TASTI 2 Searches for Chemicals in the Presence of Obstacles

R. Andrew Russell
Intelligent Robotics Research Centre, Monash University
Clayton, AUSTRALIA
andy.russell@eng.monash.edu.au

***Abstract*—This paper describes current progress in the development of a robot for locating patches of chemical deposited on the ground. It is envisaged that this form of chemical location will be useful for such tasks as verifying the effectiveness of decontamination and finding spills of non-volatile chemicals. In many cases the approximate location of the chemical is known and the search extends outwards from this point. A number of biologically inspired algorithms are available for conducting this kind of search. The aim of this project is to implement these algorithms using a specially designed robot and in particular to investigate ways in which physical obstacles can be accommodated. The design of the searching robot TASTI 2, its search algorithms and the modifications to allow obstacle avoidance are described. In addition, results of simulations and practical experiments are presented.**

I. Introduction

This project involves the creation of a mobile robot that can search for chemical deposits on the ground while also avoiding a sparse field of obstacles. It is envisaged that mobile robots equipped with tongue sensors could be used to assess the efficiency of chemical cleaning and decontamination operations. For example, in the manufacture of particularly hazardous chemicals such as cytotoxic drugs decontamination is of great importance and automated robotic systems would reduce the risk of human exposure. Robots could also locate chemical spills or chemicals carried to the surface from underground leaks. There is also increasing interest in the use of chemical signals as a form of communication between robots [1]. Non-volatile chemicals deposited on the ground could provide a long-lived method of communicating between robots.

In order to locate chemical deposits on the ground robots require two main capabilities in addition to the basic requirement for mobility. The robots should be able to detect the target chemicals and they should be capable of performing a search pattern. Biological systems are often used as inspiration by roboticists because of the simple and robust methods they employ. There are biological models for the special capabilities required by the searching robot.

A number of different creatures sample their environment by sensing non-volatile chemicals. The predatory snail *Euglandina rosea* follows the slime trails of its prey (other snails and slugs) using their long mobile lip extensions [2]. Snakes also gather information about their surroundings and track their prey by collecting scent particles from the air and ground using their forked tongue. These particles are then 'tasted' by inserting the tongue into a sensory site in the roof of the snake's mouth called the Jacobson's organ [3]. Many insects taste the ground that they stand on through their feet and use this information for locating food and avoiding unpleasant chemicals [4]. Human taste gives us first-hand experience of sensing non-volatile chemicals [5]. These examples provide a source of inspiration for sensor design.

Search is another important capability for many biological creatures. One kind of search is commonly performed to locate a burrow or nest once the creature fails to find this at the expected location. Gathering of food items is another application for search. Harvester ants execute a radial pattern when searching for food around their nest [6]. A spiral search could be another simple and efficient area search technique. There is some difference of opinion concerning whether biological creatures actually employ this kind of search [7]. However, it is entirely feasible that they could if the search was restricted to a region close to the origin so that dead reckoning errors did not become too large.

A major problem for any robot attempting to follow a predefined search pattern is negotiating obstacles. One solution would be to create a map of the search area and to superimpose the proposed search path and the obstacles. Obstacles could then be avoided by planning paths around the known obstacles linking the ends of the part of the search path that is obstructed by the obstacle. If further obstacles become apparent as the search proceeds then these could be incorporated into the map and appropriate avoidance paths planned. Such a sense-map-plan-execute scheme requires complex sensors and large computing and memory resources. As Rodney Brooks explained, robot control schemes where there is a more direct connection between sensors and actuators are faster, simpler and easier to extend [8]. Although not developed in terms of Brooks' subsumption architecture, this paper considers a method of avoiding obstacles with similar reactive control suitable for simple robots and requiring the minimum of computing/sensing resources.

The next section introduces the kind of biologically inspired search algorithms that the robot TASTI 2 employs. Section III proposes a solution to the obstacle avoidance problem. Design features of TASTI 2, the robot designed to implement the obstacle avoidance and search tasks are considered in Section IV. Results of practical experiments are presented in the following section and conclusions and proposals for further work are given in Section VI.

