

# Communication in a Swarm of Miniature Robots: The e-Puck as an Educational Tool for Swarm Robotics

Christopher M. Cianci, Xavier Raemy, Jim Pugh, and Alcherio Martinoli

Swarm-Intelligent Systems Group  
École Polytechnique Fédérale de Lausanne  
CH-1015 Lausanne, Switzerland

{chris.cianci,xavier.raemy,jim.pugh,alcherio.martinoli}@epfl.ch

**Abstract.** Swarm intelligence, and swarm robotics in particular, are reaching a point where leveraging the potential of communication within an artificial system promises to uncover new and varied directions for interesting research without compromising the key properties of swarm-intelligent systems such as self-organization, scalability, and robustness. However, the physical constraints of using radios in a robotic swarm are hardly obvious, and the intuitive models often used for describing such systems do not always capture them with adequate accuracy. In order to demonstrate this effectively in the classroom, certain tools can be used, including simulation and real robots. Most instructors currently focus on simulation, as it requires significantly less investment of time, money, and maintenance—but to really understand the differences between simulation and reality, it is also necessary to work with the real platforms from time to time. To our knowledge, our course may be the only one in the world where individual students are consistently afforded the opportunity to work with a networked multi-robot system on a tabletop. The e-Puck,<sup>1</sup> a low-cost small-scale mobile robotic platform designed for educational use, allows us bringing real robotic hardware into the classroom in numbers sufficient to demonstrate and teach swarm-robotic concepts. We present here a custom module for local radio communication as a stackable extension board for the e-Puck, enabling information exchange between robots and also with any other IEEE 802.15.4-compatible devices. Transmission power can be modified in software to yield effective communication ranges as small as fifteen centimeters. This intentionally small range allows us to demonstrate interesting collective behavior based on *local* information and control in a limited amount of physical space, where ordinary radios would typically result in a completely connected network. Here we show the use of this module facilitating a collective decision among a group of 10 robots.

## 1 Introduction

One of the aspects of swarm intelligence that makes it so exciting is that it involves an entirely different approach to problem solving than is intuitive to

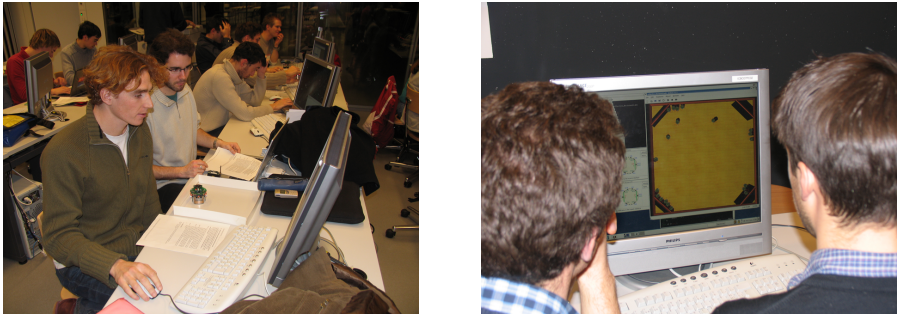
---

<sup>1</sup> <http://www.e-puck.org/>

most people [4]. Therefore, the fact that many students may not be accustomed to these different types of thinking and analysis should come as no surprise. The challenge to us is to provide them with the tools, materials, and guidance necessary to help them understand the principles of swarm intelligence and how they can be applied to an embedded real-time system such as a multi-robot swarm or a sensor network.

Our course, “Swarm Intelligence,”<sup>2</sup> includes weekly laboratory exercises (Figure 1) in which the students themselves use a combination of real robots and simulations to test and verify the topics and theories presented in lecture; this also allows them the opportunity to explore other possibilities that might not have been previously discussed. In doing so, the students are better able to assimilate theoretical concepts and understand the difficulties of implementing them. This also helps them understand the differences between various types of implementation levels; for example realistic simulation and real experiments.

