An application concept of an underwater robot society

Mika Vainio, Aarne Halme, Pekka Appelqvist, Pekka Kähkönen,
Peter Jakubik, Torsten Schönberg, Yan Wang

Automation Technology Laboratory, Helsinki University of Technology,
Otakaari 5A 02150 Espoo, FINLAND

Abstract

Cooperation as a mean to adapt to dynamic environments is well known in animal world from the social insects up to primates. But similar behavior seems to exist also in simple unicellular microbes. In this paper a novel approach for a very application oriented multi-agent system is taken. The principles of this robot society are derived from bacteria, which are here considered as multicellurar organisms. The analogy from Nature includes bacteria's ability to communicate through chemical substances, to form a colony (a society) and to act as a predator hunting for food. The concept is tested in simulations, where the behavior of the society is used to demonstrate how the extensive use of chemical in a closed water circuit for algae removal could be minimized with a collective cooperation of mobile underwater robots. Additionally, some early tests with the first generation society member will be shown in order to validate some of the simulation results including a simple topological mapping and navigation method.

Keywords: underwater autonomous mobile robots, robot society, collective intelligence, group behavior

1. Introduction

One of the main streams in today's advanced robotics research is clearly going towards replicating the structures and behaviors of highly intelligent organisms. The scaling up from more or less toy problems to more serious ones has always been very difficult; a sufficient redundancy is quite hard to build when the working environment starts to be real, i.e. really complex. In several studies, including e.g. [1] and [2], the source of inspiration has been found from social insects. The multi-agent based approach originated from Nature, as a solution for the redundancy problem has gained a lot of interest lately, and various research groups all over the world are working with the concept. One of the earliest papers [3] presented quite revolutionary ideas for the use of these systems. [4], [5] and [6] illustrate how cooperation can emerge from simple interactions between agents and the environment. In [7], [8], [9], and [10] the "social" relationships in multi-robot systems were studied. In Japan the research around the concept Cellular Robotics has been especially vivid (see, e.g. [11] and [12]) under a larger concept called Distributed Autonomous Robotic Systems (see, [13] and [14]).

In process industry the problems about how to monitor the internal state of the process in real-time and how to make local adjustments in reaction conditions are still mainly unsolved. Normally the sensors used in monitoring are fixed and thus will only provide information from certain selected parts of the process. Local adjustments have been hard to implement, if not totally impossible, and thus the control has been based more or less on the overall control of the system. This has led e.g. to an extensive use of chemicals, which is both expensive and causes often unwanted residuals and pollution. A solution for these two problems could be a robotic sensor/actuator society, which is capable of operating inside the process as a kind of local actuator or as a "precision weapon". The principles of this new distributed autonomous robotic system are derived from bacteria, which have been only recently considered as multicellurar organisms. The analogy from Nature includes bacteria's ability to communicate through chemical messages, to form a colony (a society) and to act as predator hunting for food. These behaviors will be implemented to a physical robot society operating inside a 3D demo process. The adaptation of the robot society is based on the learning the topology of the environment along with the ability to operate together.

The concept is tested in simulations, where these behaviors will be used to demonstrate how the extensive chemical use for algae removal in a closed water circuit, for example in a paper machine, could be replaced with a kind of precision weapon policy of the robot society. In the implementation phase of this project a modified robot society will be used in a pulp plant, where it first provides valuable distributed 3D sensory information from the bleaching process. Later on more active operation of the society will be addressed (i.e. the algae removal task).

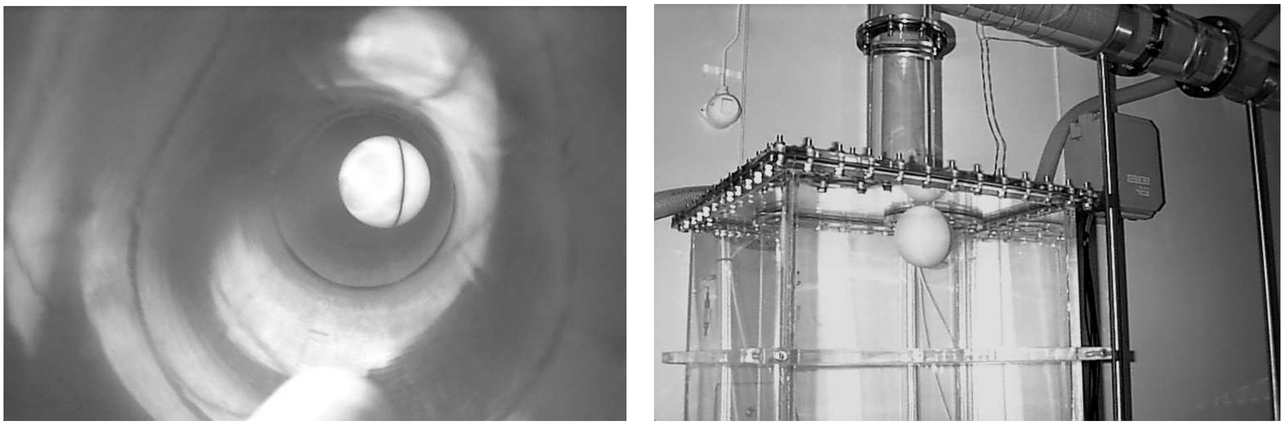


Fig. 1a - 1b. Society member inside the test environment.

2. The interesting world of bacteria

Until recently bacteria were more or less considered to be really simple unicellular microbes, even though there have been studies proofing opposite already at the beginning of this century. It took decades before the prejudices were won, and the idea of bacteria as multicellular organisms with abilities to form communities, hunt in groups and even communicate with each others were accepted [15]. All these features were before considered to exists only in higher level organisms. Some of those features are actually quite astonishing. Myxobacteria have been commonly used as an example illustrating the aggregation and motion behaviors among these small creatures. Time-lapse motion pictures where thousands of bacteria are moving through extracellular slime extracted by them in a very harmonic way, will convince even the most skeptical viewer, that there actually exists social relationships among the colony members. [16] show that Escheria coli bacteria are able to aggregate to stable patterns with remarkable regularity. This self-organization is mainly caused by a one sort of chemotaxis, where the bacteria are moving along the gradients of a chemical attractant that they are self excreting. Another fascinating feature can be found from some predatory species living in an aquatic environment. These bacteria have an elegant way to hunt and eat in complex environments. The colonies are secreting enzymes that are dissolving the protective outer layer of the prey microbes. The problem to be solved in order to guarantee successful eating, is to make sure that the enzymes will have time to effect on the prey colonies. The predator colony uses a following strategy: it surrounds the prey (actually it takes the prey inside a kind of pocket), and then it will emit the enzymes. This will provide a maximum effect and the predator can use the nutrition of the prey colony without losing valuable energy to the surrounding environment.