II. Search Algorithms

In this project the aim is to investigate simple robotic systems that require minimal sensing and computing resources. It is envisaged that simple biologically inspired search algorithms would be directly applicable to this kind of robotic system and would result in robust and effective performance. The chosen search scenario involves searching outwards from a starting point that is the initial best-guess for the location of the target. In terms of equivalent biological situations this could correspond to a creature searching for its nest or burrow having failed to find it at the expected location.

During this project a number of search algorithms have been investigated. However, because the main emphasis of this paper is searching in the presence of obstacles and not the search algorithms themselves only one representative search algorithm will be described. In all of the search algorithms implemented on TASTI 2 the search proceeds in a repeated sequence of fixed-length straight-line moves followed by an on-the-spot turn and a sensing operation.

A. The Radial Search

A radial search involves a repeated sequence of moving a set distance away from the starting point and then returning. For each outward journey a new heading is chosen. The efficiency of the radial search is improved by arranging that the outward and inward paths form a loop so that a greater area is covered.

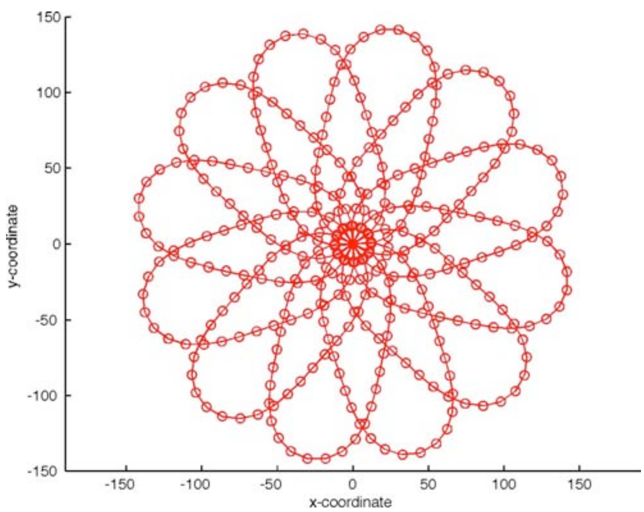

Figure 1 A radial search comprising 12 petals, 30 steps per petal and step size of 12 units (MATLAB simulation).

For this project a looping form of radial search has been developed and is encapsulated in the following equations describing the change of heading of the robot:

$$\Delta\theta_i = \frac{4\pi}{n}\left(\sin\left(\frac{2\pi(i+n/4)}{n}\right)\right)^2 \qquad (1)$$

if $\mathrm{rem}(i,n) == n/4$ then

$$\Delta\theta_i = \Delta\theta_i + \frac{2\pi}{petals} \qquad (2)$$

where:

- i = the number of steps taken by the robot
- *heading* = robot heading in radians
- *petals* = the number of petals in the search pattern
- n = the number of robot steps in one petal (should be divisible by 4)

To complete one 'flower' the robot must take *n*petals* steps. The overall size of the search pattern is governed by the distance the robot travels between sampling points and the number of loops in the pattern is controlled by the variable *petals*.

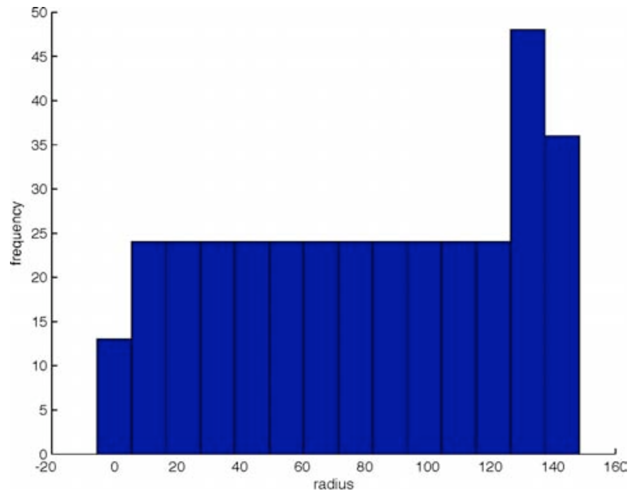

Fig. 2 A histogram of density of sensing point distribution as it varies with distance from the centre of the search area.

