Strengths and Synergies of Evolved and Designed Controllers: A Study within Collective Robotics

Gianluca Baldassarre **Stefano Nolfi**
Laboratory of Autonomous Robotics and Artificial Life,
Istituto di Scienze e Tecnologie della Cognizione,
Consiglio Nazionale delle Ricerche (LARAL-ISTC-CNR),
Via San Martino della Battaglia 44, 00185 Roma, Italy
gianluca.baldassarre@istc.cnr.it stefano.nolfi@istc.cnr.it

Abstract

This paper analyses the strengths and weaknesses of self-organising approaches, such as evolutionary robotics, and direct design approaches, such as behaviour-based controllers, for the production of autonomous robots' controllers, and shows how the two approaches can be usefully combined. In particular, the paper proposes a method for encoding evolved neural-network based behaviours into motor schema-based controllers and then shows how these controllers can be modified and combined to produce robots capable of solving new tasks. The method has been validated in the context of a collective robotics scenario in which a group of physically assembled simulated autonomous robots are requested to produce different forms of coordinated behaviours (e.g., coordinated motion, walled-arena exiting, and light pursuing).

Keywords: Neural networks, genetic algorithms, self-organisation, motor schema-based controllers, potential fields, modularity, multi-variable statistical regression

1. Introduction

In the field of autonomous robotics, approaches in which the controllers are designed by the experimenter, such as *behaviour-based robotics* (Brooks, 1986; Wang, 1991; Kube and Zhang, 1993; Arkin, 1998; Balch and Arkin, 1998; Holland and Melluish, 1999; Krieger et al., 2000; Ijspeert et al., 2001; Desai et al., 2001; Fredslund and Mataric, 2002; Wang et al., 2003; Barfoot and Clark, 2004), and approaches in which some of the characteristics of the controllers are developed through automatic procedures, such as *evolutionary robotics* (Nolfi and Floreano, 2000; Ward et al., 2001; Quinn et al., 2002; Quinn et al., 2003; Spector et al., 2005; Trianni et al., 2006; Baldassarre et al., 2006; Trianni et al., 2007), are usually seen as two alternative methods based on partially contrasting principles. This paper proposes a method for combining the strengths of automatic procedures and direct design methods. In particular it show how effective solutions discovered through an evolutionary technique can be re-coded in motor schema-based controllers which can be later manipulated and combined to produce new behaviours.

To accomplish this goal, the research presented here compares *evolved feed-forward neural-network controllers* (Cliff et al., 1993; Miglino, 1995; Nolfi and Floreano, 2000) with *hand-coded motor schema-based controllers* (Arkin, 1989; Arkin, 1998). Artificial neural networks are a formalism widely used to encode robots' controllers in evolutionary robotics research (Nolfi and Floreano, 2000). Feed-forward neural networks are the simplest type of neural controller in which the state of the motors is a function of only the current state of the sensors. Feed-forward neural controllers have been chosen because they were sufficient for the purposes of this study and because they could be easily compared with hand-coded motor schema-based controllers. Hand-coded motor-schema based controllers are a class of behaviour-based controllers (Brooks, 1986; Arkin, 1998) based on *artificial potential fields* which have been successfully used with both mobile robots (Arkin, 1989) and robotic manipulators Khatib (1986). These types of controllers have been chosen because, as feed-forward neural controllers, they involve a direct mapping between the activation of sensors and the commands issued to motors: this feature was expected to ease the comparison of the two types of controllers. Here the mapping between the two classes of controllers will be obtained with suitable mathematical multi-variable functions. The form of these functions will be directly designed, whereas its parameters, depending on the specific goal in hand, will be either hand-tuned, or obtained through suitable regressions, or searched with evolutionary/design hybrid techniques.

The test of models were conducted in the context of a collective robotics scenario (Cao et al., 1997; Kube and Bonabeau, 1998; Martinoli, 1999; Dudek et al., 2002; Grabowski et al., 2003; Dorigo and Sahin, 2004) in which a "swarm" of assembled robots (Mondada et al., 2004) is requested to display a variety of coordinated cooperative be-

haviours and in which each robot has access to only local sensory information.

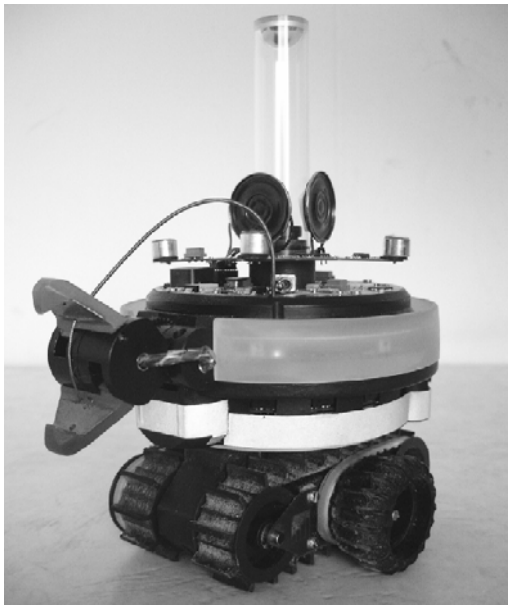


Figure 1: The robot that has been reproduced in the simulator used to carry out the experiments reported in the paper.

Beside highlighting the general strengths and weaknesses of the two approaches, the paper also shows how: (a) the solutions encoded in evolved neural controllers (section 2 illustrates the methods used for this evolution and section 3 illustrates the functioning of the evolved controllers) can be implemented in motor schema-based controllers (section 4); (b) the obtained motor schema-based controllers can be manually manipulated to identify the key functioning features of the evolved solution (section 5); (c) the obtained motor schema-based controllers can be modified to obtain new controllers able to produce new behaviours (section 6); (d) different schema-based controllers obtained from the evolved ones can be combined to develop robots able to produce more complex behaviours (section 7).

2. The experimental set-up

2.1 The robot

The research presented here was carried out within a research project, *SWARM-BOTS*, funded by the European Union (IST-FET Program; Dorigo et al., 2004; Mondada et al., 2004). The goal of this project was to develop *swarm-bots*, that is groups of fully autonomous robots able to physically connect and disconnect to form larger robotic systems. These systems can assume different physical shapes and act to solve problems that cannot be solved by single robots. This paper focuses on how a group of robots that are already assembled can accomplish common tasks such as to coordinate the motion direction in a distributed fashion (that is without a leader, cf. Baldassarre et al., 2006). Other researches carried out within the project studied how robots can self-assemble and disassemble to accomplish collective tasks (Groß et al., 2006; Tuci et al., 2007).

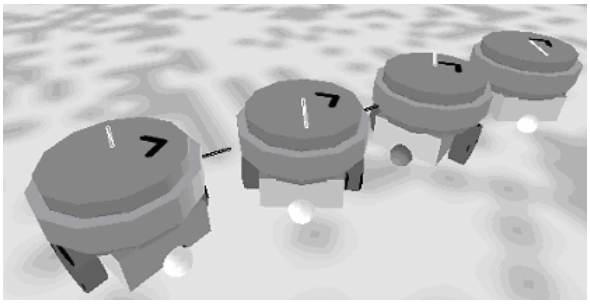


Figure 2: Four simulated robots linked up to form a linear swarm-bot. Each robot is made up by a chassis (parallelepiped) to whom two motorised cylindrical wheels and two small spherical wheels are attached (the two passive wheels have different colours, dark and light grey, to allow distinguishing the two possible chassis' fronts). The chassis is connected to a cylindrical turret. The black segment between the turrets of two robots represents a physical link between them. The white line above each robot's turret, which goes from the turret's centre to a point on its perimeter, indicates the direction of traction and, with its size, the intensity of traction.

Each robot (Figure 1; cf. Mondada et al., 2004) has a cylindrical body with a diameter of 11.6 cm and consists of a mobile base ("chassis"), and a main body ("turret"). The chassis is endowed with two motors each controlling a track and a teethed wheel. A third motor allows the turret and the chassis to actively rotate with respect to each other. The turret is provided with two grippers, one rigid and one flexible, that allow the robots to self-assemble and grasp objects. Each robot is provided with a number of different sensors (Mondada et al., 2004), but only the traction sensor described below has been simulated and used in the experiments reported in this paper.

In order to carry out the experiments reported in the paper we built a simulator of the robot based on the SDK Vortex™ toolkit (Critical Mass Labs, Canada) which allows programming realistic simulations of dynamics and collisions of rigid bodies in 3D. Given the high computational costs of simulations, only few relevant characteristics of the sensors, actuators and body of the robot were simulated; moreover the size of the robots and the gravitational acceleration coefficient were reduced to have the possibility of increasing the simulation time step without having instabilities.

The motor system of a simulated robot was modelled by four wheels connected to the chassis: two lateral motorised wheels that modelled the external wheels of the real robot and two spherical passive wheels placed at the front and at the back to stabilise the ro bot. The chassis was connected to the turret, modelled as a cylinder, through a motorised joint (Figure 2). The turret was endowed with a gripper which was modelled by creating a physical joint between the robot and other robots when needed (this joint was either rigid – in which case it will be called *rigid link* in the following sections – or possessed a free hinge with a vertical pivot – in which case it will be called *flexible link*). The

active and passive wheels had a diameter of respectively 2.30 and 1.15 cm. The turret had a diameter of 5.8 cm and a height of 4.6 cm.

During evolution, spherical collision models were used for all the wheels and for the chassis, as these speeded up computations (results equivalent to those reported below were obtained by testing the evolved controllers with the collision models shown in Figure 2). The gravitational acceleration coefficient was set at 9.8 cm/s^2. This low value, that caused a low friction of the wheels on the ground, was compensated for by setting the maximum torque of the motors at a low value, 70 dynes cm. The coefficient of friction, simulated by Vortex according to the Coulomb model, was set at 0.6. The desired speed of the wheels varied within ±5 rad/s. The desired speed applied to the turret-chassis motor was permanently set equal to the difference between the desired speed of the left wheel and right wheel times 0.26 (this setting implies that, when the chassis turns, the turret turns in the opposite direction so that its orientation does not change with respect to the environment: this greatly helps the robots to turn their chassis when they are attached to other robots). The state of the sensors and motors, and the differential equations used by Vortex to simulate the bodies' dynamics, were updated every 100 ms.

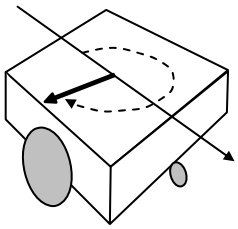


Figure 3: Traction force detected by the robot traction sensor. The parallelepiped represents the chassis. The turret has not been drawn for clarity. The large and small grey circles represent respectively the right motorised wheel and the front passive wheel. The thin arrow indicates the orientation of the chassis, the bold arrow indicates the vector of the traction force that the turret exerts on the chassis, and the dotted arrow indicates the angle of traction measured clockwise from the back of the robot.