3. The bacterium robot society

Societies are formed as collaborative structures to execute tasks that are not possible or are otherwise difficult for individuals alone. There are many types of biological societies (e.g. ants, termites, bees) but societies formed by machines or robots are still rare. The concept offers, however, interesting possibilities especially in applications where a long term fully autonomous operation is needed and/or the work to be done can be executed in a parallel way by a group of individuals [8]. The practical goal of the robot society concept is to construct a kind of “distributed robot” or “group robot”, which can execute tasks which are defined by the user or the “society controller", like in the case of a conventional industrial robot. This means that the behaviour of the society must be, to some extent, controllable from outside and that the society must thus have information connection to the controller. However, it is important from a practical point of view that this connection is not built to every member of the society, but rather to its information system. This is because a society may include a large number of members that are located in places where a communication system is difficult to build. Basically the communication in a society is done on member to member bases or in particular cases only with indirect methods, i.e. through changing the environment. The main society features are emerging system and self-organization. In an emerging system we are just dealing with the design on the member level. Without specifying any task function a meaningful operation on the society level should be observable. In a self-organizing system the society mission is achieved through an algorithmic realization of searching “functional neighbours” (e.g. task sequences, topological neighbours, information sources). Inspired by the multicellular behavior of bacteria we started to implement a robot society, which would both be an industrially oriented as well as biologically analogous to some of the fascinating features of bacteria. The society is going to be implemented in two separate phases: first phase was about building a real society of robots operating inside a demo process build in our premises, and then a modified robot society will be installed to a pulp mill.

3.1 Society member prototypes

We developed a ball-shaped autonomous underwater robot, SUBMAR (Smart Underwater Ball for Measurement and Actuating Routines), for a society member prototype. For more detailed presentation see [17]. The diameter of the robot is 11 cm and the casing is made out of plastic. The *first generation* prototype has a transputer as processor, pressure and temperature sensors and a diving tank for vertical movements. The *second generation* robot is also based on a transputer, but it has another tank for reagent transportation and a conductivity sensor. The *third generation* robot, which is now in testing phase, uses a 16-bit microcontroller with flash memory and an inductive communication unit. The robots are shown in Fig. 2. More precise description of these robot versions can be found from [18]. The following generations of robots, intended to be used in particular industrial environments, will naturally be tailored for each case separately. Harsh environmental conditions will require a lot from the casing and the components. The miniaturization technics and bio-sensor technology will provide sensors and actuators enabling large societies with really small robots in the near future.

3.2 The demo process

The first place to test robot prototypes and the concept as a whole, is a simple process environment built in our laboratory. This demo process contains several types and shapes of process parts (Fig. 3).

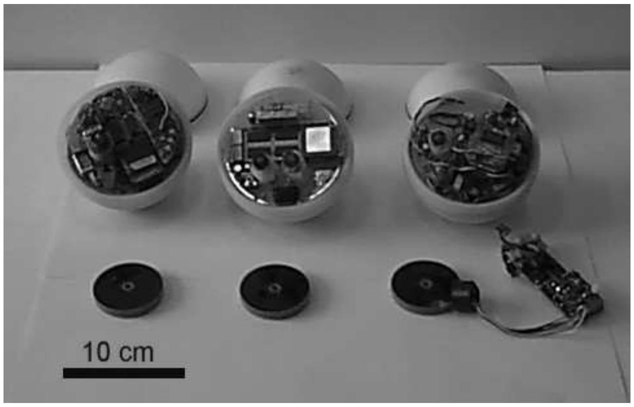


Fig 2. Three generation of SUBMARs.

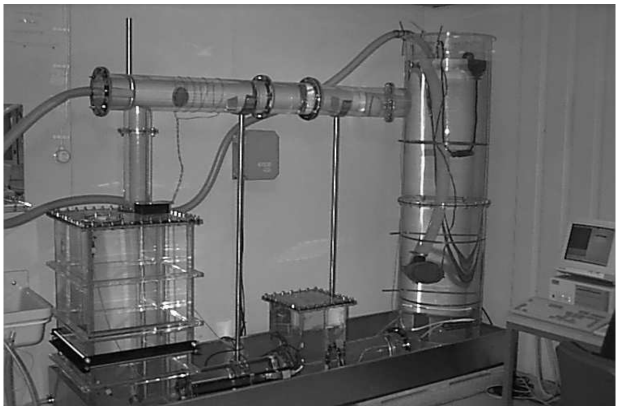
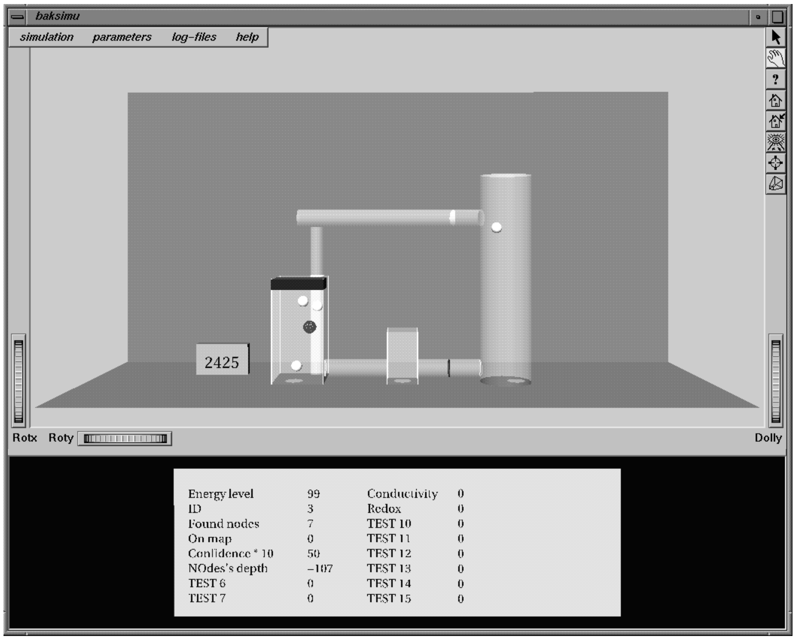


Fig. 3. The demo process

A pump is used to provide water circulation, a large resistor is installed for warming up the water, an ultrasonic flow meter and several temperature sensors are connected to a digital automation system. The whole process is made out of transparent plastic with a volume of 700 liters. The operator-society connection will be done, when needed, through a tag memory interface, i.e. in robot there is a special read/write unit, which can be used to interface an onboard tag memory. Similar read/write units are located around the process enabling messages to and from the operator.

3.3 Society simulator

In order to validate the behavior of a robot society as a solution for a complex real world problem, before we could produce ten(s) of real robots for the bacterium society, a society simulator was needed. The simulator was coded with Open GL in Silicon Graphics Indigo2 to represent the 3D world with complex dynamics (see Fig. 4). The extensive calculation makes on-line simulation quite slow. However when the graphics updating is off, the speed will be much faster and the simulator can provide the needed statistical data. Basic flow dynamics are combined to the features extracted from the real process. The other features of the simulation like robots' control structures, collision handling and diffusion of the poison are based on the testing done with real a robot. The diffusion model is a modified 3D version of 2D model presented in [19]. Additionally the flow vectors have their own impact to the diffusion rate and to its direction. The operator can adjust various parameters such as the communication range, algae growth, number of members, etc. On the lower window is shown the contents of a selected member's tag memory. This window will illustrate, for example what is the energy level of the robot. It indicates also how the member learns its environment in a form a topological map (presented later in chapter 6.). A possibility to create log files was also included to the simulator.



```
Energy level        99      Conductivity   0
ID                  3       Redox          0
Found nodes         7       TEST 10        0
On map              0       TEST 11        0
Confidence * 10     50      TEST 12        0
NOdes's depth       –107    TEST 13        0
TEST 6              0       TEST 14        0
TEST 7              0       TEST 15        0
```

Fig. 4. The outlook of a 3D society simulator.