The sampling point density histogram given in Fig. 2 shows an almost equal distribution of sampling points for a given radius over most of the range of the search pattern. Because sampling area increases with radius this indicates that the density of points per unit area decreases with increased distance from the starting point.

As a robot performs its search path in a real world situation it is very likely to come across an obstacle. Assuming that the search path does emerge from the obstacle then the aim would be to perform a detour path that connects the ends of the path where they are blocked by the obstacle.

III. Obstacle Avoidance

As explained in the previous section, the path of the searching robot consists of a repeated sequence of a movement straight ahead for a fixed distance, a turn on-the-spot, and a sensing operation. The sequence terminates when a significant chemical concentration is sensed. This path can be described as follows.

The pose p_i of the robot at step i is:

$$p_i = \begin{bmatrix} x_i \\ y_i \\ \theta_i \end{bmatrix} \quad (3)$$

where

x_i = the x-coordinate of the centre of the robot,
y_i = the y-coordinate of the centre of the robot, and
θ_i = robot heading in radians

With each step the robot moves from one sampling point (pose p_i) to the next (p_{i+1}):

$$p_{i+1} = p_i + \Delta p_i$$

$$p_{i+1} = \begin{bmatrix} x_i \\ y_i \\ \theta_i \end{bmatrix} + \begin{bmatrix} \Delta s \cos(\theta_i) \\ \Delta s \sin(\theta_i) \\ \Delta \theta_i \end{bmatrix} \quad (4)$$

where

Δs = distance between sampling points

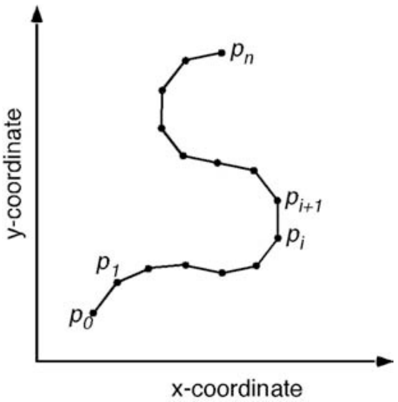

Fig. 3. The robot search path is a sequence of sampling points.

Therefore, executing the full path (Fig. 3)

$$p_n = p_0 + \sum_{i=1}^{n} \Delta p_i \quad (5)$$

If the robot encounters an isolated obstacle while performing the search path then the search will be disrupted by the least amount if the robot could detour around the obstacle and continue the desired search path on the far side of the obstacle.

Consider a detour path around an obstacle blocking the robot's search path (Fig. 4).

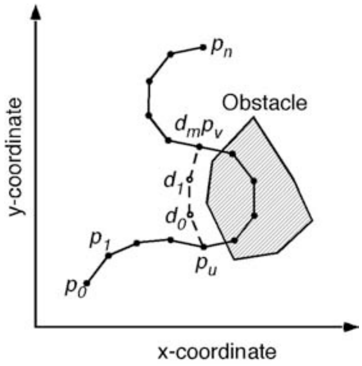

Fig. 4. Detouring round an obstacle while searching.

If such a detour path can be found then because the search path blocked by the obstacle starts and finishes at the same points as the detour path then:

$$\sum_{i=u}^{v} \Delta p_i = \sum_{j=0}^{m} \Delta d_j \quad (6)$$

For the obstacle avoidance process a number of pieces of information are required:

1) having detected an obstacle blocking the search path, which is the next search path point that is clear of the obstacle (p_v)?

2) in order to plan a collision-free path, what is the shape of the obstacle between p_u and p_v?

In keeping with the theme of biological inspiration it would be appropriate if the algorithm required the minimum of processor resources to implement.

