A realistic approach to source localization using a wireless robotic network

Christos Christodoulopoulos*, Christos Kyriakopoulos† and Athanasios G. Kanatas‡
Wireless Communication Laboratory
Department of Technology Education and Digital Systems
University of Piraeus
Email: *christos.c@ieee.org, †c.kyriakopoulos@ieee.org, ‡kanatas@unipi.gr

***Abstract*—This paper investigates the problem of target localizing by a communicating robotic swarm in an unknown environment. Robots have to collaboratively search for the target while avoiding obstacles in their way. Emphasis is given on how physical constraints such as obstacles and communication links affect the swarm's operation. Finally, simulation results of the proposed system in a small scale area are presented and evaluated and possible uses of the system are discussed.**

I. Introduction

In this paper we discuss the subject of source localization with the use of a wireless robotic network. A wireless robotic network is a swarm of mobile sensors that communicate to exchange information about their state in order to make a collective decision. Our objective is for the robotic sensors (from now on called agents) to decide on how to locate and converge to a source. A source can be anything which creates a field that can be measured with wireless sensors.

Similar work on the field includes simulations with various techniques to process the exchanged data such as methodologies on Kalman filters [1], graph searching algorithms [2] and variations of swarm optimization algorithms [3] [4].

However none of these approaches considers real life constraints that would apply to such an application. In real life application the agents would have to avoid physical obstacles that may lie in their way and still operate effectively while searching for the source. Moreover these obstacles may be blocking communication between the agents of the swarm, thus affecting its collaborative ability.

In this paper we will examine how such constraints affect the robot swarm in its goal to locate and converge to a source. In our application collective decision-making is achieved through the Particle Swarm Optimization algorithm (PSO). This algorithm is used to adjust the movement of every agent by processing data from all the agents of the network including itself.

The required data is transmitted via a wireless sensor network. Wireless Sensor Networks is an established technology, used mainly to collect and process field data. It consists of sensors which collect environmental variables and have communications infrastructure that enables them to transmit their readings to one another. The usual purpose of the network is to route all data to a data sink which will then process the data.

As mentioned above, an agent moving towards the source may find an obstacle blocking its way. Every robotic agent is equipped with sensors that allow it to scan the path of the motion. When an obstacle is sensed the agent executes an obstacle-avoidance sub-routine. Once the obstacle has been surpassed, the agent continues to move towards its initial destination (i.e. the source).

The above described system has been simulated in Java environment and our initial results are presented. A number of real-life problems can be addressed by implementing our system. These include military targets and natural emissions (i.e. heat, radiation etc.) localization, among others and will be described further on.

II. Particle Swarm Optimization Algorithm

The Particle Swarm Optimization (PSO) algorithm is part of the Swarm Intelligence algorithms family which is a form of Collective Intelligence [5]. It was initially proposed by Kennedy as a method of social behavior simulation and was established as an optimization algorithm in 1995 [6].

The idea of the method is that a swarm of agents tries to find the optimum value in a searching space by measuring the values of the agents' positions. For our application we will use a version of the algorithm known as PSO Type 1' [7]. Its differentiation from the classic algorithm is the addition of some coefficients (described later on) that help the swarm converge to a solution more efficiently.

The algorithm operates as follows: Every individual agent has a position vector ($\vec{x}$) and an adaptive velocity vector ($\vec{u}$) in the search space. Furthermore, each agent has memory of the position of its best measurement ($\vec{p}_i$) so far and acquires the position of the best measurement ($\vec{p}_g$) made throughout the swarm. The agent's movement is an acceleration vector with a direction that derives from the combination of its personal best position along with the swarm's best position. The acceleration's norm derives from the combination of the agent's current velocity vector in conjunction with the vector calculated by the algorithm.

