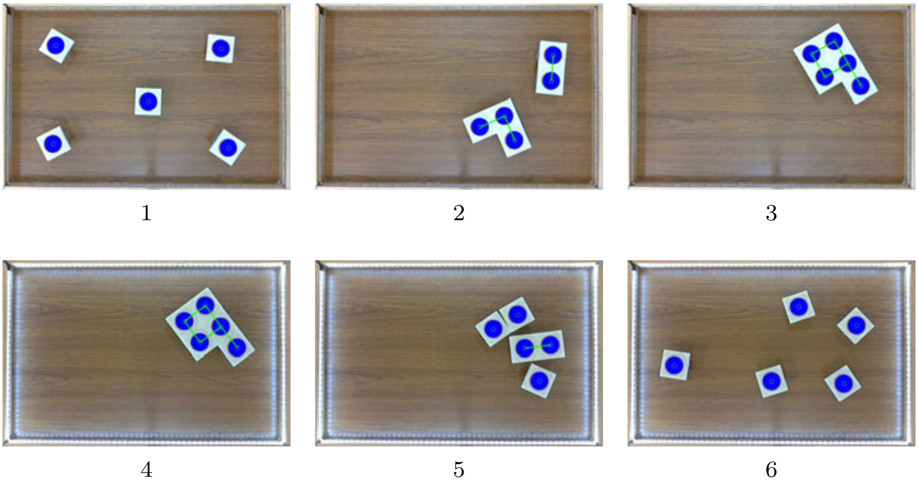


Fig. 14 A graph showing the average number of individual structures over time. The *dark shaded region* shows the values ± one standard deviation, whilst the *lighter region* highlights the period of time during which the robots self-assembly behaviour was active


1 2 3

4 5 6

Fig. 15 Still images from a video of environment driven self-reconfiguration

until the end of the run, with chance collisions occasionally and temporarily bringing modules back together. This explains the variation in the mean number of structures over the last 200 s.

In contrast to the results presented in Fig. 14, where an internal timer was used to switch between assembling and disassembling behaviours, Fig. 15 provides screenshots from a video of an experiment in which the arena's LED ring was used to trigger reconfiguration. The robots sense the LED ring by tracking changes in the ambient lighting conditions using their IR sensors. Shortly after startup, a threshold value is established for each sensor by monitoring ambient light with the LED ring turned off. At each timestep, by comparing the ambient light with their threshold value, each sensor provides one vote to determine whether or not the robot should switch between assembling and disassembling. The majority vote determines the robot's behaviour.

The video from Fig. 15 is available in the supplementary online material and can also be viewed, along with others, on the project website.[8] The robots begin in a disconnected state, executing the self-assembly behaviour (1). At first they form two independent structures (2) which later collide to form a single structure (3). At which point the LED ring is turned on (4) triggering the switch from assembly to disassembly and causing the robots to split apart (5). Eventually, the robots reach a completely disassembled state (6), at which point the whole process may be restarted by turning off the LEDs.

9 Summary and future work

In this paper, we presented an extension for the e-puck robot that transforms what is traditionally a swarm robotics platform into a mobile self-reconfigurable modular robotic system. We also presented an algorithm for controlling the collective locomotion of a group of robots equipped with the modular e-puck extension. As a consequence of the novel use of ‘virtual sensors’, without alteration, the same collective locomotion strategy was used to demonstrate self-assembling and self-reconfiguring behaviours.

In Sect. 6, we extended our earlier collective locomotion work by examining the performance of robots arranged in 15 different structural configurations. In order to fully analyse the behaviour of our alignment strategy, we chose to conduct a broad survey covering a variety of different types of structure. To minimise the risk of structures breaking apart, the speed of the individual robots was restricted, as a result, the speed at which the collective structures travel is relatively slow in comparison to the maximum speed of the e-puck. Despite their low speed, we found that in the majority of cases, even with structures containing up to nine modules, the robots were able to successfully explore an enclosed arena whilst avoiding obstacles. We observed that certain configurations were more stable than others and found that a good (though not universal) relative measure for the stability of similar-sized structures, is the average number of neighbours that the individual modules possess. The system showed promise with regards to its scalability, but it is acknowledged that further insight may be gained by conducting experiments within a larger arena, and using a greater number of modules.