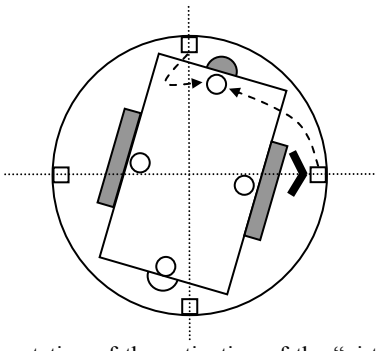


Figure 4: Computation of the activation of the "virtual light sensors" on the basis of the activation of the light sensors. The four empty squares represent the light sensors placed on the turret. The four empty circles represent the virtual light sensors located on the chassis. As an example, the dotted arrows depart from the sensors that were used to compute the activation of the virtual sensor pointed by the heads of the arrows themselves.

Each robot was provided with a *traction sensor* placed at the turret-chassis junction (Figure 3). This sensor returned the direction (angle with respect to the chassis' orientation) and the intensity of the force of traction that the turret exerted on the chassis. Traction was caused by the movements of both the connected robots and the robot's own chassis. Notice that, by being rigidly assembled to other robots, the turret of a specific robot *physically integrated* the forces produced by the other robots on it. As a consequence, the traction sensor measured the mismatch between the directions of motion of the robot and the direction of motion of the rest of the group, and hence furnished the robots an important communication channel based on *implicit communication* (Quinn et al., 2003; Tummolini et al., in press). The intensity of the traction force measured the size of this mismatch. To have more realistic simulations, a 2D noise ranging within ±5% of the maximum value was added to the traction force seen as a 2D vector.

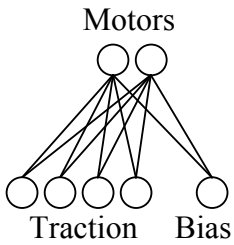


Figure 5: The neural controller of each robot consisted of a two-layer neural network with five input neurons, encoding the direction and intensity of traction plus a bias signal, and two output neurons, encoding the desired speed of the wheel motors.

Each robot was also endowed with four light sensors. These sensors were used to extend the controller in order to solve light pursuing tasks (see section 7). The sensors were positioned on the perimeter of the turret, and were simulated by using a sampling procedure applied to a real sensor (cf. Miglino et al., 1985; the sensors had a high sensitivity to the light gradient: their activation was maximum when close to the light, exponentially decreased with a decreasing distance, and achieved a value of zero at 400 cm). A noise ranging within ±5% of the maximum intensity was added to the sensors. Shadows were simulated by computing geometrical projections of obstacles in the sensors' fields. In order to provide the robot with information about the light with respect to the orientation of the chassis (this greatly eased control as the wheels were connected to the chassis), the activations of the light sensors were used to compute the activation of four "virtual light sensors". The activation of these sensors was computed on the basis of the weighted average of the activation of the two light sensors closer to the considered virtual sensor, with weights proportional to the angular distance of the latter from them (Figure 4).

2.2 The neural controller

Each robot's controller (Figure 5) consisted of a neural network with five input neurons directly connected to two output neurons. The first four input neurons encoded the traction direction on the basis of a cosine function and four different "preferred orientations". In particular, the activation x_i of the input neuron i was computed as follows:

$$x_i = [cos(ta - ta_i)]^+ \cdot ti$$

where ta is the traction angle (measured clockwise from the robot chassis' rear), ta_i is the preferred orientation of the input neuron i (set to 0 rad, $(1/2)\pi$ rad, π rad, and $(3/2)\pi$ rad respectively for the first, second, third and fourth input neuron), $cos(.)$ is the cosine function, $[.]^+$ is the identity function returning 0 for negative values, and ti is the traction intensity normalised in [0, 1]. The last input neuron, x_5, was a bias neuron always activated with one.

The activation y_j of each of the two output neurons was computed on the basis of the activation of the five input neurons x_i and a sigmoid transfer functions as follows:

$$py_j = \sum_i (w_{ji} \cdot x_i)$$
$$y_j = 1/(1+\exp(-py_j))$$

where py_j is the activation potential of the output neuron j, and exp(.) is the exponential function. The activation of the two output neurons was used to set the desired speed of the two robot's wheels by mapping it onto ±5 rad/s and was used to set the desired speed of the turret-chassis motor according to the mechanism explained in section 2.1.

The architecture of the neural controller illustrated above was chosen by testing and comparing the performance and evolvability of various feed-forward different neural networks having a variable number of hidden units. Finally, a neural network with no hidden units was chosen as this had a performance comparable to that of the other neural networks but had a higher evolvability (i.e. the genetic algorithm optimised its parameters in fewer generations).

2.3 The genetic algorithm

The connection weights of the neural controllers were evolved with a genetic algorithm (Nolfi and Floreano, 2000). Among the various available learning techniques a genetic algorithm was chosen because: (a) supervised-learning algorithms could not be used as the behaviour of the single robots leading to efficient collective performance was not known *a-priori* (cf. Yamashita and Tani, 2008); (b) reinforcement learning techniques were difficult to apply due to the high integration of the required collective behaviour and the complex dynamics of the group (cf. Matarić, 1997); (c) unsupervised learning techniques, based only on self-organising principles (cf. Janet et al., 1997), could not be used as we wanted to have a criterion with which to guide the algorithm to develop specific desired behaviours (such as the fitness of genetic algorithm).

The initial population of the genetic algorithm consisted of 100 randomly generated genotypes that encoded the connection weights of 100 corresponding neural controllers. Each connection weight was represented in the genotype by eight bits that were transformed into a number within ±10. Each genotype encoded the connection weights of four identical neural controllers which were used to control a group of four robots linked up to form the swarm-bot shown in Figure 2. Each swarm-bot was tested five times ("epochs"), each lasting 150 time steps of 100 ms. The 20 best genotypes of each generation were allowed to reproduce by generating five copies each, with 3% of their bits replaced by a new randomly selected value. The evolutionary process lasted 100 generations. The evolution was replicated 30 times with different seeds of the random number generator (as a consequence these evolutionary runs started with different random genotype populations).

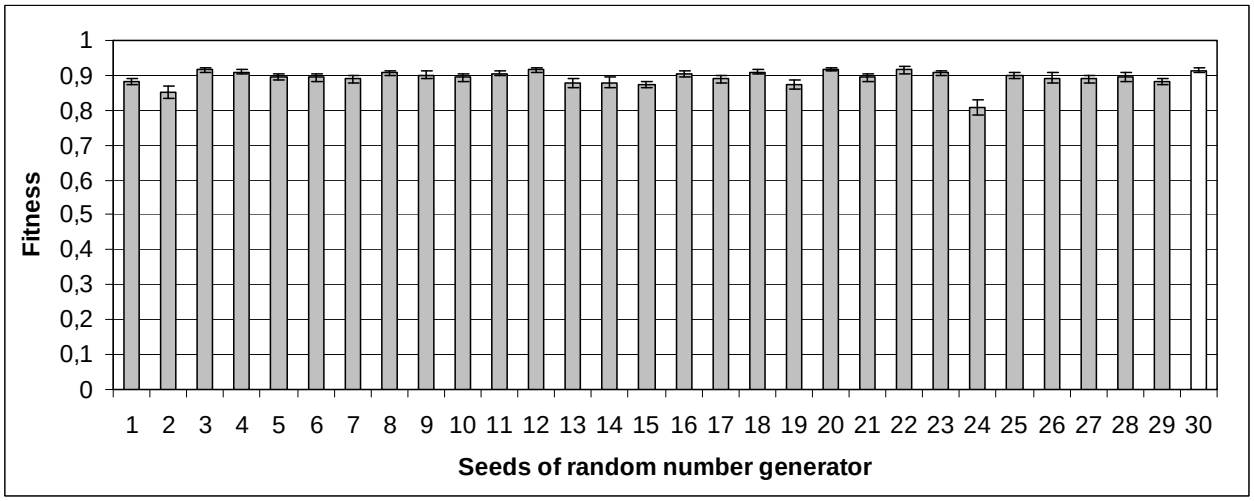
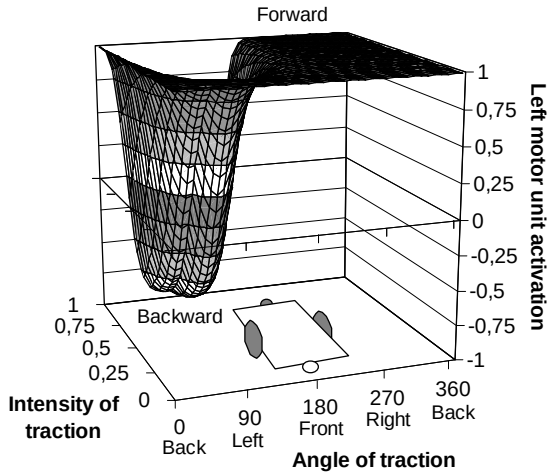
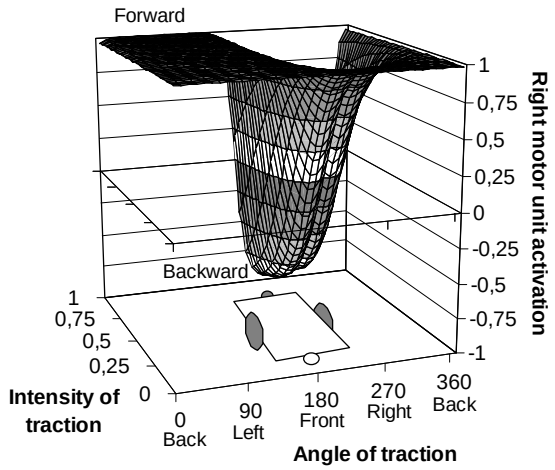


Figure 6: Average fitness and standard error, measured over 100 tests, of the best controllers of the 30 runs of the evolution. The white bar corresponds to the 30th controller, used throughout the paper.

The swarm-bots were selected for the ability to move as fast and as straight as possible. More specifically, the fitness of each swarm-bot was computed by first measuring the Euclidean distance between the centre of mass of the swarm-bot at the beginning and at the end of each one of the five epochs, and then by summing the resulting measures. To normalise the value of the fitness to one, the total fitness of one swarm-bot over five epochs was divided by the maximum distance travelled by a single robot moving straight at maximum speed for 750 (=150×5) steps.



(a)



(b)

Figure 7: The graphs show the commands that the controller corresponding to seed 30 issues to the robots' left motor (a) and right motor (b) in correspondence to a traction force having different angles and intensities. The vertical axis indicates the activation of the motor neurons which cause the robot's speed and whether its moves straight, turns left or turns right. The schematic little picture represents a chassis and should aid the "visualization" of the direction of traction with respect to the chassis itself: the white little wheel of the schematic chassis represents the rear of the chassis and corresponds to an angle of traction of 0° measured clockwise.

