The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Cooperative Chemical Concentration Map Building Using Decentralized Asynchronous Particle Swarm Optimization Based Search by Mobile Robots

Mirbek Turduev, Yunus Ataş, Pedro Sousa, Veysel Gazi, and Lino Marques

Abstract— **In this article the main objective is to perform a search in an unknown area with multiple robots in order to determine the region with highest chemical gas concentration as well as to build the chemical gas concentration map. The searching and map building tasks are accomplished by using mobile robots equipped with smart transducers for gas sensing. Robots perform the search autonomously by using their own data and the information (position information and sensor readings) obtained from the other robots. Moreover, simultaneously the robots send their sensor readings of the chemical concentration and their position data to a remote computer (a base station), where the data is combined, interpolated, and filtered to form an real-time map of the chemical gas concentration in the environment. To achieve this task as a high-level path planning algorithm we use a decentralized and asynchronous version of the Particle Swarm Optimization (PSO) algorithm which also allows for time-varying neighborhood.**

I. Introduction

In recent years the number of studies on application of robotic odor-sensing technology has increased substantially. Mobile robots equipped with chemical sensors can be useful for a number of application areas including safety, security, and environmental inspection. Instead of humans, robots can be dispatched to areas with odor contamination for inspection, or providing continuous monitoring of the contaminated environment for specific characterization of the odor. In this paper, we address the problem of determining the gas distribution in indoor environments by a swarm of mobile robots equipped with on-board gas sensors.

There are a number of works on odor source localization and mobile olfaction search [1], [2]. Most of the works on chemical sensing with mobile robots assume an experimental setup and source localization by one robot with specific navigation algorithm for exploring the environment and spreading into the environment. Moreover, some studies [2] present methods where the odor classification and gas distribution are combined for source localization and gas distribution map building. Also multiple odor sources are used and the resulting gas distribution map is combined with laser range finder and sensor data. In another work gas distribution mapping problem is solved by using concentration grid maps by mobile robot. Where mapping technique is introduced using Gaussian weighted functions to model chemical concentration data in point of measurements [3]. However, in most of the works mentioned above gas distribution map building is performed by using a single robot system.

This paper focuses on a different approach which exploits decentralized asynchronous particle swarm optimization with multiple robots to solve chemical concentration map building problem of an unknown plumed (i.e., contaminated) environment. The PSO was first introduced by Eberhart and Kennedy and used for optimization of continuous nonlinear functions [4]. Later, Marques and his colleagues [10], [11] adapted PSO to the multi-robot odor searching problem, investigating both theoretically and experimentally the application of a PSO inspired search strategy for detecting odor sources across large spaces. They also compare the PSO with other gradient based search strategies.

Some other works in adapting the PSO algorithm to multi-agent search applications were done by Pugh and Martinoli [5] and Jatmiko et al. [6]. In [5] the authors develop two different approaches which are adapting PSO to multi-robot search and adapting multi-robot search to PSO. Similar work was performed in [6] where the effect of the wind was also considered. Hereford proposed a new version of PSO algorithm called distributed PSO (dPSO) in [7]. His work is focused on both achieving scalability of the algorithm to a large number of robots and decreasing communication burden between robots. In another work [8] he developed a version of the PSO through distributing processes among several mobile bots which is called physically-embedded PSO (pePSO). A distributed PSO algorithm is implemented also in [9] where a multi-robot search algorithm based on chemotaxis behavior in bacteria is used and the parameters of the proposed algorithm are updated using distributed PSO. Doctor and his colleagues discuss using multiple robot searches involving one and more than one targets [12]. They focus on optimizing the parameters of the PSO search algorithm.

This study is continuation of the work in [13], where a new decentralized and asynchronous version of PSO is considered. It is claimed in [13] that direct application of canonical PSO to the multi-robot search may lead to an unsatisfactory performance. First of all, note that the sensing

This work was supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) under grant no 106E122 and by the European Commission under the GUARDIANS Project (FP6 contract No. 045269).