In Sect. 7, we analysed the robot’s ability to perform self-assembly, by focusing on how this behaviour relates to the implementation of ‘virtual sensors’. Virtual sensors are imaginary sensors that are notionally located in the blind spots between real sensors and allow the position of neighbouring robots to be inferred more accurately by analysing the intensity of the messages received by the real sensors. We found that in a sparse environment with five modules, robots utilising the virtual sensor approach were able to self-assemble into a single structure significantly quicker than robots that did not utilise virtual sensors. In a denser environment with ten robots, the advantage of virtual sensors was less obvious; however, in analysing the number of structures formed during the final 300 s of each run, we found that the robots were able to assemble into significantly fewer groups when using the virtual sensor approach. Whilst it is accepted that a more directed form of self-assembly could outperform our strategy in terms of speed it would be unlikely to possess the same elegant simplicity. Investigating the possibility of more directed forms of self-assembly would represent an interesting area of future work.

[8] http://www.elec.york.ac.uk/research/projects/Modular_e_puck_Extension.html.

Finally, in Sect. 8, we described a simple self-disassembling behaviour. By combining this behaviour with self-assembly, we investigated the more complex task of self-reconfiguration within a changing environment. Using a ring of LEDs as an environmental trigger, we demonstrated that the robots were able to successfully switch between self-assembling and self-disassembling behaviours when changes in the ambient lighting conditions were detected.

In comparison to similar platforms, such as the X-Cell (Hong et al. 2011), DFA (Oung and D'Andrea 2011), s-bot (Mondada et al. 2004) and foot-bot (Dorigo et al. 2013) robots, the modular e-puck extension is one of the cheapest and simplest available. Furthermore, it is one of few modular robotic systems for which the designs have been made freely available online.

The simplistic design of the extension means that, like the DFA system, the platform is well suited to the type of passive self-assembly that is more commonly utilised by externally-propelled systems. An interesting consequence of which is that the robots do not require a seed module in order to self-assemble. Assembly is performed in a bottom-up fashion, in which the robots first form pairs or small groups, and only later combine to form single structures.

Simplicity, however, comes at a cost, and the reduced sensing and actuating capabilities of the platform present certain limitations. For example, although the freedom of the rotating frame is beneficial in terms of increasing the flexibility of the system, the lack of control and the absence of the ability to sense the current orientation of the frame make the formation of specific configurations more difficult. In contrast to the modular e-puck extension, the X-Cell, s-bot and foot-bot platforms, which also possess rotating turrets, allow for fine-grained control and sensing of the orientation of their elements.

Another limitation of the platform is that, at present, the only method that has been developed to allow the robots to disassemble is the random shaking motion described in Sect. 8.1. Whilst this works well when it is desirable for an entire structure to be separated, it is harder to utilise this behaviour in order to remove only one or two modules, for example, during partial self-reconfiguration. The same problem is also suffered by the DFA platform, but is circumvented by the X-Cell, s-bot, and foot-bot platforms, in which it is possible to actively control the connection between two modules without disrupting the connections of neighbouring robots. The mirror image of this problem is the fact that, as shown in Sect. 6, it is not always possible to guarantee that the robots will remain connected, particularly as the number of modules is increased. This problem is also solved by the X-Cell, s-bot and foot-bot modules through the use of active docking elements.

In future work, we would like to investigate different tasks, such as self-repair, for which it will be necessary for the robots to form specific configurations. This may be achieved in an implicit manner using local alignment rules that have a natural tendency to produce one type of structure over another, or in a more directed fashion, like that employed by O'Grady et al. (2009), in which the modules possess an explicit representation of their own structure and use signalling methods to attract robots to the required locations. Depending upon the method employed, hardware extensions may be necessary. For example, a rotational sensor would make it easier for the robots to determine the relative heading of their neighbours. Furthermore, an actuator for controlling the rotation of the frame, and devices for signalling on specific sides of a module, would aid in directed self-assembly.

In summary, we have developed a novel control strategy that exploits the physical positioning of the sensors on board the e-puck robot. We have used the modular e-puck extension to demonstrate collective locomotion, self-assembly, self-disassembly and self-reconfiguration, both within static and dynamic environments. We have shown that the platform is capable of providing meaningful experimental results to non-trivial problems and that the simple control strategies developed scale to structures containing up to ten modules. We conclude that the modular e-puck extension represents a viable, low-cost platform for research into self-reconfigurable modular robotics.

Acknowledgements This research is supported by the SYMBRION project, within the Seventh Framework Programme (FP7). Project no. FP7-ICT-2007.8.2, grant agreement 216342. Jon Timmis is partially supported by the Royal Society. The authors would like to thank the reviewers for their detailed and comprehensive comments, which greatly helped to improve the quality of this manuscript.

