

# Exploiting Bacteria Swarms for Pollution Mapping

John Oyekan, Huosheng Hu and Dongbing Gu  
School of Computer Science and Electronic Engineering  
University of Essex, Wivenhoe Park, Colchester CO3 4SQ, U.K.  
*Email: jooyek@essex.ac.uk; hhu@essex.ac.uk; dgu@essex.ac.uk*

**Abstract**—Inspired by the simplicity of how nature solves its problems, we develop a flocking controller that would enable the localisation and subsequent mapping of environmental pollutants. Pollutants could range from chemical leaks to invisible air borne hazardous materials. We use simulation results to validate our approach and then briefly discuss how to implement the controller onto a real robotic platform. Our motivation is to use the advantages offered by swarm robotics- simple, multiple and cheap agents- to achieve a collective complex single goal of mapping an environmental pollutant spread over a large area. We aim to make our approach as simple as possible yet highly effective in generating the map.

**Keywords** Bacterium Inspired Algorithm, Environmental Monitoring Flocking.

## I. INTRODUCTION

The global warming has been a major topic in the environmental sciences in the past few years, which has led researchers in various fields to find new novel ways of monitoring the environment. One of the novel ways is the use of various multi agent theories of which swarm robotics is one of them[1]. Swarm Robotics involves the use of many agents to perform a task that is not possible or that is very difficult to achieve with a single agent[2]. By using swarm robotics, multi agents could monitor the pollution spread and how it changes with time. In addition, if one of the agents fails, the mission would still carry on with little loss in performance of the entire system. Furthermore, using swarm robotics enables the system to be everywhere at once and also enables the condition of pollutants to be viewed at various locations all at once. It also enables users to view how conditions at one location affect conditions at other locations in real time. From this, a real time 3D map of the pollutant and changes in its condition can be generated.

In order to build a map of environmental pollutants using multiple robotic systems, two problems have to be solved. Firstly, a robotic agent has to be controlled in such a way to make sure that it is placed at the best position in the environment to get a good reading of an environmental pollutant. Secondly, a way of building a map of the measured quantity is needed. In this paper, we address the first problem. There have been numerous research work done in this area.

Cortes et al in[3] use a voronoi approach to divide an area of interest into voronoi partitions and then control the robotic agents to place themselves in the center of each voronoi

partition. This they argued enables them to get the optimal reading in that sector of the environment. However, this approach requires a high computational and communication cost and can only be used in polygon derivative environments. Shucker et al in[4] use a gabriel graph theory to achieve the effective placement of agents in the environment to track a target. This approach requires that agents are able to communicate to an extent over a large distance with agents at other positions. This might not be practical in a real world scenario.

Lilienthal et al achieve an effective coverage of an area by moving their agents in a predefined manner in the area to be covered[5]. This method quickly becomes ineffective if a large distance is to be covered by the single agent. Work done by Mesquita et al uses a technique based on the bacteria chemotaxis behaviour to arrange them in the environment based on the signal to be monitored[6]. However, in their work, they assumed that the structure of the signal to be monitored is known before deployment of their agents.

Zarzhitsky et al. used a term called fluotaxis to direct a swarm of robotic agents in localizing a plume source. This term was also inspired partly by the chemotaxis behaviour of bacteria. However, they used an artificial physics framework to achieve the flocking behaviour of the swarm which is arguably not biologically plausible[7]. In our work, we investigate the use of a bacterial chemotaxis behaviour in combination with flocking algorithms to position the agents in the environment based on the density profile of the environmental pollutant to be measured. It is our aim to arrange the agents so that areas of high environmental pollutant concentration receive more agents than areas of low pollutant concentration. We believe that by doing this, areas with more interesting data are monitored closely than areas with less interesting data[8]. This makes it possible for agents to keep searching for more interesting data if there is a possibility of any appearing in the environment. In addition, we believe that our algorithm is simple to implement and not environmental specific.

In this paper, we present results of using a combination of swarm algorithm and bacterial algorithm to achieve the monitoring of pollutants. The developed controller is presented in Section II. Section III discusses the experimental setup while Section IV presents the results of our simulation. A

robotic implementation is given in Section V. Finally, a brief conclusion and future work are presented in Section VI.