B. Obstacle avoidance path

The obstacle avoidance algorithm is based on taking avoiding actions where necessary and also continuing to take the steps required by the search path where possible. These two sets of actions involve movements in approximately orthogonal directions. They were intermixed and used as required to track around the obstacle.

$$\sum_{j=0}^{m} \Delta d_j = \sum_{k=0}^{l} \Delta a_k + \sum_{i=u}^{v} \Delta p_i \quad (7)$$

It is arranged that eventually the avoiding actions taken together amount to the completion of a closed path:

$$\sum_{k=0}^{l} \Delta a_k = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (8)$$

At this point the robot rejoins the search path at the far side of the obstacle. The algorithm to generate the 'avoid' components a_k of the obstacle avoidance process can be complicated or as simple as feasible to negotiate the obstacles found in the robot's environment. For relatively small, convex obstacles it is sufficient to choose avoidance actions perpendicular to the trajectory of the robot when it first encountered the obstacle.

C. Avoid/Unavoid movements

There are a number of possible choices of obstacle avoidance actions that the robot can take. Bear in mind the requirement is that eventually the combined result of all avoid actions should be a closed path. Possibly the simplest path would be generated as follows.

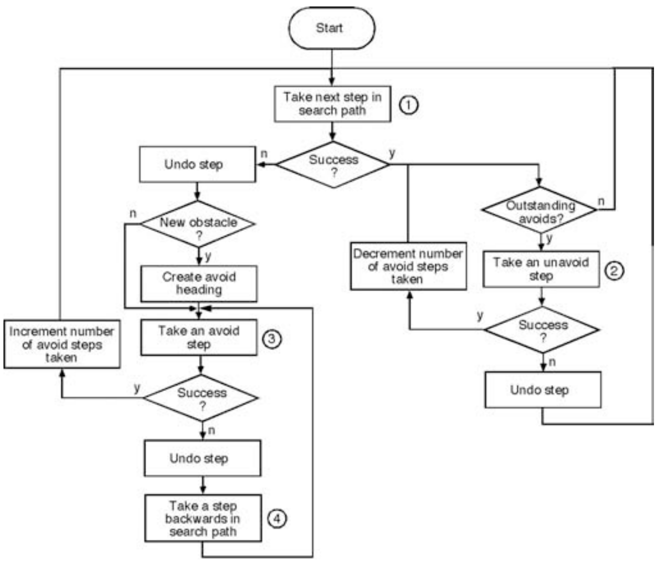

Fig. 5. Flow diagram of the search process including obstacle avoidance. The circled numbers indicate the four phases of the search process.

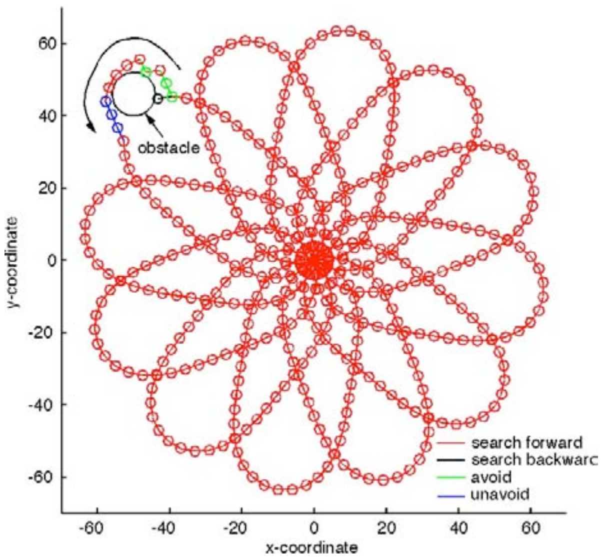

Fig. 6. Simulation of a radial search in which the robot avoids an obstacle drawn as a black circle.

Simple obstacle avoidance policy:

On the first collision an avoid direction is chosen with a heading 90˚ to the right of the current heading. Every avoid movement is made at this absolute heading and the unavoid movements in the opposite direction (90˚ to the left of the original heading). Thus, a number of avoid movements followed by an equal number of unavoid movements would bring the robot back to its starting position if they were the only movements that the robot was making (Equation 8). However, the full avoid process involves a mixture of avoid/unavoid movements interspersed with movements specified by the search path. Applying the avoid process with this choice of avoid and unavoid movements causes the robot to work its way around obstacles in an anti-clockwise direction.

