The Concept of Robot Society: Evidence Taken From Multi-Robot SLAM

Micael S. Couceiro[1], Andria R. Lopes[2],
N. M. Fonseca Ferreira[3], Anabela G. Ferreira[4], Rui Rocha[1]

[1] Institute of Systems and Robotics, University of Coimbra,
Pólo II, 3030-290 Coimbra, Portugal
{micaelcouceiro,rprocha}@isr.uc.pt

[2] Faculty of Economics of Coimbra, University of Coimbra
Av. Dias da Silva 165, 3004-512 Coimbra, Portugal
andilope@gmail.com

[3] Department of Electrotechnics Engineering,
Coimbra Institute of Engineering, Portugal
nunomig@isec.pt

[4] Coimbra School of Education,
Pólo I, Praça Heróis do Ultramar, Solum, 3030-329 Coimbra, Portugal
anabelagoncalves@esec.pt

Abstract — *In this paper, it is presented the significance and the advantages of cooperation in the different societies making the analogy in the concept of robot society.*

In order to compare the advantages of cooperative robots over a single robot, it is considered essential the development of computational simulation based on the robotic cooperation in unstructured environments. It is implemented a Multi-Robot Simultaneous Localization and Mapping (SLAM) using Rao-Blackwellized particle filter [1] in a simulation environment developed in the Player / Stage platform for robot and sensor applications.

Keywords — *robot, society, cooperation, multi-robot slam*

1 Introduction

The concept of robot society soon appeared showing the inherent advantages when comparing to single solutions [2]. Since societies are formed as collaborative structures to execute tasks which are not possible or are difficult for individuals alone, having societies formed by robots would bring at least two advantages: fault tolerance and parallelism.

At first glance, having multiple robots performing a task or a set of common tasks may seem more problematical (and challenging) than useful. Why not use a single and complex robot capable of performing all these tasks?

The answer is all around us in nature. Much of the work developed in the area of cooperative robots mention biological systems as a source of inspiration. The collective behavior of ants, bees, birds and other similar societies provide strong evidence that systems composed of simple agents can perform complex tasks in the real world. The robustness and adaptability of biological systems represent a powerful motivation to replicate these mechanisms in an attempt to generate software and hardware with features comparable to those of biological systems. These and many other reasons will be addressed in this study showing the benefits of cooperative robots over a single robot.

Cooperative robots, or Multi-Robot Systems (*MRS*), describe the situation in which a group of robots get an overall benefit. A first key issue in cooperation is whether robots should be identical (homogeneous groups) or different (heterogeneous grouping) and if the efficiency should come into consideration for the performance of the whole group or only to each individual robot. This kind of cooperation in robotics can vary from having only two robots to perform a simple task together (*e.g.*, two industrial arms manipulating a large object) [3] to a group of heterogeneous robotic agents that can connect and form a more complex structure [4]. More recently, some studies have been focused interest in *MRS* incorporating algorithms of localization and mapping [5] thus enjoying all the advantages of the cooperation between robots.

Many applications in robotics, such as search and rescue, surveillance, exploration, among others, require the exact location in unknown environments. When robots are operating in unstructured environments, in order to obtain their exact location, we need to create and analyze the map of the environment. The concept of robot society will show us the improvements of systems that require robots to operate in unstructured environment.

Section two highlights the importance of cooperation in societies. In section three we present the state of the art in the area of robotics, focusing on cooperation and sociological systems. Section four gives a brief survey of Simultaneous Localization and Mapping (*SLAM*) applied to single robots and multiple robots and in order to demonstrate the advantages of cooperative robots over a single robot, a Multi-Robot *SLAM* algorithm inspired in the work of Andrew Howard [1] is implemented in section five using the *Player / Stage* platform for robot and sensor applications. Finally, in section six outlines the main conclusions.

