Modified PSO Algorithm Based on Flow of Wind for Odor Source Localization Problems in Dynamic Environments

Wisnu Jatmiko[1], Petrus Mursanto[1], Benyamin Kusumoputro[1], Kosuke Sekiyama[2], and Toshio Fukuda[2]
[1]Faculty of Computer Science, University of Indonesia, Indonesia.
(phone: (62)-021-786-3419; fax: (62)-021-786-3415; e-mail: wisnuj@cs.ui.ac.id and r_efendi@yahoo.com)
[2]Department of Micro-Nano Systems Engineering, Nagoya University, Japan.
(e-mail: {sekiyama,fukuda}@mein.nagoya-u.ac.jp).

Abstract- A new algorithm based on Modified Particle Swarm Optimization (MPSO) in order to control autonomous vehicles for solving odor source localization in dynamic advection-diffusion environment have been developed. Furthermore an improvements of the MPSO for odor source localization, which follows a local gradient of the chemical concentration within a plume is investigated. Another popular biomimetic approach in odor source localization problem is anemotaxis. An anemotaxis-driven agent measures the direction of the fluid's velocity and navigates "upstream" within the plume. In this paper, the combination of chemotaxis "MPSO"-based algorithm and anemotaxis will be described. This method is well known in the animal kingdom as odor-gated rheotaxis (OGR). On the other hand, in real world, the odor distribution is multi peaks especially in obstacle environments. For that reason, a new environment with obstacle will be developed. The purpose of developing the environment is to bridge the gap between very complex, hard-to-understand real world problems (odor dispersion model) and overly simplistic-toy-problem (dynamic bit matching or moving parabola). Simulations illustrate that the new approach can solve Advection-Diffusion odor model problems in such a dynamic odor with obstacle-filled environments.

Key-Words: - MPSO, Anemotaxis, Chemotoxis, OGR, Advection-Diffsuion, Odor Source

I. INTRODUCTION

Most work on chemical sensing with mobile robots assume an experimental setup that minimizes the influence of turbulent transport by either minimizing the source-to-sensor distance in trail following [1][2] or by assuming a strong unidirectional air stream in the environment [3-6], including our previous work [7]. However, not much attention has been paid to the natural environment problem.

To the best of our knowledge, there is no real implementation on a mobile robot that works in the natural environment. The main problem in implementing odor source localization using gas sensor in natural environments is that the distribution of the odorant molecules is usually dominated by turbulence rather than diffusion, the latter of which is known to be a considerably slower transport mechanism for gases in general. The other problem is the influence of unstable wind. When odor distribution is very complex and the wind direction is not stable, the robot will be haphazard and desultory [1-7].

This paper focuses on our new approach that exploits particle swarm optimization with multiple robots to solve odor source localization in natural environments where the odor distribution may change over time [8-10]. Furthermore some improvement is investigated. The improvement is conducted not only to combine chemotaxis "MPSO"-based with anemotaxis becoming odor-gated rheotaxis (OGR) but also to develop the Advection-Diffusion odor model with obstacle.

II. PARTICLE SWARM OPTIMIZATION FRAMEWORK

Many complex real-word optimization problems are dynamic, and change stochastically over time. These problems require measurements that account for the uncertainty present in the real world. Evolutionary algorithms (EAs), especially Particle Swarm Optimization (PSO), have proven successful in a number of static applications as well as dynamic and stochastic optimization problems. They are particularly successful because they draw their inspiration from the principles of natural evolution, which is a stochastic and dynamic process.

The interaction of the robot with the PSO algorithm is described as follows: Suppose that a population of robots is initialized with certain positions and velocities; let $\mathbf{x}_i(t)$ and $\mathbf{V}_i(t)$ denote the position and the velocity vector of the *i*-th robot at the iteration time *t* (*t*=1,2...). In addition, let $\boldsymbol{p}_i$ and $\boldsymbol{p}_g$ be defined as the best local and the best global position found in plume distribution that is under evaluation by the robot at position $\mathbf{x}_i(t)$. The position and the velocity are updated to improve the fitness function at each time step. When a robot discovers a pattern that is better than any previously found, the positional coordinates are stored in the vector $\boldsymbol{p}_i$, the best position found by robot *i* so far. The difference between $\boldsymbol{p}_i$ and the current position $\mathbf{x}_i(t)$ is stochastically appended to the current velocity $\mathbf{V}_i(t)$. This causes a change to the trajectory the robot would take at that position. The stochastically weighted difference between the population's best position $\boldsymbol{p}_g$ and the individual's current position $\boldsymbol{x}_i$ is also added to the velocity, in order to adjust for the next time step. These adjustments to the robot behavior direct the search around two best positions.

The value of $\boldsymbol{p_g}$ (the best global position for concentration of the gas) is determined by comparing the best performances of all the members of the population. The performances are defined by indices from each population member; and the best performer's index is assigned as the variable *g*. Thus, $\boldsymbol{p_g}$ represents the best position found by all members of the population.