B. Improved Obstacle Avoidance

The simple avoidance policy works well in regions where the search path does not change direction very quickly. However, if the direction of the search path changes significantly (> 45˚) over the region of the obstacle then the detour may not track round the obstacle correctly and leave significant shadow areas that have not been scanned adjacent to the obstacle. This effect is demonstrated in Fig.7.

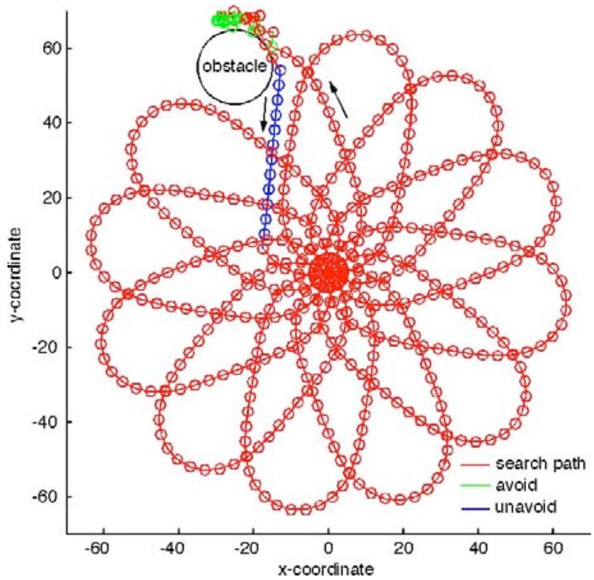

Fig. 7. Simulation of a spiral search showing a 'shadow' in the search coverage to the left of the obstacle.

One of the causes of this problem is when the search path orientation becomes aligned with the avoid heading. In order to overcome this problem the avoid policy was augmented so that when the search path and avoid headings were close to alignment (within ±45˚ of each other) then a new avoid heading was introduced at right angles to the original heading. This new avoid heading was accounted for separately and was compensated by appropriate unavoid actions after the obstacle was cleared. The result of using the modified policy is shown in Fig. 8.

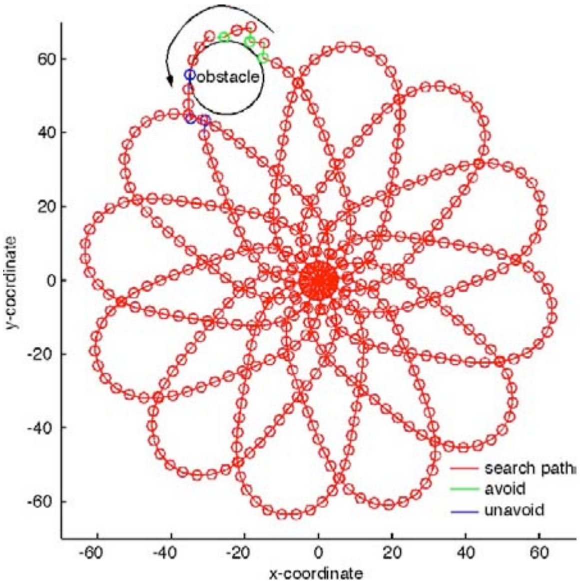

Fig. 8. Simulation of a spiral search employing the modified policy.

IV. Design Features of TASTI 2

The original licking robot TASTI was designed with the purpose of following trails of non-volatile chemicals deposited on the ground. Design features of this robot and particularly the tongue sensor are described elsewhere [9].