M. Turduev, Y. Ataş, and V. Gazi are with the Dept. of Electrical and Electronics Engineering, TOBB University of Economics and Technology, Söğütözü Cad. No: 43, 06560 Ankara, TURKEY. (email: {mturduev,yatas,vgazi}@etu.edu.tr).

P. Sousa and L. Marques are with Dept. of Electrical and Computer Engineering, Institute of Systems and Robotics, University of Coimbra, 3030-290 Coimbra, Portugal. (email: {pvsousa,lino}@isr.uc.pt).

and communication capabilities of the robots are usually limited, which may results in time dependent interactions. Furthermore, the robots cannot jump to their next way points and it may take different robots different amount of time to traverse their path from their respective current way points to their next way points. In order to overcome these shortcomings the authors propose a version of the PSO algorithm where particles are allowed to operate asynchronously and exchange information using dynamic neighborhood topology in order to improve performance of the canonical PSO. The proposed method is tested using KheperaIII robots and using experimentally collected data in order to determine the area with highest concentration.

Note that, there are important differences between this work and works in [5], [6], [7], [8], [11], [12], [13]. First of all the robots here operate in a real gas environment and perform sensing using an onboard chemical sensing hardware, whereas the mentioned works either assume that the robots have sensing capability and use previously collected and smoothed data [11], [13], or operate in a different environment such as different lightening conditions using photo sensors [5], [7], [12]. Moreover, in addition to determining the areas of high gas concentration here the robots perform real-time gas concentration map building. The obtained chemical gas concentration map is visualized in real-time on a remote computer (base station).

II. Problem Definition

Consider an application in which a group of robots are required to perform a search in an unknown environment contaminated with a chemical substance. The chemical substance is possibly dangerous for human beings. Therefore, search by autonomous robots is more suitable to the problem. Moreover, search by multiple robots in parallel can possibly lead to a faster performance. Assume that the robots are equipped with the necessary hardware equipment to sense the chemical (or set of chemicals) which contaminate the environment. The objective is to build the map of chemical concentration as well as to determine the region with the highest concentration of the contaminant. Such information might be useful for experts who may determine whether the concentration level of the chemical is within tolerable levels or constitute a danger. Moreover, composition of different chemicals can be determined. The higher chemical concentration areas usually occur very close to the sources of chemicals and determining the region with high concentration might give information of the location of the leaks (i.e., chemical sources) contaminating the environment. Such a situation can arise in, for example, a building (such as large warehouse) under fire where burning of certain chemicals might be really dangerous since certain levels of some chemicals might lead to explosion. Moreover, the composition of the chemical might give an idea of what are the burning materials. All this information can be collected by the robots without endangering the lives of firefighters and critical decisions might be taken (i.e., such as whether to go in or not) based on the information obtained by the robots.

In order to be able to perform efficient search the robots must be able to pass their sensor readings as well as other needed information among each other. Moreover, they are required to pass the information to a remote computer (a base station) where the data should be collected and combined and visual information (i.e., a 3D map) must be provided in real-time to an operator.

III. Experimental Setup

A. Environment and Robots

In this section we provide a short description of the experimental setup which is used in implementations. The experiments are performed with KheperaIII mobile robots equipped with the "kheNose" sensing system using TGS2620 alcohol sensors manufactured by Figaro. This type of chemical sensors show decreasing resistance in the presence of reducing volatile chemicals in the surrounding air. They are often used in mobile robotic systems, because they are inexpensive, highly sensitive, and relatively unaffected by changing environmental conditions such as room temperature and humidity. They are interfaced to the KheperaIII robots through interface board called "kheNose" shown on top of the robot in Figure 2(b) and described in more detail in the next subsection. The setup consists of experimental arena with no obstacles (shown in Figure 1) and $3.40m \times 2.40m \times 1.35m$ in dimensions. Moreover, the experimental arena is covered by a $transparent\ vinyl\ cover$. It constitutes a small scale representation of a large building (such as a warehouse) filled with a chemical. Ethanol, which is a volatile and colorless liquid, is used as chemical gas. For that purpose we use an air bubbler system composed of two bottles half filled with ethanol, an air pump, and plastic tubes to inject the evaporated alcohol into the arena. The pressure of the air pump can be adjusted manually between $150mbar$ - $550mbar$. In order to provide gas circulation the front right corner of the covered arena is left open to let air flow inside it.