Unfortunately, it seems that courses of this kind are rare; indeed, in the area of swarm intelligence, ours is the only one we know of to date. This is likely due in large part to the overhead of acquiring and maintaining enough hardware to provide individual students with a sufficient amount of direct contact with the equipment. Size is also a major concern specific to multi-robot systems, as it is necessary to simultaneously have several robots on a desk or tabletop. Simulation is occasionally used in courses as a substitute for real systems, but we find this to be a shame as well, since the two are not interchangeable, but rather complementary.



**Fig. 1.** Students using real and simulated e-Pucks during a laboratory exercise for the course “Swarm Intelligence” at EPFL, Fall 2005

## 2 Tools and Methods for Swarm Robotics

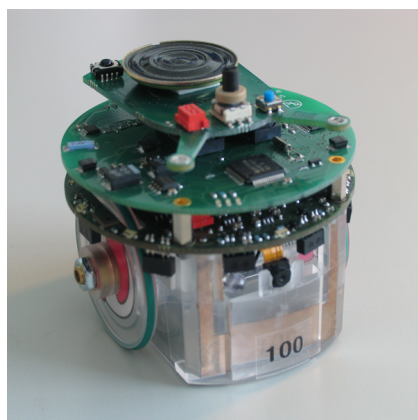
Certain tools can be extremely effective in helping one understand the principles behind swarm robotics; chief among them are naturally real robots and

<sup>2</sup> <http://swis.epfl.ch/teaching/>

simulations. However, we must respect the fact that they serve different, complementary purposes, and it is often precisely the interaction between them which can give us the greatest insight into the dynamics and subtle details of a system.

## 2.1 The e-Puck: An Educational Robot

A recent collaboration between the Autonomous Systems Laboratory (ASL),<sup>3</sup> the Swarm-Intelligent Systems group (SWIS),<sup>4</sup> and the Laboratory of Intelligent Systems (LIS)<sup>5</sup> at the École Polytechnique Fédérale de Lausanne (EPFL) has resulted in the creation of a new small-scale robotic platform for educational purposes. Central to the design of the core robot were Francesco Mondada and Michael Bonani (ASL), with some additional contributions to the base module from Xavier Raemy (SWIS), who also designed the radio communication board.



**Fig. 2.** The e-Puck: a small-scale robotic platform for education. Shown here with the radio communication board stacked between the basic module and the jumper board.

The e-Puck (Figure 2) was developed with five principle objectives in mind, for making it a high-quality teaching tool:

1. simple and sturdy electro-mechanical architecture
2. flexibility and variety in sensors, processing power, and extensions
3. minimum-hassle connectivity and usability
4. robustness sufficient to withstand use by students, and simple maintenance/repair procedures
5. sufficiently inexpensive that large numbers can be obtained so as to allow individual students direct contact with the equipment

<sup>3</sup> <http://asl.epfl.ch/>

<sup>4</sup> <http://swis.epfl.ch/>

<sup>5</sup> <http://lis.epfl.ch/>

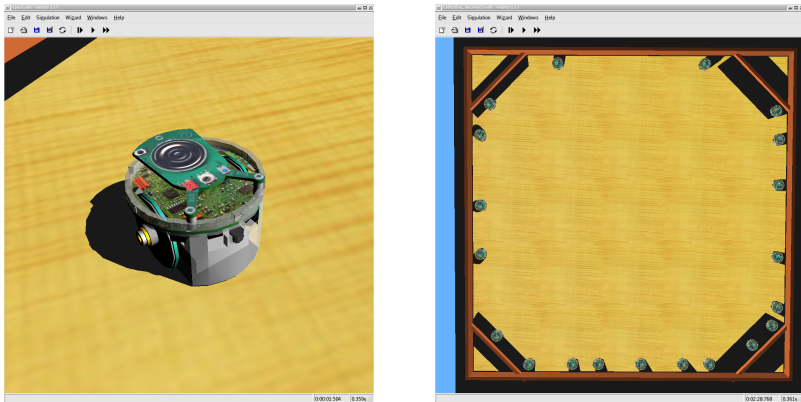
As a part of point 3, the e-Puck design includes several features which make it very well suited to multi-robot experimentation. There are no cables (programming or remote control with a computer is done via Bluetooth) and the battery is interchangeable (reducing downtime due to charging); these were the two principle drawbacks of the Khepera [17], the only previous robotic platform we know of with similar capabilities in a package this small, so as to allow the operation of at least three robots together on a portion of a desktop (the minimal number necessary for observing interesting collective effects).

