Robotics and Autonomous Systems 147 (2022) 103914

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Robotic odor source localization via adaptive bio-inspired navigation using fuzzy inference methods

Lingxiao Wang, Shuo Pang *

Department of Electrical Engineering and Computer Science, Embry-Riddle Aeronautical University, Daytona Beach, FL, 32124, USA

ARTICLE INFO

Article history:
Received 19 May 2021
Received in revised form 30 September 2021
Accepted 4 October 2021
Available online 8 October 2021

Keywords:
Odor source localization
Behavior-based navigation methods
Fuzzy-inference theories

ABSTRACT

Robotic odor source localization (OSL) has been viewed as a challenging task due to the turbulent nature of airflows and the resulting odor plume characteristics. The key to solving an OSL problem is designing an effective olfactory-based navigation algorithm, which guides a plume-tracing robot to find the odor source via tracing emitted plumes. Inspired by the mate-seeking behaviors of male moths, this article presents a behavior-based navigation algorithm for using on a mobile robot to locate an odor source in an unknown environment. Unlike traditional bio-inspired algorithms, which use fixed parameters to formulate robot search trajectories, we design a fuzzy controller to perceive the environment and adjust trajectory parameters based on the current search situation. Therefore, the robot can automatically adapt the scale of search trajectories to fit environmental changes and balance the exploration and exploitation of the search. Simulation and on-vehicle results show that compared to two classical olfactory-based navigation algorithms, the proposed algorithm is more efficient and outperforms them in terms of the averaged search time and success rate.

1. Introduction

Humans perceive the world with five senses, including sight, sound, smell, taste, and touch. In robotic applications, a machine can achieve similar perception capabilities by integrating various kinds of sensors. Autonomous driving vehicles [1], for instance, can see and hear the surrounding environment via cameras, LIDAR sensors, and sonars. Compared with other sensing abilities, the sense of smell has not been thoroughly studied in the robotic society.

Robotic odor source localization (OSL) is a technology that employs a mobile robot or an autonomous vehicle, equipped with odor detection sensors, i.e., chemical sensors, to find an odor source in an unknown environment [2]. This technology enables robots with a sense of smell, i.e., olfaction, to trace and find odor sources, which has many practical applications. Some of them that are frequently quoted include monitoring air pollution [3], locating chemical gas leaks [4], and marine surveys such as finding underwater hydrothermal vents [5].

To correctly find an odor source, an effective olfactory-based navigation algorithm is critical. Like image-based navigation algorithms, which utilize the information extracted from images as references to navigate a robot, olfactory-based navigation algorithms detect odor plumes as cues to guide a robot in finding the odor source. The challenging part of this navigation problem is to estimate the emitted odor plume locations, which are not only related to the molecular diffusion that takes plumes away from the odor source but also the advection of airflows [6].

In laminar flow environments, plume dispersal is a steady and stable process, which results in a spatially coherent plume trajectory. In this scenario, the intuitive gradient following algorithm, i.e., chemotaxis [7], is applicable for guiding a plume tracing robot to find the odor source. However, in a turbulent flow environment, odor plumes are stretched and twisted to form an intermittent concentration trajectory, which fails the chemotaxis in this environment. Alternatively, two other categories of olfactory-based navigation algorithms, namely bio-inspired and engineering-based (i.e., probabilistic) algorithms, have been researched and proposed [8].

A bio-inspired algorithm directs the robot to mimic animal olfactory behaviors. A typical bio-inspired algorithm is the moth-inspired method (i.e., anemotaxis), which is inspired by the mate-seeking behaviors of male moths [9]: a male moth flies against wind when detecting pheromone plumes emitted from female moths and across wind when plumes are absent. This behavior can be framed as a 'surge/casting' model, where a plume tracing robot moves against the wind direction when detecting plumes (i.e., 'surge') and traverses wind when missing plume contact (i.e., 'casting') [10]. By contrast, engineering-based navigation algorithms utilize mathematical and physics-based approaches to deduce possible odor source locations. Main procedures are

* Corresponding author.
E-mail addresses: lingxiaw@my.erau.edu (L. Wang), shuo.pang@erau.edu (S. Pang).

https://doi.org/10.1016/j.robot.2021.103914
0921-8890/

twofold [11]: first, possible odor source locations are estimated via source mapping algorithms; then, a path planner is employed to direct the robot to the estimated target.

