A multi-robot system for adaptive exploration of a fast changing environment: probabilistic modeling and experimental study

Aude Billard[2,1], Auke Jan Ijspeert[2,1], Alcherio Martinoli[1]

[1] LAMI, Swiss Institute of Technology (EPFL)
IN-F Ecublens, 1015 Lausanne, Switzerland
Tel: +41-21-693-39-07, Fax: +41-21-693-52-63
[2] - Instituto Dalle Molle di Studi Sull'Intelligenza Artificiale,
Corso Elvezia 36, CH-9600 Lugano
`Aude.Billard@epfl.ch`,
`http://diwww.epfl.ch/w3lami/team/billard/`

Abstract. This paper presents an experiment in collective robotics which investigates the influence of communication, of learning and of the number of robots in a specific task, namely learning the topography of an environment whose features change frequently. We propose a theoretical framework based on probabilistic modeling to describe the system's dynamics. The adaptive multi-robot system and its dynamic environment are modeled through a set of probabilistic equations which give an explicit description of the influence of the different variables of the system on the data collecting performance of the group. Further, we implement the multi-robot system in experiments with a group of Khepera robots and in simulation using Webots, a 3-D simulator of Khepera robots. The robots are controlled by a distributed architecture with an associative-memory type of learning algorithm. Results show that the algorithm allows a group of robots to keep an up-to-date account of the environmental state when this changes regularly. Finally, the results of the simulated and physical experiments are compared to the predictions of the probabilistic model. It is found that the model shows both a good qualitative and quantitative correspondence to these results. This suggests that a probabilistic model can be a good first approximation of a multi-robot system.

Keywords: Learning in a dynamic environment - multi-robot system - probabilistic modeling - local and global communication

1 Introduction

Numerous works on autonomous robot systems investigate the questions of 1) whether it is more efficient to distribute the area of expertise needed for performing a complicated task between several robots rather than designing a unique expert robot [6, 9, 16]; 2) whether the use of explicit communication could improve the performance of a group of robots in a collaborative task ([1], [2], [8], [17], [22]); 3) what learning abilities should the robot(s) be provided with for adapting to a continuously changing environment [7, 14, 18, 21].

We address these three issues in a specific task, namely learning the topography of an environment whose features, the locations of objects, change frequently. The locations of objects are learned by a group of *worker robots* which constantly search the environment, and which communicate to each other their knowledge as they meet. The information gathered by each robot is also transmitted to a static *database robot* which each robot visits regularly, and which keeps an up-to-date account of the global state of the dynamic environment. The experiments are implemented with groups of Khepera robots and in simulation using Webots, a 3-D simulator of the Khepera robots [13]. The robots are controlled by a distributed architecture, with an associative-memory type of learning algorithm. The algorithm is simple and, as such, makes no contribution to connectionist architecture. It is however well indicated for the task and allows a group of robots to keep an up-to-date account of an environment which changes regularly and very frequently (we study periods of changes of a few seconds).

We propose a theoretical framework based on probabilistic modeling to describe the system's dynamics. The aim of the model is to give an explicit description of the influence of the variables of the system, namely the number of worker robots, the frequency of environmental changes and the environment's configuration, on the data collecting performance of the group. The model can be used to analyze the experiment's dynamics, and to predict its main characteristics such as the average time necessary for a group of robots to discover all objects in an environment.

The work presented in this paper brings three new contributions to research in collective robotics: 1) the probabilistic model as an abstract representation of a multi-robot system (related work can be found in [5] for the description of ant societies, and in [11, 22] for robotic systems, but without an explicit representation of the system's dynamics), 2) the comparison of three different levels of implementation (the theoretical model, the simulation, and the physical experiment), and 3) the mapping of a *dynamic* (although very simple) environment, rather than a static one (see [4, 20]), for instance).

The rest of this paper is divided as follows. Section 2 describes the experimental set-up and the robots' controllers used in the experiments. Section 3 gives the equations of the probabilistic model. Sections 4 and 5 present the results of two sets of experiments, with local and global communication respectively, and compare the results with the predictions of the probabilistic model. Section 6 concludes the paper with a short summary and discussion of the results.