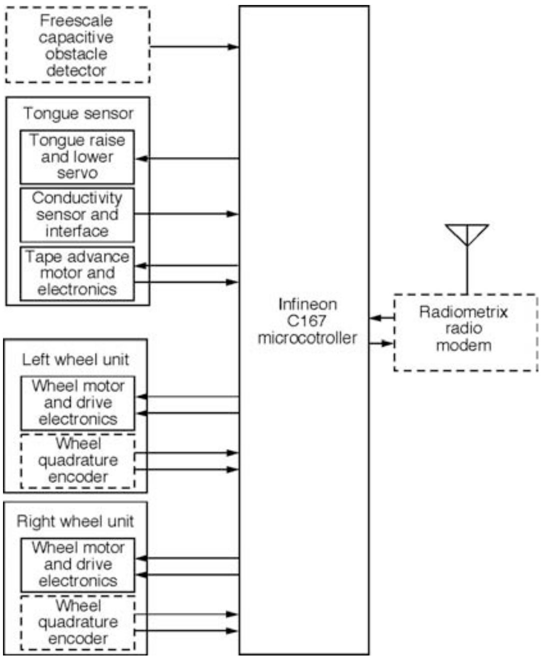

Fig. 9. Major components of the TASTI 2 robot. Items in dashed rectangles are enhancements added for the search experiments and not present in the original TASTI robot.

Attempts to use TASTI to implement search algorithms highlighted a number of areas for improvement and as a result TASTI 2 was designed (Fig. 9).

Wheel encoder resolution was improved by increasing the number of light to dark transitions for one revolution of the wheel and changing from a single phase to a two phase quadrature encoder.

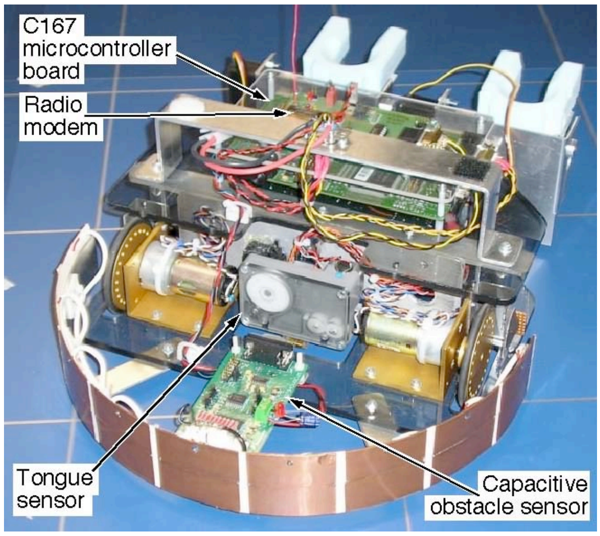

Fig. 10. The TASTI 2 gustatory robot.

Search paths are performed using dead reckoning and collisions introduce significant inaccuracy. In order to detect obstacles before a collision occurred a capacitive sensor ring was attached to the front of the robot. The capacitive sensor is based on the Freescale MC33794 electric field imaging device. This device can detect the change in capacitance as either a conductive or dielectric object approaches one of nine sensor electrodes. In TASTI 2 the nine electrodes are arranged in a 180˚ arc covering the front of the robot. When approaching a steel can the robot detects the obstacle and comes to a halt 1.5cm away from the can. The radius of the capacitive sensor is 153mm and each sensor electrode senses over a 20˚ region. In this application all of the sensor electrodes were connected in parallel to give a single proximity signal.

When performing complex search patterns the umbilical power/communications cable of TASTI became tangled. For this reason cable-free operation was designed into TASTI 2 by transferring to battery power and using a Radiometrix modem for communications.

The final improvement was to change the geometry of the robot. In TASTI the tongue sensor was mounted on the edge of the cylindrical robot 175mm from the turning centre of the robot. This made it difficult to follow a calculated search path. To overcome this problem the tongue sensor was relocated to the turning centre of the robot as shown in Fig. 10.

V. Experimental Results

In order to test the obstacle avoidance algorithm experiments were performed using TASTI 2. In the results presented here an Archimedes spiral search was performed searching for a circular patch of salt crystals 350mm in diameter. The spiral search was chosen rather than the radial search described in Section II to reduce the floor space required for the experiments. The step size of the robot search path was 141mm. For comparison initial experiments were performed without any obstacles. Figs. 11 and 12 show an example of the path of the robot and chemical sensor readings. Note that the whole of the search path consists of phase 1 movements only (take next step in the search path).