## II. DEVELOPED CONTROLLER

As mentioned previously, we use both a flocking controller and a bacteria controller to achieve the distribution of agents in the environment. We combined the output of both controllers and used gains for each output as shown in Equation 8. We shall now discuss the details of each individual controller.

### A. Bacteria Controller

Based on the Berg and Brown model[10], a bacterium motion is composed of a combination of tumble and run phases. The frequency of these phases depends on the measured concentration gradient in the surrounding environment. The run phase is generally a straight line while the tumble phase is a random change in direction with a mean of about 68 degrees in the E. Coli bacterium. If the bacterium is moving up a favourable gradient, it tumbles less thereby increasing the length of the run phase and vice versa if going down an unfavourable gradient. This behaviour was modelled by Berg and Brown by fitting the results of their experimental observations in[9] with a best fit Equation in[10]. This model is shown below:

$$\tau = \tau_o \exp(\alpha \frac{dP_b}{dt}) \quad (1)$$

$$\frac{dP_b}{dt} = \frac{k_d}{(k_d + C)^2} \frac{dC}{dt} \quad (2)$$

where  $\tau$  is the mean run time and  $\tau_o$  is the mean run time in the absence of concentration gradients,  $\alpha$  is a constant of the system based on the chemotaxis sensitivity factor of the bacteria,  $P_b$  is the fraction of the receptor bound at concentration  $C$ . In our work,  $C$  was the present reading taken by our Robotic agent.  $K_d$  is the dissociation constant of the bacterial chemoreceptor.  $\frac{dP_b}{dt}$  is the rate of change of  $P_b$ .

From our previous work in [11], we discovered that using memory aided the convergence of the Robotic agents to a source. As a result, we used the version of the Berg and Brown model that takes the effect of the previous positions into account. This is shown below:

$$\tau = \tau_o \exp(\alpha \frac{\overline{dP_b}}{dt}) \quad (3)$$

$$\frac{\overline{dP_b}}{dt} = \tau_m^{-1} \int_{-\infty}^t \frac{dP_b}{dt'} \exp(-\frac{(t-t')}{\tau_m}) dt', \quad (4)$$

where  $\frac{\overline{dP_b}}{dt}$  is the weighted rate of change of  $P_b$ , while  $\tau_m$  is the time constant of the bacterial system. The above Equations determine the time between tumbles and hence the length of runs between tumbles.

During the tumble phase, the agent can randomly choose a range of angles in the set  $\sigma \varepsilon \{0, \dots, 360\}$  by randomly choosing co-ordinates. This made it possible for our agents to backtrack if there is a favourable gradient behind it. We shall call the chosen co ordinates from the bacteria controller for agent  $i$   $x_i^b$ . From our previous work in [11], we used a adaptable velocity that is determined by

$$\beta(t) = \frac{\beta_o * v_k}{C(t)} \quad (5)$$

where  $\beta$  is a dynamic velocity that depends on the present reading  $C(t)$  of the environmental quantity,  $\beta_o$  is the standard velocity without any reading and  $v_k$  is a constant for tuning the dynamic velocity.

### B. Flocking Controller

To implement the flocking controller for agent  $x_i$ , we used a modified flocking controller as shown below [12].

$$x_i^f(t) = [-K(dist(t) - d)](x_i(t) - x_j(t)) + [H(x_i(t) - x_h(t))] \quad (6)$$

$$dist(t) = ||x_i(t) - x_j(t)|| \quad (7)$$

where  $x_h(t)$  is the position of the agent with the highest measurement in the neighbourhood of  $x_i$ . The neighbourhood is determined by the communication radius of each agent as discussed in Section III.  $dist(t) = \sqrt{(x_i(t) - x_j(t))^T (x_i(t) - x_j(t))}$ .  $K > 0$  is used to determine the magnitude of the repulsion force,  $(dist(t) - d)$ , between agent  $x_i$  and agent  $x_j$ . Constant  $H$  is the attractant gain for the force between agent  $x_i$  and the agent  $x_h$  with the highest environmental quantity measurement in the neighbourhood.