2 The experimental set-up

2.1 Set-up for experiments with local communication

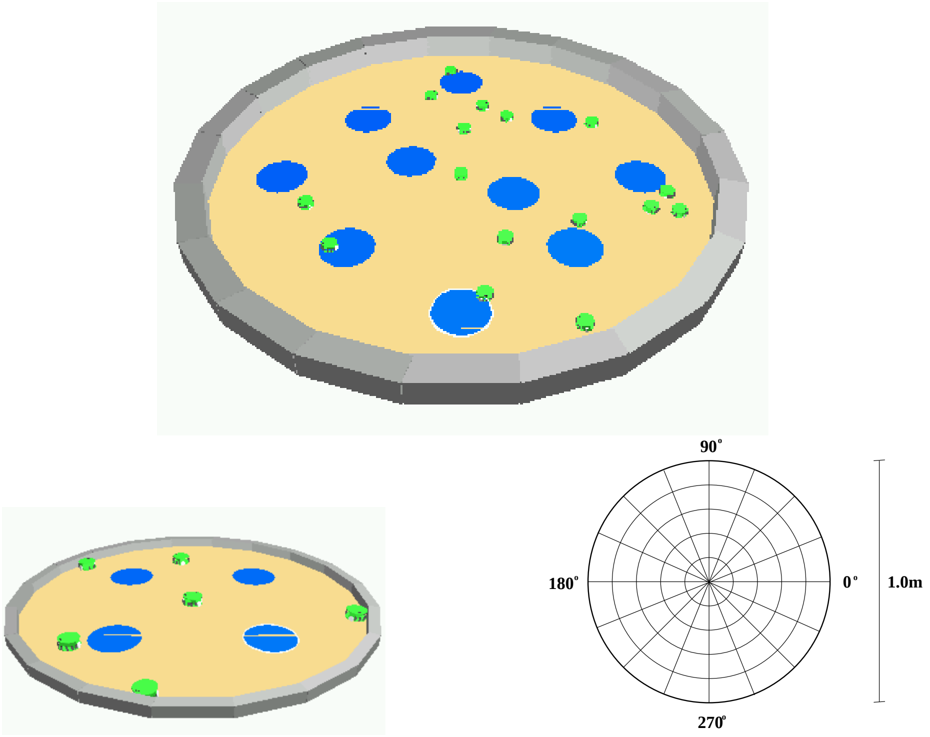

Fig. 1. Set-up of simulations of section 4: Arenas of 2 meters (top) and 1 meter (bottom left) of diameter with respectively 5 and 15 *worker* robots. The database robot stands in the center of each arena. There are 10 and 4 objects in the big (top) and small (bottom left) arenas respectively represented as patches of 0.1m and 0.07m diameter lying on the floor. **Bottom right** Division of the small arena into 80 zones.

The experiments with local communication (section 4) are carried out in Webots [13], a 3-D simulator of the Khepera [15] robots. The experiments are realized in two circular arenas of 1 meter and 2 meters diameter respectively as shown in figure 1.

The simulator gives a faithful representation of the Khepera robots [15], by introducing noise in the robots' movements and sensors measurements as measured on the real Khepera robots. Each robot is provided with 9 infra-red (IR) sensors (8 are used to detect other robots and the arena walls, the 9th IR is activated only by the walls and allows to distinguish between robots and walls), a detector of ground color (used to distinguish between zones with/without objects), a radio transceiver (418 MHz + baud rate 9600), a compass with 5^0 degrees precision and one odometry counter on each wheel.[1] Compass and odometry sensors are used by the robots to determine their location relative to the center of the arena. The robots reset their position to the correct one each time they meet the database robot or hit a wall in the arena. The odometry errors are therefore contained within a range of up to 10 percent error. The objects' locations, given as an angle and a distance relative to the center, are determined following a scaling of the arena into $5 \cdot 16 = 80$ (small arena) and $10 * 16 = 160$ (big arena) zones, see schema of figure 1. Thus, the objects' locations are known within a precision of 22.5 degrees (for the angle) and 10 cm (for the distance). Note the new object locations after each update are chosen randomly in the simulations. The database robot is a static Khepera robot placed in the middle of the arena (see figure 1). The robots' controller is described below in section2.3.

2.2 Set-up for experiments with global communication

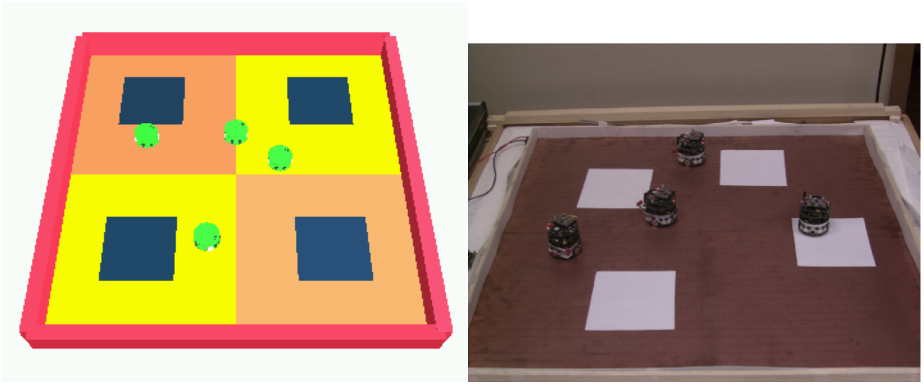

Fig. 2. **Set-up of section 5:** Square arena of 78 by 78 cm with 4 robots (Webots simulation (left) and physical set-up (right)). There are 4 objects represented as square patches.

The experiments with global communication (section 5) are carried out both with real Kheperas and in simulation, with 1 to 4 robots in a square arena of 78 by 78 cm (figure 2). In these experiments, the database robot is a workstation connected to a radio transceiver placed on the bench which runs a C program with the same learning algorithm as the one used by the worker robots (see section 2.3).

Instead of using odometry and the reseting strategy described above, the real robots use a KPS (Khepera Positioning System [10]) for determining their

---

[1] All sensors used in the simulations exist and could be used on the real Khepera robots.

position and orientation with precisions of 5mm and 5 to 10 degrees, respectively. The robots are continuously powered by the arena's floor, which is covered with electrified copper bands [12]. Although the robots are provided with ground color detectors and can thus distinguish between patches on the floor (the objects in the simulation) and the rest of the arena, the objects in the physical experiments are defined virtually (that is, the locations of the patches are predefined in each robot's controller), for simplicity reasons. This approach does not require the experimenter to manually move the patches and ensures a constant frequency of environmental update.

2.3 The robots' controllers

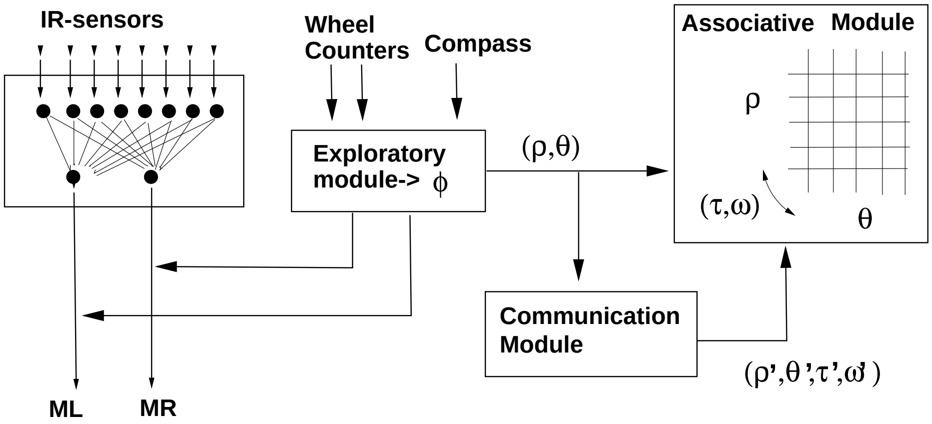

Fig. 3. The robots' controller.

