Hazardous Chemical Source Localisation in Indoor Environments Using Plume-tracing Methods

Zeqi Li

School of mechanical engineering

The University of Adelaide

South Australia 5005

Australia

A thesis submitted for fulfilment of the requirements

for the degree of Master of Philosophy in

Mechanical Engineering

on the 31th August 2020

Declaration

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in my name in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission in my name for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint award of this degree.

The author acknowledges that copyright of published works contained within this thesis resides with the copyright holder(s) of those works.

I give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library Search and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

Zeqi Li

Signed:_____

Date:_____

Acknowledgement

This thesis is finalised with the help and guidance from many people, especially my supervisors in the University of Adelaide. Therefore, I would like to thank many people for supporting my research in many ways.

I would like to thank my principle supervisor, Dr. Zhao Tian for supporting me in academic field and my daily life. Zhao, who brings me to the world of research, is the guide for my research project throughout the whole process of my master's degree. Both his professional knowledge and personal care inspire me in scientific field and on personal level significantly. Thank you, Zhao!

I would like to thank my co-supervisor Dr. Tien-fu Lu for supporting me in the professional areas of robotics and plume-tracing algorithms. Even though I started this project without sufficient knowledge in plume tracing as well as simulation skills, Tien-fu helps me kindly and patiently. Thank you, Tien-fu!

I would like to thank my co-supervisor Dr. Houzhi Wang for supporting my research in different academic areas. Houzhi supports me with his professional knowledge and positive attitude. Thank you, Houzhi!

I would also like to give thanks to my parents for guiding me throughout my whole life. Thank you, Hongwei and Mingtao!

Many thanks to my research fellows and friends for supporting me and the academia who helped me through the school seminar, major review, as well as other core components and milestones during my candidature.

Sincerely, Zeqi.

Abstract

Bio-inspired chemical plume-tracing methods have been applied to mobile robots to detect chemical emissions in the form of plumes and localise the plume sources in various indoor environments. Nevertheless, it has been found from the literature that most of the research has focused on plume tracing in free-stream plumes, such as indoor plumes where the chemical sources are located away from walls. Moreover, most of the experimental and numerical studies regarding the assessment of indoor plume-tracing algorithms have been undertaken in laboratory-scale environments. Since fluid fields and chemical concentration distributions of plumes near walls can be different from those of free-stream plumes, understanding of the performance of existing plume-tracing algorithms in near-wall regions is needed. In addition, the performance of different plume-tracing algorithms in detecting and tracing wall plumes in large-scale indoor environments is still unclear. In this research, a simulation framework combining ANSYS/FLUENT, which is used for simulating fluid fields and chemical concentration distributions of the environment, and MATLAB, with which plume-tracing algorithms are coded, is applied.

In general, a plume-tracing algorithm can be divided into three stages: plume sensing, plume tracking and source localisation for analysis and discussion. In the first part of this research, an assessment of the performance of sixteen widely-used plume-tracing algorithms equipped with a *concentration-distance obstacle avoidance method*, was undertaken in two different scenarios. In one scenario, a single chemical source is located away from the walls in a wind-tunnel-like channel and in the other scenario, the chemical source is located near a wall. It is found that *normal casting*, *surge anemotaxis* and *constant stepsize* together

performed the best, when compared with all the other algorithms. Also, the performance of the *concentration-distance obstacle avoidance method* is unsatisfactory. By applying an *along-wall obstacle avoidance method*, an algorithm called *vallumtaxis*, has been proposed and proved to contribute to higher efficiencies for plume tracing especially when searching in wall plumes. The results and discussion of the first part are presented in Chapter 4 of this thesis.

In the second part, ten plume-tracing algorithms were tested and compared in four scenarios in a large-scale indoor environment: an underground warehouse. In these four scenarios, the sources are all on walls while their locations are different. The preliminary testing results of five algorithms show that for most failure cases, the robot failed at source localisation stage. Consequently, with different searching strategies at source localisation stage, this research investigated five further algorithms. The results demonstrated that the algorithm with a specially-designed *pseudo casting* source localisation method is the best approach to localising hazardous plume sources in the underground warehouse given in this research or other similar environments, among all the tested algorithms. The second part of the study is reported in Chapter 5 of this thesis.

Contents

List of publications

Z. Li, Z. F. Tian, T.-f. Lu, and H. Wang, "Assessment of different plume-tracing algorithms for indoor plumes," *Building and Environment,* p. 106746, 2020.

Z. Li, Z. Tian, T.-f. Lu, and H. Wang, "Localisation of fire source in a warehouse using plume-tracing method," presented at *The 11th International Conference on Computational Methods (ICCM20)*, online.

List of figures

List of tables

Abbreviations

CFD	Computational Fluid Dynamics
PS	Plume Sensing
PT	Plume Tracking
SL	Source Localisation
RANS	Reynolds-Averaged Navier-Stokes
i-SCA	inverso Surge Chemo-Anemotaxis
UAV	Unmanned Aerial Vehicle

Chapter 1 Introduction

1.1 Introductory background

Unmanned mobile robots navigated by plume-tracing algorithms, which evolved from bionics, can be employed to detect and localise hazardous plume sources in indoor environments, to protect people from unsafe circumstances [3-5, 7-43]. It was first reported in a large and growing body of literature that insects and underwater organisms, such as moths and lobsters are able to track odour plumes in air or under water [24]. For instance, researchers have undertaken studies on the behaviours of male moths when they search for the source of released pheromones within plumes [44-46]. Following the moths' searching strategy of surging upwind, various plume-tracing algorithms were further developed in subsequent investigations [4, 40]. Plume-tracing algorithms have been adopted to search for contaminant and chemical sources in indoor environments such as laboratories and they have proven to be powerful search tools [15]. With efficient algorithms, plume-tracing mobile robots are promising to provide an effective approach to preventing hazards and reducing health risks in more indoor environments.

Generally, a typical plume-tracing algorithm can be divided into three stages: plume sensing (PS), plume tracking (PT), and source localisation (SL) for discussion and analysis. The boundaries between them are defined by the local concentration of the target chemical [30, 32, 38, 47]. The first stage, plume

sensing, compounds different types of *casting* behaviours aiming to find a plume under the condition that the robot locates outside the plume [28]. Plume tracking is the stage at which the robot remains within the plume and manoeuvres towards the source. Normally after plume tracking, source localisation stage takes place where the robot is deemed to be near the source and prepares to declare its location.

1.2 Plume tracing in experiments

Different plume-tracing algorithms have been proposed, assessed and applied [25]. A comparison across four plume-tracing algorithms in the localisation of a source of an ion plume was carried out experimentally by Harvey, et al. [40]. The experiments were conducted in a wind tunnel with an ion generator positioned on the central line of the wind tunnel and the wind speed was set at three different values: 0.55 m/s, 0.95 m/s, and 1.4 m/s. *Surge anemotaxis*, which is the plume-tracing algorithm that navigates the robot to surge upwind continuously, proved to be the most effective algorithm among four algorithms: *surge anemotaxis*, two *bounded search algorithms* and *counterturning* [40]. A subsequent comparison by Harvey, et al. [4] was undertaken within the same wind tunnel and location of the source. However, in this research, a robot coded with different plume-tracing algorithms was tested in a shifting wind field [4]. The experimental results showed that plume tracing in shifting wind fields is applicable [4]. Moreover, by partly improving simple algorithms, for instance, by revising turning angles modestly, the performances of the robot could be improved [4]. Lu [2] proposed a novel searching strategy whereby a robot moves at a shorter distance every step when the chemical concentration increases as the robot approaches the source to reduce the possibility of missing the source when moving near it. Moreover, some researchers have tried to apply visual

technology to estimate the position of a plume source [33]. For example, a firefighting robot with real-time visual detection technology was recently developed by Kim and Lattimer [7]. The unmanned firefighting robot performed well in searching for a fire source by analysing the images of the smoke in an on-fire building. However, this robot, despite its visual detection technology, can only be used for detecting visible smoke when the fire is large and the source obvious. The same problem applies to other investigations using visual detection technology, for instance, the firefighting robot developed in [8]. In order to minimise the damage, it is best to detect and localise the hazardous plume source at an early stage and this can be achieved by plume tracing methods.

1.3 Plume tracing in a simulation framework

Apart from the experiments, a simulation framework was developed to test and train plume-tracing algorithms [2, 11, 16, 47, 48]. Simulations were conducted in an environment created by computational fluid dynamics (CFD), providing a novel and repeatable method of testing and training robots coded with different plume-tracing algorithms. One such simulation framework, combining FLUENT (ANSYS Inc.) and MATLAB (Mathworks Inc.), was presented by Liu [10], Liu and Lu [11], Liu and Lu [16]. It has been validated that the combination of the CFD-produced environment and MATLAB-based robot(s) with plume-tracing algorithms is appropriate and applicable. For example, Lu [2] undertook a case study using this simulation framework to investigate the effects of initial location, initial orientation and moving distance on the performance of the robots with plume-tracing algorithms. Initially, the simulation framework was two-dimension (2D), meaning that the data produced in CFD were from only one particular height, focused for the simulation. However, data, including the chemical concentration distribution and the wind field from a 2D simulation framework

are not sufficient, as all plumes are strong three-dimension (3D) domains. A 3D simulation framework using data from different horizontal levels, was subsequently developed by Awadalla, et al. [49] to create a realistic environment for testing and training plume-tracing robots. Other research concerning scenarios in a 3D simulation framework can be found in the literature [22, 29, 42, 47].

1.4 Significance of this research

Despite the various attempts made by researchers to either design novel algorithms or improve existing algorithms, most of these efforts were aimed at free plumes such as plumes in an outdoor environment or indoor plumes where the chemical sources are located away from walls. To the best of the author's knowledge, none of these research projects were targeted to discover how plume-tracing algorithms and robots perform when a plume source is on or near a wall, where a wall plume forms. A possible scenario for such wall plumes is smouldering caused by a short-circuit cable on a wall. Plume-tracing robots that are designed for indoor environments should be able to avoid obstacles such as walls and still be able to trace wall plumes efficiently. Since the performance of plume-tracing robots could be different in wall plumes, investigations about how the plume-tracing algorithms and the obstacle avoidance system perform together when localising the gas sources of wall plumes in indoor environments are needed. In addition, research into the performance of plume-tracing algorithms in large-scale indoor environments such as warehouses, grocery stores and workshops are still lacking.

Therefore, the overall aim of the project is to assess the performance of widely-used plume-tracing algorithms in several indoor environments, especially for wall plumes and, based on the assessment, to propose and test more efficient

algorithms for plume-tracing robots. Employing the 3D simulation framework, this study first assesses the performance of several widely-used plume-tracing algorithms in detecting and localising the source of ion plumes in two scenarios, where the air is flowing in a wind-tunnel-like channel. In one scenario, the plume source is located at a position on the centreline of a confined environment. In the other scenario, the source is located next to one wall of the channel, where a wall plume forms from this source. Two types of plume sensing and source localisation methods, four types of plume tracking methods, and thus sixteen different plume-tracing algorithms were selected and tested in both scenarios. Robots coded with different algorithms were released from five different initial locations near the air outlet boundary. After testing and analysing all the plume-tracing algorithms, the algorithm with the lowest number of steps, costing the least and consequently showing the highest efficiency among the sixteen algorithms was identified and further improved with a special *along-wall obstacle avoidance method* (*Vallumtaxis*), which navigates the robot to surge along a detected wall instead of leaving. The searching strategy of moving along a wall or obstacle was previously called *wall following* and mentioned and developed in [10, 41, 50]. Different from the previous *wall following* searching strategies, the robot coded with *vallumtaxis* is not only capable of moving along a wall, but also able to leave the wall if the wind field or chemical concentration near the wall changes significantly when searching in wall plumes. When a wall is detected in front by ultrasonic sensors, the robot turns to the direction that is parallel to the wall, with an acute angle to the upwind direction. Moreover, when being too near to an obstacle, the robot will move away from it to provide random chances for the robot to leave a wall when searching.

In the second part, ten different plume-tracing algorithms, including *vallumtaxis* are further tested in an underground warehouse with a single hazardous plume

source in four scenarios, where the locations of the source are different. *Chemotaxis* is not tested in the second part of the research, as the robot coded with *chemotaxis* fails in all the cases in the wind tunnel. Based on the simulation results created in CFD, the fluid field, as well as the chemical concentration distribution are more complex in large-scale environments, as recalculating flows were found. Five plume-tracing algorithms, including the original *vallumtaxis* was firstly assessed. Five more plume-tracing algorithms with different searching strategies at source localisation stage, including a novel searching strategy called *pseudo casting*, were subsequently presented and tested.

Chapter 2 Literature review

In Chapter 2, a literature review regarding plumes and plume tracing, the application of plume-tracing robots and comparisons among different plume-tracing algorithms is presented. At the end of this chapter, the research gaps, the research aims and objectives are summarised and presented.

2.1 Plumes and plume tracing

It is of paramount importance to detect and localise the sources of hazardous contaminant and chemical plumes in indoor environments, which may come from explosives, early-stage fires, and gas leaks [47]. A plume, in this research, is defined as the downwind trail formed from the mixture of the chemical molecules in fluid movements [47]. Numerical models can be used to describe the time-average chemical concentration distribution in a plume and thus, the shape of a plume [51]. A free stream plume can be governed by a reactive plume

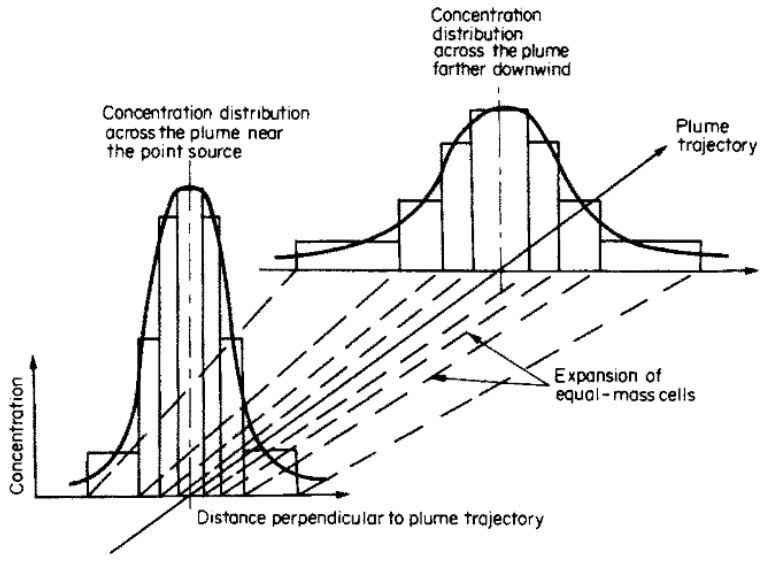


Figure 2.1: Schematic of plume expansion as a function of downwind distance [1]

model and a schematic view of the numerical model of a plume is shown in Figure 2.1 [1]. It is shown that as a plume moves towards the downwind direction, the width of the plume increases. In this model, several equal-mass cells are used to define the concentration distribution of the chemicals within the plume and the expansion of equal-mass cells is given in Figure 2.1 [1].

A cross-section of the plume is presented as an array of well-mixed cells perpendicular to the plume centreline, which contains a fixed amount of the chemical masses following the simple translation given in Equation 2.1 [1]:

$$s = \int_0^t u(t)dt \qquad (2.1)$$

where s is the cross-section of the plume, t is the time since emission and u is the wind speed along the plume trajectory. Moreover, as it is shown in Figure 2.1, the total width and depth of the plume can be governed by Equations 2.2 and 2.3 [1]:

$$W(s) = 4\sigma_y(s) \qquad (2.2)$$

and

$$H(s) = 4\sigma_z(s) \qquad (2.3)$$

where σ_y and σ_z are the horizontal and vertical dispersion coefficients described as [1]:

$$r^2 = \frac{y^2}{\sigma_y^2} + \frac{z^2}{\sigma_z^2} \qquad (2.4)$$

where r is the polar coordinate, y and z are the horizontal and vertical extents of the cells, respectively. In summary, the peak chemical concentration reduces gradually and the width of the plume increases following the plume trajectory. This model is based on the Gaussian model given in Equation 2.5 [52]:

$$C_{(x,y,z,H)} = \frac{Q}{2\pi\sigma_y\sigma_z\bar{u}} \exp\left[-\frac{1}{2}\left(\frac{y^2}{\sigma_y^2} + \frac{(z-H)^2}{\sigma_z^2}\right)\right] \qquad (2.5)$$

where $C_{(x,y,z,H)}$ is the chemical concentration at points (x, y, z, H). In Equation

2.5, x is defined as the distance in the downwind direction, and y is the crosswind distance. z is the vertical direction and H is the height above the ground. Q is the releasing rate of the chemical from the plume source, σ_y and σ_z are the same coefficients given in Equation 2.4 and $\bar{u}$ is the mean wind speed. Equations 2.1 and 2.5 define the shape of a plume that forms from a point source [52]. However, since nearly all the plumes are not laminar, turbulence influences not only the chemical concentration distribution but also the fluid field [34]. Variables at different points within a plume were measured by a number of researchers to examine the structure of a plume [34]. It was found that mean instantaneous measurements showed that the plume is highly intermittent, and concentration measurements may produce the distribution of the Gaussian model [34]. A plume forms in the downwind direction of a point hazardous plume source and according to the Gaussian model [34], the width of the plume becomes higher following the downwind direction. A point source may produce a wide and detectable plume in the downwind direction. In this case, mobile robots with chemical sensors may potentially be applied to tracing the source of a chemical plume.

The method to detect and localise the sources of various plumes is called the plume-tracing method. The plume-tracing method is a searching strategy inspired by the behaviours of different organisms on locating food and their mates over long distances [46]. It was first documented in the 1970s that some insects are able to move from an area with lower chemical concentrations to areas with higher chemical concentrations continuously within a plume under a changing wind field and concentration of the chemical [53]. Various plume-tracing algorithms were subsequently developed by researchers according to the searching strategies of different organisms, such as moths, lobsters, and birds [24, 33]. Plume-tracing robots have now been used globally in different indoor

and outdoor environments. Research concerning the application of the plume-tracing method on mobile robots has grown substantially [7, 31]. Most of the plume-tracing algorithms can be divided into two categories: *chemotaxis* and *anemotaxis* [31]. *Chemotaxis*, which is inspired by bird flocking and capable of navigating the robot to move from an area with lower chemical concentrations to that with higher chemical concentrations, was the basis for several algorithms in this research. *Chemotaxis* relies on the chemical concentration distribution only. Different from *chemotaxis*, *anemotaxis* tends to navigate the robot to move upwind within the plume. Figure 2.2 shows the searching strategy of *anemotaxis,* which is continuously moving in the upwind direction within the plume. A number of plume-tracing algorithms are designed and developed based on these two algorithms.

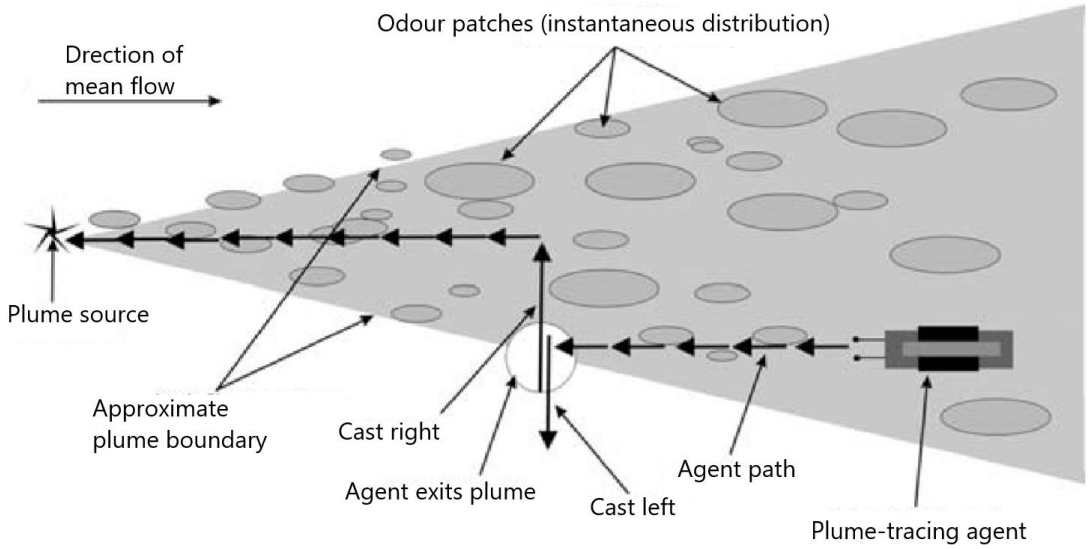


Figure 2.2: Illustration of Robotic lobster performing Odour Gated Rheotaxis based strategy (after [19])

2.2 The application of plume-tracing robots

Chemical plume-tracing algorithms have been tested in various experimental and virtual environments. The experimental research demonstrates that a plume-tracing algorithm is compatible with mobile robots equipped with wind and

chemical sensors [28, 29]. Unmanned robots equipped with plume-tracing algorithms are capable of undertaking dangerous tasks in order to keep people safe [2]. In this case, an obstacle avoidance system is needed for indoor plume-tracing robots. Various obstacle avoidances, such as the Virtual Force Field Method [54], Virtual Field Histogram Method [55], Dynamic Window Approach [56], and Fuzzy-based Obstacle Avoidance Method [57] can be used; however, they are either designed for either a robot that moves continuously or for an application in certain known environments. A Wall-following Method [10, 41, 50] has been performed for indoor plume-tracing robots, which usually move step by step. For the Wall-following obstacle avoidance method, the robot moves following a wall. To the best of the author's knowledge, most current obstacle avoidance methods for mobile robots rely on distance-measuring sensors. Other information, for example, the wind velocity near the obstacle, has not been used for obstacle avoidance systems.

An obstacle avoidance system for plume-tracing robots using ultrasonic sensors was designed by Awadalla, et al. [47] and then widely validated in subsequent works [2, 5, 14, 15, 25]. Figure 2.3 shows the covering areas of the four ultrasonic sensors in [2], and Figure 2.4 shows the covering areas of the five ultrasonic

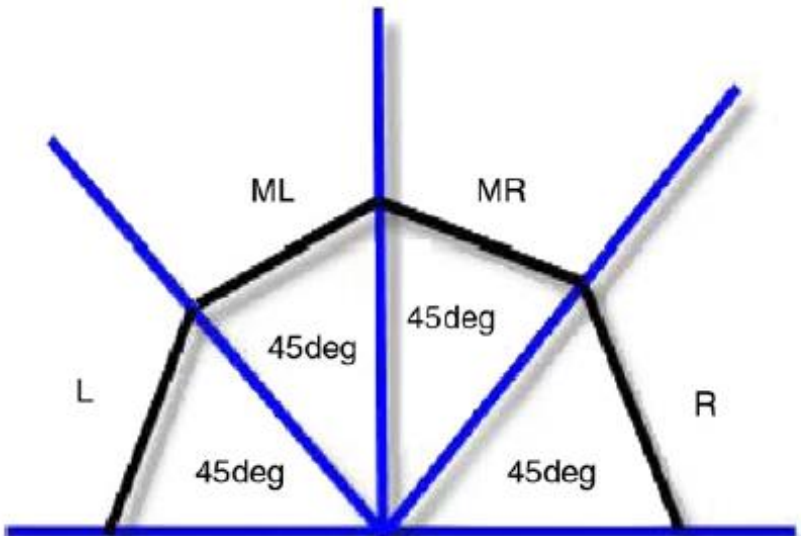


Figure 2.3: The covering angles of four ultrasonic sensors [45]

sensors in [15]. The robot reacts differently according to the data measured by these sensors. For instance, for some obstacle avoidance methods, if an obstacle is detected on the left hand side of the robot, the robot will move slightly to the right in order to avoid it [2]. Also, the robot surges along an obstacle for some other obstacle avoidance methods [10]. In this condition, the robot would calculate the angle between the obstacle and the robot heading using multiple ultrasonic sensors [10]. Equipped with such sensors, the plume-tracing robot is capable of avoiding obstacles and has the potential to be applied safely in environments with walls and obstacles.

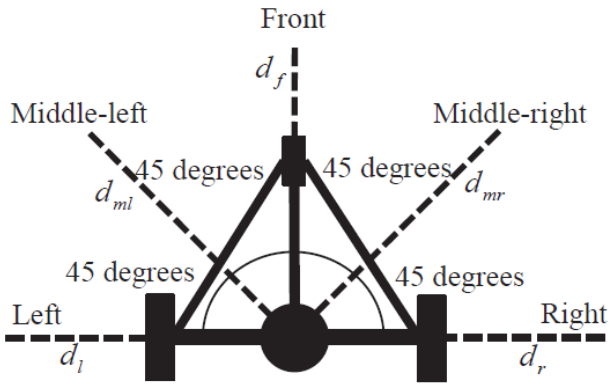
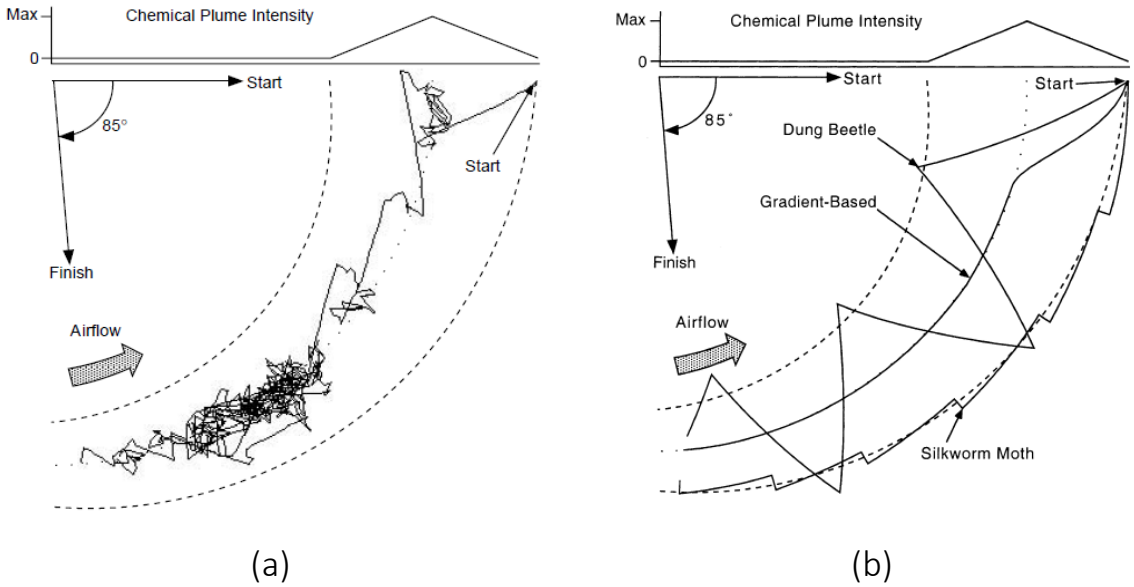


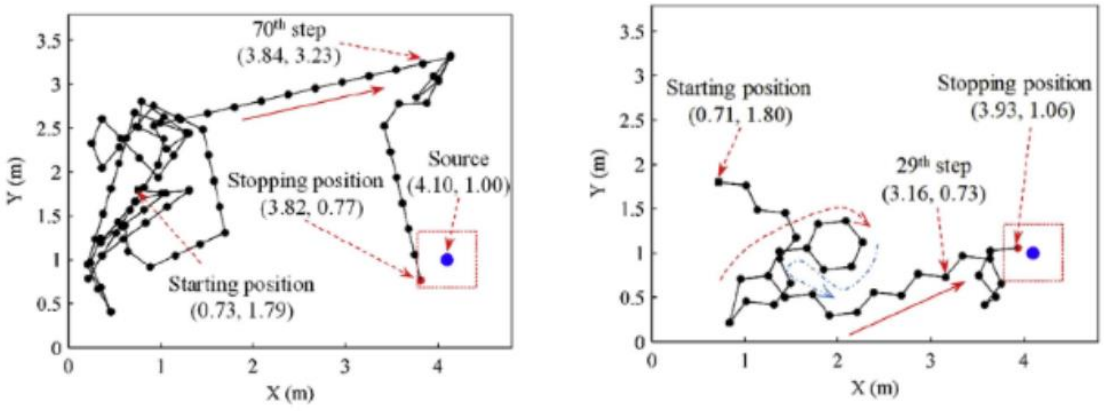
Figure 2.4: Ultrasonic sensors setting for measuring five directions [15]



(a) (b)

Figure 2.5: (a) Simulation results of the robot following a chemical trail using the E. coli algorithm, (b) Simulation results for trail following (robot released on the outer edge of the chemical plume) [38]

A number of tests regarding the application of plume-tracing algorithms on mobile robots have been performed by researchers. For instance, several plume-tracing algorithms, namely the *E. coli algorithm, dung beetle algorithm, gradient-based algorithm* and *silkworm moth algorithm* were tested using a mobile robot in [38]. Figures 2.5 (a) and (b) represent the trajectories of the robots driven by different plume-tracing algorithms. The results show that, with a steady airflow, the four plume-tracing algorithms tested are all capable of localising the source of the plume. The *E. coli algorithm*, with a randomly-moving mechanism towards the direction of the area with higher concentration, was found to require the least effort in sensing and controlling the robot [38]. However, being tested on a small scale with a laminar flow field, this *E. coli algorithm* is only proven to be effective in very small scale systems or areas with low turbulence affecting the concentration distribution [38]. Furthermore, the *silkworm moth algorithm*, which is similar to the *surge anemotaxis* performed in [40] was identified as being capable of supporting a robot to search in a turbulent chemical plume [38]. In this series of studies, plume-tracing algorithms were tested in a small area with a steady airflow and with the source away from walls.



(a) (b)

Figure 2.6: (a) A successful experiment using the E.coli algorithm, (b) A successful experiment using the hex-path algorithm [5]

Another test was performed in indoor environments with low-speed airflows (approximate 0.3 m/s) by Yang, et al. [5]. Figures 2.6 (a) and (b) show the trajectories of two successful experimental cases using a robot coded with several plume-tracing algorithms. The results show that a single plume-tracing robot is capable of detecting and localising the source of ethanol in experimental work in an environment with low wind speed; for example, an indoor environment [5]. Since the plume-tracing algorithms developed based on *surge anemotaxis* rely significantly on wind direction, this experimental study shows that a plume-tracing mobile robot can be widely used in various indoor environments with high or low airflow speed.

Despite the tests of plume-tracing algorithms in the experiments in wind tunnels, it can be challenging to undertake experiments to detect and localise hazardous plume sources or test plume-tracing algorithms in large industrial buildings, such as warehouses and workshops. Therefore, beyond these experiments, a novel simulation framework was developed to test and train plume-tracing algorithms and robots. This simulation framework was firstly presented by Liu and Lu [16] and involved combining CFD-predicted virtual environments with MATLAB-based plume-tracing algorithms. With the rapid development of computing technologies, CFD has been applied to simulate various fluid fields like chemical concentration distributions and wind fields in warehouses [47]. The CFD-produced fluid field, as well as the chemical concentration distributions, are exported to MATLAB for access. The CFD simulation framework has been validated in the work conducted by Lu [2]; Li, et al. [25] and Liu [10]. A good match of the simulated robot trajectories and measured robot trajectories was found in the validation work [2], indicating a simulation framework combining ANSYS/FLUENT and MATLAB-based plume-tracing algorithms is a valid research tool to test and train plume-tracing robots in a low-cost, fast-created and reliable

environment. Employing this simulation framework, Lu [2] proposed a novel plume-tracing algorithm, namely "inverso Surge Chemo-Anemotaxis" (*iSCA-taxis*), which depends not only on wind direction but also on chemical concentration levels. Different from *surge anemotaxis* with a *constant stepsize*, for the novel *iSCA-taxis* proposed in Lu [2], the robot moves at a larger step size when the target chemical concentration decreases. Conversely, the robot moves at a smaller step size when the local chemical concentration increases. When the chemical concentration is low, this mechanism can help the robot move faster, thereby reducing the searching time and improving the searching efficiency. Figure 2.7 shows a success case of *iSCA-taxis* in a relatively large-scale environment. The environment in this study is part of Level 2 of the Engineering South building at the University of Adelaide, and its size is 25 m on one side and 26 m on the other side, which is much larger than the arena of the previous experiments (about 9 m^2). The results show that before entering the room, the robot went in the wrong direction once and then went back because it sensed a

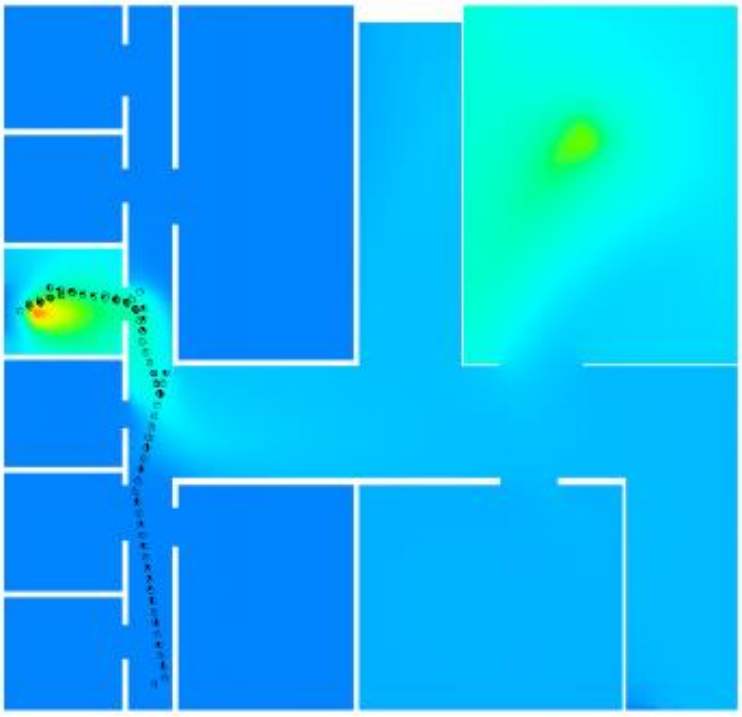


Figure 2.7: A successful case of iSCA-taxis [2]

decreasing chemical concentration (Figure 7). However, the source is still in the central part of a room, and so the performance of a plume-tracing robot in tracing wall plumes remains unclear.

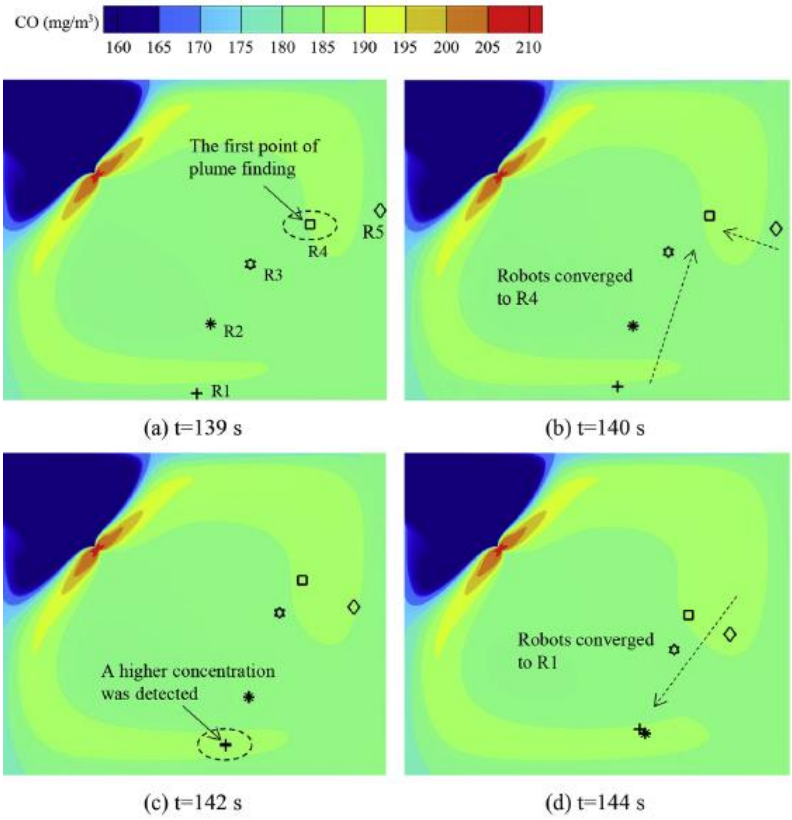
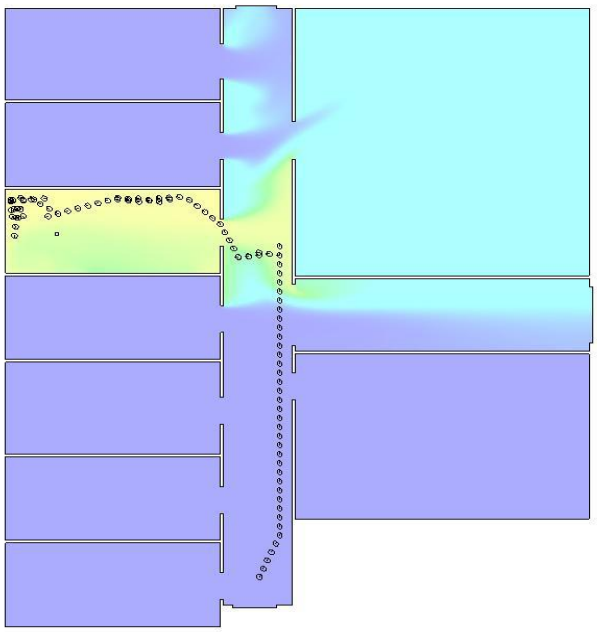


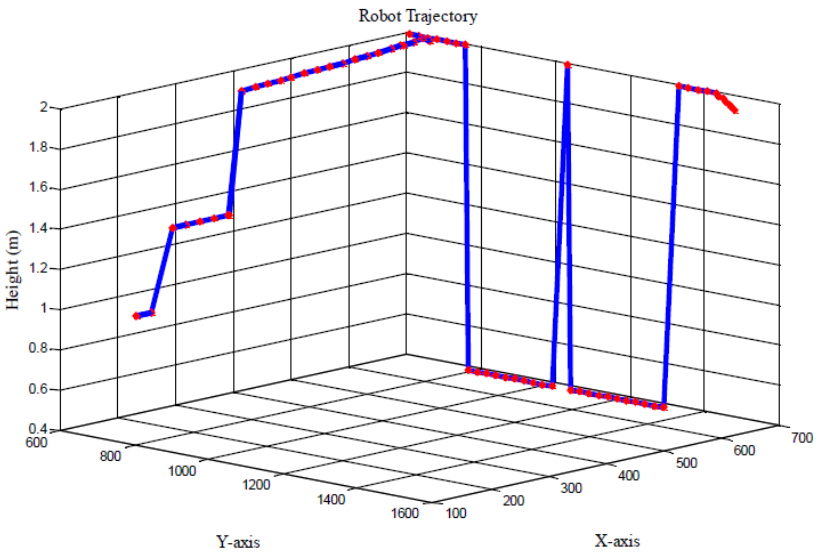
Figure 2.8: Procedure of contaminant source localisation (after [6])

Another test of a plume-tracing robot under the simulation framework is where Chen, et al. [6] investigated a plume-tracing task in a similar (small) indoor environment using the *multi-robot active olfaction method*. In this case study, FLUENT with the *k-ε* transient model was used to simulate an indoor airflow field (Figure 2.8). Compared with the environment investigated by Lu [2], the scale in this study is smaller but the inlet airspeed changes with time, resulting in a more complex wind field. Figure 2.8 shows the chemical concentration distribution, as well as the position of the plume-tracing robot at that time. Under the application of a *multi-robot active olfaction method* on a swarm of robots, the chemical source was successfully localised by the robots. This study further proved that

the application of this simulation framework on testing various plume-tracing algorithms is feasible. However, the geometry of the room in this investigation is very simple and there are no obstacles inside. Normally there are various objects and walls influencing the wind field as well as the movement of a plume-tracing robot in a room occupied by people. Hence, more tests need to be undertaken in



(a)



(b)

Figure 2.9: (a) Searching trajectory of the robot (b) 3D searching trajectory in 3D simulation framework [49]

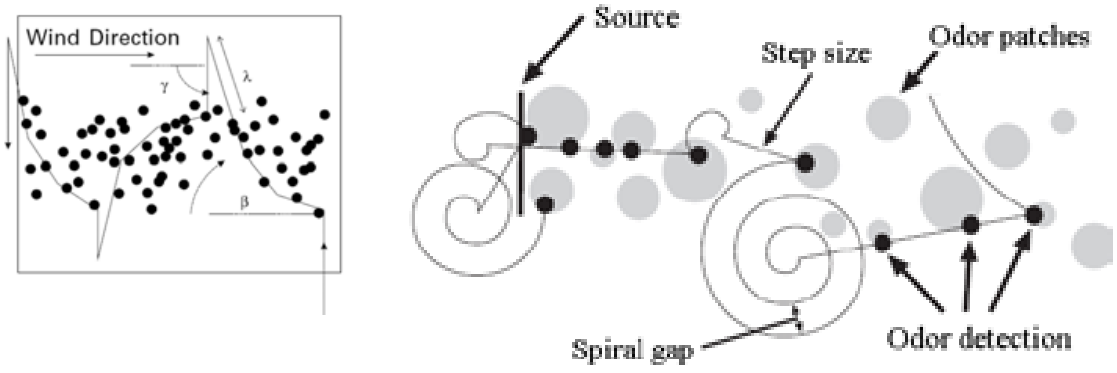
complex indoor environments because walls and obstacles can influence wind fields and chemical concentration distribution and therefore influence the performance of different plume-tracing algorithms.

All the above simulation studies are based on a 2D simulation framework, which means that only one horizontal level of CFD results, including chemical concentration distributions and wind fields, is imported into MATLAB for further processing. Since plumes are mainly strong 3D flows, a 3D simulation framework is necessary for these strong 3D plumes. Awadalla, et al. [47] developed a 3D simulation framework using ANSYS/FLUENT and MATLAB. In this 3D simulation framework, data, including chemical concentration distributions and fluid fields on multiple horizontal levels, are exported to MATLAB for further processing. It can be concluded from Figures 2.9 (a) and (b) that the application of a plume-tracing robot using a 3D simulation framework that provided wind fields and chemical concentration distributions on different horizontal levels is feasible.
Although a number of successful cases of plume-tracing robots are found in the literature, problems still remain. Firstly, the locations of sources are away from the walls. Since it can be normal that a hazardous plume source is near to or on a wall and wall plumes are different from free-stream plumes, it is important that plume-tracing algorithms can be tested on wall plumes. Moreover, most of the tests were performed in lab-scale environments. To apply plume-tracing algorithms in more indoor environments, such as warehouses, more tests need to be performed in large-scale environments.