3. The evolved controller

Figure 6 shows the average fitness and standard error of the best controllers of the 30 runs of the evolution, measured over 100 epochs. All runs produced controllers that lead the robots to coordinate so as to allow the group to move fast and straight. The rest of the paper focuses on the controller corresponding to seed 30. The weights of this controller are reported in Table 1. This controller was selected as: (a) it had a high performance (its performance was not statistically different from the controller with the highest performance, corresponding to seed 20, t-test, $p = 0.71$); (b) it implied a forward movement of the robots (recall that the robots have two possible fronts of motion: using the forward front eased the analyses); (c) it had a particularly regular shape (see Figure 7) that eased the regression process and the analysis of the controller (see sections 4 and 5).

The functioning of the evolved controller at the individual and collective level is now briefly described (for more details see Baldassarre et al., 2004, and Baldassarre et al., 2006). Direct observation of the collective behaviour of robots indicates that: (a) at the beginning of the test they start to pull/push in different directions, (b) then they orient their chassis in the direction where the majority of the other robots are moving, and then (c) they move straight along the direction that emerges from the initial negotiation in fine coordination with the other robots. The absolute direction that emerges from the robots' negotiation changes in different tests depending on the initial orientation of the robots, but the robots always converge towards a single direction of motion (see Baldassarre et al., 2006, for data).

Table 1: Weights of the neural-network controller emerged in the 30th evolutionary run. The rows indicate the weights corresponding to the two output units of the neural network whereas the columns indicate the weights corresponding to the input units (I1, I2, I3, and I4) having a particular preferential traction direction (traction from left, front, right and back), and a bias unit (Bias).

	I1 Left	**I2 Front**	**I3 Right**	**I4 Back**	**Bias**
Left motor	-10.000	-1.1718	7.5781	-5.0781	6.4843
Right motor	8.1250	-1.4062	-8.8281	-3.7500	5.3906

In order to understand how the individual controller produced this behaviour, the activation of its two output units corresponding to traction forces having different angles and intensities was measured and plotted (Figure 7). The analysis of the resulting graphs reveals that the controller works as follows. When the traction comes from the front (about 180°), the robot is oriented toward a direction that is close to the "mean" direction of motion of the other robots. In this situation the robot moves straight. When the traction comes from the left hand side (about 90°) or the right hand side (about 270°) there is a significant mismatch between the orientation of the robot and the mean orientation of the other robots. In this condition the robot turns toward the direction of the traction force, by turning left when the traction comes from the left hand side and by turning right when the traction comes from the right hand side. The speed of turning is proportional to the intensity of the traction force. When traction comes from the rear (about 0°) the robot goes straight at maximum speed independently of the intensity of traction. This might be due to the fact that when the traction comes from the left or the right hand side the robot has to respectively turn left or right, so the point cor-

responding to 0° (rear) represents the separation between the two different turning behaviours. Overall, the individual robot's behaviour might be characterised as a *conformist tendency* to follow the direction of motion of the rest of the group: indeed traction provides an indication of the average direction of motion of the other robots. At the group level, this tendency rapidly leads the robots to select the same direction of motion and to move in a coordinated fashion. Note that these individual and collective behaviours have the features of a self-organising process based on a positive feedback mechanism, likely also characterised by a phase transition phenomenon (see Baldassarre, et al. 2004b; Baldassarre, 2008; Turgut et al., 2008, for details).

Notwithstanding this analysis revealed important features of the evolved solution, it did not allow us to fully clarify the role of other characteristics. In particular, it did not allow clarifying the functional role, if any, of the following characteristics of the evolved controllers: (a) the tendency of the robots to persevere in their direction of motion when the traction comes from their rear ("stubbornness"; see Figure 7); (b) the tendency of the robots to move forward at maximum speed with traction forces coming from the front (see Figure 7); (c) the left/right asymmetry of some evolved controllers having high fitness (this implied that the effects of traction forces coming from the left or the right and side were different in these controllers, data not shown). As we will see, the method described in section 4 allowed us to better understand these features of the evolved controllers.

4. Mapping evolved neural-controllers into motor schema-based controllers

This section describes how it is possible to re-code the sensory-motor mapping implemented by the evolved neural controller into a suitable motor schema-based controller encoded as a multi-variable equation. Motor-schema based controllers (Arkin, 1989), initially developed by Khatib (1986) in particular in relation to robotic manipulators, are a type of controllers used in behaviour-based robotics (Brooks, 1986; Arkin, 1998) in which the control modules responsible for producing different elementary behaviours are expressed as mathematical equations which generate artificial potential fields capable of guiding robots' movements. For example, in the case of a robot engaged in a navigation task, if graphically visualised with a gradient graph the potential field generated by a motor schema-based controller might indicate: (a) the robot's direction of motion in the various positions in space, e.g. pointing away from obstacles and other robots, and pointing towards a navigation goal; (b) the robot's speed related to the distance to such objects. A key aspect of the motor schema-based approach is that it allows generating behaviours derived from various sources (e.g. various obstacles, resources, etc.) as a weighted sum of the different potential fields. An application of this principle will be shown in section 7.

To recode the neural controller into a motor schema-based controller a strategy was followed which should be also applicable to other problems which can be solved with feed-forward neural controllers. The strategy is based on the selection of a suitable nonlinear mathematical function and the use of a statistical regression technique to estimate its parameters on the basis of data sampled from the input-output mapping of the original neural controller. The selection of the mathematical function is the most delicate passage of the procedure. Here this selection was performed by trying to satisfy the following constraints: (a) the function should be mainly formed by summations and multiplications of Gaussian and sigmoid functions: this is an important point as these functions allow forming functional bases which on one side are suitable for statistical regressions (e.g. due to their overall simplicity, symmetry or monotonicity), and on the other side allow building universal function approximators (see Cybenko, 1989, and Park and Sandberg, 1991, for the sigmoid and Gaussian bases, respectively); (b) the capacity of the function of approximating the function expressed by the controller of interest (this capacity can be measured by the residual error of the regression); (c) the presence of parameters that allow manipulating with ease the aspects of interest of the behaviour exhibited by the controller (see sections 5 and 6 for examples of this; in this respect, consider that having more parameters than the original controller – as it happened in the example shown here where the approximation function has 16 parameters vs. the 10 parameters corresponding to the weights of the original neural controller – might allow rendering the various aspects of the controller independent between them and hence more easily modifiable). These criteria should allow applying the procedure also to problems different from those reported here to the extent that they have a similar complexity. The estimation of the function parameters reported below was performed with a Least Mean Square nonlinear regression using the Nonlinear Regression Toolbox of Matlab^{TM}.

The parameters' estimation was based on the input-output vector couples produced by the best evolved neural controller corresponding to the 30th run of the evolutionary experiment (Table 1). These input-output vector couples were sampled by systematically varying the input pattern of the evolved network and by computing the corresponding output pattern. The input pattern was varied by sampling the angle and intensity of traction over 41 values each. This sampling process produced two input-output data sets, one for the left motor and one for the right motor, each composed of 41*41=1681 elements. These two data sets are graphically displayed in Figure 7 and, partially, in Figure 8a.

As a first possible candidate function for the regression, we selected a Gaussian function with one independent variable corresponding to the angle of traction, multiplied by a three-degree polynomial function with one independent variable corresponding to the intensity of traction:

$$wds = b7 + \left(b3 \exp\left(-\frac{(ta\text{-}b1)^2}{b2} \right) \right) \left(b4\, ti + b5\, ti^2 + b6\, ti^3 \right) \qquad (1)$$

where *wds*, the dependent variable of the function, is the wheel's desired speed ranging over [-1, 1], *ta* is the traction angle ranging over [0, 1], *ti* is the traction intensity ranging over [0, 1], *b1*-*b7* are the parameters of the function which were estimated separately for the left and right wheel. The values of the parameters obtained through the regression for the two wheels are shown in Table 2. The Mean Square Error of the regression for the left and right wheels was respectively 0.0035 and 0.0030. The mapping performed by the function on the basis of the tuned parameters is shown in Figure 8b (compare it with Figure 8a relative to the evolved controller).

Table 2: Parameters of the "Gaussian controller" (equation (1)) estimated with a Least Mean Square nonlinear regression.

	b1	**b2**	**b3**	**b4**	**b5**	**b6**	**b7**
Left	0,723	0,032	2,008	0,962	-4,266	2,033	1,013
Right	0,291	0,034	2,023	0,787	-4,192	2,191	1,011

In order to improve the approximation, and to have parameters which allow an independent regulation of controllers' "stubbornness" and asymmetry, a second function was designed. This was composed by the product of three Sigmoid functions, two depending on the traction angle and one depending on the traction intensity:

$$wds = b8 + \frac{1}{\left(1 + \exp\left(-\frac{(ta\text{-}b1)}{b2} \right) \right)} \frac{1}{\left(1 + \exp\left(+\frac{(ta\text{-}b3)}{b4} \right) \right)} \qquad (2)$$

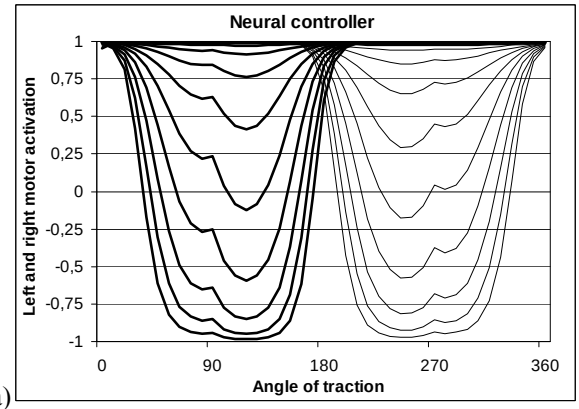
$$\frac{-(b7*2)}{\left(1 + \exp\left(-\frac{(ti\text{-}b5)}{b6} \right) \right)}$$

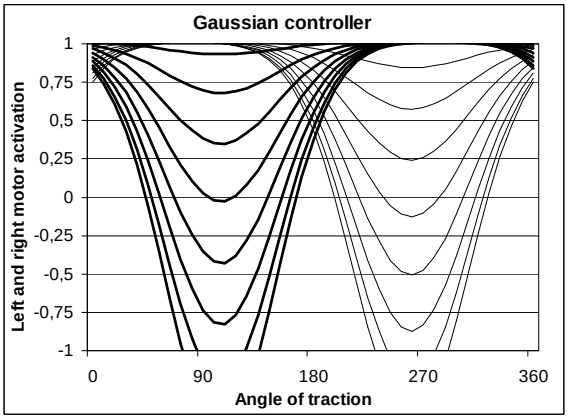
By using this function the Mean Square Error of the regressions for the left and right motor were respectively 0.0004 and 0.0007. The mapping performed by the function on the basis of the estimated parameters (see Table 3) is shown in Figure 8c.