In both experiments, all worker robots have the same controller which consists of five modules (see figure 3): 1) an *obstacle avoidance module* which consists of a one-layer real value feed forward neural network with eight input units (one for each infra-red sensor measurement) and two output units for the two motors (speed proportional control of the motors); 2) a *memory-based exploration module* which determines the robot's direction of travel when crossing the border between two zones of the arena (following the division represented in figure 1); each robot keeps track of the number of times it has crossed each zone; when it estimates that it has reached the border between two zones, the robot turns towards the zone it has less visited so far with an the angle of turn randomly chosen between 0 and PI, 3) a *communication module* for the broadcast of the object locations; in the experiments with local communication (section 4), a robot can communicate in two occasions: when it discovers an object, it broadcasts locally (within a limited range) the location of the object; when it meets another robot, it transmits (using point-to-point protocol, i.e. with acknowledgement from the receiver robot) the location of one object chosen randomly over all locations it knows; in the experiments on global communication (section 5), the robots communicate only when they discover an object (broadcast with infinite range). 4) an *odometry module* which calculates the robot's position relative to the database (center of the arena) given the measurements of the wheels' counters and of the compass; 5) a *learning module* which consists of a bidirectional associative memory; the robots keeps track of the objects' locations by associating the two outputs of the odometry module which are the angle θ and the distance ρ, i.e. the polar coordinates of the robot relative to the center of the arena. Each connection of the module between an angle and a distance measurement is bidirectional and is associated with two parameters, a weight $w_{ij} = w_{ji}$ and a time parameter $\tau_{ij} = \tau_{ji}$. The associative module takes two binary (1/0) vectors as inputs; the vectors encode the robot's measures of angle and distance (following the arena's scaling, see figure 1); there is one active bit per vector at any point of time (e.g. *dist-vect*= $[00100] = 30[cm]$ and *dist-vect*= $[00010] = 40[cm]$). The weights w and time parameters τ are two matrices of 10 by 16 units (for simulations in the big arena) and of 5 by 16 units (for simulations in the small arena). The experiments starts with all weights w and time parameter set to zero. The learning algorithm is a system of three rules:

1. **Learning by seeing:**
   If the robot detects an object, then
   $$w_{\theta,\rho} = 99 \text{ and } \tau_{\theta,\rho} = t$$
   where t is the time measured by the clock of the robot[2].
2. **Forgetting:**
   If the robot crosses a location given by θ, ρ such that $w_{\theta,\rho} > 0$ but does not detect a object, then
   $$w_{\theta,\rho} = 1 \text{ and } \tau_{\theta,\rho} = t$$
3. **Learning by hearing:**
   If the robot hears the location of an object as told by another robot, then:
   $$\text{If } \tau'_{\theta',\rho'} > \tau_{\theta',\rho'} \text{ then } w_{\theta',\rho',} = \frac{1}{2} \cdot (w_{\theta',\rho'} + w'_{\theta',\rho'}) \text{ and } \tau_{\theta',\rho'} = \tau'_{\theta',\rho'}$$
   $\theta', \rho', w'_{\theta',\rho'}, \tau'_{\theta',\rho'}$ are the distance, angle, weight and time parameter transmitted by the emitter robot.

The learning for one robot is evaluated by counting the number of correctly memorized object locations. Learning is successful when this number is equal to the number of different locations. An object located at the coordinates $\{\theta, \rho\}$ is considered as correctly memorized when the weight $w_{\theta,\rho}$ is greater than a

---

[2] The clock is incremented at each processing cycle and is set to zero when the experiment starts.

threshold H. H is calculated at each time step as a function of the current value of all the weights w: $H = \frac{\text{median}_{w>0}(w)+\text{mean}_{w>0}(w)}{2}$, where $\text{median}_{w>0}(w)$ and $\text{mean}_{w>0}(w)$ are, respectively, the median and the arithmetic mean calculated over all $w > 0$. The arbitrarily chosen H estimates the threshold between the important weights (close to 99) which correspond to strong correlations and the small weights (close to 1) which are noisy or no longer valid correlations.

The database robot's controller consists of the same learning module as that used in the worker robots. In the experiments on local communication (section4), when worker and database robots meet, the worker robot transmits to the database robot its two matrices of weights and of time parameters (all w and τ). Following rule 3, the database robot updates its knowledge by calculating an averaged value between its current set of weights and those newly transmitted, if and only if the new information is more recent than its current one. The database robot transmits then back to the worker robot the averaged matrix of weights and time parameters. After a meeting with the database, a worker robot has therefore the same global knowledge of the environment as the database. This speeds up the forgetting process as the robot can then verify more locations (all the locations which have been recorded by the group) than only those it had stored itself. This exchange of matrices weights does not occur in the experiments on global communication (section 5). In that case, the database robot stores the locations of the newly discovered objects as soon as this one is broadcasted by the robot which has discovered it, as the robots' broadcast is audible by all (unless interference). All worker robots can pick up this signal. Thus, worker robots and database robot have almost the same knowledge at all time (small differences exist due to the interferences) and there is no need for the database robot to send any information in return to the one it picks up.

For a discussion of the similitude/difference this algorithm bears to other connectionist models, the reader can refer to the description of DRAMA[3] (a connectionist architecture for on-line learning of spatio-temporal regularities and of time series in autonomous robots) of which the present algorithm is a simplification.

3 The probabilistic model

In this section, we determine sets of probabilistic equations to model the learning dynamics of the experiments using local communication (section 4) and global communication (section 5). The model is based on the assumption that the information gathering process (learning of the locations) is essentially a stochastic process based on simple geometrical considerations. It assumes that the exact trajectories of the robots, the details of the learning and communication events can be ignored and that the result of the learning can be represented as a set of probabilities of occurrence.