References

Baldassarre, G., Trianni, V., Bonani, M., Mondada, F., Dorigo, M., & Nolfi, S. (2007). Self-organized coordinated motion in groups of physically connected robots. *IEEE Transactions on Systems, Man and Cybernetics. Part B. Cybernetics*, *37*(1), 224–239.

Beni, G. (1988). The concept of cellular robotic system. In *Proceedings of the 1988 IEEE international symposium on intelligent control* (pp. 57–62). New York: IEEE Press.

Beni, G. (2005). From swarm intelligence to swarm robotics. In E. Şahin & W. Spears (Eds.), *LNCS: Vol. 3342*. *Swarm robotics* (pp. 1–9). Berlin: Springer.

Bishop, J., Burden, S., Klavins, E., Kreisberg, R., Malone, W., Napp, N., & Nguyen, T. (2005). 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* (pp. 3684–3691). New York: IEEE Press.

Campo, A., Nouyan, S., Birattari, M., Groß, R., & Dorigo, M. (2006). Negotiation of goal direction for cooperative transport. In *LNCS: Vol. 4150. Proceeding of ANTS 2006, 5th international workshop on ant colony optimization and swarm intelligence* (pp. 191–202). Berlin: Springer.

Cianci, C. M., Raemy, X., Pugh, J., & Martinoli, A. (2007). Communication in a swarm of miniature robots: the e-puck as an educational tool for swarm robotics. In *LNCS: Vol. 4433*. *Proceedings of the 2nd international conference on swarm robotics, SAB 2006* (pp. 103–115). Berlin: Springer.

Dorigo, M., Tuci, E., Trianni, V., Groß, R., Nouyan, S., Ampatzis, C., Labella, T. H., O'Grady, R., Bonani, M., & Mondada, F. (2006). SWARM-BOT: design and implementation of colonies of self-assembling robots. In G. Y. Yen & D. B. Fogel (Eds.), *Computational intelligence: principles and practice* (pp. 103–135). New York: IEEE Computational Intelligence Society. Chap. 6.

Dorigo, M., Floreano, D., Gambardella, L. M., Mondada, F., Nolfi, S., Baaboura, T., Birattari, M., Bonan, M., Brambilla, M., Brutschy, A., Burnier, D., Campo, A., Christensen, A. L., Decugnière, A., Caro, G. A. D., Ducatelle, F., Ferrante, E., Förster, A., Gonzales, J. M., Guzzi, V. L., Magnenat S, J., Mathews, N., de Oca MM O'Grady, R., Pinciroli, C., Pini, G., Rétornaz, P., Roberts, J., Sperati, V., Stirling, T., Stranieri, A., Stützle, T., Trianni, V., Tuci, E., Turgut, A. E., & Vaussard, F. (2013, in press). Swarmanoid: a novel concept for the study of heterogeneous robotic swarms. *IEEE Robotics and Automation Magazine*.

English, S., Gough, J., Johnson, A., Spanton, R., & Sun, J. (2012). Formica. http://warrantyvoidifremoved.com/formica/. Accessed 9 April 2013.

Ferrante, E., Brambilla, M., Birattari, M., & Dorigo, M. (2013). Socially-mediated negotiation for obstacle avoidance in collective transport. In *Springer tracts in advanced robotics: Vol. 83*. *Proceedings of the 10th international symposium on distributed autonomous robotic systems, DARS 2010* (pp. 571–583). Berlin: Springer.

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

Fukuda, T., & Nakagawa, S. (1988b). Dynamically reconfigurable robotic system. In *Proceedings of the 1988 IEEE international conference on robotics and automation, ICRA 1988* (pp. 1581–1586). New York: IEEE Press.

Fukuda, T., Nakagawa, S., Kawauchi, Y., & Buss, M. (1988). Self organizing robots based on cell structures—CEBOT. In *Proceedings of the 1988 IEEE international workshop on intelligent robots* (pp. 145–150). New York: IEEE Press.

GCtronic (2012). Elisa 3. http://www.gctronic.com/doc/index.php/Elisa-3. Accessed 9 April 2013.

Gilpin, K., & Rus, D. (2010). Self-disassembling robot pebbles: new results and ideas for self-assembly of 3D structures. In *Proceedings of the 2010 IEEE international conference on robotics and automation, ICRA 2010* (pp. 94–99). New York: IEEE Press.