3.4 Industrial applications

In order to avoid the scale-up problem, the concept was developed from the beginning keeping industrial applications in mind. The system is first going to be used as a distributed measurement system in a pulp bleach process. The process consists of series of high bleaching towers with long retention times. To be able to control the quality of the batch, the operators have to monitor what is going on inside these huge towers. In first phase the society will be injected into a tower from the bottom. Inside the tower there is a massflow going upwards. The normal practice is to take samples from the top of the towers. Samples are then analyzed off-line in a laboratory. With the mobile sensor society, the delay will be much shorter; the operators will get a detailed picture of what is happening inside the first tower immediately when the members emerge from the tower. In future it is possible that a communication system will be built inside the tower. In this way the operators will receive on-line 3D information from the process. Other future application will possibly be process inspection and cleaning. In large water storage, like in closed water circuit inside a paper machine, biological growth causes problems to water quality. The developed society could detect and remove unwanted algae growth with relatively small use of chemicals. The cost savings in both cases will be really substantial. In the following we consider the principles of such society.

4. Collective cooperation

In [20] some biologically inspired experiments were conducted in order to study collective and cooperative group behaviors. The authors made a distinction between a collective noncooperative behavior and a collective cooperative behavior. The former is just a case where greater number of agents will accomplish the task faster than fewer (usually up to some limit) and the latter is the actual case where some kind of cooperation between the agents is needed before the task can be finished. Usually cooperation in multi-agent systems is linked to direct communication but the cooperation can also be based on an indirect communication. This indirect communication takes place, when the agents are communicating only by changing and detecting the changes in their environment. In animal world ants communicate through chemical messages (as well as through tactile messages), and bacteria emit bacteriocin to provide some kind of information for the other members in their colony. [21] pointed out that a concept stigmergy, “the production of a certain behavior in agents as a consequence of the effects produced in the local environment by previous behavior”, is not strictly restricted to building structures, but it can also be found for example in ants trail requirements. Thus in many case stigmergy and indirect communication can be considered to represent the same phenomena.

4.1 The task domain

The indirect communication/stigmergy is also used in our case. It is implemented in an aquatic environment’s difficult conditions with chemical messages. The idea is to use the society as a distributed sensory system, which will search the environment looking for algae growth. When a growth has been detected by a single member (Fig. 5a), it starts to recruit the others by using chemical messages released to the liquid (Fig. 5b). After a while a cluster will be formed and the society will act cooperatively as a precision weapon (Fig. 5c) and remove algae growth locally with a very small amount of chemicals (Fig. 5d).

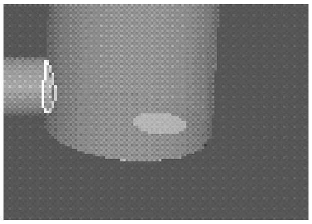


Fig. 5a. Algae growth

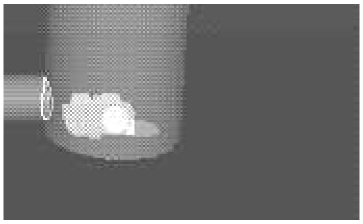


Fig. 5b. Recruitment

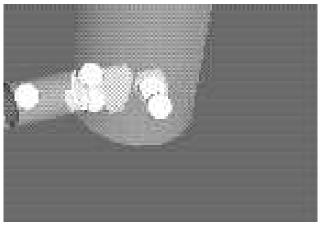


Fig. 5c. Cooperation

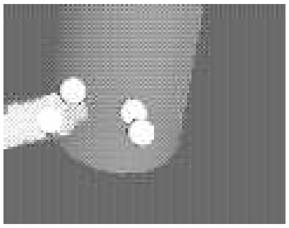


Fig. 5d. Job done

The problems in the scenario presented above are mainly caused by the poor mobility of the members and by the dynamic environment (i.e. the process flows). If a member after detecting the growth tries immediately to land on the bacterial growth, it will most likely fail and lands on a wrong place. The only possible solution is to provide the members a way to build some sort of primitive map of the environment and thus make simple planning possible.

5. Member's behavioral repertoire

Both the real and simulated robots use behavior-based control structures (see e.g. [22], [23] and [24] for overview). Used structures have similarities with both subsumption [25] and with Arkin's schema-based ([26] and [27] systems. Members have a special sensor block (i.e. *perception -behavior*), which takes care of the interface to the sensors. The used sensors include: pressure, redox, conductivity and energy status sensors. On the other hand we have implemented an another hardware related block (i.e. an *action* -behavior), which takes care of the control of the two motors (i.e. the diving tank and the poison tank), which is based on the information it receives from the parallel behavior blocks. These behaviors, which are presented later on, include: *stagnation.recovery, self-sufficiency, make.map, altruistic* and *remove.algae*.

6. Topological mapping

The members of the society, with a very limited ability to move actively, have to have some kind of spatial representation model of their environment, e.g. a topological map, in order to work efficiently together as a society. In a complex underwater environment any absolute localization method would be difficult to construct. Fortunately, the concept of robot society provides a way to solve this problem. An individual member's map need not to be highly accurate. It is enough that different process parts can be clearly recognized. The cooperation between society members will then make their maps more detailed. Needless to say, an adaptive element must be included to the used mapping. Mataric [28] pointed out, that even though fixing and finding of landmarks is usually based on vision, this isn't obligatory. Animals are known to use various of types of landmarks, including tactile and auditory. Mataric presented a neurobiologially-feasible cognitive mapping, implemented with an autonomous mobile robot, where landmarks were defined as combinations of the robot's motion and it's sensory input. The map produced by the robot contained nodes (i.e. landmarks) and topological links between different nodes, which indicate their spatial adjacency. We chose to represent the environment with a directed graph, where a node in the graph represents an event (or a landmark) and an edge stands for transit from one event to an another.

6.1 What is needed?

As an essential part of this mapping is the behavior, which prevent deadlocks. In a process environment, the deadlocks are usually related to two types of events. The robot can get stuck to the bottom/ceiling of some tank or it can end up to a strong turbulence. To cope with these problems, the robot was equipped with a so called *stagnation.recovery* -behavior. This behavior will start the whole mission after the members are put inside the process (into a tank with an open ceiling). A member stay on the surface for a while, put after N measurements (usually N = 20) with same pressure values, the *stagnation.recovery* -behavior will become active and the member will take some water inside to its tank and it dives. On the other hand, if the member is stucked to the bottom of a tank, where the currents are slow, this behavior will inject some water out of the diving tank and the robot will rise back to the current. To survive from the latter problem (i.e. from strong turbulence), the behavior just changes robot's specific weight time to time (after 10*N measurements). This behavior, designed to recover from a stagnation, is actually also increasing the searching power of the robot, and thus making it possible for the robot to explore the environment properly. While it isn't active, the robot is moving along the main process currents, but when it becomes active, it will enable the discovery of new areas outside the main flow (i.e. the ceilings and the bottoms of various tanks and areas behind some turbulence).

