Autonomous Navigation System Applied to Collective Robotics with Ant-Inspired Communication

Renato Reder Cazangi
LBiC – DCA – FEEC – Unicamp
PO BOX 6101/CEP 13083-970
Campinas, SP, Brazil
+55 (19) 3788-3885
renato@dca.fee.unicamp.br

Fernando J. Von Zuben
LBiC – DCA – FEEC – Unicamp
PO BOX 6101/CEP 13083-970
Campinas, SP, Brazil
+55 (19) 3788-3820
vonzuben@dca.fee.unicamp.br

Maurício F. Figueiredo
DIN – UEM
Av.Colombo, 5790/CEP 87020-900
Maringá, PR, Brazil
+55 (44) 263-5874
mauricio@din.uem.br

ABSTRACT
Research in collective robotics is motivated mainly by the possibility of achieving an efficient solution to multi-objective navigation tasks when multiple robots are employed, instead of a single robot. Several approaches have already been tried in multi-robot systems, but the bio-inspired ones are the most frequent. This paper proposes to augment an autonomous navigation system based on learning classifier systems for using in collective robotics, introducing an inter-robot communication mechanism inspired by ant stigmergy, with each robot acting independently and cooperatively. The navigation system has no innate basic behavior and all knowledge necessary to compose the decision-making artifact is evolved as a function of the environmental feedback only, during navigation. Repulsive and/or attractive pheromone trails are produced by the robots along navigation, following very simple rules. Basically, each robot has to perform obstacle avoidance and target search, and the status of the pheromone level at the position currently occupied by each robot will influence the coordination of the two fundamental behaviors. Experiments are performed in simulation, with comparative results indicating that the presence of the pheromone trails is responsible for significant improvements in the capture rate and in the length of the route adopted by each robot.

Categories and Subject Descriptors
I.2.9 [**Artificial Intelligence**]: Robotics – *autonomous vehicles;* I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence – *multiagent systems, intelligent agents;*

General Terms
Algorithms, Performance and Experimentation.

Keywords
Collective robotics, autonomous navigation, learning classifier systems and ant stigmergy.

1. INTRODUCTION
The problems related to autonomous robot navigation have shown to be very attractive for researchers of diverse areas because of their complexity and challenging issues. Before the last twenty years, most research efforts were concentrated on single robot systems. Just recently multi-robot systems and collective robotics are receiving a great deal of attention [2].

The reasons for this growth of interest are mainly related to limitations of single robots and advantages associated with multi-robots applications. Several tasks, like transport of objects, may require the use of various robots. In other tasks, a single robot may solve them but not as well as a collection of robots would do, especially if they cooperate.

Some approaches for collective robotics, possibly the most frequent, are the biologically inspired ones. Such approaches are based on social characteristics of insects and animals, mainly ants, bees and birds. The studies of ant societies, their structure and dynamics have provided a great deal of knowledge about how such simple insects living in such complex social organization are able to perform a plethora of tasks with high level of efficiency.

The scientific knowledge about ants has inspired researchers to devise various robotic tasks mimicking ants' behaviors. The work of Kube *et al.* [13] is a good example. They have modeled the multi-robot box-pushing task according to the behavior ants exhibit when transporting food and prey, and also explored the manner ants build their nests, called blind bulldozing. Based on ant brood sorting, the authors have developed strategies for collective robot clustering.

Beyond the aforementioned social behaviors, there is one extensively explored and of special interest in this work, the stigmergy. It consists in a form of indirect communication among individual ants mediated by secretion of chemical substances, denoted pheromones, in the environment [6].

The pheromone operates as communication signals and can have several roles (e.g., alarm, recruitment, and so on). Some socially advanced species of ants may make use of up to 20 pheromone types, each one with a different purpose. A second basic aspect related to pheromones is the concentration, which is proportional to the relevance of such signals. For example, in the case of food transport, the pheromone trail concentration may indicate the amount and quality of the food source [5].

The use of artificial pheromone trails in computational systems and algorithms has been very widespread. Some works have considered such inspiration for optimization [4][10] and clustering [1][14] problems. Among other works concerning collective robotics, Wagner and Bruckstein [16] developed an algorithm for cleaning a dirty area where multiple robots cooperated by leaving trails on the ground. Ding *et al.* [11] employed artificial pheromone as quantifiers of task difficulty, that is, the robots should try to carry out tasks and, after that, deposit an amount of pheromone proportional to the task difficulty. Harder tasks, associated to higher pheromone concentration, would attract more robots to cooperate with each other and solve them.

This work is inserted into the context of bio-inspired collective robotics as an extension of previously published works associated with single robot navigation. Cazangi and Figueiredo [7], and Cazangi *et al.* [8] presented an autonomous navigation system (ANS) based on learning classifier systems [12] and evolutionary algorithms, with simulated scenarios and also employing real robots. In contrast with other approaches in the literature [11] [17], the previously cited ANS has no innate basic behaviors, such as collision avoidance, shortest path finding, etc. All knowledge necessary to compose the robot controller is evolved according to environmental feedback during navigation.

