From Solitary to Collective Behaviours: Decision Making and Cooperation

Experimental Setup

Vito Trianni[1], Christos Ampatzis[2], Anders Lyhne Christensen[2], Elio Tuci[2], Marco Dorigo[2], and Stefano Nolfi[1]

[1] LARAL, ISTC, CNR, Italy {vito.trianni,stefano.nolfi}@istc.cnr.it
[2] IRIDIA, CoDE, ULB, Belgium {campatzi,alyhne,etuci,mdorigo}@ulb.ac.be

This document is intended to support the paper submitted to the 9th European Conference on Artificial Life (ECAL 2007) [1]. It provides a detailed description of the experimental setup. In Section 1, the configuration of the *s-bot* is given. In Section 2, we describe the controller and the evolutionary algorithm used. In Section 3, we accurately describe the evaluation function used.

1 The *S-bot*

The *s-bot* is a small mobile autonomous robot with self-assembling capabilities (see Fig. 1). The details about the hardware and electronics of the *s-bot* can be found in [2], and in the SWARM-BOTS project website (`http://www.swarm-bots.org`). In this document, we give some details about the sensors and actuators used for the experiment presented in [1].

Each *s-bot* is provided with four proximity sensors placed under the chassis—referred to as *ground sensors*—that can be used for perceiving the ground's grey level (see Fig.1 top-right). When the sensor is placed over white ground, it returns a high value due to the high reflectivity of the ground. On the contrary, if the ground colour is black, the reflectivity is low and consequently the sensor returns a value close to 0. The raw sensor readings are recorded and scaled in the interval [0,1] before being processed by the neural controller.

Each robot is also equipped with an omni-directional camera, which is used to perceive the presence and the corresponding distance of neighbouring *s-bots*. The omni-directional camera can perceive the red colour continuously emitted by the *s-bots* by means of their coloured LEDs embedded in the T-shaped ring (see Fig. 1). The circular image obtained from the camera is filtered in order to extract only the red objects. Then, it is split in 4 sectors of 90° each (front-left, front-right, rear-left, rear-right) and the distance of the closest red object in each sector is computed. With such a system, the closest *s-bot* in each sector can be perceived up to a distance of about 50 *cm*. Also in this case, distances are scaled in the interval $[0, 1]$ before being processed by the neural controller.

In order to communicate with each other, *s-bots* are provided with a very simple signalling system, which can produce a continuous tone with fixed frequency and intensity. When a tone is emitted, it is perceived by every robot in the arena, including the signalling *s-bot*. The tone is perceived in a binary way, that is, either some *s-bot* is signalling in the arena, or no one is.

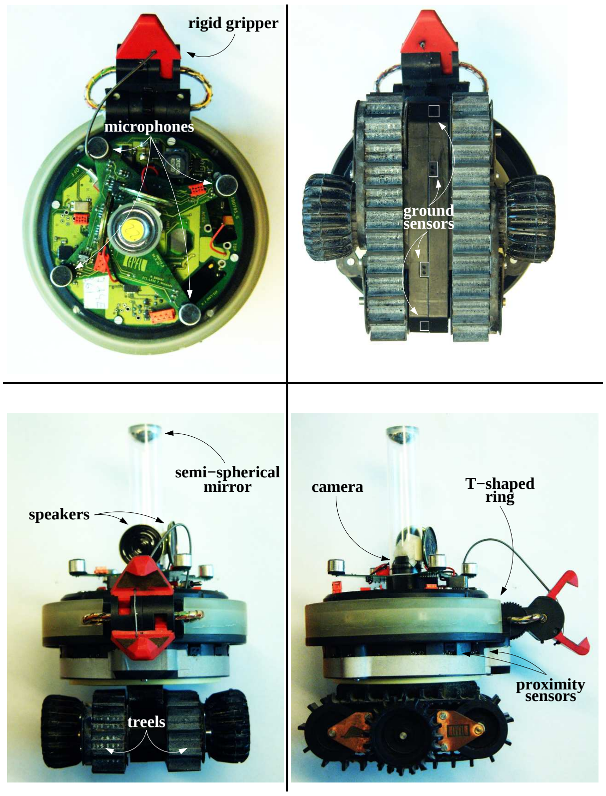

Fig. 1. View of the *s-bot* from different sides. The main components are indicated.