3.1 Local communication

The aim is to define an equation which will allow us to determine T the minimal time for the database robot to learn the locations of N_s objects, given that there are N_r worker robots, that the arena has size A and that an object covers a surface S_s.

We define the building blocks or fundamental probabilities of the model by considering the geometrical configurations of the system. We define the probability of meeting the database robot (P_{db}) as the ratio of the surface of detection of the database robot by another robot S_d over the arena's surface A, i.e. $P_{db} = S_d/A$. Similarly, the probabilities of meeting another robot ($P_r = S_r/A$), of passing across a source ($P_s = S_s/A$) or of being in the range of communication of another worker robot ($P_c = S_c/A$) are the ratios of the surfaces of each of these objects over the arena's surface ($A = \pi \cdot (r^2)$, $r = 0.5[m]$ or $r = 1[m]$ for small/big arenas), $S_d = \pi \cdot 0.1^2[m^2]$ (small arena),, $S_d = \pi \cdot 0.15^2[m^2]$ (big arena) $S_r = \pi \cdot 0.1^2[m^2]$, $S_s = 0.0038[m^2]$ (small arena), $S_s = 0.0078[m^2]$ (big arena), $S_c = \pi \cdot 0.3^2[m^2]$.

Let $P_{\text{success}}(N_r, N_s, T)$ be the probability that the event "the database robot has recorded N_s locations" has occurred after a time T. This event is true if each of the N_s locations have been seen by at least one robot and been transmitted to the database robot at least once in a time T, i.e.:

$$P_{success}(N_r, N_s, T) = 1 - (1 - P_{L-success}(N_s, T))^{N_r}$$

$P_{\text{L-success}}(N_s, T)$ is the probability that a first event "all N_s locations have been transmitted" has occurred within a period $T - t_1$, and that a second event "all N_s locations have been learned" has happened in a period t_1. This probability can be expressed as follows:

$$P_{\text{L-success}}(N_s, T) = P(\textit{see DB in } T - t_1 \textit{ (and) learn object in } t_1)$$

The two events are independent, thus the probability of their co-occurrence for a given pair $\{t_1, T - t_1\}$ is the product of each event's probability. The total probability is the sum over all possible pairs $\{t_1, T - t_1\}$ (time is discretised) of this product:

$$P_{\text{L-success}}(N_s, T) = \frac{\sum_{t_1=1}^{T} P_{\text{learn-object}}(N_s, t_1) \cdot P_{\text{see-database}}(T - t_1)}{\sum_{t_1=1}^{T} P_{\text{learn-object}}(N_s, t_1)} \qquad (1)$$

The probability of meeting the database robot, $P_{\text{see-database}}(T - t_1)$ in equation 1, is the probability of crossing the surface S_d within a period $T - t_1$:

$$P_{\text{see-database}}(T - t_1) = 1 - (1 - P_{db})^{T - t_1}$$

$P_{\text{learn-object}}(N_s, t_1)$ is the probability that the event "a robot has learned N_s object locations in a time t_1" is true. A robot learns about an object's locations if the robot either sees the object P_{S} or hears its location from another robot P_{h}.

$$P_{\text{learn-object}}(N_s, t_1) = (1 - (P_{\text{not-learn-object}})^{t_1})^{N_s}$$

$$P_{\text{not-learn-object}} = (1 - P_{\text{S}}) \cdot (1 - P_{\text{h}})^{N_r - 1}$$

The probability of hearing an object's location from another robot's broadcast is the probability that the three following events are true: 1) the listener robot is within an area S_c around the emitting robot (S_c is the surface within which the communication is audible) and 2) the emitting robot broadcasts the particular location, 3) no other robot out of the $N_s - 1$ (excluding the emitting robot, including the listener robot) is simultaneously emitting in that same area (P_{Interf} is the probability of this event).

$$P_h = P_{\text{Hear}} \cdot P_{\text{Interf}}$$

$$P_{\text{Interf}} = (1 - P_{\text{Hear}})^{N_s \cdot N_r - 1}$$

$$P_{\text{Hear}} = (\frac{Sc}{A}) \cdot (1 - P_{\text{Not-emit}})$$

Event 2 is true if the emitter robot either sees the object (it then broadcasts the location) or if the emitter robot meets another robot which transmits it that particular location. In the later case, the emitting robot chooses 1 location among the $2 * N_s$ it knows, which included the N_s correct and no longer valid locations. Event 2 can occur only if the robot has seen that object within a time $t_o < t_1$ before meeting another robot. It follows:

$$P_{\text{Not-emit}} = (1 - P_s)^{t_1} \cdot (1 - \frac{1}{2 \cdot N_s} \cdot \frac{\sum_{t_o=1}^{t_1} (1 - (1 - P_s)^{t_o}) \cdot (1 - (1 - P_r)^{t_1 - t_o})}{\sum_{t_o=1}^{t_1} 1 - (1 - P_s)^{(}t_1 - t_o)})$$

In the above equations, the unity of surface is the meter and the unity of time corresponds to the time needed to cover the surface S_s (which is the minimal surface considered in the equations). In order to convert the value of time in seconds, T has to be multiplied by $\frac{S_s}{V_r * D_R}$, where $V_r = 0.16[m/s]$ is the maximal speed of the robots and $D_r = 0.055[m]$ is the diameter of the robot.

3.2 Global communication

Experiments with global communication (section 5) are modeled by determining the probability that the event "N_s locations are correctly transmitted in a time T". The difference with the previous model is that, in this experiment, a robot broadcasts an object location only when it sees the object and not when it meets another robot. The above event is true when the broadcast of each of the N_s locations (made by one of the robot which was visiting the surface S_s)

has been correctly picked up by the database robot at least once (i.e. no interference occurred). Referring to the reasoning regarding the previous model, the probability of this event is:

$$P_{success}(N_r, N_s, T) = (1 - (1 - (P_s \cdot (1 - P_s)^{N_s \cdot (n-1)}))^{n \cdot T})_s^N \tag{2}$$

$P_s = S_s/A$. An object surface is $S_s = 0.35 * 0.35/4[m^2]$ and the arena surface is now $A = 0.78 \cdot 0.78 = 0.61[m^2]$.

