

Bio-inspired Plume Tracking Algorithm for UAVS

G. Montes, B. Letheren, T. Villa and F. Gonzalez

Queensland University of Technology, Brisbane, Australia, QLD, 4001

glen.montes@mail.escuelaing.edu.co, ben.letheren@student.qut.edu.au, tf.villa@hdr.qut.edu.au,
felipe.gonzalez@qut.edu.au

Abstract

There is an increased interest on the use of UAVs for environmental research and to track bush fire plumes, volcanic plumes or pollutant sources. The aim of this paper is to describe the theory and results of a bio-inspired plume tracking algorithm. A memory based and gradient based approach, were developed and compared. A method for generating sparse plumes was also developed. Results indicate the ability of the algorithms to track plumes in 2D and 3D.

Keywords: Bio-inspired, plume tracking, gas sensing UAVs, air quality.

1 Introduction

Bees such as the Magachile Rotundata, more commonly known as the Alfalfa Leafcutter Bee, are able to find food by tracking the concentration of particles in the air [Theobald., 2007]. Previous researchers on plume tracking have used bio-inspired methods for ground and underwater vehicles [Farrell, 2005; Porter, 2012]. Farrell for instance developed and advanced 2-D robotic implementation of an odour-based navigation system. The algorithms were inspired by moths and Antarctic procellarii (seabirds) behaviours. The male Tobacco Hornworm moth (*Manduca Sexta*) was studied by Porter [Porter, 2012]. The author discussed previous attempts before creating a plume tracking algorithm and running virtual simulations. Ishida *et al* studied the flight path of a moth also for their plume tracking algorithm [Ishida et al., 2001] and implemented it into a ground based wheeled robot [Ishida et al., 2001]. Their robot was designed to be downwind to the plume and would gather information on the distribution along with other data while tracking upwind. Chen and Moore [Chen and Moore, 2004] illustrated the usefulness of having multiple UAVs to track chemical plumes. Smídl and Hofmans [Smídl and Hofman, 2013] found that two UAVs with their plume tracking algorithms returned similar results to thirty ground based gas sensors. Neumann *et al.* proposed a pseudo-gradient-based plume tracking algorithm and a particle filter-based source declaration approach, and tested it on a gas-sensitive micro-drone. The authors compared the performance of their system in simulations and real-world experiments. The authors also suggested

the use of a zig-zag algorithm and referred to the difficulties on locating gas sources in scenarios with high turbulence and changing wind conditions.

Pang *et al* developed a plume tracking algorithm that was implemented and tested on an Autonomous Underwater Vehicle [Pang., 2007]. The algorithm was tested in the open ocean where it successfully tracked underwater chemical plumes. Smídl and Hofman expanded their plume tracking and implementation by using both ground sensors and multiple UAVs [Smídl and Hofman, 2013]. The authors also tried varying combinations and found that the UAVs were a useful addition to the ground station units especially as the UAVs do not require weather forecast information for tracking. This paper extends their work by complementing the memory based method, developing a gradient based approach and testing it for several plume modelling scenarios in 2D and 3D.

The rest of the paper is organised as follows; Section 2 describes the tracking algorithms, Section 3 discusses hardware in the loop simulation results and Section 4 concludes the paper and discusses future research.

2 Plume Tracking Algorithms

In this work the Moth flight profile was used as inspiration for developing the algorithms. Figure 1 shows a typical behaviour of the male Tobacco Hornworm moth (*Manduca Sexta*) which was studied by Porter [Porter, 2012]. The moth flight profile is divided 3 main tasks. The Casting behaviour is where the moth searches horizontally for the plume. The moth flies perpendicular to the wind direction in order to increase the chances of detection. Once it has detected the pheromone plume, the Counterturning behaviour begins, tracking into the plume whilst increasing its speed. The surging behaviour continues tracking narrower.

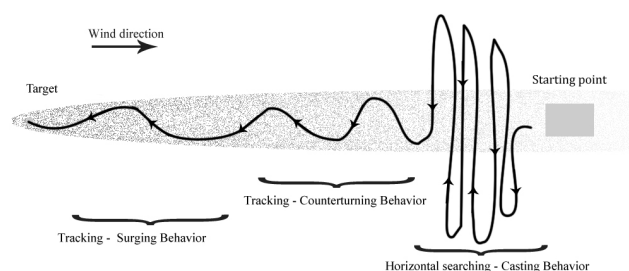


Figure 1. Example of moth behaviour tracking a pheromone plume.

Two methods were developed and tested using a bio-inspired algorithm using the moth's behaviour to track; a memory based method and a gradient based method.

2.1 Memory Based Algorithm (MA)

2.1.1 Overview

A memory based algorithm was developed. This algorithm is describes in Figure 2.