Notwithstanding the efforts to devise a precise simulation, some characteristics of the robots and of the robot-environment interaction may escape the modelling phase. For this reason, noise is used to ensure that the evolved behaviour will cope with differences between simulation and reality [3]. Except for the binary communication system, noise is simulated for all sensors and actuators, adding a random value uniformly distributed in the interval $[-5\%, 5\%]$ with respect to the maximum value.

2 The Controller and The Evolutionary Algorithm

Homogeneous groups of *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 individuals with an identical control structure. Each *s-bot* is controlled by a continuous time recurrent neural network (CTRNN) [4]. The neural network has a multi-layer topology, as shown in Fig. 2: neurons $N_{I,1}$ to $N_{I,9}$ take input from the robot's sensory apparatus, neurons $N_{O,1}$ to $N_{O,3}$ control the robot's actuators, and neurons $N_{H,1}$ to $N_{H,5}$ form a fully recurrent continuous time hidden layer. The input neurons are simple relay units, while the output neurons are governed by the following equations:

$$o_j = \sigma(O_j + \beta_j), \qquad O_j = \sum_{i=1}^{5} W_O(i,j)\, \sigma(H_i + \beta_i), \qquad \sigma(z) = \frac{1}{1+e^{-z}}, \quad (1)$$

where, using terms derived from an analogy with real neurons, O_j and H_i are the cell potentials of respectively output neuron j and hidden neuron i, β_j and β_i are bias terms, $W_O(i,j)$ is the strength of the synaptic connection from hidden neuron i to output j, and o_j and $h_i = \sigma(H_i + \beta_i)$ are the firing rates. The hidden units are governed by the following equation:

$$\frac{dH_j}{dt} = \frac{1}{\tau_j}\left(-H_j + \sum_{i=1}^{5} W_H(i,j)\sigma(H_i + \beta_i) + \sum_{i=1}^{9} W_I(i,j) I_i\right), \qquad (2)$$

where τ_j is the decay constant, $W_H(i,j)$ is the strength of the synaptic connection from hidden neuron i to hidden neuron j, $W_I(i,j)$ is the strength of the connection from input neuron i to hidden neuron j, and I_i is the intensity of the sensory perturbation on neuron i.

Four input neurons—$N_{I,1}$ to $N_{I,4}$—are set looking at the four sectors of the image grabbed by the omni-directional camera, as explained in Section 1. Four other input neurons—$N_{I,5}$ to $N_{I,8}$—are set directly from the four ground sensors. Finally, input neuron $N_{I,9}$ is a binary input set by the perception of a sound signal. The neurons $N_{O,1}$ and $N_{O,2}$ are used to set the speed of the *s-bot*'s wheels. Neuron $N_{O,3}$ is used to set the state of the loudspeaker, which is turned on if the neuron output is higher than 0.5, and off otherwise. The weights of the connection between neurons, the bias terms and the decay constants are genetically encoded parameters. Cell potentials are set to 0 each time a network is initialised or reset. State equations are integrated using the forward Euler method with an integration step-size of 0.1 seconds.

In order to set the parameters of the *s-bot*' controllers, a simple generational evolutionary algorithm is employed [5]. The population contains 100 genotypes

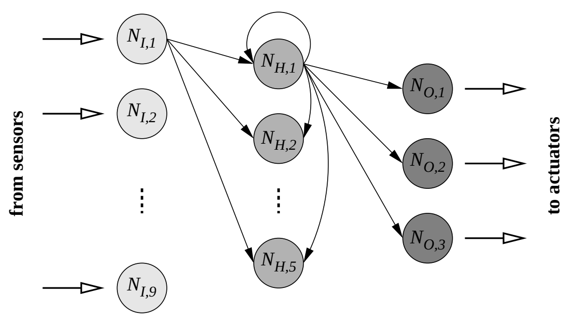

Fig. 2. The multi-layer topology of the neural controller. The hidden layer is composed of continuous time neurons with fully recurrent connections.