Each robot is equipped with an ad-hoc wireless network and global positioning system (GPS). Through the ad-hoc network, each robot transmits and collects the information about the gas concentration, while the position of the robot is determined by the GPS.

A. Standard Particle Swarm

The concept of standard PSO is described in eq. (1) and (2).

$$\begin{aligned}\mathbf{V}_i(t) = \chi\big(&\mathbf{V}_i(t-1) + c_1 rand()(\mathbf{p}_i(t-1) - \mathbf{x}_i(t-1)) \\ &+ c_2 Rand()(\mathbf{p}_g(t-1) - \mathbf{x}_i(t-1))\big)\end{aligned} \tag{1}$$

$$\mathbf{x}_i(t) = \mathbf{x}_i(t-1) + \mathbf{V}_i(t) \tag{2}$$

After finding the two best values, the particle velocity and position is updated with eq.(1) and (2). The functions *Rand*() and *rand*() are random functions returning a value between (0,1). Coefficient χ is constriction factor, which is less than 1. The coefficient c_1 and c_2 are learning parameters, where $c_1 = c_2 = 2$.

The main problem with standard PSO applications in dynamic optimization problems is that the PSO will eventually converge to an optimum; it thereby looses the diversity necessary for efficient exploration of the search space.

B. Charged PSO

Applying Coulomb's law, a charged swarm robot is introduced in order to maintain diversity of the positional distribution of the robots and to prevent them from being trapped in a local maximum. This enhances adaptability to the changes of the environment. Figure 4 shows the repulsion function for charged swarm robots. Suppose that robot *i* can observe the present position of the other robots ($\mathbf{x}_p \neq \mathbf{x}_i$) and has a constant charge Q_i in order to keep a mutual distance away and maintain positional diversity. Two types of swarm robots are defined: *neutral* and *charged* robots. For all *neutral* robots $Q_i = 0$; hence, no repulsive force is applied to the *neutral* robots. For *charged* robots, the mutual repulsive force between robots *i* and *p* is defined according to the relative distance, $|\mathbf{x}_i - \mathbf{x}_p|$ as follows;

$$\mathbf{a}_{ip} = \begin{cases} \dfrac{Q_i \cdot Q_p (\mathbf{x}_i - \mathbf{x}_p)}{r_{core}{}^2 |\mathbf{x}_i - \mathbf{x}_p|} & |\mathbf{x}_i - \mathbf{x}_p| < r_{core} \\ \dfrac{Q_i \cdot Q_p}{|\mathbf{x}_i - \mathbf{x}_p|^3}(\mathbf{x}_i - \mathbf{x}_p) & r_{core} < |\mathbf{x}_i - \mathbf{x}_p| < r_{perc} \\ 0 & r_{perc} < |\mathbf{x}_i - \mathbf{x}_p| \end{cases} \tag{3}$$

where, $(i \neq p)$, r_{core} denotes the diameter inside which a constant, strong repulsion force is applied and r_{perc} denotes the recognition range of robot. Hence, if the mutual distance is beyond r_{perc}, there exists no repulsion force between the robots. In the case of $r_{core} \leq r \leq r_{perc}$, the repulsion force is dependent on the mutual

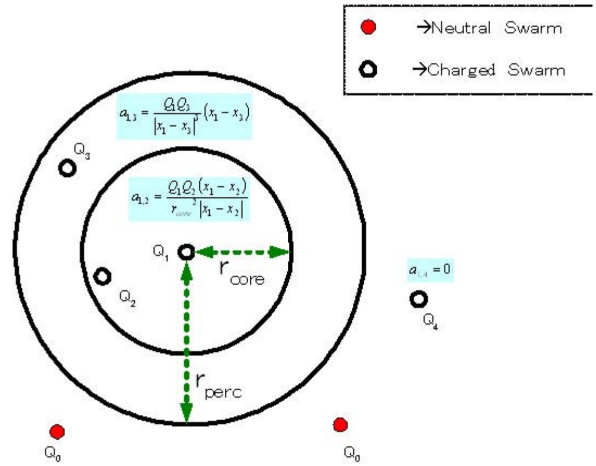


Fig. 1. Interaction of the charged swarm robots.

distance. Then, taking the summation of the mutual repulsion force, robot *i* defines collective repulsion force by:

$$\mathbf{a}_i(t) = \sum_{p \neq i}^{N} \mathbf{a}_{ip} \tag{4}$$

where *N* is number of the robots. The charged swarm robot is described in equations (5) and (6)

$$\begin{aligned}\mathbf{V}_i(t) = \chi\big(&\mathbf{V}_i(t-1) + c_1 rand()(\mathbf{p}_i(t-1) - \mathbf{x}_i(t-1)) \\ &+ c_2 Rand()(\mathbf{p}_g(t-1) - \mathbf{x}_i(t-1))\big) + \mathbf{a}_i(t)\end{aligned} \tag{5}$$

$$\mathbf{x}_i(t) = \mathbf{x}_i(t-1) + \mathbf{V}_i(t) \tag{6}$$

where, the first part of eq.(5) is responsible for finding and convergence to the optimal solution, while the second part maintains diversity of the swarm distribution and prevents robots from being trapped in a local maximum. Also, if all robots are set to the *neutral*, Charged PSO (CPSO) is reduced to the standard PSO, as described in