6.2 The mapping algorithm (i.e. make.map -behavior)

The mapping is based on only one sensor measurement, in this case a pressure sensor. These are used to detect events, where the motion character changes. In this case it means, that the movement changes from going down to halt, from halt to going down, from going up to halt and from halt to going up. The amount of data about the environment obtained by this way is very limited, and we have to accept the fact that some errors and overlapping take place. To be able to deal with this fact, the mapping has to have some kind of adaptation capability. In [29] a concept called APN (Adaptive Place Network) was presented. It provided a spatial representation and a learning capability to a mobile autonomous robot. Even though the APN uses quite a lot of information, which is difficult or in some cases impossible to obtain in our 3D environment, we implemented a modified version of the APN in our bacterium robot society. This principle is shortly described in the following.

When a new link is created it is also labeled with a confidence value $c \in [0,1]$. This value will basically estimate how real the created link is, i.e. does it really exists or is it just some kind of sensor error or due to some collision between the society members. At the beginning of the lifespan of a link the value c is set to c_{birth}. If the robot has traveled through a certain link then the value of this link is increased with the following equation:

$$c_{t+1} = \lambda + (1-\lambda)*c_t \qquad (1)$$

where λ is the learning rate. After N_{nodes} nodes have been detected the values of all links are reduced according to the following formula,

$$c_{t+1} = (1-\lambda)*c_t \qquad (2)$$

When this value goes below a threshold named as T_{kill}, the link will disappear from the adjacency matrix. If it happens that, all links that were connecting node to the graph will be deleted, then also the node will be removed from the graph. Thus the size of the matrix won't explode in case of several members are operating in the same working area. For more detailed presentation see [18]. This kind of updating of the links and nodes will provide the robot some kind of primitive way to understand what is going on around it without any active communication. Collisions between society members will create nodes, which will be eventually removed from the graph.

6.3 Preliminary results by using one real robot

The actual algorithm was initially developed with the data obtained from a real robot. The robot was operating inside the process and simultaneously measuring pressure values once in a second. In Fig. 6 is shown the data from a test run.

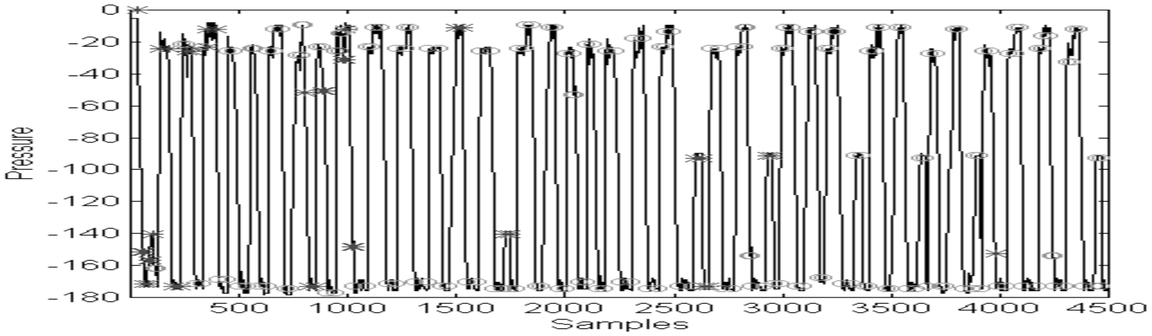


Fig. 6. An example run with following parameters: $\lambda = 0.25$, $N_{nodes} = 20$, $T_{kill} = 0.1$, $c_{birth} = 0.5$. From the pressure values (depth) various parts of the demo process can be recognized. The length of this particular run is about 1 hr. 30 minutes. A new node is always marked with a cross and a circle represents a known node (i.e. the algorithm is on the map).

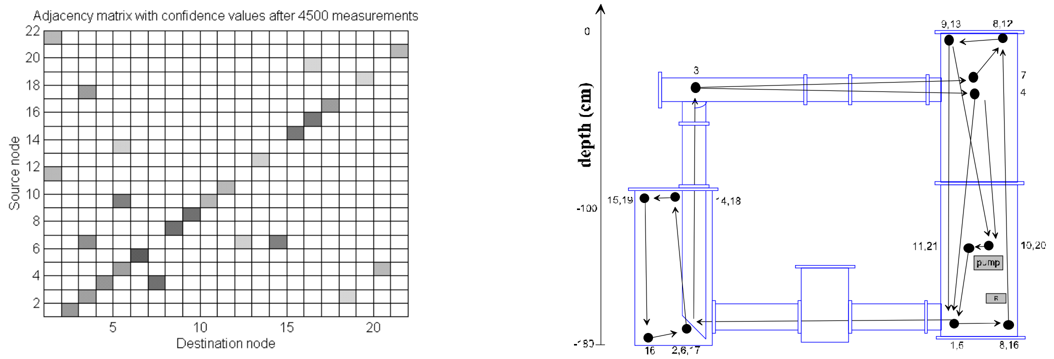


Fig. 7a- 7b. Situation after the run. The adjacency matrix with confidence values is shown left. The darker the square is, the more often SUBMAR has used that link. On the right side this topological map is plotted on top of the process picture. Circles represent nodes (indexes plotted aside), and arrows stand for possible trajectories (i.e. links) between them.

6.4 Preliminary results by using the multi-robot simulator

Along with these initial tests implemented with a real robot, the society simulator was used to study the behavior of the society. The preliminary simulations verified the obvious fact, that the parameters (like learning rate, etc.) in the make.map- algorithm had a very strong effect on the form and size of the map. In the following tests these parameters were fixed and the only variable was the number of members in the society. First we studied how a single agent behaves in a simulator. In Fig 8a the adjacency matrix is shown after 300 node creations. The matrix contains 16 nodes and in Fig. 8b only links with confidence value greater than 0.5 are illustrated. This figure shows how some kind of "basic cycle" is already detectable (i.e. the robot's main route inside the process).

		194	193	7	181	41	41	60	60	7	185	191	155	155	107	107	192	194
0	194	0	43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	193	0	0	0	**68**	**3**	0	0	0	0	0	0	33	0	0	0	0	0
2	7	43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	181	0	0	0	0	0	0	0	0	0	**68**	0	0	0	0	0	0	0
4	41	0	0	0	0	0	**53**	0	0	0	0	0	0	0	0	0	0	0
5	41	0	0	0	0	0	0	41	0	0	0	0	0	0	0	0	0	**58**
6	60	0	0	0	0	0	0	0	41	0	0	0	0	0	0	0	0	0
7	60	0	0	0	0	0	0	0	0	41	0	0	0	0	0	0	0	0
8	7	0	0	41	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	185	0	0	0	0	0	0	0	0	0	0	0	0	0	**68**	0	0	0
10	191	0	35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	155	0	0	0	0	0	0	0	0	0	0	0	0	33	0	0	0	0
12	155	0	0	0	0	0	0	0	0	0	0	33	0	0	0	0	0	0
13	107	0	0	0	0	0	0	0	0	0	0	0	0	0	0	47	0	0
14	107	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	47	0
15	192	0	47	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	194	0	**58**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig 8a. An adjacency matrix for a single agent after 1/2 simulation time. Darkened numbers represents links that are drawn in Fig. 8b.