that are evolved for 5000 generations. Each genotype is a vector of 98 real values (85 synaptic connections, 5 decay constants and 8 bias terms) that are initially chosen uniformly random from the range $[-10, 10]$. 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 4 offspring. The genotype of the selected parents is copied in the subsequent generation; the genotype of the 4 offspring is mutated with a 50% probability of adding a random Gaussian offset $N(0, 1)$ to each real-valued gene. During evolution, genotype parameters are constrained to remain within the range $[-10, 10]$. They are mapped to produce CTRNN parameters with the following ranges: connection weights $W(j, i) \in [-4, 4]$; biases $\beta \in [-4, 4]$; concerning decay constants, the genetically encoded parameters are firstly mapped onto the range $[-1, 3]$ and then exponentially mapped onto $\tau \in [10^{-1}, 10^3]$. The lower bound of τ corresponds to the integration step size used to update the controller; the upper bound is arbitrarily chosen and it is bigger than the maximum length of a trial.

3 The Evaluation Function

During the evolution, a genotype is mapped into a control structure that is cloned and downloaded in all the *s-bots* taking part to the experiment (i.e., we use a homogeneous group of *s-bots*). Groups of 3 *s-bots* are evaluated 10 times—i.e., 10 trials, 5 performed in environment A and 5 in environment B. Each trial lasts 65 seconds and differs from the others in the initialisation of the random number generator, which influences mainly the *s-bots* starting positions and orientations and the *way out* amplitude, if present.

The evaluation function takes into account the behavioural state in which the *s-bots* should be and it rewards their movements accordingly. When *s-bots* are placed in environment A, they should search for and traverse the *way out*, therefore they should always be in state $\mathcal{S}$. When *s-bots* are placed in environment B, they should initially search for the *way out*, being in state $\mathcal{S}$, and after some searching they should aggregate, therefore switching to state $\mathcal{C}$. In order to evaluate the behaviour in environment B, we ignore the time needed for searching the *way out* and we consider that an *s-bot* switches to state $\mathcal{C}$ when it encounters the circular band for the first time. In this way, we can systematically evaluate the movements of an *s-bot* according to its behavioural state:

$$\mathcal{S}(s, t) = \text{environment } A \textbf{ OR } d_M(s, t) < 1.0, \qquad \mathcal{C}(s, t) = \textbf{NOT } \mathcal{S}(s, t), \quad (3)$$

where $d_M(s, t)$ is the maximum distance from the centre reached by *s-bot* s at time t. In other words, an *s-bot* is considered to be in state $\mathcal{S}$ if it is placed in environment A or if the maximum distance it reached from the centre of the arena is smaller than one meter, which corresponds to the inner radius of the circular band. Otherwise, an *s-bot* is considered to be in state $\mathcal{C}$. Having defined the behavioural states at time t, an *s-bot* s should maximise its distance from the centre of the arena when in state $\mathcal{S}$, while it should minimise its distance from the centre of mass of the group when in state $\mathcal{C}$. Therefore, for each *s-bot* s at step t, we compute the measure $d(s,t)$ according to the following equation:

$$d(s,t) = \begin{cases} ||\mathbf{X}(s,t) - \mathbf{X}_o|| & \text{if } \mathcal{S}(s,t), \\ 1.0 - ||\mathbf{X}(s,t) - \mathbf{X}_c(t)|| & \text{if } \mathcal{C}(s,t), \end{cases} \tag{4}$$

where $\mathbf{X}(s,t)$ are the coordinates of *s-bot* s at time t, $\mathbf{X}_o$ and $\mathbf{X}_c(t)$ are the coordinates of the centre of the arena and of the centre of mass of the *s-bots*. Therefore, an *s-bot* should always maximise $d(s,t)$ in order to reach the optimal position: in state $\mathcal{S}$, an *s-bot* should move away from the centre, and it is considered successful if it reaches an optimal distance $D_O(\mathcal{S}) = 2.4$ m (i.e., $d(s,t) \geq D_O(\mathcal{S})$); in state $\mathcal{C}$, an *s-bot* should aggregate with the other robots by reducing its distance from the centre of mass of the group, and it is considered successful if it stays below an optimal distance $D_O(\mathcal{C}) = 0.25$ m (i.e., $d(s,t) \geq 1.0 - D_O(\mathcal{C})$). We measure a normalised distance $\tilde{d}(s,t)$ according to the behavioural state as follows:

$$\tilde{d}(s,t) = \begin{cases} \Theta\left(\dfrac{d(s,t)}{D_O(\mathcal{S})}\right) & \text{if } \mathcal{S}(s,t), \\ \\ \Theta\left(\dfrac{d(s,t)}{1.0 - D_O(\mathcal{C})}\right) & \text{if } \mathcal{C}(s,t), \end{cases} \tag{5}$$

where $\Theta(x)$ simply bounds the value of x in the interval $[0,1]$. In both behavioural states, $\tilde{d}(s,t) = 1$ indicates that *s-bot* s at least reached the optimal distance D_O at time t. We conventionally say that a successful *s-bot* "achieves the optimal distance D_O".

In order to evolve the desired behaviour, we compute two measures that reward the *s-bot*'s movements both for its absolute position and for the stepwise increment of the $d(s,t)$:

$$f_d(s,t) = \tau_d \cdot f_d(s,t-1) + (1-\tau_d) \cdot \tilde{d}(s,t), \tag{6}$$

$$f_i(s,t) = \frac{d(s,t) - d(s,t-1)}{2d_M} + 0.5, \tag{7}$$

where $\tau_d = 0.975$ is the time constant of a moving average, and d_M is the maximum distance increment that an *s-bot* can cover in a single simulation cycle. The measure $f_d(s,t)$ rewards the *s-bot* for the absolute position reached, and the moving average is justified by the necessity to reward behaviours that keep the optimal distance for a long time (which also justifies the high value we have chosen for the time constant τ_d). Differently, the measure $f_i(s,t)$ rewards the *s-bot* for the stepwise increments toward an optimal position. Notice that, while in state $\mathcal{S}$ robots should continue to move away from the centre of mass even if they achieved the optimal distance $D_O(\mathcal{S})$, in $\mathcal{C}$ *s-bots* cannot decrease further their distance from the centre of mass once the optimal distance $D_O(\mathcal{C})$ is reached. For this reason, we set $f_i(s,t) = 1.0$ when the *s-bot* is in state $\mathcal{C}$ and $\tilde{d}(s,t) = 1$.

Given the above measures computed for all *s-bots* and for all simulation cycles, the fitness in a trial is computed as follows:

$$F = \frac{1}{N}\sum_{s=1}^{N} f_d(s,T) \cdot \frac{1}{NT}\sum_{s=1}^{N}\sum_{t=1}^{T} f_i(s,t), \tag{8}$$

where $N = 3$ is the number of *s-bots* and $T = 650$ is the number of simulation cycles of the trial. Note that a trial is terminated whenever an *s-bot* passes over the black border of the circular band—and in this case $F = 0$—or if *s-bots* collide when in state $\mathcal{S}$.

References

1. Trianni, V., Ampatzis, C., Christensen, A.L., Tuci, E., Dorigo, M., Nolfi, S.: From solitary to collective behaviours: Decision making and cooperation. In: Advances in Artificial Life, Proceedings of the 9th European Conference on Artificial Life (ECAL'07). (2007) submitted.
2. Mondada, F., Pettinaro, G.C., Guignard, A., Kwee, I.V., Floreano, D., Deneubourg, J.L., Nolfi, S., Gambardella, L.M., Dorigo, M.: SWARM-BOT: A new distributed robotic concept. Auton. Robots **17**(2–3) (2004) 193–221
3. Jakobi, N.: Evolutionary robotics and the radical envelope of noise hypothesis. Adaptive Behavior **6** (1997) 325–368
4. Beer, R.D.: A dynamical systems perspective on agent-environment interaction. Art. Intell. **72** (1995) 173–215
5. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading, MA (1989)