The purpose here is the application of the previously developed ANS in multi-robot navigation problems, introducing a new mechanism for inter-robot communication based on ant stigmergy. The primary objectives are the same of the single robot case and are related to autonomous navigation. That is, starting with no initial knowledge and without external assistance, the robots must navigate through unknown environments accomplishing two conflicting tasks: obstacle avoidance and target seeking [3]. In such problems, the robots usually present a poor initial performance, since the learning phase will take place on-line.

The robots will be able to mark regions of the environment with artificial pheromones, according to past experiences, assisting one another in a cooperative way with indirect communication. With the presence of pheromone trails, improvements are expected in accomplishing the navigation task. Beyond a higher effectiveness in avoiding collisions and capturing targets, the minimization of distances traveled between targets and also the mapping of the environment are taken as secondary objectives, being a consequence of the accumulated deposition of pheromone.

The remainder of this paper is organized as follows. Section 2 describes the characteristics of the robots. Section 3 presents the autonomous navigation system and introduces the pheromone trail mechanism. Section 4 outlines the experimental results already obtained, and concluding remarks are delineated in Section 5.

2. THE ROBOT MODEL

The virtual robots used in simulation in this work were modeled with the same characteristics of Khepera II [8] and will be detailed below. As Figure 1 shows, there are eight infrared sensors disposed around the robot, being six at the front and two in the hinder part. They are responsible for measuring the distance from obstacles and target direction and intensity. As the robot does not move backward, the navigation system will ignore, in the applications to be presented, the measurements of obstacle distance provided by the two rear sensors. One new sensor is introduced: it reads the type and concentration of artificial pheromone present in the current position of the robot.

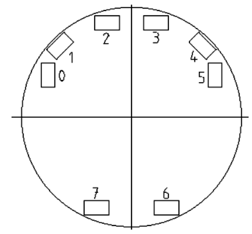

Figure 1. The organization of robot sensors. These sensors are responsible for measuring the distance to obstacles (proximity sensors) and targets (luminosity sensors). The pheromone sensor is not presented.

Regarding actuators, the adjustment of direction defined by the navigation system can range from 15 degrees (clockwise) to –15 degrees (counter-clockwise). There is also a new actuator that is responsible for depositing artificial pheromone in the environment.

Although the standard Khepera II is not able to leave marks on the ground and sense them, inserting additional accessories may give rise to such functionalities. Some works have developed and applied physical mechanisms that make marks on the floor which, though not exactly based on pheromones, represent properly the role trails have in the simulations performed here. One simpler alternative is tracing lines on the ground with a pen or chalk. A more sophisticated way involves laying down solvent substances (thinner) that discolor the floor surface (i.e., a black paper) [15].

3. NAVIGATION SYSTEM AND COMMUNICATION MECHANISM

The original autonomous navigation system presented in Cazangi and Figueiredo [7], and Cazangi *et al.* [8] was directed to robots operating in isolation. In this work, simple modifications were performed in the ANS, specifically related to the scheme of decision making. Both, the original and the new aspects involved in the communication mechanism will be described in the next sections.

3.1 Autonomous Controller

Each robot is controlled by an autonomous navigation system (ANS) that works based only on instantaneous stimuli captured from the environment and does not contain *a priori* knowledge. To accomplish the primary navigation objectives, the robot have to present different behaviors in time, sometimes searching for targets, sometimes avoiding obstacles. The nature of the behavior emerges as a consequence of the interaction of the robot with the environment. The proper behavior to be adopted during navigation cannot always be established in a straightforward manner in face of the frequent occurrence of conflicting situations. Therefore, the existence of a coordinating module tuned by a learning algorithm is required.

The ANS is based on the learning classifier system (LCS) paradigm proposed by Holland [12], an evolutionary approach to synthesize adaptive inference mechanisms capable of operating in time-varying conditions. The LCS interacts with the environment by means of detectors and actuators. Detectors receive and encode incoming messages from the environment. Actuators provide means to act on the environment, decoding and applying actions defined by the system. After acting, the system also receives an environmental feedback.

A set of classifiers composes the LCS. They are ⟨condition⟩-⟨action⟩ rules with an if–then inference mechanism. The antecedent and consequent parts of each classifier are usually binary strings. Each classifier has an associated strength that is related to its capability to act toward the achievement of predefined objectives.

There are three sub-systems in a LCS: rule and message sub-system, apportionment of credit sub-system and rule discovery sub-system. They interact, in brief words, as follows. When the detectors capture messages from the environment, they are sent to the rule and message sub-system. Then all the classifiers try to match its antecedent part with the environment message. Those classifiers with better matching take part in a competition process. Among them, the classifier that offers the highest bid (depending mainly on the classifier strength) wins and acts on the environment. The action causes an environmental feedback that is used by the apportionment of credit sub-system to readjust the strength of the winner classifier. Thereafter, the environment emits a message with its new current state that is received again by the rule and message sub-system. The process continues until an epoch of iterations is concluded. After that, the rule discovery sub-system runs aiming at producing new and improved classifiers. Usually, a genetic algorithm is responsible for the process of rule discovery, taking the individual strength as the fitness of each classifier. Detailed information about the original LCS can be found in Holland [12].