4 Experiments with local communication

This section reports on simulation experiments carried out in round arenas as shown in figure 1. Communication is local, that is the broadcast of the robot can be picked up only within an area $S_c = 2 \cdot \pi \cdot 0.3^2[m^2]$. First, a set of simulations studies is carried out to determine the minimal time delay T for learning all the four and ten locations of each arena in a static environment. The results of these experiments are compared to the prediction of the probabilistic model of section 3. We then evaluate the performance of the multi-robot system at learning a dynamic environment.

4.1 Learning in a static environment

Webots simulations were carried out in a static environment (i.e. the locations of the objects did not change). The number of worker robots was varied from 1 to 10 and from 1 to 15 in the small and big arenas respectively. For each set-up (i.e. for a given arena and a given number of robots) 10 different runs were carried out with different random seeds. A run simulated 1000 seconds. We measured the mean time delay after which the database robot knew all 4 (small arena) and 10 (big arena) object locations. In figure 4, we compare the prediction of the probabilistic model and the results of the simulations. As one would expect, the more robots, the faster the learning. However, the relation between these two variables is not linear and the increase of time efficiency saturates for important numbers of robots. Thus, if one would consider implementing the system in a real robotic set-up based on these results, one would determine the optimal number of robots by comparing the gain in time efficiency to the cost increase when augmenting the number of robots.

For both small and big arenas, the results of the probabilistic model and of the simulations are qualitatively and quantitatively similar. This means that, although the probabilistic model is a crude representation of the system, it approximates well the correlations between the main system's variables. Two aspects of the simulations are, however, not represented by the model. First, the probabilistic model assumes a uniform coverage of the space, where all points of the space are visited with the same frequency; this does not take into account the exploration strategy and the boundary effects due to the walls which make the center of the arena (i.e. the database robot's location) an area more often

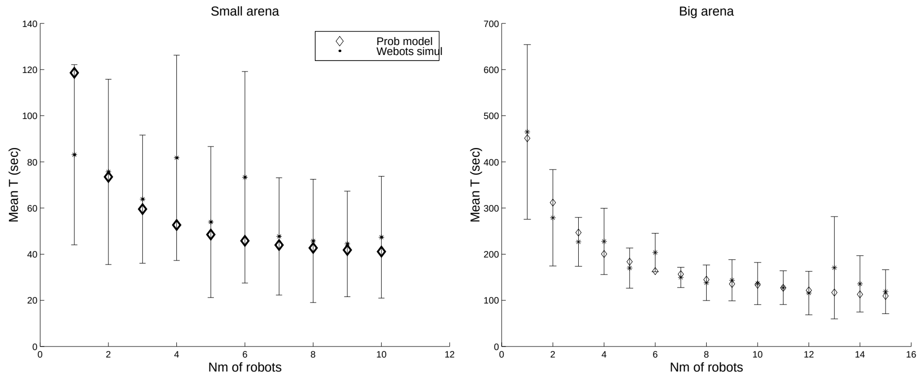

Fig. 4. The Y-axis represents the mean (over 10 runs) time delay T of the database robot to learn all object locations. The X-axis is the number of robots. Each figure compares the prediction of the probabilistic model (diamonds points) and of the Webots simulations ('$*$' point with error bars) in the small arena (left) and in the big arena (right). Error bars correspond to standard deviations.

visited than the exterior of the arena. In order to represent this effect in the model, we increased P_{db} by 20% compared to its real geometrical value so that we obtained the same probability of meeting the database robot as that measured in the simulations. Second, the probabilistic model assumes that learning of an object's location is perfect, i.e. when a robot sees the object, it learns its location without the imprecisions due to odometry errors[3].

The probabilistic model allows one to determine the optimal efficiency of the system in the ideal case; for instance, it can be used to estimate the minimal number of robots, as well as the minimal battery life time they should be provided with for the robots to collect a given number of informations from an environment of a given size.

4.2 Learning in a dynamic environment

Simulations were carried out in a dynamic environment, in which the objects changed locations periodically. For each arena, 15 different periods P of environmental change were tested, 8,16,24,...,120 seconds in the small arena and 28,56,...,420 seconds in the large arena[4]. In each case, 3 runs were carried out for a duration corresponding to $5000 \cdot P$. We ran simulations with groups of 1, 3 and 5 robots in the small arena and 5, 10 and 15 robots in the big arena. Figures 5 top left and 5 top right show the mean number of correctly and incorrectly

---

[3] The odometry errors are in fact negligible given the resetting strategy, see section 2.1.

[4] The minimal steps of 8 and 28 seconds were chosen because it was estimated to be the minimal time to finding an object; it was calculated as the mean distance between the objects divided by the distance traveled by a robot in one time step.

