# DESIGN, CONTROL, AND APPLICATIONS OF AUTONOMOUS MOBILE ROBOTS

D. FLOREANO, J. GODJEVAC, A. MARTINOLI, F. MONDADA AND J-D. NICOUD
*Micro-computing Laboratory*
*Swiss Federal Institute of Technology in Lausanne*
*LAMI-INF-EPFL, CH-1015 Ecublens*

## 1. Introduction

An autonomous robot is a machine that operates in a partially unknown and unpredictable environment. In contrast to robots used in manufacturing plants, where the environment is highly controlled, autonomous robots cannot always be programmed to execute predefined actions because one does not know in advance what will be the universe of required sensorimotor transformations required by the various situations that the robot might encounter. Furthermore, the environment might have dynamic characteristics that require rapid online modifications in the robot behaviour. For these reasons, in the last ten years several researchers have looked at novel methods for setting up autonomous mobile robots.

The basic idea behind most approaches is to break down sequential top-down programs into a set of simple, distributed, and decentralised processes that have direct access to sensors and motors of the robot. The first formalisation of this approach is the subsumption architecture proposed by Rodney Brooks at MIT [2, 3] where several local behaviours (sensorimotor modules) continuously operate in parallel using only local available information. The emergent overall behaviour is flexible, robust against environmental noise and mechanical failure, and based on compact modular codes. Another key feature of these novel approaches is bio-inspiration, that is the attempt to implement mechanisms of biological adaptive behaviour [27, 26, 4, 21]. Adaptation, combined with a decentralised bottom-up approach, is often seen as a solution to the problem of generating and maintaining stable behaviours in partially unknown and dynamic environments.

In this chapter we give an overview of recent work in autonomous mobile robotics done at our laboratory, covering hardware methods and design, adaptive control, collective autonomous robotics, and conclude with considerations on industrial applications. Other related and complementary approaches can be found in two special issues of the journals *Robotics and Autonomous System* [14] and *IEEE Transactions on Systems, Man, and Cybernetics - B* [5]. In the next section we shall introduce some important methodological issues in setting up a research tool for autonomous robotics, such as miniaturisation, modularity, and flexibility which are behind a mobile robot developed at our laboratory and currently used by several hundreds research centres. We shall then describe two experiments using neuro-fuzzy adaptive control, a methodology which provides an easy interface between user knowledge and robot autonomous operation. Some of these experiments will be later used as a starting point for introducing an evolutionary approach where genetic algorithms are used to automatically develop neurocontrollers without human intervention in a range of different environments. In a later section, we shall address hardware and control issues in autonomous collective robotics. Here, global behaviour of a team of robots emerges from dynamical interactions among several robots programmed to operate using only local information. Finally, we provide some practical considerations on industrial applications of autonomous mobile robots and conclude with a comparative summary and outlook for future developments.

## 2. Issues in hardware design

Simulation studies have for long time been considered a valid investigation methodology to develop autonomous mobile robots, both for research purposes and industrial applications. Recently, a large number of researchers [2, 3, 6, 12, 36, 38], including the authors of this chapter, have stressed the importance of using real robots for the development and validation of novel solutions to fully capture important interactions between the robot and the environment. Despite a widespread agreement, real mobile robots are not yet always employed during the initial research stage. We believe that there are some practical reasons behind this fact. Most of the researchers working in the field of artificial intelligence are software engineers or academics coming from various disciplines like biology, psychology, or anatomy. These people are not very keen to deal with mechanical and electronical problems. Mobile robots were often very unstable devices, built by hobbyists, and needing specific know-how and special care to operate them. Only few laboratories in the world have the possibility to build a robot and program it to achieve interesting research results. Furthermore, most of the commercial products available were made by roboticians for roboticians, resorted to very specific programming languages and thus were not accessible to outsiders. This situation is improving, but a majority of robots still needs specific know-how. Finally, in universities, where software writing is considered "without costs", a real robot seems often too expensive in comparison to simulations.

Lack of simple, efficient, robust and low cost robotic tools prompted us to develop the Khepera miniature mobile robot, initially designed in 1991 by E. Franzi, A. Guignard, and F. Mondada [30], based on ideas of J-D. Nicoud. Given the interest of our laboratory in innovative and multidisciplinary approaches (some of which will be outlined in the following sections), the Khepera robot has been intended right at the beginning as a tool that could be easily used by a variety of people with different backgrounds and needs, such as biologists willing to test theories of adaptation, neurophysiologists developing models of neural networks, psychologists implementing models of learning and cognition, and engineers evaluating different control strategies for specific applications. In the following subsections, we describe the essential design elements of this robotic tool.

### 2.1. MINIATURISATION

Miniaturisation of a mobile robot brings some advantages to the researcher who works with it, but it is a presents several challenges for the engineer who has to develop it. Khepera represents a viable compromise, featuring 55 mm in diameter and a variable height; it is sufficiently small to easily move on a desk and sufficiently big to be built with standard electronic and mechanical components. This size brings several important advantages. **I.** The experimenter can build complex environments on a limited surface. For a miniature robot like Khepera, indeed, a normal office desk of 0.9m x 1.8 m represents a working surface equivalent to that of a tennis court for a standard-size robot having a diameter of 55 cm. Additionally, one can use a thin suspended cable for power supply without disturbing the robot movements and also place a camera with normal lenses on the ceiling for monitoring the whole environment. **II.** Fundamental laws of physics give higher mechanical robustness to a robot of this size. In order to intuitively understand this physical phenomenon, compare a robot of 50 mm in diameter crashing against a wall at 50 mm/s with a robot of 1 m in diameter crashing against the same wall at 1 m/s. The miniature robot will resist the collision, the other robot will probably report serious damages. **III.** The price of a miniature robot like Khepera is lower than that of a robot with comparable performance, but larger size. This is due to smaller parts, smaller circuits, smaller infrastructure to build and test the robots, and simpler mechanics which allow us to use electrical part (connectors, for instance) for both electronic and mechanical functionalities (body structure, for instance). Miniaturisation brings also some drawbacks, like difficulty of mounting large devices on the robot (ultrasonic sensors, laser range finders, etc.).

### 2.2. HARDWARE AND SOFTWARE MODULARITY

Hardware modularity enables different possible configurations and experiments using the same basic components. It means also possible extensions and, globally, cheaper equipment. Software modularity means flexibility and possibilities for extensions, which enables the software developer to write only parts of the program required for the specific application. Khepera is based on this concept of modularity, both in hardware and software. At the hardware level, Khepera has an extension bus that makes it possible to add turrets on the top of the basic configuration, depending on the needs of the experiments to be done. This modularity is based on a parallel and a serial bus. The parallel bus can be used for simple extensions directly under control of the main Khepera processor. The serial bus implements a local network for inter-processor communication. Using this second bus, other processors can be connected to the main one in order to build a multiprocessor structure centred on the Khepera main processor. This kind of structure has the advantage that one can employ additional computational devices on extension modules, thus keeping the main processor free for global management of the robot behaviour.

At the software level, modularity is needed to support the multi-processor structure of the robot. It consists of a flexible protocol that recognises all added extension modules when the robot is powered, informing the main processor about all functionalities available in each extension as well as the procedures for activating these functionalities. The BIOS of the Khepera, which includes all basic procedures for robot management, is also based on a modular structure. Motors, sensors, and timing functions are grouped into distinct modules to simplify management of the robot and improve software robustness. The main software also supports remote control and down-loading of specific applications through a serial cable. Such software structure simplifies the task of the user, who can easily add her own software to the management modules already implemented on Khepera.

### 2.3. FROM SIMULATIONS TO APPLICATIONS

Several researchers in the field of autonomous mobile robots belong to computer science, artificial intelligence, neuroanatomy, biology, psychology, and many other related domains. The tool needed by this community to move from simulation models to the real world is a simple robot that can be easily

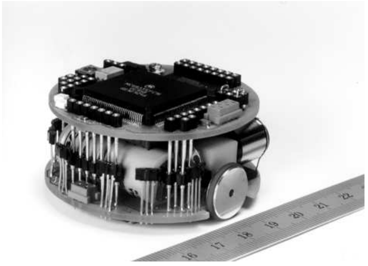


*Figure 1.* The miniature mobile robot Khepera in its basic configuration.

controlled from the desk computer in the same way in which a simulated robot is controlled.