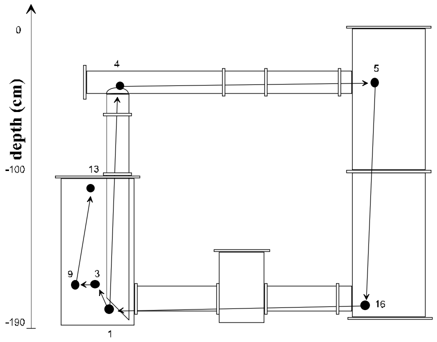


Fig. 8b. Topological map drawn on top of process picture.

When the simulation terminated, the topology of the process was quite well-known to the agent. The size of the adjacency matrix was still the same and the confidence values for the nodes were quite high, indicating thus clearly the existence of the learning capability (Fig. 9a - 9b).

		194	193	7	182	41	41	59	59	7	185	191	156	156	107	107	192	107
0	194	0	**83**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	193	0	0	0	**60**	**81**	0	0	0	0	0	0	35	0	0	0	0	**62**
2	7	**83**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	182	0	0	0	0	0	0	0	0	0	**59**	0	0	0	0	0	0	0
4	41	0	0	0	0	0	**81**	0	0	0	0	0	0	0	0	0	0	0
5	41	0	0	0	0	0	0	**81**	0	0	0	0	0	0	0	0	0	0
6	59	0	0	0	0	0	0	0	**81**	0	0	0	0	0	0	0	0	0
7	59	0	0	0	0	0	0	0	0	**81**	0	0	0	0	0	0	0	0
8	7	0	0	**81**	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	185	0	0	0	0	0	0	0	0	0	0	0	0	0	**59**	0	0	0
10	191	0	37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	156	0	0	0	0	0	0	0	0	0	0	0	0	35	0	0	0	0
12	156	0	0	0	0	0	0	0	0	0	0	35	0	0	0	0	0	0
13	107	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**59**	0	0
14	107	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**78**	0
15	192	0	**78**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	107	0	0	0	0	0	0	0	0	0	0	0	0	0	0	37	0	0

Fig 9a. An adjacency matrix for a single agent after full simulation time.

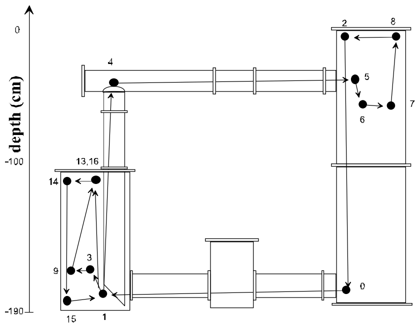


Fig.9b. Topological map drawn on top of the process picture.

Next the number of members was increased to 10. The results are shown below. The size of the adjacency matrix is obviously larger due to the multiple collisions caused by the presence of 10 society members. The confidence values are thus mainly quite low and the topological map is also very inadequate, as is shown in Fig.10a - 10b.

```
194 193 108  41  40  58  58   8 159 159 190   8 192  35  35 194 186 186 140 140 107  30  30
 0 194   0  64   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 1 193   0   0   0  48   0   0   0   0  61   0   0   0   0   0   0   0  30   0   0   0   0   0   0
 2 108   0   0   0   0   0   0   0   0   0   0   0   0  38   0   0   0   0   0   0   0   0   0   0
 3  41   0   0   0   0  48   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 4  40   0   0   0   0   0  41   0   0   0   0   0   0   0   0   0  30   0   0   0   0   0   0   0
 5  58   0   0   0   0   0   0  41   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 6  58   0   0   0   0   0   0   0   0   0   0   0   0   0  40   0   0   0   0   0   0   0   0   0
 7   8  64   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  30   0
 8 159   0   0   0   0   0   0   0   0   0  61   0   0   0   0   0   0   0   0   0   0   0   0   0
 9 159   0   0   0   0   0   0   0   0   0   0  61   0   0   0   0   0   0   0   0   0   0   0   0
10 190   0  36   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
11   8   0   0   0   0   0   0   0  49   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
12 192   0  38   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
13  35   0   0   0   0   0   0   0   0   0   0   0   0   0   0  40   0   0   0   0   0   0   0   0
14  35   0   0   0   0   0   0   0   0   0   0   0  41   0   0   0   0   0   0   0   0   0   0   0
15 194   0  30   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
16 186   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  30   0   0   0   0
17 186   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  30   0   0   0
18 140   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  30   0   0
19 140   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  30   0
20 107   0   0  30   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
21  30   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  30
22  30   0   0   0   0   0   0   0   0   0   0   0  30   0   0   0   0   0   0   0   0   0   0   0
```

Fig 10a. An adjacency matrix for an agent after
half simulation time with 10 member society

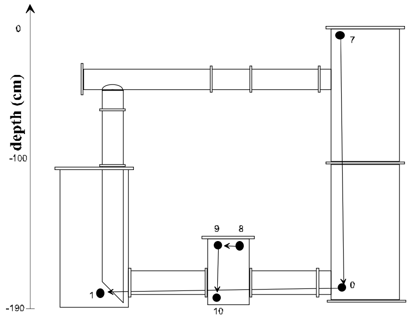


Fig. 10b. Topological map drawn on
top of the process picture.

When the simulation proceeds, the main nodes will become more evident and the topological map will get a familiar form (see Fig. 11a - 11b). Even though the members are colliding a lot, these events are forgotten quite fast.

```
       193  42  42  59  58  14 109 192 194 111 190  20 194 155 155  59  59  76  76  43  43 194
 0 193   0  69   0   0   0   0   0   0   0  57   0   0   0  42   0  30   0   0   0   0   0   0
 1  42   0   0  69   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 2  42   0   0   0  68   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 3  59   0   0   0   0  68   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 4  58   0   0   0   0   0   0   0   0   0   0   0  67   0   0   0   0   0   0   0   0   0   0
 5  14   0   0   0   0   0   0   0   0  66   0   0   0   0   0   0   0   0   0   0  50   0   0
 6 109   0   0   0   0   0   0   0  57   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 7 192  54   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 8 194  66   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 9 111   0   0   0   0   0   0  59   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
10 190  45   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
11  20   0   0   0   0   0  65   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
12 194  33   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
13 155   0   0   0   0   0   0   0   0   0   0   0   0   0   0  42   0   0   0   0   0   0   0
14 155   0   0   0   0   0   0   0   0   0   0  42   0   0   0   0   0   0   0   0   0   0   0
15  59   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  30   0   0   0   0   0
16  59   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  30   0   0   0   0
17  76   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  30   0   0   0
18  76   0   0   0   0   0   0   0   0   0   0   0   0  30   0   0   0   0   0   0   0   0   0
19  43   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  50   0
20  43   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  50
21 194   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
```

Fig 11a. An adjacency matrix for an agent after
full simulation time with 10 member society

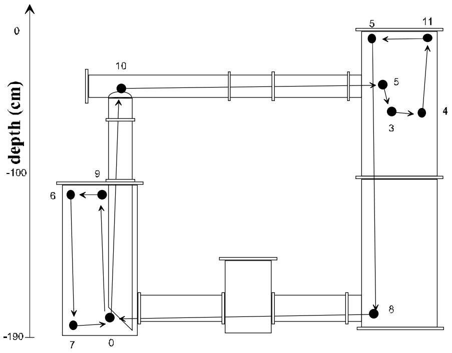


Fig.11b. Topological map drawn on
top of the process picture.