2 Cooperative Systems

Thousands of years ago, the King Solomon, who was a student of the nature, observed the humble ant, and wrote: "Go to the ant, you sluggard; consider its ways and be wise! It has no commander, no overseer or ruler, yet it stores its provisions in summer and gathers its food at harvest." [6]. In fact the ants are a perfect example of cooperation, diligence and order. In addition to work together and help each others, the ants seem to be able to find their paths (the nest to a food source and back or just getting around an obstacle) with relative ease, despite being virtually blind. Several studies have found that in many cases this capacity is the result of the interaction of chemical communication between ants (for a substance called pheromone) and emergent phenomena caused by the presence of many ants. This is the concept of stigmergy [7]. This mechanism is so efficient that there are algorithms that use this principle as is the case of the heuristic principle Ant System that simulates the behavior of a group of ants that work together to solve an optimization problem using a simple communications [8] and the case of Brood Sorting (group selection) used in swarms of robots [9].

Another very similar principle can be seen in other optimization algorithms such as genetic algorithms, evolutionary strategies and the well known *PSO* (Particle Swarm Optimization) initially proposed by Kennedy and Ebarhart [10], based on the behavior of social organisms such as birds or fishes. On cooperation and competition among the potential solutions, the optimal complex problems can be achieved more quickly. In PSO algorithms each individual of the population is called a particle and the position of these individuals is modified over time. Thus, the particles wander through the multidimensional search space. Along the way, each particle adjusts its position according to their experience and the experience of the other members of the population, taking advantage of the best position of each particle and the best position of the whole group.

Suppose the following scenario: a group of birds are randomly looking for food in an area where there is only one type of food. Although birds don't know where the food is, they know how close to the food they are at each iteration. So what is the best strategy to find the food? The most efficient one is to follow the bird that is closer to the food.

The *PSO* has been successfully used in many applications such as robotics [11][12][13] and electrical systems [14].

Another interesting engineering example based on biological cooperation is reflected in the flight of pelicans. Researchers discover that the pelicans that fly in formation earn extra boost when compared to the ones flying forward, resulting in a 15% reduction in the heart rate. In order to validate this concept, a group of engineers prepared a flight test with electronic equipment that enabled the pilot to keep the plane at a distance of 90 meters (with a small tolerance of 30 centimeters) over the plane that was ahead. What was the outcome? The plane suffered an air resistance 20% lower and it consumed 18% less fuel. These results can be used on military or civilian planes, but also in the concept of robotics to improve the dynamics of flying robots to monitor forest fires [15] or biologically inspired robots for spying [16].

However, when we speak about cooperation we should say Cooperative Systems. The cooperation is just one of the indispensable tools for the Cooperative Systems since without the collaboration between different members of a particular group or society Cooperative Systems cannot survive. On the other hand, to cooperate, the communication is essential between group members and this communication must be familiar to all of them. The coordination also plays an important tool in cooperative systems, since it organizes the group to prevent that communication and cooperation efforts are lost and that tasks are performed in the correct order, at the correct time and meeting the constraints and objectives.

The Cooperative Systems has been studied in several areas including computer science [17] and [18] and robotics [19], [20] and [5].

Inspired by the results of the existing cooperation in various societies (*e,g.*, ants, bees, plants, humans), researchers have placed a great emphasis on developing robots that can cooperate with each other and perform multiple tasks.

The Cooperative Multi-Robot Systems (*CMRS*) are based on the interception of the contribution of each member (*i.e.*, robot): if we have a group of robots cooperating to perform a given task, they need to communicate with each other in order to coordinate their actions and obtain the desired result. This concept offers a countless number of advantages similar to the benefits of Cooperative Systems in other societies that may be described in the following key factor: time. One way to circumvent the limitations inherent to the concept of time is to perform simultaneous procedures: if we have multiple robots instead of one they can act on multiple places at the same time (spatial distribution) and they can perform multiple tasks simultaneously (temporal distribution).

3 Multi-Robot SLAM

The search for a solution to the *SLAM* problem has been one of the notable successes of the robotics community over the past decade. The *SLAM* has been formulated and solved as a theoretical problem in a number of different forms being implemented in a number of different domains from indoor robots to outdoor, underwater, and airborne systems.