2.3 Comparisons among different plume-tracing algorithms

To understand the performance of different plume-tracing algorithms, assessments of different plume-tracing algorithms have been conducted

experimentally and numerically. A number of research projects assessing different plume-tracing algorithms using experimental methods can be found in the literature [5, 14, 35, 38, 39]. A parametric study regarding the speed limit of plume-tracing robots using two different algorithms *plume-maintaining behaviour with active strategy* and *spiral-surge algorithm,* was undertaken by Li, et al. [3]. Figures 2.10 (a) and (b) show the searching strategies of these two algorithms, respectively. The testing results show that in a given indoor airflow, the average approaching indexes generally increase as the maximum speed of the robot goes up [3]. Also, the overall performance of the *Spiral-surge algorithm* is better than that of *Plume-maintaining behaviour with active strategy* [3]. However, it can be seen that the plume-tracing robot in this study was tested in an indoor environment away from any walls and obstacles [3]. It can be observed from the depiction drawing of the *spiral-surge algorithm* that a large space is



(a) (b)

Figure 2.10: (a) Depiction of the plume-maintaining behaviour; (b) Depiction of the spiral-surge algorithm [3]

needed for a spiral motion. Consequently, even though the average approaching indexes of the *spiral-surge algorithm* is shown to perform better, there is not enough space for the *spiral-surge algorithm* in many indoor environments.

In addition, Harvey et al. [40] assessed several insect-inspired chemical plume-tracing algorithms using a mobile robot equipped with a wind and an ion sensor.

In this case study, the source of the plume is an ion generator, which can be treated as a point source, and a plume formed in a wind tunnel. The wind tunnel provides indoor environments with a steady airflow. Four plume-tracing algorithms, namely *surge anemotaxis*, two *bounded search algorithms* and a *counterturning algorithm* were tested in this study. The experimental results show that the *bounded triangular algorithm* presents the highest overall success rate for searching; while in terms of efficiencies, the robot with *surge anemotaxis* spent the minimum time localising the source. Across the whole process of the experiment, when the wind speed was set to be 0.95 m/s (a typical airspeed in indoor environments), the robot with *surge anemotaxis* achieved the highest success rates. Figure 2.11 shows the trajectories of the robot coded with different algorithms when searching. It could be seen from Figure 2.12 that the robot coded with *surge anemotaxis* moves in a nearly straight line after entering the plume, thereby contributing to a lower total surge distance and a higher efficiency than the robot driven by other algorithms.

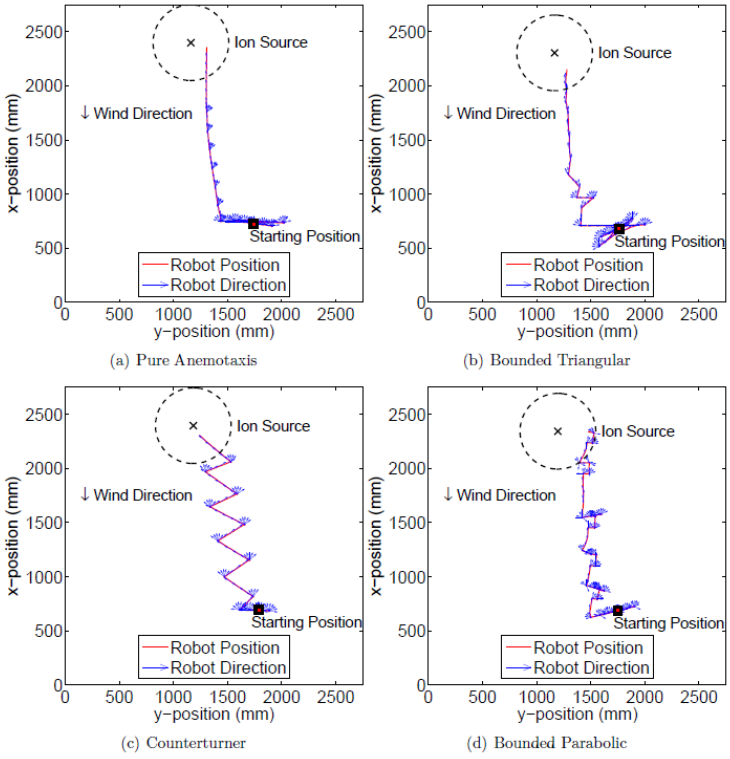


Figure 2.11: The trajectory of robots using different searching algorithms [40]

Targeting searching along an increasing gradient of chemical concentration, another novel plume-tracing algorithm called the *pseudo gradient-based algorithm*, with a special PT stage, was investigated by Neumann, et al. [35].

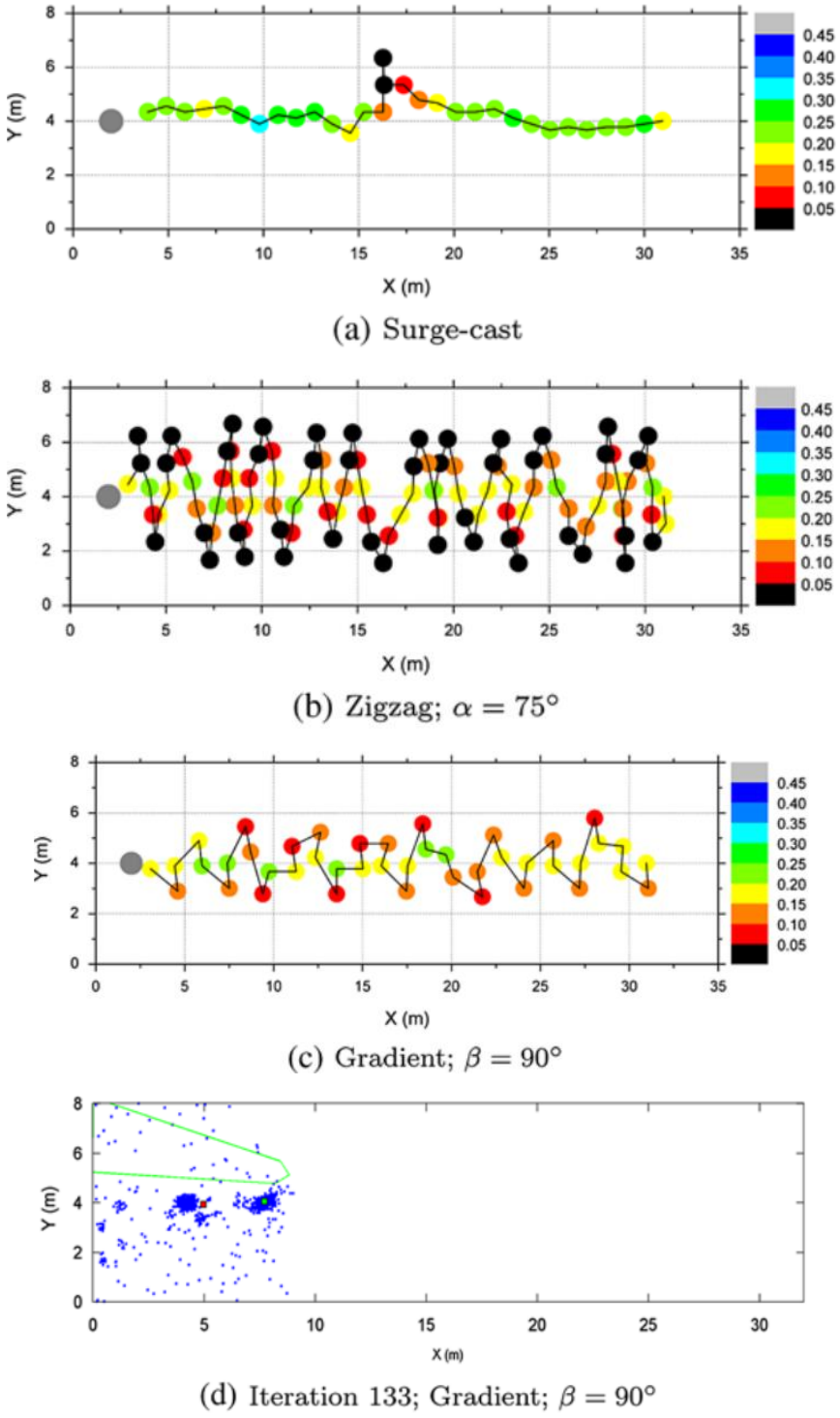


Figure 2.12: (a)–(c) Trajectories of successful simulation runs of all three bio-inspired plume-tracing algorithms, (d) Particles of the PF-based gas source localisation algorithm using the *pseudo gradient-based algorithm* after 133 iterations. [35]

Navigated by this algorithm, an unmanned aerial vehicle (UAV) at plume tracking stage moves towards the area with an increasing chemical concentration based on surging upwind. Figure 2.12 shows the trajectories of the UAV equipped with different plume-tracing algorithms. It can be seen that the *Surge-cast algorithm* (*anemotaxis*) remains the best (see Figure 2.12). This research shows that the application of the *pseudo gradient-based algorithm,* which is a novel algorithm navigating the robot to move towards the increasing concentration direction when moving upwind, on a micro-drone in an outdoor environment, is feasible. Also, since air flow usually has low speed and turbulence in indoor environments, this algorithm has the potential to be applied to plume-tracing robots in the detection and localisation of hazardous plume sources in various indoor environments.

Harvey, et al. [4] also undertook an experimental study to compare the effectiveness of different plume-tracing algorithms in a shifting wind field (Note: the direction of the wind varies with time). In this research, novel plume-tracing algorithms especially designed for a shifting wind field are presented [40]. The *surge anemotaxis with shift adjustment* is an improved *surge anemotaxis,* with a moving orientation adjustment mechanism equipped particularly for adjusting to the shifting wind direction. Various methods, including adjusting turning angles or decreasing surge time, were tested to get higher searching efficiency for plume tracing in environments with shifting wind [4]. Table 2.1 shows the success rates of the algorithms applied in this series of experiments. It can be seen that traditional simple algorithms (e.g. *surge anemotaxis*) presented by Harvey, et al. [40] still performed the best in terms of success rates (Table 2.1). All the algorithms give perform success rates higher than 0.8.

Table 2.1: Success rates of different algorithms [4]

	Anticlockwise Shift n=14	No Shift n=25	Clockwise Shift n=14	Overall n=53
Simple Surge Anemotaxis	1.00	1.00	1.00	1.00
Surge Anemotaxis With Wind Shift Adjustment	0.86	0.92	0.86	0.89
Simple Bounded Search	1.00	1.00	1.00	1.00
Bounded Search With Wind Shift Adjustment	0.93	0.76	0.79	0.81
Improved Bounded Search	0.86	0.92	0.93	0.91
Longer Surge Bounded Search	0.86	0.96	0.93	0.92
Variable Threshold Bounded Search	1.00	0.72	0.79	0.81

Naeem, et al. [24] compared two searching strategies inspired by bacterium, namely *pure random walk* and *biased random walk*. *Pure random walk* (termed *chemotaxis* in this report) relies on the gradient of increasing concentration alone. It can be observed from Figure 2.13 that it could be challenging for the robot equipped with *chemotaxis* without a biased moving strategy to find the location of the source. Figure 2.13 shows that the Biased Random Walk performed better than the Pure Random Walk in the indoor environment given in this study.

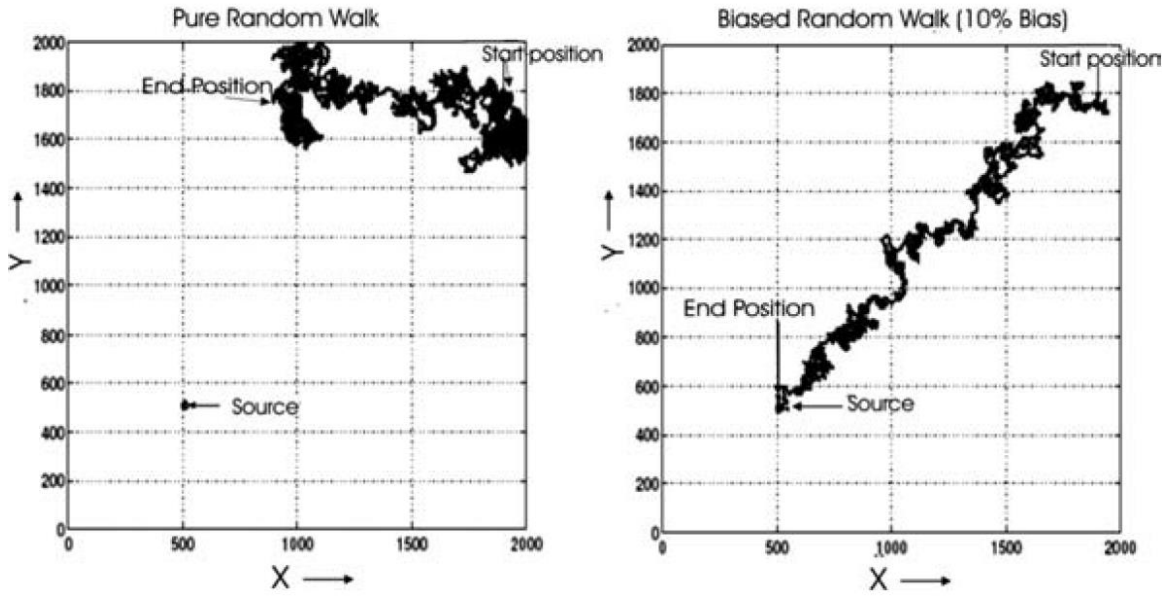


Figure 2.13: Pure random walk vs. biased random walk [23]

In summary, it has been shown that the plume-tracing method has been widely used and tested in different indoor environments. Also, the simulation framework has proved to be a reliable and useful search tool. It is promising for applying a plume-tracing robot to detect and localise hazardous plume sources

in large buildings. Also, research concerning the assessment of different plume-tracing algorithms presents better algorithms in different specific environments. However, there are still some questions in the current literature. As hazardous plume sources, such as gas leakage, explosives and fires could be near a wall, research in the scenarios where the chemical source is near a wall is definitely needed. Also, in previous investigations, plume-tracing algorithms were mostly compared in small-scale spaces, such as laboratories, offices and classrooms. Research on the assessment of plume-tracing algorithms in large-scale environments, for example, warehouses, is needed.

2.4 Research gaps

1. The performance of different plume-tracing algorithms in tracing wall plumes is not yet understood.

In previous literature, various plume-tracing algorithms have been tested and compared numerically and experimentally in a number of indoor environments, where the plume sources are usually away from walls or obstacles. However, wall plumes can be formed from hazardous plume sources such as gas leakages, smouldering and explosives that are located near a wall. Due to the current algorithms not being assessed in scenarios where the plume source is near a wall, the performance of different plume-tracing algorithms for wall plumes needs to be investigated and understood.

2. The proposed *concentration-distance obstacle avoidance method* is inefficient.

For many existing plume-tracing algorithms, when a wall or an obstacle is detected in front of a plume-tracing robot, the robot usually moves away from it. However, preliminary simulations conducted in this project showed that when a wall plume is thin (thinner than the step size of the robot), this obstacle-avoidance method does not perform well because the robot repeats moving

towards and away from the wall, sometimes remaining in an endless loop. Therefore, a novel obstacle-avoidance method based on a wall-following search strategy, is necessary especially for plume-tracing robots searching in the thin wall plumes that can be found in indoor environments.

3. The performance of several algorithms in tracing wall plumes in large indoor environments is not clear.

Plume-tracing algorithms have been tested in some indoor environments such as laboratories, offices and classrooms and have proved to be capable of detecting and localising hazardous plume sources in such indoor environments. However, to the best of the author's knowledge, there is no research targeting on the performance of different plume-tracing algorithms in tracing plumes in large-scale indoor environments such as warehouses. Study regarding the performance of different plume-tracing algorithms in large-scale environments, such as warehouses, is therefore needed.

2.5 Aim and objectives

This research aims to understand the performance of different plume-tracing algorithms in indoor plumes in both small and large environments. Based on the understanding and analysis of this scenario, a new search algorithm that outperforms the existing algorithms has been proposed and tested.

1. To understand the performance of different plume-tracing algorithms for indoor wall plumes.

To identify the most efficient plume-tracing algorithm that fits different circumstances where the plume source is either away from or near a wall, different plume-tracing algorithms need to be tested and compared in two different scenarios. In one scenario the plume source is away from walls and in the other scenario the plume source is near a wall. Under the circumstance that

currently existing algorithms do not perform well in tracing indoor plumes, a new searching strategy is needed.

2. To propose and test an efficient plume-tracing algorithm including an obstacle avoidance method.

Since the strategy of navigating the robot away from the wall does not perform well when searching in wall plumes in indoor environments, a plume-tracing algorithm, called *vallumtaxis,* which includes a novel obstacle-avoidance method, has been proposed and tested in both scenarios proposed in objective 1.

3. To test the proposed plume-tracing algorithms in the detection and localisation of hazardous plume sources in large-scale indoor environments.

A new algorithm which outperforms the existing algorithms for wall plume tracing and other plume-tracing algorithms will be further tested and analysed in large indoor environments. To be more specific, these algorithms will be further tested in different scenarios, where large-scale indoor plumes are formed from near-wall sources.

Chapter 3 Methodology

In this research, a simulation framework combining CFD-created virtual environments and MATLAB-based plume-tracing algorithms is used. In Chapter 3, the methodology including the overall procedure of the simulation framework and the simulation setups in this study is introduced.

3.1 The simulation framework

3.1.1 Introduction of the simulation framework

The simulation framework applied in this study was firstly proposed by Liu and Lu [16]. Subsequent validation of this simulation framework can be found in the work undertaken by Awadalla, et al. [49], Awadalla, et al. [47], Lu [2], Chen, et al. [15] and Li, et al. [25]. The overall procedure is given in Figure 3.1, which shows that data, including the concentration distribution of chemicals or contaminants and the flow field generated from CFD, are exported to MATLAB for further processing. The geometry design of the environment is first modelled. Boundary conditions, such as the mass flow rate of the chemical coming from the source and the inlet wind velocity are input to and processed using CFD. Steady state CFD simulation using a Reynolds-averaged Navier-Stokes (RANS) model (the k-ω-SST turbulence model), was applied in the simulation framework. Therefore, the turbulence field is a Reynolds-averaged field not changing with time and thus the plumes in the fluid field are continuous. Steady state CFD simulation has been widely used in testing plume-tracing algorithms created by steady state CFD in

[2, 10, 13, 15, 16, 19-21, 25, 26, 48, 58]. Figure 3.1 represents the overall procedure of the simulation framework. Data, including the chemical concentration distributions of the chemical, flow fields and contours of the concentration distribution of the chemicals, which are produced by CFD, are processed by MATLAB to simulate the robot tracing plumes under different virtual circumstances. The simulation framework was 2D-based at first, meaning that the concentration distribution and fluid field from only one horizontal level were processed [16]. Since plumes are all 3D domains, the simulation framework was soon developed to be 3D, which means data from several different horizontal levels could be utilised [25, 47].

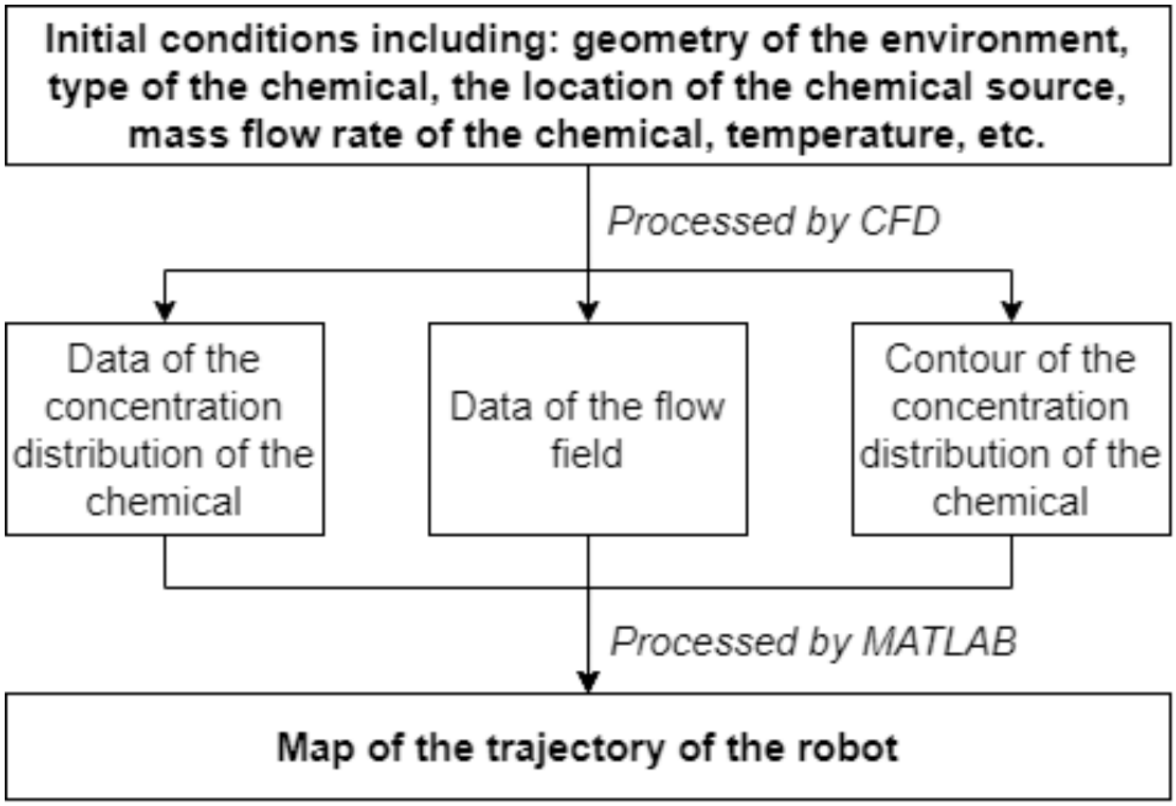


Figure 3.1: The overall procedure of the simulation framework

3.1.2 Validation of the simulation framework

Experimental validation of the simulation framework was first carried out by Liu

and Lu [16] by comparing the trajectories of robots between experimental and simulation results. The experiments are performed in an office with an ion source and a fan providing an ion plume. The simulation results are performed in a CFD-predicted virtual environment. Subsequent validations of the simulation framework, combining CFD and MATLAB, were carried out for both indoor and outdoor experiments with obstacles by Liu and Lu [11] or without obstacles by Lu [9]. In the current research, further validation was conducted based on the experiments undertaken by Harvey, et al. [40]. Figures 3.2 (a) and (b) represent the trajectories of a robot in the experiments and simulation framework. In the experiments [40], the arena is a wind tunnel. The size of the experimental area is approximately 2700 mm in the X direction and 2400 mm in the Y direction (Figure 3.2). The arena in the simulation framework is modelled as a wind-tunnel-like channel that has the same domain as the one in the experiments (Figure 3.2). In the arenas, there is an inlet boundary providing a steady airflow and an outlet boundary on the opposite side. The wind direction can be seen in Figure 3.2. The wind velocity in the CFD model is the same as that in the experiments (0.55 m/s). The ion generator, which is the plume source, is cylindrical (50 mm height and 5 mm diameter) and it emits ion at a mass flow rate of 0.449 g/s to ambient air through the cylindrical side. Data, including the flow field and ion concentration distribution generated in FLUENT 2019 R1, were imported into the plume-tracing codes based on MATLAB R2018b for processing. Figures 3.2 (a) and (b) show that the trajectories of the robot in the experiments and simulation framework are very similar. Hence, a level of confidence in the simulation framework is further obtained and the simulation framework can be regarded as a useful tool for training and testing the plume-tracing robots in this study.

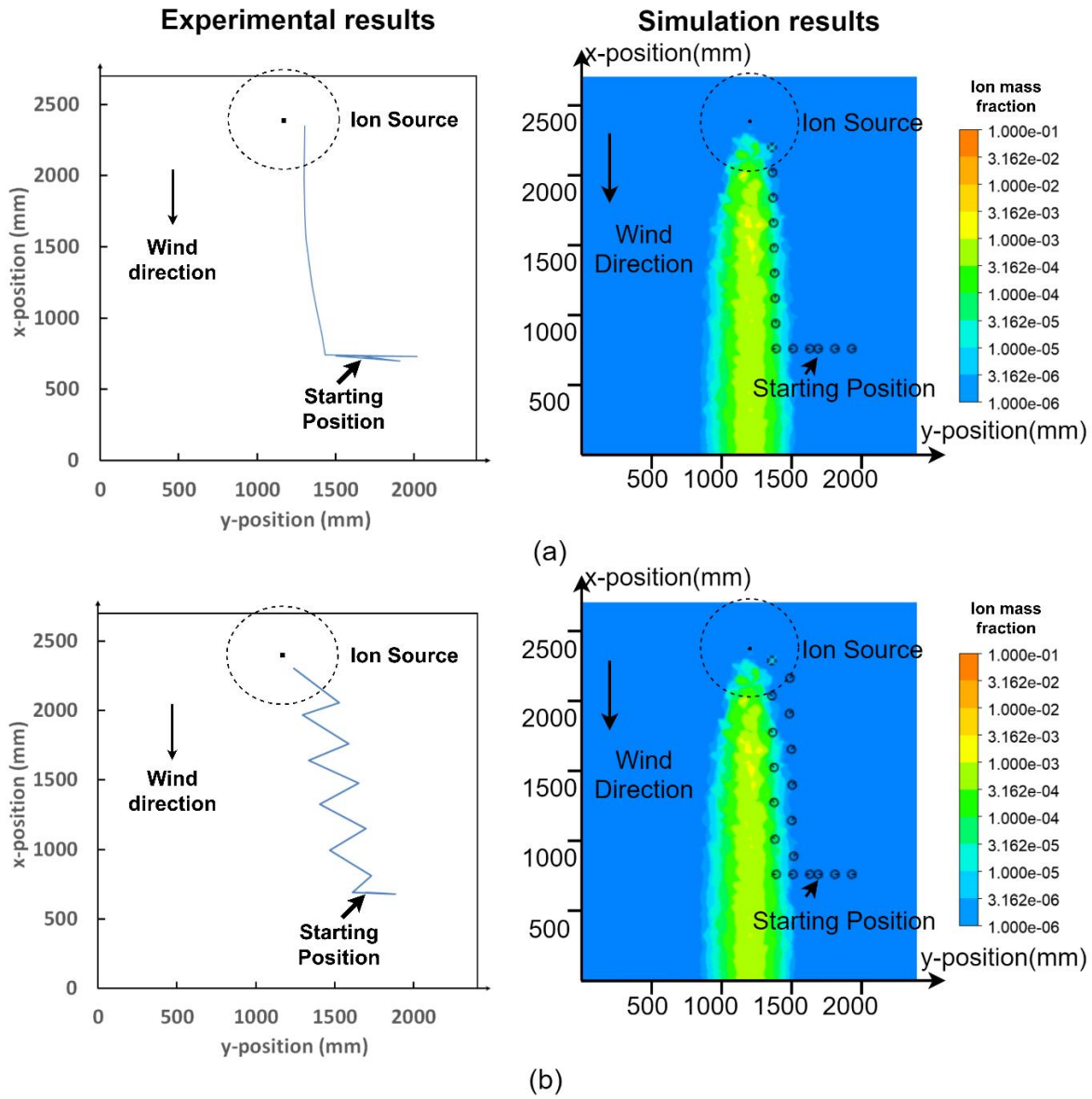


Figure 3.2: (a) Surge anemotaxis robot trajectory in experiments (after Harvey, et al. [40]) and a virtual environment, (b) Zigzags robot trajectory in experiments (after Harvey, et al. [40]) and a virtual environment

3.2 Computational fluid dynamics

Computational fluid dynamics was used to create the environments for investigating the performance of the robot coded with different plume-tracing algorithms.

3.2.1 Simulation setup of the wind tunnel

Figures 3.3 (a) and (b) present two scenarios in the flow confined in a long wind-

tunnel-like channel investigated in Chapter 4. This long channel is specially designed for testing and training plume-tracing robots referring to the domain used by Harvey, et al. [40] and Lochmatter, et al. [39]. The dimension of the current domain in each scenario are 2400 mm (millimetres) in the X direction and 8000 mm in the Y direction. Each boundary of this channel is set as a 5-pixel-thick blank area for testing the obstacle avoidance method. In this research, the source of the plume is an ion generator, which is modelled following the real ion generator applied in the previous investigations [4, 34, 40]. In Scenario M, the coordinate of the centre of the source is (1200 mm, 7715 mm, 175 mm), which is 1200 mm away from both side walls; while in Scenario S the source is located at (80 mm, 7715 mm, 175 mm), which is near the left side wall. In both Scenario M and Scenario S, there is an inlet boundary and an outlet boundary contributing to steady wind flow. The inlet velocity in both scenarios is 1 m/s. A plume that mixes air and ions forms downstream from the source. Five initial locations A-E are shown in Figure 3.3, and their coordinates are A (304 mm, 459 mm), B (650 mm, 546 mm), C (1238 mm, 459 mm), D (1819 mm, 546 mm) and E (2165 mm, 459 mm), respectively. The ion concentration distribution along a horizontal plane 15 cm above the ground is shown in Figure 3.3. The colour legend of the ion mass fraction in the air in Figure 3.3 is logarithmic for better visualisation and it applies to all the following figures in Chapter 4 of this thesis. When the robot declares a location where the distance between the location and the source is less than 350 mm in real scale, which follows the previous experimental investigation undertaken by [40], it is considered a successful search.

The Reynolds number of this confined channel flow can be governed by Equation 3.1:

$$Re = \frac{uD_H}{v} \qquad (3.1)$$

where u is the mean velocity of the flow, D_H is the hydraulic diameter of the

channel and v is the kinematic viscosity of air at room temperature. The wind velocity u is 1 m/s. The hydraulic diameter D_H can be calculated by Equation 3.2:

$$D_H = \frac{2ab}{a+b} \qquad (3.2)$$

where a and b are the width and height of the wind tunnel, respectively. Re is around 56000, which shows that the airflow is a turbulent flow. Moreover, the thickness of the boundary layer δ of the flow can be given in Equation 3.3:

$$\delta = 0.37 \frac{x}{Re^{1/5}} \qquad (3.3)$$

where x is distance downstream from the start of the boundary layer and Re is the Reynolds number. It can be calculated that the boundary layer is small

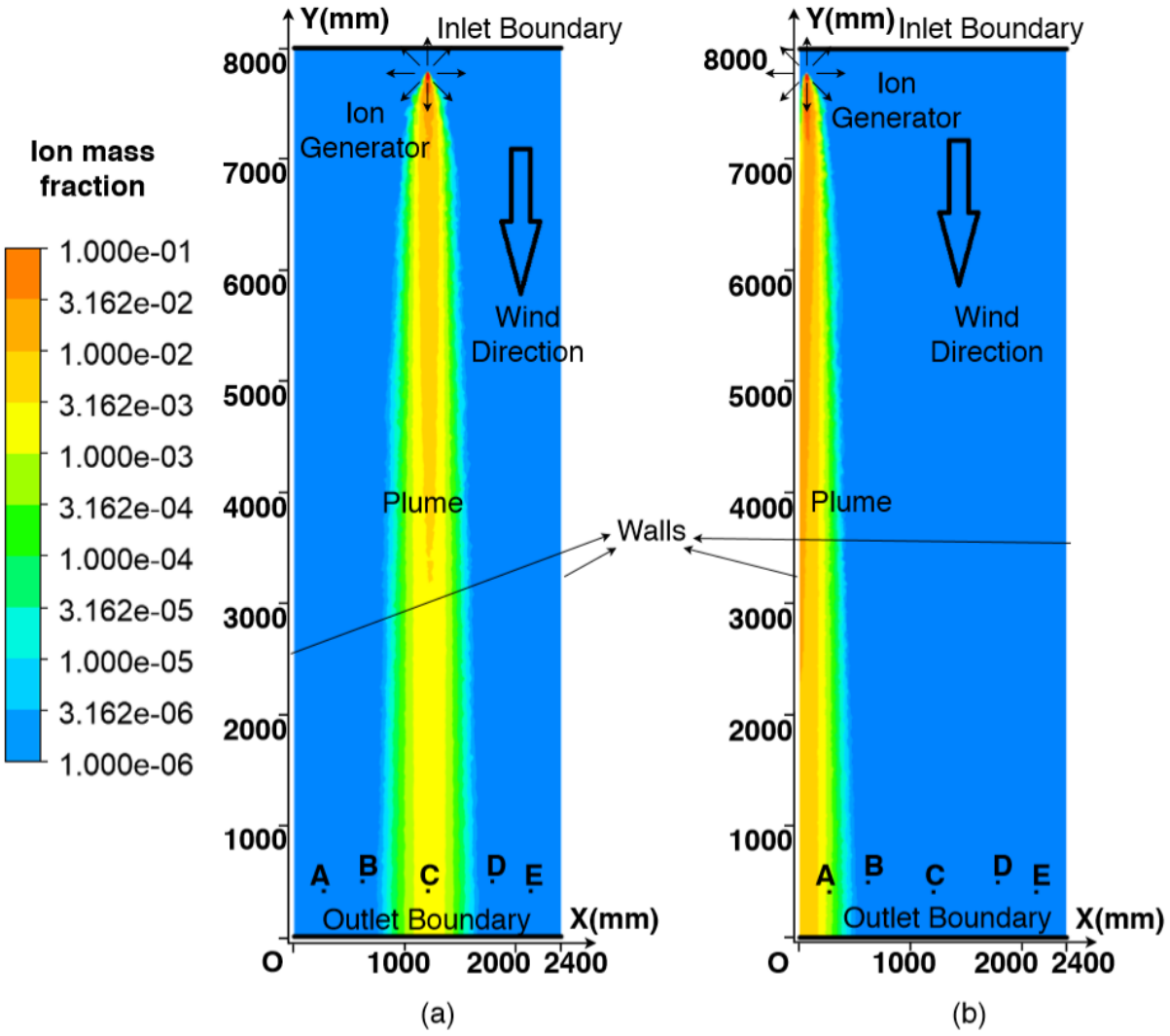


Figure 3.3: Simulation setup (a) Scenario M (b) Scenario S

(maximum 0.33 m) and does not affect the robot significantly.

The simulation framework applied in this research is based on steady state CFD. The advantage of this steady state CFD approach is that the exact same flow fields and plumes are used to compare different plume-tracing algorithms. These consistent environments make the results fair and repeatable. This CFD approach

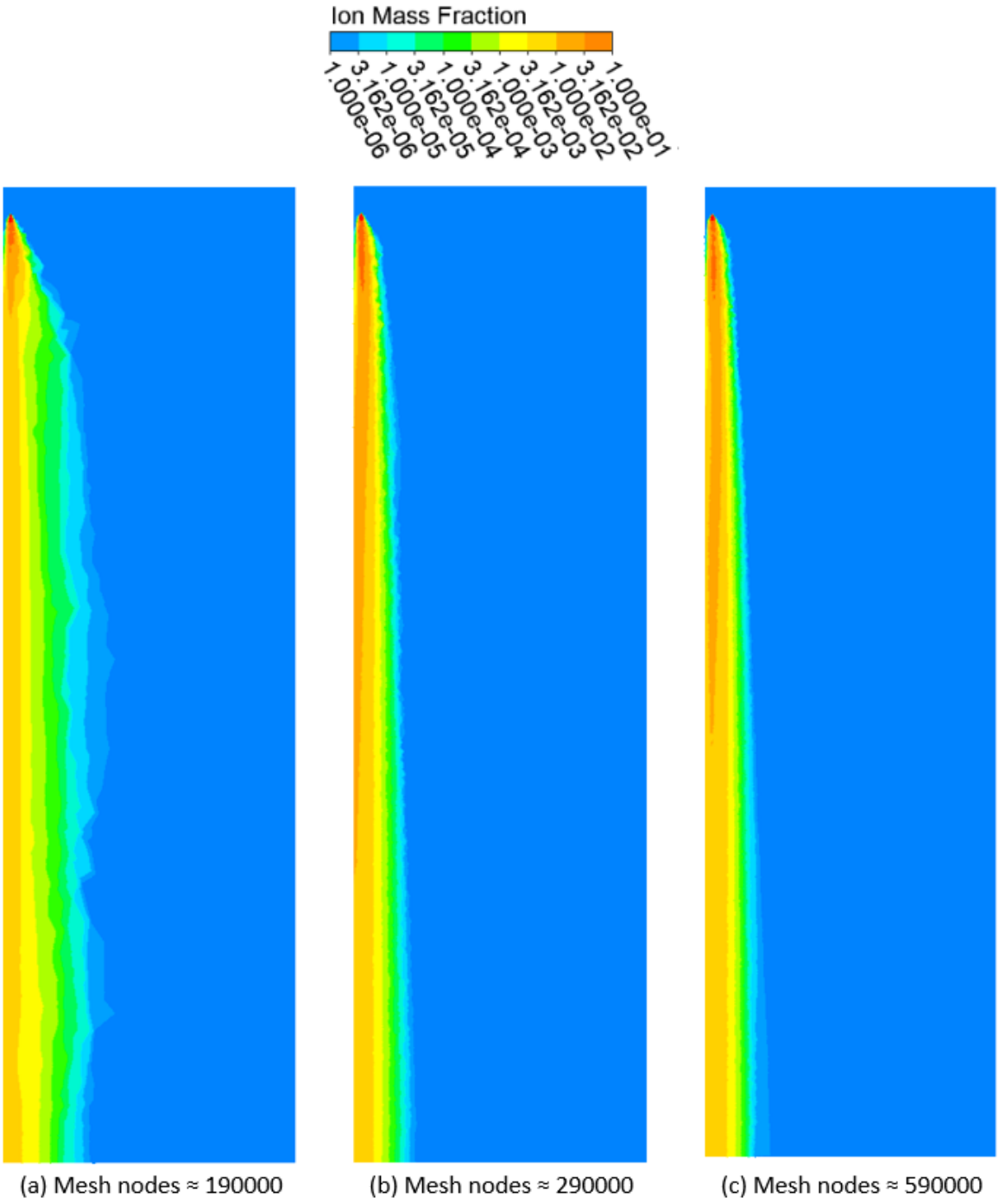


Figure 3.4: Comparison of predicted chemical concentration distribution at 0.3 m high

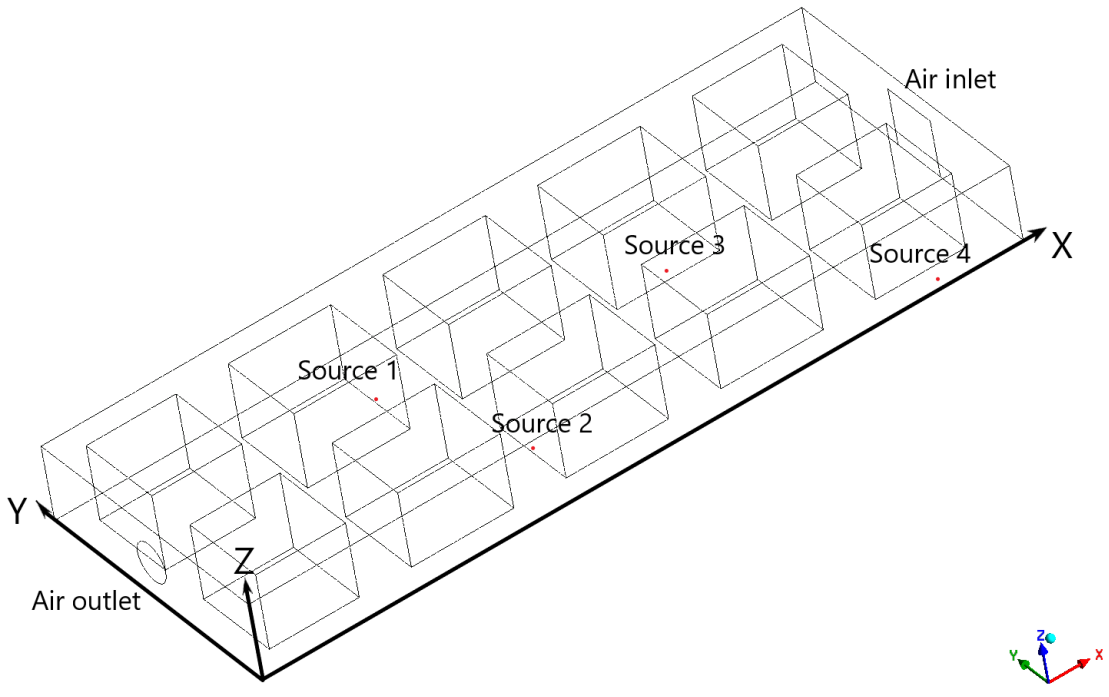
has been widely adopted for indoor airflows and plume-tracing algorithms research, for example, in [2, 10, 13, 15, 16, 19-21, 26, 45, 48]. Moreover, it has been found to be valid as shown in the validations of the simulation framework reported in the literature in Section 3.1.2.

A mesh independent test is conducted for the case in the first part of this research. An initial mesh with about 190000 nodes was generated in the computational domain. The mesh was then refined in finer meshes with 290000 and 590000 nodes. A comparison of the predicted chemical concentration distribution at 0.3 m high is presented in Figure 3.4. It can be seen that the fine mesh (590000 nodes) and medium-density mesh (290000 nodes) yield results that are close to each other. Therefore, a mesh density of 290000 is determined as sufficient and used in the subsequent part of this research. The mesh size may slightly affect the CFD results and plume tracing; however, the algorithm comparison results and the conclusion of this study are not affected by the mesh size used in the CFD simulations.

