

Self-organisation and communication in groups of simulated and physical robots

Vito Trianni · Marco Dorigo

Received: 3 August 2005 / Accepted: 15 May 2006 / Published online: 5 July 2006
© Springer-Verlag 2006

Abstract In social insects, both self-organisation and communication play a crucial role for the accomplishment of many tasks at a collective level. Communication is performed with different modalities, which can be roughly classified into three classes: indirect (*stigmergic*) communication, direct interactions and direct communication. The use of stigmergic communication is predominant in social insects (e.g. the pheromone trails in ants), where, however, direct interactions (e.g. antennation in ants) and direct communication (e.g. the waggle dance in honey bees) can also be observed. Taking inspiration from insect societies, we present an experimental study of self-organising behaviours for a group of robots, which exploit communication to coordinate their activities. In particular, the robots are placed in an arena presenting holes and open borders, which they should avoid while moving coordinately. Artificial evolution is responsible for the synthesis in a simulated environment of the robot's neural controllers, which are subsequently tested on physical robots. We study different communication strategies among the robots: no direct communication, handcrafted signalling and a completely evolved approach. We show that the latter is the most efficient, suggesting that artificial evolution can produce behaviours that are more adaptive than those obtained with

conventional design methodologies. Moreover, we show that the evolved controllers produce a self-organising system that is robust enough to be tested on physical robots, notwithstanding the huge gap between simulation and reality.

Keywords Swarm robotics · Evolutionary robotics · Self-organisation · Swarm intelligence · Swarm-bot

1 Introduction

Ants are everywhere, but only occasionally noticed. They run much of the terrestrial world as the premier soil turners, channelers of energy, dominatrices of the insect fauna [...] They employ the most complex forms of chemical communication of any animals and their organization provides an illuminating contrast to that of human beings [...] (from Hölldobler and Wilson 1990, p 1).

This way, Hölldobler and Wilson introduce their journey into the ants' world. They provide a passionate, yet rigorous description of this fascinating and intriguing animal society: a picture that serves as inspiration not only for entomologists or socio-biologists, but also for engineers and computer scientists. Indeed, the principles that lie behind the organisation of an ant colony have been so far exploited in multiple domains, resulting in the development of robust optimisation algorithms (see, for example, Dorigo and Stützle 2004) and giving birth to the *swarm intelligence* research domain (Beni and Wang 1989; Bonabeau et al. 1999). Also robotics could benefit from the biologically inspired approach,

Electronic Supplementary material Supplementary material is available in the online version of this article at <http://dx.doi.org/10.1007/s00422-006-0080-x>

V. Trianni (✉) · M. Dorigo
Université Libre de Bruxelles, IRIDIA CP 194/6,
Av. Franklin Roosevelt 50, Brussels 1050, Belgium
e-mail: vtrianni@iridia.ulb.ac.be

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

as demonstrated by the continuously growing interest for *swarm robotics* (Dorigo and Şahin 2004).

In a swarm robotic context, it is useful to allow for self-organisation while designing the different parts of the robotic system. Self-organisation can be defined as the emergence of order in a system as a result of the numerous interactions among the system's components. Although these interactions take place on a purely local basis, their effect on the whole system is the increase of its inherent order. Self-organisation is often observed in biology, and in particular in many animal societies, not limited to social insects like ants, bees or termites (Camazine et al. 2001). From an engineering perspective, there are multiple advantages in designing a self-organising system. Among these, it is worth mentioning that such a system is inherently robust to individual failures, as it is normally redundant in its constituent parts. It can adapt to varying environmental conditions and maintain its organisation notwithstanding certain external perturbations.

Another important aspect to consider is communication, which is often required for coordination of collective behaviours. Social insects make use of different forms of communication, outlined in Sect. 2. In collective robotics research, the coordination of the activities in a group of robots requires the definition of communication strategies and protocols among the individuals. These strategies and protocols need not, however, be particularly complex. In many cases, simple forms of communication – or no explicit communication at all – are enough to obtain the coordination of the activities of the group (Kube and Zhang 1997). This is the case for *swarm robotics*, which focuses on local and simple communication paradigms that can gracefully scale up with the number of agents involved.

However, designing a self-organising system and the related communication protocols for a group of simulated and/or real robots is not a trivial task. From an engineering perspective, the design problem is generally decomposed into two different phases: (1) the behaviour of the system should be described as the result of interactions among individual behaviours and (2) the individual behaviours must be encoded into controllers. Both phases are complex because they attempt to decompose a process (the global behaviour or the individual one) that is a result of dynamical interactions among its sub-components (interactions among individuals or between individual actions and the environment). These dynamical aspects are in general difficult to be predicted by the observer. In such a context, we believe that *Evolutionary Robotics* (ER) is the methodology to be exploited (Harvey et al. 1992, 2005; Nolfi and Floreano 2000). ER bypasses the problem of decomposition

at both the levels of finding the mechanisms that lead to the emergent global behaviour and of implementing those mechanisms in a controller for the robots. In fact, ER relies on the evaluation of the system as a whole, i.e., on the emergence of the desired global behaviour starting from the definition of the individual ones. Moreover, ER can exploit the richness of solutions offered by the dynamic robot–environment interaction, which may not be apparent *a priori* to the experimenter (Nolfi and Floreano 2000; Dorigo et al. 2004).

In this paper, we show how ER techniques can be used for solving a complex task, both with simulated and real robots. In our work, we study a swarm robotic system composed of a number of autonomous mobile robots, referred to as *s-bots*, which have the ability to connect one to the other forming a physical structure – referred to as a *swarm-bot* – that can solve problems the single *s-bots* are not able to cope with (see Fig. 1 and Mondada et al. 2004, for details). The physical connections among *s-bots* result in physical interactions that can be exploited for the self-organisation of the *swarm-bot*. Additionally, *s-bots* are provided with a sound signalling system that can be used for communication. The task we study requires the *s-bots* to explore an arena presenting holes in which the robots may fall. Individual *s-bots* cannot avoid holes due to their limited perceptual apparatus. On the contrary, a *swarm-bot* can exploit the physical connections and the cooperation among its components in order to safely navigate in the arena.

This paper brings forth a twofold contribution. On the one hand, we examine different communication protocols among the robots (i.e. no signalling, handcrafted and evolved signalling), and we show that a completely



Fig. 1 A *swarm-bot* moving in an outdoor environment. The traction system of each *s-bot* is composed of both tracks and wheels. On top of it, a rotating turret is mounted, which holds many sensory systems and the rigid gripper for physical connections

evolved approach achieves the best performance. This result is in accordance with the above assumption, for which evolution potentially produces a system that is more efficient than those obtained with other conventional design methodologies. Another important contribution of this paper consists in the testing of the evolved controllers on physical robots. We show that the evolved controllers produce a self-organising system that is robust enough to be tested on real *s-bots*, notwithstanding the huge gap between simulation and reality. To the best of our knowledge, only very few works can be found in the literature in which cooperative evolved behaviours have been successfully tested on a group of physical robots (see, for example, Quinn et al. 2003; Kamimura et al. 2005). Considering the difficulty of the task we face and the complex dynamics involved, we believe that we obtained the most advanced evolved group behaviours so far successfully tested on a physical robotic platform.

This paper is organised as follows. In Sect. 2, we briefly overview the different communication forms that can be found in social insects and draw a parallel with collective robotics research. A taxonomy of different communication modalities is also introduced. In Sect. 3, we describe the *s-bot* features and its simulation. In Sect. 4, we introduce the task studied and detail the experimental setup used for evolving hole avoidance behaviours. Section 5 shows the obtained results in simulation, while Sect. 6 describes the results obtained in transferring the evolved controllers on the real *s-bots*. Finally, Sect. 7 concludes the paper.

2 Communication in social insects and robots

Insect societies abound in examples of self-organising behaviours (see Camazine et al. 2001). In most of these examples, communication is present in some elementary form. Hölldobler and Wilson point to 12 functional categories of communication in ants (see Hölldobler and Wilson 1990, p 227). This wide use of communication with different modalities is justified by the fact that communication serves as a regulatory mechanism of the activities of the colony. In the following, we discuss about the different communication modalities observed in social insects.

2.1 A glance at insect societies

From the study of mass communication modalities arises the concept of *stigmergy*: it describes an indirect communication among individuals, which is mediated by the environment. Stigmergy was first introduced by Grassé

(1959), while studying the nest building behaviour of termites of the genus *Macrotermes*. Grassé suggested that the cooperation among termites in their building activities was the result of environmental stimuli provided by the work already done, i.e. the nest itself. Other examples of stigmergic communication have been observed in the foraging behaviour of many ant species, which lay a trail of pheromone, thus modifying the environment in a way that can inform other individuals of the colony about the path to follow to reach a profitable foraging area (Goss et al. 1989; Hölldobler and Wilson 1990). Stigmergy is therefore a form of communication that is, in some way, indirect and mediated by the environment.

Stigmergy is not the only way of communication that can be observed in social insects. *Direct interactions*, i.e. a form of communication that involves some physical contact, account for various social phenomena (Hölldobler and Wilson 1990). For example, in many species of ants such as *Ecophilla longinoda*, recruitment of nest-mates for the exploitation of a food source is performed by touching the nest-mate with the antennae (*antennation*) and by regurgitating a sample of the food source (*throphallaxis*). Hölldobler and Wilson (1990) report another invitation behaviour during colony emigrations in ants of the species *Camponotus sericeus*. A recruiter ant invites another individual to follow it to a new nesting site by first grasping and pulling it by the mandibles. Afterwards, the recruiter turns around and moves toward the new site, while the other ant follows maintaining physical contact with its antennae. Mandible pulling and the subsequent *tandem running* are striking examples of coordination of movements that exploit direct interactions among individuals. Similar behaviours have been observed in other ant species, associated to recruitment for both colony emigration and foraging.

Some forms of *direct communication* within insect societies have been studied, a well-known example being the waggle dance of honey bees. A bee is able to indicate to the unemployed workers the direction and distance from the hive of a patch of flowers, using a “dance” that also gives information on the quality and the richness of the food source (Seeley 1995). Another form of direct communication takes places through acoustical signals. Many ant species use sound signals – called *stridulations* – as recruiting, alarm or mating signals. In the presence of a big prey, ants of the genus *Aphaenogaster* use stridulation during nest-mates recruitment. Here, the sound signal serves uniquely as a reinforcement of the usual chemical and tactile attractors, resulting in a faster response of the nest-mates. Another form of acoustic signalling is *drumming*, i.e. vibrations produced by strokes on the surface of chambers in wooden nests (Fuchs 1976). This signal serves as a direct alarm

communication and has a modulating effect on the probability of individual workers to respond to other signals.