Basically, *SLAM* is a process by which a mobile robot can build a map of an environment and at the same time use this map to deduce its location. So, in a probabilistic form, the *SLAM* problem requires that the probability distribution (1) be computed for all times k.

$$P(x_k, m \mid Z_{0:k}, U_{0:k}, x_0) \tag{1}$$

This probability distribution describes the joint posterior density of the landmark locations and vehicle state (at time *k*) given the recorded observations and control inputs up to and including time k together with the initial state of the vehicle.

The *SLAM* approach for a single robot began to receive attention in 1990 [21]. The majority of the solutions to the *SLAM* problem are based on the implementation of the extended Kalman filter (*EKF*) that correlates the pose estimation relative to different landmarks [22] [23].

Although the *EKF* is one of the most effective approaches for map estimation, [24] proved that the *FastSLAM* performance was substantially higher than those obtained by the *EKF*. The *FastSLAM* algorithm was used for the construction of indoor maps in [25] and [26]. They used an algorithm based on occupancy grids in order to build a metric map of the environment.

A variant of the *FastSLAM* was proposed [27] combining the Rao-Blackwellized particle filter (*RBPF*) for samples of the trajectory of the robot and an *EKF* to represent the map. This algorithm contains many elements of the standard Monte-Carlo localization algorithm [28]. The challenge lies in maximizing the per-particle update speed while minimizing the corresponding storage requirements, so that the filter may run in real time and in bounded memory with a relatively large number of particles. As always, the speed and storage demands tend to conflict, and our implementation favors the former over the latter.

Based on the previous single robot *SLAM* algorithm Andrew Howard developed a similar algorithm applied to multiple robots [1].

This algorithm has two important elements:

1. robots are able to detect, identify and measure the relative pose of other robots at some time during the exploration task (when those robots are both nearby and within line-of-sight, for example). Such encounters allow robots to fuse their subsequent observations into a common map, using the measured relative pose to initialize the

filter (note, however, that only the first such encounter is used; subsequent encounters between robots are ignored);

2. the particle-filter based *SLAM* algorithm supports time-reversed updates; this generalization allows robots to incorporate observations that occurred prior to the first encounter, by treating those observations as if they came from additional "virtual" robots travelling backwards in time.

As an illustration, consider the following example: two robots are exploring an environment from distant and unknown initial locations. When robots encounter one another they measure their relative pose constructing a filter in which robot 1 has an initial pose of zero, and robot 2 has the measured relative pose. Subsequent measurements from the two robots are fed to the filter, and thereby fused into a common map. At the same time, two virtual robots are added to the filter with poses initialized as above where the previously recorded measurements are fed to the filter in reverse time-order, such that these virtual robots appear to be driving backwards through the environment. Thus, the filter incrementally fuses data from both robots, recorded both before and after the encounter, into a single map.

Let Δ_s^2 denote the relative pose of robot 2 as measured by robot 1 at time s. We wish to estimate the posterior over maps and trajectories given by:

$$
\begin{aligned}
p(x_{1:t}^1, x_{s+1:t}^2, m \mid z_{1:t}^1, u_{0:t-1}^1, x_0^1, z_{s+1:t}^2, u_{s:t-1}^2, \Delta_s^2) = \\
p(m \mid x_{1:t}^1, z_{1:t}^1, x_{s+1:t}^2, z_{s+1:t}^2) \qquad (2) \\
p(x_{1:t}^1 \mid z_{1:t}^1, u_{0:t-1}^1, x_0^1) p(x_{s+1:t}^2 \mid z_{s+1:t}^2, u_{s:t-1}^2, x_s^1, \Delta_s^2)
\end{aligned}
$$

where $x_{1:t}^1$ and $x_{s+1:t}^2$ denotes a sequence of robot 1 and 2 poses at times 1; 2; ... ; t, and s+1; s+2; ... ; t, respectively. $z_{1:t}^1$ and $z_{s+1:t}^2$ denotes the corresponding sequence of observations, and $u_{0:t-1}^1$ and $u_{s:t-1}^2$ denotes the sequence of actions executed by the robots (Fig. 1).