Due to functional purposes, the classifier system implemented in the ANS differs in some relevant aspects from Holland's LCS. At first, each classifier has two distinct antecedent parts and two distinct consequent parts, instead of just one. Moreover, not all parts are composed of binary values. Integers are also used in the codification. When the classifiers compete to act on the environment, the winner is the one that presents the best matching with the received message. There is no bid and the classifier strength does not influence the competition process. The strength is only used for the rule discovery sub-system, when computing the fitness value. This sub-system can be triggered every time one of three possible events is detected during navigation: collision, target capture, or monotony (virtual event detected when the robot presents monotonous behavior). Furthermore, for each event there is a different evolutionary algorithm with specific fitness functions and procedures, producing a new generation of rules. The complete details of the whole system are described in the following sections.

Returning to the description of the original autonomous navigation system, it can be said that it interacts with the environment by means of sensors and actuators, and it is arranged in four main components: population of rules, evaluation module, reproduction module and competition module (Figure 2). Notice that the actuators will determine the direction and speed of the robot at each navigation step. The population of rules (Section 3.1.1) represents the knowledge base and evolves during the robot navigation. The competition module receives stimuli captured by target and obstacle sensors, performs a matching with ⟨condition⟩-⟨action⟩ rules, and defines which one is going to act on the environment. This process is repeated every time a control action is required, forming a loop that is only interrupted to give rise to an evolutionary update of the population of rules, denoted here as the evolution phase.

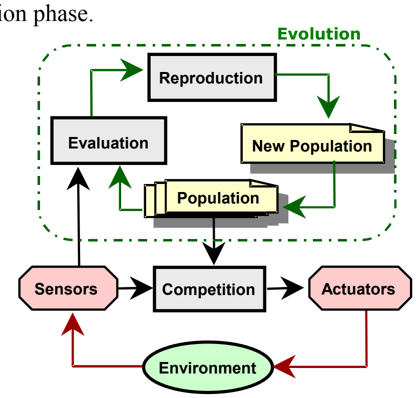

Figure 2. The main structure of the ANS.

It is important to highlight that the evolution of the set of rules (knowledge base) will depend on the interactions with the environment, and that these interactions will be determined by the sequence of rules selected to provide the control action at each instant of time.

3.1.1 Population of Rules

As in the original LCS, each individual of the population is represented by ⟨condition⟩-⟨action⟩ rules, with a *modus-ponens* inference mechanism type. Each rule can be described by a list of attributes, called a chromosome. Each chromosome, shown in Figure 3, is composed of four vectors: obstacle distance (RO: vector of integers), target light intensity (RA: vector of integers), direction adjustment (RD: binary vector) and speed adjustment (RV: binary vector). Therefore, RO and RA comprise the antecedent part and contain, respectively, six and eight components. The former corresponds to the number of proximity sensors, and the latter to the number of luminosity sensors of the robot. The RD vector (consequent part) has nine components, with the five more significant bits indicating signal, and the four less significant bits being used to indicate an absolute value. Considering the signal bits, if most of them are zero, the signal is negative, otherwise the signal is positive. Regarding the RV vector, if most of them are zero, the robot speed should be reduced, otherwise it should be increased by a fixed amount.

Several bits, together with a majority voting procedure, are adopted simply to provide a smooth transition between states in antagonism. The rules are initially constructed with totally random values in both antecedent and consequent parts.

Obstacle Antecedent (RO)

900	900	900	700	90	50

Target Antecedent(RA)

0	0	20	100	350	100	20	0

Direction Consequent (RD)

1	1	0	0	0	0	0	1	1

Speed Consequent(RV)

0	0	1	1	1

Figure 3. Example of a rule representation as a chromosome. The consequent parts determine a turn of –3º and an increase in speed.

3.1.2 Competition Module

At each robot movement, the obstacle and target sensors capture stimuli from the environment in the form of two vectors, EO and EA, respectively, sending them to the competition module.

The other piece of information used in this module is gathered by the pheromone sensor for inter-robot communication. The factors CR and CA are introduced in Equation (1) and represent an innovative aspect of the proposal when compared with the original ANS [8]. Their role and what kind of association they have with the pheromone sensor will be explained in section 3.2.

At this stage, all rules compete in a winner-takes-all process. The winner will be the rule that presents the best matching with EO and EA. The similarity of each rule r, $S(r)$, is given by:

$$S(r) = \frac{\|RO(r) - EO\|}{MaxO} CR + \frac{\|RA(r) - EA\|}{MaxA} CA \tag{1}$$

where $MaxO$ and $MaxA$ mean suitable values for normalization of the first and second terms of $S(r)$, respectively, and $\|\cdot\|$ is the Euclidean norm. The consequent of the winner rule is then used to command the actuators. The RD consequent determines the adjustment in direction, and the RV consequent establishes the variation in speed for the next movement. The speed is always modified by a constant value. This way, the winner rule just indicates if the speed increases or decreases.

3.1.3 Evolution

The evolution process, responsible for evolving the population of rules, is composed of evaluation and reproduction modules. Evolution takes place every time one of the following events is detected: collision with an obstacle, capture of target and monotony. A monotony event is triggered if the robot does not capture a target for a long time, or if the sum of direction adjustments of previous iterations exceeds a predefined threshold (to stop situations when the robot is moving in circles, for example). The events that demand evolution of the population of rules have, each one, different objectives associated with specific evolution processes. These processes are depicted during the description of the evaluation and the reproduction modules.