2.2 From insects to robots

The above examples suggest a possible taxonomy of different forms of communication in insect societies that can be borrowed for characterising a collective robotic system (Trianni et al. 2004a):

Indirect or stigmergic communication. A form of communication that takes place through the environment, as a result of the actions performed by some individuals, which indirectly influence someone else's behaviour (e.g. pheromone trails).

Direct interaction. A form of communication that implies a non-mediated transmission of information, as a result of the actions performed by some individuals, which directly influence someone else's behaviour (e.g. antennation, mandibular pulling).

Direct communication. A form of communication that implies a non-mediated transmission of information, without the need of any physical interaction (e.g. the waggle dance, stridulations).

A number of other taxonomies for communication modalities in robotic systems have been proposed in the past (see, for example, Balch and Arkin 1994; Cao et al. 1997; Dudek et al. 2002; Matarić 1998). What we propose can be considered equivalent to the taxonomy introduced by Cao et al. (1997), having adapted it to the natural examples discussed above. The terminology we used is partly borrowed from Matarić (1998).

A pioneering work on the study of biologically inspired communication in collective robotics is the one of Balch and Arkin (1994). Three tasks and three different communicative setups were considered. Balch and Arkin show that direct communication is not required if the task is characterised by some form of indirect communication that provides the same amount of information. Additionally, they show that, among the direct communication strategies, a higher complexity does not forcedly result in an advantage. Stigmergy is the main coordination mechanism employed in many other works relevant for swarm robotics research (Beckers et al. 1994; Holland and Melhuish 1999). Finally, it is worth mentioning the work of Kube and Zhang (1997) and Kube and Bonabeau (2000) that show how a self-organising behaviour observed in ants (i.e. collective transport) can be replicated in a group of robots. In this case, the robotic experiments served as an empirical model useful to uncover some interesting features of the insect behaviour.

Direct interactions are not commonly exploited in robotic systems, as in general physical contacts among robots are preferably avoided or ignored. A remarkable exception has to be found in the SWARM-BOTS project,¹ in which self-assembling robots and cooperative strategies have been studied extensively, including the work presented in this paper (for some examples, see Groß et al. 2006; Baldassarre et al. 2004).

Simple forms of direct communication modalities are often chosen in collective robotics. Hayes et al. (2000) study how a simple binary communication can result in higher performance in a collective exploration task. Ijspeert et al. (2001) show how in a strictly collaborative task (i.e. a task in which cooperation is strictly required for goal achievement) a simple form of direct communication can enhance the performance of the system. Similar to the already mentioned work of Balch and Arkin (1994), Rybski et al. (2004) study the influence of different forms of communication on the performance of a collective robotic system in a foraging task.

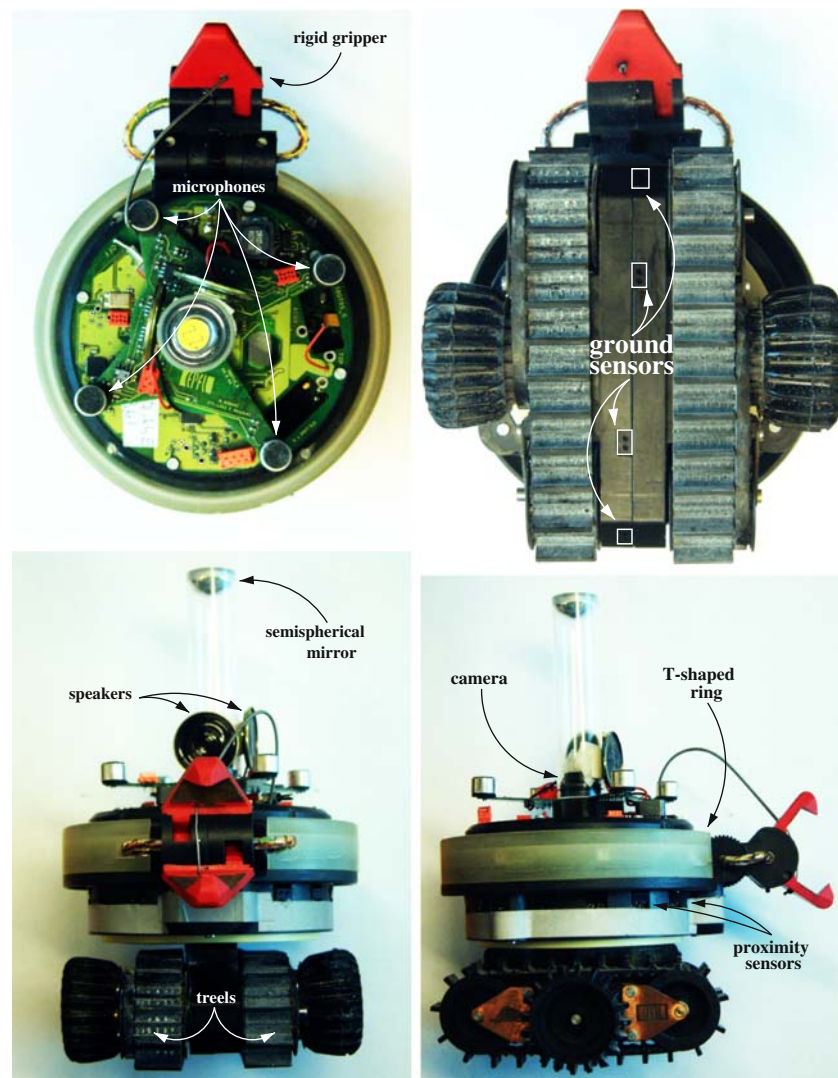
We conclude this short literature review mentioning some interesting work related to communication in an evolutionary robotics context. The pioneering work of Werner and Dyer (1991) studies evolution of communication strategies in a population of male and female artificial organisms selected for their ability to mate. More recently, Di Paolo (2000) has studied the evolution of communication between two simulated agents, whose goal was staying close to one another on the basis of acoustic signals only. Another example is given by Quinn (2001), who evolved a sort of communication strategy between two simulated robots for allocating the roles of leader and follower. All the above work in evolutionary robotics has been conducted in simulation. A remarkable exception is the work of Quinn et al. (2003), who studied the evolution of coordinated motion in a group of three simulated and physical robots. Also in this case, there is no explicit communication among the robots, but role allocation emerges from the initial interactions among the robots.

3 A self-organising artefact: the *swarm-bot*

A *swarm-bot* is a self-assembling, self-organising artefact. As mentioned above, the *swarm-bot* is a physical structure formed by a number of independent robotic units, called *s-bots*. In the *swarm-bot* form, the *s-bots* become a single robotic system that can move and reconfigure. Physical connections between *s-bots* are essential

¹ See <http://www.swarm-bots.org>.

Fig. 2 View of the *s-bot* from different sides. The main components are indicated (see text for more details)



for solving many collective tasks, such as the retrieval of a heavy object. Also, during navigation on rough terrain, physical links can serve as support if the *swarm-bot* has to pass over a hole larger than a single *s-bot* or when it has to pass through a steep concave region. However, for tasks such as searching for a goal location or tracing an optimal path to a goal, a swarm of unconnected *s-bots* can be more efficient. In the following, we describe in detail the *s-bot*'s features and the simulation model used for the experiments presented in this paper.

3.1 The *s-bot*

An *s-bot* is a small mobile autonomous robot with self-assembling capabilities, shown in Fig. 2 (Mondada et al. 2004). It weighs 700 g and its main body has a diameter of about 12 cm. Its design is innovative concerning both

sensors and actuators. The traction system is composed of both tracks and wheels, each track–wheel pair of a same side being controlled by a single motor. This combination of tracks and wheels provides the *s-bot* with a differential drive motion, which is labelled *Differential Treels*® Drive. The treels are connected to the chassis, which contains the batteries. The main body is a cylindrical turret mounted on the chassis by means of a motorised joint that allows the relative rotation of the two parts. Due to the power and control cables that connect chassis and turret, rotation of the turret must be limited in the range $[-\pi, \pi]$ rad. This constraint, hereafter referred to as *rotational limit*, must be taken into account in developing hole avoidance control strategies, as we detail in the following sections. The gripper is mounted on the turret. It can be used for connecting rigidly to other *s-bots* or to some objects. The shape of the gripper closely matches the T-shaped ring placed around the

s-bot's turret, so that a firm connection can be established. The gripper does not only open and close, but also has a degree of freedom for lifting the grasped objects. The corresponding motor is powerful enough to lift another *s-bot*.

An *s-bot* is provided with many sensory systems, useful for the perception of the surrounding environment or for proprioception. Infrared proximity sensors are distributed around the rotating turret and can be used for detection of obstacles and other *s-bots*. Four proximity sensors placed under the chassis, referred to as *ground sensors*, can be used for perceiving holes or the terrain's roughness (see Fig. 2). Additionally, an *s-bot* is provided with eight light sensors, two temperature/humidity sensors, a 3-axes accelerometer and incremental encoders on each degree of freedom.

Each robot is also equipped with sensors and devices to detect and communicate with other *s-bots*, such as an omni-directional camera, coloured LEDs around the *s-bots*' turret, microphones and loudspeakers (see Fig. 2). The loudspeaker can be used to emit a sound signal varying its frequency and intensity. The signal is perceived by the microphones and processed by the on-board CPU in order to discriminate the perceived frequency and intensity.

In addition to a large number of sensors for perceiving the environment, several sensors provide each *s-bot* with information about physical contacts, efforts and reactions at the interconnection joints with other *s-bots*. These include torque sensors on most joints as well as a *traction sensor* to measure the pulling/pushing forces exerted on the *s-bot*'s turret. The traction sensor is placed at the junction between the turret and the chassis. This sensor measures the direction (i.e. the angle with respect to the chassis orientation) and the intensity of the force of traction, hereafter called *traction*, that the turret exerts on the chassis. The turret of an *s-bot* physically executes a vectorial summation of the forces that are applied to it by other connected *s-bots*. The traction sensor plays an important role in the context of coordinated movement of a group of physically connected *s-bots*. In particular, it can be employed to provide the *s-bot* with an indication of the average direction towards which the *swarm-bot* is trying to move. Traction sensors are responsible for the detection of the direct interactions among *s-bots*. An *s-bot* can generate a traction force that is felt by the other *s-bots* connected through their grippers. This force mediates the communication among *s-bots* and can be exploited for coordinating the activities of the group: it proved to be important to evolve coordinated motion strategies in a *swarm-bot* and for collective obstacle and hole avoidance (see Baldassarre et al. 2004; Trianni et al. 2004a).