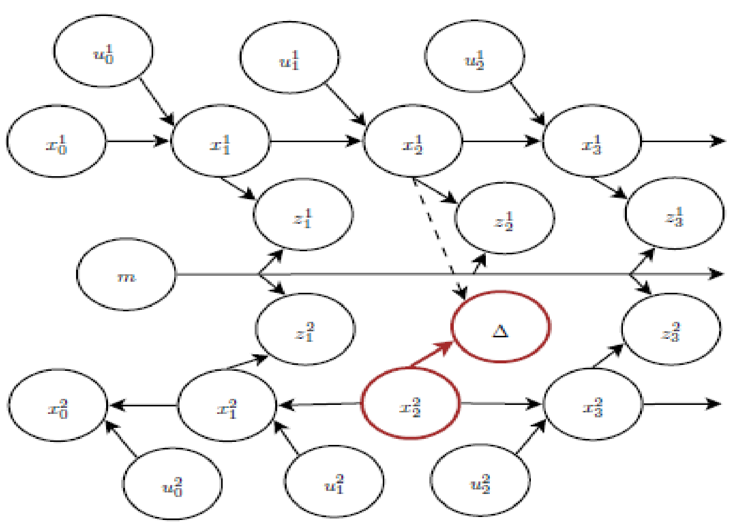

Figure 1: Bayes net for multi-robot SLAM with unknown initial poses. The robots first encounter one another at time s, recording the relative pose Δ_s^2.

This algorithm has number of attractive features. First, it is able to fuse all data from all robots into a single map, without knowing the initial robot poses. Second, it inherits the bounded-time, bounded-memory properties of the single robot *SLAM* algorithm (CPU and memory requirements do not increase with path length). Third and finally, the algorithm is fast: our implementation can fuse data from two robots in real time. Collectively, these features make the algorithm highly suitable for on-line, in-the-loop applications, such as multi-robot exploration and search tasks.

4 Experimental Results

In order to demonstrate the advantages of cooperative robots over a single robot, we implemented a single and Multi-Robot *RBPF-SLAM* algorithm in the *Player* / *Stage* platform based on the work of Andrew Howard [1].

The filter update step requires two ray-tracing operations on the occupancy grid for each and every particle: one to evaluate the sensor model and another to update the map. Since these operations are expensive, we approximate the ray-tracing step by considering only the ray end-points, and decimate the laser scans by using only one scan for every 0.50 m of distance traveled. These approximations improve processing speed by an order of magnitude or more, thereby allowing real-time operation. For each particle, we maintain a complete occupancy grid map, generally with a resolution of 0.50 m and covering an area of between 2000 and 8000 m^2.

The robots used are the *Pioneer II* with odometry and 2D laser (horizontal plane) with 1º of resolution and retro-reflective markers (for mutual recognition).

Fig. 2 shows a typical map generated by the single-robot algorithm, with all three loops closed correctly. Processing time for this map is 126 s on a 1.6GHz *Intel Centrino* using 150 particles.

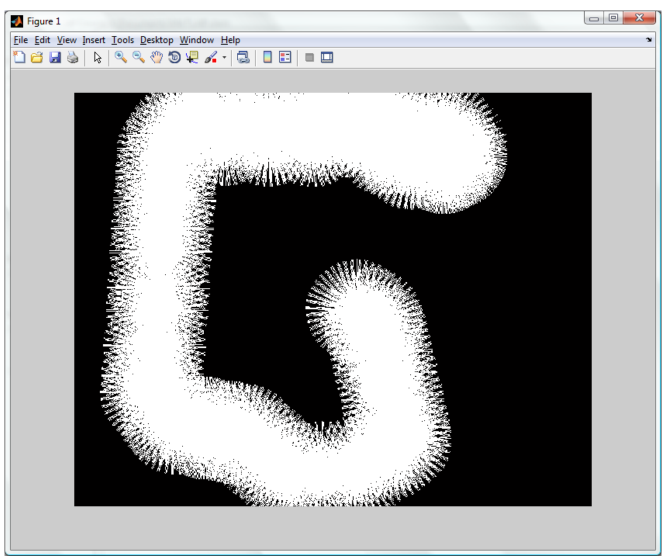

Figure 2: Map generated using the single-robot algorithm; the map is 16 m by 16 m with a resolution of 0.50 m.