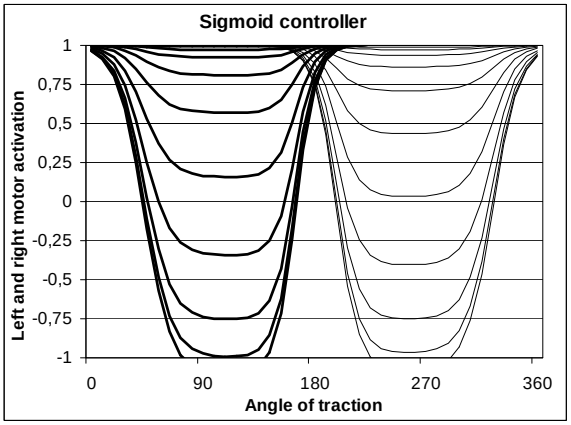
Table 3: Parameters of the "Sigmoid controller" (equation (2)) estimated with a Least Mean Square nonlinear regression.

	b1	**b2**	**b3**	**b4**	**b5**	**b6**	**b7**	**b8**
Left	0,118	0,030	0,458	0,022	0,650	0,107	1,099	1,005
Right	0,549	0,025	0,901	0,030	0,627	0,122	1,092	1,001

The estimated parameters of the "Sigmoid function controller" (Table 3) were rounded as reported in Table 4 in order to simplify their manipulation and the interpretation of the effects of such manipulations on the robots' behaviour illustrated in the next sections (the important parameters *b1* and *b3* were rounded in such a way that the "sides" of the two Sigmoid functions sensitive to the traction angle were all at a distance of 0.05 from either one of the two critical values of traction force's angle, 0 and 0.5, respectively corresponding to 0° and 180°; *b2* and *b4*, regulating the pendence of the same two functions, were rounded to 10^{-2}; the less sensitive parameters *b5*-*b7*, regulating the Sigmoid function dependent on traction intensity, and *b8*, regulating the level of the overall function, were rounded to 10^{-1}). Figure 8d displays the mapping obtained on the basis of the function resulting from these rounding.

(a)


(b)


(c)


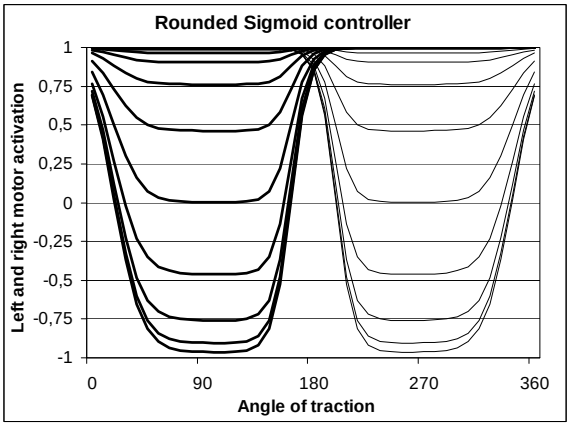
(d)


Figure 8: The sensory-motor mapping performed by four different controllers: (a) the evolved neural controller; (b) the Gaussian controller; (c) the Sigmoid controller; (d) the Rounded Sigmoid controller. Each graph encodes the desired speed of the two motors (y-axis) for traction forces with different directions (x-axis). Thick and thin lines encode the desired speed of respectively the left and right motor. The 11 curves in each graph correspond to desired speed of traction with different intensities (from –1 to +1 in intervals of 0.2).

Figure 9 compares the mean and standard error of the performance of the four controllers (i.e. respectively the neural-network controller, the "Gaussian controller", the "Sigmoid controller", and the "Rounded Sigmoid controller") in 100 trials of the coordinated motion task used to evolve the neural controller. The results show that the evolved neural controller outperforms the three equation-based controllers: the difference of performance between the former and each one of the latters is statistically significant (t-test, $p < 0.01$ in all cases). This implies that small differences in the sensory-motor mapping (as mentioned above, the residual errors after the regressions were very low) play a significant role in robots' behaviour.

Table 4: Rounded parameters of the "Sigmoid controller" (equation (2)) estimated with a Least Mean Square nonlinear regression.

	b1	**b2**	**b3**	**b4**	**b5**	**b6**	**b7**	**b8**
Left	0.05	0.03	0.45	0.02	0.60	0.10	1.00	1.00
Right	0.55	0.02	0.95	0.03	0.60	0.10	1.00	1.00

As the Rounded Sigmoid controller had a performance higher than the Sigmoid controller and the Gaussian controller (even if not statistically significant: t-test, $p > 0.1$ in both statistical comparisons), and moreover, differently from the latter two, it had some parameters that allowed changing important aspects of the robots' behaviour (see next section), it was used in all experiments presented in the remaining sections of the paper. Section 7 will show the artificial potential field generated by this controller.

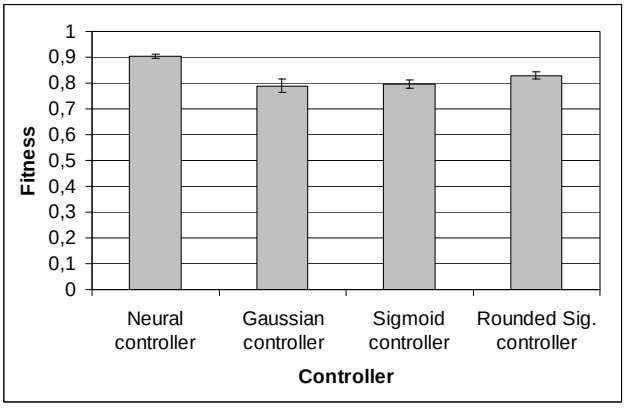


Figure 9: Average performance and standard error measured over 100 trials of robots provided with the evolved neural controller and the three equation-based controllers described in the text.

5. Manipulating the motor schema-based controller to identify crucial aspects of the evolved controller

This section describes the results of the manipulations of the Rounded Sigmoid controller developed in the previous section which were carried out to understand the role played by the various aspects of the mathematical function that it implements (see Figure 7). This analysis focused on the aspects of the controller that the experiments described in section 3 suggested to be important with respect to the coordinated behaviour displayed by the whole swarm-bot, namely: conformism/stubbornness with respect to traction from the rear, reaction to traction from the front, and symmetry/asymmetry of reaction to traction from the left and right hand side.

The effects of the manipulation of the controller on the swarm-bot's behaviour were measured in terms of the average distance covered by it in 100 trials (the same measure used in section 4). The results are summarised in Figure 10. The histogram bar "D" indicates the performance observed by the unmodified Rounded Sigmoid controller. The histogram bars A-E indicate the performance of controllers that have increasing levels (from left to right) of conformism to traction forces coming from the rear. The level of conformism was regulated by setting the parameters *b1* of the left wheel and *b3* of the right wheel, that establish the reactivity of respectively the left and right motor to traction from the rear, to the couples of values: {0.20, 0.80}, {0.15, 0.85}, {0.10, 0.90}, {0.05, 0.95}, and {0.00, 1.00} (the effects of these parameters on the mathematical function implemented by the controller are indicated by the oblique arrows on the graphs reported under the respective bars in the histogram of Figure 10). The performance of controllers corresponding to bars A, B, C, and E is statistically lower when compared to performance of the unchanged Sigmoid, bar D (t-test, $p < 0.05$ in all cases). The increase of performance from controller A to D indicates that increasing levels of conformism with respect to traction forces coming from the rear improve the capacity of the robots to coordinate quickly: the reason is that the controllers react more readily to mismatches with respect to the group's motion. The low performance of controller E (the one with the highest conformism), can be explained by considering that in this case the left wheel erroneously slows down when the traction comes from 350° or more and, similarly, the right wheel erroneously slows down when the traction comes from 10° or less (see oblique arrows on graph under bar E). These results suggest that our previous interpretations correctly attributed a central role to conformism and that stubbornness for traction forces coming from the rear does not play an important functional role (as it was erroneously suggested in a previous work, Baldassarre et al. 2003, before conducting this analysis).

Tests F-G, compared with test D, analyse the effects of manipulations that increase the interval in which robots exhibit a tendency to move forward at maximum speed for

traction forces coming from the front (see the vertical arrows on the graphs under the respective bars). These manipulations were performed by setting parameters *b3* of left right wheel and *b1* of right wheel to the values {0.35, 0.65} for F and {0.25, 0.75} for G (note that in the case of D they were set at {0.45, 0.55}). The fact that performance of controller F does not statistically differ from the unchanged controller D (t-test, $p = 0.96$) indicates that the response to traction forces coming from the front is not as important as the lack of conformist behaviour for traction forces coming from the rear. Performance is impaired only if such tendency is extended to an excessively wide range of traction forces coming from the front, as in the case of controller G (its performance is statistically lower than that of D, t-test, $p < 0.05$), likely because in this case the capability of the controller to suitably respond to traction forces coming from the left or from the right hand side is impaired.

Tests H-K were conducted to analyse the effects of asymmetries between the reactions of left and right motors to tractions coming respectively from the left and the right hand side of the robots. The controllers of tests H-J were obtained by changing in various ways the parameter *b1* and *b3* of the two wheels, with respect to the control condition D, so as to obtain different types of asymmetries (see values directly in Figure 10; the asymmetries of the controllers with respect to D are indicated by the vertical arrows on the respective lower diagrams). The performance of the three controllers was not statistically different from the performance of the controller D (t-test, $p = 0.92$, $p = 0.09$, $p = 0.11$, respectively). This indicates that small asymmetries produce little effects on performance. Performance significantly deteriorates (but surprisingly not so much) only with strong asymmetries such as that of test K: in this case the reactivity of the controller to traction forces coming from the left hand side of the robot was eliminated altogether.

Altogether these results explain why the five best-performing neural controllers evolved with different random number-generator seeds (3, 12, 20, 22, and 30, see Figure 6) vary significantly with respect to the reactivity to traction forces coming from the front and with respect to asymmetry, while they all have a high reactivity (conformism) to traction forces coming from the left or right hand side (especially near the rear), at least for one of the two motors (data not reported).