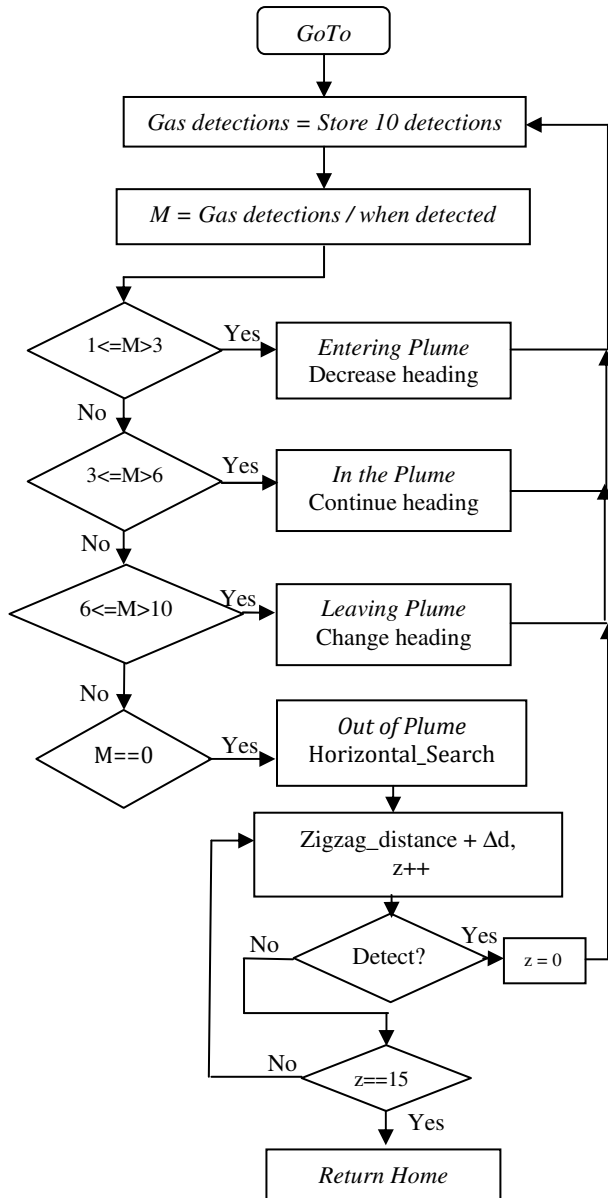


Figure 2. Memory based algorithm flow chart.

The memory based algorithm starts by reading in a predefined number of gas sensor values and averaging the number of detections with when they occurred. A memory size of ten readings was selected for the simulations to allow the UAV to be more responsive. Once this averaged value is calculated, the algorithm assesses the state of the UAV. If this value is equal to zero (occurs when no detections have been made), the UAV is considered to be out of the plume and thus the search function is activated which performs a zigzag pattern which in turn increases

until a detection is made or until the plume is considered lost returning back to launch. As the UAV enters the plume (value between 1 and 3), it decreases its heading slightly as it is therefore assumed that it has entered the side of the plume by the previous search function which uses large heading values (zigzag patterns). A value that is equal or greater than 3 but less than 6 results in the aircraft being seen as flying inside the plume and will maintain its heading. Finally with a value between 6 and 10, the UAV will be considered to be leaving the plume. This will happen when it previously maintained its heading and no detections have been made afterwards which will cause a change of heading, trying to enter the plume again. These values had been researched prior [Porter, 2012] however, to have the UAV to be more responsive to when it can make a heading change and also when it is considered to be leaving the plume, the number of samples were decreased to 10 and the range for when the UAV was leaving the plume was expanded to with the range for when the UAV is in the plume was decreased by one. Testing these changes showed promising results with these values also being used for the algorithms discussed latter. It should be noted that these values will be confirmed again during the flight testing phase. An example of the UAV's trajectory when the 'Leaving Plume' range was between seven and 10 is included below in Figure 3, it can be seen that the UAV did detect that it had left the plume (upward trend at the bottom left corner) but it had travel too far out to be able to recover the plume again before the simulation finished.

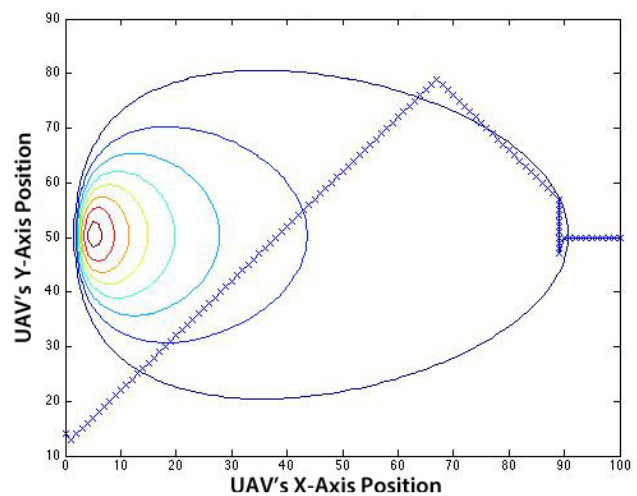


Figure 3. Memory based algorithm simulation with unrefined 'Leaving Plume' range.