Fig. 3 shows the results produced by the multi-robot algorithm for an autonomous exploration task. Two robots were deployed into this environment at distant locations, from which they executed a cooperative, but largely reactive, exploration strategy.

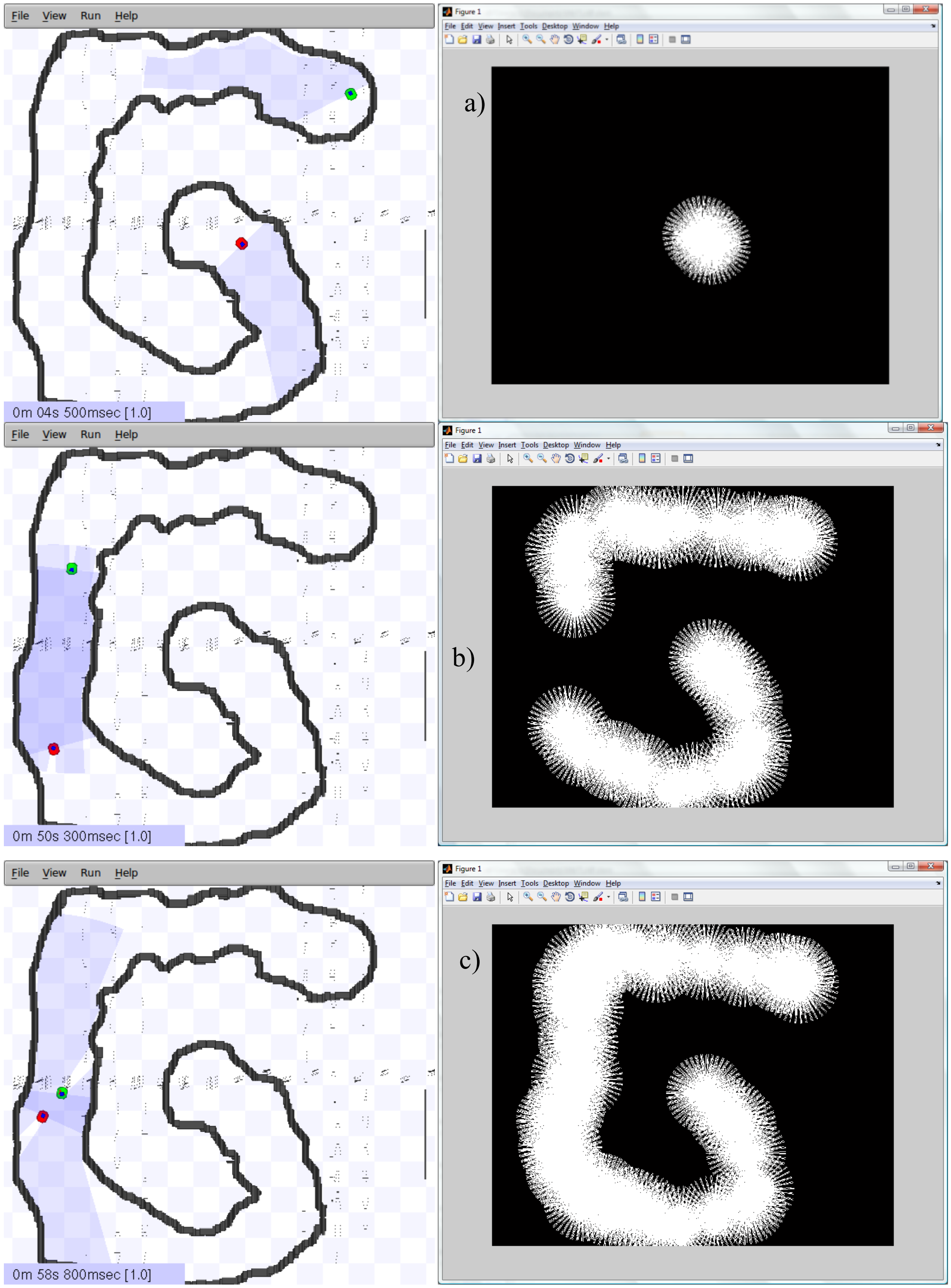

Figure 3: Sequence of events events: a) robots starts at distant locations and the global map begins being generated by the robot 1 (red) considering its initial position zero; b) robot 1 (red) encounters robot 2 (green) (due to the retro-reflective markers) at time t = 54 seconds and uses the combined information adding it to the global map; c) at time t = 62 seconds the entire map is obtained with a resolution of 0.50 m.

In the final map all the major topological features have been properly extracted and the map quality is uniformly high.

The processing time for this map is 62 seconds on a computer 1.6GHz *Intel Centrino* using 150 particles.

5 Conclusions and Discussion

The use of cooperative strategies in robotics offers several attractive features since robots are constantly interacting and communicating with each other with the dynamic environment and with other members of different societies (*e.g.*, man).

The collective intelligence emerging from cooperative strategies in robotics gives a reason to call these systems, at their highest level, as robot society. The concept of robot society shows potential in applications where the space and time distribution of single robots are restricted and also as an alternative to more complex robots.

To demonstrate possible advantages of the cooperation in robotics, a single and a multi-robot *SLAM* algorithm based on a *RBPF* was implemented on the *Player* / *Stage* platform. One of the attractive features of the multi-robot SLAM algorithm is that it's easy to implement after the implementation of the single robot algorithm. The basic elements of the algorithm - the sensor and action models, occupancy grids and ray-tracing - are easily adapted from the Monte-Carlo location algorithm. Despite possible improvements to this algorithm as discussed in [1], our results show that a cooperative exploration strategy becomes far superior to the individual one. The processing time of the map for the single-robot solution is greater than twice the processing time of the two-robots solution.

References

[1] Howard, A.: Multi-robot Simultaneous Localization and Mapping using Particle Filters. Int. Journal of Robotics Research, 25(12):1243-1256, 2006.

[2] Halme, A., Jakubik, P., Schönberg, T., Vainio, M.: The Concept Of Robot Society And Its Utilization. Proceedings of IEEE Int. Workshop on Advanced Robotics, 1993.

[3] Ferreira, N. M. F.: Sistemas Dinâmicos e Controlo de Robôs Cooperantes. The Phd thesis, in 5 of September, University of Trás-os-Montes e Alto Douro, 2006.

[4] Fukuda, T., Nakagawa, S., Kawauchi, Y., Buss, M.: Structure Decision Method for Self Organizing Robots Based on Cell Structures - CEBOT. Proc. of IEEE Int'l on Robotics and Automation, pp. 695-700, 1989.

[5] Rocha, R.: Building Volumetric Maps with Cooperative Mobile Robots and Useful Information Sharing: a Distributed Control Approach based on Entropy. PhD thesis, Faculty of Engineering of University of Porto, Portugal, May, 2006.

[6] Dean, M: Book of Proverbs. Catholic Encyclopedia. Adapted from Holman Bible Handbook on Proverbs, 1913.

[7] Aras, R., Dutech, A., Charpillet, F.: Stigmergy in Multi Agent Reinforcement Learning. Loria, Inst. Nat. de Recherche en Inf. et Autom., Nancy, France, 2004.

[8] Dorigo, M., Stützle, T.: Ant Colony Optimization. MIT Press, Cambridge, MA, 2004.

[9] Wilson, M., Melhuish, C., Sendova-Franks, A., Scholes, S.: Algorithms for building annular structures with minimalist robots inspired by brood sorting in ant colonies. Autonomous Robots, 17(2-3):115-136, 2004.