In collective robotics, several robots navigate in the environment and each one is an obstacle for the others. Thus, due to their size and limited sensorial capability, collisions between robots are frequent and may be caused by a large variety of events. So, two consecutive evolution processes tends to be separated by small intervals of time, with no chance of a proper evaluation of the current population of classifiers. Because of that, events of collision between robots tend to degenerate the population and therefore will not trigger evolution processes.

3.1.3.1 Evaluation Module

3.1.3.1.1 Collision

The aim of evolution just after each collision event is to improve the skill for obstacle avoidance. Independent evaluations are performed, one for the antecedent of the rules, and another for the consequent part. The first one sorts the individuals according to their similarity to the instantaneous collision situation. Equation (1) is used here, where EO and EA are associated with the stimuli captured at the collision instant and CR=CA=1. In the consequent evaluation, instinctive reflex is considered for the robot, after each collision, and is given by $T(r)$ in Equation (2). The idea is that the output proposed by the selected rule be altered by a fixed amount of 15, forcing the robot to point to a direction that tends to move the robot away from the obstacle.

$$T(r) = \begin{cases} \left|[RD(r)]_d - 15\right|, \text{if left collision;} \\ \left|[RD(r)]_d + 15\right|, \text{otherwise.} \end{cases} \tag{2}$$

where $[RD(r)]_d$ is a real value that represents the adjustment in direction defined by the consequent part of rule r.

It is clear that in risky situations (imminent collision) the robot should reduce its speed. Because of that, the consequent part of each rule, related to speed, is evaluated based on a Hamming distance between the $RD(r)$ vector and a pattern vector that represents speed reduction (all elements with value one).

3.1.3.1.2 Capture

The evaluation procedure here is an analogue of the collision one, aiming at improving the capture efficiency. However, the evaluation of the antecedent part takes into account the instantaneous capture situation. The evaluation of $D(r)$ (consequent RD) is given by the sensor that detected the capture, as follows:

$$D(r) = \left|[RD(r)]_d - \alpha\right|, \tag{3}$$

where $[RD(r)]_d$ is the same as in Equation (2), and α is the angle of the sensor that detected the event.

When capturing a target, the operation of speed decreasing is also necessary (see the previous section), since it is generally related to a place where a task must be carried out, such as object collecting. So, the behavior is similar to the one associated with the collision case.

3.1.3.1.3 Monotony

The monotony events are characterized by the robot presenting navigation behaviors that do not produce collision or target capture events.

At the beginning of the navigation experiment, with the rules being randomly generated, collision events are more frequent and imply intensive learning, as expected. Also, collision may not necessarily represent a risk of physical damage. A collision event can be characterized by the occurrence of a distance between the robot and an obstacle smaller then a threshold.

So, the autonomous navigation system should avoid monotony in favor of target seeking, possibly increasing the probability of collision. Monotony is generally measured as a function of a predefined number of iterations. Thus, every time that monotony is detected, the whole set of rules that are being selected to provide the control action (for example, moving the robot consistently away from the target) will participate in the reproduction stage. The remaining rules are kept fixed.

3.1.3.2 Reproduction Module

3.1.3.2.1 Collision and Capture

In the case of collision or capture evolution, the reproductive process is very similar. The only differences are the evaluation procedures performed before reproduction (details in evaluation module). Taking the evaluations as references, the next step is parent selection in pairs (roulette wheel) and application of one-point crossover. As a result, two offsprings are generated. Finally, they are mutated according to a small probability rate.

It is important to remark that the amount of offspring produced per generation is only 10 % (procreation rate) of the population's total size. This policy reduces the probability of occurring harmful interference among different kinds of sequential evolution processes. Concluding the offspring generation, new antecedent parts are randomly combined with new consequent parts, originating the final individuals that replace their corresponding parents. The remaining individuals are kept and the new generation is finally obtained.

There is still an important mechanism that plays the role of keeping the population balance between individuals associated with obstacle avoidance and target capture behaviors. A balance mechanism is necessary because there is a natural trend toward the predominance of rules associated with target capturing, simply due to the consecutive occurrence of target captures.

Table 1. Relation between the number of consecutive captures and the procreation rate.

Consecutive Captures	1	2	3	4	5	6	7	8	>8
Procreation Rate (%)	10	8	7	6	5	4	3	2	1

Rules associated with obstacle avoidance should not be discarded as a consequence of the reduction in the number of collisions. In order to implement the aforementioned balance, the procreation rate is decreased gradually, while consecutive captures take place. The progeny size is adjusted as described in Table 1. If a collision or monotony event is detected, the procreation rate is set immediately to 10%. Although the number of descendants generated in evolution processes can be reduced by the decrease of the procreation rate, the system never stops evolving. The smallest possible rate is 1%. Therefore, at least this amount of offspring rules will be always produced.

3.1.3.2.2 Monotony

Assuming that the evaluation module has sorted all rules, just those with worst evaluation are modified. The modification consists in removing those rules and inserting new random rules in replacement. This way, the rules responsible for monotonous behavior tend to be eliminated, thus suppressing the anomalous navigation behavior.

3.2 Pheromone Trail Mechanism

Considering that several robots navigate through the same environment, a mechanism for indirect communication among them has been designed. The mechanism is inspired by ants, more specifically by their pheromone trails, and will be described next.