Khepera is situated between simulations and real-world applications. If on the one hand it keeps a level of simplicity and operation modality similar to simulation tools, on the other hand it cannot reach the complexity of real-world applications. However, being a physical robot, it introduces most of the characteristics of robots used for real-world applications. This position between simulators and applications is confirmed by several Khepera users who have moved from simulations to the miniature robot and have highlighted the advantages of playing at these two levels [33, 19, 28]. Recent construction of the new larger Koala robot, which is software compatible with Khepera, enables the transfer of developments made on the Khepera to a more complex platform which can be used for real-world applications.

### 2.4. KHEPERA CHARACTERISTICS AND EXTENSIONS

The Khepera robot consists of a basic platform and a number of extension turrets. Some of them are described below and are used in the experiments described in the following sections, while other modules are under development.

In its basic configuration (figs. 1 and 2), Khepera consists of two layers corresponding to two main boards: the sensory-motor board and the CPU board. The motor system consists of two lateral wheels and two pivots on the front and back. This configuration is very good for facing complex geometric obstacles because the robot can turn in place without lateral displacement. The sensory system available in the basic configuration is placed on the lower board, consisting of 8 infrared-light proximity sensors distributed around the body, 6 on one side and two on the other (this asymmetry can be used to establish the front and back of the robot). These sensors can detect the presence of objects by emitting and measuring re-

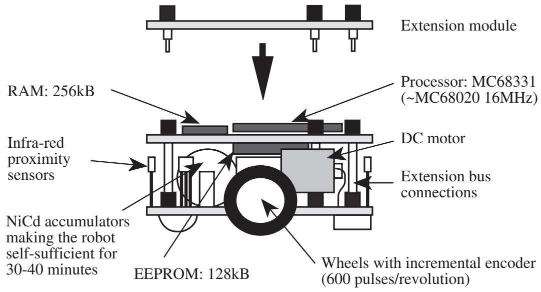
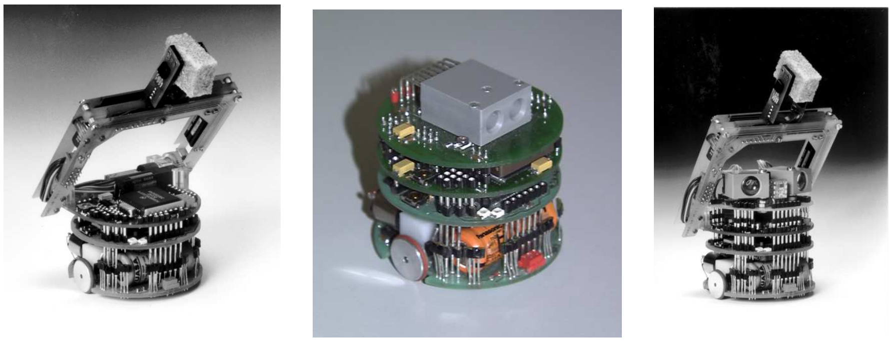



*Figure 2.* Khepera robot structure and extension possibilities.

flected light and can also be used as simple passive infrared light sensors. On the sensorimotor board are also placed NiCd batteries with a capacity of 110 mAh which allow the robot to be self-sufficient for approximately 30-40 minutes. The CPU board encloses the robot's main processor (a Motorola MC68331 with 128 K-bytes of EEPROM and 256 K-bytes of static RAM). An A/D converter allows the acquisition of analog signals coming from the sensory-motor board. An RS232 serial line is also available on the board via a miniature connector. On this same connection, a wire can also provide continuous power supply from an external source.

The electrical link between sensory-motor board and CPU board is done by connectors that also provide mechanical support. These connectors pass through the boards and form an extension bus for addition of further modules to the robot. Possible extensions are, for example, the gripper module and the linear vision module illustrated in figure3. The gripper module (figure 3a) is connected on the extension bus and has its own processor which communicates with the main Khepera processor using the local network. The gripper has two degrees of freedom: elevation and grasping. Sensors inform about elevation of the arm, position of the gripper fingers, presence of an object inside the gripper (optical barrier), and electrical resistivity of the grasped object. The K213 linear vision, which can be placed on the basic platform (figure 3, b)) or on the top of the gripper (figure 3, c)), has its own local processor which uses the local network to share information with the main processor. This module has a horizontal linear camera composed of 64 photo-receptors giving a gray-level image spanning a 36-degree visual field, and an auto-iris function for automatic adaptation of image sensitivity to changing light conditions which also provides information on global light intensity.



a) b) c)

*Figure 3.* From left to right: Khepera with a gripper extension, with a "linear vision" extension, and with both modules simultaneously.

### 2.5. IMPROVEMENTS OF THE ACTUAL DESIGN METHODOLOGIES

Widespread use of the Khepera robot (more than 500 users spread in more that 200 research centres and schools all over the world) and publication of the scientific results have clearly improved replicability and comparison of research results. However, despite this progress, it is still necessary to move in the direction of real applications, which means larger robots and more complex environments. This kind of experimentation needs considerable investments of time and money, and it cannot maintain the standardisation achieved with Khepera. If a simple Khepera environment can be easily reproduced, a larger and much more complex office environment will never be reproduced by other research groups to compare the efficiency of the control algorithms. Therefore, new tools are required.

A project currently in progress at LAMI[1] aims at testing a possible solution, based on the growing communication networks, to this methodological problem. The basic idea is to make available to the scientific community one or more complex mobile robotic set-ups through computer networks like Internet or ATM. With this equipment, every university or research group in the world could access in real time, through a network, the same robotic set-up, download algorithms into the robot, monitor the robot behaviour, or modify the environment, just as if the robot was in the next room. Comparison between different approaches in the field of autonomous mobile robotics can be made only when all conditions are perfectly identical. This means that not only the robot has to be exactly the same, but also

[1] This project, called "Sharing of Unique or Expensive Equipment for Research and Education: The Remote Manipulation Paradigm" is part of the Swiss Priority Programme SPP ICS of the Swiss National Foundation.

that the environment has to be identical, in all details from light conditions to colour and type of floor. A unique and shared robotic set-up meets these requirements and could bring a new dimension to scientific research.

In the following two sections, we will shift our attention to different types of adaptive control applied to the Khepera robot.

## 3. Fuzzy control

Fuzzy logic offers the possibility to express and implement human know-how in the form of linguistic if-then rules which can be applied for the control of nonlinear systems, such as mobile robots [39]. Every rule has two parts: the antecedent part (premise), expressed by **If...** , and the consequent part, expressed by: **then...** . The general form of a linguistic if-then rule is:

> **If** a set of conditions is satisfied **then** a set of consequences can be inferred.

The antecedent part is the description of the state of the system which is used to activate one rule, while the consequent part is the action that the operator who controls the system must take. The process states and control variables are called linguistic variables. A linguistic variable can take several linguistic values. For example, figure 4a shows a set of seven values for the linguistic variable *speed*, labelled as: *NEGATIVE BIG*, *NEGATIVE MEDIUM*, etc. These linguistic values are expressed by continuous functions, called membership functions which represent fuzzy sets. Each membership functions is defined by two parameters: its centre and width.

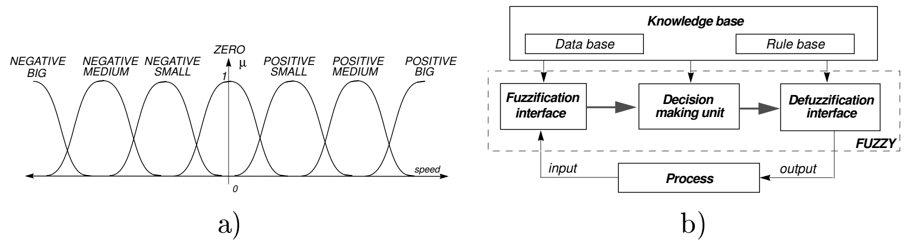



*Figure 4.* a) Universe of discourse for linguistic variable *speed*. b) Block diagram of a fuzzy controller.