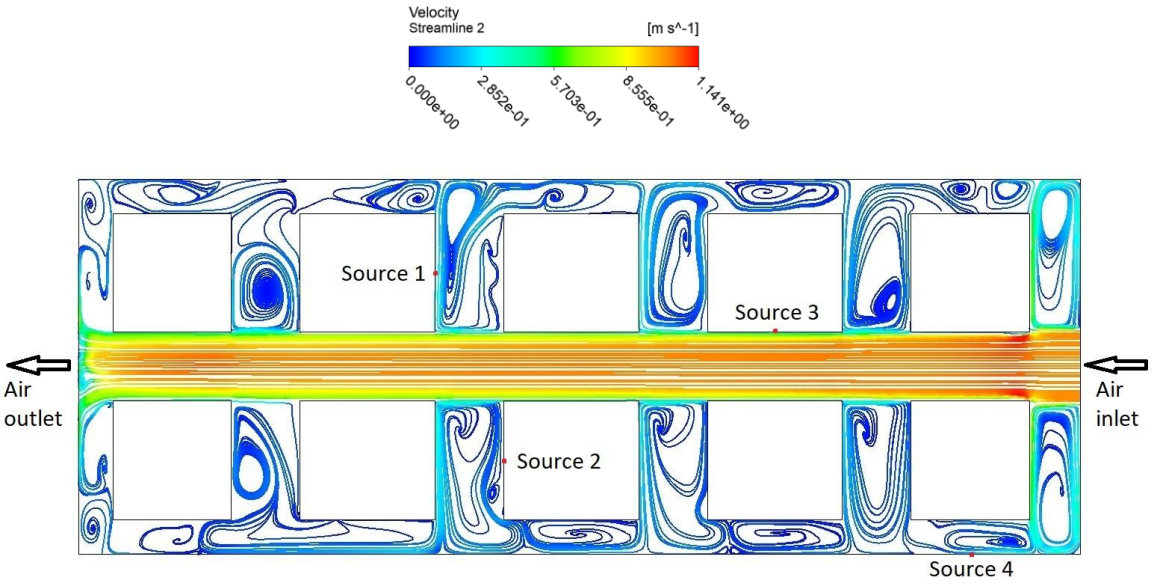
3.2.2 Simulation setup of an underground warehouse

The warehouse model in this research was developed with reference to the underground warehouse demonstrated in the previous literature [59]. The simulation is conducted in 3D framework. However, all the results are presented in 2D format for better visualisation. The geometry design of the warehouse, as well as the positions of hazardous plume sources, are shown in Figures 3.5 (a) and (b). This is a typical underground facility with an inlet boundary forcing air to enter and an air outlet boundary to maintain good ventilation (Figure 3.5 (a)). There is a door on a side wall and, since it is normally closed, it is not treated as an air outlet boundary in this research. The length of the warehouse is 78.7 m in the X direction and 29.4 m in the Y direction (Figure 3.5 (a)). The height of this

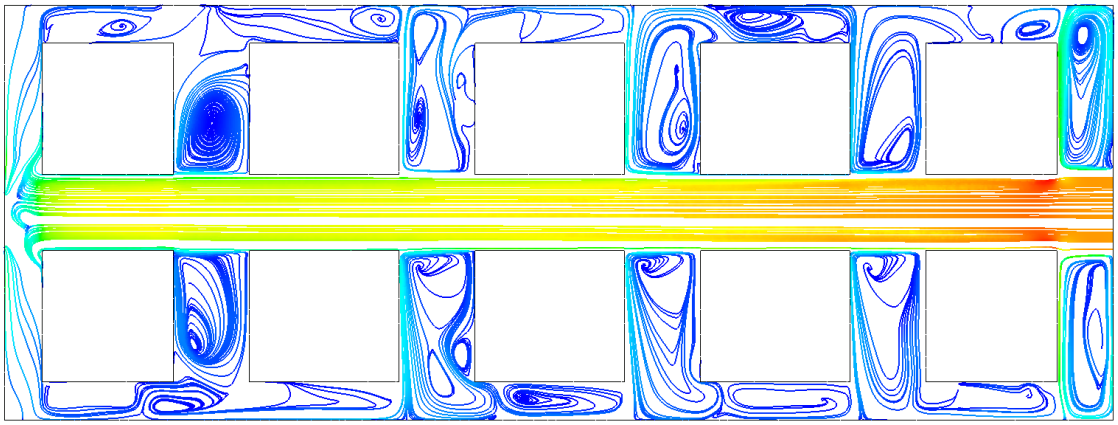
warehouse is 8 m in the Z direction (Figure 3.5 (a)). There is only one air inlet and one air outlet boundary. Fresh air comes from the air inlet boundary at 1 m/s. The area of the air inlet boundary is 36 m^2, and the area of the air outlet boundary is 12.566 m^2 (Figure 3.5 (a)). There are ten piles of cargo in total and they are assumed to be impermeable to air. Figure 3.5 (b) shows the 2D streamline at a horizontal level, 1 m above the ground. Figures 3.5 (c) and (d) show the 2D streamlines at horizontal levels, 0.3 m and 0.5 m above the ground, respectively. It can be seen from Figure 3.5 (b) that the fluid field in such a large-scale environment is complex, as many large and small recirculating airflows form between the piles of cargo. These large recirculation may significantly affect the performance of plume-tracing robots since, on the one hand, a number of plume-tracing algorithms rely on moving following the wind direction, and, on the other hand, these recalculating flows influence the chemical concentration distribution. More details regarding the geometry design can be seen in the research conducted by Stefopoulos and Damigos [59]. Sources 1, 2, 3, and 4 are



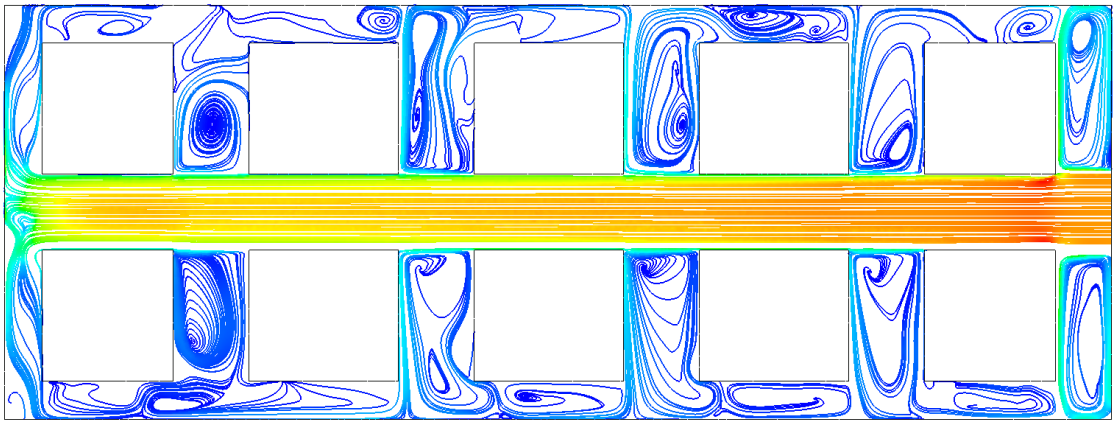
(a)



(b)



(c)



(d)

Figure 3.5: (a) The geometry design of the underground warehouse, the fluid field at the horizontal level (b) 1 m, (c) 0.3 m and (d) 0.5 m above the ground

the simulated hazardous chemical sources in Scenarios 1, 2, 3, and 4, separately.

For the coordinate system given in Figure 3.5 (a), the locations of Sources 1, 2, 3, and 4 are (28 m, 22.05 m, 0.5 m), (33.4 m, 7.35 m, 0.5 m), (54.7 m, 17.4 m, 1 m) and (70.05 m, 0 m, 1 m), respectively. Carbon monoxide comes out from the source at a mass flow rate of 0.66 g/s and a temperature of 300K. The robot was released from (1.82 m, 14.7 m) on the floor, a location near the air outlet boundary. The Reynolds number Re at the air inlet boundary, which can be calculated by Equations 3.1, 3.2 and 3.3, is 3.8×10^5. The flows near the air inlet boundary are turbulent flows. The number of the mesh nodes in the model of the underground warehouse is 1784840. A maximum-5-layer inflation is inserted for the enhancement of the mesh near the walls. A steady state k-ω SST model is used to simulate the airflows in the underground warehouse. It should be noted that, as the focus of this research is assessing the performances of different plume-tracing algorithms on wall plumes, most of the sources in this study are located on walls. When designing the cases, the sources that could be either near to or far from, or on the left side or the right hand side of the releasing location of the robot are considered.

3.3 MATLAB-based plume-tracing algorithms

A plume-tracing process can be divided into three stages for discussion: plume sensing (PS), plume tracking (PT) and source localisation (SL) [30, 32, 38, 47]. In this study, the transition between different stages is defined by the local chemical concentration. Two concentration thresholds, *threshold I* and *threshold II*, are set to distinguish different stages. All the plume-tracing algorithms with these three steps are programmed in MATLAB. Figure 3.6 explains how a robot with a three-stage plume-tracing algorithm works. When the plume-tracing robot starts working, it first detects the existence of obstacles. Having sufficient space for plume-tracing, it will then measure the local concentration. If the local

concentration is below *threshold I*, the robot is seen to be outside the plume, so the robot will move at PS stage. If the local concentration is between *threshold I* and *threshold II*, PT will be triggered. If the local concentration is higher than *threshold II*, the robot is supposed to be near the source and should get ready for source declaration, thereby moving at SL stage. For instance, in the wind-tunnel-like channel, the values of the ion mass fractions of *threshold I* and *threshold II* are set to be 10^{-5} and 0.1, respectively. *Threshold I* and *threshold II* are set to distinguish the different stages of plume tracing. When applied in other cases, the values of these two thresholds can be set case by case, according to the

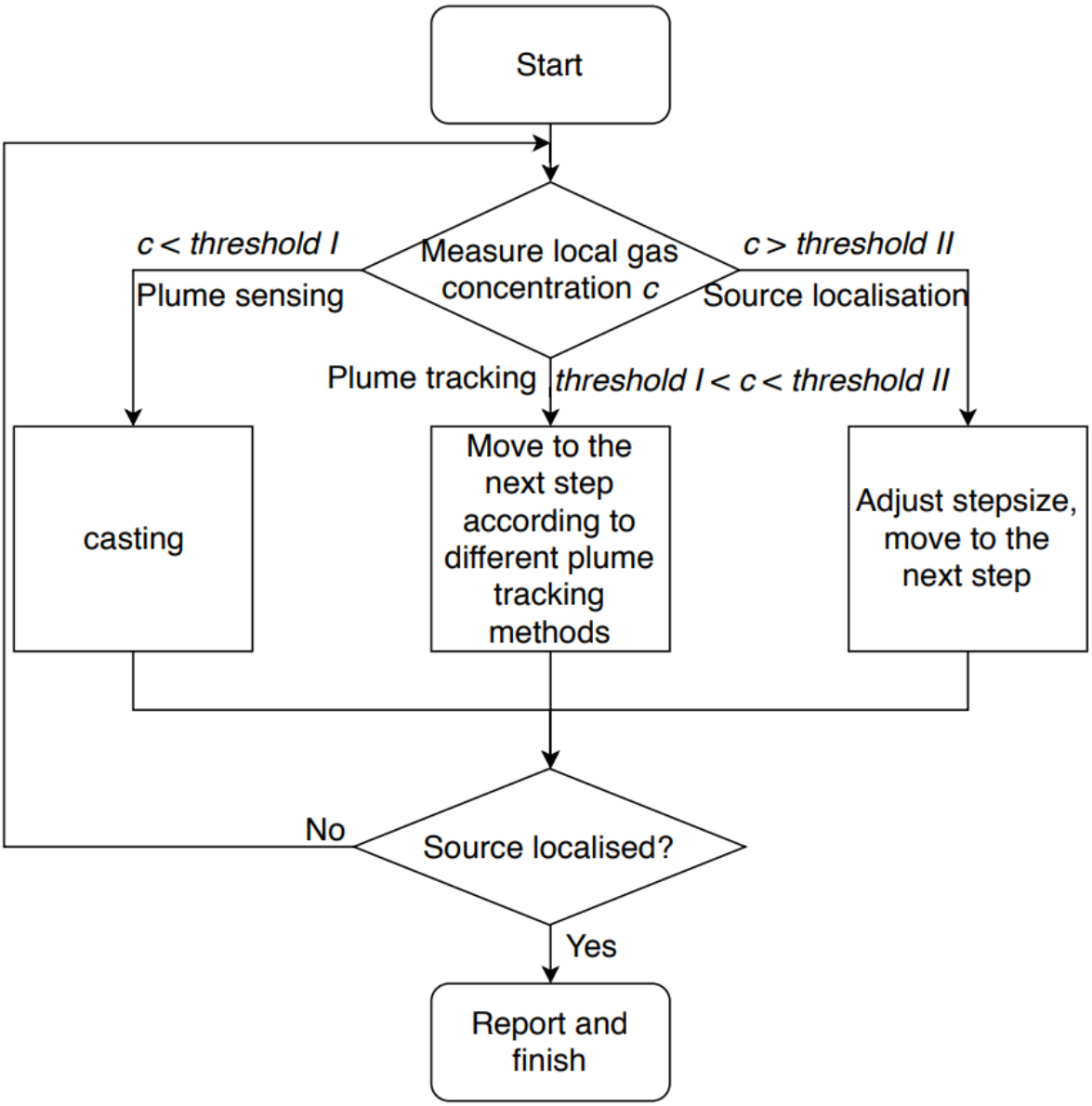


Figure 3.6: The logic flow chart of a typical plume-tracing algorithm

different chemical concentrations and environments. *Threshold I* is decided following the criteria: higher than the low limit of the range of the chemical sensor and the normal concentration of the target chemical in the environment. *Threshold II*, which distinguishes PT and SL stages, defines an area near the source and this area varies with different source strengths. Nevertheless, the robot keeps moving towards the source at either PT or SL stage, and the conclusion of the comparison of different algorithms is believed to be consistent.

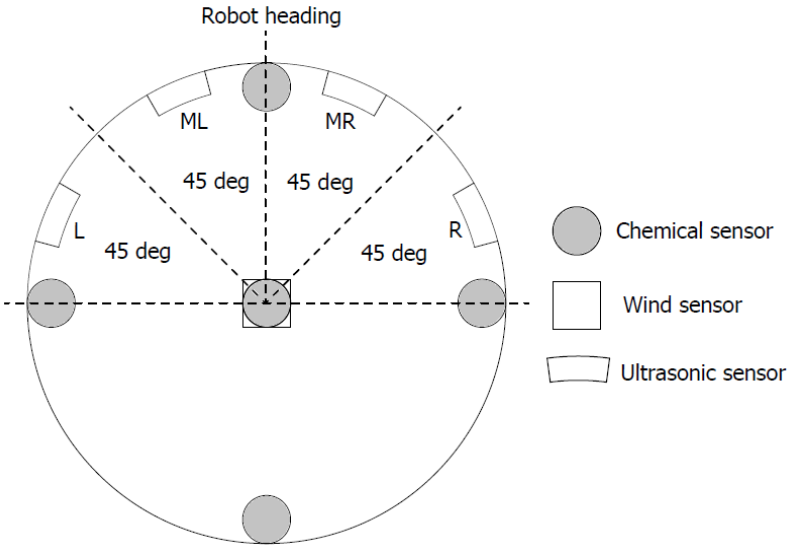
Chapter 4 Assessment of different plume-tracing algorithms in indoor plumes

The work presented in Chapter 4 is partially based on the results that have been published in [25]. In this chapter, the performance of different plume-tracing algorithms for tracing a single plume source in a wind-tunnel-like channel is investigated.

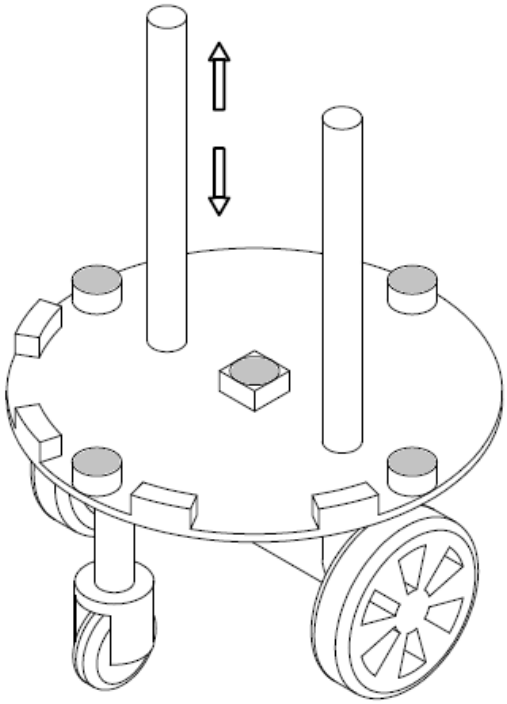
4.1 The robot

The plume-tracing robot simulated in this study is a differential drive robot equipped with chemical, wind and ultrasonic sensors and was designed in conjunction with the robots used in the studies undertaken in our wider research group by Awadalla, et al. [47] and Lu [2]. The diameter of the robot is about 10 cm, and the height of the robot is about 30 cm. As shown in Figure 4.1, two driving wheels are set on two sides of the robot to drive it the robot and two small idle wheels are set in the front and back for balance the robot. Please note that, for simplicity, the robot's control box and batteries are not included in Figure 4.1. Different from the earlier robots used by Awadalla, et al. [47] and Lu [2], which only one fixed chemical and where one fixed wind sensor was applied, the robot in this case study is equipped with multiple chemical sensors on the edge for both plume tracing and obstacle avoidance (shown in Figure 4.1). Moreover, unlike the previous robots, the locations of the sensors of the current robot are not fixed and the sensors can move vertically along two cylindrical holders (see Figure 4.1

(b)) to work at different horizontal levels. The chemical sensors are equipped to measure the local chemical concentration, and the wind sensors are equipped to measure the local wind speed and direction. Generally, plume-tracing robots move step by step, which is the definition of 'surge', because the wind and



(a)



(b)

Figure 4.1: An overview of the robot (a) top view (b) isometric view

chemical sensors need to work in a steady condition (both the robot and the sensor module plate stay still) to obtain accurate readings. In other words, at different heights, chemical concentrations and wind directions are measured after the robot stops moving horizontally and the sensor module plate stops moving vertically. As shown in Figures 4.1 (a) and (b), one chemical sensor and one wind sensor are located at the centre of the robots, and four chemical sensors are located in four directions: left, right, forward and back on the edge of the robot. Four ultrasonic sensors are used to measure the shortest distance between the robot and an obstacle in their covering sections termed sections L, ML, MR and R (illustrated in Figure 4.1). To avoid striking obstacles, a novel *concentration-distance obstacle avoidance method* was proposed using the information from the ultrasonic sensors and the chemical sensors on the edges of the robot. Here, the *responding distance* is defined as the sum of the surge distance and the diameter of the robot. If an ultrasonic sensor measures a distance longer than the *responding distance*, an obstacle is confirmed in its corresponding covering section. When the ultrasonic sensors detect the existence of an obstacle(s), the robot will tend to move away from the obstacle in order to leave sufficient space for the robot to surge following plume-tracing algorithms at the next step. For instance, when the robot is moving towards an obstacle wall in a direction normal to the wall, ultrasonic sensors will detect the wall and the chemical sensors on the edge will help navigate the robot away from the wall with respect to the concentration distribution around the robot.

The *concentration-distance obstacle avoidance method* is designed to use the concentration measured by multiple sensors. Table 4.1 shows how the robot reacts when detecting different data in different directions. In Table 4.1, using the data measured by ultrasonic sensors L, ML, MR and R, the whole obstacle avoidance method is categorised into sixteen cases. The letter Y denotes yes,

which means an obstacle is detected in the covering area. The letter N denotes that no obstacle is detected in the corresponding covering area. For example, for case 1, an obstacle is detected by Sensor L, and no obstacles are detected by the other sensors. C_f, C_r and C_l are the concentrations measured by the chemical sensor in the forward, right and left directions, respectively. Thus, for Case 8, obstacles are detected by Sensors ML and MR and no obstacles are detected by Sensors L and R. Under this circumstance, if C_r, which is the concentration measured by the chemical sensor on the right, is higher than that measured by all other sensors, the robot will turn 90° to the right and surge for a long distance. Meanwhile if C_l is the highest among all the reading, the robot will turn 90° to the left and surge for a long distance. Otherwise the robot will turn back and surge for a long distance.

Table 4.1: Algorithm of the *concentration-distance obstacle avoidance method*

Case No.	Sensor L	Sensor ML	Sensor MR	Sensor R	Condition	Action
1	Y	N	N	N	All	Turn 60° to the right and surge for a long distance
2	N	Y	N	N	All	Turn 90° to the right and surge for a long distance
3	N	N	Y	N	All	Turn 90° to the left and surge for a long distance
4	N	N	N	Y	All	Turn 60° to the left and surge for a long distance
5	Y	Y	N	N	All	Turn 90° to the right and surge for a long distance
6	Y	N	Y	N	C_f=Max	Turn 45° to the left and surge for a long distance
					Other	Turn 90° to the right and surge for a long distance
7	Y	N	N	Y	All	Surge forward for a long

						distance
8	N	Y	Y	N	C_r=Max	Turn 90° to the right and surge for a long distance
					C_l=Max	Turn 90° to the left and surge for a long distance
					Other	Turn back and surge for a long distance
9	N	Y	N	Y	C_f=Max	Turn 45° to the right and surge for a long distance
					Other	Turn 90° to the left and surge for a long distance
10	N	N	Y	Y	All	Turn 90° to the left and surge for a long distance
11	Y	Y	Y	N	All	Turn 135° to the right and surge for a long distance
12	Y	N	Y	Y	All	Turn 45° to the left and surge for a long distance
13	Y	Y	N	Y	All	Turn 45° to the right and surge for a long distance
14	N	Y	Y	Y	All	Turn 135° to the left and surge for a long distance
15	Y	Y	Y	Y	C_r=Max	Turn 150° to the right and surge for a long distance
					C_l=Max	Turn 150° to the left and surge for a long distance
					Other	Turn back and surge for a long distance
16	N	N	N	N	All	Follow the plume tracing algorithm

4.2 Plume-tracing algorithms

Figures 4.2 (a) and (b) show examples of how the plume-tracing robot work in two different scenarios. Please note that in all Figures 4.2, 4.5, 4.7, 4.8, 4.9 and 4.11, the background contour represents the chemical concentration distribution along the horizontal plane, where the simulation data was used by the robot at the last step. In both scenarios shown in Figure 4.2, the robot was released at a location outside of the plume. In both scenarios, as the chemical concentration at the releasing location is lower than *threshold I*, the robot began plume sensing to look for the plume. In Scenario M, as shown in Figure 4.2 (a), the robot moved left and right searching for the plume. After three steps, the chemical sensors detected that the ion concentration was higher than 10^{-5}, then the robot started plume tracking stage. The robot moved upward at a constant step length until

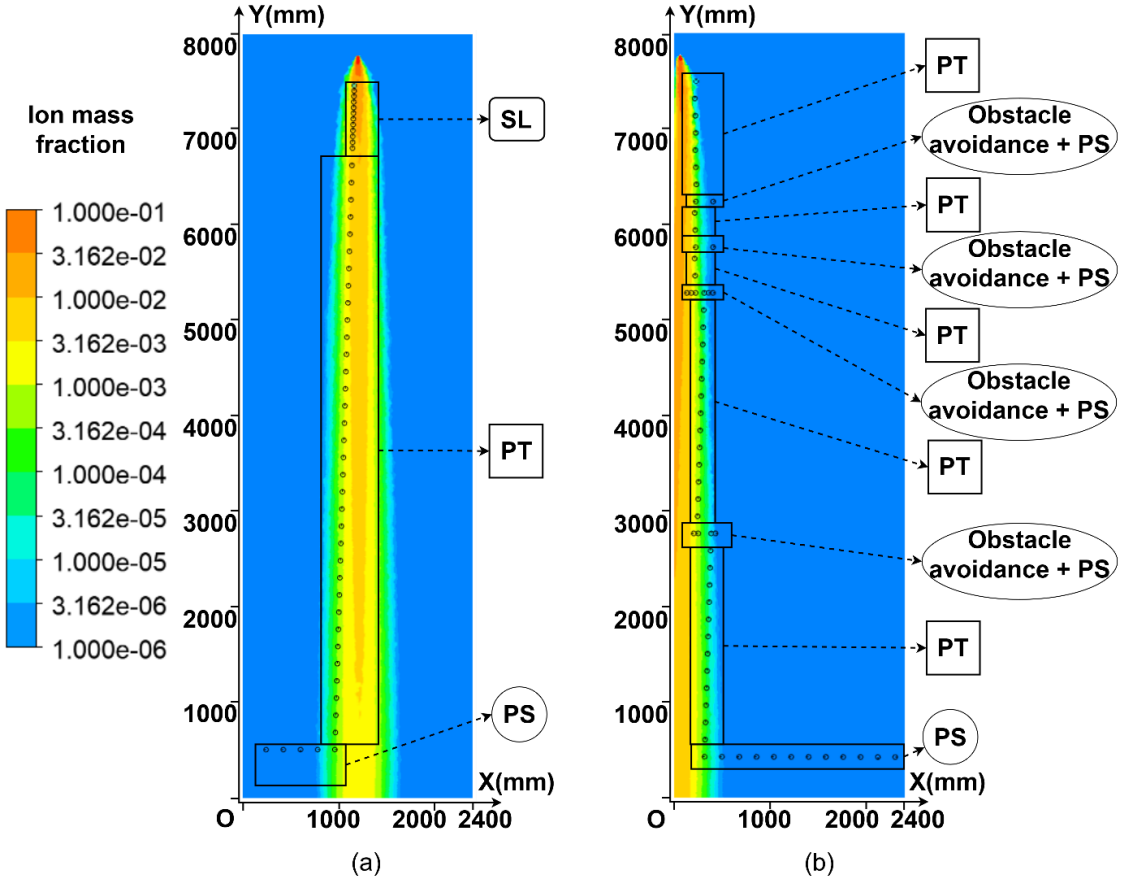


Figure 4.2: (a) An example of how the plume-tracing robot works in Scenario M, (b) An example of how the plume-tracing robot works in Scenario S

the chemical sensors detected that the ion concentration was higher than 0.1 at a location around y=5800 mm. Then the robot started source localisation stage and reduced the length of steps. After 28 steps, the distance between the robot and the source was less than 350 mm, and the robot reported that the source was detected. In Scenario S, the robot found and left the plume four times when searching because of either wind direction or obstacle avoidance and finally it ended without SL because the robot did not gain contact with the area where the chemical concentration was above *threshold II*.

4.2.1 Plume sensing

Plume sensing is the process whereby a robot that is initially located outside a plume moves to find a plume that comes from a chemical source. This process, inspired by insects, is also termed *casting*. There are two types of *casting*, *normal casting* and *special casting*, in this study that have been proposed in the previous literature. The performance of these two *casting* methods is compared by searching in both the wall plume (Scenario S) and the free stream plume (Scenario M).

4.2.1.1 *Normal casting*

This first kind of *casting* method, called *normal casting* in this chapter is the same as the *casting* method used in the experiments undertaken by Harvey, et al. [40]. During the *normal casting* process, the robot moves in a direction that is perpendicular to the wind direction and moves forth and back in a gradually increasing step length at each step. Meanwhile, the chemical sensor keeps measuring the chemical concentration and once the concentration is above *threshold I*, the robot enters a plume through one plume boundary and it will immediately stop PS and begin the next stage, either plume tracking or source localisation, depending on whether the chemical concentration is lower or higher than *threshold II*.

4.2.1.2 *Special casting*

Under the *special casting* process, the robot searches for the centreline of a plume, which is located in the middle of two well-developed boundary layers of a plume [17, 18]. Differing from *normal casting*, in *special casting* the robot keeps moving straight, aiming to find the next boundary of the plume after entering the plume through one plume boundary. As long as the concentration is above *threshold I*, the robot continues to move across the wind and the plume until either the concentration is again lower than *threshold I* or the robot meets a wall, meaning that the other boundary of the plume is found. Then the robot moves back to the centre between the two boundaries and starts the next stage. The advantage of the *special casting* is that the initial position of the next stage to be commenced will be near the centreline of the plume and theoretically the possibility of leaving the plume when surging upwind is lower than that when moving near the plume boundary, which is normally found in the case of *normal casting*.

4.2.2 Plume tracking

Plume tracking is the process by which the robot surges within the plume after finding it and normally takes a longer time, thereby influencing the efficiency significantly (see Figure 4.2). All the PT methods compared in this part of the study were previously presented either in experiments or in a simulation framework and were already proved to be applicable (see Chapter 1). The distance that the robot surged at every step was constant.

4.2.2.1 *Surge anemotaxis*

Surge anemotaxis is one of the simplest of the implemented algorithms, and it was identified as performing well in experiments [40]. Here, *surge* is defined as moving step by step. When the local chemical concentration is above *threshold I*, the robot surges towards the wind direction measured by the wind sensor. Under

surge anemotaxis, the robot continuously adjusts its direction to ensure that it is orienting to the wind with every step and then surges upwind.

4.2.2.2 *Chemotaxis*

Chemotaxis is defined as surging towards the area with the highest chemical concentration regardless of wind direction; i.e. the robot will not surge based on wind direction but will surge based on chemical concentration. In this study, there are four chemical sensors located on the edge of the robot (explained in Section 4.1) and the robot moves in the direction of the sensor that reports the highest chemical concentration among four. For example, when the robot is moving, if the sensors detect the highest concentration in the forward direction, the robot will surge forward.

4.2.2.3 *Zigzags*

The strategy of this algorithm was inspired by the moving behaviours of Dung beetles. As its name indicates, the trajectories of the robot using this algorithm are like zigzags. The overall strategy is surging upwind; however, differing from *surge anemotaxis*, after turning to face the wind, the robot continues to turn at a certain angle and then surges. The turning angle is constant in most cases. In this study, 45° was selected.

4.2.2.4 *Pseudo gradient-based algorithm*

The idea of the *pseudo gradient-based algorithm* was first presented by Lilienthal and Duckett [60], and was developed into a formal algorithm and tested in the experiments undertaken by Neumann, et al. [35]. The strategy of this specially designed PT method is actually a combination of *surge anemotaxis* and *chemotaxis*. Under the *pseudo gradient-based algorithm*, after facing upwind, the robot continues to turn β towards the area with a higher chemical concentration (as illustrated in Figure 4.3). In general, β can be calculated by using Equations 4.1 and 4.2:

$$\beta = \begin{cases} 0, & \mu = 0.5 \\ (1-2\mu) \times 90°, & (turn\ left), \quad \mu < 0.5 \\ (2\mu-1) \times 90°, & (turn\ right), \quad \mu > 0.5 \end{cases} \qquad (4.1)$$

$$\mu = \frac{C_L}{C_L + C_R} \qquad (4.2)$$

where μ is a coefficient that is used to determine the value of β. In previous experiments, C_L and C_R were measured by moving left and right but in this study they are measured by taking advantage of multiple chemical sensors. C_L is the chemical concentration measured by the chemical sensor on the left hand side, and C_R is measured by the one on the right hand side.

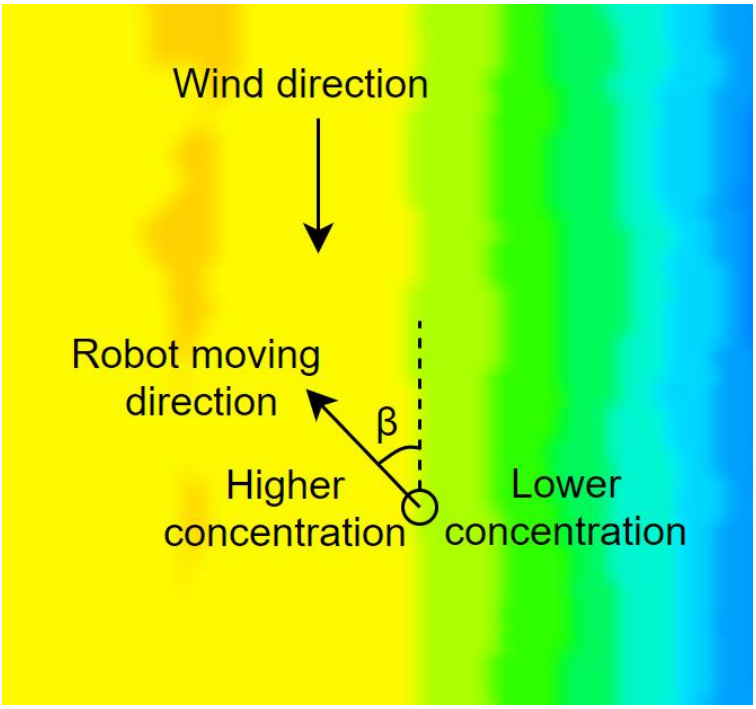


Figure 4.3: Schematic diagram of the working principle of *pseudo gradient-based algorithm*

4.2.3 Source localisation

Source localisation is the process of identifying and declaring the location of the source after the robot enters the plume zone where the chemical concentration is higher than *threshold II*. During the source localisation process, the robot can move either following *constant stepsize* or *variable stepsizes*. In this part of the

study, the *stepsize* is defined as the distance that the robot surges at each step.

4.2.3.1 *Constant stepsize*

For the *constant stepsize* SL, the robot moves in a constant *stepsize* that can be the same as that in the plume tracking stage. One drawback of the *constant stepsize* source localisation is that, when the *stepsize* is large, the robot may surge out of the plume when searching near the chemical source.

4.2.3.2 *Variable stepsize*

To solve the problem of the robot surging out of the plume near the chemical source, a novel source localisation method with a *variable stepsize* strategy was proposed by Lu [2]. During this *variable stepsize* SL, the higher the chemical concentration is, the shorter the distance the robot surges at each step. An equation between the chemical concentration and surge distance was adopted as [2]:

$$XY = C \qquad (4.6)$$

where X is the chemical concentration that is normalised by the highest concentration, i.e. the concentration at the source. Therefore, the X value is normalised within the range from 0 to 1. Y is the surge *stepsize*, and C is a constant value that can vary according to different environments. Following the work of Lu [2], $C = 5$ was selected.

4.2.4 Table of algorithms

One of the main objectives of this research is to assess the performance of the algorithms with different PS, PT or SL stages when tracing the sources of two plumes confined in a channel (Scenario S and Scenario M). Therefore, a total of 16 different combined plume-tracing algorithms are compared in this study. Table 4.2 shows a list of these 16 combined algorithms.

Table 4.2: Different plume-tracing algorithms and their origins

Algorithm No.	Plume Sensing	Plume Tracking	Source Localisation
1	*Normal casting* [40]	*Surge anemotaxis* [40]	*Constant stepsize* [2]
2	*Normal casting*	*Surge anemotaxis*	*Variable stepsize* [2]
3	*Normal casting*	*Chemotaxis* [32]	*Constant stepsize*
4	*Normal casting*	*Chemotaxis*	*Variable stepsize*
5	*Normal casting*	*Zigzags* [35, 40]	*Constant stepsize*
6	*Normal casting*	*Zigzags*	*Variable stepsize*
7	*Normal casting*	*Pseudo gradient-based algorithm* [35]	*Constant stepsize*
8	*Normal casting*	*Pseudo gradient-based algorithm*	*Variable stepsize*
9	*Special casting* [17]	*Surge anemotaxis*	*Constant stepsize*
10	*Special casting*	*Surge anemotaxis*	*Variable stepsize*
11	*Special casting*	*Chemotaxis*	*Constant stepsize*
12	*Special casting*	*Chemotaxis*	*Variable stepsize*
13	*Special casting*	*Zigzags*	*Constant stepsize*
14	*Special casting*	*Zigzags*	*Variable stepsize*
15	*Special casting*	*Pseudo gradient-based algorithm*	*Constant stepsize*
16	*Special casting*	*Pseudo gradient-based algorithm*	*Variable stepsize*

4.2.5 *Stepsize*

The *stepsize* in the experiments carried out by Harvey, et al. [40] was 18cm. In the current study, four different *stepsizes*: 9cm, 13cm, 18cm and 23cm were tested and compared in two different scenarios using Algorithm 1 (please see Table 4.2 for the details of Algorithm 1). For example, the robots in both Figures 4.2 (a) and (b) have a *stepsize* of 18 cm. It can be seen in Figure 4.4 that the total number of steps tends to decrease when the *stepsize* increases in Scenario M. In Scenario S, the total number of steps decreases when the *stepsize* increases from 9 cm to 13 cm, however, there are no obvious advantages when the *stepsize* increases from 13 cm to 18 cm. An accident happened when the *stepsize* increased to 23 cm, as Figure 4.5 shows: the robot became trapped and failed to approach the source because of the increasing *responding distance* for obstacle avoidance. To balance the performance of the robots in the two scenarios, an 18cm *stepsize* is selected. In this study, the robot is assumed to surge once at each time step, as per the work undertaken by Lu [2]. Therefore, the lower number of steps indicates the lower time spent, representing higher efficiency.

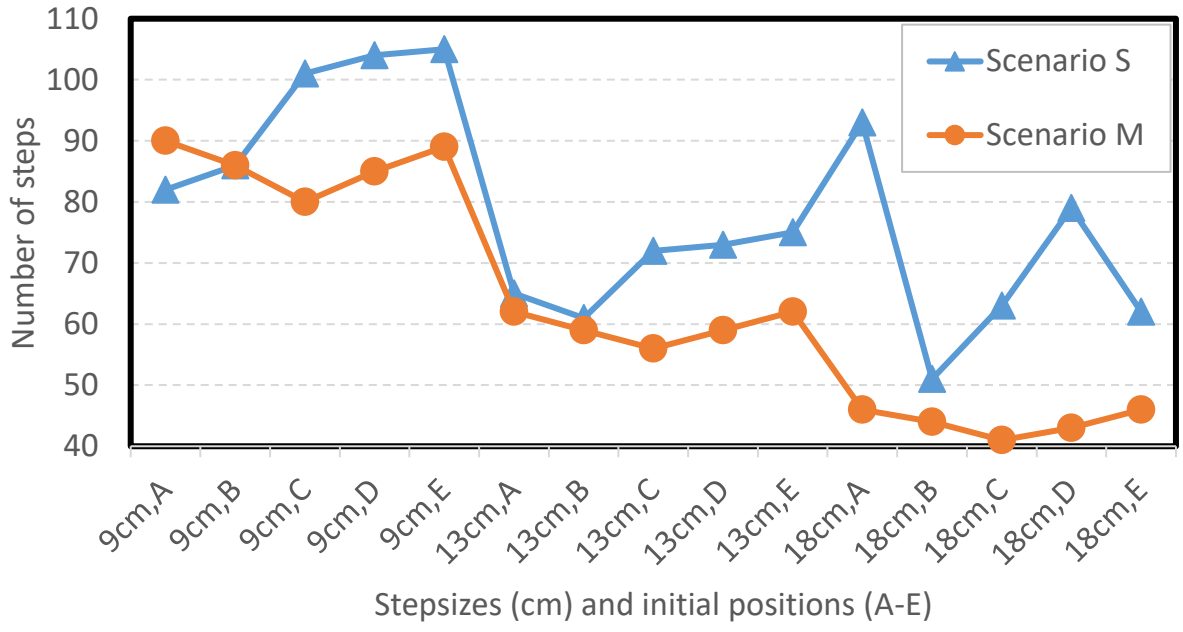


Figure 4.4: The total numbers of steps in two scenarios with different *stepsize*, searching with Algorithm 1

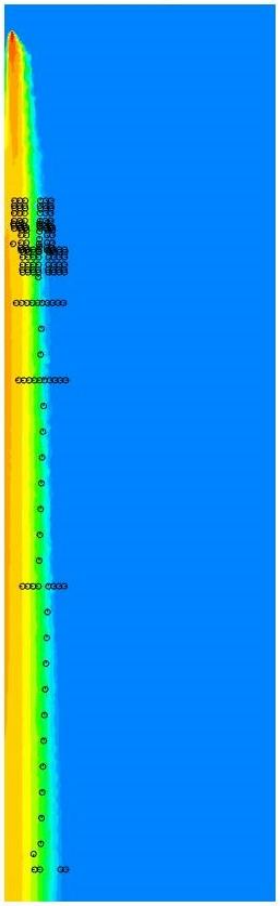
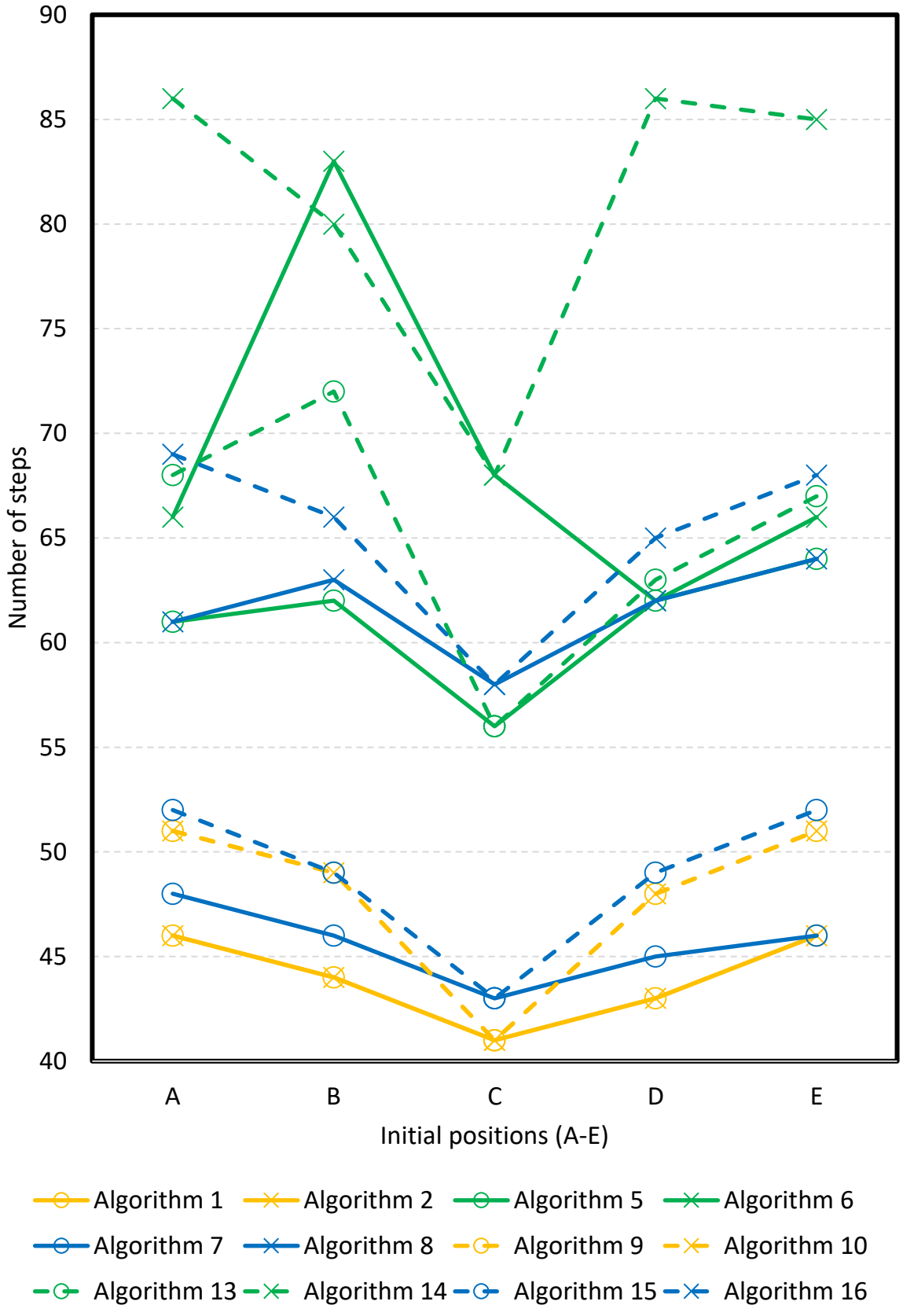


Figure 4.5: The trajectory of the robot in Scenario S when the *stepsize* is 23 cm, searching with Algorithm 1

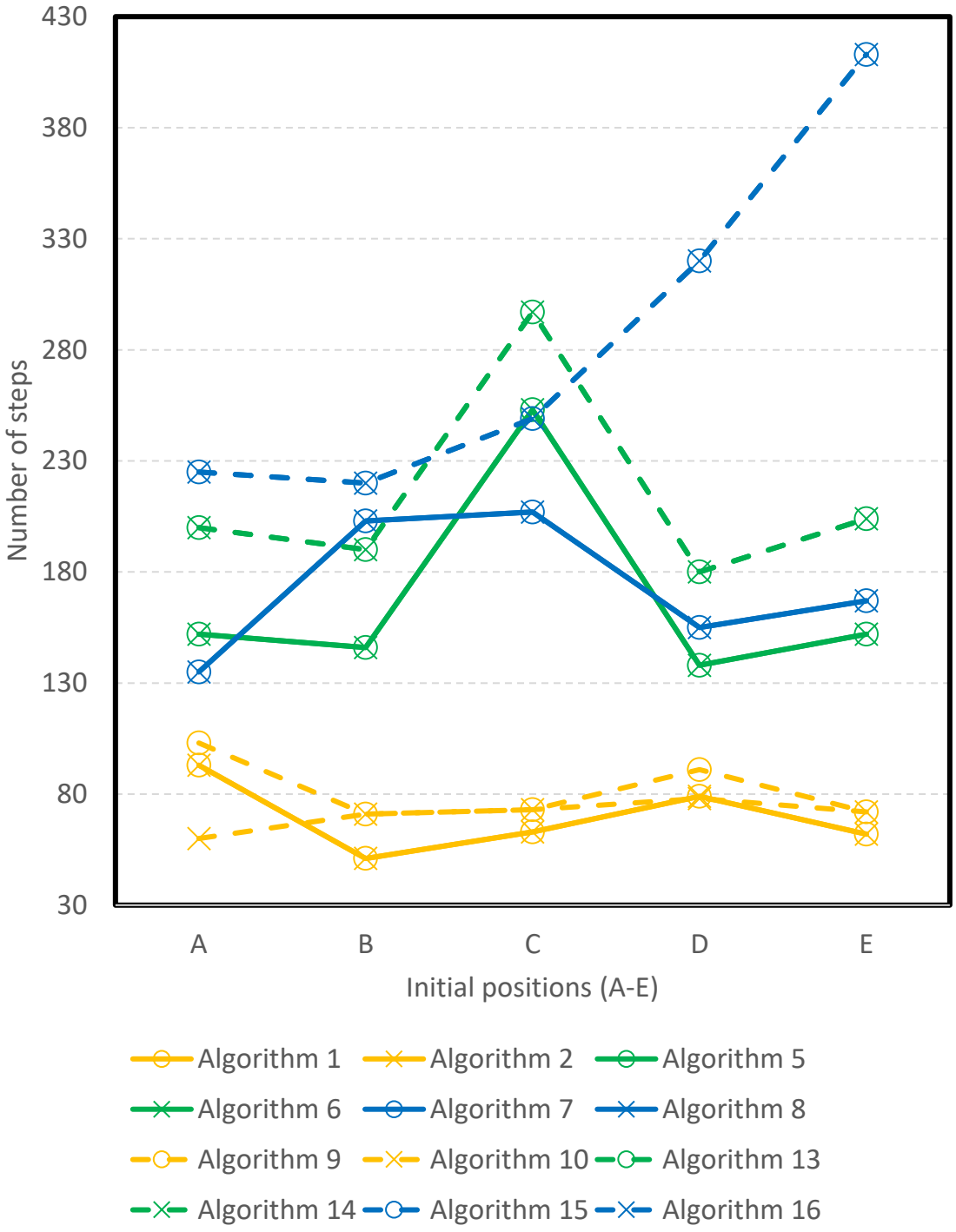
4.3 Simulation results and comparison

In this section, an overall view of all results is presented, and subsequently an analysis is offered as to whether initial locations matter and comparisons are made across the different PS, PT and SL methods. The results are shown in Figures 4.6 (a) and (b). These indicate that the robot's performances differ in different scenarios, PS, PT, SL stages and initial locations. The total number for steps of each algorithm with *chemotaxis* for PT stage is not shown because of its unsatisfactory performance (see Section 4.4.3). It costs the robot considerably more steps when searching in Scenario S, indicating that these algorithms need to be improved to fit the condition that the source is located near a wall. In Figure

4.6, the different line style indicates different SL stages. Solid lines represent the



(a)



(b)

Note: Algorithms refer to Table 4.1, solid line – *normal casting*, dash line – *special casting*, yellow – *surge anemotaxis*, green – *zigzags*, blue – *pseudo gradient-based algorithm*, O – *constant stepsize*, X - *variable stepsize*

Figure 4.6: (a) Statistical results of the number of steps in Scenario M, (b) Statistical results of the number of steps in Scenario S

algorithms with *normal casting* while dash lines represent the algorithm with

special casting. Similarly, the colour green represents the *zigzags*, and yellow represents the *surge anemotaxis* and blue means the *pseudo gradient-based algorithm* for PT stage. Also, for the markers, 'O' means the *constant stepsize* and 'X' means the *variable stepsize* for SL stage. In Figures 4.7, 4.8, 4.9 and 4.11, 'algorithm', 'scenario', 'corresponding method', 'initial location' and 'total number of steps' are labelled following every Figure in this chapter. For instance, the trajectory of the robot coded with Algorithm 5 where *normal casting* is applied and released from position E in Scenario M is shown in Figure 4.7 (a), and the total number of steps is 64. More results can be seen in Appendix A of this thesis.