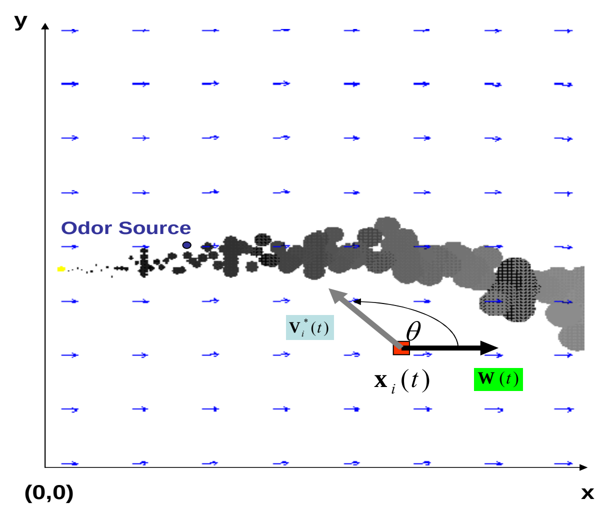
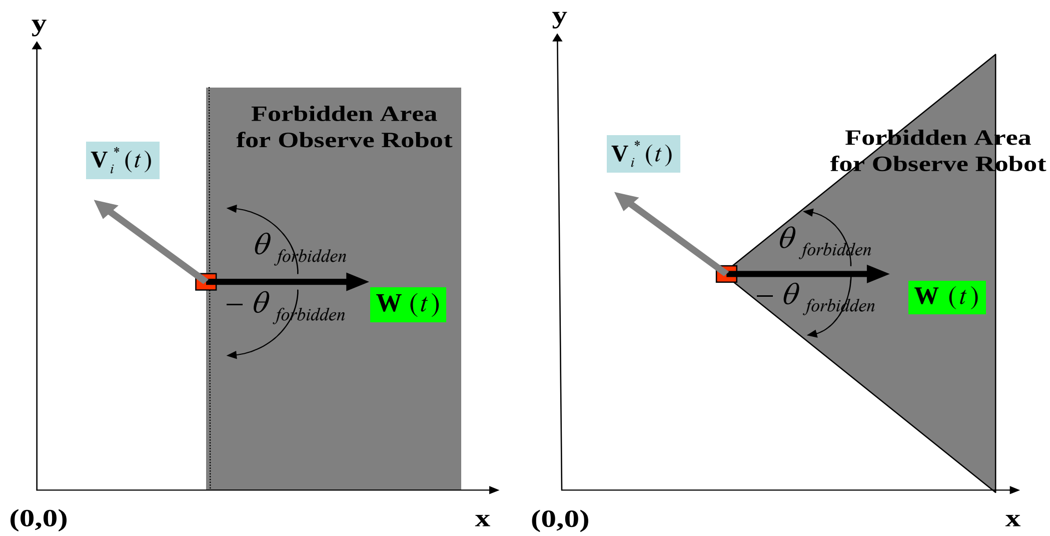


Fig. 2. Modified particle swarm optimization with wind utilization concept.



(a). $|\theta_{forbidden}| \le 90°$ (b). $|\theta_{forbidden}| \le 45°$

Fig. 3. Conceptual idea wind utilization with forbidden area (note x-axis is taken as the downwind direction)

eq. (1) and (2). The conceptual idea of Charged PSO is shown in Fig 1.

III. Extension with Wind Utilization

In this section, the integration of chemotaxis and anemotaxis properties to the PSO is introduced. Again, chemotaxis causes the Modified PSO robots to follow a local gradient of the chemical concentration, while an anemotaxis-driven PSO measures the direction of the fluid's velocity and navigates "upstream" in the plume to find the odor source. This methodology is well known as odor-gated rheotaxis (OGR) since it is employed by animals to find food.

The logic of OGR is clear. If an agent senses a plume, the mean flow of that plume must be bearing chemicals from the source of the plume toward the agent; and, therefore, a movement against the mean flow will reduce the agent's distance from the source. If the agent looses contact with a previously detected plume during its navigation upstream, the agent may overshoot the source. If he has lost contact with the plume, the agent moves back and forth across the path on which he knew the flow was located to re-contact the plume. This back and forth movement is termed casting and is a typical behavior of individualistic animals such as the American Lobster. In Particle Swarm Optimization, the algorithm not only shares individualistic information but also shares social information. This following section details the adaptation and implementation of OGR into the MPSO.