Additionally, in the interest of education and knowledge sharing, the e-Puck is based on an “open source hardware”<sup>6</sup> model, whereby all documentation relating to it may be freely distributed under a license allowing anyone to use it and develop for it.

In December of 2005, 400 units were produced for use in various courses at EPFL and elsewhere, several of which were already underway during the academic year 2005-2006.

## 2.2 The Webots<sup>TM</sup> Simulation Environment

As mentioned above, simulation also has its place; it allows us to run experiments with many more robots (at a constant price, without having to buy and maintain hundreds of real robots) and greatly increases the speed and thoroughness with which theories can be tested. For much of our realistic simulation work, we run experiments in Webots, an embodied robotic simulation environment produced by Cyberbotics Ltd. [16].



**Fig. 3.** (left) A simulated model of the e-Puck in Webots, and (right) a simulation of 20 e-Pucks in a setup similar to the collective decision experiment described in Section 5

Through a recent collaboration with Cyberbotics, we have established a preliminary framework for the integration of the open source network simulation

<sup>6</sup> <http://www.e-puck.org/> → Project → License

engine OMNeT++ [20] into Webots as a modular plug-in to allow realistic modeling of radio communication channels between simulated robots (specifically utilizing a component we implemented containing the subset of the IEEE 802.15.4 and ZigBee protocols present in our physical modules for the robots, but the existing IEEE 802.11 components or any others written for use in OMNeT++ may be used as well). This work (and the related necessary verification) is on-going, and therefore is not yet ready to be presented in detail here.

### 2.3 Correspondence Between Reality and Simulation

One of the most important points that we try to teach our students is that simulation is a necessarily simplified representation of a system, but depending on the specifics of the system being considered, certain simplifications may be acceptable or even desirable. For example, if the simplified simulated model still produces results faithful to the behavior of the real system, one can say with reasonable confidence that the neglected parameters have little if any influence on the behavior being studied.

Used in this way, simulation then becomes a tool; one which can be extremely powerful when used properly in concert with real systems. Once we are confident that the simulation results accurately and precisely reproduce the outcomes of analogous experiments with real hardware, exploring the parameter space can be significantly easier and faster than performing similar experiments in reality. Finding this trade-off between realism and speed again requires careful consideration of the specific situation at hand to determine how much (or how little) realism is really necessary to achieve the desired results.

Taking this logic one step further, we can actually formalize the varying degrees of complexity possible; ranging from realistic simulation to mathematical macroscopic models [15]. Further details and examples of this multi-level approach can be found in [9,2].

## 3 Communication and Swarm Intelligence

Much of the previous work in swarm intelligence and swarm robotics has focused on so-called ‘biologically inspired’ mechanisms (some early definitions actually *limited* the definition of swarm intelligence to “algorithms or distributed problem-solving devices inspired by the collective behavior of social insect colonies and other animal societies.” [5]). Consistent with this definition, [5] goes on to define self-organization and stigmergy as key mechanisms required in a swarm-intelligent system.

Stigmergy, or indirect communication via the environment, works well for insects, which are particularly adept in the area of mobility, but it seems clear that it is not always the most ideal communication channel for sharing information (it is typically slow, short range, untargetted, etc). Nonetheless, it has been used with some success in various robotic tasks (for example, [1]).

Self-organization is generally accepted to consist of four principle components:

**Positive feedback:** amplification, notably recruitment and reinforcement.

**Negative feedback:** the checks and balances for positive feedback mechanisms, i.e. saturation, exhaustion, and competition.

**Randomness:** unpredictability can be crucial for the explorative element of a self-organized system; the robustness often exhibited is a direct result of the sometimes seemingly inefficient behavior caused by reactions to noise in the environment.

**Multiple interactions:** for self-organization to occur, there must be at least a minimum number of mutually tolerant agents able to react to the presence or actions of the others.