Random factors $\phi_1, \phi_2 \in [0, \phi_{max}]$ are used to add some fuzziness to the acceleration vector, thus serving for better searching of the space. The final equation is:

$$\vec{u}_i^{n+1} = \chi\left(\vec{u}_i^n + \phi_1^n(\vec{p}_i^n - \vec{x}_i^n) + \phi_1^n(\vec{p}_g^n - \vec{x}_i^n)\right) \tag{1}$$

$$\vec{x}_i^{n+1} = \vec{x}_i^n + \vec{u}_i^{n+1} \tag{2}$$

where n is the current algorithm step and χ is a positive number called global constriction coefficient that makes the system to be non divergent and is represent by:

$$\chi = \begin{cases} \frac{2\kappa}{\phi - 2 + \sqrt{\phi^2 - 4\phi}} & \text{if } \phi > 4 \\ \sqrt{\kappa} & \text{else} \end{cases} \quad (3)$$

where $\kappa \in [0, 1]$ with smaller values resulting in faster convergence and $\phi = \phi_1 + \phi 2$.

The efficiency of the PSO algorithm has been tested thoroughly by optimization theorists using standard optimization test functions such as Rosenbrock, Griewangk and Rastrigin [7].

We performed further 'test runs' in Matlab using the $|sin(x)/x|$ function. The PSO's efficiency in the $|sin(x)/x|$ function was over 95% when used with more than 8 agents, whereas with more than 10 agents the efficiency rose up to 99%.

III. Communication Network

As described above, the swarm needs to exchange information between all its individual agents, for the searching algorithm to execute. To achieve that, a wireless communication network needs to be established.

The structure of the network is time variant and unstable, because the agents will constantly move and may become unreachable at some point. They operate in an environment with high uncertainty and have very little information about their surrounding space. Therefore the communication network has to be a decentralized, ad-hoc network.

For our application we propose the use of sensors which are used in wireless sensor networks and are capable of measuring and transmitting data. Wireless sensor networks use various techniques to establish an appropriate topology to route data through the network, usually with the purpose of collecting this data to a base station for further processing. In our case, we don't need a centralized data collection, but for every single agent to individually collect all the data transmitted. Furthermore we cannot create routing tables, since the agents constantly move in unequal distances and there isn't any static point of reference. Therefore we use the typical setup of wireless sensor networks and utilize agents with omnidirectional antennas broadcasting their data to every other agent in the network using time division multiple access (TDMA).

The nature of the application is tolerable to communication links being broken for a couple of the PSO algorithm's steps. Every agent uses as much information as it can gather from the rest of the swarm in order to make its decision about its movement. An agent may for some reason become permanently disconnected to the network, thus leaving the swarm; this however will have no effect on the rest of the network that will keep operating normally, broadcasting their data. Information gathered from fewer agents though, will result in poorer decision making by the PSO algorithm, but this has minor overall impact, as it has been observed during the simulation.

IV. Robotics

Every agent of the swarm consists of a robotic part which enables it to perform both the motion and the obstacle avoidance functions. The robots that implement the swarm system can be of any type, as long as they provide the sensors with the above mentioned capabilities. In our application, we use a basic wheeled robot equipped with ultrasonic sensors. For the time being our research is limited to surface robots, even though the same algorithmic principles can be extended to aerial or aquatic robots.

One fundamental task of the robotic section is self localization. In order for the PSO algorithm to operate the agents must know both their positions in space (2D points, for simplicity reasons) and the position of every other agent. One commonly used method, for this problem is GPS navigation but given the fact that in our application the agents operate in an indoor environment, this method is inappropriate. Another solution involves guiding beacons (using reflectors and laser rangefinders or RF technologies), which are pre-set in the area. In our application, however the agents must operate in unknown areas, so it is not possible for beacons to exist in the environment. The solution that we propose is the establishment of a distributed global coordination system. During an initialization phase all agents start from the same point in the area and they start to move towards a randomly selected position. As every agent moves, they know their absolute position (in relevance with the starting point) and their relative position to the other agents (provided they can communicate).

Another issue that arises when dealing with robotics is motion, in the sense of the equations that describe the direction and the position of the robot (reverse kinematics).

Due to the complex nature of reverse kinematics we use a simplified set of motion equations that describe the motion as a combination of straight motion and non-moving rotation around the vertical axis [8].