We then combined the two behaviours and calculate the new position of the agent  $x_i$  using the following Equation:

$$x_i(t+1) = \beta(t) + (F * x_i^f(t) + B * x_i^b(t)) \quad (8)$$

where constants  $F = 0.01$  and  $B = 0.94$  are gains for the flocking and bacteria behaviour respectively and  $x_i^f$  is the output of the flocking behaviour for agent  $i$ .

## III. EXPERIMENTAL SETUP

To test the algorithm, a simulated arena with a dimension of 600 pixels by 600 pixels was developed. We used kinematic models for the simulated robots based on our previous work in [13]. They had a dimension of 10 pixels by 10 pixels and an array of simulated chemical sensors in the center of the robot. This array of chemical sensors had a dimension of 4 pixels by 4 pixels. It is assumed that each individual chemical sensor making up the array returns 1 or 0 as output. If a chemical sensor detects a pollutant particle in a location, it returns a 1 and otherwise a 0. To measure the concentration of pollutants in the robot's position, the values

of each chemical sensor in the array is added up to get the total measured concentration in that location.

Each agent is able to communicate with other agents within a radius of 60 pixels. To simulate the restrictions on communication bandwidth, each agent is only able to buffer up 5 readings from its neighbourhood agents.

Other assumptions taken in our experiment include:

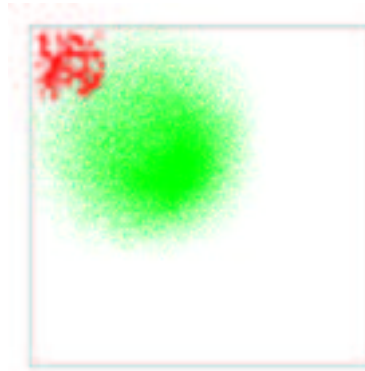
- A simple generated air pollutant having the distribution shown in Figure 1. There were no clear concentration gradient boundaries in the pollutant.
- The air pollutant is stationary and not moving. The effects of wind changes or air convection currents are not investigated but will be done in later studies.
- The robots were placed initially at the edges of the pollutant as in [6].
- Each robot knows its position in the simulated arena. This is possible in practice by using GPS.
- In the flocking implementation, the collision avoidance scheme was supposed to be obtained from the implemented flocking controller in Equation 6. On real robots, ultrasonic sensors could be used to aid the flocking controller for avoiding obstacles.
- In the bacteria chemotaxis investigations, it is assumed that once the robots find a concentration value greater than a threshold of 14 they have reached the source.
- It is assumed that the chemical sensors used in the experiment were noiseless during the simulation. We plan to introduce noise through the use of a random number generator in subsequent experiments.

#### IV. SIMULATION AND RESULTS

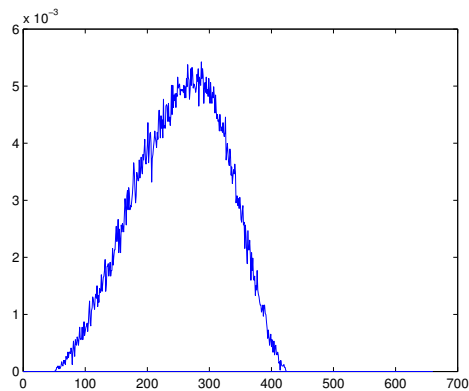
##### A. Bacterial Chemotaxis Behaviour

Before combining the two behaviours together, we tested the bacterial chemotaxis behaviour. We performed an exhaustive search to find out what value would be the best for the parameters of Equations 2, 3 and 4 using a constant velocity. The experiments were performed using 100 robots and running the simulation of each newly changed parameter value 40 times to get a good distribution of results.

We discovered that more agents were able to find the source by having smaller values of standard tumble length. Using a small standard tumble length made the agents rotate continuously on a spot due to continuous tumble phases. This led to a smaller response time leading to faster convergence [14]. The effects of a selection of other parameters are displayed in Figure 2. The  $kd$  parameter (responsible for determining how sensitive a bacterium's receptor is to attractants in the environment) increases the number of agents to a certain level after which increases in the  $kd$  value has no effect on the number of agents localizing at the optimal of the pollution source. This could be as a result of the saturation of the sensors. In other words, if the pollutant concentration were increased, then this would result in a shift in the  $kd$



(a)



(b)

Fig. 1. Simulated Arena with pollutant- Fig. (a); Distribution of pollutant- Fig. (b)

parameter curve in Figure 2. The  $\alpha$  parameter is responsible for system amplification. This could be used to amplify the response of the system in the presence of weak readings from the environment.