A. Conceptual Idea

As explained in Eq. (1) and (2) earlier, unless the position and velocity are updated in the PSO algorithm, there is no guarantee the robot direction will follow the plume upstream to the source. To combat this issue we utilized wind information.

Assume the velocity from the basic PSO becomes an intermediate velocity ($\mathbf{V}_i^*(t)$) from which the robots can know the direction of the wind ($\mathbf{W}$ (t)) at every step in time. Gas is emitted from the source of a coordinate system (x, y) as in Fig. 2, where the x-axis is taken as the downwind direction. The movement of the robot can be controlled by analyzing the angle (θ) between the intermediate velocity vector of the robot and the wind direction vector. Note that the angle is a relative direction, its mean depends on the direction of the wind at this time step. (In Figure 15, the angle between x-axis and wind direction is zero). With this concept, the robot movement not only will follow the gradient of the chemical concentration but also will follow the direction "upstream" of the wind. As a more detailed explanation, let us reformulate $\mathbf{V}_i^*(t)$ and $\mathbf{W}$ (t) as vectors defined as follows:

$$\mathbf{V}_i^*(t) = v_x \mathbf{e}_x + v_y \mathbf{e}_y \tag{7}$$

$$\mathbf{W}(t) = w_x \mathbf{e}_x + w_y \mathbf{e}_y \tag{8}$$

The angle of the two vectors $\mathbf{V}_i^*(t)$ and $\mathbf{W}$ (t) in two-dimensional space becomes an inner product and is defined as:

$$\theta = \cos^{-1}\left(\frac{\mathbf{V}_i^*(t).\mathbf{W}(t)}{\|\mathbf{V}_i *(t)\|\|\mathbf{W}(t)\|}\right) \tag{9}$$

From Fig. 2 and Eq. 7 – 9, we have many variables to control the velocity $\mathbf{V}_i(t)$ of the robot. We will explain two implementations of using the wind direction in the MPSO.

B. Implementation I: Used Forbidden Area

In wind utilization implementation I, we let the angle θ in Eq. 9 describe a forbidden area. The forbidden area represents an area where the robots have high likelihood of going the wrong direction (i.e. the robot direction will not follow the upstream to the source

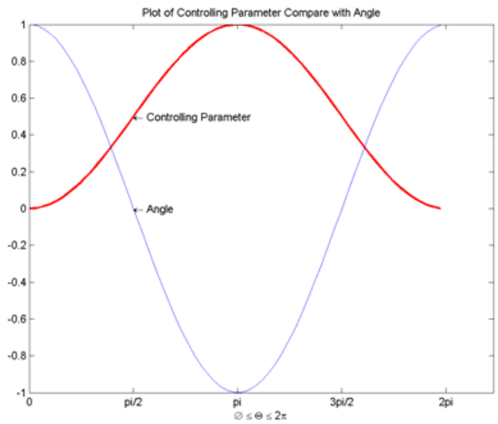


Fig. 4. Continues the function for controlling the velocity of the robots

within this area). If the angle (θ) is inside the forbidden area, there must be some action taken to avoid this area. In this simulation, for simplicity reasons, the action taken is to terminate the robot (i.e. let $\mathbf{V}_i(t)$ = 0). Otherwise, the intermediate velocity of robot ($\mathbf{V}_i^*(t)$) will become the velocity of the robots ($\mathbf{V}_i(t)$). The conceptual idea of this implementation, with a different forbidden area is shown in Fig. 3.

The modified PSO with Wind Utilization I (WUI) concept is described from Eq. 10 -11. (Other parameters still follow the basic PSO concept parameters.)

$$\mathbf{V}_i(t) = \begin{cases} 0 & \text{if } \theta < \left|\theta_{\text{forbidden}}\right| \\ \mathbf{V}_i^*(t) & \text{Otherwise} \end{cases} \tag{10}$$

$$\mathbf{x}_i(t) = \mathbf{x}_i(t-1) + \mathbf{V}_i(t) \tag{11}$$

In Fig. 3, the angle between the x-axis and the downwind direction is zero. If there is any angle between the x-axis and the downwind direction, the algorithm adapts automatically by comparing the relative angle between vectors $\mathbf{V}_i^*(t)$ and $\mathbf{W}$ (t).

C. Implementation II: Using the(χ_θ) Parameter

The weakness of implementation of Wind Utilization I is that it needs tuning of the forbidden area parameter. For implementation, we use the controlling parameter χ_θ to decide the velocity of the robot. After getting the intermediate velocity of the robot , $\mathbf{V}_i^*(t)$, the Wind Utilization II (WUII) algorithm will calculate the angle (θ) as mentioned in Eq. 9. Then the controlling parameter, χ_θ , is calculated. The continuation function for the controlling parameter χ_θ is described as follows:

$$\chi_\theta(\mathbf{W}(t), \mathbf{V}_i^*(t)) = \frac{1}{2}\left(1 - (\mathbf{W}(t), \mathbf{V}_i^*(t))\right) \tag{12}$$

where the relation of the angle θ and the controlling parameter χ_θ are shown in Fig. 4.