7. Self-sufficiency

To be able to work autonomously the society has to be self-sufficient in terms of energy consumption, i.e. there must be some way for robots to monitor and recharge their batteries during the mission. The aquatic environment with flows will require quite a lot from this kind of behavior, especially when the robots have so limited capabilities to move actively. The robots monitor the status of their batteries. At the same time when the robots are learning the topology of the process, they are looking for the energy refilling station(s). When a robot happens to come a cross a fuel station, it will receive reward in a form of an increased energy level. These kind of places are then naturally mapped, i.e. they are connected to certain nodes in the graph. After this the robot can actively try to reach (i.e. to navigate) the recharging place. So far this behavior is only implemented in simulations, but the design of the real recharging station is going on. The actual recharging will be based on inductive energy transmission between two coils, see Fig. 12. These recharging stations can be used also for unloading samples collected by the society members. The principles of this valuable “foraging” behavior are presented in [18].

```
self-sufficiency
get current.node from make.map
IF
   (energy.status.new > energy.status.old)
      energy.node = current.node
   (energy.status.new  < energy.low)
      IF
         (current.node = energy.node)
            recharge
         ELSE
            ask the closest energy.node
            go to that node
```

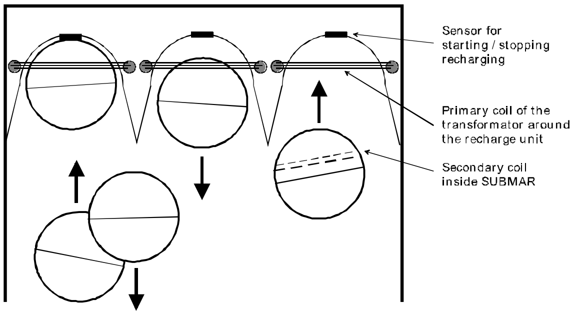


Fig. 12a-12b. The self-sufficiency behavior and the energy station. The energy transmission will be done inductively in each comb separately.

8. Algae removal - Getting the job done

The actual task for the society is to find and to destroy the algae growth that can be found mainly from places, where the flow speed is close to zero. These growths can be found by using a redox sensor in our demo process. The algae growth will "output" oxygen. (In a real industrial process the target must be detected with some other methods.) When a robot detects that dissolved oxygen level is higher than a threshold (T_{redox1}), the robot will adjust its specific weight and it will try to land on the growth (Fig. 13a). When it is on the bottom, and the level is still above the threshold, the robot will open a valve and will start to release a chemical ($KMnO_4$) from the "poison tank" as shown in Fig. 13b. By doing this it will actually start a recruiting of the other members. The rate of the dissolving (c_{unit}) is very small, but big enough to be detected locally by the others with a conductivity sensor. When an another robot detects the presence of this chemical, it will try to go to the same place. Usually the current will prevent a direct landing on the growth. Thus the only solution for a successful landing is to use a topological map in the next circulation. Due to the limited moving capabilities, the navigation strategy is very straightforward: if the target node is located deeper than the node from where there is a link to the target, the robot takes maximum amount of water in, and if the target is closer to the surface the robot will change its specific weight lighter. When the robot reaches the algae growth it will start to release the same chemical with the same rate. And the other robots will do the same. Finally the combined concentration rate will go over a threshold (c_{limit}), and the next robot to join the group will open it's poison valve to it's maximum (c_{max}) as shown in Fig. 13c.