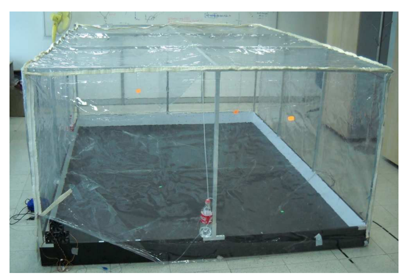

Fig. 1. Enclosed experimental setup.

For this implementation KheperaIII robots shown on Figure 2(a) are used. They are equipped with Intel PXA255 processor working at 400 MHz. Movement of the robots are provided by 2 brushless DC servo motors. Robots have 9 infrared and 5 ultrasound sensors located in the periphery. Each motor is controlled by its own PID controller implemented

(a) KheperaIII robots (b) KheperaIII with kheNose

Fig. 2. KheperaIII robots equipped with kheNose sensing system and 3DM GX2 Microstrain IMU.

in a PIC18F4431 microcontroller and this microcontroller is also used for measuring odometry information of the robot. The motor control blocks act as slave devices on an I^2C bus while communicating with a master DSPIC30F5011. The DSPIC30F5011 microcontroller is also used to obtain the sensor measurement. For robot odometry correction we used an IMU (Inertial Measurement Unit) which is manufactured by MicroStrain, shown on top of the robots in Figure 2(a). 3DM-GX2 IMU sensor is a high-performance gyro enhanced orientation sensor which utilizes miniature MEMS sensor technology. Orientation error is dominant in odometry localization system errors. Reduced error rate in orientation is inverse proportional to localization consistency. IMU is integrated to Khepera III robots to reduce error rate in robot orientation. Thresholding method is used to integrate IMU data to odometry localization. Basically, when the change in robot orientation is larger than a predefined treshold, IMU data is used for calculating the turn rate, otherwise odometry data is used in localization.

In order to navigate and avoid robot to robot collisions artificial potential functions [14] are used for low-level control of robots. With this objective in order to move the robots to their next way-points we use quadratic attractive potential function and require robots to move along their negative gradients. Also, in order to avoid collisions between the robots we use repulsive potential function which is activated when the distances between robots become smaller than a predefined constant value. The repulsive potential forces are calculated using infrared sensors. In addition, we augment the potential functions based collision avoidance with a priority based robot to robot collision avoidance. In other words, when two robots get in a close collision distance the robot with smaller ID waits until the robot with larger ID avoids the collision and continues on its path.

The robots used in this paper are unicycle agents moving in $\mathbb{R}^2$ with the dynamics

$$\begin{aligned} \dot{x}_i(t) &= \bar{v}_i(t)\cos(\theta_i(t)), \\ \dot{y}_i(t) &= \bar{v}_i(t)\sin(\theta_i(t)), \\ \dot{\theta}_i(t) &= w_i(t) \end{aligned} \tag{1}$$

where $P_i(t) = [x_i(t), y_i(t)]$ is the position vector and $\theta_i(t)$ is the steering angle of agent i at time t. It's control inputs are the linear speed $\bar{v}_i(t)$ and the angular speed $w_i(t)$.