The modified PSO with Wind Utilization II (WUII) concept is described from eq. (13) to eq. (14):

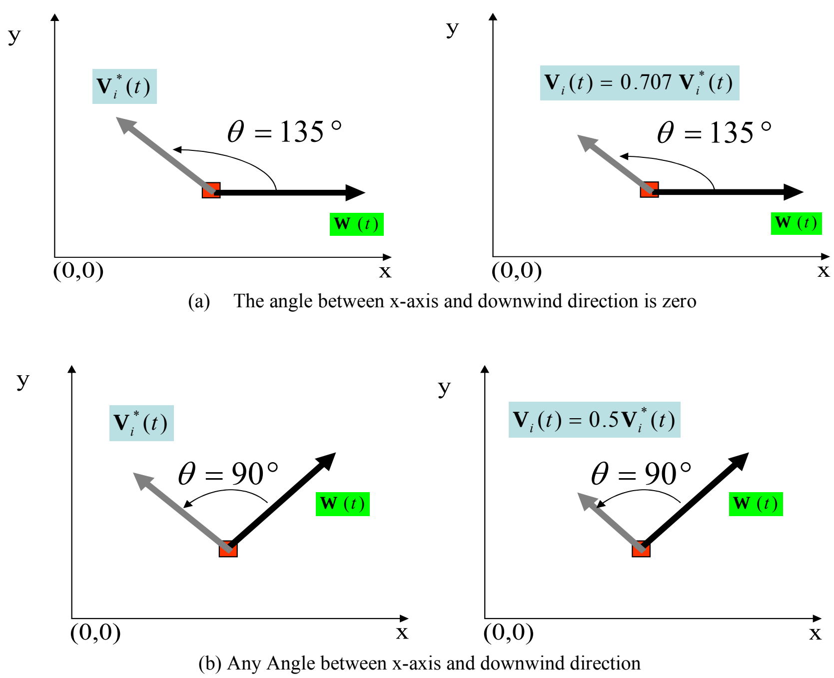


Fig. 5. Conceptual idea wind utilization with (χ_θ) parameter.

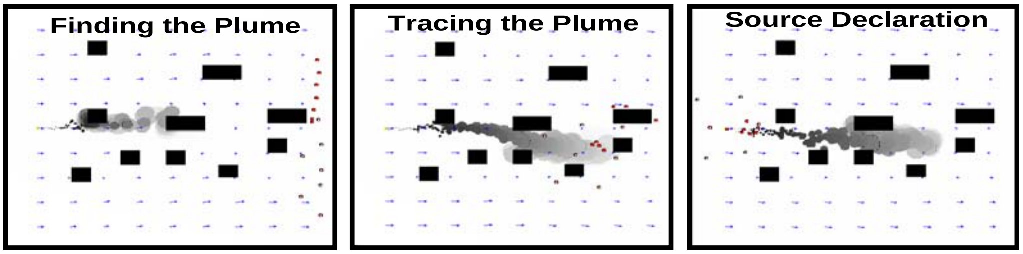


Fig. 6. Sample of visualization of proposed approaches for odor source localization in Advection-Diffusion odor model with ten obstacles.

$$\mathbf{V}_i(t) = \chi_\theta \mathbf{V}_i^*(t) \tag{13}$$

$$\mathbf{x}_i(t) = \mathbf{x}_i(t-1) + \mathbf{V}_i(t) \tag{14}$$

IV. Implementation Framework

The odor source localization problem in dynamic environments is related to several issues from biology, physical chemistry, engineering and robotics. This paper proposes a comprehensive approach to offer a sound technical basis for odor source localization in a dynamic environment.

A. Environment

In this paper, we adopted an extended Advection-Diffusion odor model by Farrell et al. [11] because of its efficiency. It represents time-averaged results for measurement of the actual plume, including chemical diffusion and advective transportation. In addition, the Advection-Diffusion odor model has a key factor to approximate the meandering nature of the plume, in that the model is sinuous.

The Advection-Diffusion model is composed of a large number of advected and dispersed filaments. Given a large number of filaments, the overall instantaneous concentration at $\mathbf{x}_o = (x, y)$ is the sum of the concentrations at that location contributed by each filament:

$$C(\mathbf{x}_o, t_o) = \sum_{t=1}^{M} C_i(\mathbf{x}_o, t_o) \tag{15}$$

where C is the concentration of the plume (*molecules/cm*^{3}), t_o is the number of iterations, and M is the number of filaments currently being simulated.