Our results from this experiment were more promising when compared to our previous results in [11] for the same time frame. The value of the median of the number of robots finding the source in our previous results was 49.5 in a time frame of 30 seconds. In our present results, we were able to achieve about a median of 97 robots in the same time period.

##### B. Combining the Behaviours

By combining the bacteria behaviour with the flocking behaviour, we were able to arrange the flock to fit the underlying environmental pollutant that was being monitored. Time steps of this is shown in Figure 3 below with their corresponding distributions.

As can be seen in Figure 3, the mean of the distribution of the flock is close to the mean of the distribution of the pollutant. We discovered that by changing the parameters of our controller, we could change the spread of the agents in the pollutant. This feature is useful in controlling the spread

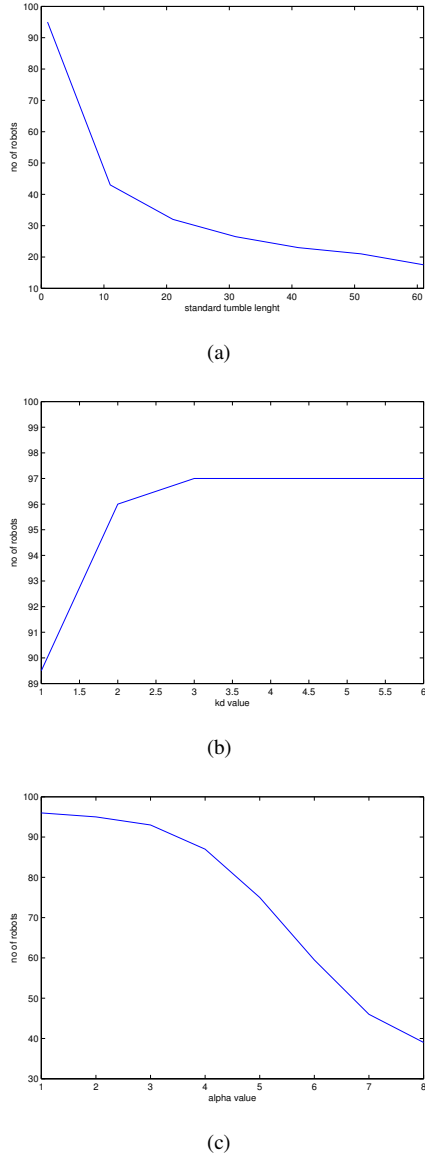


Fig. 2. Number of Robots with adjusting: the standard tumble length  $\tau_o$ -Fig. (a); the  $kd$  parameter-Fig. (b); the  $\alpha$  parameter-Fig. (c)

of the flock in the environment and hence making it possible for them to look for other sources.

### C. Controlling the agents spread

In order to make the investigation of the effects of the parameters more simplified, we changed our pollution distribution to a gaussian distribution shown in Figure 4 having a mean of 250 and a standard deviation of 57. Using a gain of 1 for both behaviours, we investigated the effects of the changes in the value of the  $\tau_o$  parameter and the value of the  $v_k$  parameter. We then measured the mean and the standard deviation of the flock after 2 minutes of running. We ran