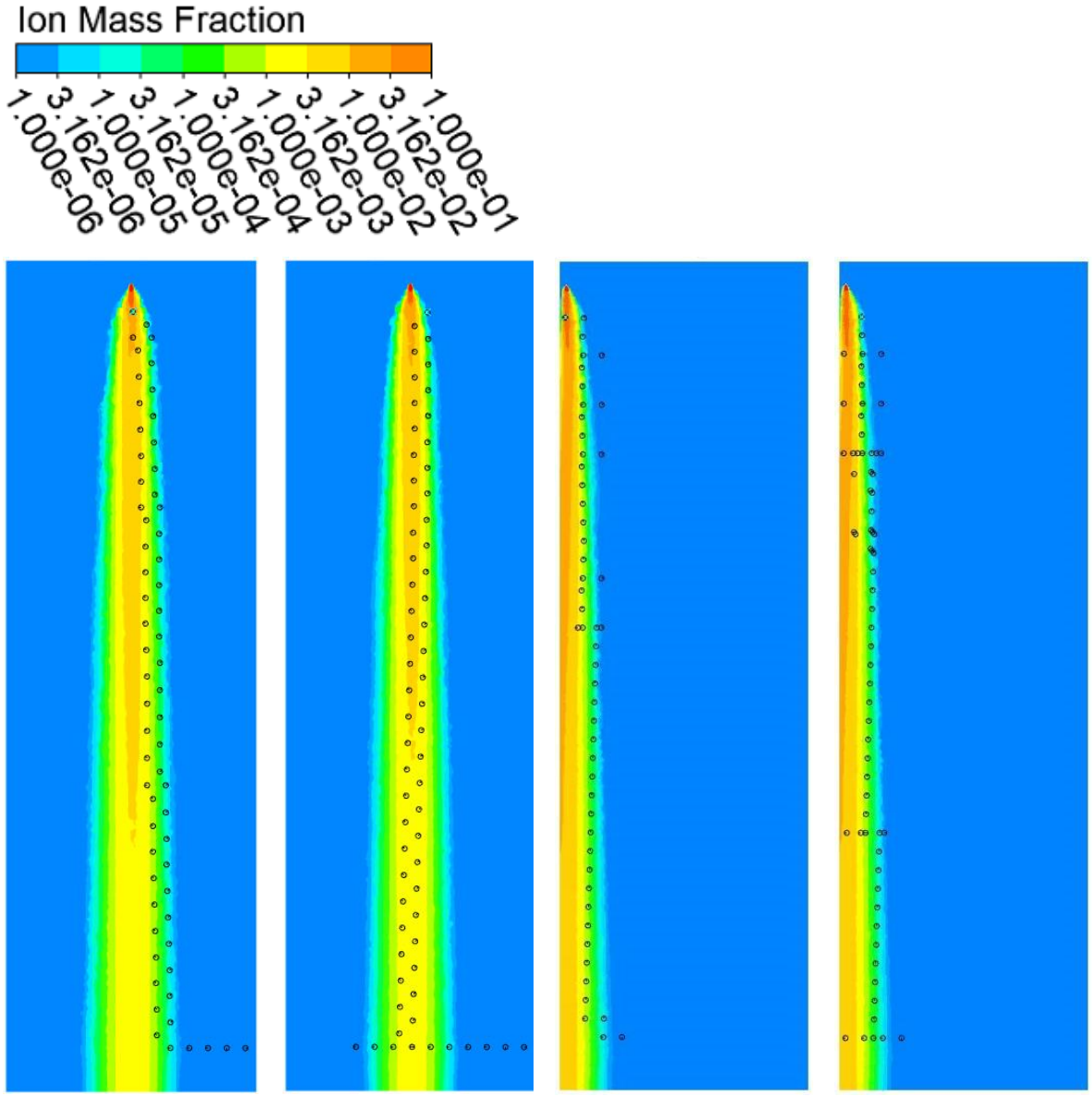
4.3.1 Initial location matters?

The question as to whether or not there are advantageous initial locations when releasing a plume-tracing robot in an environment for which the robot is without prior knowledge was proposed by Lu [2] and the simulation results indicated that there are no obvious advantageous initial positions for higher efficiency when releasing a plume-tracing robot in a corridor in a given office-like indoor environment [2]. However, the simulations conducted by Lu [2] were based on only one algorithm. To discuss this problem further, the robot was released from five different initial locations when testing different algorithms in the two scenarios in this study. (These five initial locations are illustrated in Figure 3.3) From Figures 4.6 (a) and (b), it is found that the total numbers of steps varies depending on different initial locations. In Scenario M, it can be seen that for most algorithms, the robots released from location C, which is inside the plume, have a lower total number of steps. However, in Scenario S, even though location A is inside the plume, the robot released from location A shows no advantages in terms of the total number of steps. Therefore, for an indoor environment that is

similar to the one given in this case study, it may be concluded that for a plume that is away from walls, it is better to release the plume-tracing robot from a position that is inside the plume if possible. However, when there is a wall plume, statistical results in the study indicate that the robot is not better being placed initially within the plume when searching with either the algorithms or the obstacle avoidance method mentioned in this part of the study.

4.3.2 Comparison of PS methods

According to the testing results, both *normal casting* and *special casting* prove to be capable of navigating the robot outside the plume to find the plume. Figure 4.7 shows the trajectories of the robot in four selected cases; two cases with *normal casting* and two cases with *special casting*. For the *normal casting* used in both scenarios (see Figures 4.7 (a) and (c)), once the robot found the plume, it immediately tracked within the plume. While for the *special casting* used in both scenarios (Figures 4.7 (b) and (d)), the robot continued moving forward after entering the plume until it either moved out of the plume (Figure 4.7 (b)) or reached the wall (Figure 4.7 (d)), then it moved back to the centre of the plume and started PT. The results shown in Figure 4.7 indicate that a robot with *normal casting* spends less time than that with *special casting* in total. *Special casting* did help the robot find the centreline of the plume, but this did not reduce the total time cost. To summarise, the *normal casting* is selected in terms of the total time cost.



(a) Algorithm 5, Scenario M, *normal casting*, released from location E, step number 64

(b) Algorithm 13, Scenario M, *special casting*, released from location E, step number 67

(c) Algorithm 1, Scenario S, *normal casting*, released from location B, step number 51

(d) Algorithm 9, Scenario S, *special casting*, released from location B, step number 71

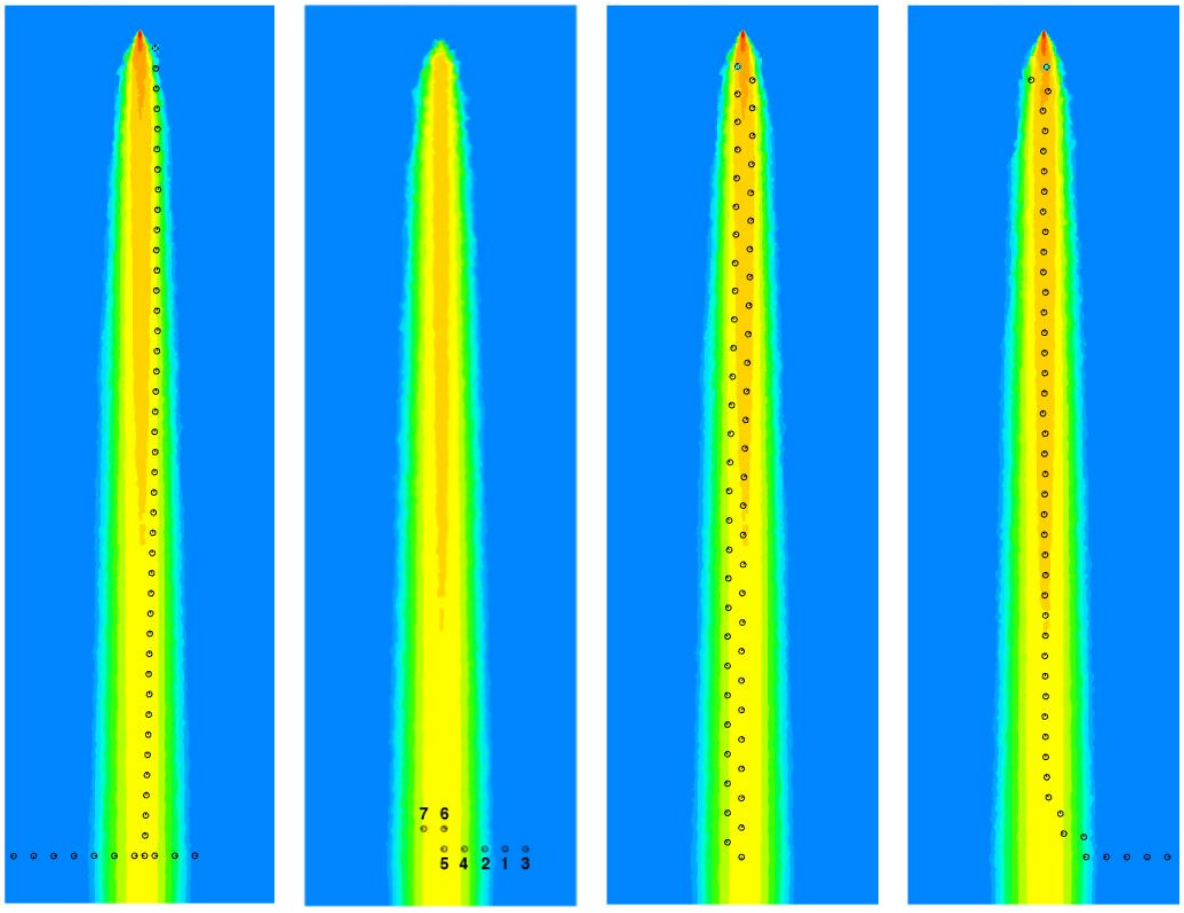
Figure 4.7: Selected cases of different PS

4.3.3 Comparison of PT methods

In total, four plume tracking methods were tested and compared in this part of

study. Among these four methods, the *surge anemotaxis*, *zigzags* and *pseudo gradient-based algorithm* proved to be capable of navigating the robot to approach the source of the plume (see Figure 4.8). For *surge anemotaxis* as shown in Figure 4.8 (a), after PS, the robot found the centreline of the plume and surged upwind for plume tracking. The trajectory of the robot was basically a straight line. Figures 4.8 (b) and (f) show that for a single robot that is coded with *chemotaxis* (illustrated in Section 4.2.2), the robot was trapped in the downstream of the plume after the PS stage. In all the tests with *chemotaxis*, regardless of PS and SL, initial locations and scenarios, the robot failed in searching. A further examination of the trajectories of the robot with *chemotaxis* shown in Figure 4.8 (b) finds that the robot entered the plume at point 4, then it moved towards point 5 and then point 6 following the measured highest chemical concentration. At point 6, the robot kept moving a step further to step 7 as the front chemical sensor detected the highest concentration. From step 6 to step 7, however, the robot passed the region where the highest concentration zone was located. At step 7, as the highest concentration was found at the back of the robot, the robot moved back to the location of step 6. The robot trajectory started repeating between step 6 and 7, and the robot was therefore trapped. The *chemotaxis* failed in Scenario S as shown in Figure 4.8 (f) as well. For the *zigzags* as shown in Figure 4.8 (c), the trajectory of the robot is a zigzag in Scenario M. However, in Scenario S (see Figure 4.8 (g)), the trajectory is initially a zigzag but then became complex after z=5000 mm. This can be attributed to the fact that the distance between the robot and the wall was getting closer after z=5000 mm, and so obstacle avoidance was frequently triggered, thereby increasing the number of steps. For the *pseudo gradient-based algorithm,* in Scenario M, the robot moved along an approximately straight line first of all and then a zigzag after z=7000 mm since the difference between the concentration

measured by the chemical sensors on the left and right hand side is more significant. In Scenario S, since the concentration near the wall is higher, the robot surged towards the wall in the first instance and then surged away from the wall because of obstacle avoidance. Hence, the total number of steps of the robots coded with the *pseudo gradient-based algorithm* in the scenario were high. In summary, considering the stability and total time cost, *surge anemotaxis* performs better than the other methods in these two scenarios.

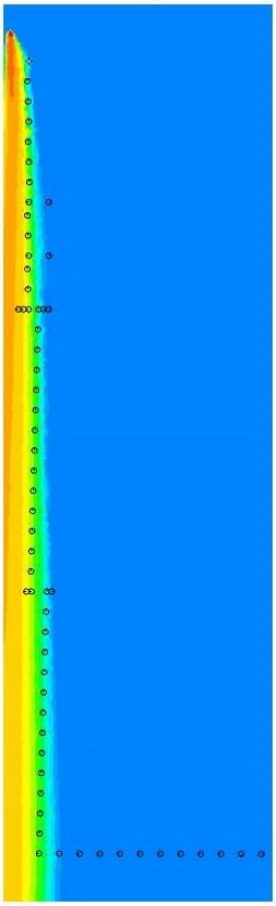


(a) Algorithm 9, Scenario M, *surge anemotaxis*, released from location A, step number 51

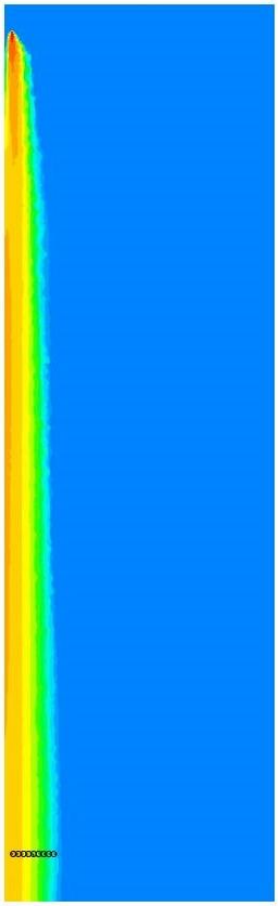
(b) Algorithm 3, Scenario M, *chemotaxis*, released from location D, failed

(c) Algorithm 5, Scenario M, *zigzags*, released from location C, step number 56

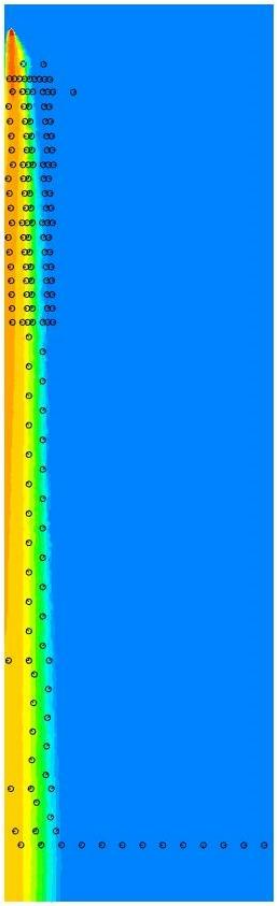
(d) Algorithm 7, Scenario M, *pseudo gradient-based algorithm*, released from location E, step number 46



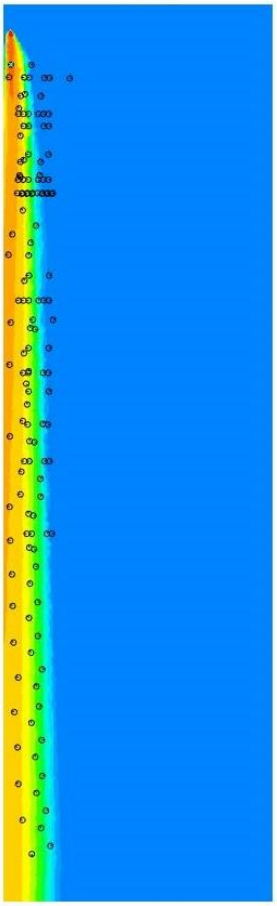
(e) Algorithm 2, Scenario S, *surge anemotaxis*, released from location E, step number 62



(f) Algorithm 4, Scenario S, *chemotaxis*, released from location A, failed



(g) Algorithm 13, Scenario S, *zigzags*, released from location D, step number 180



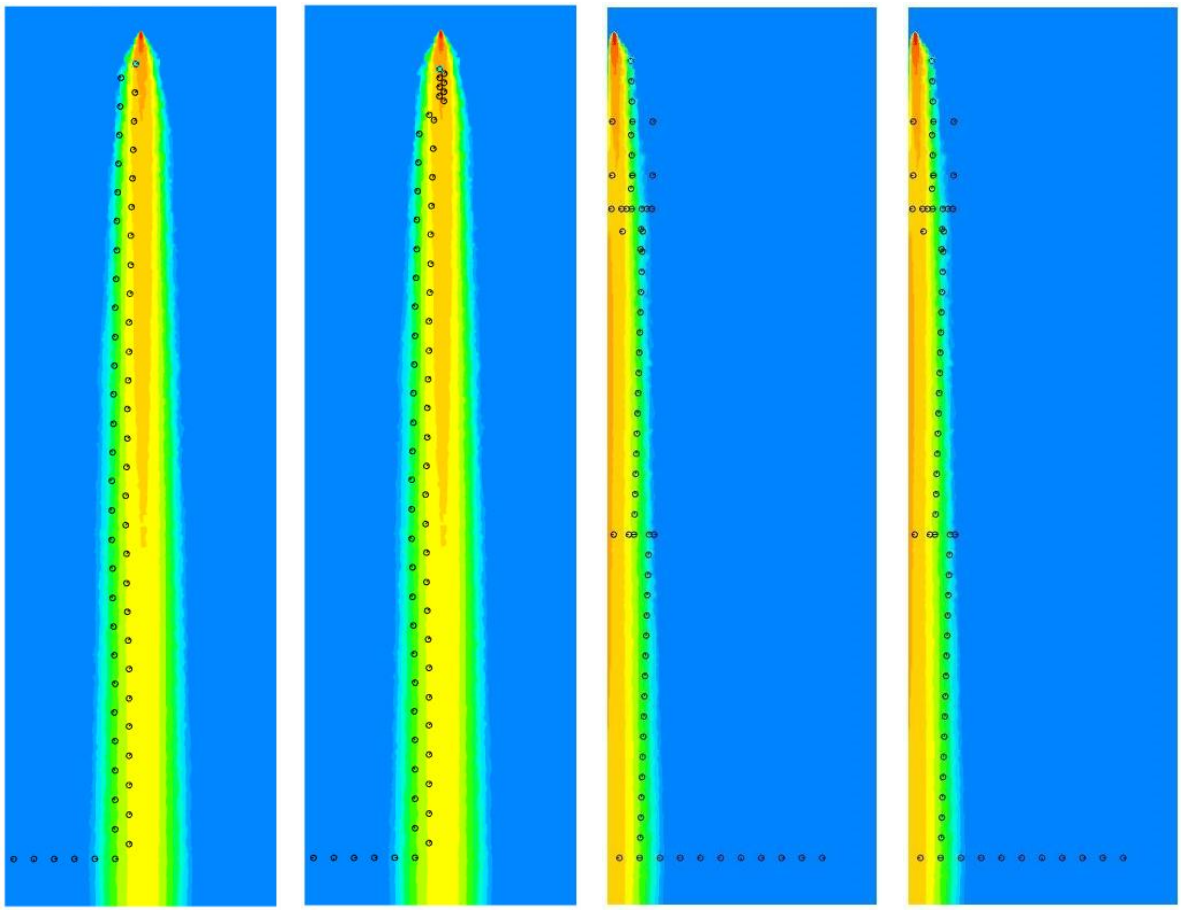
(h) Algorithm 7, Scenario S, *pseudo gradient-based algorithm*, released from location A, step number 135

Figure 4.8: Selected cases of different PT

4.3.4 Comparison of SL methods

Two types of SL, a *constant stepsize* method and a *variable stepsize* method, were tested in this study. Compared with the trajectories of the robot with *constant stepsize* shown in Figure 4.9 (a), the robot with *variable stepsize* (Figure 4.9 (b)) moved in more numerous but smaller steps when approaching the source in Scenario M. When getting closer to the source, the *stepsize* decreased drastically. It did not perform as expected (the *stepsize* gradually reducing) as per the simulation conducted by Lu [2], whose scenario was a larger indoor environment.

In Scenario S, there are no significant differences of trajectories and numbers of steps between the robots with *constant stepsize* and *variable stepsize* SL (see Figures 4.9 (c), (d) and Figure 4.6 (b)). This is because the motion of the robot is dominated by the obstacle avoidance motions when it moves close to the source in Scenario S. This observation is also confirmed in Figure 4.8. Generally, the *variable stepsize* SL stage fits a large indoor environment such as the one presented by Lu [2]. However, *constant stepsize* works better in the relatively smaller domain presented in this study. As a result, for the localisation of the indoor plume sources in this study, *constant stepsize* is selected for source localisation.



(a) Algorithm 5, Scenario M, *constant stepsize*, released from location A, step number 61

(b) Algorithm 6, Scenario M, *variable stepsize*, released from location A, step number 66

(c) Algorithm 9, Scenario S, *constant stepsize*, released from location C, step number 73

(d) Algorithm 10, Scenario S, *variable stepsize*, released from location C, step number 73

Figure 4.9: Selected cases of different SL

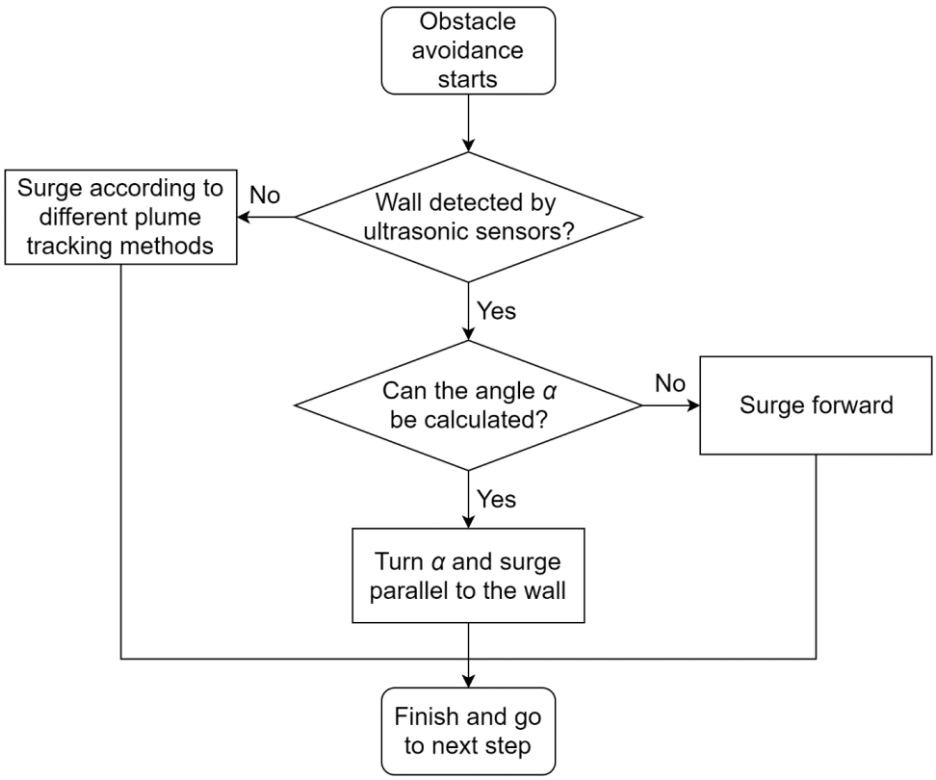
4.3.5 *Vallumtaxis*

Normally there is no prior knowledge about the possible location of the source when releasing a robot to undertake chemical plume tracing and subsequent source localisation tasks. As a result, a more comprehensive plume-tracing algorithm that is capable of navigating a robot in searching in indoor environments, where the sources are located on or near a wall, is needed, especially when an urgent situation, which needs to be solved as soon as possible, emerges. The investigations and findings presented in the previous sections indicate that *normal casting*, *surge anemotaxis* and *constant stepsize* are the best for PS, PT and SL, respectively. Therefore, a combination of all of them, Algorithm 1, was proved to perform the best and is selected for further development into a novel algorithm with higher efficiency. From the trajectories of the robot shown in Figure 4.9 (a), it is concluded that a robot navigated by Algorithm 1 approaches the source in an almost straight-line trajectory. This makes the total search distance and time cost already very low. However, in Scenario S, it can be seen that the *concentration-distance obstacle avoidance method* (explained in Section 4.1) slows down the robot by making it repeat PS (see Figure 4.8 (g)). In fact, the plume goes along the wall in Scenario S so getting away from the wall equates to leaving the plume. In this case, to achieve a higher efficiency, an algorithm called *vallumtaxis*, which is based on Algorithm 1, but with an *along-wall obstacle avoidance method*, was proposed in this study. Obstacle avoidance methods using multiple ultrasonic sensors have already been applied in many studies [2, 15, 47, 49]. Previously in this research, when detecting a wall, a robot would try to move away from the wall by moving back. Due to the characteristics of wall plumes, it was found in this study that it is more efficient for a plume-tracing robot to move along the wall when wall plumes are detected, and this was realised by the proposed *along-wall obstacle avoidance method*, which was

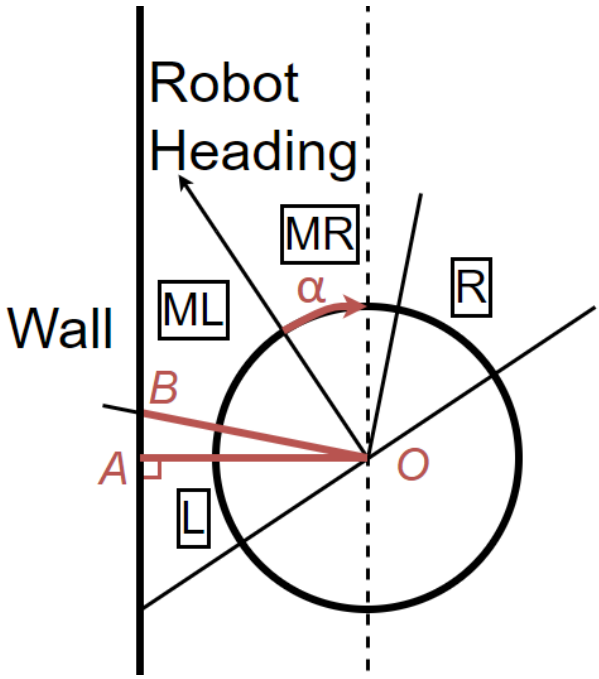
developed with the help of multiple ultrasonic sensors. The searching strategy of moving along a wall or obstacle has already been presented in the literature (see Chapters 1 and 2). The distances measured by ultrasonic sensors were used to ensure the robot moves parallel to the wall, which is the fundamental principle of this *along-wall obstacle avoidance method*. An overall logic flow of the *along-wall obstacle method* is presented in Figure 4.10 (a). Figure 4.10 (b) represents how the robot turns and surges along a wall when the wall is detected. The turning angle α is governed by the equation:

$$\alpha = \frac{\pi}{4} - \arccos\left(\frac{OA}{OB}\right) \qquad (4.7)$$

where OA is the distance between the wall and the centre of the robot measured by the ultrasonic sensor that covers the L section, and OB is the distance between Point B and the centre of the robot. Here, OB is the sum of the diameter of the robot and the distance (mentioned in Section 4.1) measured by the ultrasonic sensor that covers the ML section (see Figure 4.10 (b)). For every step



(a)



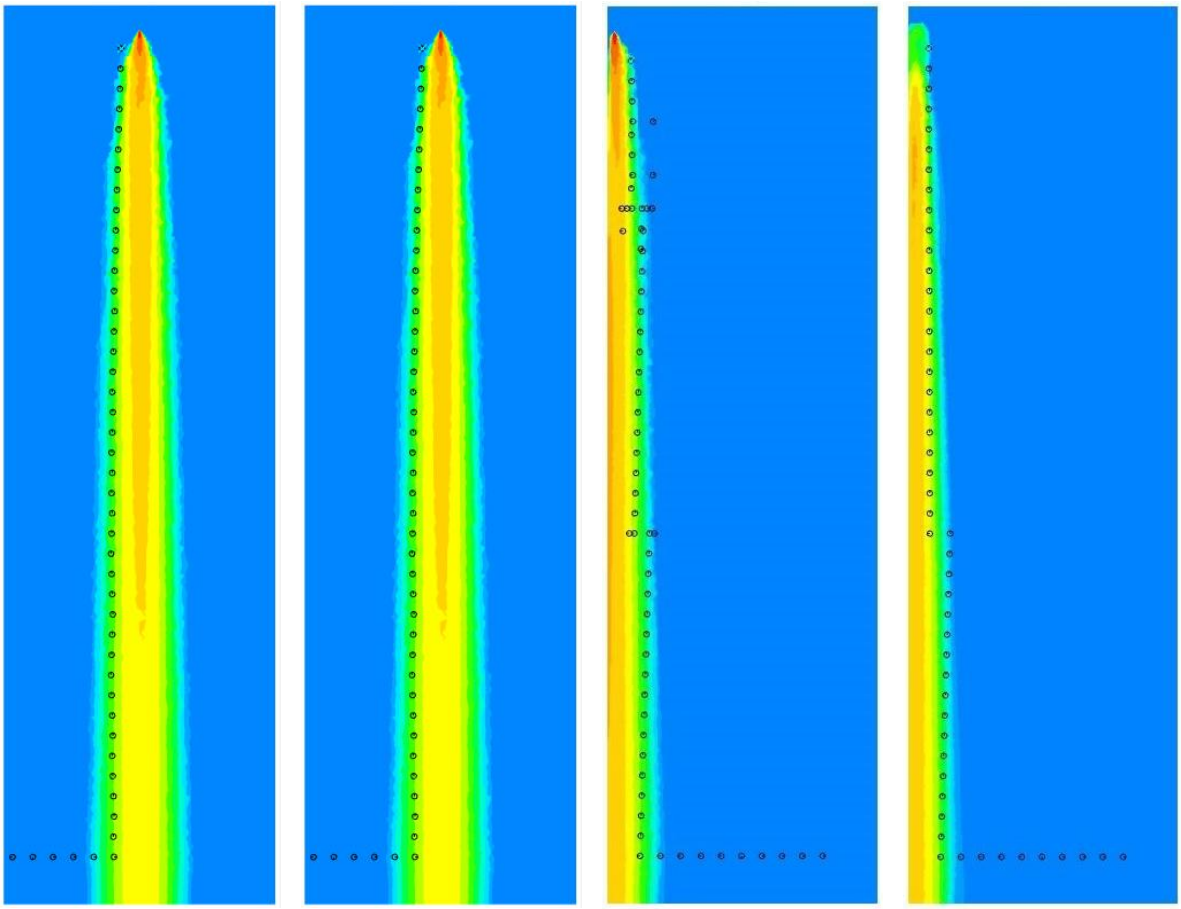
(b)

Figure 4.10: (a) The overall logic flow of the 'along-wall' obstacle avoidance method, (b) Calculation of the turning angle

surging along the wall, the robot can revise its heading to make it parallel to the wall. Either leaving the plume or the end of a wall will stop this 'along-wall' process, then the robot will return to the normal plume-tracing status. Moreover, when meeting a wall, the robot utilises not only the data measured by ultrasonic sensors, but also the wind direction measured by the wind sensor. Different from the previous *wall following* obstacle avoidance [10], when facing a wall, the robot is able to turn towards the upwind direction and surge along it. For example, when the robot heads to the downwind direction after turning α, the robot will turn back in order to surge along the wall and, meanwhile, surge towards the upwind direction.

In terms of the simulation results presented in Figure 4.11 and Figure 4.12, the robot with *vallumtaxis* is capable of detecting and localising the chemical source successfully. In Scenario M, the total number of steps as well as the trajectories of the robot coded with *vallumtaxis* are the same as that of the robot with Algorithm 1. In Scenario S, the trajectories of the robot with *vallumtaxis* show

that the robot went along the wall without moving back and with *vallumtaxis* the total number of steps significantly reduced (See Figure 4.11 and Figure 4.12). It can be concluded that, when compared with the algorithms with the *concentration-distance obstacle avoidance method* (explained in Section 4.1), *vallumtaxis* successfully avoids repeating movements near a wall (where the plume is slim) and helps to achieve higher efficiencies. Therefore, *vallumtaxis* can be adopted for plume-tracing robots for locating a gas leakage source in an indoor environment like Scenario M and Scenario S in this study, particularly when the source is near a wall, such as in Scenario S.



(a) Algorithm 1, Scenario M, concentration-distance obstacle avoidance, released from location A, step number 46

(b) *Vallumtaxis*, Scenario M, along-wall obstacle avoidance, released from location A, step number 46

(c) Algorithm 1, Scenario S, concentration-distance obstacle avoidance, released from location C, step number 63

(d) *Vallumtaxis*, Scenario S, along-wall obstacle avoidance, released from location C, step number 51

Figure 4.11: Comparison of trajectories between *vallumtaxis* and Algorithm 1

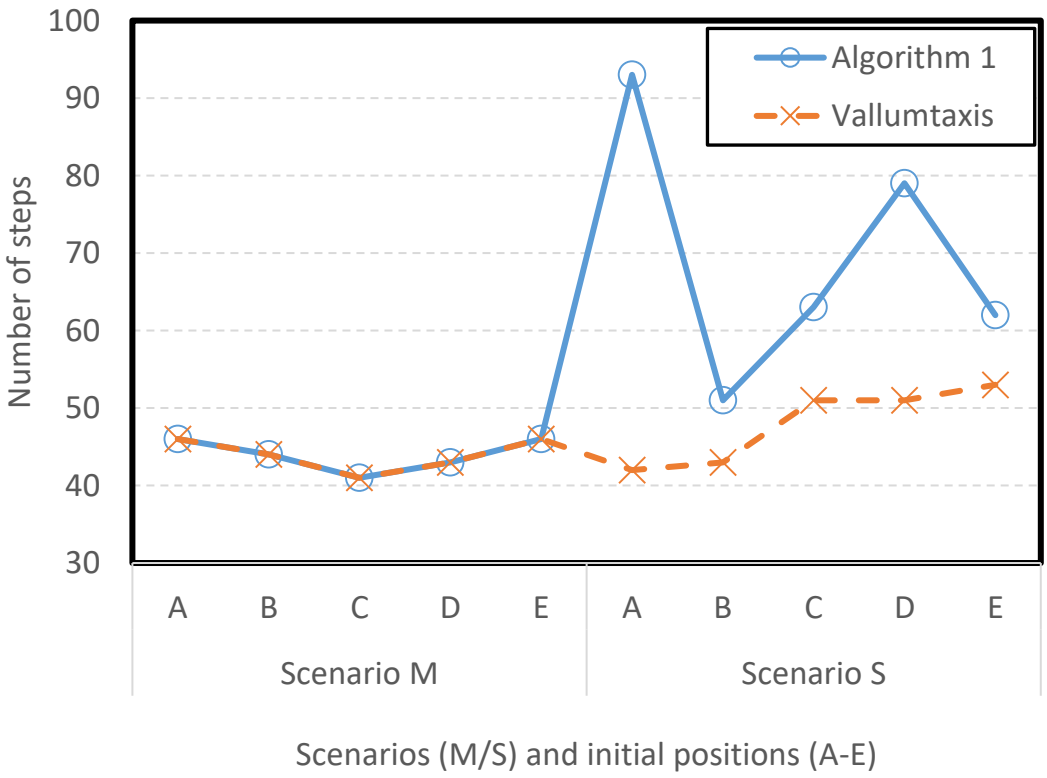


Figure 4.12: Comparison of the number of steps between Algorithm 1 and *vallumtaxis*

Chapter 5 Application of plume-tracing robots in an underground warehouse

The work presented in Chapter 5 is based on the material of a journal article that is going to be submitted. In this chapter, the performance of different plume-tracing algorithms in an underground warehouse is investigated.

5.1 The robot

As the geometry size of the environments being different, the fluid field proposed in the underground warehouse is different from that in the wind-tunnel-like channel reported in Chapter 4. The proposed mobile plume-tracing robot in Chapter 5 is also designed for plume tracing in strong 3D plumes following the robot presented in Chapter 4 of this thesis (Figure 5.1). Generally, plume-tracing robots move step by step, which is the definition of 'surge' because the wind and chemical sensors need to work in a steady condition to obtain an accurate reading. All the sensors are located on a plane as is shown in Figure 5.1, and the plane is capable of moving vertically. Therefore, the sensors are able to work at different horizontal levels. Data, including the chemical concentration and wind direction, are only measured when both the robot and the sensor module plate are still. Also, to reduce the influence of the moving plate on the measurement of the sensors, normally there is a short delay on measurement after the robot and the plate stops moving. Figure 5.1 represents the distribution of wind and chemical sensors, as well as the covering areas of four ultrasonic sensors. The

wind sensor is used to measure the wind velocity and it locates at the centre of the plane. In total four chemical sensors are equipped at the centre, in the front, and to the left hand side and right hand side, respectively. Four ultrasonic sensors are armed to measure the distance between the robot and the nearest obstacles in different directions. In this study, the distance between the robot and an obstacle together with the wind direction measured by wind sensors are used for obstacle avoidance. In real scenarios, wind sensor, ultrasonic sensors and chemical sensors measure the wind direction, working at different horizontal levels (in this part of research, 0.3 m, 0.5 m and 1 m above the ground, respectively). The robot uses the data on the height where the highest chemical concentration is detected. In the simulation framework, the fluid field and concentration distribution at the horizontal level at 0.3 m, 0.5 m and 1 m above the ground are accessed by MATLAB. According to the dimensions of this large indoor environment, the *stepsize* is determined to be 0.7 m. When the distance between the robot and the source is less than 1 m, it is considered to be a successful plume-tracing case.

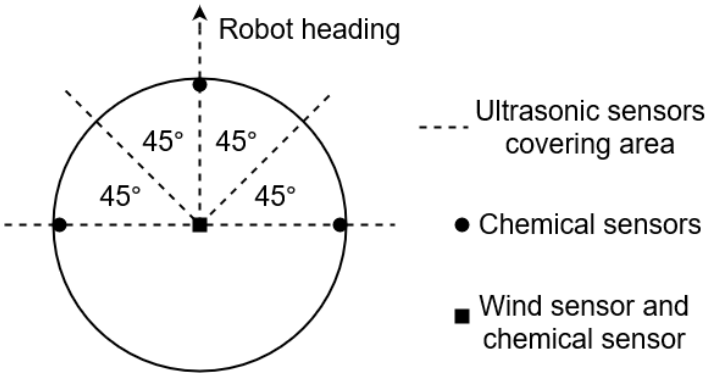


Figure 5.1: Covering areas of ultrasonic sensors and positions of wind and chemical sensors

5.2 Plume-tracing algorithms

As afore demonstrated, single plume-tracing algorithm is divided into three stages for discussion. The plume-tracing algorithms applied in this part are all

designed based on *vallumtaxis*, which is a combination of *normal casting*, *surge anemotaxis*, *constant stepsize* and *along-wall obstacle avoidance method* [2, 15, 25, 47]. In this research, when the robot gets trapped circling for many steps (more than 30 steps) or surges obviously away from the source, it is determined to be a failure case.

5.2.1 Plume sensing

Two different plume sensing methods, namely *normal casting* and *special casting* were applied. *Normal casting* and *special casting* are illustrated in Section 4.2.1.

5.2.2 Plume tracking

Three plume tracking methods including *surge anemotaxis*, *zigzags,* and *pseudo gradient-based algorithm* were tested. These three methods at PT stage are demonstrated in Section 4.2.2.