[10] Kennedy, J., Eberhart, R.: A new optimizer using particle swarm theory. Proceedings of the IEEE Sixth International Symposium on Micro Machine and Human Science, pp. 39-43, 1995.

[11] Tang, J., Zhu, J., Sun, Z.: A novel path panning approach based on appart and particle swarm optimization. Proceedings of the 2nd International Symposium on Neural Networks, LNCS 3498, pp. 253-258, 2005.

[12] Pires, E. J. S., Oliveira, P. B. M., Machado, J. A. T., Cunha, J. B.: Particle Swarm Optimization versus Genetic Algorithm in Manipulator Trajectory Planning. 7th Portuguese Conference on Automatic Control, September 11-13, 2006.

[13] Couceiro, M. S., Mendes, R., Ferreira, N. M. F., Machado, J. A. T.: Control Optimization of a Robotic Bird. EWOMS '09, Lisbon, Portugal, June 4-6, 2009.

[14] Alrashidi, M. R., El-Hawary, M. E.: A Survey of Particle Swarm Optimization Applications in Power System Operations. Electric Power Compon Syst. v34 i12. 1349-1357, 2006.

[15] J. R. Martínez-de Dios, L. Merino, F. Caballero, A. Ollero and D. X. Viegas: Experimental Results of Automatic Fire Detection and Monitoring with UAVs. V International Conference on Forest Fire Research, 2006.

[16] Couceiro, Micael S., Figueiredo, Carlos M., Ferreira, N. M. Fonseca, and Machado, J. A. Tenreiro: Biological Inspired Flying Robot. Proceedings of IDETC/CIE 2009 ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference August 30 - September 2, San Diego, 2009.

[17] Borghoff, U.M., Schlighter, J.H.: Computer-Supported Cooperative Work: Introduction to Distributed Applications. Springer, USA, 2000.

[18] Fuks, H., Raposo, A. B., Gerosa, M. A., Lucena, C.J.P.: O Modelo de Colaboração 3C e a Engenharia de Groupware. Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio), 2003.

[19] Cao, Y., Fukunaga, A., Kahng, A.: Cooperative Mobile Robotics: Antecedents and Directions. Autonomous Robots, 4:1-23, 1997.

[20] Jung, D.: An Architecture for Cooperation among Autonomous Agents. Ph.D. thesis, Dep. of Computer Science, University of Wollongong, Australia, 1998.

[21] R. Smith, M. Self, and P. Cheeseman: Estimating Uncertain Spatial Relationships in Robotics. Autonomous Robot Vehicles, pp. 167–193, 1990.

[22] M. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba: A solution to the simultaneous localization and map building (SLAM) problem. IEEE Trans. on Robotics and Automation, 17(3):229–241, 2001.

[23] S. Thrun, D. Hahnel, D. Ferguson, M. Montermelo, R. Riebwel, W. Burgard, C. Baker, Z. Omohundro, S. Thayer and W. Whittaker: A System for Volumetric Robotic Mapping of Underground Mines. In Proc. of IEEE Int. Conf. on Robotics and Automation, 2003.

[24] M. Montemerlo and S. Thrun: Simultaneous localization and mapping with unknown data association using FastSLAM. In Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA), Taipei, Taiwan, 2003.

[25] Cyrill Stachniss, Dirk Hahnel and Wolfram Burgard: Exploration with Active Loop-Closing for FastSLAM. Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2004.

[26] Cyrill Stachniss, Giorgio Grisetti and Wolfram Burgard: Recovering Particle Diversity in a Rao-Blackwellized Particle Filter for SLAM After Actively Closing Loops. Proc. of IEEE Int. Conf. on Robotics and Automation, 2005.

[27] D. Hahnel, W. Burgard and D. Fox: An efficient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements. IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, Nevada, U.S.A, Oct, 2003.

[28] S. Thrun, D. Fox and W. Burgard: Robust Monte Carlo localization for mobile robots. Artificial Intelligence Journal, vol. 128, no. 1–2, pp. 99–141, 2001.