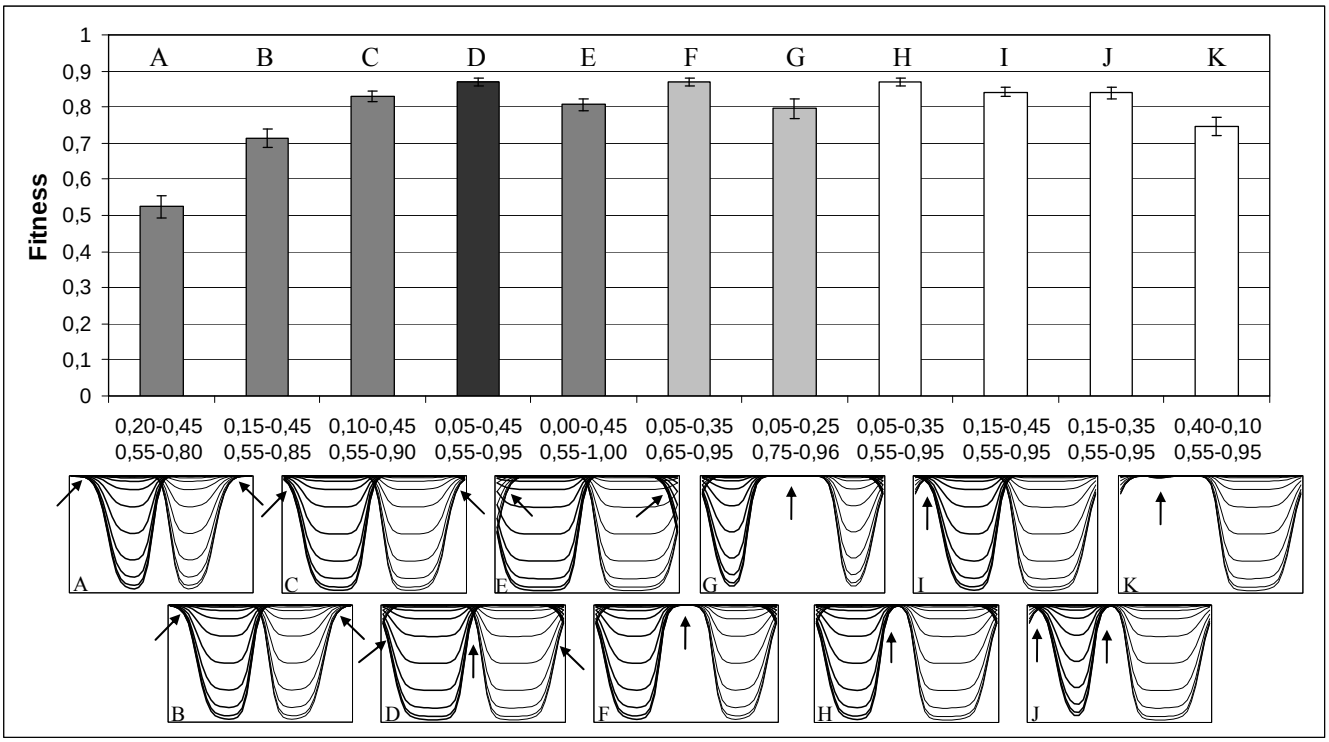


Figure 10: Average performance and standard error over 100 tests of controllers obtained by manipulating the Rounded Sigmoid controller. The function implemented by each controller is visually rendered by a small graph vertically below the respective bar of the histogram (same scale and conventions of Figure 8). The two couples of numbers under each bar indicate the parameters b1-b3 of the left wheel (first couple of numbers) and b1-b3 of the right wheel (second couple of numbers). The black bar (D) refers to the unchanged controller; dark grey bars (A, B, C, E) and light grey bars (F, G) refer to controllers used to study the effects of different levels of subbornness/conformism for traction forces coming from respectively the robots' rear or front; white bars (H-K) refer to controllers used to study the effects of different levels of asymmetry. Arrows on the small graphs highlight relevant aspects of the functions implemented by the controllers (see text).

6. Developing controllers that exhibit new behaviours

This section describes the results obtained by modifying the parameter of the Rounded Sigmoid controller in order to develop swarm-bots that display *new* types of behaviours. In particular, these experiments exemplify how new behaviours can be obtained either through direct modification of the parameters of the hand-coded controller or through an Iterated Evolutionary Computation (IEC) technique (see Takagi, 2001, for a review). The two examples used to illustrate this point, reported in sections 6.1 and 6.2, are both related to behaviours based on self-organising mechanisms guiding the whole system. In this respect, the results not only show how the hand-tuned controller might allow developing new potentially useful behaviours, but they also indicate that the technique proposed in this paper allows studying self-organising processes in collective robotic systems.

IEC is an evolutionary technique in which the selection of the best individuals is not performed automatically, on the basis of a formalised selection criterion, but rather by the experimenter, on the basis of the visual inspection of the behaviour exhibited by the robots. IEC methods have an important advantage with respect to automatic evolutionary procedures consisting in the fact that they do not require identifying a detailed "effective selection criterion" (i.e. a fitness function which maximises not only final effective solutions of the problems, but also approximate solutions in early stages of the evolutionary process which represent necessary steps to build the final solutions, cf. Nolfi and Floreano, 2000). Moreover, by exploiting human ability to judge behaviours, IEC methods allow evolving behaviour on the basis of abstract selective criteria, such as "displaying interesting behaviours" or "display synchronised behaviours", that are difficult to formalise (Funes et al., 2004). On the side of the drawbacks, IEC techniques require the experimenter to evaluate the behaviour of all the produced controllers and so are extremely time consuming and in practise can be applied only to problems with limited search spaces, that is involving limited numbers of free parameters.

6.1 Synchronised periodic behaviours

In a first experiment we tried to develop swarm-bots able to display synchronised periodic behaviours (Strogatz, 2003) relying on self-organising principles (Camazine et al., 2001; Baldassarre, et al. 2004b; Baldassarre, 2008; Turgut et al., 2008) by varying the evolved solution through an IEC technique. According to Strogatz (2003), synchronised periodic behaviours rely on two specific self-organising mechanisms: (a) an intrinsic tendency of the elements composing the collective system to generate a periodic behaviour; (b) a tendency of the elements to slow down or to accelerate the frequency of their periodic behaviour on the basis of phase mismatches. In this respect, we wanted to verify if it was possible to obtain behaviours relying upon such mechanisms through a IEC procedure directly applied to the Rounded Sigmoid controller. The experimental set-up used for this experiment involved 10 robots linked to form a linear structure.

Table 5: Parameters of the controller obtained with an Interactive Evolutionary Computation technique which lead the robots to exhibit a periodic synchronised behaviour at the group level.

	b1	**b2**	**b3**	**b4**	**b5**	**b6**	**b7**	**b8**
Left	0.038	0.040	0.462	0.020	0.618	0.083	0.975	0.541
Right	0.535	0.025	0.924	0.003	0.607	0.085	0.992	0.981

The specific IEC technique used here was implemented by first defining the goal of the exercise in an intended generic form (cf. Funes et al., 2004), in our case "trying to obtain a periodic behaviour in a group of assembled robots", and then by performing multiple times the following three operations until the obtained behaviours were "satisfactory": (a) varying the free parameters of the neural controllers by clicking on a corresponding button of the graphic interface of the simulator; (b) observing the behaviour displayed by the robots after the variation of the free parameters; (c) deciding whether to retain or discard the behavioural variant so obtained by suitably clicking either one of two buttons of the graphic interface. The program introduced the variations of the parameters by adding a random number, drawn with a uniform probability distribution over the interval [-0.05, 0.05], to each parameter of the controller and then by truncating their values, if needed, within the range [0.0, 1.0].

By following this procedure, it was indeed possible to quickly obtain a new variation of the controller that, once embodied in the 10 robots, allowed them to produce a periodic behaviour and to quickly synchronize their movements. Figure 11 illustrates a typical behaviour obtained at the end of this procedure (notice how the individual robots start with different randomly assigned orientations). The analysis of the graphs, and the visual inspection of the robots' behaviour, indicates that: (a) at the individual level the controller tends to produce periodic behaviour that consists in producing a circular trajectory by turning counterclockwise (Figure 11a); (b) robots display a conformist tendency which leads each robot to turn with a larger or smaller orientation variation depending on whether the perceived traction force comes from respectively the same or from the opposite direction with respect to the direction of turning: this implies that robots tend to accelerate or decelerate their circling behaviour on the basis of the mismatch between their phase and the phase of the rest of the group (Figure 11b); (c) as a result of these accelerations and decelerations, robots rapidly converge into a stable state in which their motions are synchronised and in which the intensity of traction forces become close to null. Notice how the points (a) and (b) indicate that the synchronised periodic behaviour observed at the group level actually relies on the self-organising mechanisms hypothesised by Strogatz (2003) (for other examples and analyses of behaviours

based on self-organising principles, and relevant for collective robotics, see: Reynolds, 1987; Kube and Zhang, 1993; Beckers et al., 1994; Holland and Melluish, 1999; Krieger et al., 2000; Baldassarre et al., 2006; Baldassarre, 2008). Also notice how these results indicate that the tendency of the robots to modify the frequency of their circling behaviour in order to "catch up" or to "wait for" the rest of the group relies upon the same conformist tendency that was used by evolved robots to display coordinated movements along a single direction.

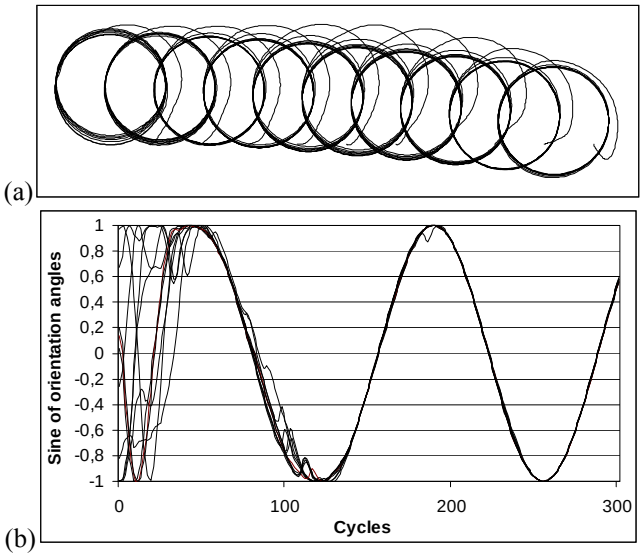


Figure 11: (a) Trajectories followed by ten robots assembled to form a linear swarm-bot during a test lasting 300 cycles, when guided by the Rounded Sigmoid controller modified through an IEC procedure. Each line corresponds to the trajectory followed by the barycentre of one robot. (b) Synchronization of the periodic circling behaviour displayed by the ten robots in the same test: the curves show the sine (y-axis) of the chassis' absolute orientation angle of the ten robots in 300 time steps (x-axis).

The analysis of the parameters of the new controllers, reported in Table 5, suggests that the most important variation of the controller's parameters with respect to their original values (cf. Table 4), consists in the variation of parameter *b8* associated with the average speed of the left wheel. This variation is responsible for the intrinsic tendency of the robots to produce a circling behaviour. Indeed, even if all parameters were reset, with the exception of parameter *b8* associated with the average speed of the left wheel, to the original values reported in Table 4, the robots still displayed the synchronised periodic behaviour.

6.2 Coordinated behaviours allowing a swarm-bot to exit from an arena

In a second experiment, we aimed at improving the swarm-bot's capability of exiting an arena surrounded by walls and with narrow ways out, by directly modifying the parameters of the Rounded Sigmoid controller. In this experiment the swarm-bot consisted of eight robots assembled through flexible links into a circular structure (Figure 12; a "flexible link" was simulated with two segments, each rigidly connected to a robot, connected between them through a passive hinge joint with one degree of freedom pivoting on the vertical axis). The environment included four walls which formed a square arena having four narrow passages located at the four corners. As the width of the passages was smaller than the diameter of the swarm-bot (when it had a regular circular shape), the swarm-bot had to appropriately deform its shape to effectively exit the arena (see Figure 12a).