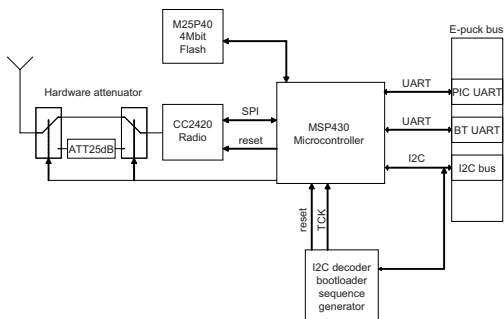
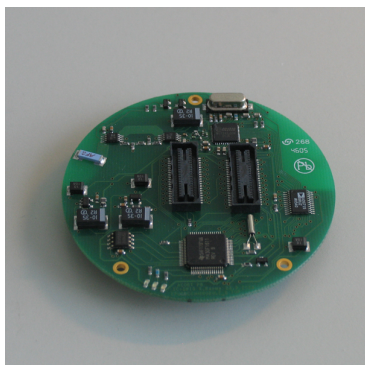
While the application of these principles to multi-agent systems is relatively straightforward (take ACO [10], for instance), when we want to apply swarm-intelligent principles to embedded platforms we need to understand the differences between natural and artificial systems, and subsequently exploit the strengths that may be present in an artificial system, to minimize the impact of accompanying weaknesses. This represents a fundamental shift ‘beyond biomimicry’ [14], and one of the most obvious areas where this may be leveraged is with respect to communication. Though the caveat clearly remains that however we choose to utilize the radio channel (or any other addition to a swarm system), we must ensure that it does not affect the scalability of the system.

Direct, in this case radio, communication is simply a more sophisticated medium for achieving “multiple interactions.” While one could argue that the inherent unreliability of communication vectors in natural systems is a large part of what forces the system to exhibit the robustness that swarm algorithms are renowned for, we believe that even using radio communication (which is not always reliable either), there remains sufficient noise in coordination and other parts of the system to provoke a collective response showing the appropriate balance of explorative and exploitative behavior for mitigating environmental unpredictability.

Without interference (be it physical or communication), the effects shown in [18] would not be possible; there would be no semblance of intelligent behavior emerging from the system. Despite the use of a decidedly non-natural element, such as radio communication, the swarm-intelligent nature of the resultant collective behavior is still utterly dependent on environmental uncertainty, noise, and self-organized coordination based on local interactions.

## 4 A Radio Communication Module for the e-Puck

To turn the e-Puck into a networked robotic system suitable for running experiments requiring local communication, we constructed a radio board (as shown in Figure 4.a) with the requirements that it be low power, as the e-Pucks run on batteries, and that it operate on standardized protocols, so as to be interoperable with our other existing robotic and sensor network platforms running TinyOS [12].



**Fig. 4.** (left) A stackable module for the e-Puck enabling local radio communication, and (right) a block diagram showing the principal components and functions of the same

#### 4.1 Hardware Design and Structure of the Module

Figure 4.b includes a block diagram illustrating the basic structure of the radio board, which is based on a modified version of the Telos (rev. B) [19] schematics provided by MoteIV. The processor is a Texas Instruments MSP430F169 with 2kB of SRAM and 60kB of flash memory (program storage), selected for its attractive energy consumption profile and the existence of a functional TinyOS port to its architecture. The physical radio is a Chipcon CC2420, an IEEE 802.15.4 and ZigBee compliant transceiver, which allows us to take advantage of the partial implementation of the IEEE 802.15.4 and ZigBee extensions already present in TinyOS. This makes hybrid communication between this radio module and any of our other platforms trivial (we have previously constructed a similar module for the Alice [8,6], and also use a sensor network composed of MICAz [11] nodes). A software selectable custom attenuation circuit is added between the transceiver and the SMD antenna (Antenna Factor ANT-2.45-CHP), for range reduction (note that this affects both reception and transmission).