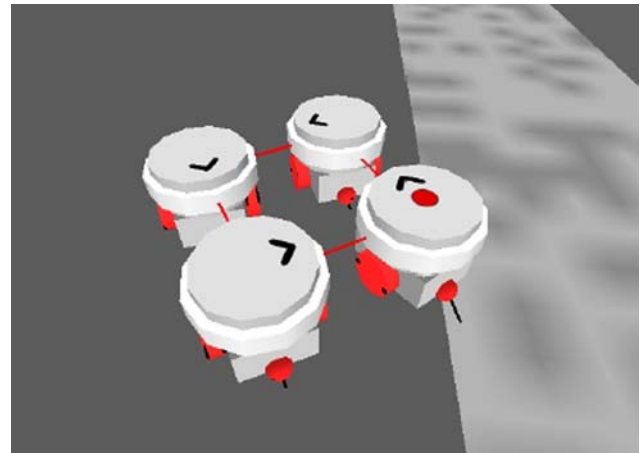


Fig. 3 A simulated *swarm-bot*, composed of four *s-bots* in square formation. The simulated *s-bot* is composed of four wheels, a chassis and a cylindrical turret (see text for details). The presence of a circle painted on top of the turret indicates that the *s-bot* is emitting a tone. The arrow on the turret indicates the position of the simulated gripper. The black line exiting from the chassis and pointing to the ground indicates a ground sensor

3.2 Simulating the *s-bot*

In order to design a controller for the *swarm-bot* through artificial evolution within a reasonable time, it is necessary to devise a simulation environment. In fact, evolution on the physical robots, besides being impractical, is extremely time-consuming.² We defined a simple *s-bot* model that at the same time allows fast simulations and preserves those features of the real *s-bot* that were important for the experiments (see Fig. 3). The developed software is based on Vortex™, a 3D rigid body dynamics simulator. Each simulation cycle corresponds to 0.1 real seconds, which provides an optimal compromise between simulation accuracy and speed. In each simulation cycle, every *s-bot* performs one control cycle, obtaining a frequency of 10 control cycles per second, which ensures a good reaction time for both simulated and physical robots.

The *s-bot*'s traction system was simulated by a chassis and four wheels. Two lateral, motorised wheels provide the required differential drive motion, modelling the lateral wheels of the treels system. Two spherical, passive wheels are placed in the front and the back and serve as support. The tracks of the treels system are not modelled, as they would have significantly reduced the simulation speed. These four wheels are fixed to the chassis, which also holds the cylindrical rotating turret that can rotate around its axis. Connections among

² One single evolutionary run, keeping the same setting we used in the experiments presented here, would require more than 110 days.

s-bots can be made dynamically, creating a joint between two *s-bots* (see Fig. 3). Differently from previous studies (see Trianni et al. 2004a), we modelled the rotational limit in the *s-bot* simulation, so that the turret can rotate only in the range $[-\pi, \pi]$ rad, consistently with the hardware counterpart. The implications of the rotational limit are explained in Sect. 4.1, in which we detail the different experimental choices we made for the evolution of behaviours that could be directly transferred on the physical *s-bots*.

Concerning the sensors, most of the physical ones have been modelled in the simulator. For the experiments presented in this paper, we mainly made use of traction and ground sensors. Four variables encode the traction force information from four different preferential orientations with respect to the chassis (front, right, back and left, see Baldassarre et al. 2004 for more details). Also the ground sensor configuration differs from what was previously used. Here it complies with the physical robot, the ground sensor being integral with the chassis and positioned in a line parallel to the main direction of motion, in the same position as shown in Fig. 2.³ Noise is simulated for all sensors, adding a random value uniformly distributed in the interval $[-5\%, 5\%]$ of the sensor saturation value.

Each *s-bot* is equipped with a loudspeaker and four microphones, used to detect the tone emitted by other *s-bots*. The speaker/microphone system is used to emit a single frequency signal that is recognised by the *s-bots* in a binary way: either one of the *s-bots* is signalling the presence of a hole – this could be the *s-bot* itself – or none of them are signalling. Therefore, sound perception or production is simulated by means of a binary variable that encodes the presence or absence of a sound signal.

4 Evolution of hole avoidance behaviours

The hole avoidance task has been defined for studying collective navigation strategies for a *swarm-bot* that moves in environments presenting holes in which it risks remaining trapped. In such a scenario, due to the limited sensory apparatus of the *s-bot*, the *swarm-bot* is more efficient than individual units. In fact, the position of the ground sensors makes it impossible for an *s-bot* to detect holes that are sidelong with respect to its direction of motion, because sensors are placed under its chassis and parallel to its tracks, as shown in Fig. 2. The *swarm-bot* can instead perform hole avoidance exploiting its larger physical structure and the cooperation among the *s-bots*.

³ In a previous work, the ground sensors were integral with the turret and distributed evenly around it (see Trianni et al. 2004a).

However, for a *swarm-bot* to perform hole avoidance, two main problems must be solved: (1) **coordinated motion** must be performed in order to obtain coherent movements of the *swarm-bot*, as a result of the actions of its components; (2) the presence of holes, which cannot be perceived by all the *s-bots* at the same time, must be communicated to the entire group, in order to trigger a change in the common direction of motion. In some preliminary studies, conducted in simulation only, we successfully evolved cooperative behaviours for the hole avoidance task (Trianni et al. 2004a, 2006). In this paper, we apply the same methodology to the evolution of behaviours that can be tested on the physical *s-bots*. In doing so, new challenges have to be faced, as the simulation model previously used was differing in some crucial aspects from the physical robot. In Sect. 4.1, we give a detailed description of the experimental choices made in order to cope with these challenges.

Moreover, in this paper we study and compare three different approaches to communication among the *s-bots*. In the first setup, *s-bots* communicate only through direct interactions, i.e. they exploit the pulling/pushing forces that one exerts on the other as a form of communication. This setup, referred to as *Direct Interactions* setup (DI), is the simplest possible for hole avoidance. The second and third setups make use of direct communication among the *s-bots* in addition to the direct interactions. In the second setup, referred to as *Direct Communication* setup (DC), the *s-bots* emit a tone as a handcrafted reflex action to the perception of a hole. On the contrary, in the third setup, which is referred to as *Evolved Communication* setup (EC), the signalling behaviour is not defined a priori, but it is left to evolution to shape the best communication protocol. In the following, we detail the experimental setup. Then, we describe the controllers and the evolutionary algorithm used, and finally we present the evaluation function defined for evolving hole avoidance behaviours.

4.1 Experimental setup

We aim at evolving hole avoidance behaviours for a group of four *s-bots* connected in a square formation. This formation was chosen in order to overcome the limitation in the perception of holes that pertains to individual *s-bots* or to *s-bots* connected forming a line. In fact, when connected in a square *swarm-bot*, the *s-bots* can detect the hole's edge and react timely notwithstanding their approaching direction.

The evolution of the hole avoidance behaviour also requires that the *swarm-bot* performs coordinated motion, i.e. the *s-bots* should prove capable of coherently moving on flat terrain. We therefore decided to let evo-

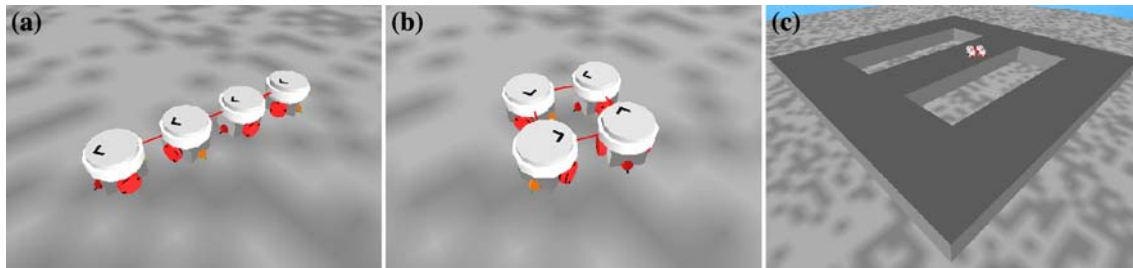


Fig. 4 Experimental conditions in which the *swarm-bot* is evolved. In conditions “a” and “b”, a *swarm-bot* is positioned on flat terrain and has to perform coordinated motion. The *swarm-*

bot shape is either a line or a square. In condition “c”, a square *swarm-bot* is positioned in an arena with open borders and holes

lution shape the neural controller testing the *swarm-bot* both in environments with and without holes (see Fig. 4). Concerning coordinated motion, Baldassarre et al. (2006) showed that robust controllers can be evolved if the *s-bots* move on a flat plane connected in a linear formation (see Fig. 4a). However, this setting alone may produce a sub-optimal behaviour, which becomes less probable if the *s-bots* are evaluated for coordinated motion connected in a square formation (see Fig. 4b). Concerning the evolution of hole avoidance, a square *swarm-bot* formation is placed in an arena presenting holes, as shown in Fig. 4c. The arena is a square of 4 m per side, with two rectangular holes and open borders. In all cases, the *s-bots* start connected in a *swarm-bot* formation, and the orientation of their chassis is randomly defined, so that they need to coordinate in order to choose a common direction of motion. In conditions “a” and “b”, once coordinated, the *s-bots* have to maintain straight motion as much as possible. In condition “c”, the *s-bots* have to explore the arena without falling into holes or out of the borders.

In all three setups (*DI*, *DC* and *EC*), *s-bots* are equipped with traction and ground sensors, as described in Sect. 3.2. In *DC* and *EC*, microphones and speakers are also used. The information provided to the controller by these sensors proved to be sufficient for the evolution of hole avoidance behaviours (Trianni et al. 2004a). However, we found that as soon as the rotational limit between turret and chassis is introduced, a perceptual aliasing problem arises (see also Baldassarre et al. 2006). In fact, the information about the angular displacement of the turret with respect to the chassis is missing, and the rotational limit can be recognised only referring to this displacement. Instead of providing this additional information to the neural controller, we decided to apply a different solution that can bypass the rotational limit. This solution, referred to as *front inversion mechanism*, was first introduced by Baldassarre et al. (2006), in order to mask the rotational limit to coordinated motion controllers evolved without taking it into ac-

count. Its working principle is very simple: whenever the turret reaches the rotational limit, the front of the *s-bot* is swapped with its back, which becomes the new principal front of motion. The front inversion involves both sensors and actuators, so that the *s-bot* ends up in a novel condition that prevents exceeding the rotational limit. A detailed explanation of the front inversion mechanism and related issues can be found in Appendix.