A fuzzy controller is composed of four principal modules, as shown in figure 4b. The *fuzzification interface* performs the transformation of crisp values into fuzzy sets. The *knowledge base* supplies the fuzzification module, the inference engine, and the defuzzification interface with necessary information (parameters of membership functions and rules) for their proper functioning. The *decision making unit*, or inference engine, computes the

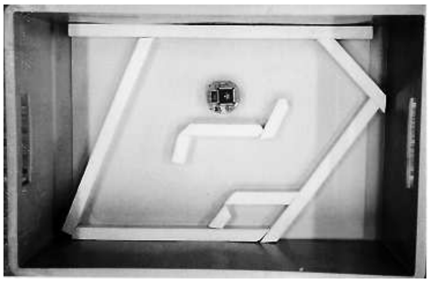


*Figure 5.* Circular maze employed for navigation experiments described in sections3.1 and 4.1.

meaning of the set of linguistic rules. The *defuzzification interface* transforms the union of fuzzy sets (individual contributions of each rule in the rule base) into a crisp output.

Although one can implement a simple controller for obstacle avoidance on the Khepera with few rules, the main effort is that of designing the appropriate membership functions and choosing the rules.

### 3.1. NEURO-FUZZY CONTROL FOR OBSTACLE AVOIDANCE

We have applied an automatic method for the design and the analysis of a fuzzy controller for two different behaviours: obstacle avoidance and wall-following. Since this method is based on a supervised learning procedure developed in the frame of neural networks, we will call it a *neuro-fuzzy* controller.

The first step in the design of a neuro-fuzzy controller is to establish a set of initial rules and parameters of the membership functions. The second step is the implementation of the controller. During the system operation, a supervisor guides the robot through the environment while the parameters of the membership functions and of the rules are adjusted by a learning algorithm based on stochastic approximation method [15]. After learning, the parameters define a new set of rules which can be used for autonomous operation of the robot. Additionally, the learned rules can be extracted in a form intelligible for a human being.

The training environment used for the two experiments consists of a circular maze with several sharp corners (figure 5). The results of the experiments are shown in figure 6. The chosen neuro-fuzzy controller has 4 linguistic variables as inputs: *distance from the left* $D_l$, *from the front* $D_f$, *from the right* $D_r$, and *from the back* $D_b$ of the robot. These are calculated

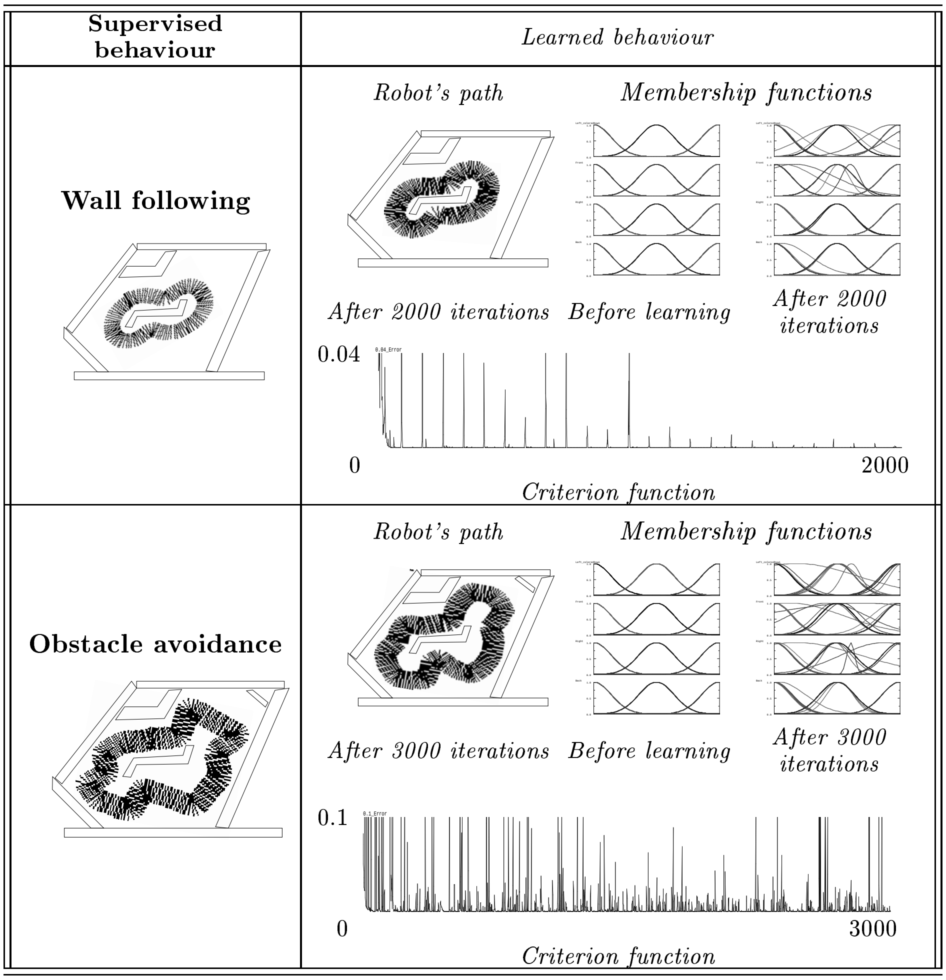



*Figure 6.* Results of adaptive neuro-fuzzy experiments.

in the following way:

$$D_l = S_0,\ D_f = \frac{S_2 + S_3}{2},\ D_r = S_5,\ D_b = \frac{S_6 + S_7}{2}, \tag{1}$$

where $S_i$ are the sensor values normalised within the interval [0,1]. For each input variable, we defined three linguistic values in order to classify the distances as: *Big*, *Average*, and *Small*. These values are sufficient to capture the various states of the sensors for this environment. The parameters of the membership functions are initialised so that they uniformly cover the input space. The controller also has two output linguistic variables: *left motor speed* and *right motor speed*. They can take seven linguistic values.

The first column of figure 6 plots a trace of the robot's path during the learning phase while the robot is guided to avoid obstacles or follow the central wall of the environment shown in Figure 5. The Khepera's path is displayed as a series of segments that link the contact points of the wheels on the floor. The motion is computed by odometry using the internal data from the incremental sensors of the wheels. Since the sampling period is constant, closer lines mean slower robot motion. The second column displays the robot paths after learning without human control, the membership functions before and after the learning phase, and error values during learning.

In the wall-following experiment depicted in the first row, learning is successfully accomplished after 1000 iterations (sensorimotor loops), whereas in the obstacle avoidance experiment learning can be considered satisfactory after 1500 iterations. These experiments show that the learning method can be successfully applied to a real robot with noisy and imprecise sensors. The robot learns only the real state vectors corresponding to those situations which are encountered in its real environment. An interesting feature of adaptive neuro-fuzzy control is that learning can start from a set of rules that the engineer might think are important (instead of starting from random parameters) and modify them as required by the environment. Seldom encountered situations need long time to be learned, as shown by the error peaks for the obstacle avoidance experiment after 1500 iterations. This drawback is the price that we pay for using on-line learning instead of memorising learning vectors which would take large memory and computational resources.

The modified membership functions can be used for off-line extraction of linguistic rules for analysis purposes. The possibility to obtain a linguistic representation of the transfer characteristics is an interesting properties that several other learning methods –such as artificial neural networks- do not offer. When extracting the rules associated with the learned parameters, one not only learns about the fuzzy controller, but also about the sensorimotor transformations required by the robot for a specific task.

## 4. Evolutionary Robotics

Evolutionary Robotics is a technique for automatic creation of control systems for autonomous robots that is inspired upon the Darwinian principle of selective reproduction of the fittest individuals. Compared to the neuro-fuzzy system described above, an evolutionary approach does not requires less human knowledge.

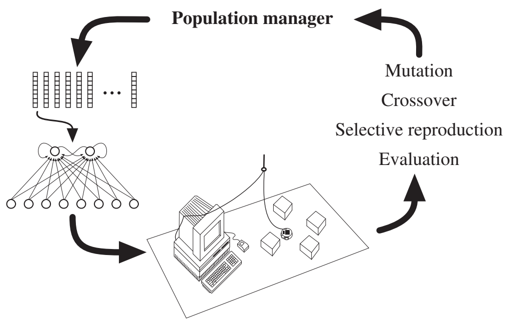



*Figure 7.* Evolutionary robotics on a single robot. Each individual is in turn decoded into a corresponding neurocontroller and the robot is let free to move while its fitness is automatically evaluated.