The robots are able to deposit artificial pheromone in any position of the environment in which they are navigating. There are two types of pheromone: repulsive and attractive. In the simulator, the repulsive one is represented by red dots and the attractive by blue dots. Both marks can have a stronger or lighter tint according to their concentration.

The repulsive pheromone indicates a "dangerous region" from where the robot should run away. Such type of pheromone is deposited when the robot collides with obstacles. Immediately after the collision, the robot repeats backwardly the last 10 movements, marking each discrete point of the trajectory (performed just before the collision) with repulsive pheromone. In case of attractive pheromone, it indicates interesting regions for navigation. The attractive pheromone is deposited in two ways. The first way is analogous to the case of collision, but the 10 points of the trajectory marked are the ones just before a target capture. The second way is more sophisticated and depends on the "intact trajectories".

The intact trajectories are the routes executed by robots between two consecutive target captures. Consecutive here means that no events of collision or monotony happen between the two captures. Thus, every time a robot captures any two targets consecutively, all points of the referred trajectory are marked with attractive pheromone. In brief words, intact trajectories become attractive pheromone trails, having one target as starting point and another as ending point. Furthermore, each intact trajectory has a length that represents the distance traveled by the robot to go from the first target to the second one. Every length is computed to get the mean length between targets and consequently the mean distance covered between target captures. The mean distance has a general unit called d.u. (distance unit).

The artificial pheromone deposited has type and concentration as attributes. So, each position of the environment is associated with the type and the concentration of pheromone. The concentration is an integer value and is incremented by one each time the respective position receives a pheromone deposit of the same type previously released. The maximum concentration of pheromone is fixed and denoted σ_{max}. If a deposit of repulsive pheromone occurs over an attractive one, the latter is ignored and the repulsive predominates. On the contrary, nothing changes. Because the collision event is more critical and undesirable, the repulsive marks will always prevail.

The whole pheromone trail mechanism affects simply just one component of the robots' navigation system: how the rule that will actuate at each iteration is selected. Such decision is taken in the competition module (Section 3.1.2) by means of Equation (1). The pheromone trails provide information for determining the values of *CR* and *CA*, favoring rules more suitable to avoid obstacles or capture targets. In every robot movement (iteration), the sensor reads the type and concentration of artificial pheromone available at the current position. After that, it is possible to calculate *CR* and *CA* as follows:

$$\begin{cases} CR = 1 + \dfrac{concentration}{10} \text{ and } CA = 1, \text{ if } type = \text{repulsive;} \\ CA = 1 + \dfrac{concentration}{10} \text{ and } CR = 1, \text{ if } type = \text{attractive;} \\ CA = 1 \text{ and } CR = 1, \text{ otherwise;} \end{cases} \quad (4)$$

In general terms, the idea of the pheromone trail mechanism is to consider experiences already lived by the set of robots for assisting them in taking more suitable decisions when navigating.

4. RESULTS

The following results were obtained with the navigation system using the pheromone trail mechanism developed as means of indirect communication among several robots. The experiments to be presented are organized in three stages, aiming at achieving the proposed objectives (obstacles avoidance, targets capturing, distance optimization and environment mapping) and the improvements provided by the implemented communication mechanism in collective robotics.

All experiments were performed using a simulator implemented by the authors. Each simulation was repeated three times and the mean of the resulting values was considered. To measure the robots' performance, three indices were adopted: cumulated number of events of collision with obstacles, capture of targets and monotony. Results considered satisfactory are those that present as many events of capture as possible and as few events of collision and monotony as possible.

Although several robots navigate simultaneously across the environments, they are, each one, controlled by their own autonomous navigation system. Furthermore, the robots do not communicate directly. Other distinction is related to pheromone evaporation. Due to the reduced size of the environment, the small number of robots interacting, and the objective of environment mapping via accumulated pheromone, the evaporation of pheromone is not considered in this work.

Every environment set up is closed and has obstacles (black rectangles), targets (circles) and robots (triangles) disposed arbitrarily. Robots must capture the targets in a fixed and serial sequence. When robots complete the sequence of captures, it is restarted. Thus, it is important to make clear that the robots do not compete for targets, they just have to capture them in a pre-established sequence. No target is removed or inserted during simulation. Moreover, the maximum concentration of pheromone (σ_{max}) is set to 40.

The first stage consists in comparing the robots mean performance with and without the pheromone trail mechanism. The environment employed is shown in Figure 4 and is composed of three central obstacles and six targets that should be captured in a fixed and cyclical sequence (1 to 6). A hundred and fifty thousand iterations was the duration of each one of the six experiments done and there were four autonomous robots (initially disposed arbitrarily) navigating through the environments. It is important to highlight that the robots do not have significant initial knowledge, so that *a priori* behaviors or navigation strategies have not been incorporated. Consequently, they are expected to suffer numerous events of collision with obstacles and monotony until they become able to navigate satisfactorily. Such evolution of basic behaviors is best presented and explored in Cazangi and Figueiredo [7].