5.2.3 Source localisation

Initially, two different searching strategies at source localisation stage were applied. *Constant stepsize* means that at source localisation stage the robot moves at the same *stepsize* as that at plume tracking stage. For *variable stepsize*, the higher the concentration is, the smaller the *stepsize* is. The *stepsize* in *variable stepsize* in this part can be calculated by the following equations:

$$S_{sl} = \begin{cases} \left(\frac{1}{3} + \frac{2(\log T_3 - \log C)}{3(\log T_3 - \log T_2)}\right) \times S_{pt}, & T_2 \leq C \leq T_3 \\ \frac{S_{pt}}{3}, & C > T_3 \end{cases} \quad (5.1)$$

where S_{sl} is the *stepsize* at source localisation stage, and S_{pt} is the *stepsize* at plume tracking stage. C is the local chemical concentration measured by the chemical sensor in the centre of the robot. T_2 and T_3 are the two thresholds set for plume tracing. T_2 is the threshold that distinguishes plume tracking and source localisation stages. T_3 is a value higher than T_2 and a novel threshold set

for the *variable stepsize* in this part of the research to ensure that the robot is moving at a proper stepsize when approaching the source. Equation 5.1 is designed referring to [2] and [25]. All the variables regarding the chemical concentration are logarithmic for better performances. It is different from the calculation in Section 4.2.3, [2] and [25] because it was found in [25] that the *stepsize* could decrease significantly when approaching the source, which lowers the efficiencies of algorithms with *variable stepsize*. Therefore, in this part of the research, a minimum *stepsize* was set. When the concentration C is between T_2 and T_3, the robot gradually moves more slowly when the concentration increases. However, when the concentration C is higher than T_3, the *stepsize* remains at $S_{pt}/3$. Under this circumstance, the robot will still move at a proper speed when the environmental concentration is high. T_2 and T_3 are subject to simulation studies before any applications of the plume-tracing robot.

5.2.4 Plume-tracing algorithms

This part of the research aims to test different plume-tracing algorithms in large-space areas, where large recalculating flows and complex fluid fields emerge. Table 5.1 presents searching methods at different stages for Algorithms 1 to 5. All the algorithms are designed based on the *vallumtaxis* conducted in [25],

Table 5.1: Searching methods of Algorithms 1 to 5

Algorithm No.	Plume sensing	Plume tracking	Source localisation
Algorithm 1	*Normal casting*	*Surge anemotaxis*	*Constant stepsize*
Algorithm 2	*Special casting*	*Surge anemotaxis*	*Constant stepsize*
Algorithm 3	*Normal casting*	*Zigzags*	*Constant stepsize*
Algorithm 4	*Normal casting*	*Pseudo gradient-based algorithm*	*Constant stepsize*
Algorithm 5	*Normal casting*	*Surge anemotaxis*	*Variable stepsize*

Algorithm 1 is the original *vallumtaxis*. *Vallumtaxis* has proved to be an effective tool for plume tracing in wall plumes in laboratory-scale environments [25]. In this part of study, different searching methods for the plume sensing, plume tracking and source localisation stage are applied for further investigation. Algorithm 2 is the *vallumtaxis* equipped with *special casting*. Algorithms 3 and 4 are the *vallumtaxis* equipped with different plume tracking methods, respectively, and Algorithm 5 is *vallumtaxis* equipped with a different SL method.

5.3 Results and discussion

Table 5.2 shows the preliminary results in all four scenarios. The letter *s* in Table 5.2 denotes 'steps' and the numbers denote the numbers of steps from the initial position to the source. Referring to previous research [2, 4, 25, 40], when the distance between a robot and a source is lower than 1 m in real scenarios, it is seen as a success case. The 'Algorithm No.' corresponds to Table 5.1 in Section 5.2.4. A higher number of steps for plume tracing indicates higher time cost and lower efficiency. Fewer steps denotes less plume-tracing time and therefore a higher efficiency. It can be seen that it is easier (with higher success rates) for the robot to locate the source in Scenarios 2 and 3 than in Scenarios 1 and 4. Also, it

Table 5.2: Assessed results of Algorithms 1 to 5 in four scenarios

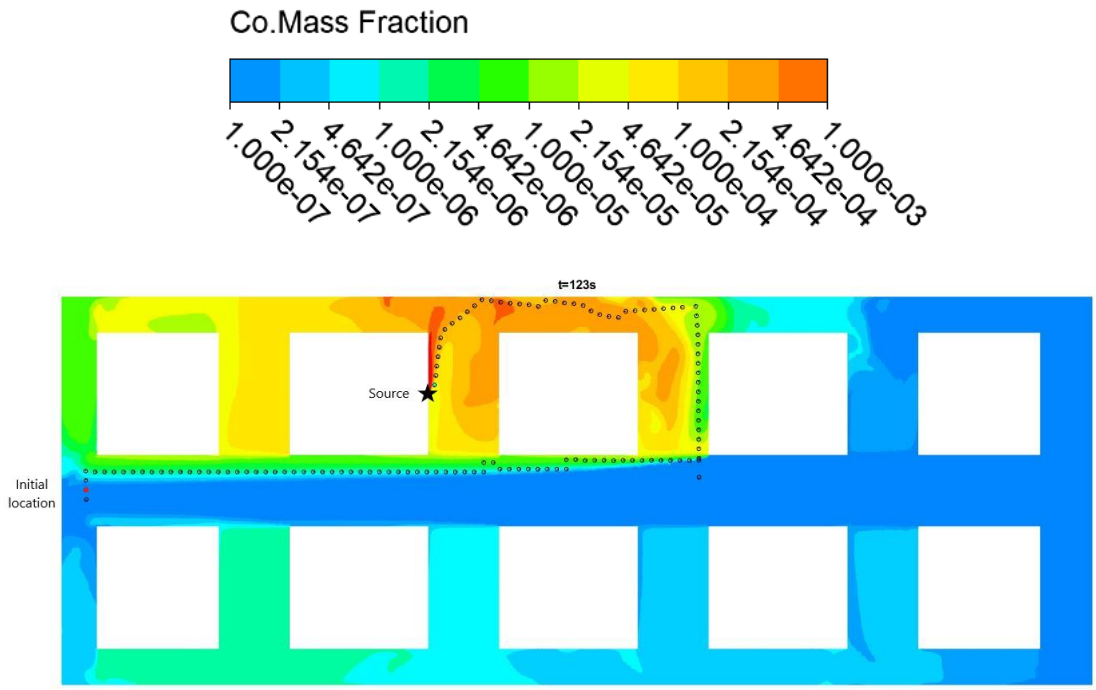
Algorithm No.	Scenario A	Scenario B	Scenario C	Scenario D
Algorithm 1	123 s	57 s	80 s	F-SL
Algorithm 2	F-SL	F-SL	86 s	F-SL
Algorithm 3	F-SL	F-PT	107 s	F-SL
Algorithm 4	F-PT	60 s	F-SL	F-SL
Algorithm 5	127 s	58 s	80 s	F-SL

Note: 's' equals 'steps'. 'F-PS' equals 'Failed at plume sensing stage'. 'F-PT' equals 'Failed at plume tracking stage'. 'F-SL' equals 'Failed at source localisation stage'.

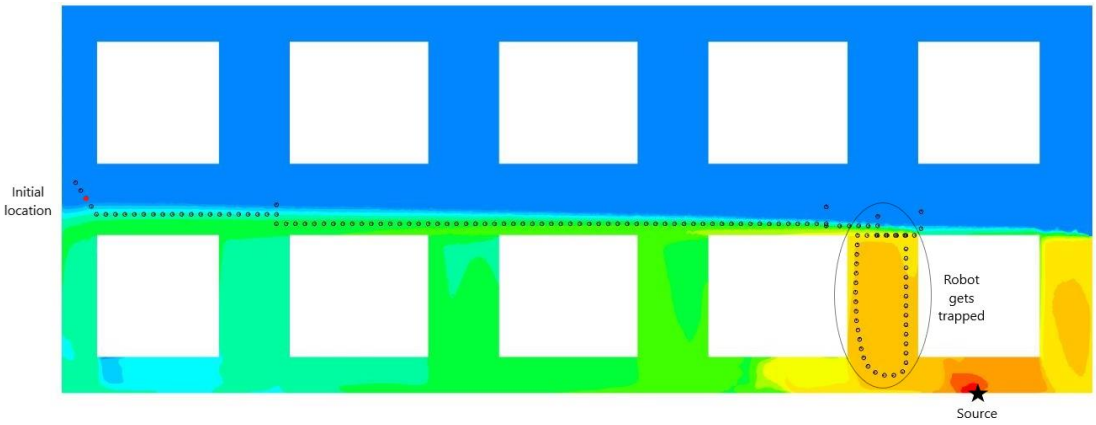
costs the robot with Algorithm 1 less time to localise the source than the robot using other algorithms in the same scenario. The stage at which the robots failed is also shown in Table 5.2. For instance, the robot coded with Algorithm 1 failed in Scenario 4 at source localisation. More results are given in Appendix B.

5.3.1 Assessment on plume sensing

Figures 5.2 (a) and (b) present a success case and a failure case of the robot coded with Algorithm 1 searching in Scenarios 1 and 4, respectively. Please note that all following figures in Chapter 5 show the contours of the concentration distribution of carbon monoxide in the warehouse. The colour legend in Figure 5.2, which is logarithmic for better visualisation, applies to Figures 5.2-5.11. It can be seen from the results of Algorithm 1 and 2 that no evidence showing that *special casting* helps improving the efficiency of the plume-tracing robot working in large-scale environments, was found (Table 5.2, Figures 5.2 (a), (b), 5.3 (a) and (b)). Originally, the *special casting* was designed for finding the centreline of the plume before the next stage. The concentration distributions shown in all of Figures 5.2 and 5.3 indicate that the shape of the plumes in the large-scale warehouse can be irregular and influenced significantly by the geometry of the environment. It can be seen from Figures 5.3 and 5.4 that with *special casting*, the robot is able to find the centreline between the plume boundaries and walls. In Figures 5.3 (a) and (b), the robot left the plume five and three times, respectively, and then came back and started the next stage at the centre of the two plume boundaries. However, compared with Algorithm 1, the robot with *special casting* did not perform at a lower time cost. Under this circumstance, there is no evidence that *special casting* helps reduce the time cost of plume-tracing robots in large-scale environments.

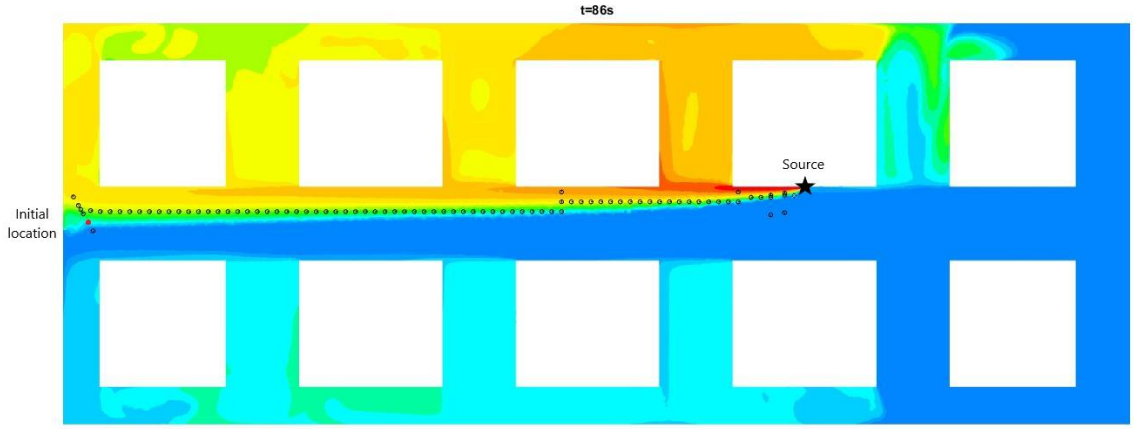


(a) t= 123 steps

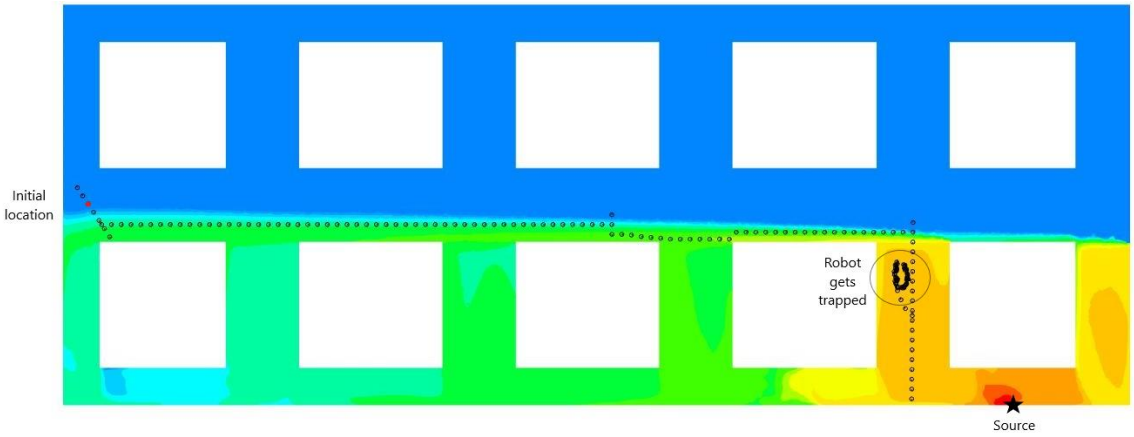


(b) failed

Figure 5.2: The trajectories of the (a) robot with Algorithm 1 searching in Scenario 1, (b) Robot with Algorithm 1 searching in Scenario 4



(a) t= 86 steps



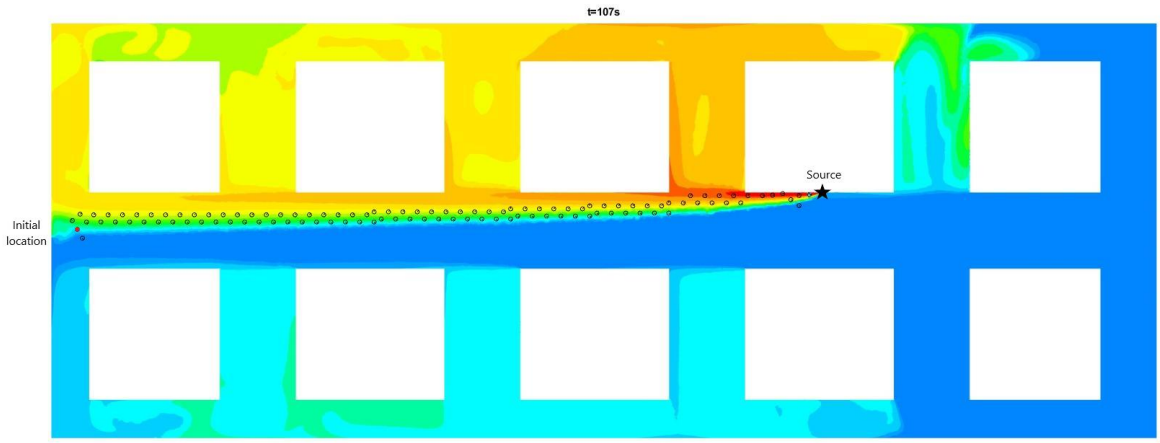
(b) failed

Figure 5.3: The trajectories of the (a) robot with Algorithm 2 searching in Scenario 3, (b) robot with Algorithm 2 searching in Scenario 4

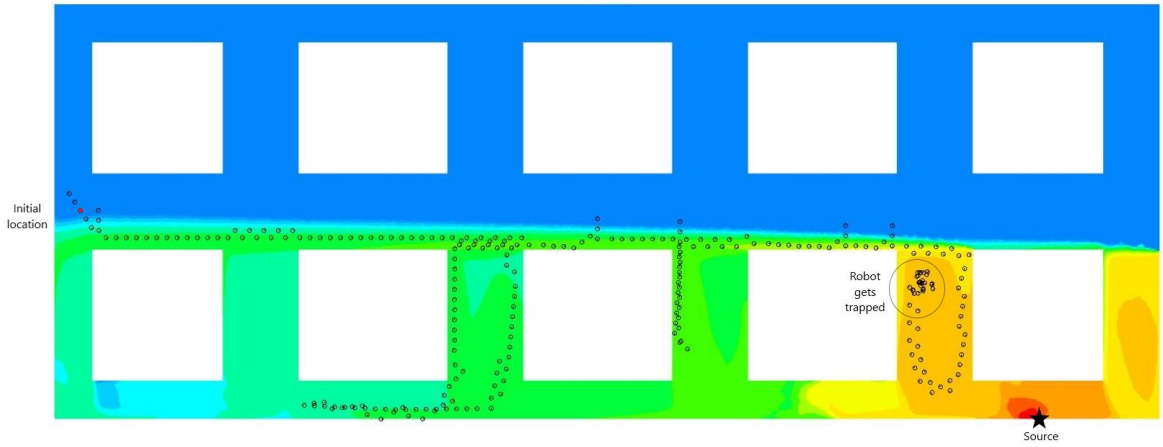
5.3.2 Assessment on plume tracking

Typical testing results of *vallumtaxis* with *zigzags* and a *pseudo gradient-based algorithm* are shown in Figures 5.4 and 5.5, respectively. According to Table 5.2, both *zigzags* and the *pseudo gradient-based algorithm* show lower success rates than that of the original *vallumtaxis* (Algorithm 1). Table 5.2 reveals that *zigzags* and the *pseudo gradient-based algorithm* do not perform with a higher efficiency and greater success rates than *surge anemotaxis*. It can be seen in Figure 5.4 (a)

that the robot succeeds in localising the source following *zigzags* at plume tracking stage. However, it cost more time for the robot with Algorithm 3 to localise the source than that with Algorithm 1. Moreover, it can be observed from Figures 5.5 (a) and (b) that the robot with Algorithm 4 is able to turn to the direction with higher concentrations; it then moves and localises the source in Scenario 2. However, the robot with Algorithm 4 failed with endless circling in Scenario 1 (Figure 5.5 (b)). It can be seen from Figures 5.2, 5.3, 5.4, and 5.5 that, at plume tracking stage, by moving along the walls, the robot successfully avoided most influences from the recalculating flows in the fluid field. However, according to the failure cases shown in Figures 5.2 (b), 5.3 (b), 5.4 (b), and 5.5 (b), the robots all ended by being with trapped in an endless circle. In Figures 5.2 (b), 5.3 (b), and 5.4 (b) the robot failed at SL stage while in Figure 5.5 (b) the robot lost at PT stage. According to the testing results above, the distance that the robot with *surge anemotaxis* at plume tracking stage moves made shorter due to following upwind direction. Therefore, it can be concluded that *surge anemotaxis* is the most successful searching strategy at plume tracking stage among the compared ones for a robot searching in this underground warehouse.



(a) t= 107 steps



(b) failed

Figure 5.4: The trajectories of the (a) robot with Algorithm 3 searching in Scenario 3, (b) robot with Algorithm 3 searching in Scenario 4

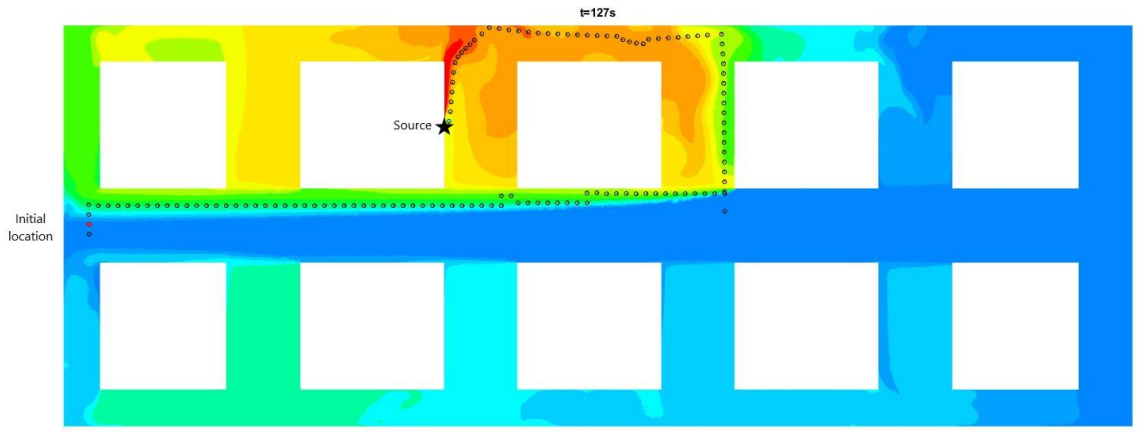
(a) t = 60 steps

(b) failed

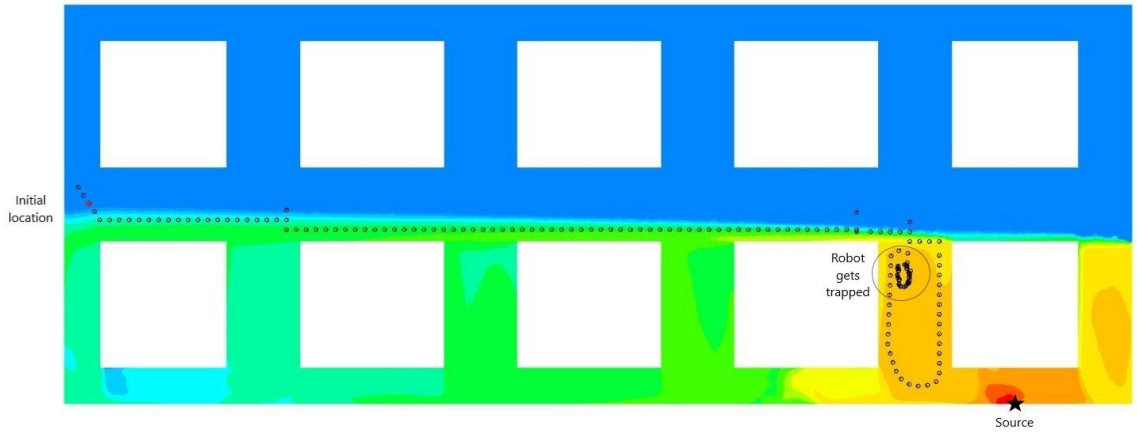
Figure 5.5: The trajectories of the (a) robot with Algorithm 4 searching in Scenario 2, (b) robot with Algorithm 4 searching in Scenario 1

5.3.3 Assessment on source localisation

Figures 5.6 (a) and (b) show that *variable stepsize* navigates the robot to move more slowly when near the source. However, the robot still failed and ended up with circling in Scenario 4, where all the tested algorithms failed. In conclusion, it is shown in Table 5.2 that, for most of the failure plume tracing cases, the robot failed at source localisation stage, which indicates that the current source localisation method is not able to trace plumes in large-scale environments. In this situation, to obtain higher success rates and efficiencies for plume-tracing algorithms searching in large-scale environments, it is recommended to focus on enhancing the method at source localisation stage. Hence, several more plume-tracing algorithms, all with the same *normal casting* and *surge anemotaxis* but with different searching methods at source localisation stage, are further assessed.



(a) t= 127 steps



(b) failed

Figure 5.6: The trajectories of the (a) robot with Algorithm 5 searching in Scenario 1, (b) robot with Algorithm 5 searching in Scenario 4

5.4 Follow-up research on source localisation

It had already been found in Section 5.3 that for most failure cases, the robot failed at source localisation stage ending up circling. Therefore, a method that could help the robot get out of the circle is needed. Searching methods, such as *variable stepsize, zigzags, pseudo gradient-based algorithm* and *chemotaxis* are able to navigate the robot to surge in another direction instead of following the upwind direction completely. Consequently, different methods at source localisation stage, on which this research focuses, mixing with *normal casting*, *zigzags*, and *along-wall obstacle avoidance method*, were tested further. This research applies more different methods at source localisation to get higher efficiency when searching in large-scale indoor environments. Table 5.3 shows the searching methods of Algorithms 6 to 10 at different stages and Table 5.4 shows the assessed results of these algorithms in four different scenarios. It can be seen that for Scenario B, the robot with Algorithms 6 to 10 succeeded in plume-tracing, while in Scenario D, only the robot with Algorithm 10 succeeded. Algorithm 10 performed the best among all other tested algorithms on its success

rate.

Table 5.3: Searching methods of Algorithms 6 to 10

Algorithm No.	Plume sensing	Plume tracking	Source localisation
Algorithm 6	*Normal casting*	*Surge anemotaxis*	*Variable stepsize & zigzags*
Algorithm 7	*Normal casting*	*Surge anemotaxis*	*Pseudo gradient-based algorithm*
Algorithm 8	*Normal casting*	*Surge anemotaxis*	*Variable stepsize & Pseudo gradient-based algorithm*
Algorithm 9	*Normal casting*	*Surge anemotaxis*	*Surge anemotaxis & Chemotaxis*
Algorithm 10	*Normal casting*	*Surge anemotaxis*	*Surge anemotaxis & Pseudo casting*

Table 5.4: Assessed results of Algorithms 6 to 10 in four scenarios

Algorithm No.	Scenario A	Scenario B	Scenario C	Scenario D
Algorithm 6	130 s	62 s	83 s	F-SL
Algorithm 7	F-PT	57 s	F-SL	F-SL
Algorithm 8	F-SL	57 s	136 s	F-SL
Algorithm 9	F-SL	57 s	80 s	F-SL
Algorithm 10	178 s	57 s	80 s	124 s

Note: 's' equals 'steps'. 'F-PS' equals 'Failed at plume sensing stage'. 'F-PT' equals 'Failed at plume tracking stage'. 'F-SL' equals 'Failed at source localisation stage'.

Figures 5.7 to 5.11 present selected results of applying *variable stepsize & zigzags, pseudo gradient-based algorithm, variable stepsize & pseudo gradient-based algorithm, surge anemotaxis & chemotaxis and surge anemotaxis & pseudo casting* on *vallumtaxis*, respectively. *Variable stepsize & zigzags* is the searching strategy that combines *variable stepsize* and *zigzags* together meaning that the robot moves following *zigzags*, and meanwhile the *stepsize* varies with

concentration. *Pseudo gradient-based algorithm* is the same as the one demonstrated in Section 4.3.2 of this thesis, only triggered at source localisation stage. *Variable stepsize & pseudo gradient-based algorithm* is that the robot moves following the mechanism of *pseudo gradient-based algorithm* while the *stepsize* varies with concentration following Equation 5.1. Algorithms 9 and 10 are equipped with a feedback regulation, which means when the robot is detecting a gradual decrease of concentration when moving, the robot will move back to the previous step and then move following *surge anemotaxis & chemotaxis* (Algorithm 9) or *surge anemotaxis & pseudo casting* (Algorithm 10). For *surge anemotaxis & chemotaxis*, the robot normally moves following *surge anemotaxis* at a constant *stepsize*. After moving back due to the decreasing chemical concentration, the robot will move following *chemotaxis* (illustrated in Section 5.2) at the next step. While for *surge anemotaxis & pseudo casting*, after getting back one step, the robot will soon move across the wind (in the direction perpendicular to the wind direction) to gather the concentration at several locations at the cross-section linear area. The robot stops moving when detecting a wall or leaving the plume and then turns back and begins to search on the other side. *Pseudo casting* ends with finishing searching both the right and left hand sides and moving back to the location with the highest concentration. This searching strategy is named *pseudo casting* due to being inspired by *casting* at plume sensing stage. Details of *surge anemotaxis*, *chemotaxis*, and *casting* are illustrated in Section 4.2.

For Algorithm 6, as was demonstrated in Tables 5.2 and 5.4 and Figure 5.7 that the success rate of Algorithm 6 is the same as that of Algorithm 1 while it costs the robot with Algorithm 6 a little more time to localise the source than the robot with Algorithm 1. *Variable stepsize & zigzags* still did not help the robot localise the source in Scenario 4 successfully. For Algorithm 7 with *pseudo gradient-based*

algorithm for source localisation, the robot could go in the wrong direction (away from the source) near the source. Figures 5.8 and 5.9 indicate that both *variable stepsize & pseudo gradient-based algorithm* and *pseudo gradient-based algorithm* at source localisation stage did not perform better than the *variable stepsize*.

Since the robot should be able to avoid missing the source, searching strategies that combine wind direction and concentration data together could be appropriate approaches. It is shown in Figure 5.10 that for *anemotaxis & chemotaxis*, the robot could change its pathway by moving back one step and then moving towards the area with higher concentration. However, it can be found from Table 5.4 and Figure 5.10 that *anemotaxis & chemotaxis* did not achieve higher success rate or efficiencies.

Figures 5.11 (a) and (b) show the trajectories of the robot with Algorithm 10 searching in Scenario 1 and Scenario 4, respectively. It is shown in Table 5.4 that Algorithm 10, with *pseudo casting*, has the highest success rate among all ten tested algorithms in this part of study. In Scenario 1, it costs more time for the robot with Algorithm 10 to reach the source than the robot with Algorithm 1. As is demonstrated in Figure 5.11 (a), *pseudo casting* was triggered several times as the chemical concentration distribution could be complex due to large eddies. Finally the robot successfully localised the plume source in Scenario 1. The trajectories of the robot with Algorithm 10 in Scenarios 2 and 3 are similar with that of Algorithm 4 and Algorithm 2 in Figures 5.5 (a) and 5.3 (a), respectively. In Scenario 4, it can be seen in Figure 5.11 (b) that after leaving the wall near the source, the robot sensed a negative gradient of concentration, moved back and then moved in the direction perpendicular to the wind direction. Eventually, the robot ended with the location with the highest concentration and succeeded in localising the source. Analysing the chemical concentration at several different

positions, *pseudo casting* successfully navigated the robot with Algorithm 10 to localise the source in Scenario 4, and this is the only success case for plume tracing in Scenario 4.

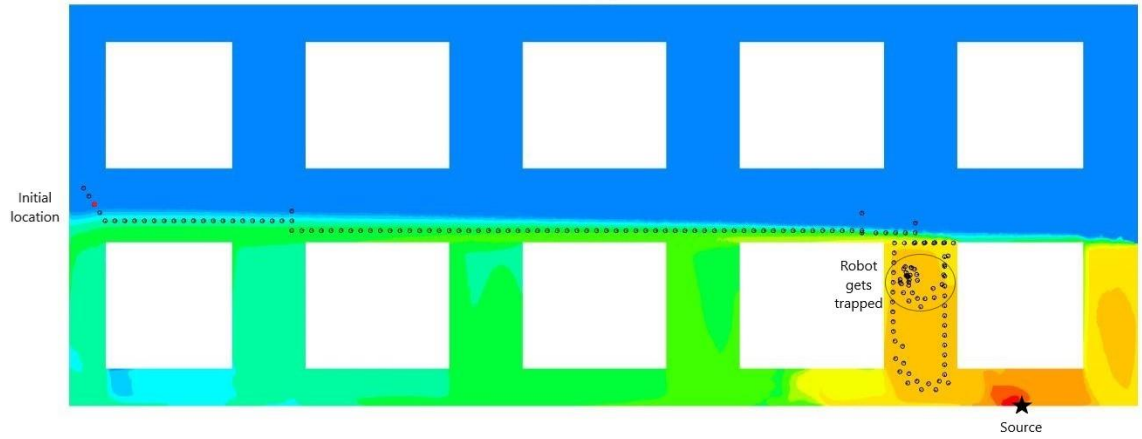


Figure 5.7: The trajectory of the robot with Algorithm 6 searching in Scenario 4 (failed)

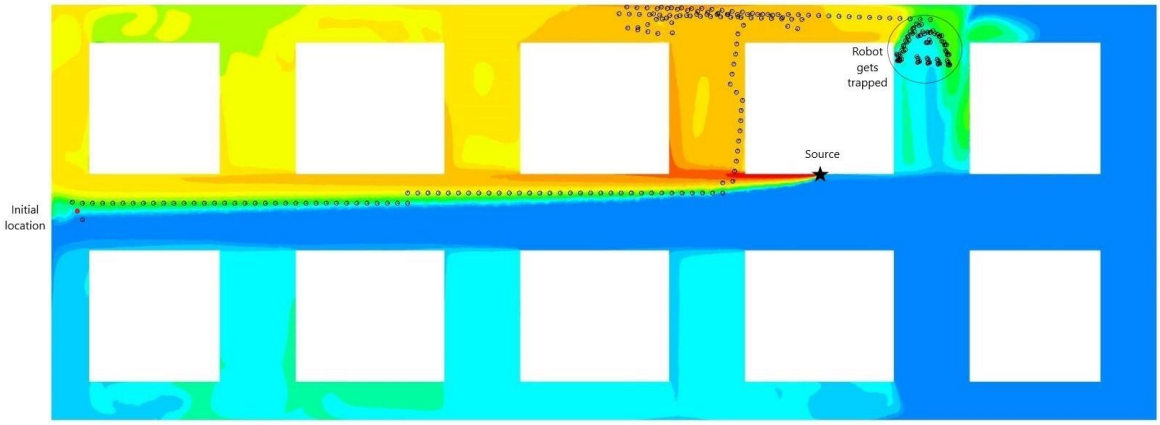


Figure 5.8: The trajectory of the robot with Algorithm 7 searching in Scenario 3 (failed)

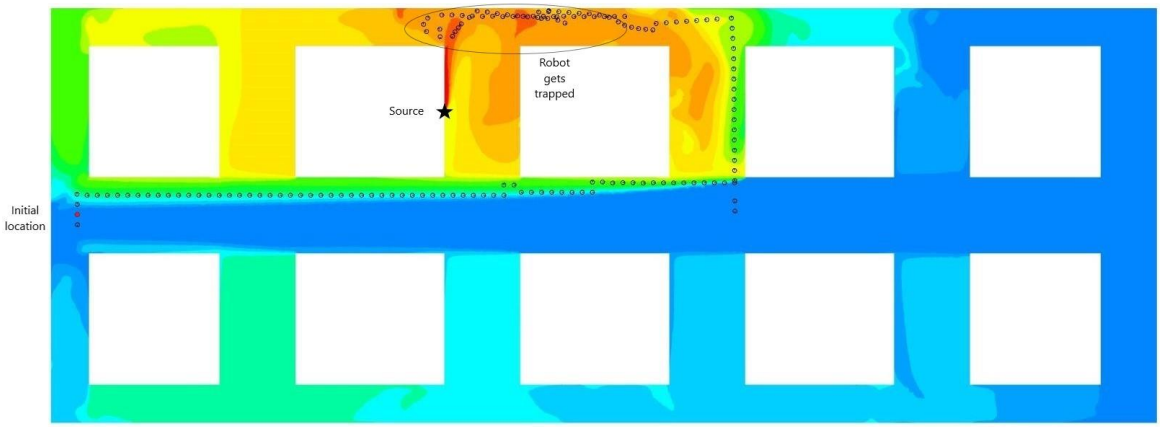


Figure 5.9: The trajectory of the robot with Algorithm 8 searching in Scenario 1 (failed)

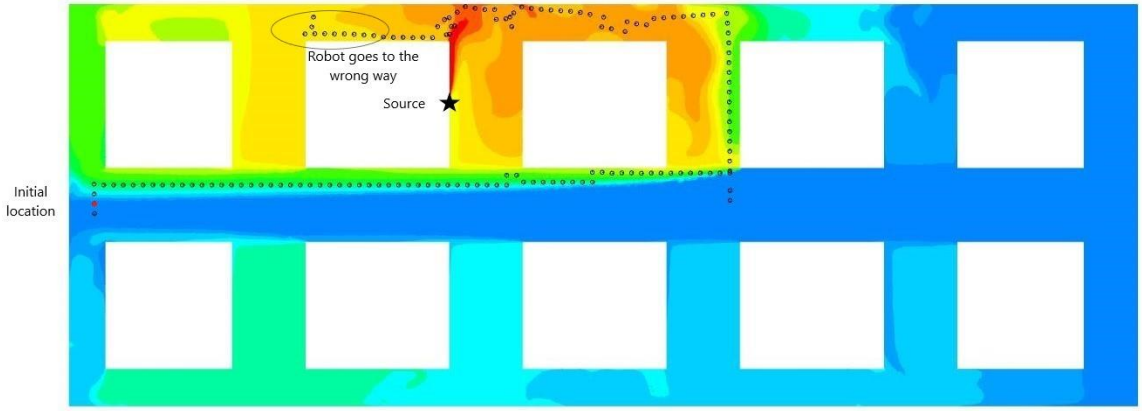
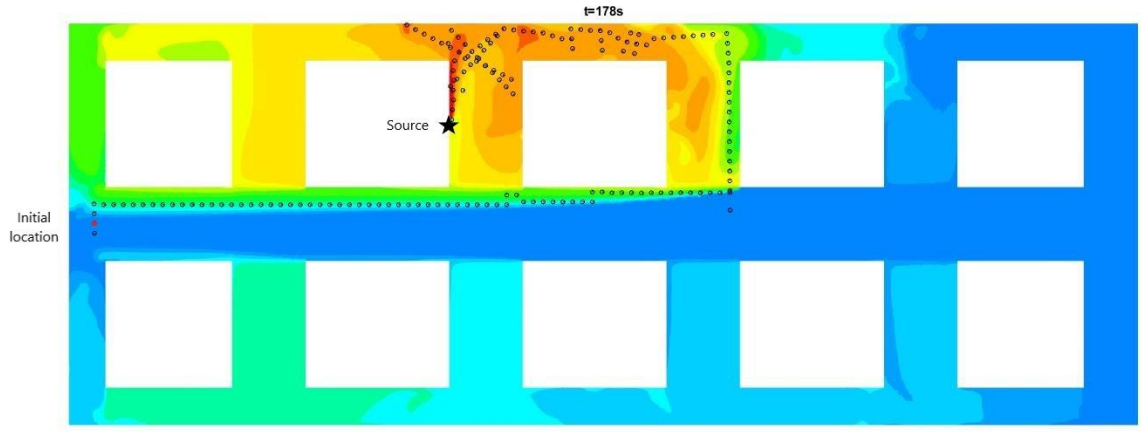
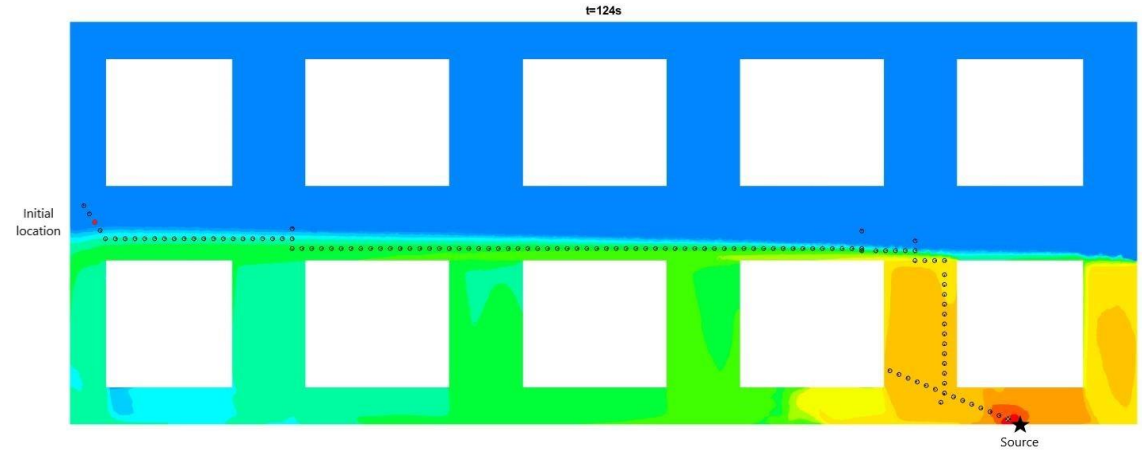


Figure 5.10: The trajectory of the robot with Algorithm 9 searching in Scenario 1 (failed)



(a) t= 178 steps



(b) t= 124 steps

Figure 5.11: The trajectories of the (a) robot with Algorithm 10 searching in Scenario 1, (b) robot with Algorithm 10 searching in Scenario 4

Chapter 6 Conclusion and future work

6.1 Conclusion

In conclusion, a total of twenty-one plume-tracing algorithms were tested in two indoor environments this research. Sixteen plume-tracing algorithms were tested in two laboratory-scale environments, and ten were tested in four scenarios in an underground warehouse. In particular, a novel plume-tracing algorithm called *vallumtaxis* has been proposed and tested in both the channel flow and the underground warehouse.

In the first part of this research, *normal casting*, *surge anemotaxis*, and *constant stepsize* together performed the best, when compared with other algorithms. When a plume is away from walls, a plume-tracing robot released from a position that is inside the plume resulted in higher efficiencies in terms of total step numbers. While for a wall plume, no advantageous initial locations for the robot were found. The *concentration-distance obstacle avoidance method* proved to be able to protect the robot from obstacles; however, its performance was unsatisfactory when the robot is searching in wall plumes. Summarising all the comparison results, a novel plume-tracing algorithm called *vallumtaxis*, which is made up with *normal casting*, *surge anemotaxis*, *constant stepsize* and a specially designed *along-wall obstacle avoidance method* was proposed. It is found that *Vallumtaxis* achieves higher efficiencies than other tested algorithms especially when searching in wall plumes in this laboratory-scale indoor environment.

In the second part, the detection and localisation of hazardous plume sources in an underground warehouse, such as the one given in Chapter 5, using a plume tracing robot proves to be applicable. There are both free-stream plumes and wall plumes in this large-scale environment. It was found that algorithms based on *vallumtaxis,* coupled with the *along-wall obstacle avoidance method* successfully avoided most recalculating flows in the underground warehouse by moving along the walls. However, the robot still ended up circling in recalculating flows under many circumstances. The plume sensing method including *special casting* and plume tracking methods, such as including *zigzags* and a *pseudo gradient-based algorithm* show no enhancements on plume tracing in large-scale indoor environments. Also, the robot did not benefit from applying a searching method called *variable stepsize* at source localisation stage. Furthermore, since source localisation is the most important stage based on preliminary testing results of Algorithms 1-5, this research further focuses on different methods at source localisation stage, for plume tracing in this large-scale indoor case. Different searching strategies were undertaken at source localisation stage to help the robot move out of recirculating airflows and it was found the algorithm with the novel *pseudo casting* is capable of achieving a higher success rate of plume tracing in the underground warehouse.

In conclusion, the confined flow simulated in the current study is a fundamental and essential flow that is often encountered in indoor environments. Also, the warehouse is a typical underground facility, presented in several previous case studies. The conclusion and findings of this research could, therefore, be applied to many other laboratory-scale or large-scale indoor environments.