A population of different artificial chromosomes, each encoding the instructions to build a different neurocontroller, is decoded and tested on the robots. While each robot freely interacts with the environment according to the decoded control system, a "fitness function" automatically assesses its performance. New populations of increasingly better individuals are created by repetitively applying selective reproduction, crossover, and mutation to the evaluated chromosomes for several generations [18] (figure 7).

The role of the engineer in evolutionary experiments is to design a fitness function that measures the performance of the robots with respect to the desired behaviour or task. Artificial evolution automatically develops suitable control systems while the robot autonomously interacts with the specific environment where it is situated incorporating the physics and dynamics of the environment which often are not available to the engineer. In the next subsections we shall describe some results in evolutionary robotics that show how different environmental constraints automatically shape significantly different behaviours without requiring major changes in the fitness function of an autonomous robot trying to keep itself in a viable state.

### 4.1. A SIMPLE EXPERIMENT

As a simple comparative example, let us consider evolution of straight navigation and obstacle avoidance for the Khepera robot in the same environment (figure 5) already used for the experiments with fuzzy logic described in section 3.1. The fitness function $\Phi$ to be maximised is based on three variables which are measured at each time step using sensor and motor activations, as follows,

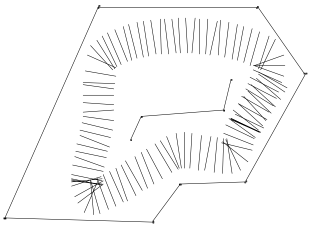


*Figure 8.* The trajectory performed by one of the evolved robots measured using a laser positioning device. Segments represent successive displacements of the axis connecting the two wheels. The direction of motion is anti-clockwise.

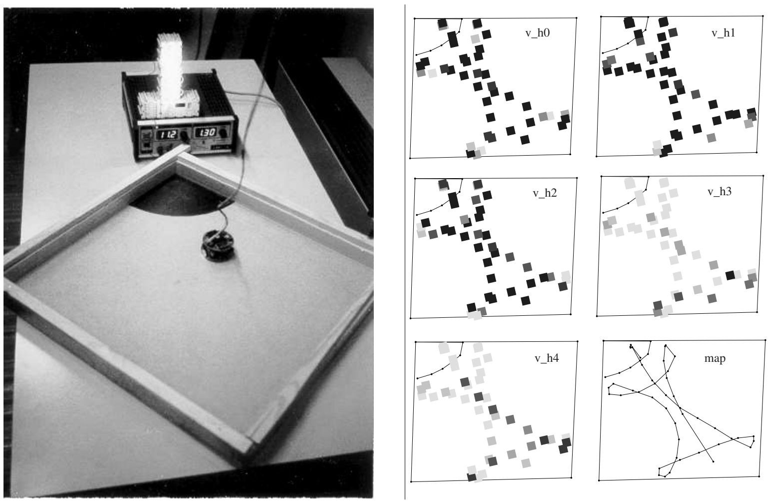
$$\Phi = V \left(1 - \sqrt{\Delta v}\right)(1 - i) \tag{2}$$

where $V$ is a measure of the average rotation speed of the two wheels, $\Delta v$ is the absolute value of the algebraic difference between the signed speed values of the wheels (positive is one direction, negative the other) and $i$ is the normalised activation value of the proximity sensor with the highest activity. The fitness values are accumulated during the "life" of each individual and then divided by the total number of actions performed. The function $\Phi$ has three components: the first one is maximised by speed, the second by straight direction, and the third by obstacle avoidance. Each artificial chromosome encodes the synaptic weights and unit thresholds of a neurocontroller with eight input units clamped to the Khepera's sensors and two motor units with recurrent connections, each controlling one motor of the robot. All the 80 chromosomes of the initial population are randomly generated and each individual is separately tested in the environment. The whole procedure is entirely automated and we can observe the population statistics on our workstation a few offices away.

After approximately 45 generations, during which the average population fitness steadily increases, an individual is born which exhibits smooth navigation around the maze (figure 8) [7]. Despite the simplicity of this experiment, which is merely intended to explore the feasibility of the evolutionary approach, there are at least three interesting emergent properties of the evolved system which were not predefined by the engineer. The first concerns the development of appropriate recurrent connection strengths on the motor neurons which prevents the robot from getting stuck in situations when contralateral sensors are equally activated, therefore generating behaviours which are more efficient than a feed-forward controller [29]. The second properties is the direction of motion. Although the robot is perfectly circular and the fitness function does not specify the direction of motion, the evolved controller always moves in the direction with higher sensor density which gives a better resolution of encountered obstacles. Finally, the evolved controller displays an optimal cruising speed that, although it is not the maximum available speed, it well matches the geometry of the environment, the sensor characteristics, and the sensor update rate. If the robot moved faster, it would crash into a wall before having the possibility of detecting it.

### 4.2. COMPLEX ENVIRONMENTS GENERATE COMPLEX CONTROLLERS

The experiment described above indicates that evolution can automatically develop a set of smart and efficient solutions tailored for the environment where the robot operates. Here we describe another experiment where some changes in the environment and in the robot characteristics lead to different behavioral strategies without making the fitness function more complex and/or changing the evolutionary algorithm. The new environment consists




a) b)

*Figure 9.* a) The environment of the experiment on battery recharge. The light tower is positioned in the far corner over the recharging area which is painted black. There are no other light sources in the room. b) Activation of five internal neurons plotted every 100 ms while the robot navigates in the environment with the robot starting on the lower right corner; darker squares indicate higher activation values. The last frame plots only the trajectory of the robot.

of a 40x45 cm arena delimited by walls of light-blue polystyrene (figure 9a), as in the previous experiment. A 25 cm high tower equipped with 15 small DC lamps oriented toward the arena is placed in one corner and the room does not have other light sources. Under the light tower, a black-painted sector is intended to simulate the platform of a battery charger under construction. When the robot happens to be over the black area, its simulated battery is instantaneously recharged. Beside the eight infrared sensors, the neurocontroller receives input also from two ambient light sensors, each on one side of the body. Additionally, another ambient light sensor is placed under the robot platform, pointing downward, to detect when the robot arrives on the black-painted sector. The robot is provided with a simulated battery characterised by a fast linear discharge rate (max duration: approx. 20 seconds), and with a simulated sensor giving information about the battery status.

The fitness function is a simplified version of that used in the previous experiment, without the second component of equation 4.1 which accounted for straight navigation. The fitness function is accumulated at every time steps. Therefore, robots that could learn to discover the position of the battery charger and periodically return to it would survive longer. It should be noticed that when the robot is on the battery charger, the fitness value is zero because the robot is very close to the walls ($i = 1$).

After 240 generations of continuous fitness progress without human intervention, a robot is born which could navigate in the environment and periodically return to the charging station. In order to maximise the fitness function, the robot returns to the charging station just 1 second before battery failure. By analysing the evolved neurocontroller dynamics and correlating the neuron activation with the robot position while it is freely moving in the environment (thanks to a laser positioning device), it was found that some neurons are responsible for obstacle avoidance and others for homing to the recharger. One of the latter units displays a coding of the environment structure and charger position which is used by the robot to locate its own position, correlate it with remaining energy, and decide whether it is necessary to return to the station [8]. Localisation of the charging station, straight trajectories, self-localisation, and exact timing of the recharge phase are all emergent properties which evolved out of a very simple fitness function in order to satisfy the constraints posed by the environment in which the robot operated.

## 4.3. CO-EVOLUTIONARY ROBOTICS

In the effort to make the environment more complex and dynamic, and at the same time attempt to reduce human design even further (which up to now had been the formulation of simple fitness functions), we turned our attention to co-evolution of competitive robots. In the simplest scenario of two co-evolving and competing populations (for example, predator and prey), fitness progress of one species is achieved at disadvantage of the other population's fitness.