#### 4.2 Software Control of the Radio Board

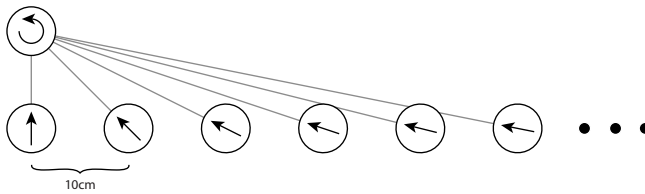
A firmware controller based on TinyOS was prepared to allow the module to act in accordance with high-level commands issued to it by the e-Puck via the I<sup>2</sup>C bus. Appropriate primitives were then written and integrated into the e-Puck API for the sending and receiving of messages, as well as the modification of control parameters (such as the transmission power, etc).

Existing modules were used wherever possible (i.e. **GenericBase**), and all of the necessary parameters were encapsulated to allow runtime modification from the e-Puck. However, as the implementation of the I<sup>2</sup>C protocol [13] within the current distributed version of TinyOS only supports operating as a bus master, the slave layer had to be written and integrated so that the module could be properly

accessed by the e-Puck. At present, since the radio functions as a slave, polling is necessary for message reception, but in the future, if a full implementation of the multi-master mode (as provided for in the I<sup>2</sup>C specification) can be integrated into TinyOS, interrupts will be able to pass in both directions, easing the computational burden on the e-Puck and making control simpler and more intuitive.

### 4.3 Measurement of Physical Characteristics

A number of tests have been run for ascertaining the performance and limits of the device. Preliminary measurements of power consumption indicate that when not in use, the module draws less than 1.4mW, and with the radio on (ready to receive) and the processor under heavy load, approximately 76.2mW.

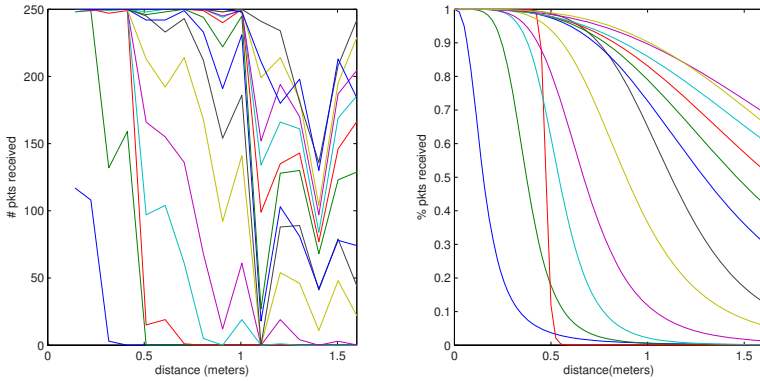


**Fig. 5.** Physical arrangement of robots for the range tests. Sixteen receivers are placed in a line at known distances from a transmitter, which is rotating to average out irregularities based on orientation.

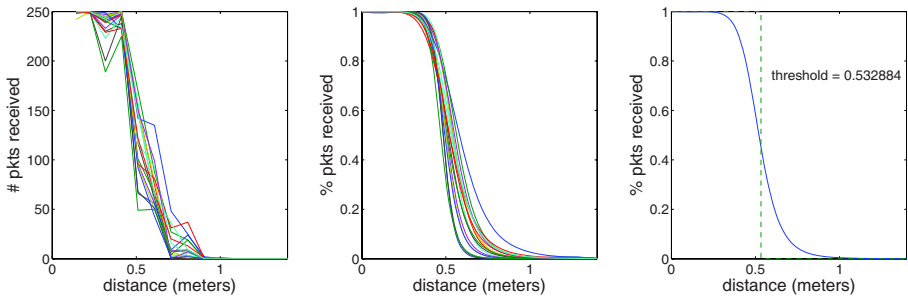
The output transmission power of the CC2420 is specified by an integer register value between 3 and 31, minimum and maximum, respectively (these numbers are an artifact of the radio hardware; see the CC2420 datasheet [7] for more information). For measuring the effective transmission range at various power settings, 17 robots (1 emitter and 16 receivers each oriented towards the emitter) were arranged as shown in Figure 5. During each iteration, the emitter would spin in place (so as to average out any possible anomalous effects of orientation) while transmitting 250 packets. Each of the receivers would then count the number of packets received, yielding a reception rate at each receiver location (Figure 6.a), the collection of which was then fed into a sigmoidal regression to determine an approximate probability density function of distance (Figure 6.b). Fifteen such iterations were performed per experiment, one for each of the odd numbered transmission power settings in the set of allowed values (3–31). This experiment was repeated three times; with the hardware attenuator active on the sender, the receivers, and on both the sender and the receivers. Only the results from the symmetric case (both the sender and the receiver using the -25dB attenuator) are shown here, as it would not be possible to implement the case study presented in Section 5 with an asymmetric attenuator configuration.