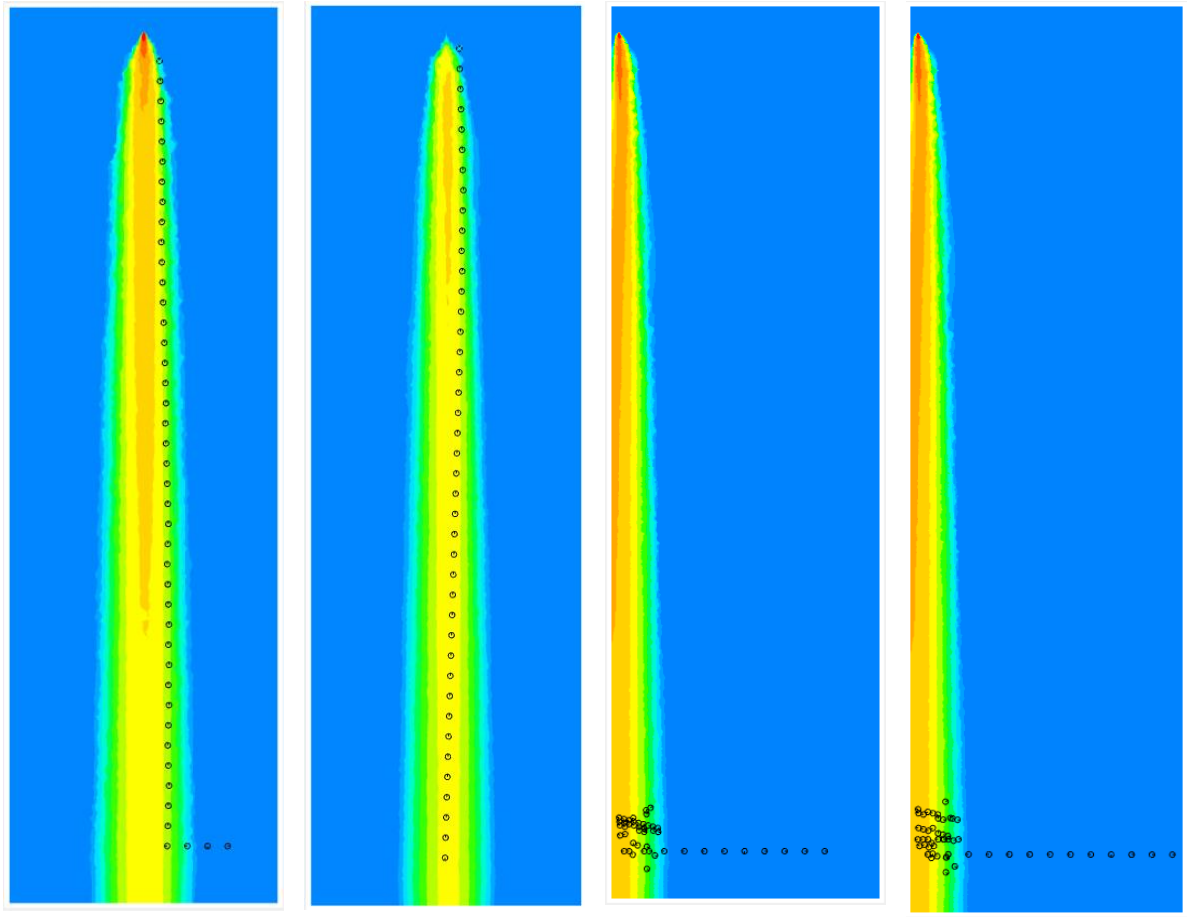
6.2 Future work

Plume-tracing methods have been applied to micro drones searching for

hazardous plume sources in outdoor environments [42]. For some large-scale indoor environments, such as the underground warehouse given in Chapter 5, the height of the domain can be much larger than the height of the robot. Testing the performance of different plume-tracing algorithms indoor plumes with multiple sources is one of the future tasks deriving from this study. Also, it is challenging for robots that move on the ground to search for hazardous plume sources that are high above the ground (for instance, higher than 2 m). Under this circumstance, a micro drone with the capability of searching at different horizontal levels may provide a better approach for plume tracing. Secondly, a potential approach to reducing the total time for searching for such plume-tracing robot is to use more sensors instead of moving the sensor module. For example, using multiple sensors at different horizontal levels could save time of the plume-tracing progress. Nevertheless, the cost of applying multiple sensors can be much higher than that of the current design. Therefore it is recommended that the optimisation of the setup of sensors without high cost could be investigated in the future. Thirdly, in Chapter 4, *vallumtaxis* is only tested in a simple wind-tunnel-like channel and in Chapter 5, the *vallumtaxis* with *pseudo casting* at SL stage is tested in different scenarios with only one hazardous plume source. Since wind fields and chemical concentration distributions could vary significantly if multiple sources emerge, *vallumtaxis* will soon be tested in other scenarios where there are multiple sources. Moreover, according to Chapter 2, most of the studies regarding plume tracing are performed in indoor environments. Therefore, more tests and assessments of the performance of different plume-tracing algorithms in outdoor environments should be investigated.

Appendix A Case 1

In Appendix A, supplementary results of testing the performances of different plume-tracing algorithms in the wind-tunnel-like channel are provided. The algorithm No. refers to Table 4.2.

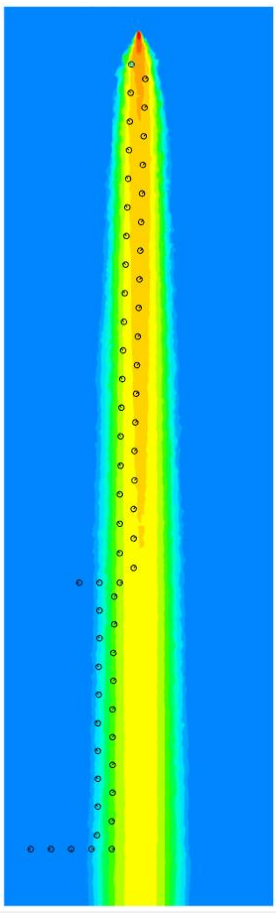


(a) Algorithm 1, Scenario M, released location D, step number 43

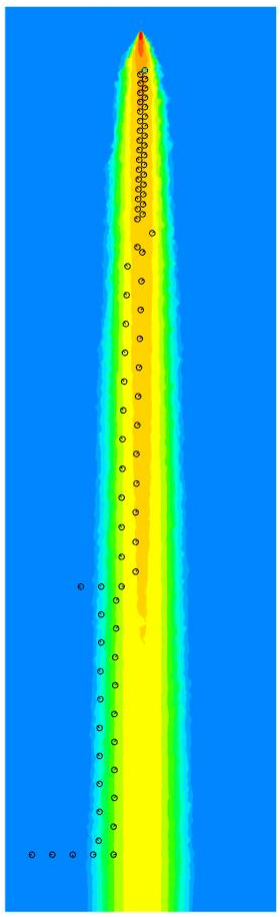
(b) Algorithm 2, Scenario M, released location C, step number 41

(c) Algorithm 3, Scenario S, released location C, failed

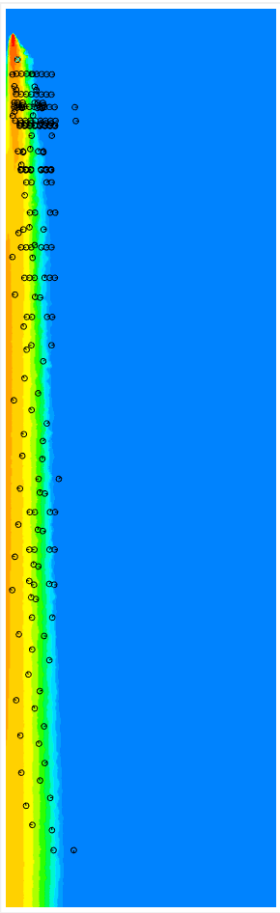
(d) Algorithm 4, Scenario S, released location D, failed



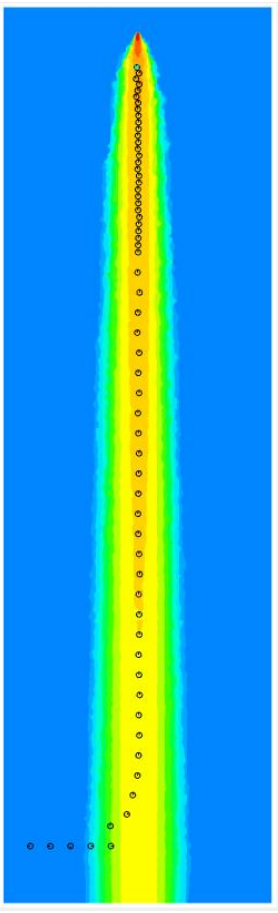
(e) Algorithm 5, Scenario M, released location B, step number 62



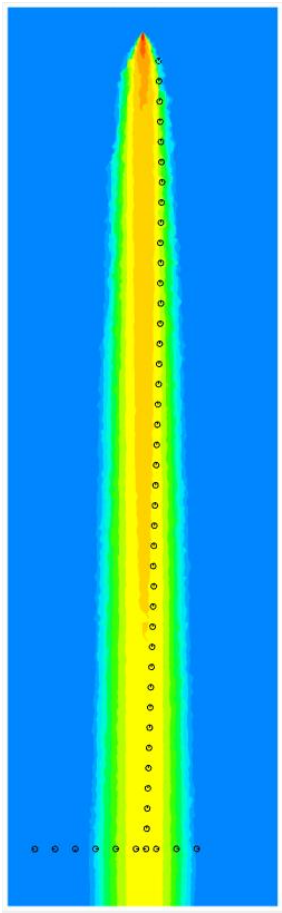
(f) Algorithm 6, Scenario M, released location B, step number 83



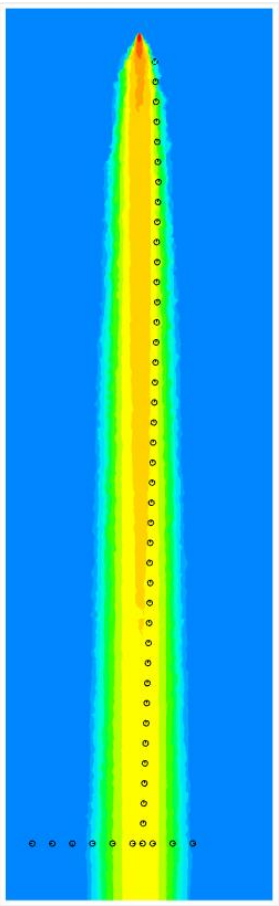
(g) Algorithm 7, Scenario S, released location B, step number 203



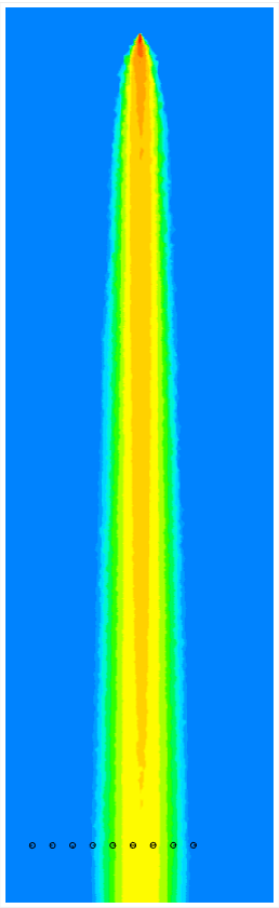
(h) Algorithm 8, Scenario M, released location B, step number 63



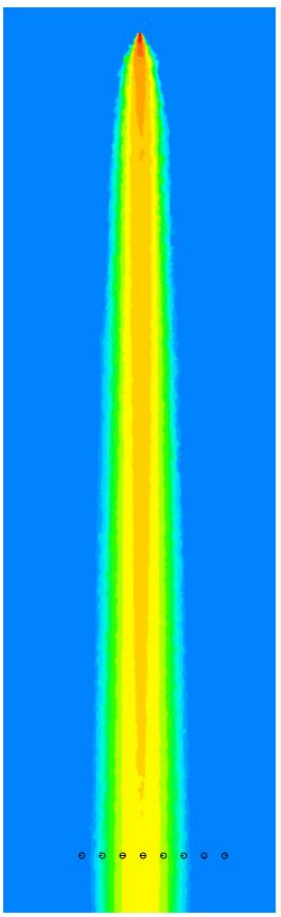
(i) Algorithm 9, Scenario M, released location D, step number 49



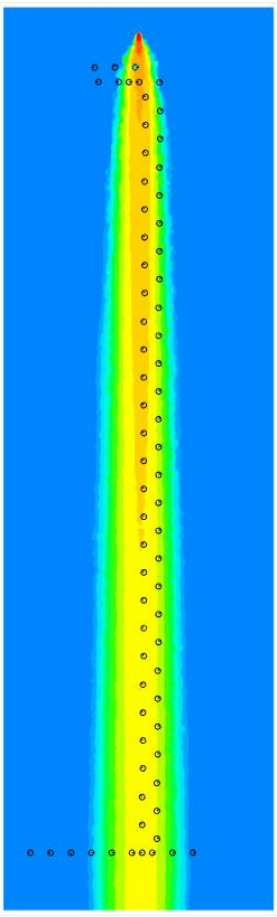
(j) Algorithm 10, Scenario M, released location B, step number 49



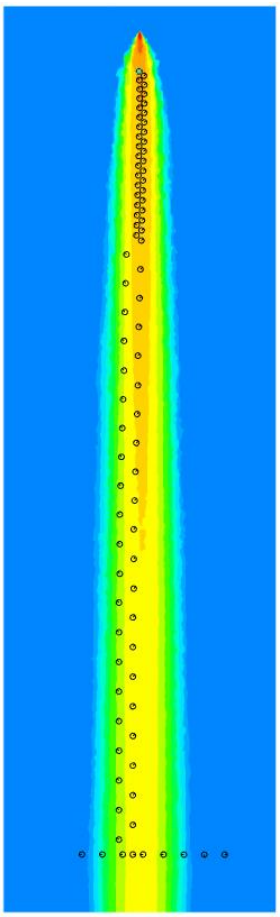
(k) Algorithm 11, Scenario M, released location B, failed



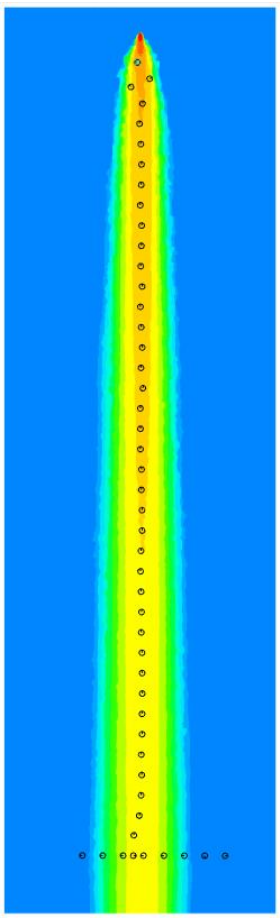
(l) Algorithm 12, Scenario M, released location D, failed



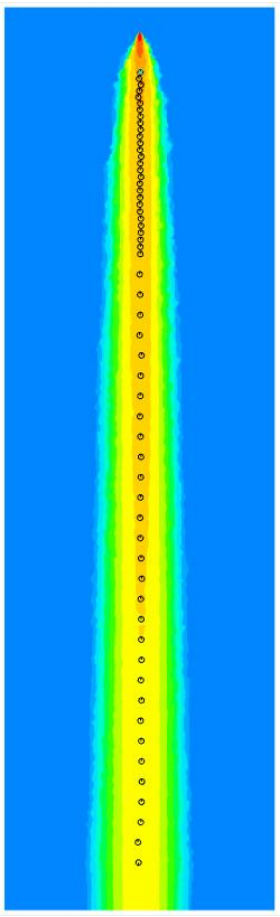
(m) Algorithm 13, Scenario M, released location B, step number 72



(n) Algorithm 14, Scenario M, released location D, step number 86



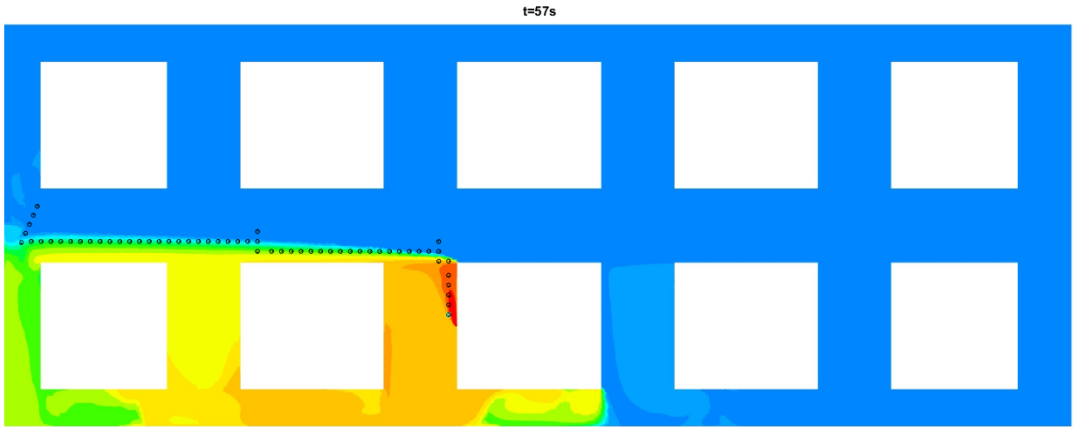
(o) Algorithm 15, Scenario M, released location D, step number 49



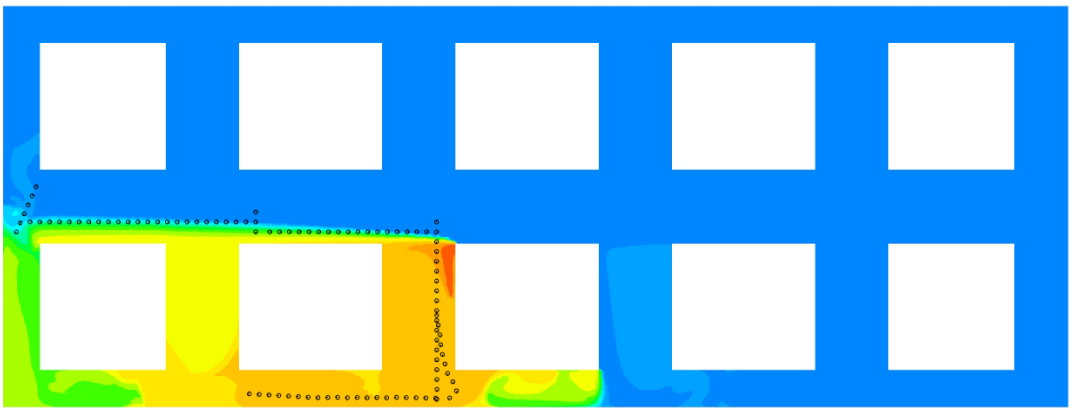
(p) Algorithm 16, Scenario M, released location C, step number 58

Appendix B Case 2

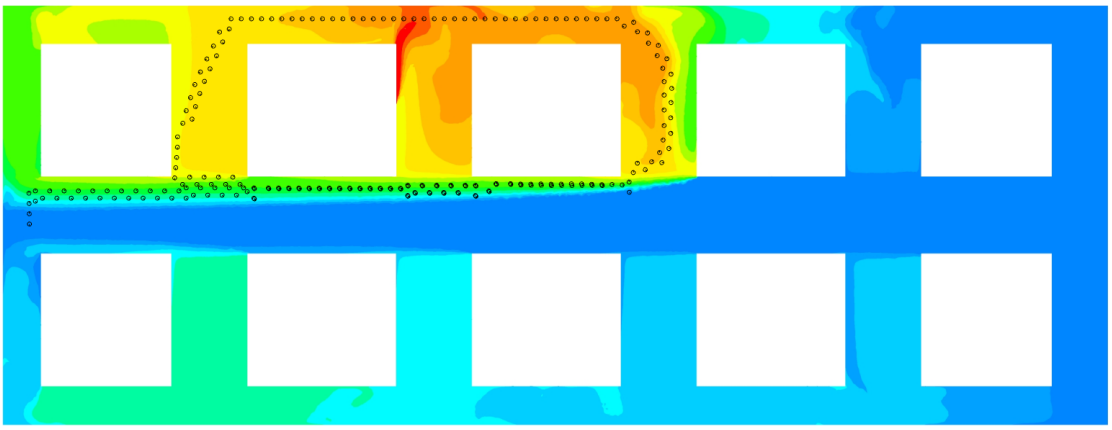
In Appendix B, supplementary results of testing the performances of different plume-tracing algorithms in the underground warehouse are provided. The algorithm No. refers to Tables 5.1 and 5.3.



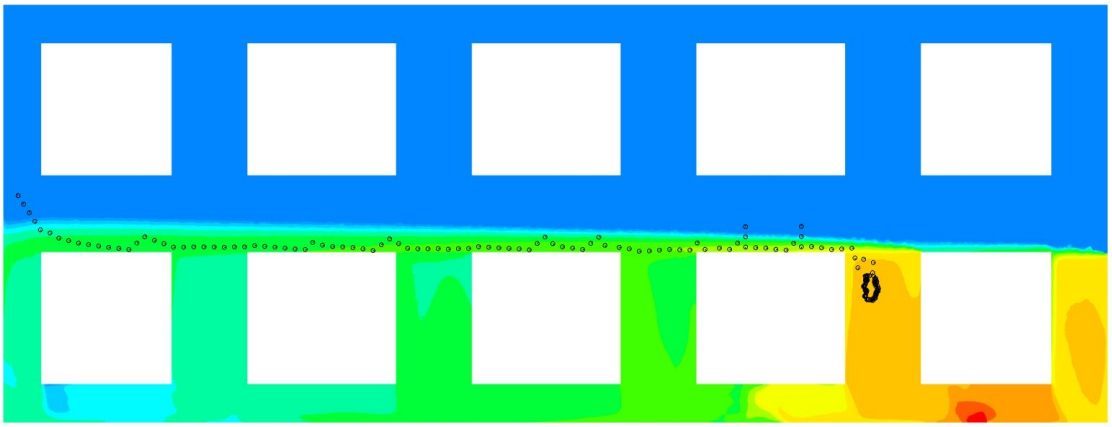
(a) The trajectory of the robot with Algorithm 1 searching in Scenario 2 (t= 57 steps)



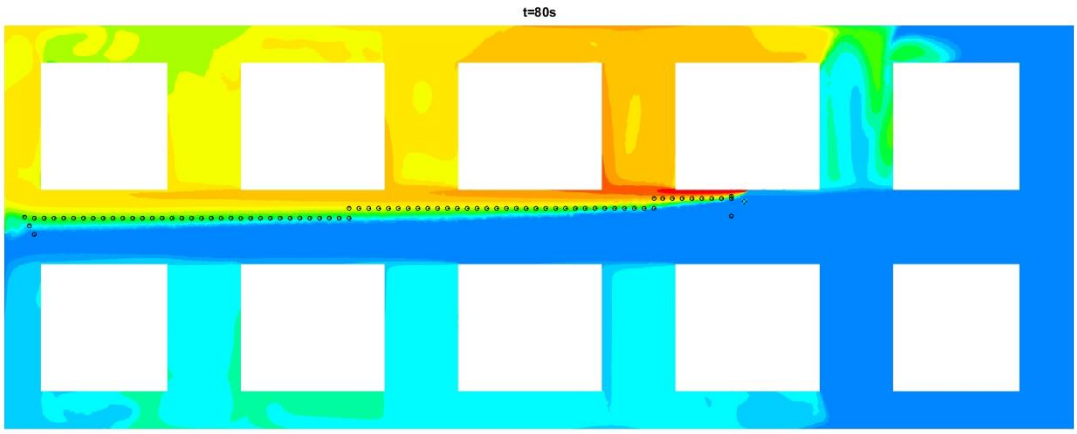
(b) The trajectory of the robot with Algorithm 2 searching in Scenario 2 (failed)



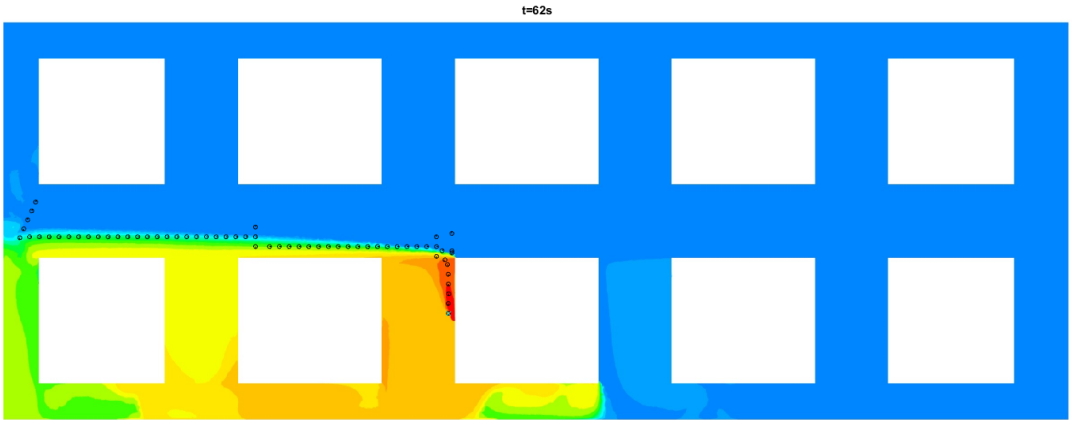
(c) The trajectory of the robot with Algorithm 3 searching in Scenario 1 (failed)



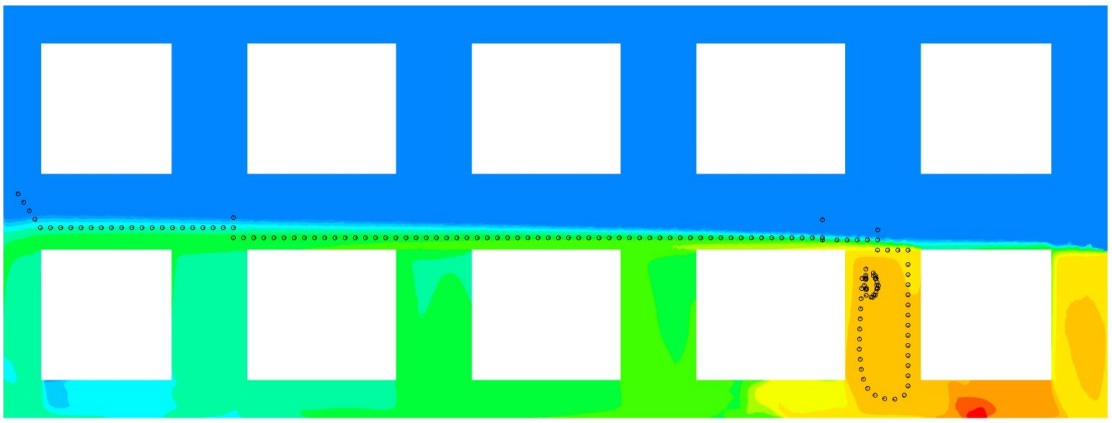
(d) The trajectory of the robot with Algorithm 4 searching in Scenario 4 (failed)



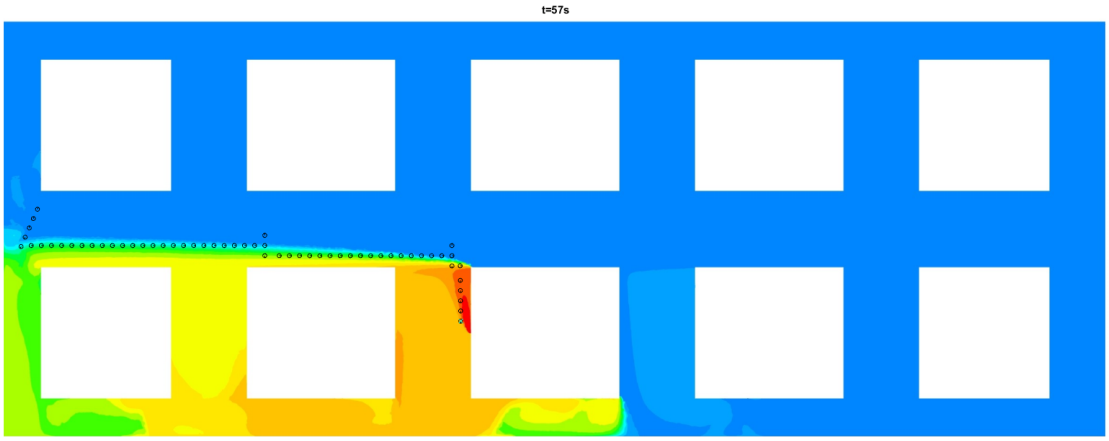
(e) The trajectory of the robot with Algorithm 5 searching in Scenario 3 (t= 80 steps)



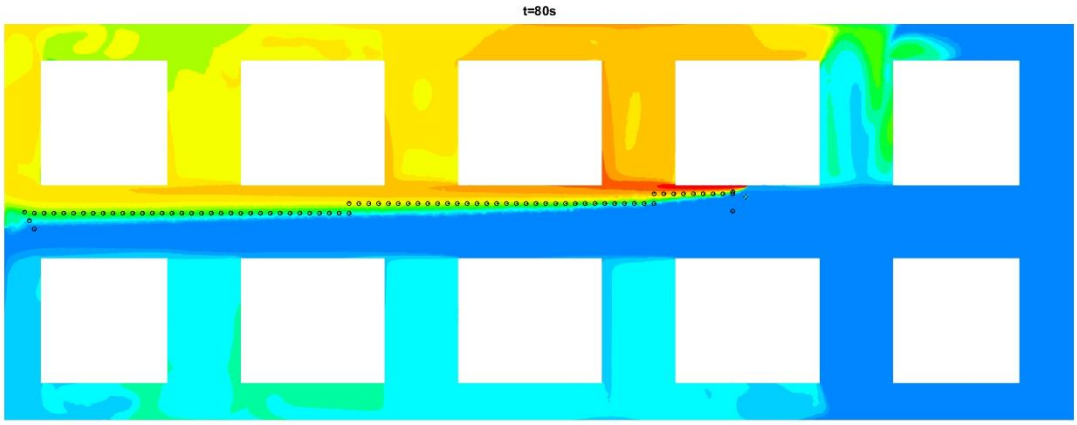
(f) The trajectory of the robot with Algorithm 6 searching in Scenario 2 (t= 62 steps)



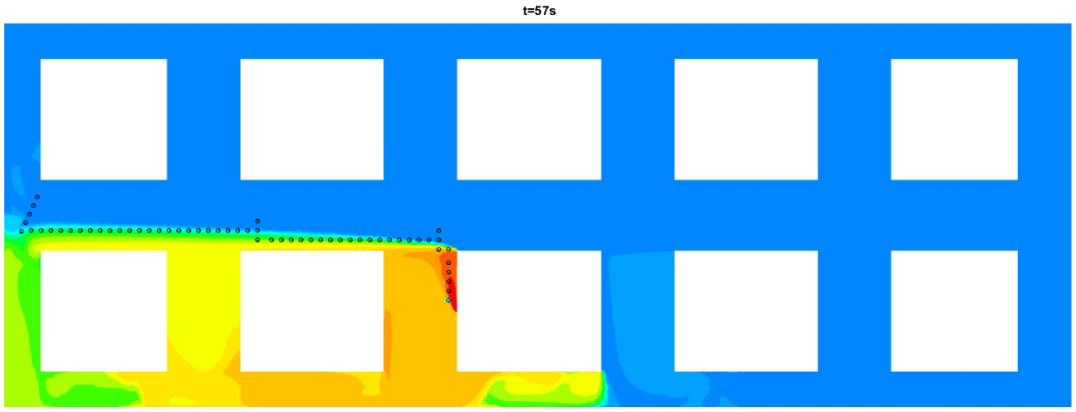
(g) The trajectory of the robot with Algorithm 7 searching in Scenario 4 (failed)



(h) The trajectory of the robot with Algorithm 8 searching in Scenario 2 (t= 57 steps)



(i) The trajectory of the robot with Algorithm 9 searching in Scenario 3 (t= 80 steps)



(j) The trajectory of the robot with Algorithm 10 searching in Scenario 2 (t= 57 steps)

Appendix C Journal article published

In Appendix C, the journal article that has been published with *Building and Environment* is attached [25].

Building and Environment 173 (2020) 106746

Contents lists available at ScienceDirect

Building and Environment

journal homepage: http://www.elsevier.com/locate/buildenv

Assessment of different plume-tracing algorithms for indoor plumes

Zeqi Li [a], Zhao Feng Tian [a,*], Tien-fu Lu [a], Houzhi Wang [b]

[a] School of Mechanical Engineering, The University of Adelaide, Adelaide, SA, 5000, Australia
[b] School of Chemical Engineering and Advanced Materials, The University of Adelaide, Adelaide, SA, 5000, Australia

ARTICLE INFO

Keywords:
Plume-tracing algorithm
Robot
CFD
Indoor environment
Plume

ABSTRACT

Bio-inspired chemical plume-tracing methods have been applied in robots to detect chemical emissions and to localise the plume sources in both indoor and outdoor environments. In the first part of this study, a comparison of performance of several widely used plume-tracing algorithms was conducted. A plume-tracing algorithm can be divided into three stages for analysis: plume sensing, plume tracking and source localisation. These algorithms, which had been previously presented and tested in either simulation framework in 2D scenarios or experiments, were tested and compared in two different 3D scenarios in this study. In one scenario, a chemical source is located away from walls in a channel and in the other scenario, the chemical source is located near a wall. This is the first time that the performance of different plume-tracing algorithms in wall plumes has been tested and assessed and included in the literature. Sixteen different algorithms were tested and compared and the algorithm constituted by *normal casting*, *surge anemotaxis* and *normal stepsize* performed the best among all. In the second part of the study, this algorithm was further optimised by an 'along-wall' obstacle avoidance method and finally a novel algorithm, named *vallumtaxis*, was proposed and shown to achieve higher efficiency.

1. Introduction

Unmanned robots have been used globally for detecting plumes and localising sources of toxic gas leakages, fires and explosives in buildings in order to protect people from hazardous circumstances [1–10]. These robots have made good progress in disaster avoidance in both indoor and outdoor environments. The source of a hazardous plume, which is defined as a mixture of chemical molecules in fluid movement downstream of the source, needs to be located as quickly as possible to maintain the safety of indoor environments. With better algorithms, plume-tracing mobile robots, which are already used to localise the indoor gas sources to reduce the health risk in indoor environments, are promising to provide an effective approach to preventing hazards.

As a large and growing body of literature has reported, insects like moths and underwater organisms like lobsters are able to track odour plumes in air or underwater [11]. For instance, researchers have undertaken studies on male moths' behaviours when they search for the source of released pheromones in plumes [12–14]. Subsequent investigations further developed plume-tracing algorithms following the strategy of surging upwind and have applied the algorithms to different mobile robots. The plume-tracing algorithms, which have proved to be useful search tools, are used to help people search for various gas sources in indoor environments. Generally, a typical plume-tracing algorithm can be divided into three stages: plume sensing (PS), plume tracking (PT) and source localisation (SL)[1] and the boundaries between them are defined by chemical concentration [15–18]. The first stage, called plume sensing, compounds different types of casting behaviours aiming to find the plume under the condition that the robot is outside the plume [19]. Plume tracking is the process in which the robot remains within the plume and manoeuvres towards the source. Normally after plume tracking, the source localisation stage takes place when the robot is deemed to be near the source and prepares to declare the location of it.

The performance of algorithms is different in diverse environments, consequently, in order to get better algorithms with higher efficiencies, comparisons between different plume-tracing algorithms continued experimentally and numerically. A comparison across four plume-tracing algorithms in the localisation of a source of an ion plume was carried out in experiments by Harvey, Lu and Keller [20]. This series of experiments was conducted in a wind tunnel with an ion generator in the central part of a wind tunnel and the wind speed was set at three different values: 0.55 m/s, 0.95 m/s and 1.4 m/s. The *surge anemotaxis*, which navigates the robot surging upwind continuously, proved to be

* Corresponding author.
E-mail address: zhao.tian@adelaide.edu.au (Z.F. Tian).
[1] PS: plume sensing, PT: plume tracking, SL: source localisation, Colours are used in print for all Figs. 1, 2, 5–15

https://doi.org/10.1016/j.buildenv.2020.106746
Received 10 October 2019; Received in revised form 10 February 2020; Accepted 12 February 2020
Available online 20 February 2020

the most effective algorithm of four: namely *surge anemotaxis*, two *bounded search* and *counterturning* (called *zigzags* in this paper) [20]. A subsequent comparison by Harvey, Lu and Keller [21] was undertaken in the same wind tunnel with the same location of the source but with a shifting wind field. According to the experimental results, by improving simple algorithms on revising turning angles, the performances of robots could be modestly improved in a shifting wind field [21]. Another comparison across four algorithms: the *E. coli algorithm*, the *silkworm moth algorithm*, the *dung beetle algorithm* and the *gradient-based algorithm*, was undertaken by Russell, Bab-Hadiashar, Shepherd and Wallace [16]. The *E. coli algorithm*, with a randomly-moving mechanism towards the direction of high concentration was identified as requiring the least effort in sensing and in controlling the moments of the robot [16]. However, being tested in a small scale with a laminar flow field, this algorithm is only proven to be effective to very small scale systems or in the areas with low turbulence affecting the concentration distribution [16]. Furthermore, the *silkworm moth algorithm*, which is similar to the *surge anemotaxis* performed by Harvey, Lu and Keller [20], was identified as being capable of supporting a robot to search in a turbulent chemical plume [16]. Using information of the gradient of concentration, another novel plume-tracing algorithm called the *pseudo gradient-based algorithm*, with a special PT stage, was investigated by Neumann, Hernandez Bennetts, Lilienthal, Bartholmai and Schiller [22] and this algorithm provided a novel pathway to optimise plume-tracing algorithms. Navigated by this algorithm, a robot moves tending to the area with a higher chemical concentration based on surging upwind at PT stage and this research concluded that the application of this *pseudo gradient-based algorithm* on a micro-drone in an outdoor environment is feasible. Also, apart from PT stage, to enhance SL particularly, Lu [23] proposed a novel strategy where the robots move more slowly when the gas concentration increases as the robot approaches the source to reduce the possibility of missing the source when surging. Other researchers also tried to combine visual detection and plume-tracing methods together to estimate a more precise position of plume sources [24]. To summarise, various attempts have been made by researchers to design novel algorithms or to improve existing ones. However, these efforts aimed for either free plumes such as plumes in an outdoor environment or indoor plumes where the chemical sources are located away from walls. None of these projects were targeted to discover how plume-tracing algorithms and robots perform when a plume source is on or near a wall and a wall plume is formed. One common example of plume sources located near or on a wall is a smouldering source on an indoor wall; such smouldering can be caused by a short-circuit cable [25]. Another possible scenario of plume sources near or on a wall is a gas leakage from pipes attached to walls in indoor environments such as warehouses. Plume-tracing robots that aim to be applied in indoor environments should be able to avoid obstacles such as walls but meanwhile be able to trace the wall plumes efficiently. Hence, investigations about how the plume-tracing algorithms and robots perform when localising the chemical sources of wall plumes in indoor environments are needed.

Recently, apart from experiments, simulation framework has been well developed to test and train plume-tracing algorithms [6,18,23,26,27]. Simulations conducted in an environment that is created by computational fluid dynamics (CFD) provide a novel and repetitive means to testing and training robots with different plume-tracing algorithms. One such simulation framework, combining Fluent (ANSYS Inc.) and Matlab (Mathworks Inc.), was presented by Liu [5], Liu and Lu [6], Liu and Lu [26]. Validation has confirmed that the combination of a CFD-produced environment and Matlab based robot(s) with plume-tracing algorithms is appropriate and applicable. The CFD-generated data, including gas concentration distribution, wind field and geometry information, were imported to Matlab to be accessed when needed. Initially, the simulation framework was 2D meaning that the data produced in CFD were from only one particular height focused for the simulation. For instance, Lu [23] did a case study taking advantage of this simulation framework to investigate the effects of initial location, orientations and surge distance on the performance of plume-tracing algorithms and robots. It was concluded that there are neither no obvious advantageous initial positions nor orientations but that the surge distance affected both the search time and the success rate. However, data including gas concentration distribution and wind field from 2D simulation framework is not sufficient as all plumes are 3D. A 3D simulation framework making use of data from 3D CFD results was then developed by Awadalla, Lu, Tian and Dally [28] to create a more realistic environment for testing and training plume-tracing robots. Subsequent research concerning 3D scenarios can be found in the studies conducted by Soares, Marjovi, Giezendanner, Kodiyan, Aguiar, Pascoal and Martinoli [29]; Neumann, Kohlhoff, Hüllmann, Lilienthal and Kluge [30]; Eu and Yap [31].

Employing the 3D simulation framework, this study assesses the performance of several promising plume-tracing algorithms in localising plume source in two indoor scenarios. Air is flowing in a wind-tunnel-like channel in both scenarios. In one scenario, the plume source is located at a point on the centreline of the confined environment while in the other scenario, the source is located next to one wall of the channel and a wall plume is formed from this source. Two types of PS and SL, four types of PT and thus 16 different plume-tracing algorithms with a specially designed concentration-based obstacle avoidance method were applied and tested in both scenarios. Robots with each different algorithm were released from five different initial locations. After testing and analysing all the plume-tracing algorithms, the one with the lowest total number of steps and consequently the highest efficiency from amongst the 16 algorithms was identified and further improved with a special 'along-wall' obstacle avoidance method. Equipped with this obstacle avoidance method, the novel algorithm, named *valluntaxis*, was confirmed to achieve higher efficiencies than all the other algorithms under comparison.

2. The simulation framework validation

Experimental validation of the simulation framework was first carried out by Liu and Lu [26] by comparing the trajectories of robots between experimental results in an office with an ion source and a fan providing an ion plume and simulation results in a CFD-predicted virtual environment. Subsequent validations between the simulation framework combining CFD and Matlab were also carried out for both indoor and outdoor experiments with or without obstacles by Liu and Lu [6] and Lu [4]. In this study, a further validation was conducted based on the experiments undertaken by Harvey, Lu and Keller [20]. Fig. 1 (a) and (b) represent the trajectories of a robot in the experiments and simulation framework. In the experiments, the arena is a wind tunnel and the size of the area provided for experiment is approximately 2.7 m (metres) in the X direction and 2.4 m in the Y direction (See Fig. 1) under the condition of neglecting the marginal area that the robot didn't get. The arena in simulation framework is modelled as a wind-tunnel-like channel whose size is 2.7 m in the X direction and 2.4 m in the Y direction (see Fig. 1). In both arenas, there is an inlet boundary providing steady air flow and an outlet boundary on the opposite side. The wind direction can be seen in Fig. 1. The wind speed in the CFD-predicted model was also set to be the same as that in the experiments (0.55 m/s). The ion generator, which is the plume source, is cylindrical (50 mm height and 5 mm diameter) and it emits ion at a mass flow rate of 0.449 g/s to ambient directions through the cylindrical side. They together provided an environment where the ion concentration distribution and the flow field are similar to that in the experiments. Data including flow field and ion concentration distribution generated in Fluent 2019 R1, were imported into Matlab R2018b for accessing. Fig. 1 (a) and (b) show that the trajectories of the robot in the experiments and simulation framework are quite close. Hence, the simulation framework can be regarded as a useful tool for training and testing the plume-tracing robots in this study.