2.1.2 Test Case 1

Figure 3 shows the simulation of the memory based algorithm using a Gaussian plume model created by [Holzbecher, 2007] where each 'x' represents a gas sensor reading, this is true for all of the simulations ran using MATLAB. Although the figures axis's are not in meters, a visualisation of the distance travelled can be achieved by considering the distances between the x's is the time between sensor reading which equates to approximately 60 meters (a 5 second sampling time and a speed of 12m/s). It can be seen that a similar behaviour to that of the Moth represented in Figure 1 can be seen in Figure 4. In addition, the heading angles allow the UAV to make a

tighter turn and ensure that it reaches the source of the plume.

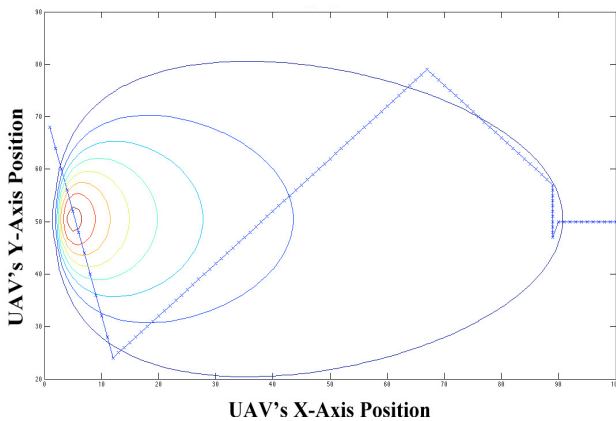


Figure 4. Memory based algorithm simulation with refined heading adjustments.

The aircraft is able to successfully navigate and fly through the centre of the source of the plume. The initial flight segment is the aircraft taking its ten memory readings before calculating that it is not in the plume and taking a heading change by which time it is in the plume and thus flies through the plume to take another set of data values and making a heading decision which, was that it was flying in the plume.

Once the aircraft leaves the plume, it is still gathering data; however the decision for a new direction is not taken before the end of the sampling procedure. At this stage, with the new information collected by the sensor, the UAV can recognize its position outside of the plume, and so heads in a new direction. During its memory gathering, the aircraft enters the plume, and continues on its current trajectory until it eventually leaves the plume again (visible near the bottom left corner). At this final heading change, the aircraft, having flown down and left the plume, decides to fly upwards reacquiring the plume.

2.2 Gradient Based Algorithm (GrA₁)

2.2.1 Overview

A gradient based algorithm was created and compared with the memory based algorithm. The algorithm starts by reading in the gas sensor value and compares it to the previous reading. If the current reading is less than the previous reading, the UAV is considered to be on a heading that is leaving the plume, which results in a heading change being required. If the current reading is higher than the previous reading, the UAV is considered to be flying towards the source of the plume and this heading is maintained. The last option is actioned if the gas sensor reading is below the threshold, the UAV is considered to be out of the plume and the search function is activated. Figure 5 shows the flow chart for the gradient based algorithm.

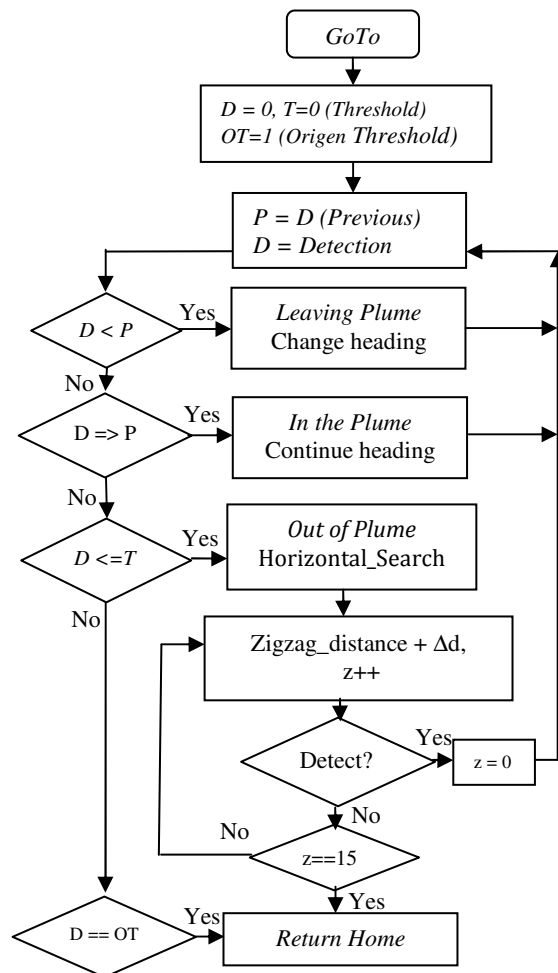


Figure 5. Task flow chart algorithm GrA₂.

The difference between the gradient algorithm and the memory based algorithm is that the memory based algorithm relies on the average value to decide whether it is leaving, entering or continuing in the plume, whereas the gradient based algorithm uses the difference between the previous and the current gas sensor value to make this decision. With this, the aircraft can react faster and will travel smaller distances between actions but means that it is also more exposed to acting on gaps in the plume and treating them as leaving the plume whereas the memory based algorithm will most likely ignore this (unless the plume is of a large size).