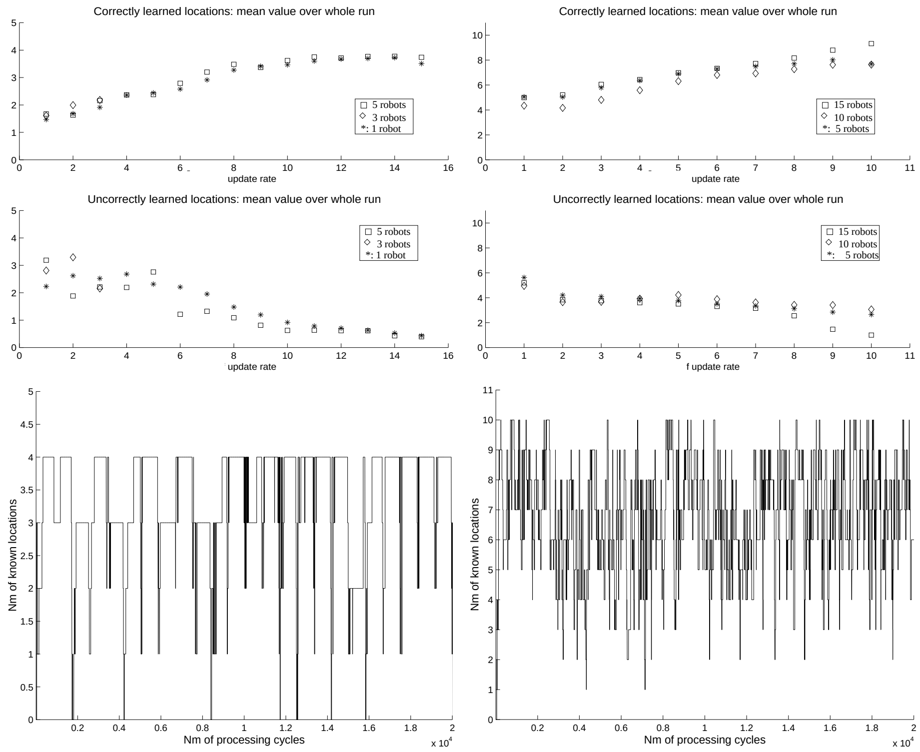

Fig. 5. **Top:** Mean number of correctly and incorrectly learned locations over the whole run for small (left) and big arenas (right); superposition of the results for three robots' configurations (left: 1,3,5 robots; right: 5,10,15 robots). X-axis is the update rate equal to $P/5$. **Bottom** State of the database's knowledge (number of known locations) along a run. 5 robots configuration in small arena (left) and 10 robots configuration in big arena (right). 1 processing cycle is 0.05 second.

learned locations over the whole run for the small and big arenas respectively. The results for each three configuration of robots are superimposed. For P less than 40sec. (small) and 140sec. (big), the database knows on average about 50% of the correct locations, while still taking for correct almost 50% of the locations which are no longer valid. For those periods, the environment changes faster than the minimal time delay T required for the robots to learn all the locations. The minimal T was measured in the simulations of section 4.1 (see figure 4) as a minimum of 40 and 120 seconds for small and big arenas respectively (the measures were consistent with the probabilistic predictions). For P greater than 40 seconds (small arena) and 140 seconds (big arena), the proportion of correctly learned locations increases steadily while the proportion of incorrectly learned locations decreases by the same proportion. There is almost no difference between the three different robot configurations in each case. This is due to the fact that the minimal time delay for finding all objects is almost the same for these three robot configurations (see figure 4)). Thus, it appears that there is almost no benefit in using 15 rather than 5 robots in the big arena and 5 robots rather than 1 in the small arena. However, figure 4 shows that it is more efficient to use 5 robots rather than 1 in the big arena.

Figures 5 bottom shows the progression of the learning of the database robot along a run (results of the simulation in small (left) and big (right) arenas with 5 and 10 robots respectively for $P = 40 and 140$). The curve varies from zero (no locations known yet) to the maximum (4 and 10 correctly known locations in small and big arenas). In the simulations, the objects are not displaced simultaneously. The period at which each object is displaced is constant (it is P) but the phase at which the first displacement occurs is different for each object. This explains the fact that the learning curve does not always decrease until zero (the new locations being discovered and transmitted before all locations have been changed). These results demonstrate that a multiple robots system based on an associative memory learning algorithm, as described in section 2.3, is successful at learning the topography of an environment and updating its knowledge when the environment constantly changes.

5 Experiments with global communication

In this section, experiments are carried out with robots communicating globally in a square arena (see figure 2) with four objects. Similarly to experiments of section 4, we first carried out a set of simulations studies and physical experiments in a static environment in order to determine the minimal time delay for learning the four locations, and then implemented the experiment in a dynamic environment.

5.1 Learning in a static environment

40 runs were carried out in simulation and then in a physical set-up using groups of 1,2,3 and 4 robots (10 runs for each robot configuration, with different random seeds in the simulations and different starting positions for the robots in the physical experiments). Figure 6 shows superimposed the mean time delay for learning all the four locations.

We observe a good agreement between the three plots, with the different implementations (physical set-up, Webots simulations and probabilistic model) showing the same behavior qualitatively. As expected, the mean value of minimal time T decreases with an increase of the number of robots. The standard deviations also decrease with an increase of the robots' number. This means that the learning redundancy due to having more robots reduces the influence of the randomness of the robots' trajectories on the learning success.

Note that, the probability of seeing a source P_s had to be increased by 20% compared to its original value in order to get a good quantitative fit of the probabilistic model (if the probability P_s is not increased, the probabilistic model gives an estimation of T 20% bigger than the measured one). The value of P_s

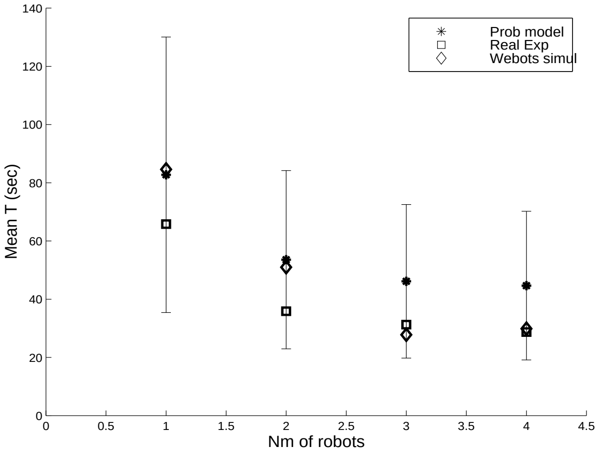

Fig. 6. The Y-axis represents the mean (over 10 runs) time delay T and the X-axis is the number of robots. The prediction of the probabilistic model is compared to the results of the Webots simulations and the physical experiments.

corresponds then to that measured in the simulations. Similarly to the experiments with local communication, the increase of P_s is probably necessary to represent the boundary effects due to the walls (i.e. a reduction of the visited surface of the arena A, and therefore an increase of $P_s = S_s/A$), which are significant here because of the reduced arena size.