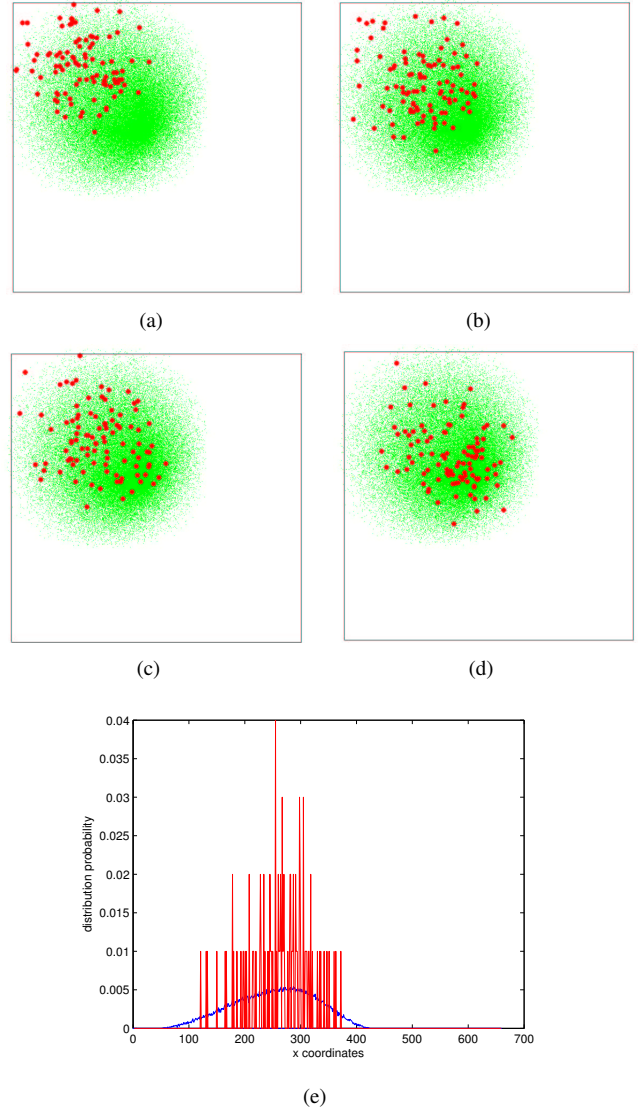


Fig. 3. Showing various time frames for the flock distribution in pollutant-Fig. (a) to Fig. (d); the probability of the flock distribution and pollutant distribution-Fig. (e)

the simulation 40 times to get a good representation of the experimental results for each change in parameter value.

It can be seen from Figure 5 that the mean of the flock (localisation property) reduces as  $\tau_o$  increases while the standard deviation (spread of the flock about the mean) increases. From the graphs above, it can be seen that to get a balance of both properties a value of 60 can be chosen.

In addition, we discovered that changing the value of the  $v_k$  parameter of Equation 5 also has effect on the spread of the agents as can be seen in Figure 5. However, changing this parameter is not really practical in real world applications due to energy and velocity restrictions of the present platforms. Nevertheless, by choosing the right values for both  $v_k$  and  $\tau_o$

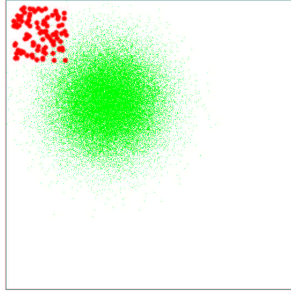
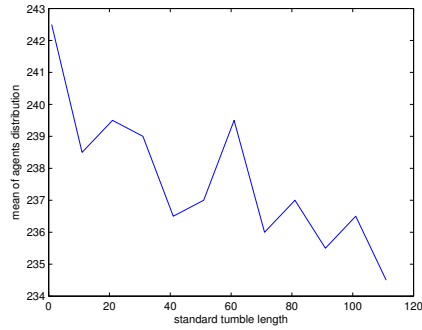
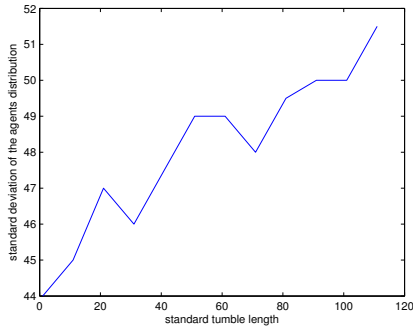


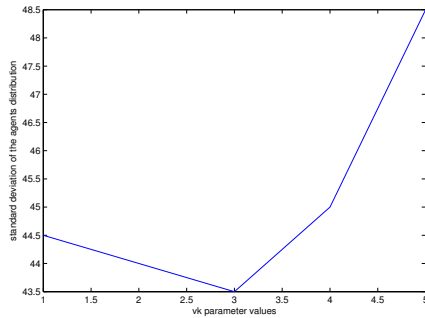
Fig. 4. Pollutant with a gaussian profile.