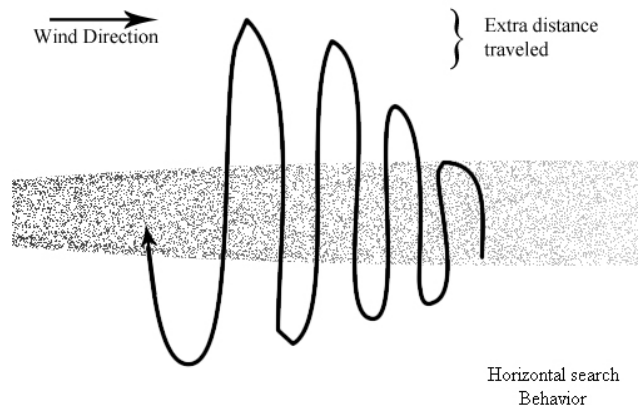


Figure 6. Illustration of the Horizontal Search routine.

2.2.2 Test Case 2

The gradient algorithm was tested for the same conditions as the memory based algorithm. The gradient algorithm was able to make decisions much faster, however it also travelled smaller distances between decisions and thus was more likely to enter the horizontal search function which is time consuming. This is visible in Figure 7 which relates to Moth's behaviour.

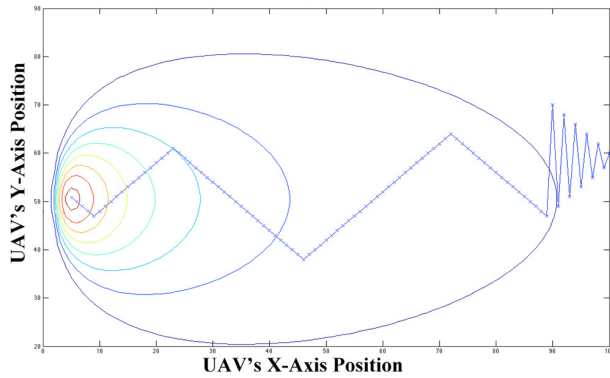


Figure 7. Gradient algorithm simulation.

The first Section of Figure 7 is where the aircraft starts its flight and the search for the plume. As the aircraft is out of the plume and had yet to make a detection, it calculated that it is out of the plume and activates the horizontal search function. This function runs until a detection is made or until a set number of 15 waypoints is reached. In this case, the plume was detected within the 10th waypoint and thus the algorithm detects that it has entered the plume. From here on, the algorithm reads the gas sensor value at each waypoint and calculates whether the gas concentration is increasing or decreasing. This is represented in Figure 7 by the long zigzag pattern through the plume where each peak, the aircraft has found the gas value is lower than the previous waypoint and changed its heading. Near the end of the plume, the UAV is close to the source and starts to track in until it reaches the approximate source of the plume.

2.3 Comparison Between The Two Algorithms

When comparing Figures 4 and 7, it can be seen that both algorithms were able to locate the approximate location of the plume and to act on their own to calculate the new heading. The memory based algorithm (Figure 4) compares the beginning of each simulation and maintains the aircraft on the same heading until it finishes a sampling period. On the other hand the gradient algorithm (Figure 7) reacts as soon as it detects that the UAV position outside the plume and so it therefore triggers the horizontal search (Zigzag pattern). As Figure 4 shows the UAV travels a greater distance between heading changes, and is able to reach the plume on the first waypoint generated by the horizontal search whereas the gradient algorithm (Figure 7) spends more time in the horizontal search before it locates the plume. From this point onwards, the memory algorithm travels further across the

plume and leaves it twice. The gradient algorithm, on the other hand, as it does not need to leave the plume to adjust its heading, is able to travel a more direct path towards the plume and does not leave the plume once it enters and locates it.

These algorithms need to be considered in a real world situation. Both algorithms rely on the UAV's location and the gas sensor reading to calculate a new waypoint. The memory algorithm travels a longer distance between heading changes which means that if the UAV is in the plume but goes through a small part of the plume where there is no gas, it would be less likely to be affected and instead would continue its current path. On the other hand the gradient based algorithm would most likely accept this gap as leaving the plume and make a heading change. This results in the first algorithm seeming more stable; however, it has to gather multiple data readings before it can make a decision. If the gas sensor unit has a long delay in sampling time the aircraft would travel large distances and may leave the plume entirely before doing a heading change.

The gas sensor considered for this work [Malaver et al., 2012] has a delay of less than five seconds between readings; however, just taking ten readings would result in up to fifty seconds having passed and at a speed of approximately twelve meters per second the UAV would have travelled over half a kilometre. The second algorithm needs to take only one reading, which, in the same circumstances, means it can make a decision within approximately 60 meters. Because of the results are represented in a normalized manner (i.e size of the plume from 0 to 100 or concentration values from 0 to 1) calculations are not directly reflected in the Figures; comparisons can be done using this standardization.