The implementation operates as follows: robots start the search from entrance of the experimental area. Their first way-points are assigned intentionally far away from each other whereas the second way points are assigned randomly. The objective of this step is to spread the robots and cover the area as much as possible at start for better mapping of the gas concentration and better performance of the asynchronous PSO algorithm. Robots communicate with each other using the TCP/IP protocol over a wireless ad-hoc network to share their acquired information of gas concentration and position. They also send the acquired information to a nearby computer (base station). After the two initial steps the robots start asynchronous PSO as higher level path planning and determine their next way-points using the algorithm by exploiting their measurements/knowledge and the measurements/knowledge of other robots in the system. The robots move from their current way-points to the next way-points while avoiding obstacles by means of potential functions. This algorithm is expected to achieve convergence of the robot swarm into the region of highest chemical concentration. Moreover, a remote PC is used as "Base Station" for gathering data from robots and obtaining a real-time $3D$ map of the chemical concentration. Matlab and a trial version of the *Golden Surfer9* software are used as the main processing and visualization tools in the base station.

B. Chemical Sensors and kheNose

KheNose is a modular olfactory sensing system for Khepera III mobile robots. This device is composed by a main board, with robot interfacing and processing capabilities, and an array of gas sensing nostrils, a temperature and humidity sensor, and up to three small thermal anemometer boards.[1] Each sensing board contains a Transducer Electronic Data Sheet (TEDS) stored in an EEPROM memory, providing plug-and-play capabilities to the system [18]. The TEDS contains relevant information about the transducers, namely their type, target gases, range and calibration data. The main board contains a Microchip dsPIC33F controller which acquires all the analog and digital information from the sensors, processes that data and sends it to the Khepera III KoreBot extension board through an I^2C interface. The whole system architecture is inspired by IEEE1451.4 standard for smart sensors. The system can operate as an odor compass, being able to measure airflow intensity and direction while it classifies the detected odors [19]. The odor classification is achieved by means of a feedforward neural network. This classification can be done using the steady-state response of the gas sensor array or, for faster classification, using the coefficients of the discrete wavelet transform of the sensor array transient response [20].

1) Calibration of multiple olfactory systems: The conductance G of tin oxide gas sensors varies with the concentration C of a target reducing gas accordingly with following relationship [21]

$$G = G_1 {P_R}^n \tag{2}$$

[1] In the current work, only the information provided by a Figaro TGS2620 metal oxide gas sensor was employed.

where G_1 is the conductance for a small concentration C_1 of the reducing gas, $P_R = C/C_1$ is the relative concentration of the gas, and n is a constant dependent from the gas and from the sensor.

In multiple robot olfactory experiments it is fundamental to have the sensing systems calibrated against the same standard values, so the measurements can be merged in a single concentration map. In the current experiments, the response from a single metal oxide gas sensor per "kheNose" was employed, so the following fast calibration method was implemented:

1) All the employed systems were placed in an enclosed environment where fixed amounts of a target gas (ethanol vapor) could be inserted.
2) The conductance in clean air was registered as G_{air}.
3) A small volume of ethanol vapor was inserted into the calibration space and, after stabilization of the sensors, the conductance corresponding to the existing atmosphere concentration C_1 was registered as G_1.
4) The same volume of ethanol was inserted into the calibration space and the conductance G_2 corresponding to the concentration $C_2 = 2C_1$ was registered.

Fixed amounts of ethanol vapor can be inserted into the calibration space using a large syringe or a Mass Flow Controller, as shown in Figure 3. In that figure the robots are inside an acrylic glass calibration box, containing a fan mixer to homogenize the atmosphere.

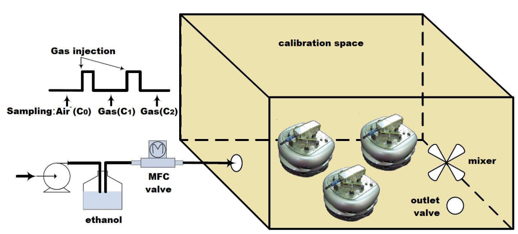

Fig. 3. Calibration setup.

After the previous calibration procedure, the constant n of each sensor could be determined. In operation, for concentrations above C_1, equation (2) was employed and for concentrations below that value, a linear interpolation between the output to clean air and the output to C_1 was employed.