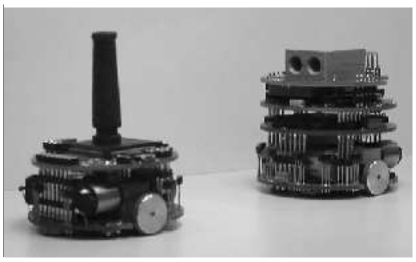


*Figure 10.* Right: The Predator is equipped with the vision module described in section 2.4. Left: The Prey has a black protuberance which can be detected by the predator everywhere in the environment, but its maximum speed is twice that of the predator. Both Predator and Prey are equipped with 8 infrared proximity sensors (max detection range was 3 cm in our environment).

For this experiment we employ two Khepera robots, a predator and prey, as depicted in figure 10 which evolve in parallel within a square arena of 47 x 47 cm. Each individual of one species is tested against the best individuals of the competitor species. A competition ends either when the predator touches the prey or after 500 sensorimotor updates. The fitness function $\Phi_c$ for each competition $c$ does not require any sensor or motor measurement, nor any global position measure; it is simply $TimetoContact$ normalised by the maximum number of sensorimotor updates $TtC$ for the predator $pr$, and $1 - TtC$ for the prey $py$.

After evolution, each best individual is tested against all the best individuals of the other species; the average fitness of this "Master Tournament" for each best individual across generations is plotted on top of figure 11). Such a Master Tournament tells us two things: At which generation we can find the best prey and the best predator, and at which generation we are guaranteed to observe the most interesting tournaments. The first aspect is important for optimisation purposes and applications, the latter for pure entertainment. The best individuals are those reporting the highest fitness when also the competitor reports the highest fitness (marked by letters A and B in the graph). Instead, the most entertaining tournaments are those between individuals that report the same fitness level, because these are situations where both species have the same level of ability to overcome the competitor.

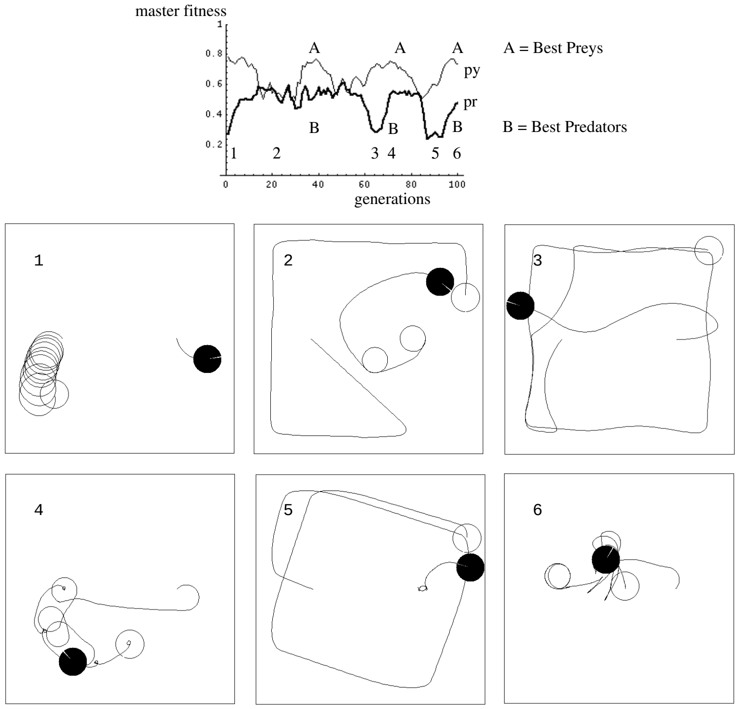



*Figure 11.* Top: Fitness of best individuals in Master Tournament for the two species. Letters indicate position of best prey and best predators. Numbers indicate position of individuals whose tournaments are displayed below. Bottom: Behaviours recorded at interesting points of co-evolution, representing typical strategies. Black disk is predator, white is the prey. See text for a description.

In the lower part of figure 11, behaviours of best competitors at critical stages of co-evolution, as indicated by Master Tournament data, give a more intuitive idea of how pursuit-evasion strategies are co-evolved.

Initially, the predator tends to stop in front of walls while the prey moves in circles (box 1). Later, the prey moves fast at straight trajectories avoiding walls while the predator tracks it from the centre and quickly attacks when the prey is closer (box 2). In box 3, the predator intercepts the prey which accelerates causing the predator to crash against the wall. Around generation 75, we have a typical example of the best prey (box 4); it moves in circles and, when the predator gets closer, it rapidly avoids it. Prey that move too fast around the environment sometimes cannot avoid an approaching predator because they detect it too late (IR sensors have lower sensitivity for a small cylindrical object, like another robot, than for a white flat wall). Therefore, it pays off to wait for the slower predator and accurately avoid it. However, some predators become smart enough to perform a small circle once they have missed the target, and re-attack until, by chance, the prey displays a side without IR sensors. As soon as the prey begin again moving around the environment, the predator develops a "spider strategy" (box 5): it slowly backs until it finds a wall where it waits for the fast-approaching prey. However, this strategy does not pay off when the prey stay in the same place. Finally, at generation 99 we have a new interesting strategy (box 6): the predator quickly tracks and reaches the prey which quietly rotates in small circles. As soon as the prey senses the predator, it backs and then approaches the predator (without touching it) on the side where it cannot be seen; consequently, the predator quickly turns in the attempt to visualise the prey which rotates around it, producing an entertaining dance.

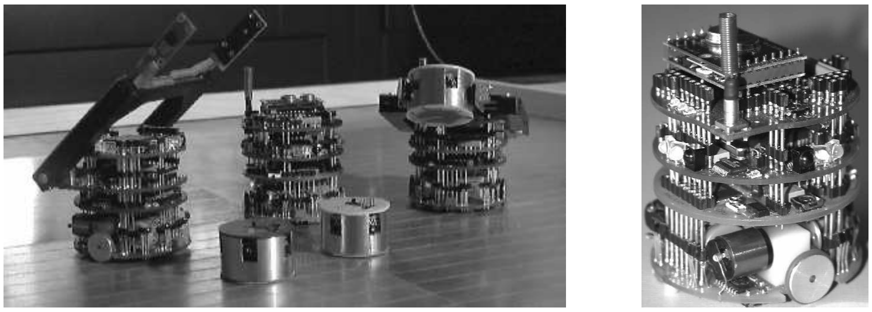
These experiments indicate that competitive co-evolution is a promising technique for automatic gradual evolution of complex behaviours without effort in fitness design [11, 10]. In the following section we shall turn our attention to the study of systems that include more than two autonomous robots.

## 5. Issues in Collective Autonomous Robots

*Collective autonomous robotics* deals with teams of several autonomous robots which are involved in a shared mission. Design and control of the robot group requires the analysis of several collective mechanisms such as communication, interference, and cooperation. One way to tackle the problem is to take inspiration from collective intelligence displayed by social insects [1]. *Bio-inspired collective robotics* favours decentralised solutions and focuses on robot-robot and robot-environment interactions which can potentially lead to robust, goal-oriented, and emergent group behaviours.

Suitable team-behaviours can be achieved both by explicit programming or by adaptation. If the control solution is decentralised, explicitly programming each single robot is easy, but can become prohibitively difficult as the complexity of the desired team behaviour increases. Evolutionary or learning techniques, such as genetic algorithms or reinforcement learning methods, can help the engineer to select adequate behaviours for individual robots in the team. We believe that the integration of adaptive methods with explicit programming can strongly contribute to design a team of self-programming robots for a predefined task.

If only team performance is measured, robots are faced with a *credit assignment problem*, that is the problem of deciding to what extent their own behaviour has contributed to the team overall score [40]. Two ways for bypassing this problem have been proposed. One consists of exploiting



a) b)

*Figure 12.* a) Three Kheperas equipped with different combinations of modules on the energy supply board (from left to right): gripper and infrared modules, infrared and radio modules, and gripper and radio modules. The active seeds complete the set-up picture. b) A closer look at the Khepera with infrared and radio modules.

*global communication* among teammates [35] so that each robot is aware of the action effects of its fellows on the environment. However, this is not a completely decentralised solution and global communication among the robots is not always possible or does not scale well to large team sizes. Another solution consists in measuring the performance of each individual robot instead of team performance [25]. A potential drawback of this approach is that it might force collective behaviour to be the sum of identical individual behaviours, which is not necessarily the optimal strategy for every task constraints. We can achieve *real team solutions* only at the price of dealing with the credit assignment problem. Attempts in this direction in simulated environments have been recently published [40, 31].