2.4 Test Case 3

The situation where the UAV flies through the plume and enters a gap in the plume is of special interest. The gradient algorithm was tested with the same plume model as before but with a hole inserted in the middle of the plume, visible as the square in the middle of Figure 8.

The UAV is still capable of reaching the approximate location of the plume even though it flies through the hole in the plume. The flight path begins as in Figure 7 until the hole is found. When this happens, the algorithm calculates that the concentration has gone below the chosen threshold and thus it begins the search function which is represented by the expanding zigzag pattern. This continues until the aircraft has left the hole and continues the tracking function.

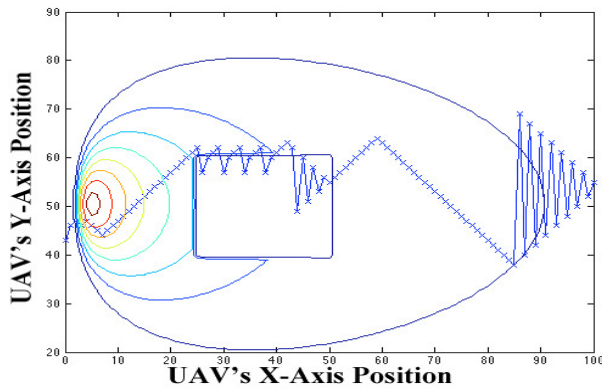


Figure 8. Gradient algorithm simulation with a gap of concentration in the plume.

The UAV continues to fly down which results in it re-entering the hole but as it has not flown far inside the hole, it is able to leave it on the first waypoint generated by the search function. This pattern continues until the UAV has flown past the hole and is able to continue its path towards the source of the plume.

2.5 Modified Holzbecher Model and improvements to the gradient algorithms (GrA₂)

Although Gaussian distribution is common in plume modelling it does not accurately reflect real world situations. In order to create a more realistic plume and simulation environment, the Holzbecher model algorithm [Holzbecher, 2007] was modified by adding the property to change the wind speed to create in turn gaps with zero concentration proportionally with how far from the origin they were. In addition the improvements also generated the capability to generate a random plume every time the algorithm was executed. With these new conditions, the algorithm described in Section 2.2 was tested. The results can be seen in the Figure 9.

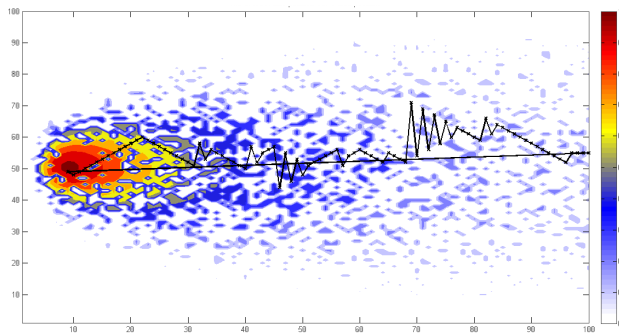


Figure 9. Dispersed plume and 2D flight path using gradient algorithm GrA₁.

Although the gradient based algorithm is able to find its target, the changes in the plume bring new challenges to the gradient algorithm. As previously expected in Section 2.2 each gap generates frequent heading changes. Therefore modifications to the gradient algorithm were necessary which was not ideal for the UAV in order to avoid this situation. To make the gradient algorithm immune to gaps in the plume, a buffer which will store how many times the algorithm was able to find consecutive ‘no detections’ was implemented. This buffer is a temporary variable that controls when the next heading is allowed to change. For instance a buffer equal

to 5 would make the UAV maintain its heading until 5 consecutive ‘no detections’ have been made by the sensor.

Pseudo-code for GrA₂

```

if Sensor_detection==0
    NoDetection=NoDetection+1;
    Actual_Heading=Last_Heading;
    if NoDetection==5
        goto Horizontal_Search;
    end
end
end
    
```

Figure 10. Pseudo code for GrA₂

After having implemented this addition to the gradient algorithm, the simulation was executed again using the same plume as Figure 9 in order to compare the results. A buffer equal to 8 was used in this application.

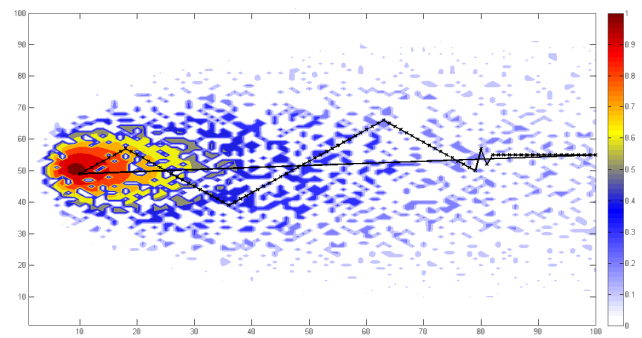


Figure 11. Dispersed plume and 2D flight path using gradient algorithm GrA₂ with a buffer of 8.

The flight path in Figure 11 shows a notable improvement in the algorithm when compared to Figure 9, adjusting to perturbations due to gaps in the plume.