(a)



(b)



(c)

Fig. 5. Fig. (a) shows the mean of the flock distribution and Fig. (b) shows the standard deviation of the flock distribution as  $\tau_o$  increases. Fig. (c) shows how the standard deviation (spread) of the flock distribution varies with changes in  $v_k$ .

parameters, it is possible to use them to reinforce each other. The  $v_k$  did not have much effect on the mean of the flock distribution.

Having the ability to control the spread of the flock enables the map of the pollutant can be built in varying degrees. The spread can be controlled to either that readings from interesting points near the source of the pollutant or could be controlled to get readings from a larger pollutant coverage area.

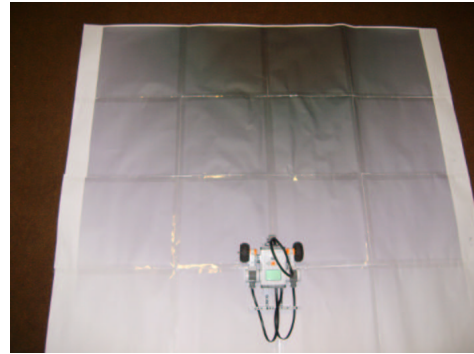


Fig. 6. Lego mindstorm platform with light sensor.

## V. IMPLEMENTATION ON A LEGO MINDSTORM ROBOT

In order to investigate the effects of our algorithm on various platforms, we implemented the algorithm on a Lego mindstorm platform having a differential drive system as shown in Figure 6. We used Lejos (A java derivative programming language to program the robot. For the environmental pollutant, we printed a gradient of black color on paper. We used a light sensor to read the values of the color from the paper and then responded accordingly. The values of the reading from the light sensor was between 0 and 65. We had to use a large  $\alpha$  value of 1000 during the control of our system. We discovered that a range of  $\alpha$  values might be necessary because of varying light levels and battery level changes. We also used a  $kd$  parameter of 2 and did not vary the velocity of the platform according to the light sensor readings. To simulate a random change in direction, we generated and used random angle values in the set  $\sigma \in \{0, \dots, 360\}$  for each wheel during a tumble. During the tests, we placed the robot so that there was an amount of pollutant on all sides of the platform.

Even the presence of noisy readings (as result of the wrinkled paper) the robot was still able to localize the source of the pollutant. We discovered that by using a smaller value of tumble length  $\tau_o$ , faster localisation at the source is achieved. This is shown in Figure 7. However, we found out that care must be taken when choosing this value on this particular platform. This is because a small value of  $\tau_o$  would result in the platform spinning around on its axis and making

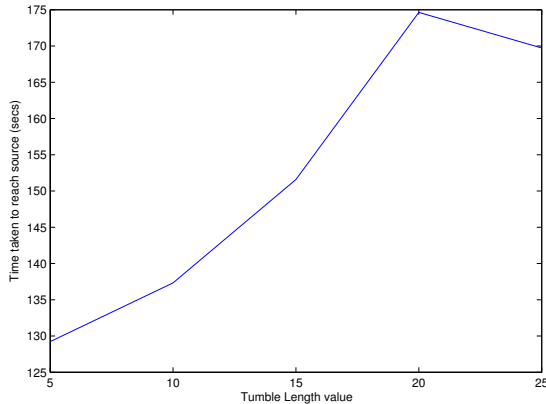


Fig. 7. Fast localisation with light sensor.

no progress. We plan to investigate other parameters in future experiments.

## VI. CONCLUSION AND FUTURE WORK

We have presented and used a controller that makes it possible to find a pollution source using a bacteria inspired algorithm. We have shown it is possible to use this controller with a flocking algorithm to direct a flock of agents to a pollution source. We have also shown that it is possible to control the spread of the agents by changing the various parameters of our flocking controller. This property would make it possible to control the degree of the mapping of an environmental pollutant. We have shown that the bacteria controller can be implemented on a robotic platform. In addition, the effects of changing pollutant profiles has been carried out with very promising results. This was done by changing from a biased gaussian function in Figure 1 to the one in Figure 4.