At our laboratory we have conducted experiments with real robots who behave according to local programs [23, 22, 24] and are currently investigating the efficiency of genetic algorithms to adapt team behaviour for certain task boundary conditions.

### 5.1. SPECIFIC HARDWARE TOOLS FOR AUTONOMOUS COLLECTIVE ROBOTICS

In evolutionary or learning single-robot experiments, the robot is connected to a workstation through a cable which supplies required energy and supports intensive computing (see sections 2 and 4). However, it is impossible to use such connections for groups of more than two robots because the cables would become entangled.

Figure 12 shows the set-up developed at our laboratory for experiments in collective robots. To achieve extended autonomy we have developed a

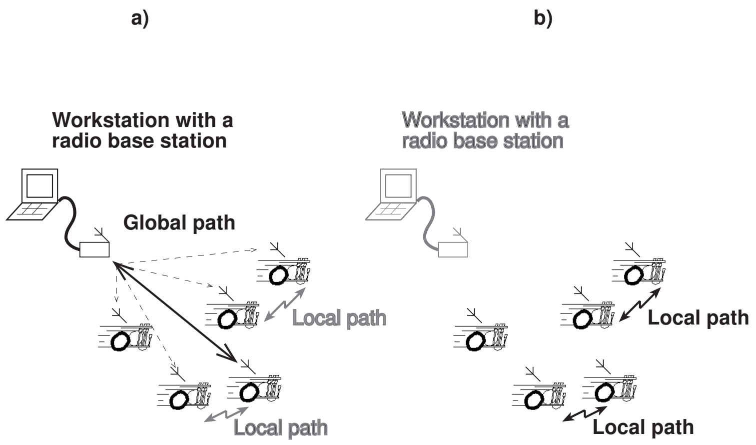


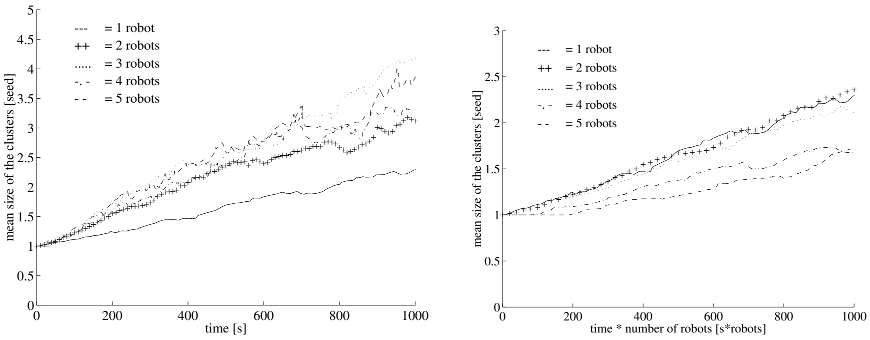

*Figure 13.* a) Global communication path. b) Local communication path.

special floor board as an interface between an external energy supply source and the robots. The required energy is acquired through electrical contacts placed underneath the robot platform, regardless of its position. Communication among robots is limited to neighbouring teammates. Furthermore, a communication link between workstation and robots enables supervision of the adaptive process. Therefore, we have a hierarchical communication strategy which optimises robot-to-robot (local path, via infrared link) and workstation-to-robot (global path, via radio link) communication (see figure 13). Finally, the infrared turret allows the Khepera to distinguish teammates from other objects in the environment.

The special pucks shown in figure 12, which we call "active seeds", introduce the possibility of increasing the environment complexity without increasing the complexity of the robot hardware in object-gathering experiments. The active seeds are capable of synchronously responding to the IR pulses of Khepera's proximity sensors. For instance, active seeds which respond with one pulse every two received pulses are seen by the robot as "blinking" objects. The blinking rate can be changed at pleasure, allowing the presence of several different detectable objects in the environment.

### 5.2. A SIMPLE BIO-INSPIRED COLLECTIVE EXPERIMENT

We present here a biologically inspired experiment concerned with clustering and gathering of scattered passive seeds (small wood cylinders). It is worth emphasising that in both experiments the robots operate completely autonomously using a local subsumption architecture [2] for each robot; there is no explicit communication (IR or radio link) with other robots or




a) b)

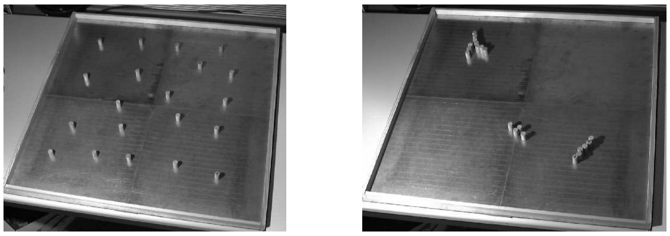
*Figure 14.* (a)Absolute performance of the group with increasing number of robots (1 to 5) on an arena of $80 \times 80$ cm and 20 seeds to be gathered. (b) Relative performance of a single robot within the group (1000 s of work using a single robot have to be compared with 500 s of work using 2 robots, with 333 s using 3 robots, and so on).

with the workstation. The only possible interactions among robots are reciprocal avoidance of collisions and modifications of the environment due to the displacement of the objects by other robots.

The control program that each robot executes can be described as follows. The robot moves on the arena looking for seeds. When its sensors are activated by an object, the robot begins the discriminating procedure. Two cases can occur: if the robot is in front of a large surface (a wall, another robot, or an array of seeds), the object is considered as an obstacle and the robot avoids it. In the second case, the object is identified as a seed. If the robot is not already carrying a seed, it grasps the seed with its gripper; if the robot is carrying a seed, it drops the seed close to the one it has found; then, in both cases, it performs a 180-degree rotation and begins resumes the search.

The experiments are conducted with a group of 1 to 5 Kheperas equipped with the gripper module and 20 scattered seeds in an arena of $80 \times 80$ cm. The measured team performance is the average size of the clusters created in about 17 minutes. Each experiment is repeated 5 times. The average performance of the 5 runs is plotted in figure 14. Typical patterns of seed scatter observed at the beginning and at the end of a longer experiment (120 minutes, repeated 3 times) are shown in figure 15.

Despite the simplicity of these experiments we can derive some interesting conclusions. The probability of incrementing the mean size of the clusters is always bigger than that of decrementing it. On the sole basis of simple local interaction rules, it is therefore possible to create clusters starting from a random placement of the seeds on the arena (see figure 14a).



a) b)

*Figure 15.* Seed scattering a) at beginning of the experiment and b) after 2 hours, at the end of the longest experiment. The experiment has been conducted with the help of the extended autonomy tool.

Figure 14b clearly shows that there is no superlinearity in the team performance. On the contrary, groups with 4 and 5 robots show sublinear team performances due to interferences among teammates. After about one hour, the average cluster size reaches a saturation zone where interference and building gradient contributions are in equilibrium. We never observed all the seeds gathered in a single cluster.

The results indicate that in this kind of experiments with no explicit communication among robots and with no adaptation in robot control, a larger number of robots does not necessarily help to increase the team fitness. The introduction of adaptation could, for instance, allow individual robots to switch from an active phase to an inactive one when the ratio between the amount of work (in our case the seed finding rate) and the interference (in our case the encounter rate with other teammates) decreases under a given threshold. Similar mechanisms are supposed to play a crucial role in ant colonies [34].

## 6. Applications of Autonomous Robots

Despite the potentials of the approaches and algorithms described in the previous sections, industrial applications of autonomous mobile robots are not yet widespread. However, the conditions might be ready for an explosion of applications in the next few years.

Given the low complexity of current applications, a traditional engineering approach is typically employed to program basic behaviours (usually in C language) for the various situations that the robot might face. The control program is extensively tested on the robot in order to adjust poor reactions to unplanned situations. Most prototypes developed so far have been based on the layered subsumption architecture proposed by Brooks [2, 3]. These robots display behaviours similar to those described in the previous sections, such as obstacle avoidance, object search, but the environment around the robot is clearly defined and the behaviour is continuously verified.