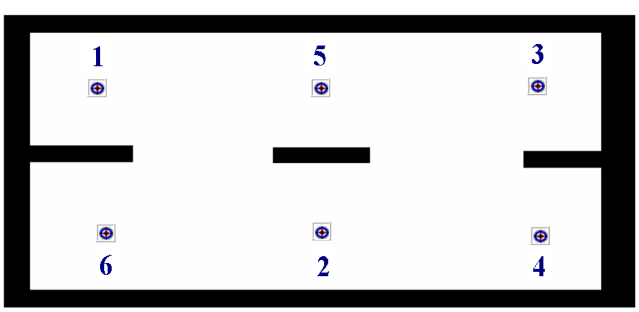

Figure 4. Initial environment and target sequence.

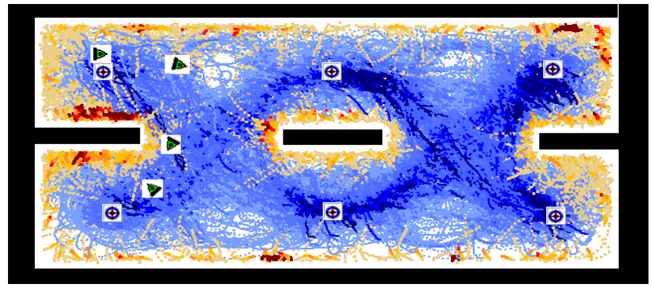

Figure 5. Concentration of pheromone trails after simulation.

Figure 5 depicts the accumulated concentration of pheromone deposited in the environment by the four robots. As already explained, there are two kinds of pheromones: attractive (blue tones) and repulsive (red tones, usually located close to obstacles). In both cases, stronger trails or regions mean larger concentrations; otherwise lighter ones indicate smaller concentrations. Trails from target 2 to 3, from 3 to 4 and from 4 to 5 are more concentrated, indicating that those paths were the most visited. It is also interesting to notice the environment mapping built by the robots through deposit of pheromone.

Table 2 contains the results of the experiments at the first stage. Note that the presence of pheromone trail improved the robots' performance in all criteria, when compared with the absence of a pheromone trail mechanism. In average, the amount of collision events was reduced by 6.44%, and of monotony events by 6.38%. The best remark was the growth of almost 36% in capture of targets.

Table 2. Statistics of the first set of experiments.

Pheromone Trail	Collision	Capture	Monotony
off	509	446	261
off	518	489	241
off	587	548	234
Mean	538	494.33	245.33
on	487	580	250
on	578	770	210
on	445	666	229
Mean	503,33	672	229,66
Increment (%)	-6.44	+35.94	-6.38

Next, the second stage focuses on evaluating the robots' capability of minimizing distances. To analyze this property, a closed environment was designed, shown in Figure 6(a), with a central obstacle between two targets which should be captured alternately. The simulations lasted 40 thousand iterations and there were five robots navigating.

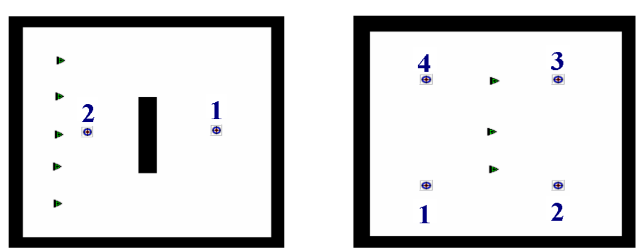

Figure 6. Organization of the environments (a) and (b).

Table 3. Statistics of the second set of experiments.

Pheromone Trail	Collision	Capture	Monotony	Intact Trails	Mean Distance
off	139	156	105	30	617.31
off	99	109	81	51	640.86
off	118	140	85	72	548.72
Mean	118.67	135.00	90.33	51	602.29
on	70	234	58	163	494.53
on	128	183	87	103	404.29
on	148	293	49	205	344.58
Mean	115.33	236.67	64.67	157	414.46
Increment (%)	–2.81	+75.31	–28.40	+307.84	–31.18

Table 3 presents the results of the six simulations and the improvements provided by the pheromone trail mechanism. The number of collision and monotony events were reduced by 2.81% and by 28.40%, respectively. The capture events were 75.31% higher, and the number of target captures accomplished by means of intact trajectories was triplicated. Finally, the mean distance of intact trails (between the two targets) decreased 31.18%. This result confirms the role of the developed mechanism to minimize the navigated extension from one target to another.

The mean distance variation during the simulation is presented in Figure 7 for two chosen cases (from Table 3): the third off (square) and the second on (circle). The *circle* marked curve (○) shows the mean distance converging more quickly than the *square* marked one (□) and toward smaller values. Furthermore, the frequency of occurrence of intact trails (marked with circles and squares in the curves) is higher when the pheromone mechanism is active. But, when it is inactive, as represented by the *square* marked curve, the interval between occurrences is less regular. It is interesting to note that the first captures may occur casually resulting in abrupt variation of mean distance, as seen in the initial 15 thousand iterations.

The third stage addresses the optimization aspect, where some experiments were prepared aiming at verifying the system efficiency in searching for optimum values. That is, in addition to navigating attending the objectives of capturing targets and avoiding obstacles, the robots will also be optimizing trajectories and minimizing distances.