2.6 Modified Holzbecher Model in 3D with Vertical Search algorithm (GrA₃)

Until now the gradient algorithm has been focused on 2D tracking. The gradient based algorithm and Holzbecher model [Holzbecher, 2007] were modified in order for the UAV to do 3D tracking and adding an extra dimension without losing the plume property of concentration versus distance. The results of a modified Holzbecher model can be seen in Figure 12.

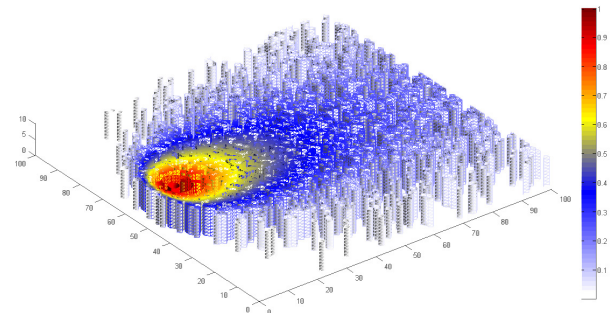


Figure 12. Modified Holzbecher model in 3D.

The gradient algorithm was modified to allow searching at different altitudes with these 3D modifications on the plume. Undoubtedly plumes have different characteristics

depending of what they are made of. The algorithm works for different types of plumes, however for practical reasons, the algorithm assumes that the plume tends to rise at the beginning and fall down when the concentration is low. For this reason, when the algorithm searches horizontally with no results, it will decrease the altitude with the purpose of keeping tracking active. Figure 13 shows the pseudo code of GrA₃.

Pseudo-code for Vertical search

```

Vertical_Search==1;
Fly_to>Last_Detection)
While Vertical_Search==1 and loiter_count<=3
    Decrease_altitude();
    Fly_around(radius);
    if (Detection)
        Vertical_Search=0;
    else
        radius=radius*2;
        loiter_count++;
end
end
    
```

Figure 13. Pseudo-code vertical search GrA₃.

The vertical search is activated after the tracking, searching horizontally and backtracking do not result in a detection. When the vertical search is called, the algorithm asks the UAV to return where the last detection was made. As soon as the UAV is near to these coordinates, it starts to fly around this point with a certain radius whilst also decreasing its altitude. The radius is increased 3 times before the vertical search stops. If no detections are made the UAV will return home.

Figure 14 describes the GrA₃ algorithm in a flow chart. The blue colour represents the Tracking function in which the heading is adjusted depending of the current detection. The brown colour represents the Horizontal Search function which increases the ground covered (in a zigzag pattern) while operating. The orange colour represents Backtracking, which forces the UAV to return to where the last detection was made in order to relocate the plume. Finally the gray colour represents the vertical searching function which decreases the altitude as well as flying around the last detection, increasing the search radius.

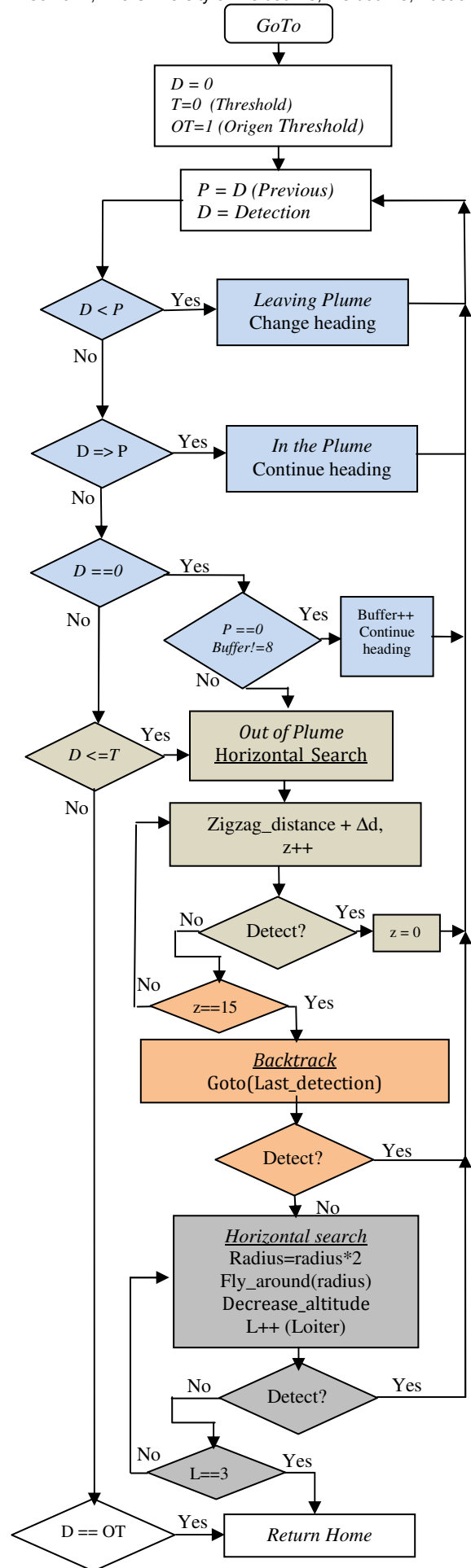


Figure 14. Flow chart algorithm GrA₃.