Another aspect of the robotic motion is speed. Or, to be more exact, the time needed to perform one step of the algorithm. In the standard PSO algorithm there is no limit for each agent's speed (beside the global constriction coefficient). However, in a real world application we have to work with agents that might be somehow restricted concerning their speed. For this reason we insert a timeout limit in the main algorithm. This limit forces the agents to perform each step of the algorithm within a certain time frame. When the timeout occurs, the current algorithm step stops and the next one begins. The speed of each agent is determined accordingly. After calculating the new position, the agent adjusts its speed in order to reach the position within the given time frame.

One final issue that needs to be addressed is obstacle avoidance. By obstacle we define any static object of a certain size (larger than ~ 60cm., e.g. walls, large furniture). In our application the obstacle detection system consists of three ultrasonic sensors placed at 90° intervals. The obstacle avoidance routine is divided into two sub-routines: "obstacle detection" and "obstacle avoidance". During the detection subroutine the

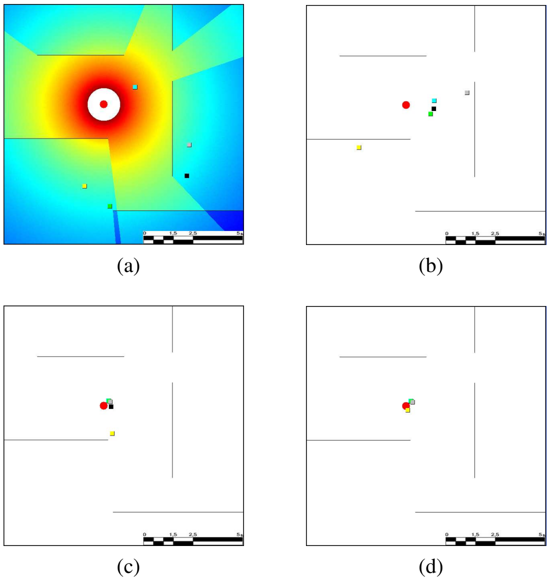

Fig. 1: Simulation in indoor environment with 5 agents. The source is located in the middle of the space. Pictures (a) to (d) depict the time progression at steps 1, 5, 10 and 15 respectively.

ultrasonic sensor scans the area in front of the robot. After each scan the sensors return a distance measurement. Normally the measurement should be the maximum scanning distance of the sensor. An obstacle is detected when the sensor returns a measurement smaller than the previous and at the same time smaller than a predefined threshold [9].

When an obstacle is detected the agent passes into the obstacle avoidance subroutine. First the agent must position itself and start moving parallel to the obstacle. Then the side sensor scans the obstacle as the agent moves. If the sensor detects an opening that is large enough for the agent, the agent turns and moves through the opening. Finally the agent returns to its previous state and continues towards its original destination. During the obstacle avoidance subroutine the front sensor continues to operate in case there is another obstacle in front of the agent. In such cases (for instance a wall corner), the agent turns at the opposite direction and continues the same subroutine.

We must point out that the same timeout limit that exists for the agents' motion applies to the obstacle avoidance routine as well. Thus, if an agent cannot avoid an obstacle within the specified time frame, the obstacle avoidance routine stops and the agent returns to the execution of the next step of the main algorithm.

V. Simulation and Results

Having established the fundamentals of the PSO Algorithm, the Robotic Swarm and the Communication Network, we examined the "proof of concept" of the system by developing a simulation tool. The tool simulates the behavior of the swarm in an indoor environment assigned to the task of locating a radio antenna.