IV. Asynchronous PSO

In this section we briefly describe the PSO algorithm which is used as the main high-level path planning algorithm. In particular, first we describe the standard PSO algorithm following which we discuss also its modified version used in this article. PSO is a population based direct search algorithm which is in general suitable for multi-robot search applications. However, there are also important differences which are discussed in [5]. Two of the main differences can be stated as: (i) the robots cannot jump to the next position, whereas the particles can do; (ii) the communication range of the robots can be limited, whereas there is no such constraint for the particles. (See [5] for more details.) These differences can lead to degradation of performance or even failure of the standard PSO when it is applied directly to multi-robot search without any modification. We discuss the differences in detail in the subsection on decentralized asynchronous PSO. However, for completeness let us first discuss the standard PSO algorithm.

A. Canonical PSO based search algorithm

In this article we use the PSO version proposed by Clerc and Kennedy in [17]. It uses a "constriction coefficient" to prevent the "explosion behavior" of the particles (see [17] for more details). At the k^{th} iteration the update of particle i, $i = 1, \ldots, N$, can be described as

$$\begin{aligned} v_i(t^i_{k+1}) &= \chi \Big[v_i(t^i_k) + \varphi^i_1(t^i_k)\Big(b_i(t^i_k) - p_i(t^i_k)\Big) \\ &\qquad + \varphi^i_2(t^i_k)\Big(g_i(t^i_k) - p_i(t^i_k)\Big)\Big] \\ p_i(t^i_{k+1}) &= p_i(t^i_k) + v_i(t^i_{k+1}) \end{aligned} \tag{3}$$

where t^i_k is the update time. Note that in this article each robot is considered as a particle from PSO view-point. The robot dynamics operates in continues time t and the instances at which robot/particle i performs its k^{th} iteration is denoted with t^i_k. Here $p_i(t^i_k) \in \mathbb{R}^2$ represents the position (way point) of the i'th particle at time t^i_k, $b_i(t^i_k) \in \mathbb{R}^2$ represents the best position of the i'th particle from time $t = 0$ to time $t = t^i_k$, $g_i(t^i_k) \in \mathbb{R}^2$ represents the best position of the neighborhood of the i'th particle from time $t = 0$ to time $t = t^i_k$. The value $p_i(t^i_{k+1}) \in \mathbb{R}^2$ which is calculated in (3) is the next (desired) way point to which the robot should move. The learning coefficients $\varphi^i_1(t^i_k) \in [0, \bar{\varphi}_1]^2$ and $\varphi^i_2(t^i_k) \in [0, \bar{\varphi}_2]^2$ are two dimensional uniform random vectors. At each iteration these random vectors respectively determine the relative significance/weight of the cognitive and social components in the iteration. The constant parameter $\chi > 0$ is the constriction parameter which prevents the explosion behavior, i.e., particles having high velocities leading to their scattering in the search space. For efficient performance and prevention of the explosion behavior in (3) we use the components of the $\varphi^i_1(t^i_k)$ and $\varphi^i_2(t^i_k)$ learning coefficient vectors proposed by [17] as

$$\varphi^i_{1j}(t^i_k), \varphi^i_{2j}(t^i_k) \in [0, 2.05], j = 1, 2; i = 1, \ldots, N. \tag{4}$$

The constriction parameter $\chi > 0$ is calculated using (refer to [17])

$$\chi = \begin{cases} \frac{2\kappa}{\varphi - 2 + \sqrt{\varphi^2 - 4\varphi}}, & \text{if } \varphi > 4, \\ \kappa, & \text{else.} \end{cases} \tag{5}$$

Here $\varphi = \bar{\varphi}_1 + \bar{\varphi}_2$ and $\kappa \in [0, 1]$. Similar to the work in [13], considering $\bar{\varphi}_1 = \bar{\varphi}_2 = 2.05$ and $\kappa = 1$, the constriction parameter is calculated as 0.7298 for this implementation.