Next, based on these results, three representative transmission power settings were selected (3, 7, and 31), and a more detailed test was performed, the results of which are shown for the value 7 in Figure 7 (the results of the remaining two





**Fig. 6.** (left) Raw received packet counts at each location for 15 different transmission powers, and (right) sigmoidal regression representing an approximate PDF. The apparent variance in sensitivity between nodes prompted the subsequent more specific tests, as illustrated in Figure 7.



**Fig. 7.** (left) Raw received packet counts at each location at transmission power 7 for 16 different receiver orderings, (center) sigmoidal regressions for each of the 16 iterations, and (right) aggregate regression on all 256 data points, with corresponding Heaviside approximation (equal area under curve)

experiments are extremely similar in form to those shown, and therefore will be quoted numerically only).

Sixteen iterations were run using the same transmission power, but with each receiver eventually occupying every possible receiver location, to remove anomalous contributions from manufacturing heterogeneities in the hardware (which, as can be seen in Figure 7.a, are present, but do not shift the basic shape of the curve). The corresponding regressions were calculated (Figure 7.b), and all 256 data points were used to create a master regression, shown in Figure 7.c with its Heaviside approximation around the definite integral (from 0 to  $\infty$ ), which can be used in geometric modeling as an estimated radius of communication.

For the three values tested in this manner, the associated approximate radii are 0.150987, 0.532884, and 4.84131 meters, respectively.

## 5 Case Study: Collective Decision

One of the most basic examples of swarm intelligence is the emergence of a consensus, or collective choice, in a distributed system. This fundamental question has recently been highlighted in the context of the European project LEURRE,<sup>7</sup> among others [3].

### 5.1 Experimental Setup

For testing our module in an experimental application, we have set up a simple environment in which independent robots, using only the local information available to them, interact in such a way as to exhibit convergence to a self-organized collective decision.

In a round arena approximately 1 meter in diameter, the robots each initially select at random to execute either left or right wall following, and periodically announce their current preference over the radio. Upon reception of one or more such messages, if the perceived majority opinion is not the same as the robot's current opinion, it makes a probabilistic decision to possibly change its current direction. This behavior will eventually cause the system to converge to a state where all the robots are traveling in the same direction (Figure 8.a). And as one might naturally imagine, given that the decisions to switch directions are based on partial perception, the time required for reaching convergence will depend on the accuracy with which the local perception reflects the global state of the system. Here, that translates directly to the range of communication.

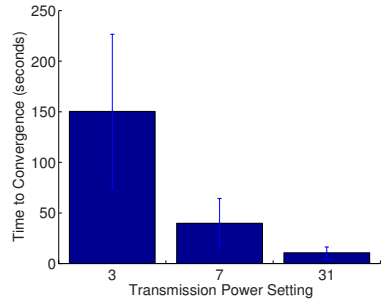
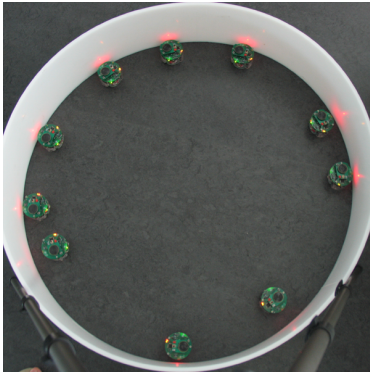
Note that this setup displays all the habitual signatures of a self-organized system: multi-stability (system may converge to either left or right wall following), positive feedback (the number of neighbors influences the probability of being convinced to change direction), negative feedback (there are a limited number of robots; resource exhaustion), randomness (non-deterministic decisions, formation of local subgroups due to partial perception), and multiple interactions (radio messages and physical detection/avoidance/following).