4.2 The controllers and the evolutionary algorithm

The *s-bots* are controlled by artificial neural networks, whose parameters are set by an evolutionary algorithm. A single genotype is used to create a group of *s-bots* with an identical control structure, a homogeneous group. Each *s-bot* is controlled by a fully connected, single layer feed-forward neural network, a perceptron (see Fig. 5). Each input is associated with a single sensor, receiving a real value in the range [0.0, 1.0], which is a simple linear

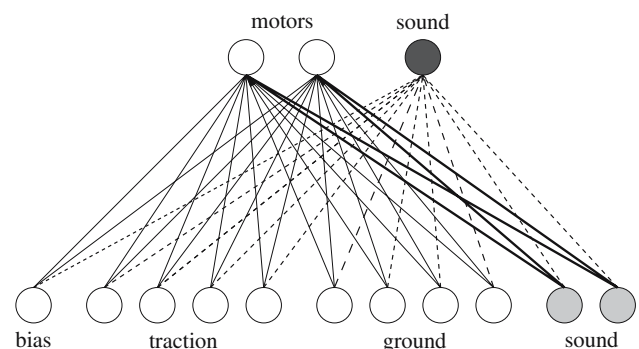


Fig. 5 The neural controller. Circles represent neurons, while lines represent weighted connections from input to output neurons. The empty circles and normal lines refer to neurons and connections used in the *DI* setup: the neural controller takes as input the traction and ground sensors, plus a bias, and it controls the two wheels and turret/chassis motor. The bold lines and light grey neurons are added in the *DC* setup: the neural controller also receives as input the perceived sound signals. The dashed lines and the dark grey neuron are further added in the *EC* setup: the neural network now controls the sound emitter also

scaling of the reading taken from its associated sensor. Additionally, the network is provided with a bias unit, i.e. an input unit whose activation state is clamped to 1. The activation of the output neurons is computed as the weighted sum of all input and bias units, filtered through a sigmoid function. The activations of the output neurons are real-valued numbers in the range $[0.0, 1.0]$ and are used to control the effectors of the *s-bot*.

In the basic *DI* setup, the traction and the ground sensors are used as inputs. Specifically, four inputs of the perceptron are dedicated to the traction sensor, encoding the traction force intensity and direction into four variables, as already mentioned in Sect. 3.2. Four other inputs are dedicated each to one ground sensor. Concerning the actuators, the two outputs of the perceptron are used to control the left and right wheels, scaling the activation values in the range $[-\omega_M, \omega_M]$, where $\omega_M \approx 3.6$ rad/s. Additionally, the same two outputs control the turret–chassis motor. The desired speed of the turret–chassis motor was set equal to the difference between the desired speed of the left and right wheels times a constant $k = r_w/2d_w$, where r_w is the radius of the wheels and d_w the distance between the two wheels. This setting leads to a movement of the turret with respect to the chassis that counter-balances the rotation of the chassis produced by the wheels' motion. This is useful to help the rotation of the chassis with respect to the turret when the *s-bots* are physically connected to one another.

In the *DC* setup, two additional binary inputs encode the information perceived by the microphones, as shown in Fig. 5. We use two inputs (instead of one) in order to cope with the rotational limit and the front inversion mechanism. One input is active when the *s-bot* uses the principal front, while the other is active when the *s-bot* is using the inverted front. In this way, it is possible to evolve controllers that can cope with the front inversion mechanism (see Appendix). These inputs are set to 1 if at least one *s-bot* is signalling, while they are set to 0 if no sound signal is perceived. The activation of the loudspeaker has been handcrafted in this setup, simulating a sort of reflex action: an *s-bot* activates the loudspeaker whenever one of its ground sensors detects the presence of a hole. Thus, the neural network does not control the emission of a sound signal. However, it receives the information coming from the microphones, and evolution is responsible for shaping the correct reaction to the perceived signals.

On the contrary, in the *EC* setup the sound emitter is controlled by an additional output added to the neural network, along with all the required connections (see Fig. 5). Whenever the activation of this additional neuron is greater than 0.5, a tone is emitted. Therefore, in

this setup evolution is responsible for shaping not only the response to the emission of a signal, but also the signalling behaviour. In other words, the complete communication paradigm – the signalling and the reaction to the perceived signal – is under the control of evolution.

The weights of the perceptron's connections are genetically encoded parameters. In all three setups a simple generational evolutionary algorithm is used. Initially, a random population of 100 genotypes is generated. Each genotype is a vector of binary values, 8 bits for each parameter. The genotype is composed of 144 bits for *DI*, 176 for *DC* and 264 for *EC*. Subsequent generations are produced by a combination of selection with elitism and mutation. Recombination is not used. At every generation, the best 20 genotypes are selected for reproduction, and each generates four offspring. The 80 offspring, each mutated with a 5% probability of flipping each bit, together with the 20 parents form the population of the subsequent generation. One evolutionary run lasts 200 generations.

4.3 Fitness evaluation

During evolution, a genotype is mapped onto a control structure that is cloned and downloaded onto all the *s-bots* taking part in the experiment (i.e. we make use of a homogeneous group of *s-bots*). Each genotype is evaluated 12 times, i.e. 12 trials. Each trial is characterised by a different seed for the initialisation of the random number generator, which influences both the initial position of the *swarm-bot* and the initial orientation of each *s-bot*'s chassis. Each trial lasts $T = 400$ control cycles, each corresponding to 0.1 simulated seconds. As already mentioned, we have defined three different conditions for the evolution of both coordinated motion and hole avoidance (see Fig. 4). Conditions “a” and “b” are intended to evolve robust coordinated motion strategies on flat terrain. Condition “c” is devoted to the evolution of hole avoidance. During evolution, the *swarm-bot* is initialised to one of these different conditions for 4 trials, thus obtaining 12 trials in total per genotype.

The behaviour produced by the evolved controller is evaluated according to a fitness function that takes into account only variables directly accessible to the *s-bots* (see Nolfi and Floreano 2000, p 73). In each simulation cycle t , for each *s-bot* s belonging to the *swarm-bot* S , the individual fitness $f_s(t)$ is computed as the product of three components:

$$f_s(t) = \omega_s(t) \Delta \omega_s(t) \gamma_s(t), \quad (1)$$

where:

- $\omega_s(t)$ accounts for fast motion of an *s-bot*. It is computed as the sum of the absolute values of the angular speed of the right and left wheels, linearly scaled in the interval $[0, 1]$:

$$\omega_s(t) = \frac{|\omega_{s,l}(t)| + |\omega_{s,r}(t)|}{2\omega_M}, \quad (2)$$

where $\omega_{s,l}(t)$ and $\omega_{s,r}(t)$ are, respectively, the angular speed of the left and right wheels of *s-bot* s at cycle t , and ω_M is the maximum angular speed achievable.

- $\Delta\omega_s(t)$ accounts for the straightness of the *s-bot*'s motion. It is computed as the difference between the angular speeds of the different wheels, as follows:

$$\Delta\omega_s(t) = \begin{cases} 0 & \text{if } \omega_{s,l}(t)\omega_{s,r}(t) < 0 \\ 1 - \sqrt{\frac{|\omega_{s,l}(t) - \omega_{s,r}(t)|}{\omega_M}} & \text{otherwise} \end{cases}. \quad (3)$$

This component is different from zero only when the wheels rotate in the same direction, in order to penalise any turning-on-the-spot behaviour. The square root is useful to emphasise small speed differences.

- $\gamma_s(t)$ accounts for coordinated motion and hole avoidance. It is computed as follows:

$$\gamma_s(t) = 1 - \max(\mathcal{F}_s(t), \mathcal{G}_s(t), \mathcal{S}_s(t)), \quad (4)$$

where $\mathcal{F}_s(t)$ is the intensity of the traction force perceived by the *s-bot* s at time t , $\mathcal{G}_s(t)$ the maximum activation among the ground sensors of *s-bot* s at time t and $\mathcal{S}_s(t)$ a binary value corresponding to 1 if *s-bot* s is emitting a tone at time t and 0 otherwise. This component favours coordinated motion as it is maximised when the perceived traction is minimised, which corresponds to a coherent motion of the *swarm-bot*. It also favours hole avoidance because it is maximised if the *s-bots* stay away from the holes. Finally, the component referring to the speaker has been designed to minimise the usage of direct communication, in order to signal only when it is necessary.

Given the individual fitness $f_s(t)$, the fitness F_θ of a trial θ is computed as follows:

$$F_\theta = \begin{cases} 0 & \text{if fall} \\ \frac{1}{T} \sum_{t=1}^T \min_{s \in S} f_s(t) & \text{otherwise} \end{cases}, \quad (5)$$

where T is the maximum number of simulation cycles. This fitness computation strongly penalises every fall of the *swarm-bot*, in order to evolve robust avoidance behaviours. However, given that many trials are performed on a flat plane, genotypes that result in a good coordinated motion strategy are still rewarded. Additionally, at each simulation cycle t we select the minimum among the individual fitnesses $f_s(t)$, which refers to the worst-performing *s-bot*, therefore obtaining a robust overall fitness computation. As a final remark, it is worth noting that in all the three setups the same evaluation function is used. Even if it may appear that the fitness evaluation has been designed explicitly for the *EC* setup, it ensures a fair comparison of the three setups. In fact, in *DI* sound is not used, so that $\mathcal{S}_s(t)$ is always 0, while in *DC* sound is used corresponding to the maximum activation of the ground sensors, so that both $\mathcal{S}_s(t)$ and $\mathcal{G}_s(t)$ are equal to 1, therefore the handcrafted emission of a tone is not penalised more than in the *EC* setup.

5 Results

For all setups – *DI*, *DC* and *EC* – the evolutionary experiments were replicated 10 times, so that 30 evolutionary runs have been performed on the whole. In all cases, a successful hole avoidance behaviour was evolved. The average performance of the best individuals of all evolutionary runs is close to 0.5, where a value of 1 should be understood as a loose upper-bound to the maximum value the fitness can achieve.⁴

5.1 Behavioural analysis

Looking at the behaviours produced by the evolved controllers, we observe no particular difference among the three setups for what concerns the initial coordination phase that leads to the coordinated motion of the *swarm-bot*. This is not surprising, because coordinated motion results mainly from the evaluation of the controllers on a flat terrain, namely, in conditions “a” and “b” shown in Fig. 4. In these conditions, the use of direct communication does not lead to any particular advantage, and the performance achieved by the three different setups is comparable. Therefore, in the following we describe the initial coordination phase referring to one particular controller evolved in the *DI* setup, as the other controllers produce similar behavioural strategies.