B. Asynchronous PSO based search algorithm

As mentioned before there are important differences between robotic search and PSO search. The robots have to traverse the entire path between the current way-point and the next way-point calculated by the PSO algorithm. Since the distance between the current positions and the next way-points can be different for different robots in the standard PSO a robot which arrives at its way-point earlier than the other robots has to wait for them in order to perform its PSO iteration. Moreover, since the communication range of the robots is usually limited and the area to be searched can be large during the search process the robots may exit the communication range of their neighbors, which may result in indefinite wait and stall by the system. Furthermore, permanent communication and robot failures can lead to failure of the overall search. Realizing these potential problems a decentralized asynchronous PSO inspired multi-robot search algorithm which allows also for dynamic neighborhood and possible time delays is proposed in [13] (which is also inspired by the earlier works in [15], [16]). The algorithm can be briefly described by the pseudocode given in Table I which is taken from [13]. In Table I $S(b_i)$ refers to the sensor reading at the best position of the robot and $S(g_i)$ is the sensor reading at the global best position. Note that this version of PSO does not suffer from above mentioned problems since the robots do not have to synchronize and can continue operation using the available inforamtion only.(See [13] for more details.) Also, mentioned above before initiating the algorithm in Table I the robots move to points away from each other for better coverage of the area. During this step they also continuously sense the environment and send the information to the base station.

TABLE I

PSEUDOCODE OF THE ALGORITHM

Initialize the variables
Assign the first way points randomly
while (Stopping criteria is not satisfied) **do**
  while (Agent has not arrived to its way point) **do**
    Move towards the desired way point
    Read concentration data from the environment
    Update $S(b_i)$
    Listen to data from other robots
    Send data to base station
    if (Collision distance to other robots) **then**
      Apply priority based collision avoidance
    end if
  end while
  Broadcast own $S(b_i)$
  if ($S(b_{i\_other}) > S(g_i)$ or $S(b_i) > S(g_i)$) **then**
    Update $S(g_i)$
  else
    Use previous $S(g_i)$
  end if
  Calculate a new way point using Equation (3)
end while

V. EXPERIMENTAL RESULTS

In this section we present the results obtained in the implementation considered. We use three Khepera III robots in the experiments. Initially the robots are located near to the entrance of the experimental area. Their first way-points are assigned intentionally far away from each other in order to enclose the environment as much as possible for better map building of the gas distribution. After reaching the first way-point the next way-points are assigned randomly(as seen in Table I). In order to share their acquired information robots communicate with each other using the TCP/IP protocol over wireless ad-hock network. Moreover, they also send their position and "kheNose" sensor data gathered from the environment to a remote computer (base station) during the search. The data received by the base station is combined, interpolated, and filtered to form a real-time map of the gas concentration. We used krigging estimation and a gaussian filter with a $[5, 5]$ mask.

The map is visualized in real-time as a three dimensional plot to be viewed by an operator. We performed experiments with several different gas source locations in the environment two of which are shown here. For the first case presented in Figures 4 and 5 three sources are placed in the environment at the locations $(1.5, 1.6)$, $(1.65, 1.6)$, and $(1.6, 1.4)$. Figure 4 shows the trajectories of the robots superimposed on the contour plot of the extracted environmental map for an example run and bold stars represent the source locations in the environment. The paths of different robots are represented with different types of curves. The $3D$ plot of the obtained ethanol concentration in the environment is shown in Figure 5. As can be seen from the figures the robots gather in an area with high gas concentration (which, in general,

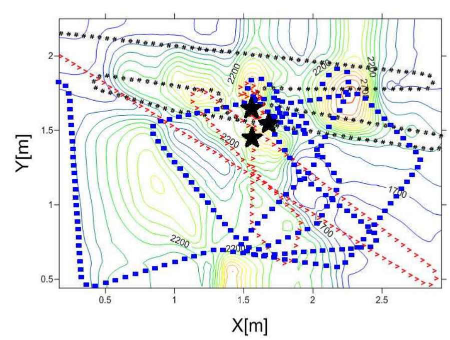

Fig. 4. Trajectories of robots.