At the moment of writing this chapter, we are not aware of any application where an autonomous robot automatically improves its own behaviour by modifying internal parameters according to experienced situations. Neural networks, genetic algorithms, and neuro-fuzzy approaches do not offer yet enough guarantee of stability in unsupervised situations for being accepted as a viable method in critical applications where risks and costs rank high. For real-world applications the robot must be sold in numerous copies to customers who will read a short set of recommendations, power the robot, and check from time to time that the work is being properly done. Customers are not willing to spend time in instructing their robot, letting it carefully explore the environment, and buy the risk of sub-optimal performance. Processors, interfaces, and sensors are sufficiently reliable nowadays, but not so is complex software. One possibility is that of combining deterministic and adaptive software, but much work in this direction remains to be done.

Another challenge for applications is energy autonomy which is currently supported by solar cells or on-board accumulators, but could be extended by including a behavioural module that guides the robot to a recharging station when necessary. Power requirements are negligible for sensor reading and processing, but become considerable when it comes to move actuators, such as wheels and grippers. Current limitations in energy autonomy naturally favour "white-collar" applications of autonomous robots, such as surveillance.

## 6.1. FULLY AUTONOMOUS APPLICATIONS

A fully autonomous, widely marketed robot is the Husqvarna lawn-mower for flat and prepared terrains. The robot is 15 cm high and has a surface of 80 cm by 40 cm covered by solar power cells which let it work for several months when the sun is high over the sky (a small battery back-up is used by the processor when sun light is not strong enough to power the robot). The robot moves randomly, exploiting small irregularities of the terrain, while checking for an electric wire (solar powered too) positioned on the perimeter by the owner. In case the robot gets stuck in unexpected situations, it starts beeping and waiting for human help. Lawn-mowing is a simple navigation task where random walk seems acceptable; furthermore, since the wheels move faster when the lawn is cut, the robot tends to spend more time on areas not yet cleared. Pool cleaning robots share some characteristics with the autonomous lawn-mower. Although several types are available on the market (e.g., see `http://h2o-marketing.com/aquabot/aqua.html`), they generally perform a random walk on the bottom and on the sides while scrubbing, vacuum cleaning, and filtering the pool. They are generally powered via a cable hanging from the centre of the pool and can also be remotely controlled, if necessary. A more systematic cleaning of the pool inner surface can be achieved by pressure sensors which exploit the regularities of tiles.

In both the applications described above, the environment is simple and stable enough to expect reliable operation. Unfortunately, not the same applies for home vacuum cleaners. Here the robot cannot rely on a regular environment, on wires delimiting perimeter and obstacles, on uniform surfaces, and on solar cells or a cable. Energy requirements are very stringent and batteries with a reasonable size might last only 5 minutes before needing a lengthy recharge. Currently, autonomous vacuum cleaning robots are restricted to specific large environments, such as airport lounges (Narita airport in Japan, e.g.). A prototype robot in the Paris metro was designed to follow a line buried in the ground, whereas the recently completed CLEAN Eureka project (nr. EU-1094 on the Eureka database: `http://www.eureka.be`) has attempted to develop a robot for cleaning hyper-market surfaces by exploiting pre-positioned active landmarks (see the concept of active seeds for extended sensors described in section 5.1).

Autonomous mobile robots seem a very suitable application for searching anti-personnel landmines [32]. However, several factors make this application very difficult: the terrain is often impracticable for wheeled robots (such as tall vegetation on abandoned fields), there are not yet powerful mine sensors available, and an almost systematic search is required to guarantee that no landmines are left behind (for military applications, such as breaching, not all mines must be neutralised: human and material losses are weighted against the strategic value of crossing the field). Recently, a legged water-proof robot has been developed for landmines positioned on the surf zone [17], which is a rather regular and defined terrain. All these robots are supposed to blow up mines by hitting them (and being destroyed in the meanwhile).

### 6.2. PARTIALLY AUTONOMOUS APPLICATIONS

The Mars Sojourner (`http://mpfwww.jpl.nasa.gov/default.html`) is a popular example of a robot with full energy autonomy, but limited behavioural autonomy. Since it moves very slowly, a rather small area of solar cells is sufficient to power the robot. It receives instructions from Earth on its destination, but it has to get there autonomously.

Another application with similar behavioral requirements is the autonomous wheelchair [16]. Several handicapped persons find it difficult to steer precisely their own wheelchair to get around corners or passing through doorways. By supplying these chairs with additional sensors and appropriate control systems that support semi-autonomous navigation, the owner can instruct the chair on the desired destination and let it get there autonomously.

Semi-autonomous mobile robots have a large potential market, from rescue robots to robots for maintenance of nuclear plants, and have several military applications, such as reconnaissance flying drones. An application that has attracted the interest of several industries, research institutes, and funding agencies is a semi-autonomous vehicle capable of navigating in daily traffic as well as on rough terrains. A well-known example is NavLab [20], developed by Carnegie Mellon University. The Swiss Serpentine [37] is a urban semi-autonomous vehicle designed for accommodating several standing persons which follows an inductive track providing power, self-localisation, and general directives on the task to be accomplished.

As it can be seen from the examples reported here, safety compliances, costs, stability, and risks are important issues in industrial applications where often a traditional engineering approach is still preferred over advanced learning abilities. However, certain niches of the market seem ripe for new forms of autonomous robot featuring some form of self-organisation. Entertainment and companion robots have recently attracted the interest of several small and large companies [13]. Here, the learning aspect is important both for adapting the robot to the requirements of the owner and for providing it with novel behaviours added to the old ones, which would otherwise become soon boring. Similarly, space exploration and asteroid mining will profit from robots equipped with higher autonomy and decision abilities.

## 7. Conclusion

In this chapter we have provided an overview of some important issues in mobile autonomous robotics, emphasising the directions explored at our laboratory from hardware to software development. At the hardware level, we have insisted on the development of appropriate research tools to investigate approaches and applications on an incremental and comparative fashion. The success of the concept behind the miniature mobile robot Khepera is indeed its strategic position between simulations and applications, between research development and industrial application, and between roboticians and people from other disciplines.

At the software level, three different approaches have been described here in further detail: neuro-fuzzy, evolutionary, and collective systems. Each of these fields of investigations has its own interest, advantages, and drawbacks, which become evident when all approaches are tested and compared on the same robot platform. Neuro-fuzzy systems offer the unparallelled feature of combining a low-level learning algorithm which operates on a set of knowledge-based rules. The result of the learning phase can then be transformed back into easily interpretable rules, as opposed to the often unintelligible pattern of synaptic connections in neural networks. However, the presence of a human supervisor during the learning phase limits somewhat the autonomy and applicability of the method. Evolutionary systems are powerful algorithms to develop interesting behaviours without much human intervention and knowledge. A clear drawback, however, is that they require considerable training time during which power must be somehow externally supplied. Collective systems, a relatively unexplored terrain, introduce new complexities which require a reconsideration of several issues at hardware and software level. Study of decentralised control in colonies of autonomous robots can provide interesting surprises, such as those outlined in section 5.2, which can be appropriately used as a starting point when developing applications requiring several interacting robots. Despite the obvious research interest and biological relevance, it is not yet clear whether application potentials can match hardware costs, both at the development phase and during final operation.

As outlined in the previous section on applications, it is clear that research on autonomous robots capable of self-organisation is still far ahead of industrial applications. We are still at a crucial investigation phase where it would be deleterious and short-minded to declare what are the most promising directions and what are those that should be abandoned. Probably, some combinations of these and other techniques will reciprocally overcome their own respective drawbacks. Some investigations are already underway on the combination of evolution and fast learning algorithms [9], on fuzzy logic to analyse and control the result of autonomous learning, and on adaptive local methods embedded in collective systems.