Gilpin, K., Kotay, K., & Rus, D. (2007). Miche: modular shape formation by self-dissasembly. In *Proceedings of the 2007 IEEE international conference on robotics and automation, ICRA 2007* (pp. 2241–2247). New York: IEEE Press.

Gilpin, K., Kotay, K., Rus, D., & Vasilescu, I. (2008). Miche: modular shape formation by self-disassembly. *The International Journal of Robotics Research*, *27*, 345–372.

Gilpin, K., Knaian, A., & Rus, D. (2010). Robot pebbles: one centimeter modules for programmable matter through self-disassembly. In *Proceedings of the 2010 IEEE international conference on robotics and automation, ICRA 2010* (pp. 2485–2492). New York: IEEE Press.

Goldstein, S., Campbell, J., & Mowry, T. (2005). Programmable matter. *Computer*, *38*(6), 99–101. doi:10.1109/MC.2005.198.

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

Groß, R., & Dorigo, M. (2009). Towards group transport by swarms of robots. *International Journal of Bio-Inspired Computation*, *1*(1–2), 1–13.

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., Mondada, F., & Dorigo, M. (2006b). Transport of an object by six pre-attached robots interacting via physical links. In *Proceedings of the 2006 IEEE international conference on robotics and automation, ICRA 2006* (pp. 1317–1323). New York: IEEE Press.

Groß, R., Magnenat, S., Küchler, L., Massaras, V., Bonani, M., & Mondada, F. (2011). Towards an autonomous evolution of non-biological physical organisms. In *LNCS: Vol. 5777. Proceedings of the 10th European conference on artificial life, ECAL 2009* (pp. 173–180). Berlin: Springer.

Gutiérrez, A., Campo, A., Dorigo, M., Donate, J., Monasterio-Huelin, F., & Magdalena, L. (2009a). Open e-puck range & bearing miniaturized board for local communication in swarm robotics. In *Proceedings of the 2009 IEEE international conference on robotics and automation, ICRA 2009* (pp. 3111–3116). New York: IEEE Press.

Gutiérrez, A., Tuci, E., & Campo, A. (2009b). Evolution of neuro-controllers for robots alignment using local communication. *International Journal of Advanced Robotic Systems*, *6*(1), 25–34.

Hong, W., Wang, S., & Shui, D. (2011). Reconfigurable robot system based on electromagnetic design. In *Proceedings of the 2011 international conference on fluid power and mechatronics, FPM 2011* (pp. 570–575). New York: IEEE Press.

Jørgensen, M., Østergaard, E., & Lund, H. (2004). Modular ATRON: modules for a self-reconfigurable robot. In *Proceedings of the 2004 IEEE/RSJ international conference on intelligent robots and systems, IROS 2004* (Vol. 2, pp. 2068–2073). New York: IEEE Press. doi:10.1109/IROS.2004.1389702.

K-Team Corporation (2012). K-team mobile robotics. http://www.k-team.com. Accessed 10 February 2012.

Kernbach, S. (2012). Jasmine swarm robot platform. http://www.swarmrobot.org. Accessed 10 February 2012.

Kernbach, S., Scholz, O., Harada, K., Popesku, S., Liedke, J., Raja, H., Liu, W., Caparrelli, F., Jemai, J., Havlik, J., Meister, E., & Levi, P. (2010). Multi-robot organisms: state of the art. In *Proceedings of the 2010 IEEE international conference on robotics and automation, ICRA 2010*, Workshop on "Modular robots: state of the art", Anchorage, Alaska (pp. 1–10).

Liu, W., & Winfield, A. F. (2011). Open-hardware e-puck Linux extension board for experimental swarm robotics research. *Microprocessors and Microsystems*, *35*(1), 60–67. doi:10.1016/j.micpro.2010.08.002.

Liu, W., & Winfield, A. F. T. (2010). Autonomous morphogenesis in self-assembling robots using IR-based sensing and local communications. In *LNCS: Vol. 6234. Proceedings of ANTS 2010, 7th international conference on swarm intelligence* (pp. 107–118). Berlin: Springer.

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

Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klaptocz, A., Magnenat, S., christophe Zufferey, J., Floreano, D., & Martinoli, A. (2009). The e-puck, a robot designed for education in engineering. In *Proceedings of the 9th conference on autonomous robot systems and competitions, IPCB: Instituto Politécnico de Castelo Branco* (pp. 59–65).