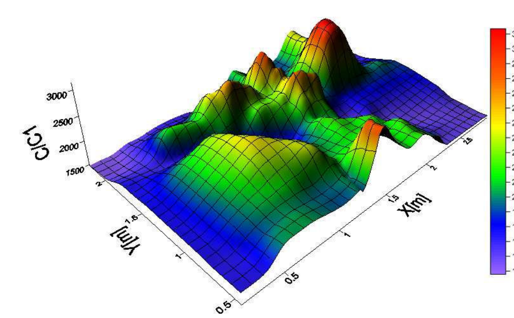

Fig. 5. 3D representation of chemical concentration.

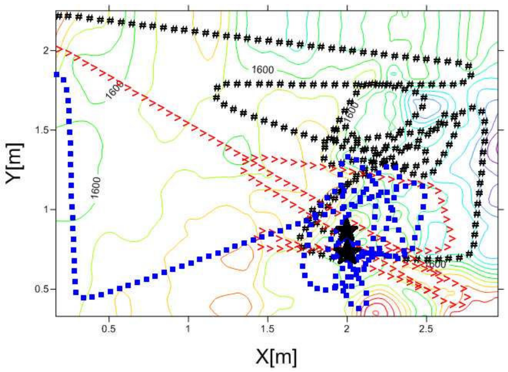

Fig. 6. Trajectories of robots.

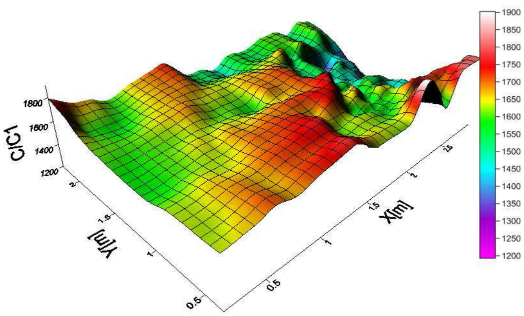

Fig. 7. 3D representation of chemical concentration.

may not necessarily be the global maximum). The plots presented in Figures 6 and 7 show the results for another run in which there are two gas sources in the environment located at $(2, 0.9)$ and $(2, 0.7)$. As can be seen from the figures the robots aggregate at the area of high concentration of the ethanol gas in the environment. In both of the experiment a map of the ethanol concentration is obtained. It is a realistic map since the peaks of the map occur very close to the gas sources. Only three robots where used in the experiments due to lack of resources. It might be possible to obtain better results with higher number of robots.

VI. Concluding Remarks

In this study cooperative chemical concentration map building using Decentralized Asynchronous Particle Swarm Optimization inspired search is performed by mobile robots. The implementation is realized in a laboratory setting with Khepera III robots equipped with "kheNose" chemical sensing system with Figaro alcohol sensors and real ethanol gas. The results show that the robots succeed in extracting a three dimensional map of the concentration and aggregate in regions which they perceive as areas of high concentration. Possible applications might include search in large buildings such as warehouse under fire. Future research might focus on comparing the performance of the algorithm with the performance of alternative methods. Affords in this direction continue.

References

[1] A. T. Hayes, A. Martionoli and R. M. Goodman, "Distributed Odor Source Localization," in *IEEE Sensors Journal*, vol. 2, no. 3, June 2002.

[2] A. Loutfi, S. Goradeschim, A. J. Lilienthal and J. Gonzales, "Gas Distribution Mapping of Multiple Odour Sources using a Mobile Robot," in *Robotica (2009)*, vol. 27, pp. 311–319 Cambridge University Press, March 2009.

[3] A. Lilienthal, T. Ducket, "Building gas concentration gridmaps with mobile robot," in *Robotic and Autonomous Systems*, vol. 48, no. 1, pp. 3-16, May 2004.

[4] J. Kennedy and R. C. Eberhart, "Particle Swarm Optimization," in *IEEE International Conference on Neural Network*, pp. 1942–1948, 1995.

[5] J. Pugh and A. Martinoli, "Inspiring and Modeling Multi-Robot Seach with Particle Swarm Optimization," in *Proceedings of the 4th IEEE Swarm Intelligence Symposium*, pp. 332–339, Honolulu, HI, USA, April 2007.