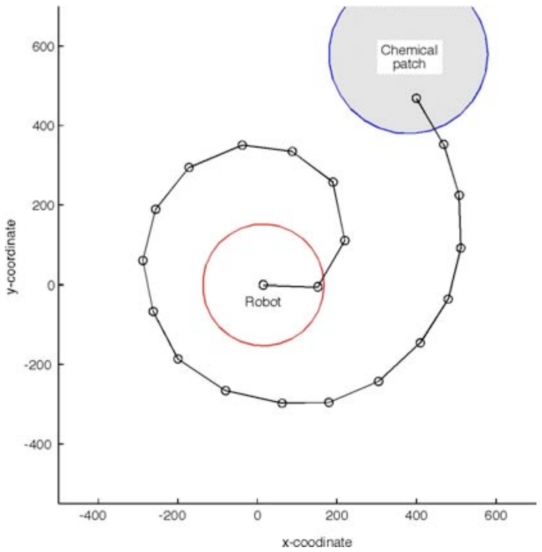

Fig. 11. TASTI 2 performs a spiral path searching for a patch of chemical 350mm diameter. Black circles indicate the location of sensing operations.

As shown in Fig. 13 TASTI 2 is successful in avoiding an obstacle in its path and continuing its search to find the chemical patch. The data displayed in Fig. 14 shows that the search algorithm moves to phases 3 and 4 during obstacle avoidance. Note that while negotiating round an obstacle the robot sometimes samples the chemical concentration at one point more than once if it makes a move that is blocked and must return to a previous point. Odometry is quite accurate until the robot encounters the obstacle.

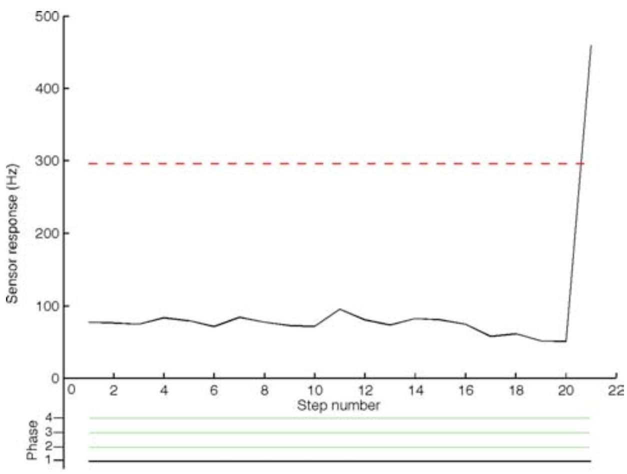

Fig. 12. Chemical sensor response and search algorithm phase recorded during the spiral search. The search terminates when the sensor reading exceeds a threshold of 296Hz.

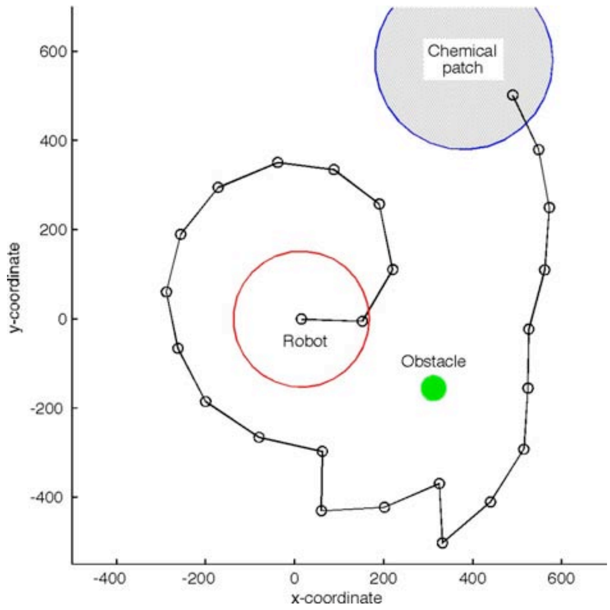

Fig. 13. Spiral search path avoids an obstacle.