## References


1. E. G. Bonabeau and Theraulaz G. *Intelligence Collective*. Hermès, Paris, France, 1994.
2. R. A. Brooks. A robust layered control system for a mobile robot. *IEEE Robotics and Automation*, RA-2:14–23, March 1986.
3. R. A. Brooks. Intelligence without representation. *Artificial Intelligence*, 47:139–59, 1991.
4. D. Cliff, P. Husbands, J. A. Meyer, and S. W. Wilson, editors. *From Animals to Animats: Proceedings of the Third International Conference on Simulation of Adaptive Behavior*, Cambridge, MA, 1994. MIT Press/Bradford Books.
5. M. Dorigo. Special issue on learning autonomous robots. *IEEE Transactions on*

*Systems, Man and Cybernetics-Part B*, 26:361–364, 1993.
6. M. Dorigo and U. Schnepf. Genetic-based machine learning and behavior based robotics: a new synthesis. *IEEE Transactions on Systems, Man and Cybernetics*, 23:141–154, 1993.
7. D. Floreano and F. Mondada. Automatic Creation of an Autonomous Agent: Genetic Evolution of a Neural-Network Driven Robot. In D. Cliff, P. Husbands, J. Meyer, and S. W. Wilson, editors, *From Animals to Animats III: Proceedings of the Third International Conference on Simulation of Adaptive Behavior*, pages 402–410. MIT Press-Bradford Books, Cambridge, MA, 1994.
8. D. Floreano and F. Mondada. Evolution of homing navigation in a real mobile robot. *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, 26:396–407, 1996.
9. D. Floreano and F. Mondada. Evolution of plastic neurocontrollers for situated agents. In P. Maes, M. Mataric, J-A. Meyer, J. Pollack, H. Roitblat, and S. Wilson, editors, *From Animals to Animats IV: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*, pages 402–410. MIT Press-Bradford Books, Cambridge, MA, 1996.
10. D. Floreano and S. Nolfi. Adaptive behavior in competing co-evolving species. In P. Husbands and I. Harvey, editors, *Proceedings of the 4th European Conference on Artificial Life*, Cambridge, MA, 1997. MIT Press.
11. D. Floreano and S. Nolfi. God save the red queen! competition in co-evolutionary robotics. In J. Koza, K. Deb, M. Dorigo, D. Fogel, M. Garzon, H. Iba, and R. L. Riolo, editors, *Proceedings of the 2nd International Conference on Genetic Programming*, San Mateo, CA, 1997. Morgan Kaufmann.
12. N. Franceschini, J.-M. Pichon, and C. Blanes. Real time visuomotor control: From flies to robots. In *Proceedings of the Fifth International Conference on Advanced Robotics*, pages 91–95, Pisa, June 1991.
13. M. Fujita and K. Kageyama. An open architecture for robot entertainment. In *ACM Conference on Autonomous Agents*, pages 435–442, Marina Del Rey, February 1997.
14. P. Gaussier. Special Issue on Animat Approach to Control Autonomous Robots interacting with an unknown world. *Robotics and Autonomous Systems*, 16, 1995.
15. J. Godjevac. *A Method for the Design of Neuro-Fuzzy Controllers; an Application in Robot Learning*. PhD Thesis N° 1602, École Polytechnique Fédérale de Lausanne, 1997.
16. T. Gomi and K. Ide. The development of an intelligent wheelchair. In *Intelligent Vehicles Symposium*, Tokyo, 1996.
17. H. Greiner, C. Angle, T. Freed, J. Jones, P. Ning, R. Elsley, and G. Bane. Autonomous legged underwater vehicles for near land warfare. In *Autonomous Vehicles in Mine Countermeasures Symposium*, pages 14–22, Monterey, CA, 1995.
18. J. H. Holland. *Adaptation in natural and artificial systems*. The University of Michigan Press, Ann Arbor, 1975.
19. N. Jakobi. Half-baked, ad-hoc and noisy: Minimal simulations for evolutionary robotics. In P. Husbands and I. Harvey, editors, *Proceedings of the 4th European Conference on Artificial Life*, Cambridge, MA, 1997. MIT Press.
20. A. Kelly and A. Stenz. Analysis of requirements for high speed rough terrain autonomous mobile robots. In *IEEE Conference on Robotics and Automation*, pages 3318–3333, Albuquerque, 1997.
21. P. Maes, M. Mataric, J-A. Meyer, J. Pollack, H. Roitblat, and S. Wilson, editors. *From Animals to Animats: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*, Cambridge, MA, 1996. MIT Press/Bradford Books.
22. A. Martinoli, E. Franzi, and O. Matthey. Towards a reliable set-up for bio-inspired collective experiments with real robots. In *Proc. of the Fifth International Symposium on Experimental Robotics ISER-97*, Barcelona, Spain, June 1997. In Press.

23. A. Martinoli and F. Mondada. Collective and cooperative group behaviours: Biologically inspired experiments in robotics. In *Proceedings of the Fourth International Symposium on Experimental Robotics*, Stanford, U.S.A., 1995. Springer Verlag.
24. A. Martinoli, M. Yamamoto, and F. Mondada. On the modelling of bioinspired collective experiments with real robots. In *Proceedings of the Fourth European Conference on Artificial Life ECAL-97*, Brighton, UK, July 1997. `http://www.cogs.susx.ac.uk/ecal97/present.html`.
25. M. J. Matarić. Learning in multi-robot systems. In G. Weiss and S. Sen, editors, *Adaptation and Learning in Multi-Agent Systems*, volume 1042, pages 152–163. Springer Verlag, Lecture Notes in Artificial Intelligence, 1996.
26. J. A. Meyer, H. L. Roitblat, and S. W. Wilson, editors. *From Animals to Animats: Proceedings of the Second International Conference on Simulation of Adaptive Behavior*, Cambridge, MA, 1993. MIT Press/Bradford Books.
27. J. A. Meyer and S. W. Wilson, editors. *From Animals to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior*, Cambridge, MA, 1991. MIT Press/Bradford Books.
28. O. Michel. Evolutionary neurogenesis applied to mobile robotics. In M. Patel and V. Honovar, editors, *Advances in Evolutionary Synthesis of Neural Systems*. MIT Press, Cambridge, MA, 1997.
29. F. Mondada and D. Floreano. Evolution of neural control structures: some experiments on mobile robots. *Robotics and Autonomous Systems*, 16:183–195, 1995.
30. F. Mondada, E. Franzi, and P. Ienne. Mobile robot miniaturization: A tool for investigation in control algorithms. In T. Yoshikawa and F. Miyazaki, editors, *Proceedings of the Third International Symposium on Experimental Robotics*, pages 501–513, Tokyo, 1993. Springer Verlag.
31. A. Murciano and J. del R. Millán. Learning signaling behaviors and specialization in cooperative agents. *Adaptive Behavior*, 5(1):5–28, 1997.
32. J. D. Nicoud. Vehicles and robots for humanitarian demining. *Industrial Robot Journal*, 24(2):168, 1997.
33. S. Nolfi, D. Floreano, O. Miglino, and F. Mondada. How to evolve autonomous robots: Different approaches in evolutionary robotics. In R. Brooks and P. Maes, editors, *Proceedings of the Fourth Workshop on Artificial Life*, pages 190–197, Boston, MA, 1994. MIT Press.
34. S. W. Pacala, D. M. Gordon, and H. C. J. Godfray. Effects of social group size on information transfer and task allocation. *Evolutionary Ecology*, 10:127–165, 1996.
35. L.E. Parker. The effect of action recognition and robot awareness in cooperative robotic teams. In *Proceedings of IEEE International Conference on Intelligent Robots and Systems IROS-95*, volume 1, pages 212–219, Pittsburgh, PA, August 1995. Springer Verlag.
36. R. Pfeifer. Cognition – perspectives from autonomous agents. *Robotics and Autonomous Agents*, 15:47–70, 1995.
37. B. Saugy and R. Rovira. The Serpentine, a step forward in intelligent transport system. In *Proceedings of the Conference on Robotics and Intelligent Systems (IROS97)*, Grenoble, France, September 1997.
38. L. Steels. The Artificial Life Roots of Artificial Intelligence. *Artificial Life*, 1:75–110, 1994.
39. M. Sugeno and K. Murakami. An Experimental Study on Fuzzy Parking Control Using a Model Car. In Michio Sugeno, editor, *Industrial Applications of Fuzzy Control*, chapter 8, pages 125–138. North-Holland, 1985.
40. C. Versino and L. M. Gambardella. Ibots learn genuine team solutions. In M. Van Someren and G. Widmer, editors, *Proceedings European Machine Learning ECML-97*, pages 298–311, Kyoto, Japan, 1997. Springer Verlag. Lecture Notes in Artificial Intelligence.