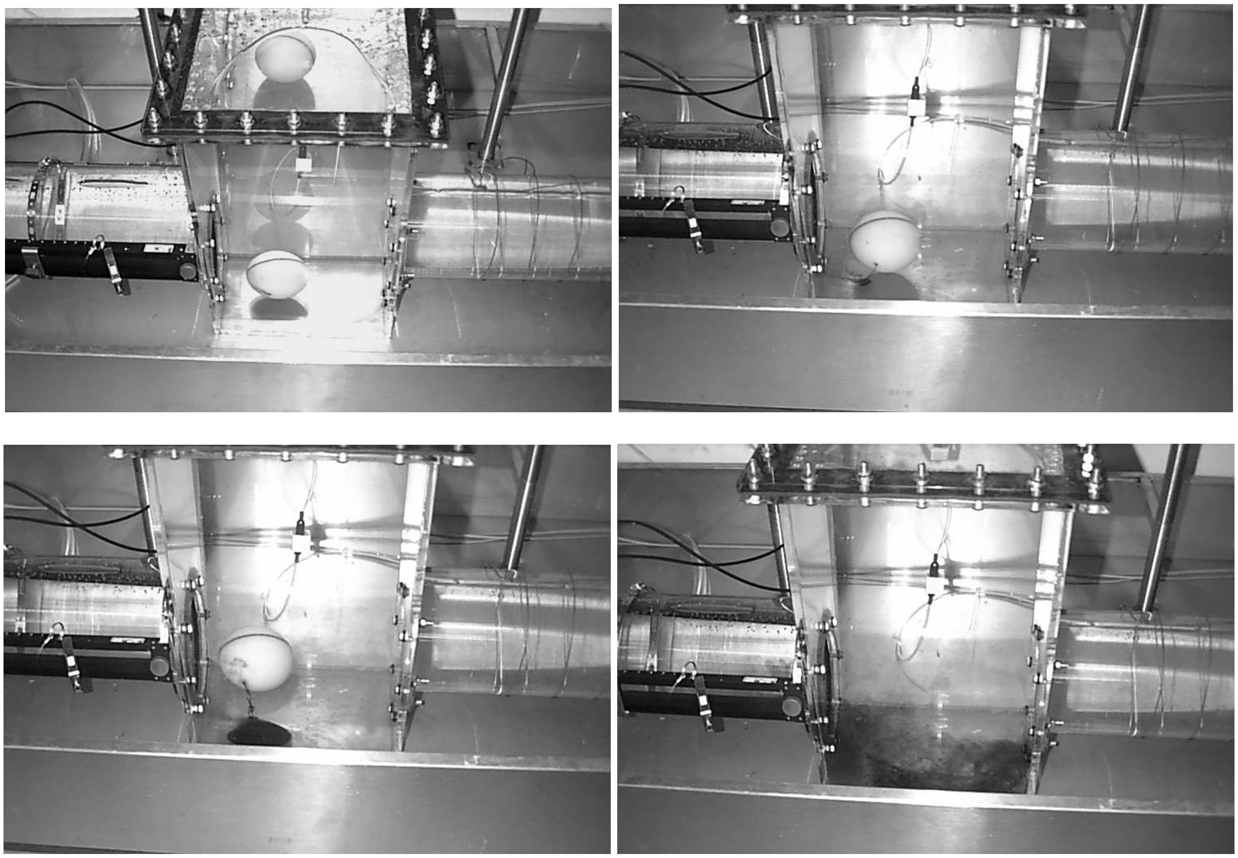


Fig. 13a - 13d. A single society member performing the remove algae -behavior. In Fig. 13a the robot has detected an algae growth and lands on the bottom. In next figure it starts to output chemical in order to recruit other members. Below on the left the member will output the chemical with maximum speed, and in the last picture the robot has already left the tank and the "poison" is slowly dispersing.

This will raise the total concentration above threshold (c_{limit}) also in the cluster. When the other robots detect this they will also release all their chemicals. This will provide the cooperation, and the society is acting as a precision weapon (Fig. 13d). With small amount of poison the society is able to remove algae from a process, thus replacing the normal massive use of chemical. To replace the society with a single more sophisticated robot having a large chemical tank, would also mean that the size of this robot would be unsuitable for process environment (narrow pipelines). Besides, a malfunction in this robot would cause a failure in the mission. The *remove.algae* -behavior is shown below.

```
remove.algae

  get current.node
   IF
     (conductivity < limit.small)                 --no concentration found
        IF
          (dissolved oxygen = TRUE)               --the robot detects an algae growth and starts to recruit
             node.work = current.node             --other members
             go to node.work in next cycle
             release poison (small output)
     (conductivity > limit.large)                 -- high concentration of chemical in the water nearby
        node.work = current.node
        go to node.work in next cycle
        release poison (large output)
     (conductivity > limit.small)                 --small concentration of chemical in the water nearby
        node.work = current.node
        go to node.work in next cycle
        release poison (small output)
```

9. Communication through tag memories

9.1 Fixed memories

The third generation members of the society are equipped with a tag memory and with the ability to read and write these memories. Onboard the robot there is a special read/write -unit, which consists of a simple microcontroller and an antennae. These are used to provide the members a way to communicate. Around the process there are a few communication stations. These stations use coil antennas wrapped around the process pipes. These antennas are connected to read/write -units. When a member wants to send a message to another members of the society, it will leave this message to the communication station. When another member of the society happens to pass the scene, it will receive the message while passing through the antennae loop. The system is more precisely presented in [18]. This kind of communication requires more or less active operation from outside (i.e. the operation of the read/write -units), but when the idea is refined ahead a more passive system can be done. The idea behind this passive system is to use only tag memories located to strategic places inside the process. They will improve the society's performance by providing a media through which the members can communicate. A logical place for a tag memory will be inside the energy recharging station. While loading the energy, the member can also read the messages and maybe leave some of its own messages. The constant flow of energy through this reloading will make this type of communication possible. These tag memory locations will be discovered by the members moving in a process and they are then marked to the map for further use.

9.2 Members as mobile tag memories

The communication can be further improved by implementing a concept, which could be called as altruistic behavior in a society. A related idea was earlier presented in [30], where the robots were able to transport Intelligent Data Carriers (IDCs) to their operation environment, and thus make the operation more efficient through the creation of an own local world based on the known landmarks. Here the robots won't leave any memories, but instead some members of the society will sacrifice their own work for the sake of the society, and they will start to operate as fixed blackboards (landmarks). The place is chosen from the member's map and then the robot will go to that node and stay there. After it has settled to the location, it will start to send a signal out indicating its existence. This message will be send out until a certain number of members have visited the location, and thus make the location as a part of their map. When another member will detect this "indication" message, it can leave a message to this storage or read the previous messages left there by the other members of the society. This whole concept follows the same idea as the society as a whole: it doesn't matter even though some of the members will die, the rest of them will still do the job. The main problem is to chose a correct mechanism, which will make the decision that a member should start to act as a blackboard: Member with some sort of malfunction or almost without energy will start to behave as a blackboard. Furthermore if a member has permanently got stucked, it can still be useful for the society by starting to act as a blackboard. Other possibilities to be studied include the case, where the decision has been linked directly to the development of the map: if the map seems to stay unchanged for a long time (i.e. the mapping is complete), the member will start to operate as a blackboard. Nevertheless the decision must be made in a member without any knowledge from the other members. If the member stays unnoticed for a certain period of time, the location is useless, and the member has to change its position to a better one. From an engineering point of view, the operator can actively guide the creation and locations of blackboards in the process.

```
altruistic -behavior

IF
  ((running.totally.out.of.energy) OR (pressure stays unchanged for 20*N measurements))
        start.altruism = TRUE
  ((start.altruism) AND (no handshaking with another member for 20*N measurements))
        start.altruism = FALSE
```

When a blackboard is active, the first task for the other members of the society is to match their maps. After a long run, the individual maps should be quite similar, and thus make it possible to compare these maps. The main nodes should be clearly visible on each map, and based on these few nodes the matching should be possible. After this matching when an active member comes to a blackboard it can leave some information about its own map, e.g. some new branches of the graph can be added to the graph on the blackboard. When the next member will pass the blackboard it can compare the map on the board to its own map. By comparing the maps some other members can use the information stored in this map; when two maps are matched, the other active member can get valuable information for example about existing recharging places etc. If this idea is taken even further, it could be arranged so that the blackboard members will locate themselves near the refueling station. This way the environment near the meaningful locations would change and the society would learn something just by detecting the environment.

10. Conclusions and future work

In this paper a biologically inspired concept of a bacterium robot society was presented. The principles of this robot society are derived from bacteria, which are here considered as multicellular organisms. The analogy from Nature includes society’s ability to communicate through chemical messages, to form a swarm and to act as predator hunting for food (i.e to remove algae growth). The use of these behaviors in process industry has been described. Additionally the adaptation of the robot society inside a real process based on the topological learning of the environment along with the ability to cooperate without an active communication is presented.

In near future the construction of the robot society with approximately 10 members will be completed. The recent topological map, which uses the graph representation of the environment will be improved by an altruistic behavior, which will be based on the exchange of information between the society members as they individually perceive the environment. The structure of the simulator will be improved so that the operation of large societies with 100 or more members can be studied. Large population size will obviously make the analogy between bacteria and robot society even more stronger. It is also possible that some complex behaviors found in bacteria can thus emerge in artificial robot societies.

Acknowledgments

The authors would like to thank Johanna Nikkinen for her participation in the development of SUBMAR, Tommi Tuovila and Antti Matikainen for their development of the simulator software. Also we would like to thank James Shapiro for very illuminating conversations concerning the behaviors of bacteria.