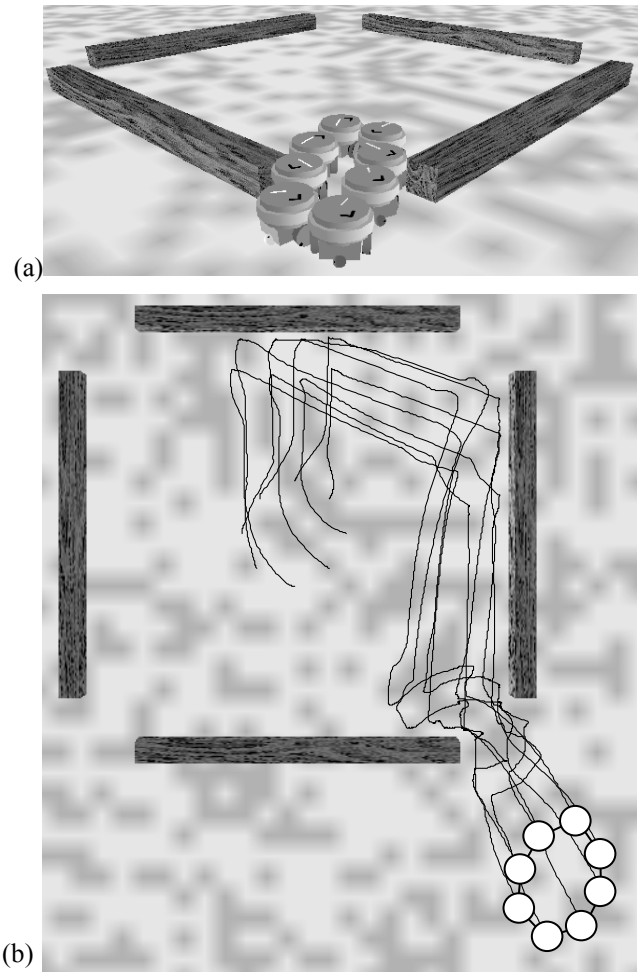


Figure 12: (a) A swarm-bot deforms its structure and succeeds in exiting a walled arena with four narrow passages. (b) The traces left by a swarm-bot engaged in exiting the arena.

As described in Baldassarre et al. (2003), swarm-bots provided with the evolved neural controller described in Sections 2 and 3 display an ability to generalise their coordinated motion in situations in which they are connected through flexible links; moreover, they spontaneously exhibit a coordinated obstacle avoidance behaviour as a result of the traction forces produced by the collisions between the robots' turrets and obstacles. As a result of the combination of these abilities, plus the spontaneous deformation of the swarm-bots' shape caused by the robots' collisions with walls, the evolved swarm-bot already displayed a good ability to solve the task described above (see Figure 12b). The

swarm-bot maintained these capabilities also when equipped with the Rounded Sigmoid controller. In particular, the swarm-bot provided with such controller managed to exit the arena in 33.5% of cases when tested for 200 trials lasting 3000 cycles each.

To modify the Rounded Sigmoid controller so as to enhance swarm-bot's ability to exit from the arena we tried to increase the level of stubbornness of the controller. The reason of this choice was the hypothesis that a higher level of stubbornness would have reduced tendency of robots to avoid obstacles which in turn would have increased the swarm-bot tendency to deform its shape, as a result of the collision with obstacles, so as to conform its shape to the characteristic of the passage and exit the arena more easily.

To verify this hypothesis, the test illustrated above was repeated with the Rounded Sigmoid controller where *b1* of the left wheel was set at 0.10 and *b3* of the right wheel was set at 0.90. The result of the test confirmed the hypothesis: the modified controller outperformed the baseline controller by exiting the arena 52% of times (t-test, $p < 0.01$).

This experiment shows how the hand-coded controller allows directly modifying its parameters so as to have a particular desired behaviour at the group level. This can be done, as it usually happens for hand-coded controllers, because it is sometimes possible to have a sufficiently accurate intuition about the causal relationship existing between the parameters of the controller, the behaviour of the single robots, and the behaviour of the whole group.

7. Using the motor schema-based controller as a building block in behaviour-based controllers

The Rounded Sigmoid controller developed for coordinated motion tasks could also be used as a building block to design controllers capable of solving more complex tasks. To illustrate this more in detail, we considered an experimental set-up where a linear swarm-bot formed by four robots assembled to form a linear structure had to coordinate to move in space and search and approach a light target. In previous research this behaviour was evolved from scratch (Baldassarre et al., 2004; given that swarm-bots as those used here exhibit spontaneous obstacle avoidance, as illustrated in section 6, this work used this behaviour to tackle a light searching task in a maze). This section shows how the whole behaviour can be implemented by using a modular architecture formed by two motor schemas, each based on the Rounded Sigmoid controller, producing respectively a coordinated motion behaviour and a coordinated light pursuing behaviour.

The first motor schema, capable of performing coordinated motion, was implemented by using the unmodified Rounded Sigmoid controller described in section 4. The second motor schema, capable of performing coordinate light-pursuing, was implemented by using a copy of the same Rounded Sigmoid controller that took as input the direction and intensity of the light (encoded as illustrated in section 2.1) instead of the direction and intensity of the traction force. The reason why the Rounded Sigmoid controller, suitable for producing a coordinated motion behaviour, could be re-used to produce a light pursuing behaviour is that both behaviours represent a form of *taxis*, that is behaviours that drive the robots toward a certain direction on the horizontal plane.

The arbitration between the two motor schemas of the controller was accomplished by averaging the output produced by them. This is a typical solution adopted for motor schema-based controllers (cf. Arkin, 1989; Arkin, 1998; note that other solutions have been proposed within the behaviour-based robotics literature, such as the hierarchical arbitration mechanism used in the popular subsumption architecture, Brooks, 1986). Note that a sort of averaging arbitration mechanism also emerged in the neural controllers evolved from scratch (cf. Baldassarre et al., 2004).

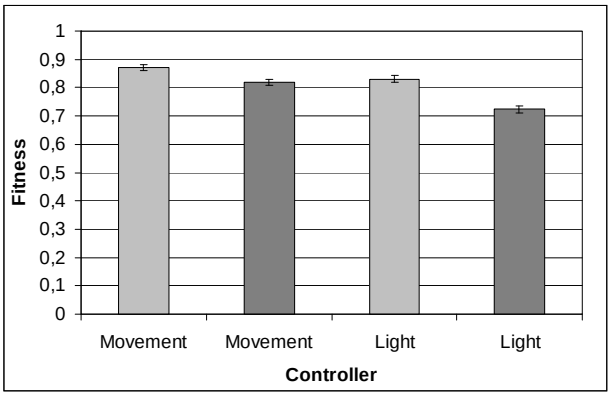


Figure 13: Performance (y-axis) of the evolved neural-network controller (light grey bars) and of the double motor schema-based controller (dark grey bars) in a coordinated motion test ("Movement") and in a light approaching test ("Light"). Each histogram bar reports the average performance and the standard error of the controllers obtained in 100 trials. The data relative to the evolved neural controllers are those produced by the experiments described in detail in Baldassarre et al. (2004).

Visual inspection of the behaviour exhibited by robots provided with this modular controller indicates that when robots do not perceive the light, they display a smooth coordinated motion behaviour. As soon as robots start to detect the light, that is as soon as the distance between them and the light is below 400 cm and they are not shadowed by other robots, they starts moving toward the direction of the light. This motion generates traction forces that are detected by the other robots of the group (in particular by those in shadow) so that they turn accordingly and the whole group ends up moving toward the light in a finely coordinated fashion.

In order to quantify the performance of this behaviour, the new motor schema-based controller was tested with and without the light target, and the results obtained by robots provided with the double motor schema-based controller we compared with those obtained with neural controllers evolved from scratch (data from Baldassarre et al., 2004). As shown in Figure 13, in both tests the performance of the

double motor schema-based controller is rather good but lower than the performance displayed by the evolved neural controller (t-tests, $p < 0.01$).

To analyse the role of each of the two motor schema controllers and to understand how the average of their output could produce an effective behaviour, we plotted the potential gradient fields generated by the coordinated motion schema-based controller alone (Figure 14a), the coordinated light-pursuing schema-based controller alone (Figure 14b), and by the combination of the two schema-based controllers arbitrated by averaging their output patterns (Figure 14c). As specified above, the two schema-based controllers were implemented by using two Rounded Sigmoid controllers which respectively received as input the direction and the intensity of traction and the direction and intensity of light.

Concerning the coordinated motion controller Figure 14a shows the motor reactions produced by the controller for traction forces with different intensities and directions. Each arrow of the graph represents the controller reaction to a certain intensity and direction of traction. In particular, a certain intensity of traction is proportional to the closeness of the position of the arrow to the *centre* of the graph, whereas the direction of traction is represented by the direction going from the arrow position to such centre. The length and the orientation of each arrow represent, respectively, the change of position and the change of orientation of the robot that is produced by setting (for 0.75 s) the desired speed of its two wheels to the values produced by the controller with the intensity and direction of traction corresponding to the arrow position. This analysis confirms that, as indicated in section 3, when traction comes from the robot's front or rear, or when its intensity is low, the robot tends to move straight. On the contrary, when traction comes from either the robot's left or right hand side, and the intensity of the traction is significant, the robot tends to turn toward the direction of the traction (by consequently also reducing the extent of the displacement).

In relation to the coordinated light-pursuing behaviour, Figure 14b shows the reactions produced by the controller for different sensory states corresponding to different orientations and distances of the light target. The distance (which was varied within the range of [0, 141] cm) of the light target is represented by the distance of the arrow position from the *centre of the north-wall side* of the graph, whereas the direction of the light target is represented by the direction going from the arrow position to such centre. Analogously to what done for the previous graph, the reaction of the controller is indicated by the length and the orientation of the arrows. These represent, respectively, the change of position and the change of orientation of the robot produced in 0.75 s by setting the desired speed of the two wheels to the value produced by the controller with a light position and distance corresponding to the arrow position. The graph show that a robot moves fast towards the light when this is located in front of it, whereas it turns toward the direction of the light when this is located on its left or right hand side.