### 5.2 Results

Sixteen experiments were run for each of three transmission powers: 3, 7, and 31 (48 runs in total). A nearby MICAz node acted as an eavesdropper (with its antenna, it was able to reliably overhear even the minimum power messages from the robots, so long as it was near the arena), and counted the time between the start signal and when all received messages from the robots indicated that they were traveling in the same direction. The mean and standard deviation of these completion times are shown in Figure 8. At first glance, the deviation may seem a little large, but is likely due to the random nature of the initial conditions and the interactions between the agents.

---

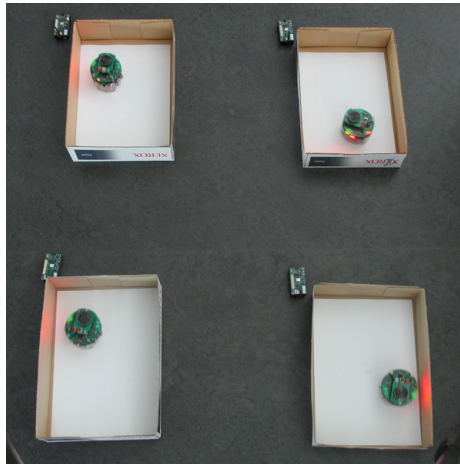
<sup>7</sup> <http://leurre.ulb.ac.be/>



**Fig. 8.** (left) Overhead view of ten e-Puck robots after convergence to left wall-following, and (right) mean and standard deviation of convergence times for 48 experimental runs (16 each power)

### 5.3 Related Hybrid Network Example: Isolated Collective Decision

A similar setup which we presented to the students in a course laboratory exercise<sup>8</sup> involved a hybrid network, as pictured in Figure 9; each robot was isolated in its own miniature arena (still executing left or right wall following), but in one corner of its arena, it would be close enough to a fixed node in a sensor network



**Fig. 9.** Alternative setup to Figure 8: a hybrid network wherein isolated robots communicate via a sensor network backchannel

<sup>8</sup> [http://swis.epfl.ch/teaching/swarm\\_intelligence/ay\\_2005-06/exercises/SI\\_05-06\\_labhwk10\\_assignment.pdf](http://swis.epfl.ch/teaching/swarm_intelligence/ay_2005-06/exercises/SI_05-06_labhwk10_assignment.pdf)

which could act as a relay tower. Therefore, during a short section of its trip around the track, it had the opportunity to send and receive messages with other robots (which were also be close enough to their respective sensor nodes) via the ‘backbone network’ provided by the sensor nodes. While systematic testing is not shown here, this system also yielded convergence to a collective decision, in a network of 15 robot/node pairs.

## 6 Conclusion

Teaching and research activities in swarm intelligence and swarm robotics require tools; among these, we have found that simulation and physical hardware are both beneficial and mutually complementary in an educational setting. For reasons of accessibility (equally from the perspectives of cost, pedagogy, and usability), the e-Puck platform, particularly when equipped with our local communication module (range adjustable between about 15cm and 4.8m), promises to serve as a powerful addition to the toolset in this context. In the example scenario demonstrated here, collective decision occurs in groups of up to fifteen robots.

## Acknowledgments

The authors are currently sponsored by a grant from the Swiss National Science Foundation (Contract Nr. PP002-68647). Additionally, portions of the works described received partial funding from the Fond d’Innovation pour la Formation (FIFO) and the School of Computer and Communication Sciences, both at EPFL. We would like to thank especially Francesco Mondada and Michael Bonani, ultimately responsible for the design and creation of the e-Puck, and also Olivier Michel, Yvan Bourquin, and Alexei Kounine, without whose assistance the modification and integration of the realistic network simulation engine OMNeT++ into Webots<sup>TM</sup> would not have been possible. Inspiration for the experimental case study was initially seeded during an informal discussion between Alcherio Martinoli and Guy Theraulaz.