Due to the gradient based method of searching, the bacterial controller might fail in a high turbulent environment. We believe that using this controller with a flocking controller performance might be improved due to the advantage of cooperative foraging. We also believe that by investigating the parameters of the bacterial controller closely, we can adapt it to work in a high turbulent environment. Other future work would involve generating a map of the environmental pollutant at a base station. This might sound trivial but when a limited number of agents is used to generate a map of an environmental pollutant spread over a large area, this might become challenging. We plan to use the real time dynamic changes in the pollutant, the readings from the sensor and mapping building methods to generate a map. Presently, most map generating techniques are for robot localisation.

We also believe that we can improve the performance of our controller by replacing the exponential function on our controller by another function. This would be done at run

time using a learning algorithm which we would develop. This would also involve dynamic adaptation of the  $\alpha$  value to cope with environmental and platform noise. Investigation of the effects of the parameters on a real robotic platform still has to be investigated and so is the effect on flocking. Other areas include further investigation of collision avoidance between agents, and implementing a biased random angle during tumbles as is observed in a real life E.Coli bacterium.

## ACKNOWLEDGEMENTS

We would like to thank Renzo De Nardi for all his numerous advice.

## REFERENCES

- [1] M.V.Ramana, V.Ramanatham, D.Kim, G.C.Roberts and C.E.Corrigan, *Albedo, atmospheric solar absorption and heating rate measurements with stacked UAVs*, Quarterly Journal of the Royal Meteorological Society, 1913-1931, 2007.
- [2] E. Sahin, *Swarm robotics: From Sources of Inspiration to Domains of Application*, Swarm robotics, Springer Berlin/Heidelberg, Vol. 3342/2005, January 2005.
- [3] J. Cortes, S. Martinez, T. Karatas, F. Bullo, *Coverage Control for Mobile Sensing networks*, IEEE Transactions on Robotics and Automation, vol. 20, no. 2, pp. 243-255, April 2004.
- [4] B. Shucker, T. Murphey, and J.K. Bennett, *An Approach to Switching Control Beyond Nearest Neighbor Rules*, American Control Conference, 16 June 2006.
- [5] A. Lilienthal and T. Duckett, *Building gas concentration grid maps with a mobile robot*, Robotics and Autonomous Systems, vol. 48, no. 1, pp.3-16, 2004.
- [6] A.R. Mesquita, J.P.Hespanha, and K. Astrom, *Optimotaxis: A stochastic Multi-agent Optimization Procedure with Point Measurements*, Springer-Verlag Berlin Heidelberg 2008.
- [7] D. Zarzhitsky, D.F.Spears and W.M. Spears, *Swarms for Chemical Plume Tracing*, In Proc of the IEEE Swarm Intelligence Symp. (SIS'05)
- [8] M. Schwager, F. Bullo, D. Skelly and D. Rus, *A Ladybug Exploration Strategy for Distributed Adaptive Coverage Control*, Proceedings of International Conference on Robotics and Automation, Pasadena, CA, May 2008.
- [9] H.C. Berg and D.A. Brown, *Chemotaxis in Escherichia coli analysed by 3D tracking*, Nature vol.239, October 1972.
- [10] D.A. Brown and H.C. Berg, *Temporal Stimulation of Chemotaxis in Escherichia coli*, Proc. Nat. Acad. Sci. USA Vol. 71, No. 4, pp. 1388-1392, April 1974.
- [11] J.O. Oyekan and H. Hu, *Toward Bacterial Swarm for Environmental Monitoring*, IEEE ICAL,2009, Awaiting acceptance.
- [12] K.M. Passino, *Biomimicry for Optimization, Control and Automation*, Springer-Verlag London Limited, 2005.
- [13] J. Oyekan and H. Hu, *Towards Autonomous Petrol Behaviours for UAVs*, Proceedings of UK EPSRC Workshop on Human Adaptive Mechatronics, Staffordshire University, Stafford, U.K., 15-16 January 2009.
- [14] G.A Jackson, *Simulating chemosensory responses of marine organisms*, Limnol. Oceanogr., 32(6), 1987, 1253-1266, 1987.