The Advection-Diffusion gas concentration at the location $\mathbf{x}_o$ due to the *i*-th filaments is expressed by:

$$C_i(\mathbf{x}_o, t_o) = \frac{q}{\sqrt{8\pi^3}} \exp\left[\frac{-r_i^{\,2}(t_o)}{R_i^{\,2}(t_o)}\right] \tag{16}$$

$$r_i(t_o) = \left|\mathbf{x}_o - \mathbf{P}_i(t_o)\right| \tag{17}$$

where q is the amount of odor released, R_i is the parameter controlling the size of the *i*-th filament; and $\mathbf{P}_i$ is changing positions of the *i*-th filament. (For further explanation on this model, see [11], section two and three.)

This model generates plumes that meander; in addition, the meander is coherent with the flow fields in the sense that downwind odor distribution from the source is the result of advection by the flow. Therefore, we extend the original equations from [11] to incorporate the obstacles in the environment. As a result, the environment becomes more realistic and complicated.

B. Robot Behavior

The gas source localization algorithm used in this work can be divided into three subtasks: plume finding, plume traversal and source declaration. Random search is employed until one robot encounters the plume. After finding the plume, the second task of the plume traversal proceeds. Particle Swarm concept will be applied to following the cues determined from the sensed gas distribution toward the source. The last task is the source declaration based on the certainty that the gas source has been found. If a robot senses the gas density that is beyond a certain threshold value, it means that the gas source location is specified; and hence, the searching behavior is terminated. Moreover, the search is terminated if the swarm robots fail to localize the odor source by the maximum iteration time step.

To ensure that the performance of proposed strategies is applicable to the hardware experiments, the simulation must contain the key features of the hardware setup. Firstly, the robot has a maximum velocity at which it can move. Hence, the value of velocity

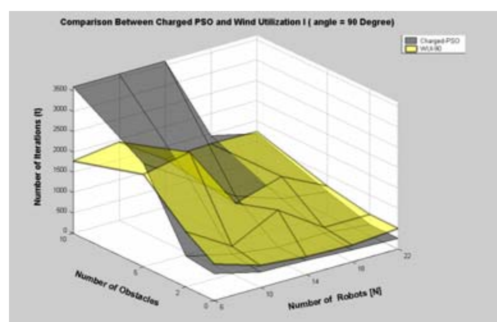


Fig. 7. Average convergence time in an obstacle environment, given the algorithm Wind Utilization I with 90° forbiden area.

vector can be restricted to the range [-*V*max, *V*max]. In this simulation, the maximum velocity is set to 0.05 (*m/s*), by following definition:

$$\mathbf{V}_i(t) = \min\left(\mathbf{V}_i(t), \mathbf{V}_{\max}\right) \tag{18}$$

Secondly, in order to incorporate a collision avoidance mechanism, which is not considered in the standard PSO algorithm, we assume that infrared sensors are equipped on each robot. Then the parameters of sensor noise and threshold value are added to model sensor responses. Assume that iteration time t of the robot in eq. (1) to (6) and iteration time t_o in eq. (15) to (17) is different time step

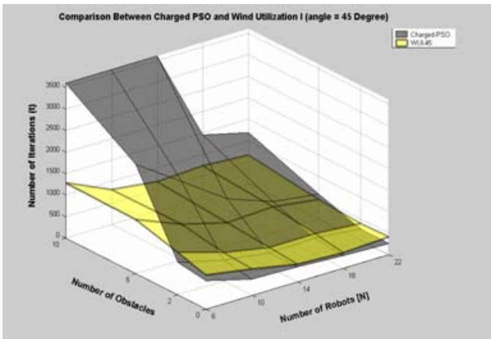


Fig. 8. Average convergence time in an obstacle environment, given the algorithm Wind Utilization I with 45° forbiden area.

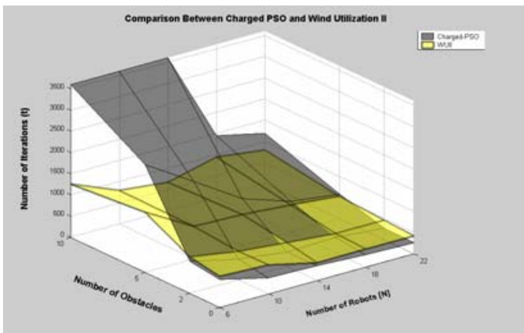


Fig. 9. Average convergence time in an obstacle-filled-environment, given the algorithm Wind Utilization II

resolution. Time correlation between time step t and time step t_o is explained as follow: The time scale of t has higher resolution than that of time step t_o and count up is represented as:

$$t_o + 1 = t_o + \Delta t \tag{19}$$

$\triangle t$ is the interval time step t_o in terms of time step t. Hence; t_o is represented with t by:

$$t_o = \left[\frac{t}{\Delta t} \right] \tag{20}$$

where $[X]$ is the Gauss's symbol. The sensor response is defined by:

$$S(t) = \begin{cases} C\left(\left[\dfrac{t}{\Delta t}\right]\right) + e(t) & \text{If } C > \tau \\ 0 & \text{Otherwise} \end{cases} \tag{21}$$

is the sensor's response, C is the gas concentration, e is the random sensor noise with $e << C$, and τ is sensors threshold.