2.6.1 Test Case 4

Figures 15-17 show an example of the execution of the vertical search task. It is indicated with the symbol \otimes .

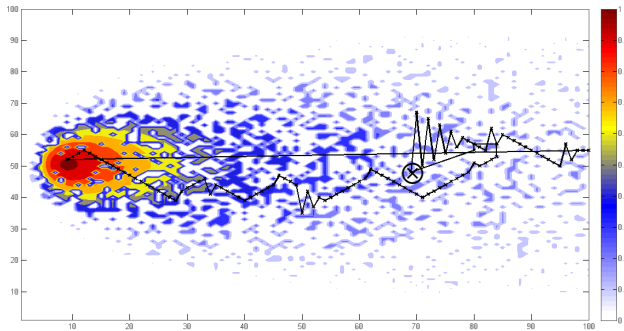


Figure 15. 3D algorithm implemented.

It is possible to identify each task of the algorithm in Figures 15 and 16 as well as the Moth's behaviour described in Section 2:

- Tracking: Straight lines with little heading changes
- Horizontal Searching: Zigzag pattern.
- Backtracking: Return to the last detection
- Vertical Searching: Flying around one point and decrease the altitude while increment the search radius by 2.

In Figure 17, it is shown how the algorithm decreases altitude at the same location that it is returning to the last detection (X-axis 70).

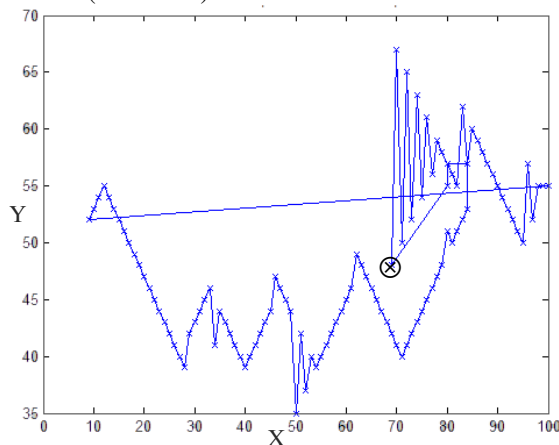


Figure 16. Projection of 3D path onto X-Y axes.

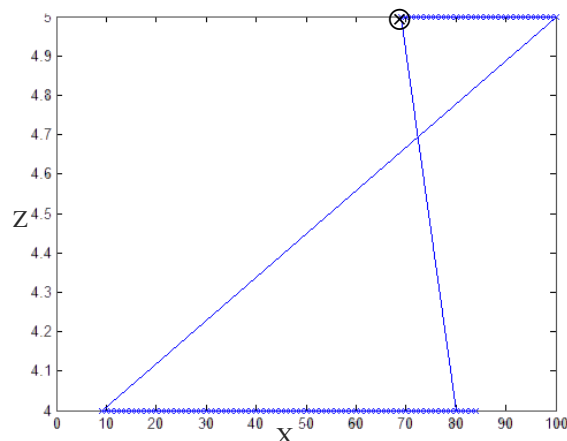


Figure 17. Projection of 3D onto X-Z axes.

3. Hardware In The Loop Implementation

Hardware in the loop testing is the last stage in testing the algorithm before flight testing. Hardware in the loop works by integrating an autopilot module with a ground control station (GCS). The GCS runs the plume-tracking algorithm and interacts with the autopilot and the flight simulator. The autopilot, thanks to the flight simulator, acts as if it is in a flying aircraft and responds to the commands from the tracking algorithm sent through the GCS. This allows a visual representation, which aids in debugging and refinement of the algorithm before physical flight-testing.

During the hardware in-the-loop simulations the initial and previous gas sensor values were set to ensure the algorithm could test cases described. The UAV's starting heading and required heading change is displayed to the user. The initial heading change is set to zero degrees as the algorithm must read the gas sensor value before making a decision. After the gas sensor value has been read, the algorithm compares this value to the previous value using the method outlined in Section 2.2. This is illustrated in Figure 18 where the algorithm selects the out of the plume function (0,1,0,0) which in turn decides on a new heading of 230 degrees. After this, the algorithm re-evaluates the situation and continues the pattern until the source of the plume is found.

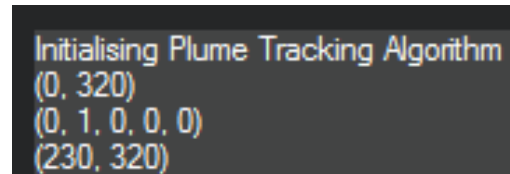


Figure 18. GCS showing the output of the plume tracking algorithm being activated.

A result of the hardware in loop simulations, Figure 19, shows the projection of the plume onto Google Earth and the path followed by the UAV for test case 4. The path followed is the same as the results obtained in Figures 15 and 16.