References

[1] S. Goss, J.-L. Deneubourg, R. Beckers, J.-L. Henrotte, Recipes for Collective Movement, *Proceedings of the 2nd European Conf. on Artificial Life* (preprints), 1993, pp. 400-410.

[2] J.-L. Deneubourg, G. Theraulaz, R. Beckers, Swarm-Made Architectures, *Toward a Practice of Autonomous Systems* (1st European Conf. on Artificial Life), F. J. Varela, P. Bourgine (Eds.), MIT Press, Cambridge, MA, 1992, pp. 123-133.

[3] R. A. Brooks, A. Flynn, *Fast, Cheap and Out of Control*, Memo 1182, Massachusetts Institute of Technology Artificial Intelligence Lab. Cambridge, Massachusetts, 1989.

[4] M. J. Mataric, Designing Emergent Behaviors: From Local Interactions to Collective Intelligence, *From Animals to Animats 2* (2nd Int. Conf. on Simulation of Adaptive Behavior), J.-A. Meyer, H. L. Roitblat, S. W. Wilson (Eds.), MIT Press, Cambridge, MA, 1993, pp. 432-441.

[5] M. J. Mataric, *Interaction and Intelligent Behavior*, MIT EECS PhD Thesis, 1994.

[6] C. R. Kube, H. Zhang, Collective Robotics: From Social Insects to Robots, *Adaptive Behavior*, Vol. 2, No. 2, 1993, pp. 189-219.

[7] T. Balch, R. C. Arkin, Communication in Reactive Multiagent Robotic Systems, *Autonomous Robots*, Vol.1, No.1, 1994, pp. 27-52.

[8] A. Halme, P. Jakubik, T. Schönberg, M. Vainio, The Concept of Robot Society and Its Utilization, *Proceedings of IEEE Int. Workshop on Advanced Robotics*, Tsukuba, Japan, 1993, pp. 29-35.

[9] M. J. Mataric, Learning to Behave Socially, *From Animals to Animats 3* (3rd Int. Conf. on Simulation of Adaptive Behavior), D. Cliff, P. Husbands, J.-A. Meyer, S. W. Wilson (Eds.), MIT Press, Cambridge, MA, 1994, pp. 453-462.

[10] K. Dautenhahn, Getting to Know Each Other- Artificial Social Intelligence for Autonomous Robots, *Robotics and Autonomous Systems*, Vol. 16, Nos. 2-4, 1995, pp. 333 - 356.

[11] T. Fukuda, S. Nakagawa, Y. Kawauchi, M. Buss, Structure Decision Method for Self Organizing Robots based on Cell Structure -CEBOT, *Proceedings of 1989 IEEE International Conf. on Robotics and Automation*, 1989, pp. 695-700.

[12] T. Fukuda, G. Iritani, Intention Model and Coordination for Collective Behavior in Robotic System, *Distributed Autonomous Robotic Systems*, H. Asama, T. Fukuda, T. Arai, I. Endo (Eds.), Springer-Verlag, Tokyo, 1994, pp. 255-266.

[13] *Proceedings of the 1st Int. Symposium on Distributed Autonomous Robotic Systems*, Saitama, Japan, 1992.

[14] *Distributed Autonomous Robotic Systems*, H. Asama, T. Fukuda, T. Arai, I. Endo (Eds.), Springer-Verlag, Tokyo, 1994.

[15] J. A. Shapiro, Bacteria as Multicellular Organisms, *Scientific American*, Vol. 258, No. 6, 1988, pp. 62 - 69.

[16] E. O. Budrene, H. C. Berg, Dynamics of formation of symmetrical patterns by chemotactic bacteria, *Nature*, Vol. 376, No. 6, 1995.

[17] P. Appelqvist, *Design of an Intelligent Sensor/Actuator Robot for Process Control and Robot Societies Studies*, Master's Thesis, Helsinki University of Technology, 1996.

[18] A. Halme, P. Appelqvist, P. Jakubik, P. Kähkönen, T. Schönberg, M. Vainio, Bacterium Robot Society- A Biologically Inspired Multi-Agent Concept for Internal Monitoring and Controlling of Processes, *Proceedings of 1996 IEEE/RSJ Int. Conf. on Intelligent Robots and Sytems*, Osaka, Japan, 1996 (to appear).

[19] L. Buscemi, M. Prati, G. Sandini, Cellular Robotics: Behavior in Polluted Environments, *2nd International Symposium on Distributed Autonomous Robotic Systems (preprints)*, Saitama, Japan, 1994.

[20] A. Martinoli, F. Mondada, Collective and Cooperative Behaviors: Biologically Inspired Experiments in Robotics, *4th Intl. Symposium on Experimental Robotics (preprints)*, Stanford, California, 1995.

[21] R. Beckers, O. E. Holland, J. L. Deneubourg, From Local Actions to Global Tasks: Stigmergy and Collective Robotics, *Artificial Life IV*, R.A. Brooks, P. Maes (Eds.), MIT Press, Cambridge, MA, 1995, pp. 181-189.

[22] R. A. Brooks, *Intelligence without reason*, Memo 1293, Massachusetts Institute of Technology Artificial Intelligence Lab. Cambridge, Massachusetts, 1991.

[23] P. Maes, Behavior-Based Artificial Intelligence, *From Animals to Animats 2* (2nd Intl Conf. on Simulation of Adaptive Behavior), J.-A. Meyer, H. L. Roitblat, S. W. Wilson (Eds.), MIT Press, Cambridge, MA, 1993, pp. 2-10.

[24] L. Steels, The Artificial Life Roots of Artificial Intelligence, *Artificial Life: An Overview*, C. G. Langton (Ed.), MIT Press, Cambridge, MA, 1995, pp. 75-110.

[25] R. A. Brooks, A Robust Layered Control System For a Mobile Robot, *IEEE Journal of Robotics and Automation*, Vol. RA-2, No.1, 1986, pp. 14-23.

[26] R. C. Arkin, Motor schema-based mobile robot navigation, *Int. J. Robotics Res*., Vol. 8, No. 4, 1989, pp. 92-112.

[27] R. C. Arkin, Integrating behavioral, perceptual, and world knowledge in reactive navigation, *Designing Autonomous Agents*, P. Maes (Ed.), MIT Press, Cambridge, MA, 1990, pp. 105-122.

[28] M. J. Mataric, Navigating With a Rat Brain: A Neurobiologically-Inspired Model for Robot Spatial Representation, *From Animals to Animats* (1st Int. Conf. on Simulation of Adaptive Behavior), J.-A. Meyer, S. Wilson (Eds.), MIT Press, Cambridge, MA, 1991, pp. 169-175.

[29] B. M. Yamauchi, *Exploration and Spatial Learning in Dynamic Environments*, Case Western Reserve University, PhD Thesis, 1995.

[30] T. v. Numers, H. Asama, T. Fujita, S. Kotosaka, S. Miyao, H. Kaetsu, I. Endo, An Intelligent Data Carrier System for Local Communication Between Cooperative Multiple Mobile Robots and Environment, *2nd IFAC Conf. on Intelligent Autonomous Vehicles*, Espoo, Finland, 1995, pp. 366- 371.