Finally, the basic concept of PSO algorithm uses a common assumption, that all robots have accurate GPS that give the robot its global location and no error model is used for the position. These errors should be modeled as well to provide a more realistic situation. For this reason, a random position error sensor was added, as defined by:

$$\mathbf{x}_i(t) = (\mathbf{x}_i(t-1) + \mathbf{V}_i(t)) + \mathbf{D}_{error} \tag{22}$$

$$x_{error} = ((\frac{rand()}{RAND\_MAX}) \times 100) - RE \tag{23}$$

$$y_{error} = ((\frac{rand()}{RAND\_MAX}) \times 100) - RE \tag{24}$$

where RE is the range of error, and $x_{error} \neq y_{error}$ is an error position.

V. Experimental Example

To demonstrate capability our new comprehensive approach we run the simulation in different scenarios. Four scenario environments, one without obstacle and three with obstacles, are used for all the experiments. Figure 6 shows a sample of visualization a new algorithm to solve odor source localization with obstacle.

All experiments were run 25 times; the results reported are the averages. Then from the averages of convergence time in dynamical change of environment was plotted, as shows in Fig. 7. From Fig. 7, there is any intersection result, for free obstacle and simple obstacles (two obstacles) the CPSO is superior compare WUI-90, but for complex obstacles environment (five and ten obstacles) the WUI-90 is superior compare CPSO. It is therefore not surprising result, probably because the stopping behavior of the robot or the large of forbidden area as we mention in previous section.

The result the of the algorithm using the forbidden area function $\left|\theta_{forbidden}\right| \leq 45°$ compared to the results of using CPSO are shown in Fig. 8. In Fig. 8, the results intersect. This is partly due to a difference the gap space between obstacles, which depends on the number of obstacles in the environment. For an environment with only two obstacles, the results for the CPSO and WUI algorithms were very similar. However, for and environment with five to ten obstacles (a complex environment), the WUI-45 is obviously superior compared to the CPSO algorithm.

The weakness of implementing Wind Utilization I, is the need for tuning the forbidden area parameter (i.e. tuning from $\left|\theta_{forbidden}\right| \leq 30°, \left|\theta_{forbidden}\right| \leq 45°, \left|\theta_{forbidden}\right| \leq 60°$**).** If the robot stays in the area $\left|\theta_{forbidden}\right| \leq 45°$ the results are promising. For implementation, we use the controlling parameter χ_θ to decide the velocity of the robot. Then we compared the results with the CPSO algorithm, as shown in Fig. 9. In Fig. 9, the results again intersect. For an environment with only two obstacles, the results for the CPSO and WUI algorithms were very similar. However, for and environment with five to ten obstacles (a complex environment), the WUII is superior compared to the CPSO.

For more representative analysis the effect of error of position and error of odor sensor was investigated. Table 5.1 shows typical time development of global best coping with dynamical change of environment used WUII algorithm with employed uncertain sensor parameters with various odor sensor and position errors. Fourteen robots with two obstacles employed in this result. From Table 1, it can be concluded; adding the positioning error will slightly decrease the performance. The positioning error of 50 (cm) is a realistic assumption, considering the dimension of the robot (10 cm).

Table 1. Average of convergence time in obstacle environment used WUII with employed uncertain sensor parameters. (Repeated 25 times)

Position Error (cm)	**Odor Sensor Error (ppm)**		
	0	**0.2**	**1**
0	664 ± 179	794 ± 297	894 ± 199
50	984 ± 511	1071 ± 814	1100 ± 714
100	1150 ± 514	1210 ± 712	1325 ± 613

VI. Conclusions and Prospect

The PSO was implemented for controlling autonomous robots to search for an odor source in dynamic, obstacle-filled environments . When comparing CPSO and DR PSO results, the CPSO gave better results for the convergence time to find an odor source location. A wind-utilization function for OGR was also implanted with the CPSO algorithm and compared in the WUI-45 and WU-II analyses. Both these analyses WUI-45 and WUII showed promising results. Furthermore, the WUII algorithm, which uses the parameter χ_θ for controlling the velocity of the robot, had success even without a tuning parameter for noise and variable sensor parameters.

These proposed approaches can solve such dynamic environment problems but in practical, for real natural environment, the robot will find various situations related with multi study from biology, physical chemistry, engineering and robotic. Unresolved problem still find in implementation phase. Most of those could be grouped into one of the following categories:

1. Environments

The previous report used Advection-Diffusion equation to model the environment. Since the Gaussian and Advection-Diffusion equation does have only one optimum, it means, as long as such model is used for the environment, no matter which algorithm is used, the agents will eventually find the odor source. How to cope with some issues like two or more different locations having the similar optimum? We also try to solve such kind of multiple location of odor in future work.