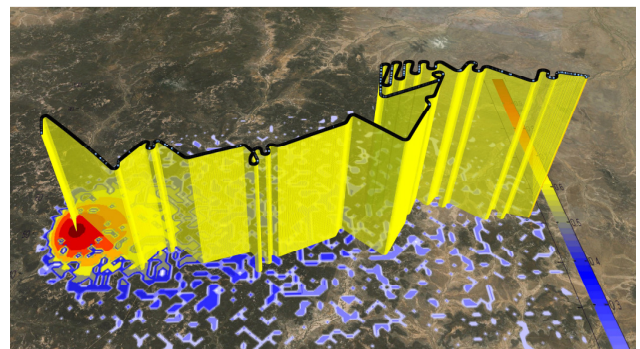


Figure 19. Flight path of algorithm GrA₃.

4. Conclusion and Future Work

This project expands on existing plume tracking research by designing the algorithm for real world environmental roles whilst using a bio-inspired algorithm. The plume tracking memory-based algorithm is able to locate the

source of the plume however its reaction time is too slow for this application. The Gradient based algorithm works well in 2D but needs modifications in 3D plumes which are an ongoing process.

Before creating a smoke plume in a controlled environment and performing real world testing the hardware in the loop simulation must be implemented with the configuration file of the UAV which will do the tracking in order to refine the flight time, maximum and minimum performance.

Further work might consider including multiple UAVs to track the plume. A possible scenario would have one of the UAVs flying upwind and one downwind in order to track the plume from different angles and so being able to predict the plume path. This would help to provide an early warning of likely areas to be affected.

New developments will explore the use of Genetics Algorithms to obtain faster reaction times. In fact these kinds of algorithms have proven being a valuable tool in terms of effectiveness and efficiency in identifying optimal parameters for different possible problems.

References

- [Chen and Moore, 2004]. Kevin L. Moore and Yang Quan Chen. *Model-Based Approach To Characterization Of Diffusion Processes Via Distributed Control Of Actuated Sensor Networks*. 1st IFAC Symposium on Telematics Applications In Automation and Robotics, Helsinki University of Technology Espoo, Finland, 2004.
- [Farrell, 2005] J. A. Farrell. *Chemical plume tracing via an autonomous underwater vehicle*. IEEE Journal of Ocean Engineering, 30(32):428—442, 2005.
- [Holzbecher, 2007] Ekkehard Holzbecher. *Environmental Modeling*. Springer Science+Business Media, 2007.
- [Ishida., 2005] Hiroshi Ishida, Takamichi Nakamoto and Moriizumi. *Controlling a Gas/Odor Plume-Tracking Robot Based on Transient Responses of Gas Sensors*. IEEE Sensors Journal, 5(3), 2005.
- [Ishida., 2001] Hiroshi Ishida, Nakamoto T, Toyosaka Moriizumi, Timo Kikas, and Janata J. *Plume-Tracking Robots: A New Application of Chemical Sensors*, 2001. The biological bulletin 2001 Apr;200(2):222-6.
- [Malaver., 2012] A. Malaver, F. Gonzalez, N. Motta, A. Depari and P. Corke. *Towards the Development of a Gas Sensor System for Monitoring Pollutant Gases in the Low Troposphere Using Small Unmanned Aerial Vehicles*. Workshop on Robotics for Environmental Monitoring, 11 July 2012, Sydney University, N. S. W.
- [Marques, 2002] L. Marques. *Olfaction-based mobile robot navigation*. Thin Solid Films, 418:451-458, 2002.
- [Neumann, 2013] P. Neumann, V. Bennets, A. Lilenthal, M. Bartholmai, J. Schiller. *Gas source localization with a micro-drone using bio-inspired and particle filter-based algorithms*. Advanced Robotics, Vol. 27, Issue 9, page 725-738, 1 June 2013.
- [Osório, 2010] Luís Osório, Gonçalo Cabrita, and Lino Marques. *Mobile Robot Odor Plume Tracking Using Three Dimensional Information*, ECMR, page 165-170. Learning Systems Lab, AASS, Örebro University, 2011.
- [Porter, 2012] Porter, M. J. *Bio-Inspired Odor-Based Navigation*, BiblioScholar, 2012.
- [Pang, 2007] Shuo Pang; Farrell, Jay A.; Wei Li. *AUV-Based Chemical Plume Tracing*, Sea Technology, Vol. 48 Issue 7, p43, 2007.
- [Smídl and Hofman, 2013] Václav Smídl and Radek Hofman. *Tracking of atmosphere release of pollution using unmanned aerial vehicles*. Atmospheric Environment, Elsevier, 2013.
- [Theobald., 2007] Jamie Carroll Theobald, William T. Wcislo and Eric J. Warrant. *Flight performance in night-flying sweat bees suffers at low light levels*. The Journal of Experimental Biology, 210(Pt 22):4034-42, 2007.
- [Young, 2010] George S. Young. *UAV Navigation by Expert System for Contaminant Mapping*. Expert Systems with Applications, Volume 37, Issue 6, Pages 4687-4697, Elsevier, 2010.