## References

1. W. Agassounon and A. Martinoli. Efficiency and robustness of threshold-based distributed allocation algorithms in multi-agent systems. In *Proc. of the Int. Conf. on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 1090–1097, Bologna, Italy, 2002. ACM Press.
2. W. Agassounon, A. Martinoli, and K. Easton. Macroscopic modeling of aggregation experiments using embodied agents in teams of constant and time-varying sizes. *Autonomous Robots*, 17(2–3):163–191, 2004.
3. J. M. Amé, J. Millor, J. Halloy, and J.-L. Deneubourg. Collective decision-making based on individual discrimination capability in pre-social insects. In *Proc. Simulation of Artificial Behaviour*, pages 700–711. Springer Verlag, Berlin, 2006.

4. G. Beni. From swarm intelligence to swarm robotics. In *Proc. of the SAB 2004 Workshop on Swarm Robotics, Santa Monica, CA, USA, July, 2004*, volume 3342, pages 1–9, 2005.
5. E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. SFI Studies in the Science of Complexity, Oxford University Press, New York, NY, USA, 1999.
6. G. Caprari and R. Siegwart. Mobile micro-robots ready to use: Alice. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 3295–3300, Canada, 2005.
7. CC2420: 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver, Chipcon Products from Texas Instruments. <http://www.chipcon.com/>, 2005.
8. N. Correll and A. Martinoli. Comparing coordination schemes for miniature robotic swarms: A case study in boundary coverage of regular structures. In *Proc. of the Int. Symp. on Experimental Robotics (ISER)*, Rio de Janeiro, Brazil, 2006. Springer Tracts for Advanced Robotics (STAR), to appear.
9. N. Correll and A. Martinoli. Modeling and optimization of a swarm-intelligent inspection system. In *Proc. of the Int. Symp. on Distributed Autonomous Robotic Systems (DARS)*, pages 369–378. Springer Distributed Autonomous Systems VI, 2006.
10. M. Dorigo and G. D. Caro. The ant colony optimization meta-heuristic. *New Ideas in Optimization*, pages 11–32, 1999.
11. J. Hill and D. Culler. Mica: A wireless platform for deeply embedded networks. *IEEE Micro*, 22(6):12–24, 2002.
12. J. Hill, R. Szcwzyk, A. Woo, S. Hollar, D. Culler, and K. Pister. System architecture directions for network sensors. In *Architectural Support for Programming Languages and Operating Systems (ASPLoS)*, 2000.
13. The I2C-bus specification, version 2.1, Philips Electronics N.V. [http://www.semiconductors.philips.com/products/interface\\_control/i2c/index.html](http://www.semiconductors.philips.com/products/interface_control/i2c/index.html), 2000.
14. IEEE Swarm Intelligence Symposium, Call for papers. <http://www.ieeeswarm.org/>, 2005.
15. A. Martinoli, K. Easton, and W. Agassounon. Modeling of swarm robotic systems: A case study in collaborative distributed manipulation. *Int. Journal of Robotics Research*, 23(4):415–436, 2004.
16. O. Michel. Webots: Professional mobile robot simulation. *Journal of Advanced Robotic Systems*, 1(1):39–42, 2004.
17. F. Mondada, E. Franzini, and P. Ienne. Mobile robot miniaturization: a tool for investigation in control algorithms. In *Proc. of the Int. Symp. on Experimental Robotics (ISER)*, pages 501–513. Springer Verlag, Berlin, 1993.
18. J. Nembrini, A. Winfield, and C. Melhuish. Minimalist coherent swarming of wireless connected autonomous mobile robots. In *Proc. Simulation of Artificial Behaviour (SAB)*, pages 273–382, Edinburgh, 2002.
19. J. Polastre, R. Szcwzyk, and D. Culler. Telos: Enabling ultra-low power wireless research. In *IEEE/ACM Int. Conf. on Information Processing in Sensor Networks (IPSN-SPOTS)*, 2005.
20. A. Varga. Software tools for networking: “OMNeT++”. *IEEE Network Interactive*, 16(4), 2002.