Murata, S., Kurokawa, H., & Kokaji, S. (1994). Self-assembling machine. In *Proceedings of the 1994 IEEE international conference on robotics and automation, ICRA 1994* (pp. 441–448). New York: IEEE Press.

Murray, L., Timmis, J., & Tyrrell, A. (2012). Self-reconfigurable modular e-pucks. In *LNCS: Vol. 7461. Proceedings of ANTS 2012, 8th international conference on swarm intelligence* (pp. 133–144). Berlin: Springer.

O'Grady, R., Christensen, A., & Dorigo, M. (2008). Autonomous reconfiguration in a self-assembling multi-robot system. In M. Dorigo, M. Birattari, C. Blum, M. Clerc, T. Stützle, & A. Winfield (Eds.), *LNCS: Vol. 5217. Proceedings of ANTS 2008, 6th international conference on ant colony optimization and swarm intelligence* (pp. 259–266). Berlin: Springer.

O'Grady, R., Christensen, A., & Dorigo, M. (2009). SWARMORPH: multirobot morphogenesis using directional self-assembly. *IEEE Transactions on Robotics*, *25*(3), 738–743.

Oung, R., & D'Andrea, R. (2011). The distributed flight array. *Mechatronics*, *21*(6), 908–917. doi:10.1016/j.mechatronics.2010.08.003.

Pamecha, A., Chiang, C. J., Stein, D., & Chirikjian, G. (1996). Design and implementation of metamorphic robots. In *ASME design engineering technical conference and computers in engineering conference*, ASME, Irvine, California.

Pololu Corporation (2012). Pololu robotics & electronics. http://www.pololu.com. Accessed 10 February 2012.

Rubenstein, M., Hoff, N., & Nagpal, R. (2011). *Kilobot: a low cost scalable robot system for collective behaviors* (Tech. rep. TR-06-11). Harvard University. ftp://ftp.deas.harvard.edu/techreports/tr-06-11.pdf. Accessed 11 February 2012.

Ryland, G., & Cheng, H. (2010). Design of iMobot, an intelligent reconfigurable mobile robot with novel locomotion. In *Proceedings of the 2010 IEEE international conference on robotics and automation, ICRA 2010* (pp. 60–65). New York: IEEE Press.

Şahin, E. (2005). Swarm robotics: from sources of inspiration to domains of application. In *LNCS: Vol. 3342*. *Proceedings of the 2004 international conference on swarm robotics, SAB 2004* (pp. 10–20). Berlin: Springer.

Schweikardt, E. (2011). Modular robotics studio. In *Proceedings of the 5th international conference on tangible, embedded, and embodied interaction* (pp. 353–356). New York: ACM.

Suh, J., Homans, S., & Yim, M. (2002). Telecubes: mechanical design of a module for self-reconfigurable robotics. In *Proceedings of the 2002 IEEE international conference on robotics and automation, ICRA 2002* (pp. 4095–4101). New York: IEEE Press.

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

Trianni, V., Nolfi, S., & Dorigo, M. (2006). Cooperative hole avoidance in a swarm-bot. *Robotics and Autonomous Systems*, *54*(2), 97–103.

Weel, B., Hoogendoorn, M., & Eiben, A. (2012). On-line evolution of controllers for aggregating swarm robots in changing environments. In C. Coello, V. Cutello, K. Deb, S. Forrest, G. Nicosia, & M. Pavone (Eds.), *LNCS: Vol. 7492*. *Proceedings of the 12th international conference on parallel problem solving from nature, PPSN XII* (pp. 245–254). Berlin: Springer.

Wei, H., Chen, Y., Tan, J., & Wang, T. (2011). Sambot: a self-assembly modular robot system. *IEEE/ASME Transactions on Mechatronics*, *16*(4), 745–757.

Yim, M., Shen, W. M., Salemi, B., Rus, D., Moll, M., Lipson, H., Klavins, E., & Chirikjian, G. (2007). Modular self-reconfigurable robot systems [grand challenges of robotics]. *IEEE Robotics & Automation Magazine*, *14*(1), 43–52.

Yim, M., White, P., Park, M., & Sastra, J. (2009). Modular self-reconfigurable robots. In R. A. Meyers (Ed.), *Encyclopedia of complexity and systems science* (pp. 5618–5631). Berlin: Springer.

Zykov, V., Chan, A., & Lipson, H. (2007). Molecubes: an open-source modular robotics kit. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems, IROS 2007*, Workshop on self-reconfigurable robotics & systems and applications.