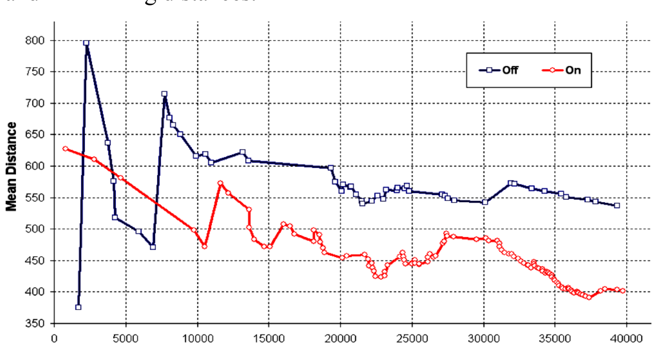

Figure 7. Curves of mean distance versus iterations.

A closed environment was designed with four targets (sequence 1 to 4), each one disposed 200 d.u. far from its precedent target. As illustrated in Figure 6(b), the environment does not contain obstacles in a way that the best trajectory between two targets is always a straight line of length equal to 200 d.u. Moreover, there are three robots and the simulations lasted 50 thousand iterations.

The optimum length of the complete sequence (from target 1 to 4) measures 800 d.u., being that the optimum average distance between targets is 200 d.u. Looking at Table 4, where the results are placed, it can be noticed that a value very close to the optimum (201.91) was achieved when the pheromone trail mechanism is active. Although the average performance is not so close to the optimum, the pheromone trail mechanism can be considered relevant to suggest shorter trajectories to be followed by multiple robots.

Yet detailing the simulation case in which the minimal mean distance was obtained, Figure 8 presents a curve with its variation along 50 thousand iterations. Similar to what commonly happens in navigation experiments, the learning of basic behaviors (like obstacle avoidance and target capturing) takes place along the first iterations (until 24 thousand, in this case). Just after that period, the behaviors are well established, so the intact pheromone trails begin to appear and become more and more concentrated. As well as the concentration increases, the robots start to adopt shorter paths. The tendency of minimizing the distance traveled is clear after iteration 24 thousand, when the mean distance is greatly reduced and the frequency of occurrence of intact trails increase (accumulated circles).

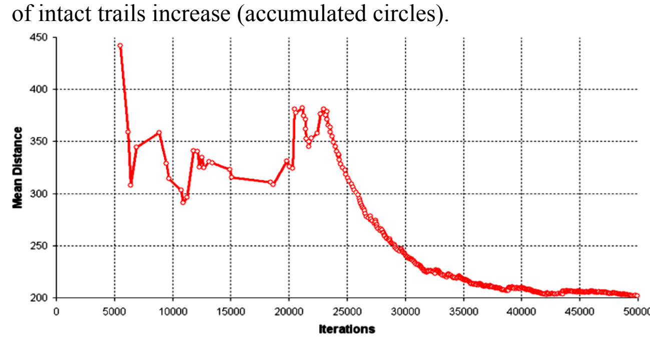

Figure 8. Evolution of the mean distance in the best case.

The same experiment associated with the graphic of Figure 6(b) has Figure 9 as final image. It shows the pheromone trails concentration and the most used trajectories after 50 thousand iterations. In terms of mapping the environment, it is also possible to divide the process in two distinct phases. The first iterations are important mainly to mark regions with repulsive pheromone because most collisions occur in such phase. In the subsequent iterations, the role changes and it is time to mark regions mainly with attractive pheromone, as a consequence of a constant production of intact pheromone trails.

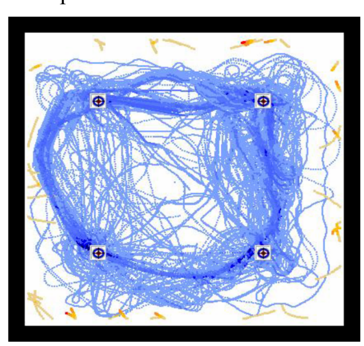

Figure 9. Concentration of pheromone trails after simulation.

Table 4. Statistics of the third set of experiments.

Pheromone Trail	Collision	Capture	Monotony	Intact Trails	Mean Distance
off	39	249	57	204	291.23
off	45	233	84	184	231.46
off	21	258	82	239	235
Mean	35	246.67	74.33	209	252.56
on	25	468	30	444	234.52
on	40	344	55	305	220.84
on	40	425	37	396	201.91
Mean	35	412.33	40.67	381.67	219.09
Increment (%)	0.0	+67.15	–45.28	+82.61	–13.25

Examining the remaining results of Table 4, once more the developed mechanism provided good improvements over the absence of pheromone, except in number of collision events. The mean distance of the on-cases was 219.09 d.u., difference of only 19 d.u. from the expected value (200). It is important to emphasize that the navigation system has no rewards or specific artifices toward the minimization of distances. This kind of artifices may further improve the results.

5. CONCLUSIONS

In this work a mechanism for collective robot navigation with indirect communication was proposed, inspired by pheromone trails in ant systems. Each robot was controlled by the autonomous navigation system (ANS) presented in Cazangi and Figueiredo [7] and Cazangi *et al.* [8], extended with an additional sensor for detecting the kind and level of pheromone present at the current location of the robot, and also an additional actuator to deposit pheromone following very simple rules. The robots have no initial knowledge and learning is accomplished by means of a classifier system composed of if-then rules with a particular configuration for the antecedent and consequent parts.