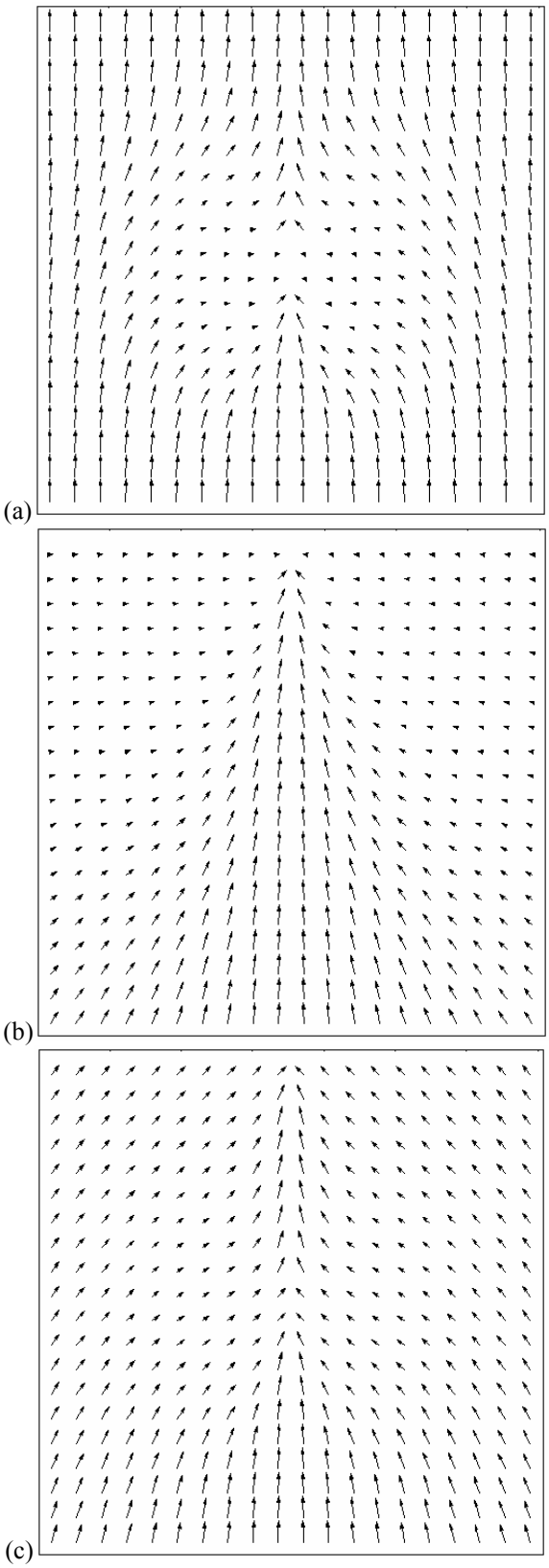


Figure 14: Potential fields generated by different motor schemas built on the basis of the Rounded Sigmoid controller. (a) Motor schema able to produce group coordination behaviours. (b) Motor schema able to produce light approaching behaviours. (c) Controller formed by the two motor schemas shown in graphs "a" and "b" arbitrated by averaging their output patterns.

Finally, with regards to the overall controller, Figure 14c shows the average of the output patterns produced by the two behaviours for different direction and intensities of the traction (represented, as in the case of Figure 14a, with respect to a traction coming from the centre of the graph) and different orientations and distances of the light target (imagined to be positioned, as in the case of Figure 14a, at the centre of the north side of the graph). The graph shows how the tendency to navigate towards the light target and the tendency to align with the rest of the group are smoothly integrated in the resulting potential field. For example, in the outer portions of the graph, which correspond to a situation in which the robot is rather aligned with the rest of the group (signalled by a low traction intensity), the reaction of the robot mainly take into account the direction of the light. Conversely, in the central part of the graph, which corresponds to situations in which there is a significant mismatch between the orientation of the robot and that of the rest of the group (signalled by a high traction intensity), the robot mainly responds to the direction of traction. Interestingly, in situations in which there is a potential conflict between the direction of the light and the direction of the group motion (corresponding to top-central part of the graph) the potential field resulting from the average of the outputs of the two controllers does not lead to fixed points or limit-cycle behaviours which could prevent the accomplishment of the task (these are typical problems which can arise when schema-based controllers are combined, Ge and Cui, 2002).

To summarise, the results shown in this section demonstrate how evolved controllers can be exploited as templates for designing hand-coded controllers within the framework of the motor schema-based approach so as to solve new problems without the need to re-evolve solutions from scratch.

8. The path to implementation in real robots

The arguments presented in the paper rely on experiments carried out on simulated swarm-bots. The reason for using simulations in this work resided primarily in its peculiarly theoretical and methodological aims. In previous works, the evolved controller capable of performing coordinated motion was successfully tested in hardware in a wide variety of conditions (Baldassarre et al., 2007). These tests demonstrated the robustness of the controller evolved in simulation by showing how it was possible to transfer it to real robots with full success without the need of any modification. The strength of the core mechanisms underlying the coordinated motion controller presented here were also tested in other real robotic set-ups that used them as "building-blocks" for evolving more complex behaviours (e.g., Trianni et al., 2006). Finally, the flexibility of the real s-bot developed within the project SWARM-BOTS was also tested with hand-coded controllers (e.g., Groß et al., 2006). The work that led to this wide spectrum of results (cf. the web site of the project for other references of works carried out with real s-bots: http://www.swarm-bots.org/) on one side indicated the importance of expanding the theoretical understanding of the evolved controllers through new methods of analysis (like those described here) and, on the other side, suggested to carry out a close comparison of evolved controllers with hand-coded ones.

9. Discussion and conclusions

At the current state of the art, self-organising methods, such as those proposed within the evolutionary robotics literature, and direct design methods, such as behaviour-based robotics, have both strengths and weaknesses. A major strength of self-organising methods, as shown in this paper, is the ability to discover effective solutions that exploit properties of the system that can hardly be identified by a human designer. With this respect, the experiment described in section 3 shows how evolving robots were able to discover a simple strategy that allows a group of physically assembled robots to produce a very efficient coordinate motion behaviour. Some fine details of the found solution were hard to identify. This was demonstrated by the fact that the hand-coded controllers, obtained by approximating the sensory-motor mapping produced by evolved neural controllers with a suitable statistical procedure, produced a performance lower than the performance of the latter ones (see sections 5 and 7), even if the residual error of the regression was very low (see section 4). In general, the reason why self-organising methods such as those used by evolutionary robotics can generate solutions that are difficult to imagine for an human designer is that, through random variation and selection, they can discover and capitalize on useful properties emerging from the complex fine-grained interactions between the robots and their environment, including the social environment formed by other robots (Nolfi and Floreano, 2000; Funes et al., 2004; Nolfi, 2006). The main weakness of self-organising methods is that they do not guarantee that a given problem will be actually solved even if an effective solution exists. In fact, to be successful evolutionary methods not only require that an effective solution of the problem exists, but also that a chain of intermediate adaptive solutions, that represent crucial steps toward the final solution, can be discovered through progressive variations (Nolfi and Floreano, 2000).

The main strength of hand-coded controllers reside in the fact that, by being easier to understand from the point of view of the experimenter and by usually presenting a strong modular organization, they can be directly programmed, modified and combined to produce new controllers, or new variations of existing controllers, in an incremental fashion. Hence, in general, hand-coded controllers can be more easily scaled to increasing complex tasks. On the other side, the main weakness of hand-coded controllers resides in the difficulty of identifying the "micro rules" that should regulate the fine-grained interaction between the robot and the environment that can lead to the desired behaviour.

This paper proposed a method for combining the strengths of self-organising methods and direct design

methods. In particular it showed how effective solutions discovered through an evolutionary technique can be recoded in equation-based controllers, such as motor schema-based controllers, that can be later manipulated and combined to produce new behaviours. In this respect, as demonstrated in section 4, equation-based controllers functionally analogous to evolved feed-forward neural controllers can be obtained by: (a) identifying a function which approximates the general characteristics of the mapping performed by the evolved neural controller; (b) approximating the function parameters through non-linear regression methods. The recoding of the neural controllers in equation-based controllers can produce a loss of performance. However, as we have shown, the process of recoding can favour the identification of crucial parameters that can be later varied to analyse the characteristics of the evolved solution and to develop new controllers able to produce new behaviours. With respect to the last point, section 6 and 7 showed how it is possible to obtain new variations of evolved behaviours, and completely new behaviours, by varying the parameters of the equation-based controllers or by combining different equation-based controllers.

The parameters of the equation-based controllers can be varied either manually (section 6.2), after their functional role has been identified (section 5), or through semi-automatic techniques based on interactive evolutionary computation algorithms (section 6.1). In the latter case the parameters are varied randomly but variations are retained or discarded on the basis of their effects evaluated by visually inspecting the resulting behaviours. In both cases, the identification of few crucial parameters is essential to keep the search space within a reasonable size. Finally, as shown in section 7, equation-based controllers which approximate evolved solutions can be fruitfully used as building blocks and combined to solve problems that require different related abilities (e.g., coordinated motion and coordinated light approaching).

The proposed method should have, at least in principle, a general validity and could therefore be applied successfully to different robotic set-ups and to problems different from those studied in the present paper, at least in the cases in which the evolved neural controllers consist of feed-forward neural architectures.

To the best of the authors knowledge, this paper presents the first work which explicitly combines evolutionary synthesis approaches with human design or "guidance" (as done in IEC techniques) not only at the level of fine-grained solutions but also at the level of the whole robot control architecture.

Acknowledgements

This research has been supported by the "SWARM-BOTS" and "SWARMANOID" projects funded by the Future and Emerging Technologies program (IST-FET) of the European Commission under grants IST-2000-31010 and IST-022888, respectively.

References

1. Arkin, R. C. (1989). Motor schema based mobile robot navigation. *International Journal of Robotics Research*. Vol. 8, 92-112.
2. Arkin, R.C. (1998). *Behaviour-Based Robotics*. Cambridge, MA: MIT Press.
3. Balch, T., Arkin, R. C. (1998). Behaviour-based formation control for multirobot teams. *IEEE Transactions on Robotics and Automation*. Vol. 14, no. 6, pp. 926-939.
4. Baldassarre, G. (2008). Self-organization as phase transition in decentralized groups of robots: a study based on Boltzmann entropy. In Prokopenko, M. (ed.), *Advances in Applied Self-Organising Systems*, pp. 127-146. Berlin: Springer-Verlag.
5. Baldassarre, G., Nolfi, S., Parisi, D. (2003). Evolution of collective behaviour in a group of physically linked robots. In Raidl, G., Guillot, A., Meyer, J.-A. (eds.), *Applications of Evolutionary Computing - Proceedings of the Second European Workshop on Evolutionary Robotics,* pp. 581-592. Berlin: Springer Verlag.
6. Baldassarre, G., Nolfi, S., Parisi, D. (2003). Evolving mobile robots able to display collective behaviour. *Artificial Life.* Vol. 9, pp. 255-267.
7. 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 8: Proceedings of the 8th International Conference on Simulation of Adaptive Behaviour*, pp. 385-394. Cambridge, MA: MIT Press.
8. Baldassarre, G., Parisi, D., Nolfi, S. (2004b). Measuring coordination as entropy decrease in groups of linked simulated robots. In Minai, A., Bar-Yam, Y. (eds.), *Proceedings of the Fifth International Conference on Complex Systems (ICCS2004)*. Cambridge, MA: The New England Complex Systems Institute. Published online at http://www.necsi.org/events/iccs/2004proceedings.html.
9. Baldassarre, G., Parisi, D., Nolfi, S. (2006). Distributed coordination of simulated robots based on self-organization. *Artificial Life*. Vol. 12, no. 3, pp. 289-311.
10. Baldassarre, G., Trianni, V., Bonani, M., Mondada, F., Dorigo, M., Nolfi, S. (2007). Self-organised coordinated motion in groups of physically connected robots. *IEEE Transactions in Systems, Man and Cybernetics – Part B Cybernetics.* Vol. 37, no. 1, pp. 224-239.
11. Barfoot, T. D., Clark, C. M. (2004). Motion planning for formations of mobile robots. *Journal of Robotics and Autonomous Systems.* Vol. 46, pp. 65–78.
12. Beckers, R., Holland, O., Deneubourg, J.-L. (1994). From local actions to global tasks: Stigmergy and collective robotics. In Brooks, R. A., Maes, P. (eds.), *Proceedings of the 4th International Workshop on the Synthesis and Simulation of Living Systems (ALife IV)*, pp. 181–189. Cambridge, MA: MIT Press.
13. Bonabeau, E., Dorigo, M., Theraulaz, G. (1999). *Swarm intelligence: from natural to artificial system*. Oxford: Oxford University Press.