⁴ This maximum value could be achieved only if all *s-bots* start with their chassis already aligned in the same direction and always move in a flat environment, without holes.

Table 1 Average and standard deviation of the performance F of the best evolved controllers for each evolutionary run in the three different setups. For each controller, the percentage of falls is also shown

Run	DI setup		DC setup		EC setup	
	F	Falls (%)	F	Falls (%)	F	Falls (%)
1	0.43 ± 0.06	41	0.48 ± 0.06	0	0.51 ± 0.06	0
2	0.45 ± 0.07	33	0.50 ± 0.08	22	0.49 ± 0.06	2
3	0.43 ± 0.07	34	0.49 ± 0.06	2	0.50 ± 0.06	0
4	0.47 ± 0.07	56	0.47 ± 0.06	1	0.48 ± 0.08	1
5	0.44 ± 0.07	47	0.51 ± 0.06	1	0.50 ± 0.06	1
6	0.45 ± 0.07	37	0.50 ± 0.05	0	0.55 ± 0.06	2
7	0.44 ± 0.07	39	0.47 ± 0.06	0	0.53 ± 0.05	0
8	0.44 ± 0.06	41	0.48 ± 0.06	1	0.50 ± 0.06	1
9	0.46 ± 0.08	23	0.44 ± 0.08	7	0.51 ± 0.06	2
10	0.45 ± 0.08	30	0.50 ± 0.06	0	0.51 ± 0.06	0

At the beginning of a trial, the *s-bots* start to move in the direction in which they were initially positioned, resulting in a rather disordered overall motion. Within one or two control cycles the physical connections transform this disordered motion into traction forces that are exploited to coordinate the group. When an *s-bot* feels a traction force, it rotates its chassis so as to reduce this force. Once the chassis of all the *s-bots* are oriented in the same direction, the traction forces disappear and the coordinated motion of the *swarm-bot* starts (see also Baldassarre et al. 2004; Trianni et al. 2004a).

The differences between the three setups appear once the hole avoidance behaviour is considered. In the *DI* setup, *s-bots* can rely only on direct interactions in the form of traction forces in order to communicate the presence of a hole and consequently avoid falling into it. The *s-bot* that first detects a hole immediately inverts its direction of motion, and therefore produces a traction force that is perceived by the other *s-bots*. Exploiting this force, a new coordination phase is triggered, which results in a new direction of motion that leads the *swarm-bot* away from the hole. However, *s-bots* are not always capable of avoiding falling. In fact, the avoidance behaviour is based on a delicate balance of the forces involved – i.e. motors, traction and friction forces – which does not always ensure a prompt reaction to the detection of the hole.

A faster reaction to the detection of a hole is achieved in the *DC* and *EC* setups, in which *s-bots* have the possibility to exploit direct communication mediated by sound signals.⁵ This is always the case in all the controllers evolved in different evolutionary runs. In the *DC* setting, the activation of the speaker is handcrafted and corresponds to the perception of a hole with any of the ground sensors, while the response to this signal is

shaped by evolution. In most of the different evolutionary runs, the perception of the signal corresponds to the rotation on the spot of the chassis of all the *s-bots* but the one that perceives the hole. This latter one tries to move away from the arena border and, in doing so, it does not encounter much resistance from the others, until it ends up not detecting the hole any more. At this point, the signalling ceases and the group reorganises moving in a new direction.

The situation is much more complex for the *EC* setup. In fact, in this case evolution is in charge of shaping both the signalling mechanisms and the response to the perceived signals. It is very interesting to notice how evolution produced a variety of behaviours, all well adapted to the hole avoidance task. A detailed description of all the communication and behavioural strategies corresponding to the different evolutionary runs is out of the scope of this paper. It is anyway interesting to highlight some of the common points that characterise these behaviours, which seem to be the cause of the better performance achieved in this setup, as we show in the following:

1. Signalling is associated with the perception of a hole, similar to the *DC* setup. However, not all ground sensors are associated with a signalling behaviour, but only those corresponding to the direction of motion. In this way, *s-bots* do not influence each other if they perceive a hole while they are moving away from it.
2. The signalling behaviour is not only linked to the perception of a hole, but is influenced by other factors also, such as the traction force perceived and the perception of sound signals. In particular, in some cases, a high traction force inhibits the production of the signal. The adaptive function of this inhibition consists in the fact that in the absence of sound signals, the *s-bots* try to coordinate based on traction only,

⁵ Falls are also registered for these setups, even if much more sporadically than in the *DI* case (see also Table 1).

which may lead to a faster choice of a new direction of motion away from the hole.

- Similar to point 2, signal production is in some cases also inhibited by sound perception. In particular, when the perception of the self-emitted sound inhibits its production, an *s-bot* performs an alternate signalling, switching the loudspeaker on and off every control cycle. In this way, the *s-bots'* behaviour is influenced only in part.

The above mechanisms contribute to achieve a fast and reliable reaction to the perception of a hole, a reaction that in general results in an efficient avoidance.

From the qualitative analysis, the use of direct communication seems to confirm our expectations: it results in a faster reaction to the detection of a hole and therefore in a more efficient avoidance behaviour. Additionally, the evolved communication strategy appears more adaptive than the handcrafted solution. In order to assess the performance difference between the different setups, we performed a quantitative analysis, described in the following.

5.2 Quantitative analysis

We performed a post-evaluation analysis and compared the results obtained with the three setups. For each evolutionary run, we selected the best individual of the final generation and re-evaluated it 100 times. Each performance evaluation is the average of the fitness scored in three trials, one for each experimental condition encountered during evolution and shown in Fig. 4. In each trial, characterised by a different random initialisation, the performance is measured using Eq. (5). All individuals are tested against the same set of trials, using the same random initialisation. On the whole, the selected controllers are evaluated in 300 trials, obtaining 100 performance values that characterise their behaviour with respect to both coordinated motion and hole avoidance. A box-plot summarising the performance of these individuals is shown in Fig. 6. It is possible to notice that *EC* generally performs better than *DC* and *DI*, while *DC* seems to be generally better than *DI*.

Table 1 reports the average performances obtained from the post-evaluation analysis, along with the number of falls registered in condition “c”. This seems to confirm that the use of direct communication has a relevant effect on the performance. We can also notice that in the *DI* setup, the *swarm-bot* is often unable to avoid falling. In the other setups, the *swarm-bot* falls only sporadically.

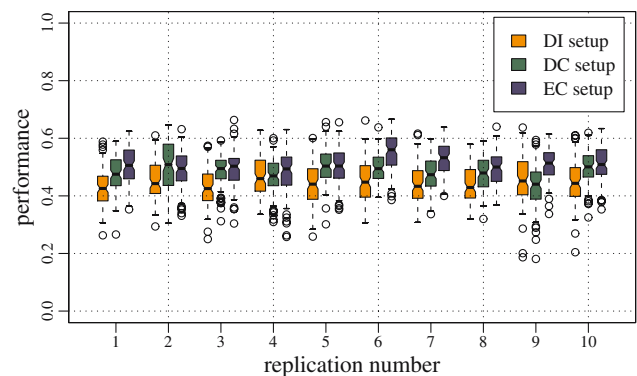


Fig. 6 Post-evaluation analysis of the best controller produced by all evolutionary runs of the three different setups. Boxes represent the inter-quartile range of the data, while the horizontal lines inside the boxes mark the median values. The whiskers extend to the most extreme data points within the inter-quartile range from the box. The empty circles mark the outliers

Table 2 Analysis of variance for the effect of the setups

	<i>df</i>	Partial SS	MS	<i>F</i>	<i>P</i>
Setups	2	1.823	0.911	279.43	<0.0001
Trials	99	4.153	0.042	12.86	<0.0001
Total	101	5.9760	0.059	18.14	<0.0001
Error	2,898	9.4525	0.003		

On the basis of these data, we performed a two-way analysis of variance (Montgomery 1997) to test if there is a significant difference in performance among the three setups. The analysis considers 3 factors (the setups), 100 blocks (the testing trials) and 10 replications for each combination of factor/block (the evolutionary runs). The applicability of the method was checked looking at the residuals coming from the linear regression modelling of the data: no violation of the hypothesis to use the analysis of variance was found. The result of the analysis, summarised in Table 2, allows us to reject the null hypothesis that there is no difference among the three setups.

The above analysis tells us that there is a statistical difference among the three setups, but it does not show which setup is different. Therefore, we performed pairwise Tukey’s tests among the three setups. The obtained results show with 99% confidence that the behaviours evolved within the *EC* setup perform significantly better than those evolved with both the *DI* and *DC* setups. The latter in turn results to be significantly better than the *DI* setup. We can conclude that the use of direct communication is clearly beneficial for hole avoidance. In fact, it speeds up the reaction to the detection of a hole and makes the avoidance action more reliable. We have also shown that evolving the communication protocol leads to a more adapted system. In the following,

Table 3 Results of the post-evaluation using the performance based on the integrated trajectory. Average performance and standard deviation are displayed for the best evolved controllers of

each evolutionary run. For each controller, the percentage of falls is also shown. The individuals chosen for transfer to the physical *s-bots* are displayed in bold

Run	<i>DI</i> setup		<i>DC</i> setup		<i>EC</i> setup	
	$\bar{\mathcal{T}}_\theta$	Falls (%)	$\bar{\mathcal{T}}_\theta$	Falls (%)	$\bar{\mathcal{T}}_\theta$	Falls (%)
1	0.10 ± 0.21	69	0.48 ± 0.28	5	0.63 ± 0.31	8
2	0.08 ± 0.15	62	0.19 ± 0.26	52	0.46 ± 0.31	13
3	0.10 ± 0.20	66	0.53 ± 0.30	5	0.48 ± 0.32	15
4	0.04 ± 0.12	76	0.32 ± 0.28	28	0.54 ± 0.33	10
5	0.10 ± 0.15	52	0.40 ± 0.27	14	0.43 ± 0.31	14
6	0.11 ± 0.20	61	0.49 ± 0.24	0	0.50 ± 0.28	8
7	0.12 ± 0.19	60	0.43 ± 0.25	0	0.58 ± 0.31	9
8	0.09 ± 0.18	63	0.54 ± 0.27	2	0.46 ± 0.32	14
9	0.28 ± 0.26	29	0.43 ± 0.32	21	0.42 ± 0.37	29
10	0.13 ± 0.24	63	0.56 ± 0.26	1	0.57 ± 0.30	5

we show how these behaviours can be efficiently transferred to the physical robots.