[6] W. Jatmiko, K. Sekiyama and T. Fukuda "A PSO-based Mobile Sensor Network for Odor Source Localization in Dynamic Environment: Theory, Simulation and Measurement," in *IEEE Congress on Evolutionary Computation*, pp. 1036–1043, Vancouver, BC, July 2006.

[7] J. M. Hereford, "A distributed Particle Swarm Optimization algorithm for swarm robotics application," in *Congress on Evolutionary Computation*, pp. 6143–6149, Vancouver, BC, July 2006.

[8] J. M. Hereford and M. Siebold, "Multi-robot search using a physically-embedded Particle Swarm Optimization," in *International Journal of Computational Intelligence Research*, vol. 4, pp. 197–209, 2008.

[9] J. Pugh and A. Martinoli, "Distributed Adaptation in Multi-Robot Search using Particle Swarm Optimization," in *Proceedings of the 10th International Conference on the Simulation of Adaptive Behavior*, Lecture Notes in Computer Science, pp. 393–402, 2008.

[10] L. Marques, U. Nunes and A. T. de Almedia " Finding Odours Across Large Search Spaces: A Particle Swarm-Based Approach," in *Climbing and Walking Robots: Proc. of the 7th Int. Conf. CLAWAR 2004*, pp. 419–426, Springer, 2005.

[11] L. Marques, U. Nunes and A. T. de Almedia "Particle swarm-based olfactory guided search," in *Autonomous Robots*, vol. 20, no. 3, pp. 277–287, June 2006.

[12] S. Doctor, G. K. Venayagamoorthy and V. G. Gudise, "Optimal PSO for Collective Robotics Search Applications," in *Proceeding of the IEEE Congress on Evolutionary Computation*, pp. 1390–1395, Portland, OR, USA, June 2004.

[13] S. B. Akat, V. Gazi and L. Marques "Asynchronous Particle Swarm Optimization Based Search with a Multi-Robot System: Simulation and Implementation on Real Robotic System," in *Turkish Journal of Electrical Engineering and Computer Sciences (accepted for publication)*, Turkey, 2010.

[14] V. Gazi, "Swarm Aggregations Using Artificial Potentials and Sliding Mode Control," in *IEEE Transactions on Robotics*, vol. 21, no. 6, pp. 1208–1214, December 2005.

[15] S. B. Akat and V. Gazi, "Particle Swarm Optimization with Dynamic Neighbourhood Topology: Three Neighborhood Strategies and Preliminary," in *IEEE Swarm Intelligence Symposium (SIS-2008)*, St. Louis, Missouri, September 2008.

[16] S. B. Akat and V. Gazi, "Decentralized Asynchronous Particle Swarm Optimization," in *IEEE Swarm Intelligence Symposium (SIS-2008)*, St.Louis, Missouri, September 2008.

[17] M. Clerc and J. Kennedy, "The Particle Swarm - Explosion, Stability, and Convergence in a Multidimensional Complex Space," in *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, February 2002.

[18] J. Pascoal, P. Sousa and L. Marques, "kheNose - A smart transducer for gas sensing," in *Proc. of the 11th Int. Conf. on Climbing and Walking Robots and the Support Technologies for Mobile Machines, (CLAWAR2008)*, Coimbra, Portugal, 2008.

[19] L. Marques, N. Almeida and A. T. de Almeida, "Olfactory Sensory System for Odour-Plume Tracking and Localization," in *IEEE Int. Conf. on Sensors*, Toronto, Canada, 2003.

[20] N. Almeida and L. Marques and A. T. de Almeida, "Fast Identification of Gas Mixtures Through the Processing of Transient Responses of an Electronic Nose," in *Proc. of EuroSensors*, Guimaraes, Portugal, 2003.

[21] N. Barsan and A. Tomescu, "Calibration procedure for S_nO_2-based gas sensors," *Thin Solid Films*, vol. 259, no. 1, pp. 91–95, 1995.