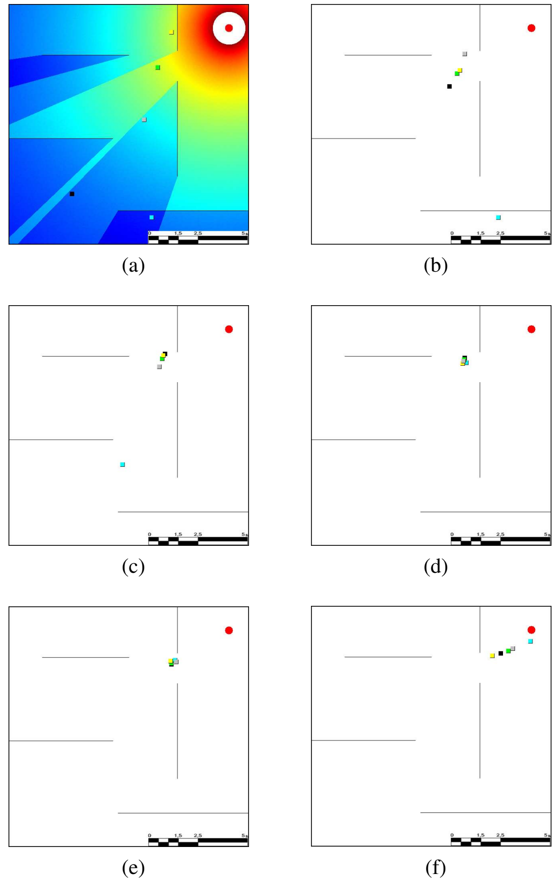

Fig. 2: Simulation in indoor environment with 5 agents. The source is located at the upper right corner of the space making the localization even more difficult. Pictures (a) to (f) depict a 5 step-at-a-time progression. Some agents fail to reach the source within the 30 step limit (picture f).

The purpose of the simulation was to evaluate the effectiveness of the basic PSO algorithm (along with some alterations which are presented later on) along with any constrains imposed by the indoor environment, the network communication and the robotic motion. At this point we overlook the issues concerning the field measurement (i.e. the sensing) -assuming that every agent is able to collect a field measurement at any time. One final concession has been made concerning the issue of localization, where we assume that the agents use a global coordinate system and are able to know their absolute position at any time.

The main algorithm that the agents follow during the simulation is the PSO Type 1' although there are some changes in our implementation. First, we have inserted a time frame within which, a step of the algorithm is executed. The duration of the time frame is associated with the speed of the actual robots and the dimensions of the environment. It is calculated as the time needed for a robot to traverse through $1/6^{th}$ of the operating space. This distance is considered to be the length of a typical large-scale obstacle (e.g. a wall) that the agent needs to overcome within the time frame. Another

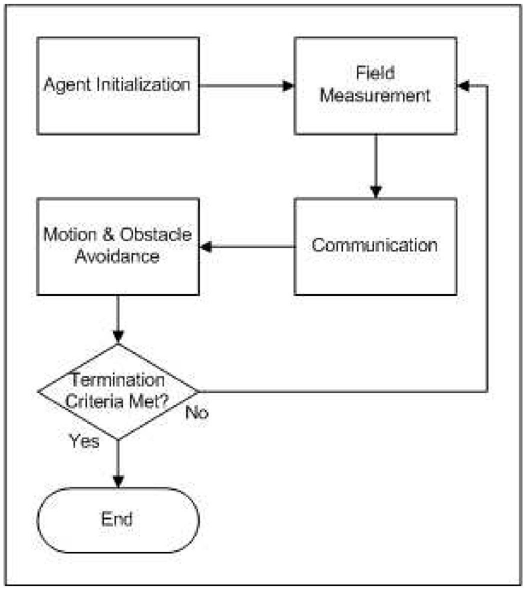

Fig. 3: PSO Algorithm Flowchart

modification deals with the obstacle avoidance subroutine (as described in a previous section). While moving during the execution of the main algorithm, an agent may encounter an obstacle that it must surpass. When such an event occurs the main algorithm is interrupted and the agent enters the obstacle avoidance subroutine. When the current time frame ends the agent returns to its previous state, whether it has surpassed the obstacle or not.

Another tweak is about the data exchanged. Instead of the agents transmitting their best measurements according to the PSO algorithm, they transmit their local copy of the best swarm value measured. This results in best values propagating to agents that cannot communicate directly with each other, thus counter-balancing the degradation of the system's efficiency due to communication failures. Fig. 3 depicts the modified PSO algorithm flowchart.

Finally the simulation provides a realistic propagation model of the electromagnetic waves transmitted by the antenna. For this purpose we use a set of alternative indoor propagation models. In particular we use the COST-231 Multi Wall Model [10], the Keenan-Motley Model [11], the Single Slope Indoor Model [10] and the Linear Attenuation Model [10].