The good correspondence between the results of the Webots simulations and those of the physical experiments further confirm the quality of that simulator (it had already been demonstrated in a multiple robots clustering task [11]). It is however important to stress the fact that the physical experiments still used a number of artifacts, such as a virtual definition of the object locations and the use of an external positioning system instead of odometry calculation (as was done in the simulations). Further experiments using robots relying only on physical sensors should be carried out in order to validate the learning system for real world applications.

5.2 Learning in a dynamic environment

12 (4 times 3) runs were carried out in the physical set-up of Khepera robot, using groups of 1,2,3 and 4 robots respectively with 3 different periods of environmental changes (50, 100 and 200 seconds). A run lasted for 10 environmental changes. Figure 7 shows the learning performance (mean number of correctly and incorrectly learned locations over the whole run) of the 4 groups of robots for the 3 update periods. The plotted data are the locations recorded by the database robot (here the stand-alone radio station connected to a workstation) following the broadcast of the worker robots (see explanations of section 2.2).

The learning performances of the four different groups of robots are qualitatively and quantitatively similar; the correctedness of the learning improves when the period of environmental changes increases (leaving more time for the robots to discover all sources). The more robots, the better the learning on average, i.e. the better the ratio between percentage of learning success (figure 7 top) and learning failure (figure 7 bottom). However, the gain in using more robots is not important (the standard deviations of the four curves superimpose[5]), as this was shown previously in figure 6.

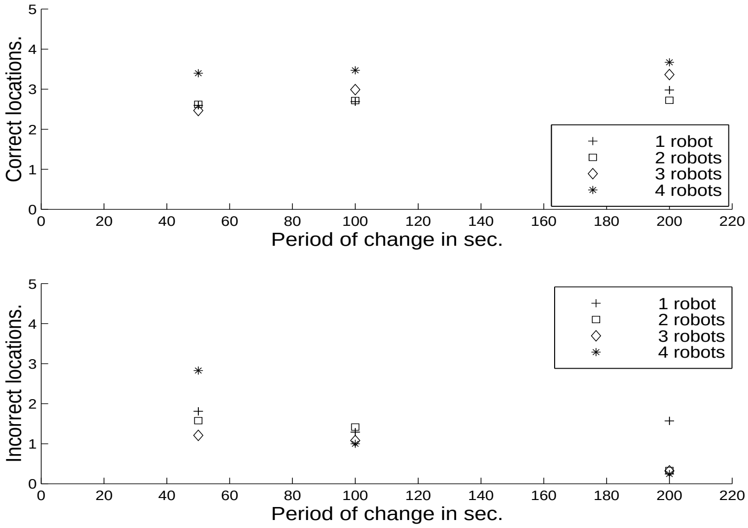

Fig. 7. Mean number of correctly and incorrectly learned locations over the whole run; superposition of the results for experiments with 1,2,3 and 4 robots.

Figure 8 shows the number of correctly learned locations (as recorded by the workstation) along a run for the four different robot configurations. In all graphs, the period of change was 100 seconds, which is bigger or equal to the average minimal time required for learning all locations (as shown in figure 6). We observe that the fluctuations of the learning decreases as the number of robots increases. Similarly to section 5.1, the redundancy in the learning due to using more robots reduces the variability of the results linked with the randomness of each robot's trajectory.

---

[5] We did not plot the error bar of the graph in figure 7 for clarity reasons.

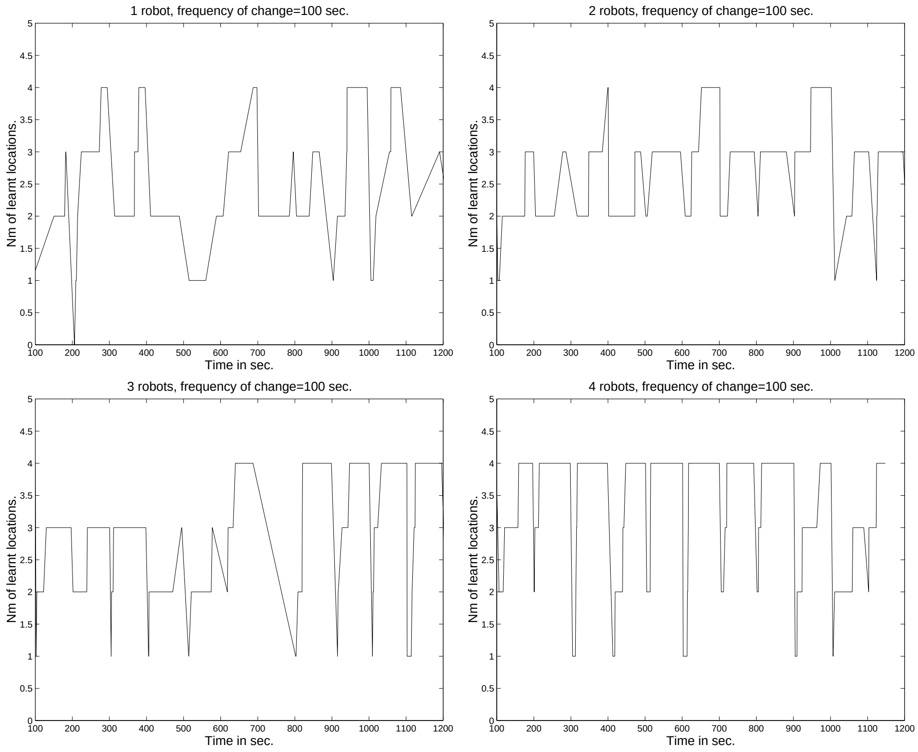

Fig. 8. State of the robots' knowledge (number of known locations) along a run. From left to right, top to bottom: experiments with 1,2,3 and 4 robots when the object locations change with a period of 100 seconds.