6 Transfer on physical *s-bots*

So far, we have shown how evolution can synthesise neural controllers that produce coordinated, cooperative behaviours in a group of simulated robots. We have also shown that evolution can shape the communication protocol in order to maximise the performance of the robotic system. In this section, we show how the controllers evolved in simulation can smoothly transfer to the real world. In order to do so, we first describe the methodology applied for choosing the individuals to test in reality. Then, we describe some issues related to the porting of the evolved controllers on physical robots. Finally, we present the results obtained with the physical robots and compare them with the simulation.

6.1 Selection of the controllers

In order to test the evolved behaviours on the physical robots, a choice had to be made among the available controllers, because testing all the best evolved neural networks in a sufficient number of trials would have been impractical and more time-consuming. We therefore decided to test a single controller per setup and compare its performance between simulation and reality.

We based the selection of the best controller on a different performance metric with respect to what was used during evolution. In fact, the function defined in Eq. (5) is a very conservative evaluation of the hole avoidance behaviour. It always takes into account the worst performing individual of the group and makes a product of measures that are based on individual sensor readings, which are affected by high levels of noise in the

real world. Therefore, when computed on data obtained from the physical robots, F_θ resulted in very low values, and a comparison with simulation results was not fair. The new performance metric \mathcal{T} gives a more informative measure of the controller's quality with respect to hole avoidance and allows making a fair comparison between simulation and reality. This performance metric corresponds to the distance covered by the *swarm-bot* and is computed integrating the trajectory covered by the centre of mass of the *s-bots* during a trial θ . This metric is computed as follows:

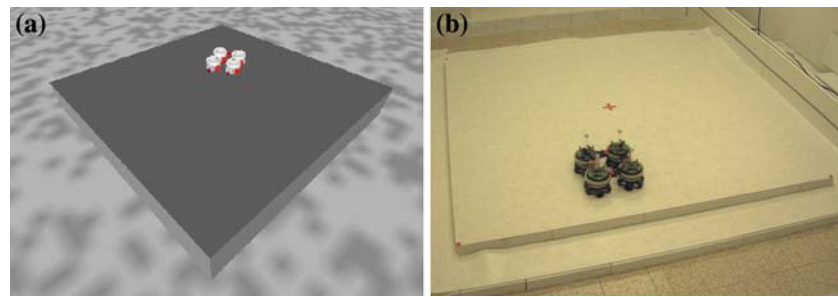
$$\mathcal{T}_\theta = \begin{cases} 0 & \text{if fall} \\ \frac{1}{D_M} \sum_{t=1}^T \|\mathbf{X}_S(t) - \mathbf{X}_S(t-1)\| & \text{otherwise} \end{cases}, \quad (6)$$

where $\mathbf{X}_S(t)$ is the coordinate vector of the centre of mass of the *swarm-bot* S at cycle t , T the number of control cycles performed and D_M the maximum distance that an *s-bot* can cover moving straight at maximum speed in T control cycles.

Using Eq. (6), we performed a post-evaluation analysis of all the best controllers evolved in the 30 evolutionary runs. The *swarm-bot* was put in a small square arena, its side measuring 180 cm, shown in Fig. 7a. A real version of this arena was built, making the comparison between simulation and reality possible (see Fig. 7b). The results obtained from the post-evaluation are summarised in Table 3. Both the average performance and the number of times the *swarm-bot* fell out of the arena are shown.⁶ It is possible to notice that the number of

⁶ Using these data we performed the same statistical analysis described in Sect. 5.2, and also in this case we obtained a significant difference among the setups, confirming that *EC* is the best setup, followed by *DC* and *DI* (data not shown).

Fig. 7 The square arena used for the comparison between simulation and physical *s-bots*. **a** The simulated arena; **b** the real arena



falls is rather high for the *DI* setup, and in general much lower for the *DC* and *EC* setups.

The choice of the best controller for each setup should be based on its performance. However, other factors are also relevant when considering porting on real robots. In our case, we were mainly interested in avoiding damage to the *s-bots*, therefore we decided to select those controllers that resulted in the least number of falls. In case of multiple possibilities, as for the *DC* setup, a choice based on the highest mean performance has been performed. Consequently, we chose the controllers evolved in the ninth, sixth and tenth evolutionary runs, respectively, for the *DI*, *DC* and *EC* setups.

6.2 Issues in porting on physical *s-bots*

The neural network controller is used on the real *s-bots* exactly in the same way as in simulation. The values returned by the various sensors are read every 100 ms, they are scaled in the range [0,1] and finally fed to the neural network. The outputs of the network are used to control the wheels and the turret–chassis motor. There are only two differences compared to the simulation. First of all, an exponential moving average is applied to the outputs of the neural network that controls the wheels and the turret–chassis motor:

$$\omega(t) = \tau y(t) + (1 - \tau)\omega(t - 1), \quad (7)$$

where $\omega(t)$ is the desired angular speed of the wheels at time t , $y(t)$ the set-point defined by the neural controller and $\tau = 0.8$ is the time constant used. This average is required to avoid damage to the robots if the network output varies too much, and it adds to the smoothing of the wheels' speed performed by the PICTM controller of the motors. Moreover, we added a recovery function that is necessary to avoid damage of the *s-bots* due to excessive efforts by the motors of the wheels. This function constantly monitors the torque applied by the motors of the left and right wheels, and in case the torque exceeds a given threshold for a long time, the speed of the wheels is set to 0. Both these modifications make the system somewhat less reactive to external stimuli,

but they are required in order to avoid an excessive strain of the motors.

No parameter tuning was required except for the maximum traction force \mathcal{F}_M . This parameter is used for scaling the readings $\mathcal{R}(t)$ of the traction sensor:

$$\mathcal{F}(t) = \begin{cases} -1 & \text{if } \mathcal{R}(t) < -\mathcal{F}_M \\ \frac{\mathcal{R}(t)}{\mathcal{F}_M} & \text{if } |\mathcal{R}(t)| \leq \mathcal{F}_M \\ 1 & \text{if } \mathcal{R}(t) > \mathcal{F}_M \end{cases}, \quad (8)$$

where $\mathcal{F}(t)$ is the normalised value fed to the neural controller. The optimal value of \mathcal{F}_M depends on the neural controller, the individual properties of the *s-bots* (level of noise, effective power of the motors) and the friction coefficient of the ground, which can vary due to dust or humidity. Therefore, we tuned this parameter independently for each neural controller in order to maximise its performance.

6.3 Results

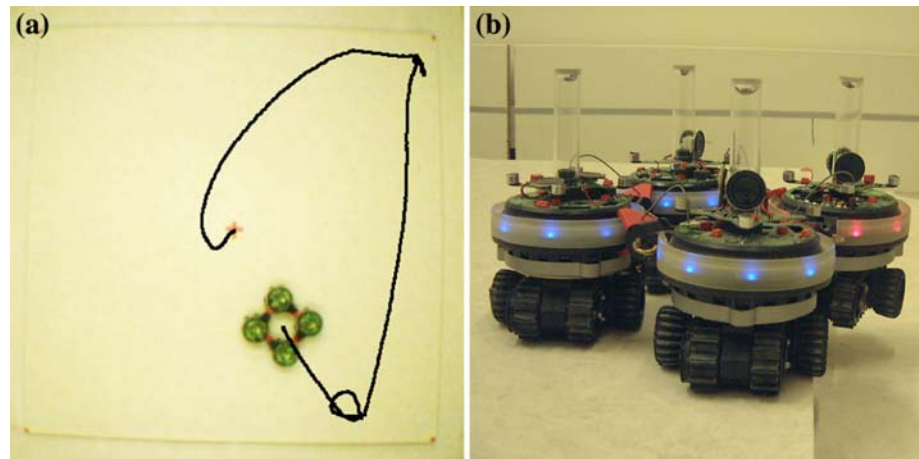
Each selected controller was evaluated in 30 trials, always starting with a different random initialisation. A square *swarm-bot* was placed in the centre of the square arena shown in Fig. 7b. The behaviour of the *swarm-bot* was recorded using an overhead camera and its trajectory obtained using the tracking software SwisTrack,⁷ which proved to be a valuable tool for tracking a swarm of robots (Correll and Martinoli 2006). Figure 8a shows an example of the trajectory extracted using the tracking software. The obtained data were used to compute the performance of the system using Eq. (6).

Qualitatively, the behaviour produced by the evolved controllers tested on the physical *s-bots* is very good and closely corresponds to that observed in simulation⁸ (see Fig. 8). *S-bots* coordinate more slowly in reality than in simulation, taking a few seconds to agree on a common

⁷ A software developed by the Swarm-Intelligent Systems Group, EPFL, <http://swistrack.sourceforge.net>.

⁸ Movies of these behaviours are available in the electronic supplementary material.

Fig. 8 **a** View of the arena taken with the overhead camera. The *black line* corresponds to the trajectory of the *swarm-bot* in a trial lasting 900 control cycles. **b** A physical *swarm-bot* while performing hole avoidance. It is possible to notice how physical connections among the *s-bots* serve as support when a robot is suspended out of the arena, still allowing the whole system to work. Notwithstanding the above difficult situation, the *swarm-bot* was able to avoid falling



direction of motion, also due to the smoothing of the wheel speed discussed in Sect. 6.2. Some problems are caused by the front inversion mechanism, which sometimes leads to a loss of coordination, due mainly to the high friction of the treels system. Hole avoidance is also performed with the same modalities as observed in simulation. With the *DI* controller, the combination of tracks and wheels of the traction system brings an advantage compared to the results recorded in simulation, in which the tracks are not modelled (see Sect. 3.2). In fact, differently from what happens in simulation, the *s-bot* that perceives the hole can produce a traction force even if it is nearly completely suspended out of the arena. Moreover, the higher friction provided by the tracks allows to produce higher traction forces that can have a greater influence on the behaviour of the rest of the group. Similarly, the treels system is advantageous for the *DC* controller, in which the *s-bot* perceiving the holes pushes the other *s-bots* away from the arena border while emitting a sound signal. Concerning the *EC* controller, on the contrary, the treels system does not lead to a clear advantage from a qualitative point of view. In both *DC* and *EC*, we recorded some communication failures, in which an *s-bot* misses either to switch on the loudspeaker or to perceive an emitted signal. In particular, in the *EC* setup, failures are more frequent whenever an *s-bot* tries to switch the loudspeaker on and off at a high pace.