Beyond the aim of extending and validating the previously developed ANS in collective scenarios, the indirect mechanism for communication promotes an overall better performance, optimization of trajectories, and environment mapping.

Even though not every aspect employed in simulation is actually bio-inspired, the extensions and additional modules prove to be helpful in navigation tasks with multiple objectives and incremental learning. Partial bio-inspirations are acceptable considering that many algorithms or systems developed based on biological phenomena are so improved and extended that the original analogy becomes just an inspiration [9].

Three sets of experiments were performed. When comparing the ANS with and without the use of pheromone trails, all experiments guide to improvements caused by the presence of pheromone. Mainly in terms of target capturing, the gain was very significant.

Related to trajectory optimization, the results indicate that the system is able to reduce the distances traveled between targets. Although it can be considered just as a side effect, the system capability of minimizing distances is promising, especially regarding that the ANS was not designed toward this end. The same can be said about the environment mapping obtained through accumulation of pheromones (see Figures 5 and 9).

The future perspectives are associated with preparing the ANS for application in real optimization problems and also in clustering. Other collective tasks will be considered (e.g., box-pushing) and co-evolutionary scenarios can also be envisaged.

6. REFERENCES

[1] Abraham, A. and Ramos, V. Web usage mining using artificial ant colony clustering and linear genetic programming. *In Proceedings of CEC '03*. Canberra, Australia, 2003, (2) 1384 – 1391.

[2] Arai, T., Pagello, E., and Parker, L. Guest editorial, Advances in multi-robot systems. *IEEE Transactions on Robotics and Automation, 18*(5), 2002, 655-661.

[3] Arkin, R.C. Behavior-Based Robotics, The MIT Press, 1998.

[4] Bauer, A., Bullnheimer, B., Hartl, R.F. and Strauss, C. An ant colony optimization approach for the single machine total tardiness problem. *In Proceedings of CEC'99*. Piscataway, USA, 1999, 1445-1450.

[5] Caetano, F.H., Klaus, J. and Zara, F.J. Ants: Biology and Anatomy. (in Portuguese) Editora da UNESP, Rio Claro, 2002.

[6] Camazine, S., Deneubourg, J.L., Franks, N.R., Sneyd, J., Theraulaz, G. and Bonabeau, E. *Self-Organization in Biological Systems*, Princeton University Press, 2001.

[7] Cazangi, R.R. and Figueiredo, M.F. Simultaneous Emergence of Conflicting Basic Behaviors and Their Coordination in an Evolutionary Autonomous Navigation System, *In Proceedings of CEC´02*, Honolulu, USA, 2002, 466 – 471.

[8] Cazangi, R.R., Von Zuben, F.J. and Figueiredo, M.F. A Classifier System in Real Applications for Robot Navigation. *In Proceedings of the CEC´03*, Canberra, Australia, 2003, (1) 574 – 580.

[9] de Castro, L.N. and Von Zuben, F.J. Biologically Inspired Collective Robotics. In L.N. de Castro & F.J. Von Zuben, *Recent Developments in Biologically Inspired Computing*, Idea Group Inc., Chapter 1, 2004.

[10] França, F.O., Von Zuben, F.J., and de Castro, L.N. Definition of Capacitated p-Medians by a Modified Max Min Ant System with Local Search. *In Proceedings of 11th International Conference on Neural Information Processing*, Calcutta, India, 2004, 1094-1100.

[11] Ding, Y., He, Y. and Jiang, J. Multi-Robot Cooperation Method Based on the Ant Algorithm. *In Proceedings of the 2003 IEEE Swarm Intelligence Symposium*, Indianapolis, USA, 2003.

[12] Holland, J.H. Escaping Brittleness: The Possibilities of General-Purpose Learning Algorithms Applied to Parallel Rule-Based Systems. *In Michalsky, R.S., Carbonell, J.G. and Mitchell, T.M., (eds.), Machine Learning: An Artificial Intelligence Approach*, Morgan Kaufmann, 1986.

[13] Kube, C.R., Parker, C.A.C., Wang, and T., Zhang, H. Biologically Inspired Collective Robotics. In L.N. de Castro & F.J. Von Zuben, *Recent Developments in Biologically Inspired Computing*, Idea Group Inc., Chapter 15, 2004.

[14] Sherafat, V., de Castro, L.N. and Hruschka, E.R. TermitAnt: An Ant Clustering Algorithm Improved by Ideas from Termite Colonies. *In Proceedings of 11th International Conference on Neural Information Processing*, Calcutta, India, 2004, 1088-1093.

[15] Svennebring, J. and Koenig, S. Building Terrain-Covering Ant Robots. Technical Report GIT-COGSCI-2002/10, College of Computing, Georgia Institute of Technology, Atlanta, USA, 2002.

[16] Wagner, I.A. and Bruckstein, A.M. Cooperative Cleaners: A Case of Distributed Ant-Robotics. In *Communications, Computation, Control, and Signal Processing: A Tribute to Thomas Kailath*. Kluwer Academic Publishers, The Netherlands, 1997, 289 – 308.

[17] Watabane, Y., Ishiguro, A. and Uchikawa, H. Decentralized Behaviour Arbitration Mechanism for Autonomous Mobile Robot Using Immune Network, *In D. Dasgupta (Editor), Artificial Immune Systems and their Applications*, Springer, 1999.