Based on this simulation tool, we performed a series of trials using the following set of parameters which correspond to real-life scenarios. The simulated environment was a 12 by 12 meters room, with an antenna source transmitting with 1W power at 2.4GHz and the COST 231 Multi-Wall Model with thin concrete (interior) walls for both the agents' signal propagation simulation and the source cost function. The swarm comprised of five agents, which is a fairly small number compared to typical swarm sizes of the PSO algorithm mathematical model. Standard PSO parameters values of κ and ϕ_{max} (0.5 and 4.1 respectively) [7] were used. The agents broadcasted at 1.0 mW with a sensing sensitivity of -94dBm [12]. The results from the simulation were very satisfactory. In most cases, where the source was located in easy-to-reach places (e.g. Fig. 1) the agents experienced no problems in reaching the source and, most of the time, in less than 20 steps of the algorithm. In more challenging scenarios where the source was in a hard-to-reach places, the agents experienced certain difficulties. Specifically in Figure 2, where the source is located near the back wall and also behind a vertical wall, some agents could not converge towards the source within the maximum step limit. However, in such challenging scenarios, since at least some of the agents gathered to the source (within a range of 0.5m), we consider the execution successful.

VI. Conclusion

In this paper we have presented an efficient method for locating and reaching a source in a realistic environment by an unsupervised wireless robotic network. This method can be applied to various scenarios containing hostile or unreachable environments to humans. Such scenarios include locating radioactive sources, enemy communication infrastructures on the battlefield and even locating valuable resources on an extraterrestrial environment.

Our work has focused on simulating a basic configuration of a wireless robot network searching in a typical environment. Future work includes scaling up the search space, fine tuning the searching algorithm, implementing smart routing protocols (allowing for communication between agents that are not within range of each other), embodying into the simulation tool more real life constrains (such as the proposed distributed global positioning system) and implementing the robotic swarm.

References

[1] A. W. Stroupe, M. C. Martin, and T. Balch, "Distributed sensor fusion for object position estimation by multi-robot systems," in *IEEE International Conference on Robotics and Automation*, May 2001.

[2] G. A. S. Pereira, V. Kumar, and M. F. M. Campos, "Localization and tracking in robot networks," in *11th International Conference on Advanced Robotics (ICAR 2003)*, July 2003.

[3] W. Jatmiko, Y. Ikemoto, T. Matsuno, T. Fukuda, and K. Sekiyama, "Distributed odor source localization in dynamic environment," in *IEEE Sensors*, November 2005.

[4] J. Wessnitzer and C. Melhuish, *Advances in Artificial Life*. Springer / Heidelberg, 2003, ch. Collective Decision-Making and Behaviour Transitions in Distributed Ad Hoc Wireless Networks of Mobile Robots: Target-Hunting, pp. 892–902.

[5] R. C. Eberhart, Y. Shi, and J. Kennedy, *Swarm Intelligence*. Morgan Kaufmann, 2001.

[6] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *IEEE International Conference on Neural Networks*, November 1995.

[7] M. Clerk and J. Kennedy, "The particle swarm - explosion, stability, and convergence in a multi dimensional complex space," *IEEE Transactions on Evolutionary Computation*, pp. 58–73, 2002.

[8] G. Dudek and M. Jenkin, *Computational Principles of Mobile Robotics*. Cambridge University Press, 2000.

[9] J. Borenstein and Y. Koren, "Obstacle avoidance with ultrasonic sensors," *IEEE Journal of Robotics and Automation*, vol. 4, no. 2, pp. 213–218, 1988.

[10] COST-231, *Digital Mobile Radio Towards Future Generation Systems (Final Report)*. European Commission, 1999.

[11] J. L. Keenan and A. J. Motley, "Radio coverage in buildings," *British Telecom Technology Journal*, pp. 19–24, 1990.

[12] Moteiv-Corp., "Tmote sky: Datasheet," http://moteiv.com/products/docs/tmote-sky-datasheet.pdf, 2006.