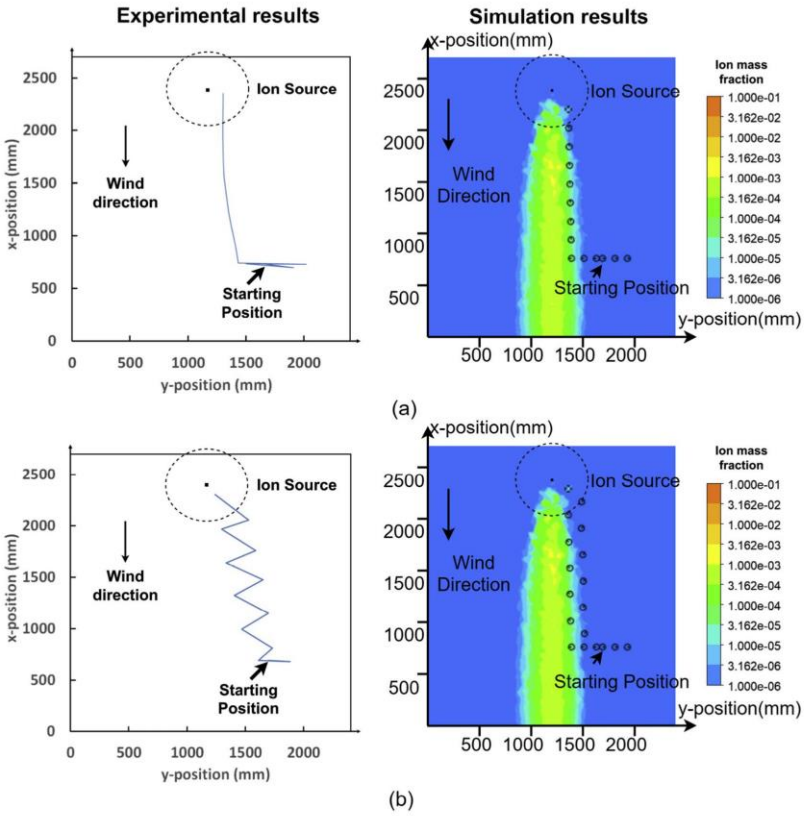


Fig. 1. (a) Surge anemotaxis robot trajectory in experiment (after Harvey, Lu and Keller [20]) and a virtual environment, (b) Zigzags robot trajectory in experiment (after Harvey, Lu and Keller [20]) and a virtual environment.

3. Simulation setup

3.1. Scenarios

Fig. 2 (a) and (b) present two scenarios investigated in this paper, both representing a flow confined in a long wind-tunnel-like channel. This long channel is specially designed for testing and training plume-tracing robots referring to the domain used by Harvey, Lu and Keller [20] and Lochmatter, Raemy, Matthey, Indra and Martinoli [32]. The dimension of the domain in each scenario is 2400 mm (millimetres) in the X direction and 8000 mm in the Y direction. Each boundary of this channel is set as a 5-pixel-thick blank area for testing the obstacle avoidance method (see Section 3.2). In this research, the source of the plume is an ion generator, which was modelled following the real ion generator applied in our previous investigations [20,21,33]. In scenario M, the coordinate of the centre of the source is (1200 mm, 7715 mm, 175 mm), which is 1200 mm away from both side walls and in scenario S, the source is located at (80 mm, 7715 mm, 175 mm), which is 80 mm away from the left side wall. In both scenario M and scenario S, there is an inlet boundary and an outlet boundary contributing to a steady wind flow. The air inlet velocity in both scenarios is 1 m/s. A plume that mixes air and ions forms downstream the source. Five initial locations A-E are shown in Fig. 2 and their coordinates are A(304 mm, 459 mm), B(650 mm, 546 mm), C(1238 mm, 459 mm), D(1819 mm, 546 mm) and E (2165 mm, 459 mm), respectively. The ion concentration distribution along a horizontal plane 15 cm above the ground is shown in Fig. 2. The colour legend of ion mass fraction in air in Fig. 2 is logarithmic for better visualisation and it applies to all the following figures in this paper. When the robot declares a location where the distance between the location and the source is less than 350 mm in real scale, which follows the previous experimental investigation undertaken by Ref. [20], it is considered as a successful searching.

The simulation framework applied in this research is based on steady state CFD simulations using a Reynolds-averaged Navier-Stokes (RANS) model (k-ω-SST model). The simulated plumes are therefore continuous and the turbulence field is a Reynolds-averaged field not changing with time. The advantage of this CFD approach is that the exact same flow fields and the plumes are used to compare different plume-tracing algorithms. These consistent environments make the results fair and repeatable. This CFD approach has been adopted widely in indoor airflows and the plume-tracing algorithms research, for example, [1,5,8,13,23,26,27,34–37]. Moreover, it has been found to be valid as shown in the validations of simulation framework reported in the paper.

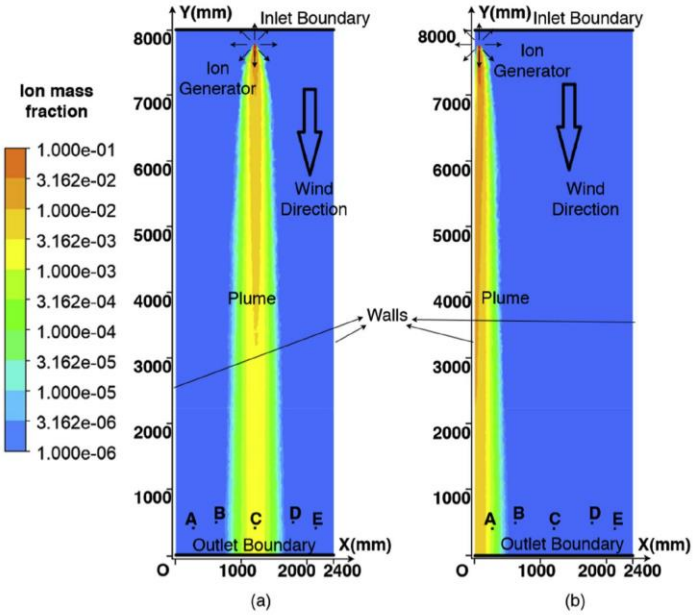


Fig. 2. Simulation setup (a) scenario M (b) scenario S.

3.2. The robot

The plume-tracing robot simulated in this study is a differential drive robot equipped with chemical, wind and ultrasonic sensors and it is designed referring to the robots applied in the studies undertaken in our group by Awadalla, Lu, Tian, Dally and Liu [18] and Lu [23]. As shown in Fig. 3, two driving wheels are set on two sides of the robot to drive the robot and two small idle wheels are set in the front and back to balance the robot. Please note that for simplicity, the control box of the robot and batteries are not included in Fig. 3. Different from the previous robots used by Awadalla, Lu, Tian, Dally and Liu [18] and Lu [23] that only one fixed chemical and one fixed wind sensor are applied, the robot in this case study is equipped with multiple chemical sensors on the edge for plume tracing and obstacle avoidance (shown in Fig. 3). Moreover, unlike the previous robots, the location of the sensors on the current robot is not fixed and all the sensors can move vertically along two cylindrical holders to work at different horizontal levels. Chemical sensors are equipped to measure the local chemical concentration and wind sensors are equipped to measure the local wind speed and direction. As shown in Fig. 3 (a) and (b), one chemical sensor and one wind sensor are located at the centre of the robots and four chemical sensors are located in four directions: left, right, forward and back on the edge of the robot. Four ultrasonic sensors are used to measure the shortest distance between robot and an obstacle in their covering sections term as section L, ML, MR and R (illustrated in Fig. 3). To avoid striking on obstacles, a novel concentration-based obstacle avoidance method was proposed by using the information from ultrasonic sensors and the chemical sensors on the edge. Here, *responding distance* is defined as the sum of the surge distance and the diameter of the robot. If an ultrasonic sensor measures a distance longer than the *responding distance*, an obstacle is confirmed in its corresponding covering section. When the ultrasonic sensors detect the existence of obstacle(s), the robot will tend to move away from the obstacle in order to leave sufficient space for the robot to surge following plume-tracing algorithms at the next step. For instance, when the robot is moving towards an obstacle wall in a direction normal to the wall, ultrasonic sensors will detect the wall and the chemical sensors on the edge will help navigate the robot to get away from the wall with respect to the concentration distribution around the robot.

4. Plume-tracing algorithms

4.1. Three stages

A plume-tracing process can be divided into three stages for discussion: plume sensing, plume tracking and source localisation [15–18]. In this study, the transition between different stages is defined by local chemical concentration. Two concentration thresholds, *threshold I* and *threshold II* are set to distinguish different stages. Fig. 4 explains how a robot with a three-stage plume-tracing algorithm works. When the plume-tracing robot starts working, it firstly detects the existence of obstacles. Having sufficient space for plume-tracing, it will then measure the local concentration. If the local concentration is below *threshold I*, the robot is seen to be outside the plume, so the robot will move in PS stage. If the local concentration is between *threshold I* and *threshold II*, PT will be triggered. If the local concentration is higher than *threshold II*, the robot is supposed to be near the source and should get ready for source declaration thereby moving at SL stage. In this study, the values of the ion mass fractions of *threshold I* and *threshold II* are set to be 10^{-5} and 0.1, respectively. *Threshold I* and *threshold II* are set to distinguish different stages of plume tracing. It can be seen in Fig. 2 that 10^{-5} for ion mass fraction, which is the *threshold I* in this study, defines a clear boundary of the plume and provides a reasonable area for plume tracing. While when the local ion mass fraction is higher than *threshold II*, which is 0.1 in this study, the robot is believed to be significantly close to the source (in this study, lower than about 1000 mm according to the concentration distribution). When applied in other cases, the values of these two thresholds can be set case by case according to different chemical concentration and environments. *Threshold I* is decided following the

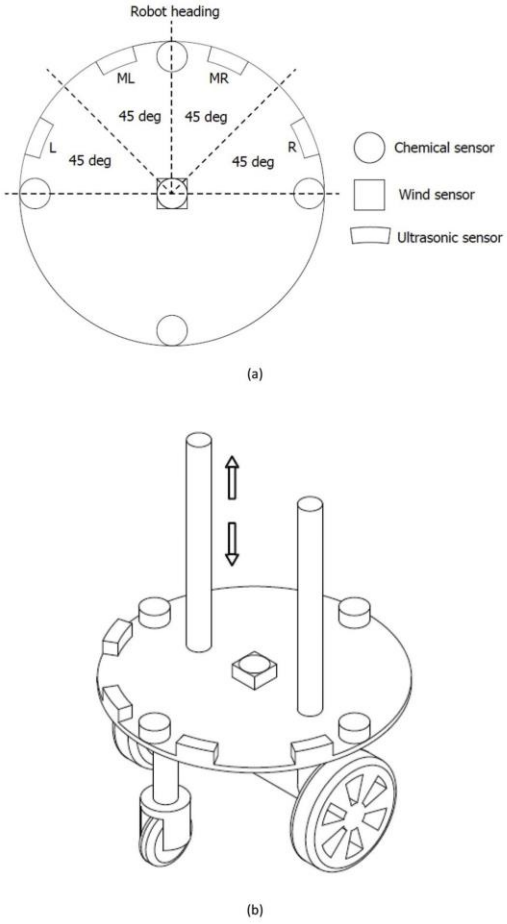


Fig. 3. An overview of the robot (a) top view (b) isometric view.

criteria: higher than the low limit of the range of the chemical sensor and the normal concentration of the target chemical in the environment. *Threshold II*, which distinguishes PT and SL stage, defines an area near the source and this area vary with different source strengths. Nevertheless, the robot keeps moving towards the source at either PT or SL stage and the conclusion of the comparison of different algorithms is believed to be consistent.

Please note that for all Figs. 5, 8, 10–12, 14, the background contour represents the chemical concentration distribution along the horizontal plane, where the simulation data was used by the robot at the last step. Fig. 5 (a) and (b) show examples of how the plume-tracing robot works in two different scenarios. In both scenarios, the robot was released at a location outside of the plume. In both scenarios, as the chemical concentration at the releasing location is lower than *threshold I*, the robot began plume sensing to look for the plume. In scenario M shown in Fig. 5 (a), the robot moved left and right searching for the plume. After four steps, the chemical sensors detected that the ion concentration was higher than 10^{-5}, then the robot started plume tracking stage. The robot moved upward at a constant step length until the chemical sensors

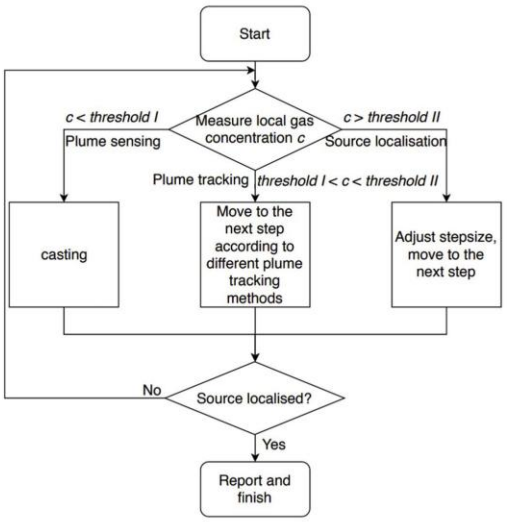


Fig. 4. The logic flow chart of a typical plume-tracing algorithm.

detected that the ion concentration was higher than 0.1 at a location around y = 6800 mm. Then the robot started source localisation stage and reduced the length of steps. After 12 steps, the distance between the robot and the source is less than 350 mm, and the robot reported that the source was detected. In scenario S, the robot found and left the plume four times when searching because of either wind direction or obstacle avoidance and finally it ended without SL because the robot didn't get contact with the area where the chemical concentration is above *threshold II*.

4.2. Plume sensing

Plume sensing is the process whereby the robot that is initially located outside a plume moves to find the plume that comes from a chemical source. This process, inspired by insects, is also termed as 'casting'. There are two types of casting, namely *normal casting* and *special casting*, in this study and have been proposed in previous literature. The performance of these two casting methods is compared by searching in both the wall plume (scenario S) and the free stream plume (scenario M).

4.2.1. Normal casting

This first kind of casting method, called *normal casting* in this paper is the same as the casting method used in the experiments undertaken by Harvey, Lu and Keller [20]. During the *normal casting* process, the robot moves in a direction that is perpendicular to the wind direction and moves forth and back in a gradually increasing step length at each step. Meanwhile, the chemical sensor keeps measuring chemical concentration and once the concentration is above *threshold I*, the robot is seen to enter a plume through one plume boundary and it will immediately stop PS and begin the next stage, either plume tracking or source localisation, depending on whether the chemical concentration is lower or higher than *threshold II*.

4.2.2. Special casting

Under the *special casting* process, the robot searches for the centreline of a plume, which is located in the middle of two well-developed boundary layers of a plume [38,39]. Differing from the *normal casting*,

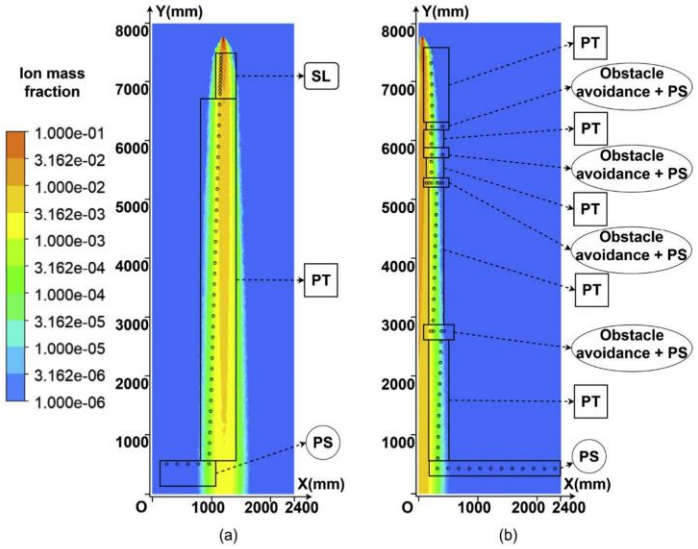


Fig. 5. (a) An example of how the plume-tracing robot works in scenario M, (b) An example of how the plume-tracing robot works in scenario S.

in the *special casting* the robot keeps moving straight aiming to find the next boundary of the plume after entering the plume through one plume boundary. As long as the concentration is above *threshold I*, the robot continues to move across the wind and the plume until the concentration is again lower than *threshold I* or the robot meets a wall, meaning that the other boundary of the plume is found. Then the robot moves back to the centre between the two boundaries and starts the next stage. The advantage of the *special casting* is that the initial position of the next stage to be commenced will be near the centreline of the plume and theoretically the possibility of leaving the plume when surging upwind is lower than that when moving near the plume boundary, which is normally found in the *normal casting* cases.

4.3. Plume tracking

Plume tracking is the process by which the robot surges within the plume after finding it and normally takes a longer time thereby influencing the efficiency significantly (see Fig. 5). All the PT methods compared in this study were previously presented either in experiments or simulation framework and were already proved to be applicable (See Section 1). The distance that the robot surged at every step was constant.

4.3.1. Surge anemotaxis

The *surge anemotaxis* is one of the simplest of the implemented algorithms and it was identified to perform well in experiments [20]. Here, *surge* is defined as moving step by step. When the local chemical concentration is above *threshold I*, the robot surges towards the wind direction measured by the wind sensor. Under the *surge anemotaxis*, the robot continuously adjusts its direction to ensure orienting to the wind every step and then surges upwind.

4.3.2. Chemotaxis

The *chemotaxis* is defined as surging towards the area with the highest chemical concentration regardless of wind direction, i.e. the robot will not surge based on wind direction but will surge based on chemical concentration. In this study, there are four chemical sensors located on the edge of the robot (explained in Section 3.2) and the robot moves in the direction of the sensor that reports the highest chemical concentration among four. For example, when the robot is moving, if the sensors detect the highest concentration in the forward direction, the robot will surge forward.

4.3.3. Zigzags

The strategy of this algorithm was inspired by the moving behaviours of Dung beetles. As its name indicated, the trajectories of the robot using this algorithm are like zigzags. The overall strategy is surging upwind, however, differing from the *surge anemotaxis*, after turning facing the wind, the robot continues to turn a certain angle and then surges. The turning angle is constant in most cases and in this study, 45° was selected.

4.3.4. Pseudo gradient-based algorithm

The idea of the *pseudo gradient-based algorithm* was first presented by Lilienthal and Duckett [40] and then it was developed into a formal algorithm and tested in the experiments undertaken by Neumann, Hernandez Bennetts, Lilienthal, Bartholmai and Schiller [22]. The strategy of this specially designed PT method is actually a combination of the *surge anemotaxis* and the *chemotaxis*. Under the *pseudo gradient-based algorithm*, after facing upwind, the robot continues to turn β toward the area with higher chemical concentration (as illustrated in Fig. 6). In general, β can be calculated by using the following equations:

$$\beta = \begin{cases} 0, & \mu = 0.5 \\ (1-2\mu)\times 90^{\circ}, & (turn\ left),\ \ \mu < 0.5 \\ (2\mu-1)\times 90^{\circ}, & (turn\ right),\ \ \mu > 0.5 \end{cases} \tag{1}$$

$$\mu = \frac{C_L}{C_L + C_R} \tag{2}$$

where μ is a coefficient that is used to determine the value of β. In previous experiments C_L and C_R are measured by moving left and right but in this study they are measured by taking advantage of multiple chemical sensors. C_L is the chemical concentration measured by the chemical sensor on the left hand side and C_R is measured by the one on the right hand side.

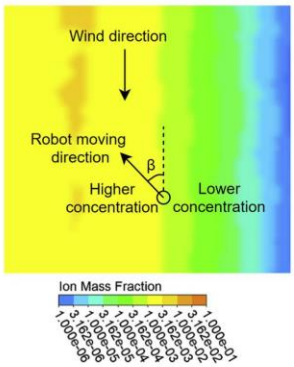


Fig. 6. Schematic diagram of the working principle of pseudo gradient-based algorithm.

4.4. Source localisation

Source localisation is the process of identifying and declaring the location of the source after the robot enters the plume zone where the chemical concentration is higher than *threshold II*. During the source localisation process, the robot can move either in a constant stepsize or variable stepsizes. In this study, stepsize is defined as the distance that the robot surges at each step.

4.4.1. Constant stepsize

For the *constant stepsize* SL, the robot moves in a constant stepsize that can be the same as that in the plume tracking stage. One drawback of the constant stepsize source localisation is that, when the stepsize is large, the robot may surge out of the plume when searching near the chemical source.

4.4.2. Variable stepsize

To solve the problem of the robot surging out of the plume near the chemical source, a novel source localisation method with a variable stepsize strategy was proposed by Lu [23]. During this *variable stepsize* SL, the higher the chemical concentration is, the shorter the distance the robot surges at each step. An equation between the gas concentration and surge distance was adopted as [23]:

$$XY = C \tag{3}$$

where X is the chemical concentration that is normalised by the highest concentration, i.e. the concentration at the source. Therefore, the X value is normalised within the range from 0 to 1. Y is the surge distance and C is a constant value that can vary according to different environments. Following the work of Lu [23], $C = 5$ was selected.

4.5. Table of algorithms

One of the main objectives of this research is to assess the performance of the algorithms with different PS, PT or SL stages in tracing for the sources of two plumes confined in a channel (scenario S and scenario M). Therefore, there are in total 16 different combined plume-tracing algorithms compared in this study. Table 1 shows a list of these 16 combined algorithms.

4.6. Stepsize

The stepsize (surge distance) in the experiments carried out by Harvey, Lu and Keller [20] was 18 cm. In the current study, four different stepsizes: 9 cm, 13 cm, 18 cm and 23 cm were tested and compared in two different scenarios using Algorithm 1 (please see Table 1 for the details of Algorithm 1). For example, the robots in both Fig. 5 (a) and (b) are with a stepsize of 18 cm. It can be seen in Fig. 7 that the total number of steps tends to decrease when the stepsize increases in scenario M. In scenario S, the total number of steps decreases when the stepsize increases from 9 cm to 13 cm, however, there are no obvious advantages when the stepsize increases from 13 cm to 18 cm. An accident happened when the stepsize increased to 23 cm as Fig. 8 shows. The robot got trapped and failed to approach the source because of the increasing *responding distance* for obstacle avoidance. To balance the performance of the robots in two scenarios, an 18 cm stepsize is selected. In this study, the robot is assumed to surge once at each time step referring to the work undertaken by Lu [23]. Therefore, the lower number of steps indicates the lower time spent thereby causes a higher efficiency.

Table 1
Different plume-tracing algorithms and their origins.

Algorithm No.	Plume Sensing	Plume Tracking	Source Localisation
1	Normal casting [20]	Surge anemotaxis [20]	Constant stepsize [23]
2	Normal casting	Surge anemotaxis	Variable stepsize [23]
3	Normal casting	Chemotaxis [15]	Constant stepsize
4	Normal casting	Chemotaxis	Variable stepsize
5	Normal casting	Zigzags [20,22]	Constant stepsize
6	Normal casting	Zigzags	Variable stepsize
7	Normal casting	Pseudo gradient-based [22]	Constant stepsize
8	Normal casting	Pseudo gradient-based	Variable stepsize
9	Special casting [38]	Surge anemotaxis	Constant stepsize
10	Special casting	Surge anemotaxis	Variable stepsize
11	Special casting	Chemotaxis	Constant stepsize
12	Special casting	Chemotaxis	Variable stepsize
13	Special casting	Zigzags	Constant stepsize
14	Special casting	Zigzags	Variable stepsize
15	Special casting	Pseudo gradient-based	Constant stepsize
16	Special casting	Pseudo gradient-based	Variable stepsize

5. Simulation results and comparison

In this section, an overall view of all results is presented, and subsequently whether initial locations matter and comparisons among different PS, PT and SL methods are discussed. Results shown in Fig. 9 (a) and (b) indicate that the robot's performances differ in different scenarios, PS, PT, SL stages and initial locations. The total number of steps of each algorithm with the *chemotaxis* for PT stage is not shown because of its unsatisfactory performance (see Section 5.3). It costs the

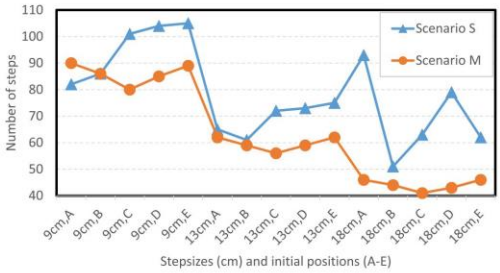


Fig. 7. The total numbers of steps in two scenarios with different stepsizes, searching with Algorithm 1.

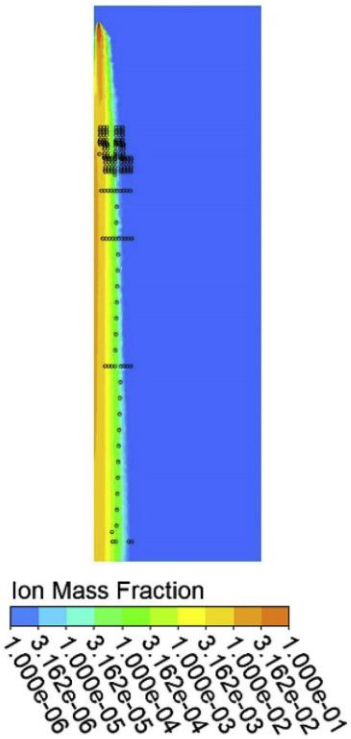


Fig. 8. The trajectory of robot in scenario S when stepsize is 23 cm, searching with Algorithm 1.

robot considerably more steps when searching in scenario S indicating that these algorithms need to be improved to fit the condition that the source locates near a wall. In Fig. 9, different line style means different SL stages. Solid lines represent the algorithms with *normal casting* while dash lines represent the algorithm with *special casting*. Similarly, the colour green represents the *zigzags*, yellow represents the *surge anemotaxis* and blue means the *pseudo gradient-based algorithm* for PT stage. Also, for the markers, the 'O' means the *constant stepsize* and 'X' means the *variable stepsize* for SL stage. In Figs. 10–12 and 14, 'algorithm', 'scenario', 'corresponding method', 'initial location' and 'total number of steps' are labelled following every figure. For instance, the trajectory of the robot coded with Algorithm 5 where the *normal casting* is applied and released from location E in scenario M is shown in Fig. 10 (a) and the total number of steps is 64.

5.1. Initial location matters?

The question whether there are advantageous initial locations when releasing a plume-tracing robot in an environment without prior knowledge was proposed by Lu [23] and simulation results indicated that there are no obvious advantageous initial locations for higher efficiency when releasing plume-tracing robot in a corridor in a given office-like indoor environment. However, the simulations conducted by Lu [23] were based on only one algorithm. To further discuss this problem, the robot was released from five different initial locations when testing different algorithms in the two scenarios in this study (These five initial locations are illustrated in Fig. 2.). From Fig. 9 (a) and (b), it is concluded that the total numbers of steps vary depending on different initial locations. In scenario M, it can be seen that for most algorithms, the robots released from location C, which is inside the plume, have lower total number of steps. However, in scenario S, even though location A is inside the plume, the robot released from location A shows no advantages in terms of total number of steps. Therefore, for an indoor environment that is similar to the one given in this case study, it may be concluded that for a plume that is away from walls, it is better to release the plume-tracing robot from a position that is inside the plume if possible. However, when there is a wall plume, statistical results in the study did not indicate that there are any obvious advantageous initial locations when searching with the algorithms and the obstacle avoidance method mentioned in this study.

5.2. Comparison of PS methods

According to the testing results, both *normal casting* and *special casting* prove to be capable of navigating the robot outside the plume to find the plume. Fig. 10 shows the trajectories of the robot in four selected cases; two cases with *normal casting* and two cases with *special casting*. For *normal casting* used in both scenarios (see Fig. 10 (a) and (c)), once the robot found the plume, it immediately tracked within the plume. While for *special casting* used in both scenarios (Fig. 10 (b) and (d)), the robot continued moving forward after entering the plume until it moved out of the plume (Fig. 10 (b)) or it reached the wall (Fig. 10 (d)), then it moved back to the centre of the plume and started PT. The results shown in Fig. 9 indicate that the robot with *normal casting* spends less time than that with *special casting* in total. The *special casting* did help the robot find the centreline of the plume but this did not reduce the total time cost. To summarise, the *normal casting* is selected in terms of total time cost.

5.3. Comparison of PT methods

In total, four plume tracking methods were tested and compared in this study. Among these four methods, the *Surge anemotaxis*, *zigzags* and *pseudo-gradient-based algorithm* proved to be capable of navigating the robot to approach the source of the plume (see Fig. 11). For *surge anemotaxis* as shown in Fig. 11 (a), after PS, the robot found the centreline of the plume and surged upwind for plume tracking. The trajectory of the robot was basically a straight line. Fig. 11 (b) and (f) show that for the single robot that is coded with the *chemotaxis* (illustrated in Section 4.3.2), the robot was trapped in the downstream of the plume after PS stage. In all the tests with the *chemotaxis* regardless of PS and SL, initial locations and scenarios, the robot failed in searching. A further examination of the trajectories of the robot with the *chemotaxis* shown in Fig. 11 (b) finds that the robot entered the plume at point 4, then it moved towards point 5 and then point 6 following the measured highest gas concentration. At point 6, the robot kept moving a step further to step 7 as the front chemical sensor detected the highest concentration. From step 6 to step 7, however, the robot passed the region where the highest concentration zone locates. At step 7, as the highest concentration was found at the back of the robot, the robot moved back to the location of step 6. The robot trajectory started repeating between step 6 and 7, and the robot was therefore trapped. The *chemotaxis* failed in scenario S as shown in Fig. 11 (f) as well. For the *zigzags* as shown in Fig. 11 (c), the trajectory of the robot is a zigzag in scenario M. However, in scenario S (see Fig. 11 (g)), the trajectory is initially a zigzag but then become complex after z = 5000 mm. This can be attributed to the fact that the distance between the robot and the wall was getting closer after z = 5000 mm, and the obstacle avoidance was frequently triggered therefore significantly increasing the number of steps. For the *pseudo gradient-based algorithm*, in scenario M, the robot moved along an approximately straight line firstly and then a zigzag after z = 7000 mm since the difference between the concentration measured by the chemical sensors on the left and right hand side is more significant. While in

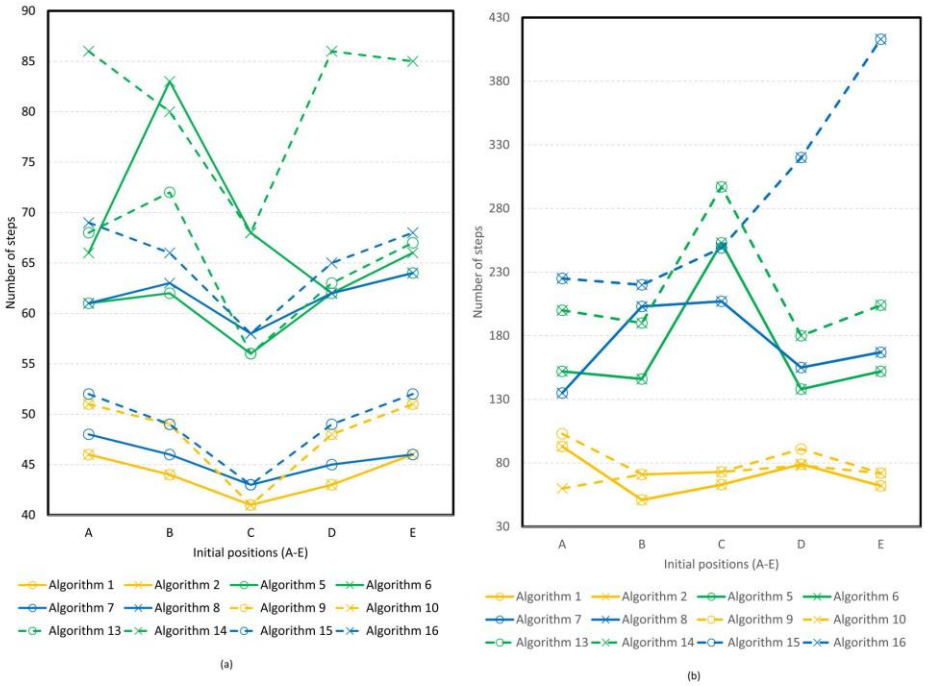


Fig. 9. (a) Statistical results of the number of steps in scenario M, (b) Statistical results of the number of steps in scenario S.
Note: Algorithms refer to Table 1, solid line – normal casting, dash line – special casting, yellow – surge anemotaxis, green – zigzags, blue – pseudo-gradient based algorithm, O – constant stepsize, X - variable stepsize.

scenario S, since the concentration near the wall is higher, the robot surged towards the wall firstly and then surged away from the wall because of obstacle avoidance. Hence, the total number of steps of the robots coded with the *pseudo gradient-based algorithm* in scenario were high. In summary, considering the stability and total time cost, the *surge anemotaxis* performs better than other methods in these two scenarios.

5.4. Comparison of SL methods

Two types of SL, a *constant stepsize* method and a *variable stepsize* method, were tested in this study. Compared with the trajectories of the robot with *constant stepsize* shown in Fig. 12 (a), the robot with *variable stepsize* (Fig. 12 (b)) moved in more but smaller steps when approaching the source in scenario M. When getting closer to the source, the stepsize decreased drastically. It did not perform as expected (the stepsize gradually reducing) like the simulation conducted by Lu [23], whose scenario was a larger indoor environment. In scenario S, there are no significant differences of trajectories and numbers of steps between the robots with *constant stepsize* and *variable stepsize* SL (see Fig. 12 (c), (d) and Fig. 9 (b)). It is because the motion of the robot is dominated by the obstacle avoidance motions when it moved close to the source in scenario S. This observation is also confirmed in Fig. 9. Generally, *variable stepsize* SL stage fits a large indoor environment such as the one presented by Lu [23]. However *constant stepsize* works better in the relatively smaller domain presented in this study. For the localisation of the indoor plume sources in this study, *constant stepsize* is selected for source localisation.

5.5. Vallumtaxis

Normally there is no prior knowledge about the possible location of the source when releasing a robot to undertake chemical plume tracing and subsequent source localisation tasks. As a result, a more comprehensive plume-tracing algorithm that is capable of navigating the robot in searching in indoor environments, where the sources are located on or near a wall, is needed especially when an urgent situation, which needs to be solved as soon as possible, emerges. Investigations and findings presented in the previous sections indicate that *normal casting*, *surge anemotaxis* and *constant stepsize* are the best for PS, PT and SL, respectively. Therefore, the combination of them, Algorithm 1 proves to perform the best and is selected for further development to a novel algorithm with a higher efficiency. From the trajectories of robot shown in Fig. 12 (a), it is concluded that the robot navigated by Algorithm 1 approaches the source in almost a straight-line trajectory. This makes the total search distance and time cost already very low. However, in scenario S, it can be seen that the concentration-based obstacle avoidance (explained in Section 3.2) slows down the robot by making it repeat PS (see Fig. 11 (g)). Actually the plume goes along the wall in scenario S so getting away from the wall is therefore leaving the plume. In this case, to achieve a higher efficiency, a novel algorithm called *vallumtaxis*, which is based on Algorithm 1, but with an 'along-wall' obstacle avoidance method, was proposed in this study. Obstacle avoidance methods using multiple ultrasonic sensors have already been applied in many studies [1,18,23,28]. However, in previous studies, when detecting a wall, a robot would try to move away from the wall by moving back. Due to the characteristics of wall plumes, it was found in this study that it is more efficient for a plume-tracing robot to move along the wall when wall

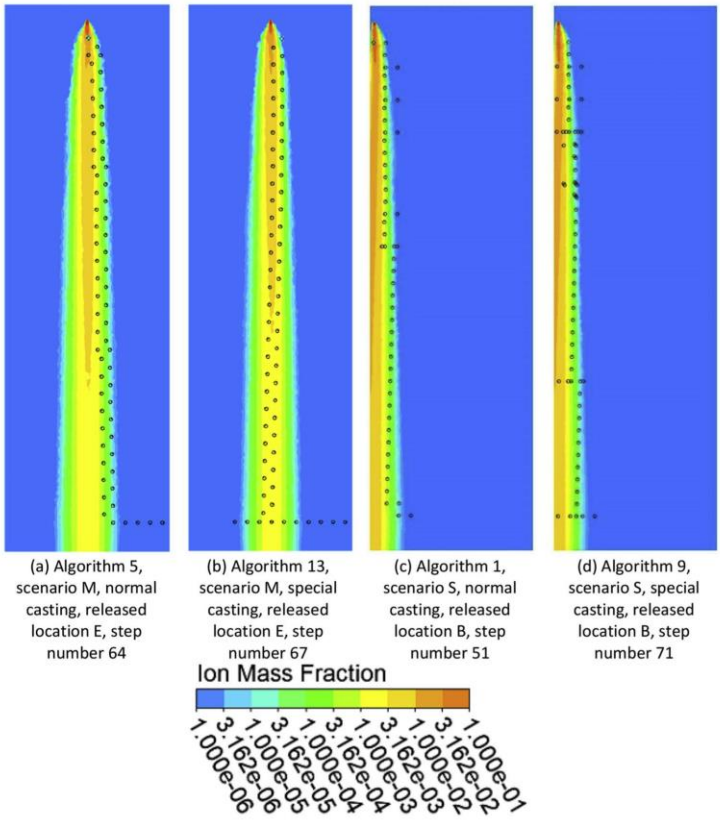


Fig. 10. Selected cases of different PS.

plumes are detected and this was realised by the proposed 'along-wall' obstacle avoidance method, which was developed with the help of multiple ultrasonic sensors. The distances measured by ultrasonic sensors were used to ensure the robot moving parallel to the wall, which is the fundamental principle of this 'along-wall' obstacle avoidance method. To the best of authors' knowledge, the proposed 'along-wall' obstacle avoidance method for plume-tracing robots has not been reported in literature yet. An overall logic flow of the 'along-wall' obstacle method is presented in Fig. 13 (a). Fig. 13 (b) represents how the robot turns and surges along a wall when the wall is detected. The turning angle α is governed by the equation:

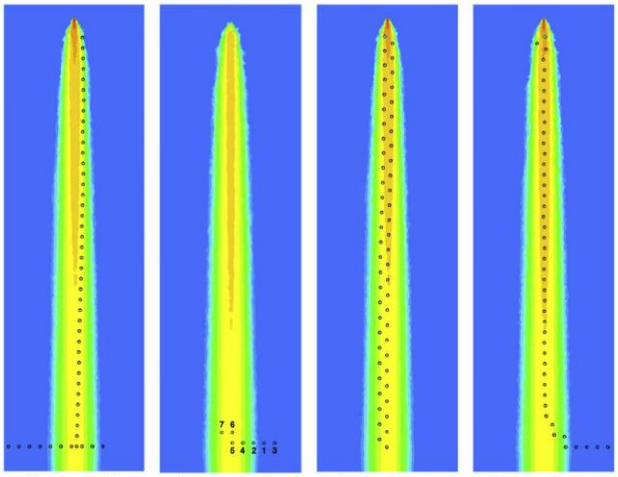
$$\alpha = \frac{\pi}{4} - \arccos\left(\frac{OA}{OB}\right) \tag{4}$$

where *OA* is the distance between the wall and the centre of the robot measured by the ultrasonic sensor that covers L section and *OB* is the distance between Point B and the centre of the robot. Here, *OB* is the sum of the diameter of the robot and the distance (mentioned in Section 3.2) measured by the ultrasonic sensor that covers ML section (see Fig. 13). For every step surging along the wall, the robot can revise its heading to make it parallel to the wall. Either leaving the plume or the end of a wall will stop this 'along-wall' process then the robot will return to the normal plume-tracing status.

In terms of the simulation results presented in Figs. 14 and 15, the robot with *vallumtaxis* is capable of detecting and localising the chemical source successfully. In scenario M, the total number of steps as well as the trajectories of the robot coded with *vallumtaxis* are the same as that of the robot with Algorithm 1. In scenario S, the trajectories of the robot with *vallumtaxis* show that the robot went along the wall without moving back and with *vallumtaxis* the total number of steps significantly reduced (See Figs. 14 and 15). It can be concluded that compared to the algorithms with the concentration-based obstacle avoidance method (explained in Section 3.2), *vallumtaxis* successfully avoids repeating movements near a wall (where the plume is slim) and helps to achieve higher efficiencies. Therefore, *vallumtaxis* can be adopted for plume-tracing robots for locating a gas leakage source in an indoor environment like scenario M or scenario S in this study, particularly when the source is near a wall such as scenario S.

6. Conclusions

The research in this study tested and compared 16 different plume-tracing algorithms in two different scenarios in a 3D simulation framework. It is concluded that *normal casting*, *surge anemotaxis* and *constant stepsize* performed the best compared with other algorithms for PS, PT and SL stages respectively. When a plume is away from walls, releasing the plume-tracing robot from a position that is inside the plume resulted in higher efficiencies in terms of total step numbers. For a wall plume, no

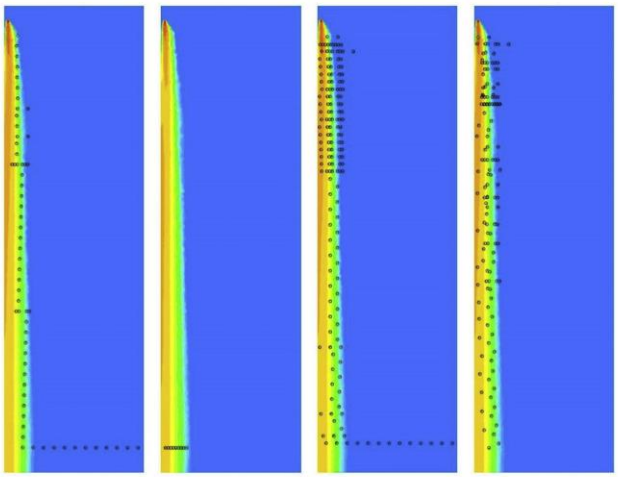


(a) Algorithm 9, scenario M, surge anemotaxis, released location A, step number 51

(b) Algorithm 3, scenario M, chemotaxis, released location D, failed

(c) Algorithm 5, scenario M, zigzags, released location C, step number 56

(d) Algorithm 7, scenario M, pseudo-gradient-based algorithm, released location E, step number 46



(e) Algorithm 2, scenario S, surge anemotaxis, released location E, step number 62

(f) Algorithm 4, scenario S, chemotaxis, released location A, failed

(g) Algorithm 13, scenario S, zigzags, released location D, step number 180

(h) Algorithm 7, scenario S, pseudo-gradient-based algorithm, released location A, step number 135

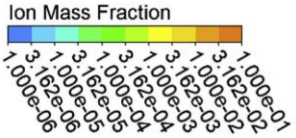
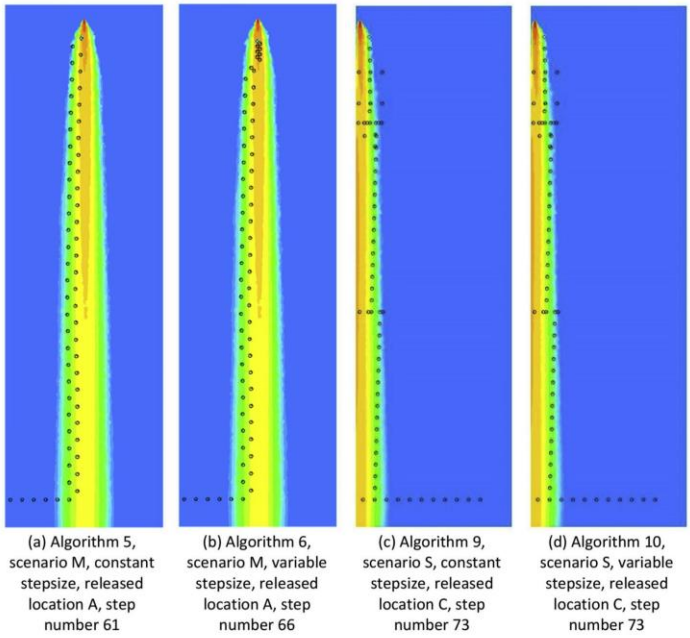


Fig. 11. Selected cases of different PT.