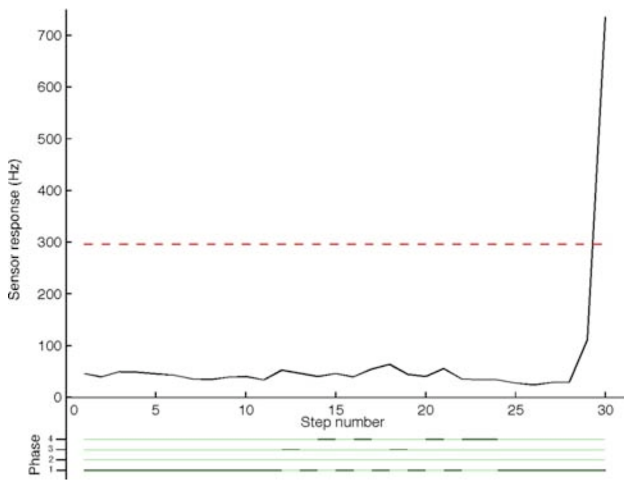

Fig. 14. Data gathered during a spiral search that includes avoiding an obstacle located in the search path.

For the two paths shown in Figs. 11 and 13 the discrepancy between the two paths just before the obstacle (step 13) was less than 10mm. However, after performing obstacle avoidance and arriving at the chemical patch the error had risen to 100mm. Obstacle avoidance involves many additional turns and straight line movements for the robot (in this case 30 as opposed to 21) coupled with the larger angles turned through both increase the accumulated odometry errors.

As mentioned in Section IV the robot TASTI 2 incorporated some improvements to reduce odometry errors. However, there is still room for improvement and the literature on robot odometry including [10] will be consulted to improve this aspect of TASTI 2.

VI. Conclusion

An obstacle avoidance scheme has been developed that allows a robot to circumnavigate obstacles while following a search path. The obstacle avoidance algorithm requires little computational resources on the searching robot and would be applicable for simple robots. Both simulation and practical experiments have confirmed that the algorithm functions as expected.

Odometry errors are inevitable and always accumulate as a robot moves and turns. If a robot's search area extended beyond the range where odometry could be relied upon then some external means of localization would be required. One potential solution for the searching task is for the robot to drop off one or more beacons at the start of the search and to use this/them to correct subsequent odometry errors. This is a possibility that will be investigated further in this project.

Acknowledgment

The author thanks Jaurie Wijaya for his excellent work constructing TASTI 2.

References

[1] A.H. Purnamadjaja, and R.A. Russell, "Guiding robots' behaviours using pheromone communication", *in Autonomous Robots*, vol. 23, 2007, pp. 113-130.

[2] K.T. Clifford, L. Gross, K. Johnson, K.J. Martin, N. Shaheen, and M.A. Harrington, "Slime-trail tracking in the predatory snail, *Euglandina rosea*", Behavioral Neuroscience, 117(5), pp. 1086-1095, 2003.

[3] L.M. Klauber, *Rattlesnakes: Their Habits, Life Histories and Influence on Mankind*, University of California Press, 1972.

[4] V.G. Dethier, *To Know a Fly*, Holden-Day Inc. San Francisco, 1962.

[5] F.A. Geldard, *The Human Senses, 2nd. Edn.*, John Wiley & Sons, N.Y., 1972.

[6] B. Hölldobler, and E.O. Wilson, *The Ants*, Harvard University Press, Cambridge, Mass., 1990.

[7] D.B. Dusenbery, *Sensory Ecology*, W.H. Freeman and Co., N.Y., 1992.

[8] R.A. Brooks, "A robust layered control system for a mobile robot", IEEE Journal of Robotics and Automation, RA-2 (1), 1986, pp. 14-23.

[9] R.A. Russell, "TASTI follows chemical trails with its robot tongue", *Proc. IEEE International Conference on Robotics and Biomimetics*, Kunming, China, 2006, pp. 257-262.

[10] J. Borenstein, and L. Feng, "Measurements and correction of systematic odometry errors in mobile robots", *IEEE Transactions on Robotics and Automation*, 12 (6), December 1996, pp. 869-880.