From a quantitative point of view, it is possible to recognise some differences between simulation and reality, as shown in Fig. 9 and Table 4. We compare the performance T_θ recorded in 100 trials in simulation with the one obtained from the 30 trials performed in reality. Generally, we observe a decrease in the maximum performance, mainly due to a slower coordination among the *s-bots*. This means that physical *s-bots* start moving coordinately later than the simulated ones, both at the

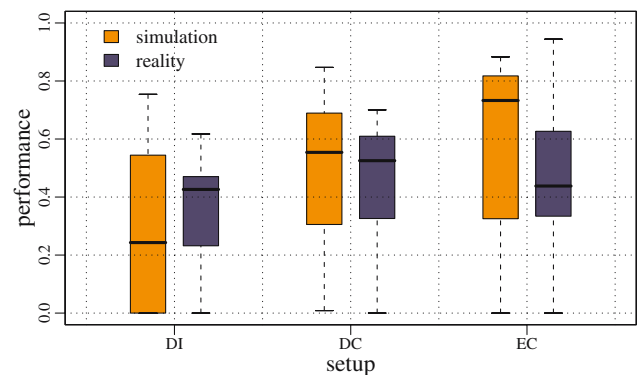


Fig. 9 Comparison of the performance produced in the different settings by the selected controllers tested both in simulation and reality. For an explanation of the plot, see Fig. 6

beginning of a trial and after the perception of a hole. This influences the performance, as the *swarm-bot* cannot cover long distances until coordination among the *s-bots* is achieved.

Looking at Fig. 9 and Table 4, we can notice that the performance of the *DI* controller is better in reality, thus confirming the qualitative analysis for which the treels system allows to enhance the direct interactions among the *s-bots*, therefore leading to a better avoidance behaviour. This is also confirmed by the percentage of falls, which is lower in reality than in simulation. Concerning the *DC* controller, the performance difference between simulation and reality is minimal. In this case, we observed that the possible performance drop due to communication failures was compensated for by the higher force transmitted from one *s-bot* to the other due to the high friction of the treels system. Here, only one fall was observed out of the 30 trials performed. On the contrary, the best controller of the *EC* setup does not perform as well in reality as in simula-

Table 4 Average and standard deviation of the performance obtained by the selected controllers tested both in simulation and reality. The percentage of falls is also shown

	DI setup		DC setup		EC setup	
	\mathcal{T}_θ	Falls (%)	\mathcal{T}_θ	Falls (%)	\mathcal{T}_θ	Falls (%)
Simulation	0.28 ± 0.26	29	0.49 ± 0.24	0	0.57 ± 0.30	5
Reality	0.33 ± 0.20	20	0.47 ± 0.18	3.3	0.45 ± 0.21	6.6

tion. *S-bots* are always able to coordinate and perform coordinated motion and hole avoidance. However, we observe here that *s-bots* are slower in avoiding holes due mainly to some failures in the communication system, which is fundamental to trigger and support the avoidance action. For this reason, quantitatively the performance decreases. However, the behaviour is altogether good, and the percentage of falls is in line with the results obtained in simulation, as shown in Table 4.

7 Conclusions

The definition of collective behaviours based on self-organisation is a problem of particular interest for many researchers. In this paper, we show that through artificial evolution it is possible to synthesise controllers that achieve very good performance in simulation and that can be smoothly ported to physical robots.

We have shown that the use of direct communication among the *s-bots* is particularly beneficial in the case of hole avoidance. It is worth noting that direct communication acts here as a reinforcement of the direct interactions among the *s-bots*. In fact, *s-bots* react faster to the detection of the hole when they receive a sound signal, without waiting to perceive a traction strong enough to trigger the hole avoidance behaviour. However, traction is still necessary for avoiding the hole and coordinating the motion of the *swarm-bot* as a whole. Additionally, the statistical analysis of the results obtained in simulation showed that the completely evolved setup outperforms the setup in which direct communication is handcrafted. This result is in our eyes particularly significant because it shows that artificial evolution can synthesise solutions which would be very hard to design with conventional approaches. In fact, the most effective solutions discovered by evolution exploit some interesting mechanisms for the inhibition of communication that would have been difficult to devise without any a priori knowledge of the system's dynamics.

The neural controllers synthesised by artificial evolution proved to be robust enough to be tested on physical robots, notwithstanding the huge gap between the simulation model used for the evolution and the actual *s-bot*.

The neural controller produced a qualitatively equivalent behaviour to what was observed in simulation. The performance measured in the real world was somewhat affected by various factors, but the difference with simulation was never higher than 20% on average. We can therefore conclude that we succeeded in transferring an evolved self-organising behaviour from simulated to physical *s-bots*. To the best of our knowledge, no other comparably advanced behaviour has been evolved in simulation and successfully tested on physical robots.

In future work, we will continue the development of self-organising, communicating behaviours for the *swarm-bot* and, possibly, other robotic platforms. Mainly, two issues attract our attention: on the one hand, we are interested in studying the formation of a physically assembled structure, i.e., a *swarm-bot*, as an adaptive response to adverse environmental contingencies that prevent the single individuals, i.e. the *s-bots*, to accomplish their task. This process goes under the name of *functional self-assembly* (Trianni et al. 2004b): The design of suitable controllers for functional self-assembly is a problem still at an early stage of study approaching it using evolutionary techniques is our main goal. On the other hand, we are interested in the evolution of communication for synchronisation of the activities of a group. As in the experiments presented in this paper, the evolution of communication should be functional to the achievement of a particular goal, e.g. efficiently avoiding holes. In the case of synchronisation of the group activities, communication should emerge as a means for reducing waste of energy at the individual level when strictly cooperative actions are required.

Acknowledgements This work was supported by the SWARM-BOTS project and by the ECagents project, two projects funded by the Future and Emerging Technologies programme (IST-FET) of the European Commission, under grant IST-2000-31010 and 001940, respectively. The information provided is the sole responsibility of the authors and does not reflect the Community's opinion. The Community is not responsible for any use that might be made of the data appearing in this publication. Vito Trianni thanks Thomas Halva Labella, Shervin Nouyan, Michael Bonani, Anders Lyhne Christensen, Roderich Groß, Sara Mitri, Elio Tuci and Francesco Mondada for their help, the stimulating discus-

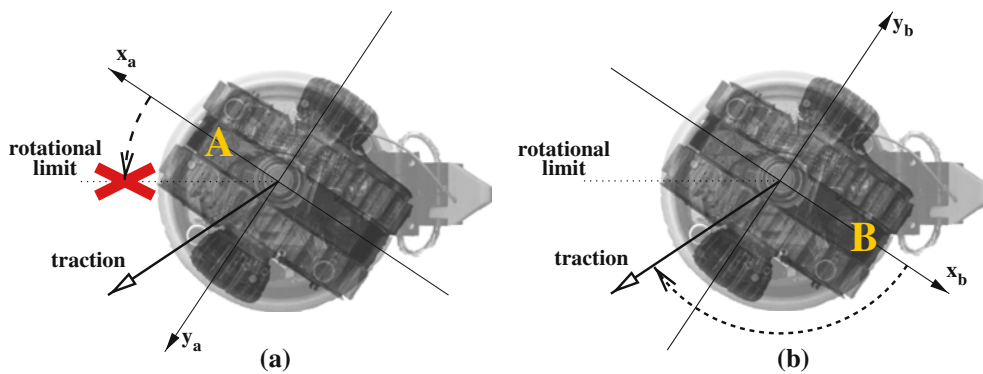


Fig. 10 The front inversion mechanism is explained using an image of the *s-bot*'s as seen from the top. The turret is made transparent in order to show the chassis also. **a** The *s-bot* is using the front **A**, therefore inputs to and outputs from the controller

are relative to the reference frame $x_a y_a$ integral with the chassis. **b** The *s-bot* is using front **B**. In this case, inputs to and outputs from the controller must be relative to the reference frame $x_b y_b$. See text for more details

sions and for their feedback during the preparation of this paper. The authors also thank Nikolaus Correll and Alcherio Martinoli for providing the tracking software used in the experiments presented in this paper. Marco Dorigo acknowledges support from the Belgian FNRS, of which he is a Research Director, through the grant "Virtual Swarm-bots", contract no. 9.4515.03, and from the "ANTS" project, an "Action de Recherche Concertée" funded by the Scientific Research Directorate of the French Community of Belgium.

Appendix

The front inversion mechanism

Suppose that an *s-bot* is driven by a controller that rotates the chassis toward the traction direction, covering the minimum angular distance. This controller takes as input the intensity and direction of the traction force computed from a reference frame integral with the chassis. Moreover, this controller does not take into account the rotational limit. Suppose also that the *s-bot* finds itself in the situation depicted in Fig. 10a: the chassis is oriented in the direction indicated as **A** and a traction force is perceived as indicated. Driven by its controller, the *s-bot* rotates the chassis counterclockwise, but it encounters the rotational limit and gets stuck. Now, suppose that the traction force stays the same, while the chassis is oriented in the opposite direction, indicated by **B** in Fig. 10b. In this case, the controller rotates the chassis clockwise and reaches the desired position without encountering the rotational limit.

In the situation depicted in Fig. 10, **A** and **B** correspond to the two directions – hereafter called *fronts* – of the *s-bot*'s chassis: one corresponds to forward motion, the other to backward motion. The symmetry of the chassis allows to make no distinction between these two fronts. The front inversion mechanism consists in

swapping front **A** with front **B** and vice versa every time the rotational limit is encountered. With respect to the above example, when the *s-bot* is in the situation depicted in Fig. 10a, it is exploiting the front **A** as main direction of motion and turns counterclockwise, until the rotational limit is encountered. At this point, the front inversion mechanism swaps the fronts, so that the *s-bot* exploits front **B** as the main direction of motion. As the traction force comes now from the left, the *s-bot* rotates clockwise and reaches the desired orientation.