(a) Algorithm 5, scenario M, constant stepsize, released location A, step number 61

(b) Algorithm 6, scenario M, variable stepsize, released location A, step number 66

(c) Algorithm 9, scenario S, constant stepsize, released location C, step number 73

(d) Algorithm 10, scenario S, variable stepsize, released location C, step number 73

Fig. 12. Selected cases of different SL.

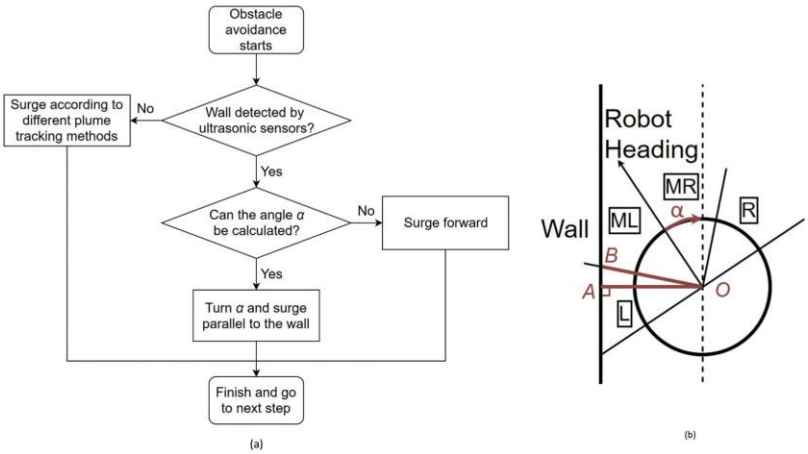
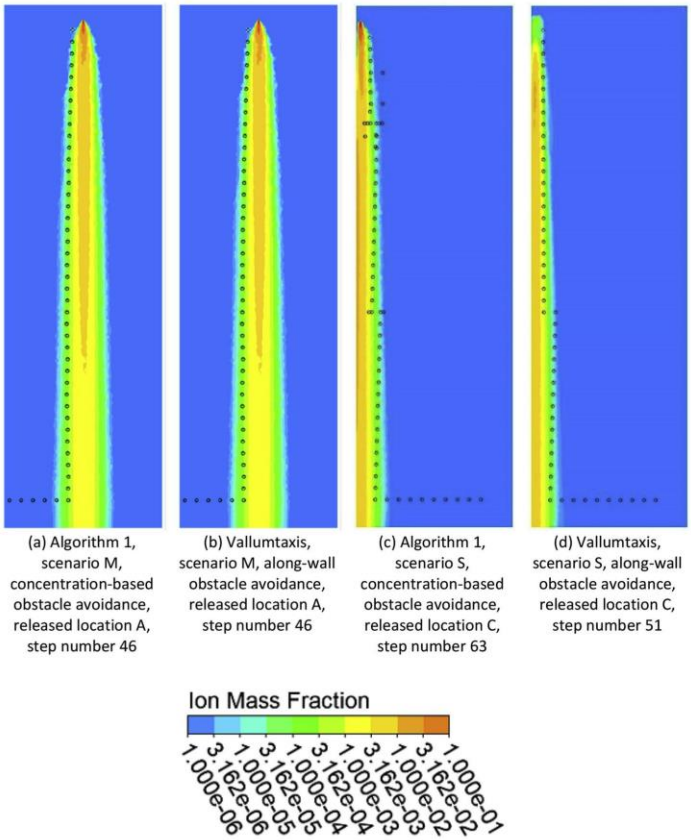


Fig. 13. (a) The overall logic flow of the 'along-wall' obstacle avoidance method, (b) Calculation of the turning angle.



(a) Algorithm 1, scenario M, concentration-based obstacle avoidance, released location A, step number 46

(b) Vallumtaxis, scenario M, along-wall obstacle avoidance, released location A, step number 46

(c) Algorithm 1, scenario S, concentration-based obstacle avoidance, released location C, step number 63

(d) Vallumtaxis, scenario S, along-wall obstacle avoidance, released location C, step number 51

Fig. 14. Comparison of trajectories between *vallumtaxis* and Algorithm 1.

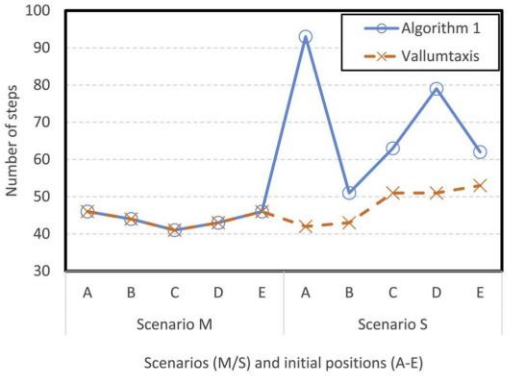


Fig. 15. Comparison of number of steps between Algorithm 1 and vallumtaxis.

advantageous initial locations were found. The concentration-based obstacle avoidance method was proved to be able to protect the robot from hitting obstacles however its performance was unsatisfactory when the plume source is located near the wall. Taking advantage of all the comparison results, the novel *vallumtaxis* method, with a specially designed 'along-wall' obstacle avoidance method was proposed and confirmed to contribute to higher efficiencies especially when searching in a wall plume. The confined flow simulated in the current study is a fundamental and essential flow that is often encountered in indoor environments. The conclusion and findings of the paper can therefore be applied to many other indoor environments. For future work, *vallumtaxis* will be tested in other virtual and realistic environments to test its feasibility and for further improvements.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The financial support from Australia Research Council (ARC)

Industrial Transformation Training Centres (IC170100032) is acknowledged.

References

[1] Y. Chen, H. Cai, Z. Chen, Q. Feng, Using multi-robot active olfaction method to locate time-varying contaminant source in indoor environment, Build. Environ. 118 (2017) 101–112.

[2] J.-H. Kim, B.Y. Lattimer, Real-time probabilistic classification of fire and smoke using thermal imagery for intelligent firefighting robot, Fire Saf. J. 72 (2015) 40–49.

[3] J. Penders, L. Alboul, U. Witkowski, A. Naghsh, J. Saez-Pons, S. Herbrechtsmeier, M. El-Habbal, A robot swarm assisting a human fire-fighter, Adv. Robot. 25 (1–2) (2012) 93–117.

[4] T.-F. Lu, Indoor odour source localisation using robot: are there advantageous initial locations?, in: IEEE International Conference on Robotics and Biomimetics, IEEE, 2011, 2011, pp. 384–389.

[5] Z. Liu, Odour Source Localization Using Multiple Plume-Tracing Mobile Robots, School of Mechanical Engineering, The University of Adelaide, 2010.

[6] Z. Liu, T.-F. Lu, Multiple robots plume-tracing in open space obstructed environments, in: IEEE International Conference on Robotics and Biomimetics (ROBIO), IEEE, 2009, 2009, pp. 2433–2439.

[7] Y.-D. Kim, Y.-G. Kim, S.-H. Lee, J.-H. Kang, J. An, Portable fire evacuation guide robot system, in: The 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems St. Louis, USA, 2009.

[8] D. Zarzhitsky, D.F. Spears, W.M. Spears, Swarms for chemical plume tracing, in: Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005., IEEE, 2005, pp. 249–256.

[9] Q. Feng, C. Zhang, J. Lu, H. Cai, Z. Chen, Y. Yang, F. Li, X. Li, Source localization in dynamic indoor environments with natural ventilation: an experimental study of a particle swarm optimization-based multi-robot olfaction method, Build. Environ. (2019) 106228.

[10] Y. Yang, Q. Feng, H. Cai, J. Xu, F. Li, Z. Deng, C. Yan, X. Li, Experimental study on three single-robot active olfaction algorithms for locating contaminant sources in indoor environments with no strong airflow, Build. Environ. 155 (2019) 320–333.

[11] W. Naeem, R. Sutton, J. Chudley, Chemical plume tracing and odour source localisation by autonomous vehicles, J. Navig. 60 (2) (2007) 173–190.

[12] M.A. Willis, R.T. Cardé, Pheromone-modulated optomotor response in male gypsy moths, Lymantria dispar L. :Upwind flight in a pheromone plume in different wind velocities, J. Comparatove Physiol. A. 167 (1990) 699–706.

[13] C.T. David, J.S. Kennedy, A.R. Ludlow, Finding of a sex pheromone source by gypsy moths released in the field, Nature 303 (1983) 804–806.

[14] T.C. Baker, K.F. Haynes, Pheromone-mediated optomotor anemotaxis and altitude control exhibited by male oriental fruit moths in the field, Physiol. Entomol. 21 (1996) 20–32.

[15] X.-x. Chen, J. Huang, Odor source localization algorithms on mobile robots: a review and future outlook, Robot. Autonom. Syst. 112 (2019) 123–136.

[16] R.A. Russell, A. Bab-Hadiashar, R.L. Shepherd, G.G. Wallace, A comparison of reactive robot chemotaxis algorithms, Robot. Autonom. Syst. 45 (2) (2003) 83–97.

[17] R.A. Russell, D. Thiel, R. Deveza, A. Mackay-Sim, A robotic system to locate hazardous chemical leaks, in: Proceedings of 1995 IEEE International Conference on Robotics and Automation, IEEE, 1995, pp. 556–561.

[18] M. Awadalla, T.-F. Lu, Z.F. Tian, B. Dally, Z. Liu, 3D framework combining CFD and MATLAB techniques for plume source localization research, Build. Environ. 70 (2013) 10–19.

[19] T. Lochmatter, A. Martinoli, Tracking Odor Plumes in a Laminar Wind Field with Bio-Inspired Algorithms, Experimental Robotics, Springer, 2009, pp. 473–482.

[20] D.J. Harvey, T.-F. Lu, M.A. Keller, Comparing insect-inspired chemical plume tracking algorithms using a mobile robot, IEEE Trans. Robot. 24 (2) (2008) 307–317.

[21] D.J. Harvey, T.-F. Lu, M.A. Keller, Effectiveness of insect-inspired chemical plume-tracking algorithms in a shifting wind field, IEEE Trans. Robot. 24 (1) (2008) 196–201.

[22] P.P. Neumann, V. Hernandez Bennetts, A.J. Lilienthal, M. Bartholmai, J.H. Schiller, Gas source localization with a micro-drone using bio-inspired and particle filter-based algorithms, Adv. Robot. 27 (9) (2013) 725–738.

[23] T.-F. Lu, Indoor odour source localisation using robot: initial location and surge distance matter? Robot. Autonom. Syst. 61 (6) (2013) 637–647.

[24] H. Ishida, H. Tanaka, H. Taniguchi, T. Moriizumi, Mobile robot navigation using vision and olfaction to search for a gas/odor source, Aut. Robots 20 (3) (2006) 231–238.

[25] G. Rein, Smouldering combustion phenomena in science and technology, Int. Rev. Chem. Eng. 1 (2009) 3–18.

[26] Z. Liu, T.-F. Lu, A simulation framework for plume-tracing research, in: Australasian Conference on Robotics and Automation, Canberra, Australia, 2008, pp. 1–7.

[27] T.-F. Lu, H.N. Le, Human-in-the-loop chemical plume tracking using robots - mapping while localizing source, in: Tokyo International Conference on Engineering and Applied Sciences, Japan, 2014.

[28] M. Awadalla, T.-F. Lu, Z.F. Tian, B. Dally, CFD modeling of 3D indoor gas contaminant plumes for testing search algorithm of mobile robots, in: Ninth International Conference on Computational Fluid Dynamics in the Minerals and Process Industries, Melbourne, Australia, 2012.

[29] J.M. Soares, A. Marjovi, J. Giezendanner, A. Kodiyan, A.P. Aguiar, A.M. Pascoal, A. Martinoli, Towards 3-D distributed odor source localization: an extended graph-based formation control algorithm for plume tracking, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2016, 2016, pp. 1729–1736.

[30] P.P. Neumann, H. Kohlhoff, D. Hüllmann, A.J. Lilienthal, M. Kluge, Bringing Mobile Robot Olfaction to the next dimension—UAV-based remote sensing of gas clouds and source localization, in: IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2017, 2017, pp. 3910–3916.

[31] K.S. Eu, K.M. Yap, Chemical plume tracing: a three-dimensional technique for quadrotors by considering the altitude control of the robot in the casting stage, Int. J. Adv. Rob. Syst. 15 (1) (2018), 1729881418755877.

[32] T. Lochmatter, X. Raemy, L. Matthey, S. Indra, A. Martinoli, A comparison of casting and spiraling algorithms for odor source localization in laminar flow, in: IEEE International Conference on Robotics and Automation, IEEE, 2008, 2008, pp. 1138–1143.

[33] D.J. Harvey, An Investigation into Insect Chemical Plume Tracking Using a Mobile Robot, 2007.

[34] D. Zarzhitsky, D.F. Spears, W.M. Spears, Distributed robotics approach to chemical plume tracing, in: 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2005, pp. 4034–4039.

[35] D. Zarzhitsky, D.F. Spears, W.M. Spears, D.R. Thayer, A fluid dynamics approach to multi-robot chemical plume tracing, International Conference on Autonomous Agents, in: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-, 2004, pp. 1476–1477.

[36] D.V. Zarzhitsky, D.F. Spears, D.R. Thayer, Experimental studies of swarm robotic chemical plume tracing using computational fluid dynamics simulations, Int. J. Intell. Comput. Cybern. 3 (4) (2010) 631–671.

[37] J.M. Soares, A.P. Aguiar, A.M. Pascoal, A. Martinoli, An Algorithm for Formation-Based Chemical Plume Tracing Using Robotic Marine Vehicles, OCEANS 2016 MTS/IEEE Monterey, IEEE, 2016, pp. 1–8.

[38] Q. Liao, E.A. Cowen, The information content of a scalar plume–A plume tracing perspective, Environ. Fluid Mech. 2 (1–2) (2002) 9–34.

[39] W. Li, Identifying an odour source in fluid-advected environments, algorithms abstracted from moth-inspired plume tracing strategies, Appl. Bionics Biomechanics 7 (1) (2010) 3–17.

[40] A. Lilienthal, T. Duckett, Experimental analysis of gas-sensitive Braitenberg vehicles, Adv. Robot. 18 (8) (2012) 817–834.

Appendix D Conference paper published

In Appendix D, the conference paper that was presented in *ICCM 2020* is given [43].

Localisation of fire source in a warehouse using plume-tracing method

***Zeqi Li[1], †Zhao Tian[1], Tien-fu Lu[1]and Houzhi Wang[2]**

[1]School of Mechanical Engineering, The University of Adelaide, Australia.
[2]School of Chemical Engineering and Advanced Materials, The University of Adelaide, Australia

*Presenting author: zeqi.li@adelaide.edu.au
†Corresponding author: zhao.tian@adelaide.edu.au

Abstract

There has been a high demand for fire prevention in large warehouses and workshops for many years. It is important to localise inconspicuous potential fire, which is called smouldering, timely and precisely in order to prevent significant damage. Different from a flaming fire source, smouldering source emits less smoke and sustains little heat so it can be challenging for smoke detectors and heat detectors to detect. Recent research shows that unmanned robots that are equipped with plume-tracing algorithms are capable of sensing chemical emission and localise its source. Under the condition that gas and smoke may also form plume, it is promising to apply plume-tracing robots in the localisation of smouldering source. In this paper, a novel approach to localise early stage fire using plume-tracing robot is presented. The robot is tested in a virtual warehouse environment created by computational fluid dynamics (CFD) and proved to be capable of detecting and localising the smouldering source in a warehouse environment successfully.

Keywords: Computational fluid dynamics; Robotics; Fire prevention; Plume-tracing algorithms

Introduction

To minimise the damage caused by industrial fires, it is best to detect and localise the fires at an early stage. In this circumstance, this paper proposes a novel approach to detecting and localising fires at an early stage, particularly the slow smouldering fires, by using plume-tracing methods. The slow smouldering fires are dangerous and can cause significant damage if growing into flaming fires. Smouldering is often characterised by low-temperature and flameless, which make it difficult to detect by using convention fire prevention equipment such as heat detectors and smoke detectors [1]-[3]. Often locating inside the cargo, smouldering sources can also be difficult to capture by using cameras. Moreover, smouldering sources usually emit a variety of gases and this provides the potential for detection and localisation of smouldering source by using mobile robots that are equipped with gas sensors. In this case, the expectation is that the plume-tracing robots will detect and localise smouldering sources by tracking in the plume.

Analysis of gases from smouldering sources shows that smouldering sources emit a higher level of carbon monoxide concentration than that of a flaming combustion [2][3]. Unlike carbon dioxide, the concentration of carbon monoxide in the air is very low, therefore being an ideal target gas species for plume-tracing proposes. In this paper, carbon monoxide is selected as the target gas species, and a virtual robot equipped with carbon monoxide sensors and wind sensors is adopted. A bio-inspired plume-tracing search algorithm [4] is introduced to navigate the robot to approach the smouldering source by continuously moving upwind the plume. People found that chemicals like pheromone may influence the moments of insects like moths [5]. A great deal of investigations on plume-tracing behaviours have been carried out and then the plume-tracing algorithms were summarised and proposed [5][6]. Unmanned

Simulation Setup

CFD is used to simulate the airflows in a warehouse with a smouldering source inside. The commercial CFD software, ANSYS Fluent 19.2, is used to simulate the gas concentration distribution as well as the wind fields in the warehouse by using the k-ω SST model. The combination of CFD-generated data and Matlab search algorithms has been validated in [4][7][8], confirming that testing the robot in virtual environment is applicable. The data generated by CFD including gas concentration, wind fields and geometry information are imported to the Matlab for testing and training the robot. A general view of the warehouse geometry is shown in Figure 2, and the CFD-predicted wind field and gas concentration distribution of carbon monoxide at 0.3m high is shown in Figure 3. The chemical concentration legend is logarithmic to present a clearer distribution. The warehouse is 100 metres long in the X direction and 60 metres long of the long margin in the Y direction (see Figure 2). The length of the wall with air inlet windows is 40m and the height is 6m, while the height of the part with reserved load area is 8m. A smouldering source is maintained on a stack of cargo and is modelled as a small surface continuously emitting carbon monoxide and carbon dioxide at the temperature of about 300°C. A steady state model is used with an air inlet speed of 1m/s at four windows and one small door as shown in Figure 2. The large door at the other side of the warehouse is set as a pressure outlet boundary (see Figure 2). The initial position of the robot is near the air outlet door.

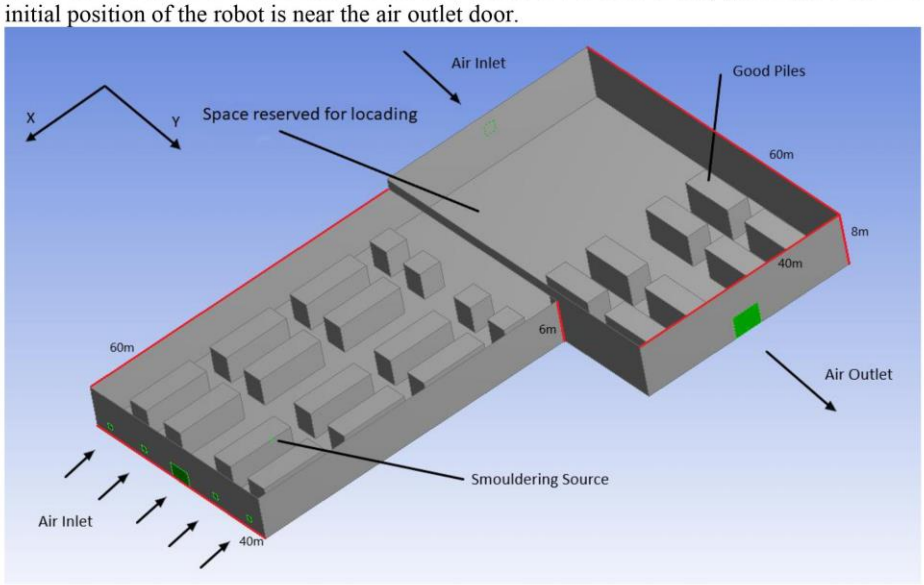


Figure 2: The geometry model of the warehouse

3D Simulation Framework Consisting of Matlab and CFD

In addition to the iSCA-taxis, a 3D simulation framework combining Matlab and CFD together, which was firstly performed by [7], is applied. For the 3D simulation framework, the CFD data at different horizontal levels (for example in this paper, 0.3 m, 0.5 m, 1.0 m and 1.5 m above the ground) are used for the Matlab search algorithms. In general, 3D framework is more realistic than 2D framework in testing the search algorithms as plumes are always three-dimensional. More details of the 3D framework can be found in [7].

robots equipped with plume-tracing algorithms are widely used in the detection and localisation of the sources of gas leakage, explosive and fires. Based on the simplest implementation of plume–tracing algorithms, called 'surge anemotaxis', an algorithm that makes robot surge upwind when maintaining in the plume, novel localisation and framework have been added to improve the searching efficiency. In this case study, the variable step size mechanism [4] and 3D simulation framework [7] are combined to provide a novel approach to detecting and localising a smouldering fire at its early stage in a warehouse environment. The plume-tracing search algorithm is tested in a virtual environment created by CFD, which already proved to be a good simulation tool and used to simulate various indoor and outdoor gas propagation [4][7][8].

Methodology

Plume-tracing Algorithm

The plume-tracing algorithm used in this paper was firstly investigated by [4] and proved to be a good tool for detecting and localising H_2S leakage sources in an office-like indoor environment. Figure 1 shows how this plume-tracing algorithm works. The surge anemotaxis plume-tracing algorithm, which navigates the robot to surge upwind every step, has been optimised in the current study. Unlike the normal surge anemotaxis investigated in [9][10] with the constant step size, for the novel iSCA-taxis [4] with variable step sizes, the robot will increase step sizes when the concentration of the target chemical decreases. Also, the robot will decrease the step size when the local gas concentration increases. This mechanism helps the robot move faster by surging a longer distance when species concentration is low, therefore reducing the searching time and improving the searching efficiency. A simple equation to determine the surge distance is adopted as:

$$XY=C \tag{1}$$

where X is the chemical concentration and Y is the normal surge distance. C is constant and can be adjusted case by case.

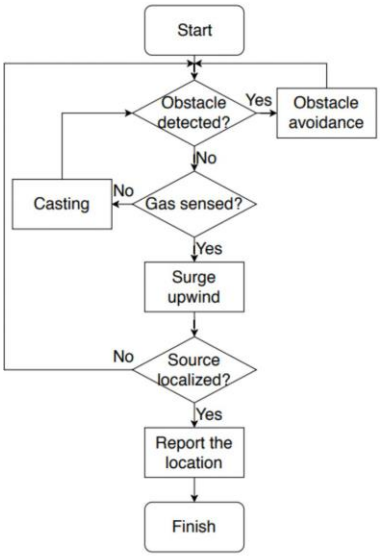


Figure 1: Mechanism of a typical plume-tracing process

Results and Discussion

Figure 3 shows the CFD-predicted wind field and concentration of carbon monoxide along a horizontal plane with a height of 0.3 meters above the ground. The smouldering source is located at a top edge of a goods , as shown in Figure 2. The wind enters the warehouse from the windows and the door on the right wall. A plume of CO emitted from the smouldering source is formed downstream of the source. Figure 4 shows the trajectory of the robot tracking the plume in the warehouse. The robot was released from a location closed to the outlet door (see Figure 4). The trajectory demonstrates that the robot controlled by the

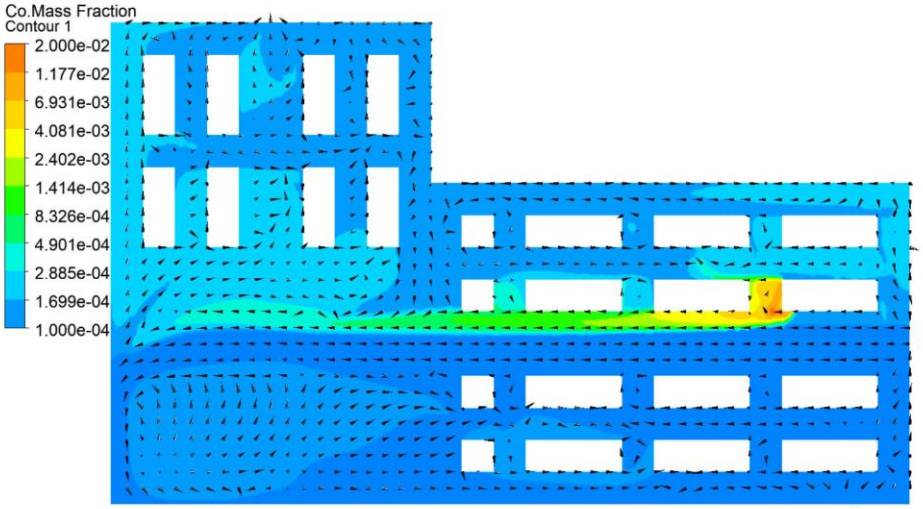


Figure 3: Concentration of carbon monoxide and wind field at height of 0.3m

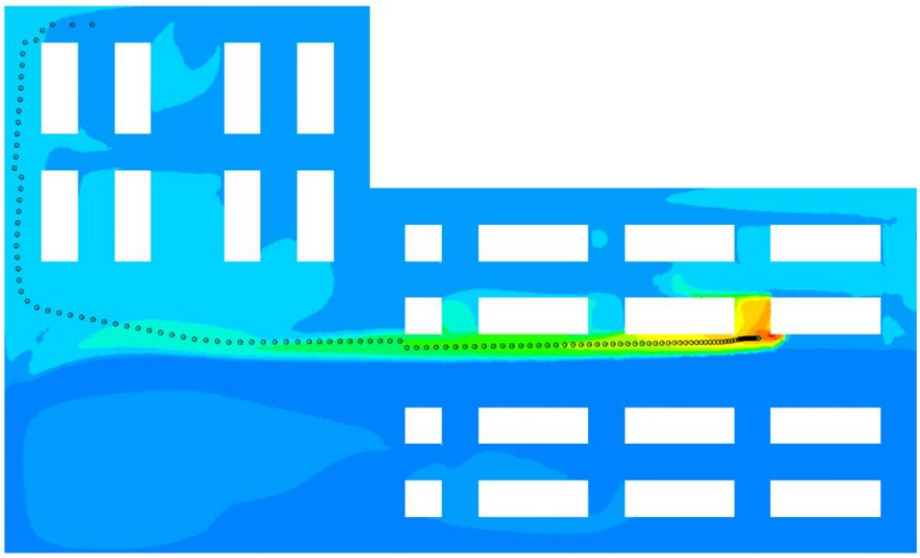


Figure 4: Trajectory of a plume-tracing robot searching for a smouldering source in a warehouse

plume-tracing algorithm successfully localised the smouldering source. The robot initially surged in a large step size (first two steps from the releasing point) in the region close to the outlet door where very low CO concentration is found. After the plume was detected (the third step shown in Figure 4), the robot surged in a smaller but constant step size within the plume. When approaching the source, the local concentration increases and the robot continued moving in a shorter step size to avoid missing the source when surging. Finally the robot found the smouldering source successfully.

Conclusions

To summarise, as afore demonstrated, the plume-tracing robot successfully detected and localised a smouldering source in a large warehouse. The iSCA-taxis [4] with a variable step size mechanism proved to be a suitable solution for enhancing the plume-tracing algorithm. More tests will be carried out to test the effect of releasing locations, source locations, wind fields, etc. on the performance of the robot in detecting and localising the smouldering source. Furthermore, as more literatures on plume-tracing algorithms are carried out, novel search algorithms with higher searching efficiency have the potential to be applied on fire prevention. Hence, future works of this study conclude testing more times with the smouldering source located at different positions within warehouses with different geometry design and optimise the plume-tracing algorithm with recently-proposed improvement methods.

References

[1] G. Rein, "Smouldering Combustion Phenomena in Science and Technology," International Review of Chemical Engineering, vol. 1, pp. 3-18, 2009.

[2] H. Wang, P. J. van Eyk, P. R. Medwell, C. H. Birzer, Z. F. Tian, and M. Possell, "Identification and Quantitative Analysis of Smoldering and Flaming Combustion of Radiata Pine," Energy & Fuels, vol. 30, no. 9, pp. 7666-7677, 2016.

[3] H. Wang, P. J. van Eyk, P. R. Medwell, C. H. Birzer, Z. F. Tian, and M. Possell, "Effects of Oxygen Concentration on Radiation-Aided and Self-sustained Smoldering Combustion of Radiata Pine," Energy & Fuels, vol. 31, no. 8, pp. 8619-8630, 2017.

[4] T.-F. Lu, "Indoor odour source localisation using robot: Initial location and surge distance matter?," Robotics and Autonomous Systems, vol. 61, no. 6, pp. 637-647, 2013.

[5] M. A. Willis and R. T. Cardé, "Pheromone-modulated optomotor response in male gypsy moths, Lymantria dispar L. :Upwind flight in a pheromone plume in different wind velocities," Joumal of Comparatove Physiology A, vol. 167, pp. 699-706, 1990.

[6] T. C. Baker and K. F. Haynes, "Pheromone-mediated optomotor anemotaxis and altitude control exhibited by male oriental fruit moths in the field," Physiol. Entomol, vol. 21, pp. 20-32, 1996.

[7] M. Awadalla, T.-F. Lu, Z. F. Tian, B. Dally, and Z. Liu, "3D framework combining CFD and MATLAB techniques for plume source localization research," Building and Environment, vol. 70, pp. 10-19, 2013.

[8] T.-F. Lu, "Indoor odour source localisation using robot: Are there advantageous initial locations?," in 2011 IEEE International Conference on Robotics and Biomimetics, 2011, pp. 384-389: IEEE.

[9] D. J. Harvey, T.-F. Lu , and M. A. Keller, "Comparing Insect-Inspired Chemical Plume Tracking Algorithms Using a Mobile Robot," IEEE Transactions on Robotics, vol. 24, no. 2, pp. 307-317, 2008.

[10] Z. Li, Z. F. Tian, T.-f. Lu, and H. Wang, "Assessment of different plume-tracing algorithms for indoor plumes," Building and Environment, p. 106746, 2020.

References

[1] D. A. Stewart and M.-K. Liu, "Development and application of a reactive plume model," *Atmospheric Environment (1967),* vol. 15, no. 10-11, pp. 2377-2393, 1981.

[2] T.-F. Lu, "Indoor odour source localisation using robot: Initial location and surge distance matter?," *Robotics and Autonomous Systems,* vol. 61, no. 6, pp. 637-647, 2013.

[3] J.-G. Li, J. Yang, S.-G. Cui, and L.-H. Geng, "Speed limitation of a mobile robot and methodology of tracing odor plume in airflow environments," *Procedia Engineering,* vol. 15, pp. 1041-1045, 2011.

[4] D. J. Harvey, T.-F. Lu , and M. A. Keller, "Effectiveness of Insect-Inspired Chemical Plume-Tracking Algorithms in a Shifting Wind Field," *IEEE Transactions on Robotics,* vol. 24, no. 1, pp. 196-201, 2008.

[5] Y. Yang *et al.*, "Experimental study on three single-robot active olfaction algorithms for locating contaminant sources in indoor environments with no strong airflow," *Building and Environment,* vol. 155, pp. 320-333, 2019.

[6] Y. C. Chen, H. Cai, Z. L. Chen, and Q. L. Feng, "Using multi-robot active olfaction method to locate time-varying contaminant source in indoor environment," *Building and Environment,* vol. 118, pp. 101-112, Jun 2017.

[7] J.-H. Kim and B. Y. Lattimer, "Real-time probabilistic classification of fire and smoke using thermal imagery for intelligent firefighting robot," *Fire Safety Journal,* vol. 72, pp. 40-49, 2015.

[8] J. Penders *et al.*, "A Robot Swarm Assisting a Human Fire-Fighter," *Advanced Robotics,* vol. 25, no. 1-2, pp. 93-117, 2012.

[9] T.-F. Lu, "Indoor odour source localisation using robot: Are there advantageous initial locations?," in *2011 IEEE International Conference on Robotics and Biomimetics*, 2011, pp. 384-389: IEEE.

[10] Z. Liu, "Odour Source Localization Using Multiple Plume-Tracing Mobile Robots," Doctor of Philosophy Thesis, School of Mechanical Engineering, The University of Adelaide, 2010.

[11] Z. Liu and T.-F. Lu, "Multiple robots plume-tracing in open space obstructed environments," in *2009 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2009, pp. 2433-2439: IEEE.

[12] Y.-D. Kim, Y.-G. Kim, S.-H. Lee, J.-H. Kang, and J. An, "Portable Fire Evacuation Guide Robot System," presented at The 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, USA, October 11-15, 2009.

[13] D. Zarzhitsky, D. F. Spears, and W. M. Spears, "Swarms for chemical plume tracing," in *Proceedings 2005 IEEE Swarm Intelligence Symposium,* 2005, pp. 249-256: IEEE.

[14] Q. Feng *et al.*, "Source localization in dynamic indoor environments with natural ventilation: An experimental study of a particle swarm optimization-based multi-robot olfaction method," *Building and Environment,* p. 106228, 2019.

[15] Y. Chen, H. Cai, Z. Chen, and Q. Feng, "Using multi-robot active olfaction method to locate time-varying contaminant source in indoor environment," *Building and Environment,* vol. 118, pp. 101-112, 2017.

[16] Z. Liu and T.-F. Lu, "A Simulation Framework for Plume-Tracing Research," presented at the Australasian Conference on Robotics and Automation,

Canberra, Australia, 2008.

[17] Q. Liao and E. A. Cowen, "The information content of a scalar plume–A plume tracing perspective," *Environmental Fluid Mechanics,* vol. 2, no. 1-2, pp. 9-34, 2002.

[18] W. Li, "Identifying an odour source in fluid-advected environments, algorithms abstracted from moth-inspired plume tracing strategies," *Applied Bionics and Biomechanics,* vol. 7, no. 1, pp. 3-17, 2010.

[19] D. Zarzhitsky, D. F. Spears, W. M. Spears, and D. R. Thayer, "A fluid dynamics approach to multi-robot chemical plume tracing," in *International Conference on Autonomous Agents: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-*, 2004, vol. 3, pp. 1476-1477.

[20] D. V. Zarzhitsky, D. F. Spears, and D. R. Thayer, "Experimental studies of swarm robotic chemical plume tracing using computational fluid dynamics simulations," *International Journal of Intelligent Computing and Cybernetics,* vol. 3, no. 4, pp. 631-671, 2010.

[21] D. Zarzhitsky, D. F. Spears, and W. M. Spears, "Distributed robotics approach to chemical plume tracing," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp. 4034-4039: IEEE.

[22] K. S. Eu and K. M. Yap, "Chemical plume tracing: A three-dimensional technique for quadrotors by considering the altitude control of the robot in the casting stage," *International Journal of Advanced Robotic Systems,* vol. 15, no. 1, p. 1729881418755877, 2018.

[23] J. A. Farrell, S. Pang, W. Li, and R. Arrieta, "Chemical plume tracing experimental results with a REMUS AUV," in *Oceans 2003. Celebrating the Past... Teaming Toward the Future (IEEE Cat. No. 03CH37492)*, 2003, vol. 2, pp. 962-968: IEEE.

[24] W. Naeem, R. Sutton, and J. Chudley, "Chemical Plume Tracing and Odour Source Localisation by Autonomous Vehicles," *Journal of Navigation,* vol. 60, no. 2, pp. 173-190, 2007.

[25] Z. Li, Z. F. Tian, T.-f. Lu, and H. Wang, "Assessment of different plume-tracing algorithms for indoor plumes," *Building and Environment,* p. 106746, 2020.

[26] J. M. Soares, A. P. Aguiar, A. M. Pascoal, and A. Martinoli, "An algorithm for formation-based chemical plume tracing using robotic marine vehicles," in *OCEANS 2016 MTS/IEEE Monterey*, 2016, pp. 1-8:.

[27] G. Kowadlo and R. A. Russell, "Using naïve physics for odor localization in a cluttered indoor environment," *Autonomous Robots,* vol. 20, no. 3, pp. 215-230, 2006.

[28] T. Lochmatter and A. Martinoli, "Tracking odor plumes in a laminar wind field with bio-inspired algorithms," in *Experimental Robotics*, 2009, pp. 473-482:.

[29] J. M. Soares *et al.*, "Towards 3-D distributed odor source localization: an extended graph-based formation control algorithm for plume tracking," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 1729-1736: IEEE.

[30] R. A. Russell, D. Thiel, R. Deveza, and A. Mackay-Sim, "A robotic system to locate hazardous chemical leaks," in *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, 1995, vol. 1, pp. 556-561: IEEE.

[31] J. Wisnu, S. Kosuke, and F. Toshio, "A PSO-Based Mobile Robot for Oder Source Localization in Dynamic Advection-Diffusion with Obstacles Environment: Theory Simulation and Measurement," *IEEE Computational Intelligence Magazine,* vol. 2, no. 2, pp. 37-51, 2007.

[32] X.-x. Chen and J. Huang, "Odor source localization algorithms on mobile robots: A review and future outlook," *Robotics and Autonomous Systems,* vol.

112, pp. 123-136, 2019.

[33] H. Ishida, H. Tanaka, H. Taniguchi, and T. Moriizumi, "Mobile robot navigation using vision and olfaction to search for a gas/odor source," *Autonomous Robots,* vol. 20, no. 3, pp. 231-238, 2006.

[34] D. J. Harvey, "An investigation into insect chemical plume tracking using a mobile robot," Doctor of Philosophy Thesis, School of Mechanical Engineering, The University of Adelaide, 2007.

[35] P. P. Neumann, V. Hernandez Bennetts, A. J. Lilienthal, M. Bartholmai, and J. H. Schiller, "Gas source localization with a micro-drone using bio-inspired and particle filter-based algorithms," *Advanced Robotics,* vol. 27, no. 9, pp. 725-738, 2013.

[36] D. J. Pack, R. Avanzato, D. J. Ahlgren, and I. M. Verner, "Fire-Fighting Mobile Robotics and Interdisciplinary Design-Comparative Perspectives," *IEEE Transactions on Education,* vol. 47, no. 3, pp. 369-376, 2004.

[37] K. Miyazawa, "Fire robots developed by the Tokyo Fire Department," *Advanced Robotics,* vol. 16, no. 6, pp. 553-556, 2002.

[38] R. A. Russell, A. Bab-Hadiashar, R. L. Shepherd, and G. G. Wallace, "A comparison of reactive robot chemotaxis algorithms," *Robotics and Autonomous Systems,* vol. 45, no. 2, pp. 83-97, 2003.

[39] T. Lochmatter, X. Raemy, L. Matthey, S. Indra, and A. Martinoli, "A comparison of casting and spiraling algorithms for odor source localization in laminar flow," in *2008 IEEE International Conference on Robotics and Automation*, 2008, pp. 1138-1143: IEEE.

[40] D. J. Harvey, T.-F. Lu , and M. A. Keller, "Comparing Insect-Inspired Chemical Plume Tracking Algorithms Using a Mobile Robot," *IEEE Transactions on Robotics,* vol. 24, no. 2, pp. 307-317, 2008.

[41] Y.-L. Chen, J. Cheng, C. Lin, X. Wu, Y. Ou, and Y. Xu, "Classification-based

learning by particle swarm optimization for wall-following robot navigation," *Neurocomputing,* vol. 113, pp. 27-35, 2013.

[42] P. P. Neumann, H. Kohlhoff, D. Hüllmann, A. J. Lilienthal, and M. Kluge, "Bringing Mobile Robot Olfaction to the next dimension—UAV-based remote sensing of gas clouds and source localization," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 3910-3916: IEEE.

[43] Z. Li, Z. Tian, T.-f. Lu, and H. Wang, "Localisation of fire source in a warehouse using plume-tracing method," presented at the The 11th International Conference on Computational Methods (ICCM20), online,

[44] M. A. Willis and R. T. Cardé, "Pheromone-modulated optomotor response in male gypsy moths, Lymantria dispar L. :Upwind flight in a pheromone plume in different wind velocities," *Joumal of Comparatove Physiology A,* vol. 167, pp. 699-706, 1990.

[45] C. T. David, J. S. Kennedy, and A. R. Ludlow, "Finding of a sex pheromone source by gypsy moths released in the field," *Nature,* vol. 303, pp. 804-806, 1983.

[46] T. C. Baker and K. F. Haynes, "Pheromone-mediated optomotor anemotaxis and altitude control exhibited by male oriental fruit moths in the field," *Physiol. Entomol,* vol. 21, pp. 20-32, 1996.

[47] M. Awadalla, T.-F. Lu, Z. F. Tian, B. Dally, and Z. Liu, "3D framework combining CFD and MATLAB techniques for plume source localization research," *Building and Environment,* vol. 70, pp. 10-19, 2013.

[48] T.-F. Lu and H. N. Le, "Human-in-the-loop chemical plume tracking using robots - mapping while localizing source," the Tokyo International Conference on Engineering and Applied Sciences, Japan, December 2014.

[49] M. Awadalla, T.-F. Lu, Z. F. Tian, and B. Dally, "CFD modeling of 3D indoor gas contaminant plumes for testing search algorithm of mobile robots," the Ninth International Conference on Computational Fluid Dynamics in the Minerals and

Process industries, Melbourne, Australia, 10- 12 December 2012.

[50] T. Lochmatter, N. Heiniger, and A. Martinoli, "Localizing an odor source and avoiding obstacles: Experiments in a wind tunnel using real robots," in *AIP Conference Proceedings*, 2009, vol. 1137, no. 1, pp. 69-72: American Institute of Physics.

[51] J. Hunt, "Diffusion in the stable boundary layer," in *Atmospheric Turbulence and Air Pollution Modelling*: Springer, 1984, pp. 231-274.

[52] J. S. Elkinton and R. T. Cardé, "Odor dispersion," in *Chemical ecology of insects*: Springer, 1984, pp. 73-91.

[53] S. R. Farkas and H. H. Shorey, "Chemical trail-following by flying insects: A mechanism for orientation to a distant odor source," *Science,* vol. 178, pp. 67-68, 1972.

[54] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots," *IEEE Transactions on Systems, Man, and Cybernetics,* vol. 19, no. 5, pp. 1179-1187, 1989.

[55] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," in *ICRA*, 1991, vol. 2, pp. 1398-1404.

[56] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine,* vol. 4, no. 1, pp. 23-33, 1997.

[57] K.-H. Kim and H. S. Cho, "An obstacle avoidance method for mobile robots based on fuzzy decision-making," *Robotica,* vol. 24, no. 5, pp. 567-578, 2006.

[58] M. Katsev, A. Yershova, B. Tovar, R. Ghrist, and S. M. LaValle, "Mapping and pursuit-evasion strategies for a simple wall-following robot," *IEEE Transactions on Robotics,* vol. 27, no. 1, pp. 113-128, 2011.

[59] E. Stefopoulos and D. Damigos, "Design of emergency ventilation system for an underground storage facility," *Tunnelling and underground space*

technology, vol. 22, no. 3, pp. 293-302, 2007.

[60] A. Lilienthal and T. Duckett, "Experimental analysis of gas-sensitive Braitenberg vehicles," *Advanced Robotics,* vol. 18, no. 8, pp. 817-834, 2012.