14. Brooks, R. A. (1986). A robust layered control system for a mobile robot. *Journal of Robotics and Automation*. Vol. 2, pp. 14-23.
15. Camazine, S., Deneubourg, J. L., Franks, N. R., Sneyd J., Theraulaz, G., Bonabeau, E. (2001). *Self-organization in biological systems*. Princeton, NJ.: Princeton University Press.
16. Cao, Y. U., Fukunaga, A. S., Kahng, A. B. (1997). Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*. Vol. 4, pp. 1-23.
17. Cliff, D., Harvey, I., Husbands, P. (1993). Explorations in evolutionary robotics. *Adaptive Behaviour*. Vol. 2, pp. 73-110.
18. Cybenko, G. (1989). Approximation by superposition of sigmoidal functions. *Mathematics of control, signals, and systems*. Vol. 2, pp. 303-314.
19. Desai, J. P., Ostrowski, J. P., Kumar, V. (2001). Modeling and control of formations of nonholonomic mobile robots. *IEEE Transactions on Robotics and Automation*. Vol. 17, no. 6, pp. 905-908.
20. Dorigo, M., Sahin, E. (2004). Swarm robotics – Special issue editorial. *Autonomous Robots*. Vol. 17, no. 2-3, pp. 111-113.
21. Dorigo, M., Trianni, V., Sahin, E., Groß, R., Labella, T. H., Baldassarre, G., Nolfi, S., Deneubourg, J.-L., Mondada, F., Floreano, D., Gambardella, L. M. (2004). Evolving self-organising behaviours for a swarm-bot. *Autonomous Robots*. Vol. 17, no. 2-3, pp. 223-245.
22. Dudek, G., Jenkin, M., Milios, E. (2002). A taxonomy of multirobot systems. In Balch, T., Parker, L. E. (eds.), *Robot teams: From diversity to polymorphism*. Wellesley, MA: A K Peters Ltd.
23. Fredslund, J., Mataric, M. J. (2002). A general algorithm for robot formations using local sensing and minimal communication. *IEEE Transactions on Robotics and Automation*. Vol. 18, no. 5, pp. 837-846.
24. Funes, P., Orme, B., Bonabeau, E. (2004). Shaping collective behaviour: An exploratory design approach. In Pollack, J., Bedau, M., Husbands, P., Ikegami, T., Watson, R. (eds.). *Artificial Life IX: Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems*, pp. 232-237. Cambridge, MA: MIT Press.
25. Ge, S., Cui, Y. (2002). Dynamic motion planning for mobile robots using potential field method. *Autonomous Robots*. Vol. 13, pp. 207-222.
26. Grabowski, R., Navarro-Serment, L. E., Khosla, P. K. (2003). An army of small robots. *Scientific American*, November, pp. 42-47.
27. Groß, R., Bonani, M., Mondada, F., Dorigo, M. (2006). Autonomous self-assembly in mobile robotics. In Murase, K., Sekiyama, K., Kubota, N., Naniwa, T., Sitte, J. (eds.), *Proceedings of the Third International Symposium on Autonomous Minirobots for Research and Edutainment,* pp. 314-322. Berlin: Springer Verlag.
28. Holland, O., Melhuish, C. (1999). Stigmergy, self-organization, and sorting in collective robotics. *Artificial Life*. Vol. 5, no. 2, pp. 173-202.
29. Ijspeert, A. J., Martinoli, A., Billard, A., Gambardella L. M. (2001). Collaboration through the exploitation of local interactions in autonomous collective robotics: the stick pulling experiment. *Autonomous Robots*. Vol. 11, pp. 149-171.
30. Janet, J. A., Gutierrez, R., Chase, T. A., White, Mark, W., Sutton, J. C. (1997). Autonomous mobile robot global self-localization using Kohonen and region-feature neural networks. *Journal of Robotic Systems.* Vol. 14, no. 4, pp. 263-282
31. Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research.* Vol. 5, no. 1, pp. 90-98.
32. Krieger, M. J. B., Billeter, J.-B., Keller, L. (2000). Ant-like task allocation and recruitment in cooperative robots. *Nature*. Vol. 406, pp. 992–995.
33. Kube, C. R., Zhang, H. (1993). Collective robotics: From social insects to robots. *Adaptive Behaviour.* Vol. 2, no. 2, pp. 189-219.
34. Kube, R. C., Bonabeau, E. (1998). Cooperative transport by ants and robots. *Robotics and autonomous systems.* Vol. 30, pp. 85-101.
35. Martinoli, A. (1999). *Swarm intelligence in autonomous collective robotics: From tools to the analysis and synthesis of distributed control strategies*. Ph.D. dissertation. Lausanne: Computer Science Department, Ecole Polytechnique Fédérale de Lausanne.
36. Matarić, M. J. (1997). Reinforcement learning in the multi-robot domain. *Autonomous Robots*, Vol. 4, no. 1, pp. 73-83.
37. Miglino, O., Lund, H. H., Nolfi, S. (1995). Evolving mobile robots in simulated and real environments. *Artificial Life.* Vol. 4, 417-434.
38. Mondada, F., Pettinaro, G. C., Guignard, A., Kwee, I. V., Floreano, D., Deneubourg, J.-L., Nolfi, S., Gambardella, L. M., Dorigo, M. (2004). SWARM-BOT: A new distributed robotic concept. *Autonomous Robots*. Vol. 17, no. 2-3, pp. 193-221
39. Nolfi, S. (2006). Behaviour as a complex adaptive system: on the role of self-organization in the development of individual and collective behaviour. *ComplexUs*. Vol. 2, no. 3-4, pp. 195-203.
40. Nolfi, S., Floreano, D. (2000). *Evolutionary Robotics: the biology, intelligence, and technology of self-organising machines*. Cambridge, MA: MIT Press.
41. Park, J., Sandberg, I. (1991). Universal approximation using radial-basis-function networks. *Neural computation*. Vol. 3., pp. 246-257.
42. Quinn, M., Smith, L., Mayley, G., Husband, P. (2002). Evolving teamwork and role allocation with real robots. In Standish, R. K., Bedau, M. A., Abbass, H. A. (eds.), *Proceedings of the 8th International Conference on Artificial Life,* pp. 302-311. Cambridge, MA: MIT Press.
43. Quinn, M., Smith, L., Mayley, G., Husbands, P. (2003). Evolving controllers for a homogeneous system of physical robots: Structured cooperation with minimal sensors. *Philosophical Transactions of the Royal Society of London, Series A*. Vol. 361, pp. 2321-2344.

44. Reynolds, C. W. (1987). Flocks, herds, and schools: a distributed behavioural model. *Computer Graphics*. Vol. 21, no. 4, pp. 25-34.
45. Spector, L., Klein, J., Perry, C., Feinstein, M. (2005). Emergence of collective behaviour in evolving populations of flying agents. G*enetic Programming and Evolvable Machines*. Vol. 6, no. 1, pp. 111-125.
46. Strogatz, S. (2003). *Sync: the emerging science of spontaneous order*. New York: Hyperion.
47. Takagi, H. (2001). Interactive evolutionary computation: fusion of the capabilities of EC optimization and human evaluation. *Proceedings of the IEEE*. Vol. 89, pp. 1275-1296.
48. Trianni, V., Ampatzis, C., Christensen, A. L., Tuci, E., Dorigo, M., Nolfi, S. (2007). From solitary to collective behaviours: decision making and cooperation. In Almeida e Costa, F., Rocha Luis, M., Costa, E., Harvey, I., Coutinho, A. (eds.), *Proceedings of the 9th European Conference on Artificial Life (ECAL 2007)*, pp. 575-584. Berlin: Springer Verlag.
49. Trianni, V., Dorigo, M. (2006). Self-Organisation and Communication in Groups of Simulated and Physical Robots. *Biological Cybernetics*. Vol. 95, pp. 213-231.
50. Tuci, E., Groß, R., Trianni, V., Mondada, F., Bonani, M., Dorigo, M. (2006). Cooperation through self-assembling in multi-robot systems. *ACM Transactions on Autonomous and Adaptive Systems*. Vol. 1, no. 2, pp. 115-150.
51. Tummolini, L., Midolli, M., Castelfranchi, C. (in press). Stigmergic cues and their uses in coordination: an evolutionary approach. In Uhrmacher, A. M., Weyns, D. (eds.), *Agents, Simulations and Applications.* London: Taylor & Fransis.
52. Turgut, A. E., Huepe, C., Çelikkanat, H., Gökçe, F., Sahin, E. (2008). Modeling phase transition in self-organised flocks. In Dorigo, M., Birattari, M., Blum, C., Clerc, M., Stützle, T., Winfield, A. F. T. (ed.), *Proceedings of the 6th International Conference on Ant Colony Optimization and Swarm Intelligence (ANTS 2008),* LNCS 5217, pp. 108-119. Berlin: Springer-Verlag.
53. Wang, P. K. C. (1991). Navigation strategies for multiple autonomous mobile robots moving in formation. *Journal of Robotic Systems*. Vol. 8, no. 2, pp. 177-195.
54. Wang, Z. D., Nakano, E., Takahashi, T. (2003). Solving function distribution and behaviour design problem for cooperative object handling by multiple mobile robots. *IEEE Transactions on Systems, Man and Cybernetics - Part A*. Vol. 33, no. 5, pp. 537-549.
55. Ward, C. R., Gobet, F., Kendall, G. (2001). Evolving collective behaviour in an artificial ecology. *Artificial Life*. Vol. 7, no. 1, pp. 191-209.
56. Yamashita, Y., Tani, J. (2008). Emergence of functional hierarchy in a multiple timescale neural network model: a humanoid robot experiment. *PLoS Computational Biology*. Vol.4, no. 11.