Technically, inverting the front from **A** to **B** or vice versa involves a 180° rotation of the chassis' reference frame, therefore passing from $x_a y_a$ to $x_b y_b$, as in Fig. 10. The inputs of the controller must be computed referring to the new reference frame. In particular, the traction encoding must be inverted:

$$\mathcal{F}_b = -\mathcal{F}_a,$$

where \mathcal{F}_a is the traction as perceived by the traction sensor and \mathcal{F}_b the value fed to the controller. If other sensors are used, their readings must be swapped with respect to both x and y axes before using them as input to the controller.⁹ Concerning the wheels, using front **B** instead of front **A** requires that the controller outputs are inverted as well:

$$\omega_{b,l} = -\omega_{a,r},$$

$$\omega_{b,r} = -\omega_{a,l},$$

where $\omega_{a,-}$ is the angular speed defined by the controller, while $\omega_{b,-}$ is the angular speed set to the wheel.

The precondition for the application of the front inversion mechanism is the central symmetry of the sen-

⁹ This applies to ground sensors in the experiments presented in this paper.

sory motor equipment, because it allows a 180° rotation of the reference frame. Moreover, the controller must be somewhat “symmetric” itself: in the inverted condition, the controller should produce an action that is opposite with respect to the non-inverted condition. For example, a controller that rotates the chassis clockwise for every perceptual condition is not symmetric. In such a case, swapping the fronts does not lead to any advantage. A symmetric controller would turn counterclockwise when using **A** and clockwise when using **B** for a given perceptual state, similar to the situation depicted in Fig. 10. Notice that the controller does not have to be perfectly symmetric, but it is sufficient that it results in a “qualitatively” symmetric action with respect to symmetric perceptual conditions.

The necessity of having a symmetric controller when using the front inversion mechanism justifies the introduction of two sound inputs in the *DC* and *EC* setups (instead of adding only one), as mentioned in Sect. 4.2. Having a single input would lead to a single action, no matter which front is used. For example, if the response to a perceived signal is a clockwise turn, it would not change when inverting the fronts. Therefore, we make use of two inputs, which are alternately set whether the active front is **A** or **B**. This allows to obtain a symmetric behaviour with respect to the perception of sound signals (i.e. a clockwise rotation using **A** and a counterclockwise rotation using **B**). Finally, it is worth noting that it is up to the evolutionary algorithm to synthesise a symmetric controller: in fact, if the controller were not symmetric, the front inversion mechanism would not work properly and the *s-bot* would get stuck on the rotational limit.

References

- Balch T, Arkin RC (1994) Communication in reactive multiagent robotic systems. *Auton Robots* 1(1):27–52
- Baldassarre G, Parisi D, Nolfi S (2004) Coordination and behaviour integration in cooperating simulated robots. In: Schaal S, Ijspeert A, Billard A, Vijayakumar S, Hallam J, Meyer J.-A (eds) From animals to animats VIII. Proceedings of the 8th international conference on simulation of adaptive behavior. MIT Press, Cambridge, pp 385–394
- Baldassarre G, Parisi D, Nolfi S (2006) Distributed coordination of simulated robots based on self-organisation. *Artif Life* pp 2558–2564
- Beckers R, Holland OE, Deneubourg J-L (1994) From local actions to global tasks: stigmergy and collective robotics. In: Brooks RA, Maes P (eds) Proceedings of the 4th international workshop on the synthesis and simulation of living systems (artificial life IV). MIT Press, Cambridge, pp 181–189
- Beni G, Wang J (1989) Swarm intelligence. In: Proceedings of the seventh annual meeting of the Robotics Society of Japan, RSJ Press, Tokyo, pp 425–428
- Bonabeau E, Dorigo M, Theraulaz G (1999) Swarm intelligence: from natural to artificial systems. Oxford University Press, New York
- Camazine S, Deneubourg J-L, Franks N, Sneyd J, Theraulaz G, Bonabeau E (2001) Self-organization in biological systems. Princeton University Press, Princeton
- Cao YU, Fukunaga AS, Kahng AB (1997) Cooperative mobile robotics: antecedents and directions. *Auton Robots* 4:1–23
- Correll N, Martinoli A (2006) Collective inspection of regular structures using a swarm of miniature robots. In: Ang MH Jr, Khatib O (eds) Experimental Robotics IX. Proceedings of the ninth international symposium on experimental robotics (ISER-04). vol 21 Springer tracts in advanced robotics, Springer, Berlin Heidelberg New York, pp 375–385
- Di Paolo EA (2000) Behavioral coordination, structural congruence and entrainment in a simulation of acoustically coupled agents. *Adapt Behav* 8(1):25–46
- Dorigo M, Şahin E (2004) Swarm robotics – special issue editorial. *Auton Robots* 17(2–3):111–113
- Dorigo M, Stützle T (2004) Ant colony optimization. MIT Press/Bradford Books, Cambridge
- Dorigo M, Trianni V, Şahin E, Groß R, Labella TH, Baldassarre G, Nolfi S, Deneubourg J-L, Mondada F, Floreano D, Gambardella LM (2004) Evolving self-organizing behaviors for a swarm-bot. *Auton Robots* 17(2–3):223–245
- Dudek G, Jenkin M, Miliot E (2002) A taxonomy of multirobot systems. In: Balch T, Parker LE (eds) Robot teams: from diversity to polymorphism. AK Peters Wellesley
- Fuchs S (1976) The response to vibrations of the substrate and reactions of specific drumming in colonies of carpenter ants (*Camponotus*, Formicidae, Hymenoptera). *Beha Ecol and Sociobiol* 1(2):155–184
- Goss S, Aron S, Deneubourg J-L, Pasteels JM (1989) Self-organized shortcuts in the argentine ant. *Naturwissenschaften* 76:579–581
- Grassé PP (1959) La reconstruction du nid et les coordinations interindividuelles chez *Bellicositermes natalensis* et *Cubitermes* sp. La théorie de la stigmergie: Essai d'interprétation du comportement des termites constructeurs. *Insectes Sociaux* 6: 41–81
- Groß R, Tuci E, Bonani M, Mondada F, Dorigo M (2006) Object transport by modular robots that self-assemble. In: Proceedings of the 2006 IEEE international conference on robotics and automation (ICRA'06). IEEE Computer Society Press, Los Alamitos pp 2558–2564
- Harvey I, Husbands P, Cliff D (1992) Issues in evolutionary robotics. In: Meyer J.-A, Roitblat H, Wilson S (eds) Proceedings of the 2nd international conference on simulation of adaptive behavior. MIT Press, Cambridge, pp 364–373
- Harvey I, Di Paolo EA, Wood R, Quinn M, Tuci E (2005) Evolutionary robotics: a new scientific tool for studying cognition. *Artif Life* 11(1–2):79–98
- Hayes AT, Martinoli A, Goodman RM (2000) Comparing distributed exploration strategies with simulated and real autonomous robots. In: Parker LE, Bekey G, Bahren J (eds) Proceedings of the fifth international symposium on distributed autonomous robotic systems (DARS-00) Springer, Berlin Heidelberg New York, pp 261–270
- Holland O, Melhuish C (1999) Stigmergy, self-organization, and sorting in collective robotics. *Artif Life* 5(2):173–202
- Hölldobler B, Wilson EO (1990) The ants. Belknap Press Harvard University Press, Cambridge
- Ijspeert AJ, Martinoli A, Billard A, Gambardella LM (2001) Collaboration through the exploitation of local interactions in autonomous collective robotics: the stick pulling experiment. *Auton Robots* 11 (2):149–171

- Kamimura A, Kurokawa H, Yoshida E, Murata S, Tomita K, Kokaji S (2005) Automatic locomotion design and experiments for a modular robotic system. *IEEE/ASME Trans on Mechatron* 10 (3):314–325
- Kube CR, Bonabeau E (2000) Cooperative transport by ants and robots. *Robot Auton Syst* 30(1–2):85–101
- Kube CR, Zhang H (1997) Task modelling in collective robotics. *Auton Robots* 4:53–72
- Matarić MJ (1998) Using communication to reduce locality in distributed multiagent learning. *J Exp Theor Arti Intell Spe Issue Learn DAI Syst* 10 (3):357–369
- Mondada F, Pettinaro GC, Guignard A, Kwee IV, Floreano D, Deneubourg JL, Nolfi S, Gambardella LM, Dorigo M (2004) SWARM-BOT: a new distributed robotic concept. *Auton Robots* 17(2–3):193–221
- Montgomery DC (1997) Design and analysis of experiments, Wiley New York, 5th edn.
- Nolfi S, Floreano D (2000) Evolutionary robotics: the biology, intelligence, and technology of self-organizing machines. MIT Press/Bradford Books, Cambridge
- Quinn M (2001) Evolving communication without dedicated communication channels. In: Kelemen J, Sosik P (eds) *Advances in artificial life: sixth european conference on artificial life (ECAL2001)* Springer, Berlin Heidelberg New York, pp 357–366
- Quinn M, Smith L, Mayley G, Husbands P (2003) Evolving controllers for a homogeneous system of physical robots: structured cooperation with minimal sensors. *Philos Trans R Soc Lond Ser A Math Phys Eng Sci* 361:2321–2344
- Rybski P, Larson A, Veeraraghavan H, LaPoint M, Gini M (2004) Communication strategies in multi-robot search and retrieval: experiences with MinDART. In: Alami R (ed) *Proceedings of the seventh international symposium on distributed autonomous robotic systems (DARS-04)*, Toulouse, France, 23–25 June 2004 pp 301–310
- Seeley T (1995) *The wisdom of the hive*. Harvard University Press, Cambridge
- Trianni V, Labella TH, Dorigo M (2004) Evolution of direct communication for a swarm-bot performing hole avoidance. In: Dorigo M, Birattari M, Blum C, Gambardella LM, Mondada F, Stützle T (eds) *Ant colony optimization and swarm intelligence – proceedings of ANTS 2004 – fourth international workshop vol 3172*. Lecture notes in computer science Springer, Berlin Heidelberg New York, pp 131–142
- Trianni V, Tuci E, Dorigo M (2004b) Evolving functional self-assembling in a swarm of autonomous robots. In: Schaal S, Ijspeert A, Billard A, Vijayakamur S, Hallam J, Meyer J-A (eds) *From animals to animats 8. Proceedings of the eighth international conference on simulation of adaptive behavior (SAB04)*, MIT Press, Cambridge pp 405–414
- Trianni V, Nolfi S, Dorigo M (2006) Cooperative hole avoidance in a swarm-bot. *Robot Auton Syst* 54(2):97–103
- Werner GM, Dyer MG (1991) Evolution of communication in artificial organisms. In: Langton C, Taylor C, Farmer D, Rasmussen S (eds) *Artificial life II. vol X SFI Studies in the science of complexity*, Addison-Wesley, Redwood City, pp 659–687