6 Conclusion

This paper presented a multi-robot system capable of learning the topography of an environment whose features changed regularly. A learning algorithm was proposed, composed of learning and forgetting processes. It was implemented in simulation with 1 to 15 robots and in a real set-up of 1 to 4 Khepera robots. Results showed that the multi-robot system was able of keeping an up-to-date account of the environmental state when this changes regularly. A probabilistic model was developed to represent the system's dynamics. The probabilistic equations give an explicit description of the correlations between the different variables of the system. It was used to predict the minimal time delay for correct learning in different configurations of robots and environments. The prediction of the probabilistic model were shown to agree qualitatively and quantitatively to the results of simulated and physical experiments, demonstrating that the probabilistic model is a good first approximation of this multi-robot system.

7 Acknowledgement

Many thanks to Luca Gambardella and anonymous reviewers for useful comments which helped improving the writing of this paper. Lots of thanks to the LAMI, EPFL which provided the facilities and technical support for these experiments. This research was supported by a grant of the Swiss National Research Foundation, for the project "A methodology for collective robotics design".

References

1. T. Balch & R. C. Arkin, (1994), 'Communication in reactive multiagent robotic systems', *Autonomous robots Journal*, 1, pp. 27-52.
2. Billard A. & Dautenhahn K., (1999), 'Experiments in social robotics: grounding and use of communication in autonomous agents', accepted for publication in *Adaptive Behavior* (August 1998).
3. Billard, A. and Hayes, G., 1999, "DRAMA, a connectionist architecture for control and learning in autonomous robots", *Adaptive Behavior*, vol. 7:1.
4. W. W. Cohen, (1996), 'Adaptive mapping and navigation by teams of simple robots', *Robotics and autonomous systems*, 18, pp. 411-434.
5. E. Bonabeau, G. Theraulaz & J-L Deneubourg, (1998), 'Fixed response thresholds and the regulation of division of labor in insect societies', *Bulletin of Mathematical Biology*, **60**, pp. 753-807, Article No. bu980041.
6. G. Dudeck, M. Jenkin, E. Milios and D. Wilkes, (1996), 'A taxonomy for multi-agent robotics', *Autonomous Robots*, 3, pp.375-397.
7. Floreano, D., (1998), 'Evolutionary Robotics in Artificial Life and Behavior Engineering', in T. Gomi (Ed.), Evolutionary Robotics, Ontario (Canada): AAI Books, 1998.
8. M. J Mataric, (1998), 'Using Communication to Reduce Locality in Distributed Multi-Agent Learning', *Journal of Experimental and Theoretical Artificial Intelligence*, special issue on Learning in DAI Systems, Gerhard Weiss, ed., **10(3)**, Jul-Sep, pp. 357-369.
9. M. J Mataric, (1995), 'Issues and Approaches in the Design of Collective Autonomous Agents', *Robotics and Autonomous Systems*, **16(2-4)**, Dec. 1995, pp. 321-331.
10. A. Martinoli, "Swarm Intelligence in Autonomous Collective Robotics: From Tools to the Analysis and Synthesis of Distributed Control Strategies", PhD Thesis, Computer Science Department, EPFL, Lausanne, oct. 1999. To appear.
11. A. Martinoli, A. Ijspeert, F. Mondada, (1999), 'Understanding collective aggregation mechanisms: from probabilistic modelling to experiments with real robots', *Robotics and Autonomous Systems*, Elsevier (to appear).
12. A. Martinoli, E. Franzi, and O. Mattey. "Towards a reliable set-up for bio-inspired collective experiments with real robots." In A. Casals and A. T. de Almeida, editors, *Proc. of the Fifth International Symposium on Experimental Robotics ISER-97*, pages 597-608, Barcelona, Spain, June 1997. Springer-Verlag.
13. O. Michel, (1998), 'Webots: a Powerful Realistic Mobile Robots Simulator', Proceeding of the Second International Workshop on RoboCup, LNAI, Springer-Verlag. http://www.cyberbotics.com/
14. A. Murciano & J. del R. Millan, (1997), 'Learning signalling behaviour and specialisation in Cooperative agents', *Adaptive Behavior*, **5:1**, pp. 5-28.

15. F. Mondada, E. Franzi & P. Ienne, (1993), 'Mobile Robot Miniaturisation: a Tool for Investigation in Control Algorithms', Proceedings of ISER'93, Kyoto, Japan, October 1993. Also at http://diwww.epfl.ch/lami/robots/K-family/.
16. L. E. Parker, (1994), 'An Experiment in Mobile Robotic Cooperation', Proceedings of the ASCE Speciality Conference on Robotics for Challenging Environments, February, pp. 131-139.
17. L. E. Parker, (1994), 'Heterogeneous Multi-Robot Cooperation', PhD-thesis, Massachusetts Institute of Technology, January, 1994.
18. Pfeifer, R. and Scheier, C., (1998), 'Sensory-motor coordination: the metaphor and beyond', R. Pfeifer, R. Brooks (eds.), *Robotics and Autonomous Systems*, special issue on practice and future of autonomous agents, **20:2-4**.
19. Premvuti, S. and Wang, J., "A medium access protocol, supporting wireless inter-robot communication in distributed robotic systems", proceedings of IEEE 1994, pp. 1050-4729.
20. I. M. Rekleitis, G. Dudeck, E. E. Milios, "Multi-robot exploration of an unknown environment, efficiently reducing the odometry error", Proc. of Int. Joint Conf. on Artificial Intelligence, Nagoya, Japan, August 1997. Morgan Kaufmann, pp. 1340-1345.
21. Thrun, S., (1996), 'Explanation-Based Neural Network Learning - A Lifelong Learning Approach', Kluwer Academic, Boston, MA.
22. E. Yoshida, T. Arai, M. Yamamoto and J. Ota, (1998), 'Local communication of multiple robots: Design of optimal communication area for cooperative tasks',*Journal of Robotics Systems* 15(7), pp. 407-419.