2. Performance Measurement in Dynamic Environment

Some researches argue that the performance measure for static problem, such as with MEAN statistic, cannot directly be applied to dynamic environment. These measures are used either to report the performance of an algorithm after a fixed number of iterations (or function evaluations in our case the odor distribution functions), or to produce a performance profile over all iterations. In the first case, if applied to the dynamic environment, the final performance will not be an accurate account of performance under all the environment changes. The ability to recover from environment changes is not reflected. Morrison provides a summary of performance measures for dynamic environments with respect to evolutionary algorithms. We also try to analyze the feasibility conjectures referred to above, in future work.

3. Algorithm Optimization

The common problem using PSO lies in the parameter tuning to find the optimal solution. Most of the researchers use the cross validation or try and error methods to tune parameters. Further algorithm development in simulation will include online learning, which system can learn parameters from environment.

4. Real Hardware Implementation:

An important stage will be on porting the simulation to actual robots in a laboratory experiment. Multiple autonomous mobile robots will use for actual robot experiment. This robot can move autonomously, additionally a robot can communicate each other by wireless LAN. The robot used TGS-822 gas sensor for alcohol and volatile vapor detection from Figaro Inc. The sensing element of TGS-822 gas sensors is a tin dioxide (SnO2) semiconductor that has low conductivity in clean air. In the presence of a detectable gas, the sensor's conductivity increases depending on the gas concentration in the air. A simple electrical circuit can convert the change in conductivity to an output signal which corresponds to the gas concentration. The TGS-822 has high sensitivity to the vapors of organic solvents as well as other volatile vapors. It also has sensitivity to a variety of combustible gases such as carbon monoxide, making it a good general purpose sensor. Via the ad-hoc wireless LAN, each robot can collect the gas concentration value and choose the best one. Then the position of the robot can be determined covering camera.

The final goal of the research is construct a whole system, software and hardware implementation, for solving distributed odor source localization in real environment through a newly proposed Modified Particle Swarm Optimization algorithm (MPSO).

Acknowledgment

The authors are grateful to Prof. J. A. Farrell from the University of California, Riverside, U.S.A., for his support in advanced turbulent-environment source-code.

References

[1] M. Wandel, A. Lilienthal, T. Duckett, U. Weimar, and A. Zell. Gas distribution in unventilated indoor environments inspected by a mobile robot. In Proceedings of the IEEE International Conference on Advanced Robotics (ICAR'03), 2003.

[2] X. Cui, C. T. Hardin, R. K. Ragade, and A. S. Elmaghraby. A swarm-based fuzzy logic control mobile sensor network for hazardous contaminants localization. In Proceedings of the IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS'04), 2004.

[3] Ishida, H. Nakayama, G. Nakamoto and T. Moriizumi, T. Controlling a gas/odor plume-tracking robot based on transient responses of gas sensors", IEEE Sensors Journal, Vol. 5. No.3. June 2005.

[4] Adam T. Hayes, A. Martinoli and R. M. Goodman, "Distributed Odor Source Localization," IEEE Sensors Journal, Vol. 2. No.3. June 2002.

[5] D. Zarzhitsky, D. Spears, and W. Spears. Distributed Robotics Approach to Chemical Plume Tracing. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'05), 2005.

[6] R. A. Russell and A. H. Purnamadjaja. Odor and airflow: Complementary senses for a humanoid robot. In Proceedings of the 2002 IEEE International Conference on Robotics and Automation, 2002.

[7] W. Jatmiko, T. Fukuda, T. Matsuno, F. Arai and B. Kusumoputro," Robotic Applications for Odor-Sensing Technology: Progress and Challenge, WSEAS Transaction on System、Issue 7, Volume 4, July 2005.

[8] W. Jatmiko, K. Sekiyama and T. Fukuda," A PSO-based Mobile Sensor Network for Odor Source Localization in Dynamic Environment: Theory, Simulation and Measurement", Proc. of the IEEE CEC-WCCI "Congress Evolutionary Computation - Word Congress on Computational Intelligence" pp. 3781-3788, 2006.

[9] W. Jatmiko, K, Sekiyama and T. Fukuda," A Mobile Robots PSO-based for Odor Source Localization in Dynamic Environment Advection - Diffusion Environment", Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robotics and Systems (IROS 2006), pp. 4527-4532, 2006.

[10] W. Jatmiko, K**.** Sekiyama and T. Fukuda, "Modified Particle Swarm Robotic for Odor Source Localization in Dynamic Environment ", The International Journal of Intelligent Control and Systems: Special Issue on Swarm Robotic, Vol. 11, No 3, pp.176-184, September 2006.

[11] Jay A. Farrel et all , "Filament-based atmospheric dispersion model to achieve short time-scale structure of odor plumes," Environment Fluid Mechanics , vol. 2, pp. 143-169, 2002.