Compared the aforementioned algorithms, chemotaxis is barely researched since it is not applicable in turbulent flow environments. Bio-inspired algorithms are simple and easy to implement but not as effective as an engineering-based algorithm in highly turbulent flow environments. When the airflow field varies significantly, the simple 'surge/casting' model can barely keep the robot continuously detecting plumes [12]. Engineering-based algorithms, on the other hand, can estimate odor source or plume locations to facilitate the plume tracing process and improve the search efficiency [13], but the high computational cost to online estimate source or plume locations impedes its applications on robotic platforms, which have limited computational resources. In summary, a desired olfactory-based navigation algorithm should be light-weighted, i.e., it does not require high computational resources, while efficient and capable enough to find odor sources in different airflow environments.

Motivated by this consideration, the objective of this work is to design an olfactory-based navigation algorithm that compromises benefits from both bio-inspired and engineering-based methods. We want to utilize the simple and concise framework of a bio-inspired method to reduce the algorithm complexity and absorb the intellectual ability (i.e., the ability to mathematically estimate environmental changes) from engineering-based methods to improve the search performance. To achieve this goal, the proposed olfactory-based navigation algorithm is constructed based on the 'surge/casting' behavior framework and is enhanced with a decision-making approach via the fuzzy inference theory to perceive the environment and adjust robot search trajectories accordingly. Specifically, this article discusses the following topics to present the proposed algorithm:

1. Develop an intelligent behavior-based navigation algorithm that improves a traditional bio-inspired plume tracing algorithm based on fuzzy inference theories;
2. Design a fuzzy controller to adjust parameters in search behaviors according to the current search condition, which is estimated by the proposed fuzzy controller based on the onboard sensor measurements;
3. Implement the proposed navigation algorithm in varying airflow environments to evaluate its validity and compare the results with traditional bio-inspired and engineering-based algorithms.

The remainder of this paper is organized as below: Section 2 reviews the recent published olfactory-based navigation algorithms; Section 3 shows the overview of the proposed algorithm; Section 4 reviews robot search behaviors in a behavior-based navigation algorithm; Section 5 presents the design of the fuzzy controller; Section 6 demonstrates experiment designs and results.

2. Related works

Robotic OSL is a heated research topic in recent decades. Thanks to the development of technologies in robotics and autonomous systems, implementing a mobile robot or an autonomous vehicle to trace and locate a chemical (or odor) source in a hash environment becomes feasible. Over the past three decades, a significant amount of research works have proposed various kinds of olfactory-based navigation algorithms, which can be grouped into three categories, namely chemotaxis, bio-inspired, and engineering-based algorithms.

Early works of robotic OSL attempt to complete this task via a simple gradient following algorithm, i.e., chemotaxis. A common implementation of this algorithm is to install a pair of chemical sensors on the left and right sides of a plume-tracing robot, where the robot is commanded to steer toward the side with the higher concentration measurement [14]. Many experiments [15–18] have proved the validity of chemotaxis in laminar flow environments (i.e., low Reynolds numbers). However, this method is not applicable in an environment with turbulent flows (i.e., high Reynolds numbers) since odor plumes are congregated into packets and the gradient of concentration becomes a patchy and intermittent signal [19]. To approach the problem of plume tracing in turbulent flow environments, bio-inspired and engineering-based methods are proposed.

Bio-inspired methods direct the plume-tracing robot to mimic animal odor search behaviors, which can be summarized into two common features: upwind search when plume is detected, and a local search after losing plume. Bio-inspired methods include moth-inspired, lobster-inspired, and beetle-inspired methods [8]. Moth-inspired methods can be summarized as a 'surge/casting' behavior pattern. Lochmatter et al. [20] implemented this behavior pattern on a ground wheeled vehicle to find an odor source in a laminar flow environment. In the underwater environment, Li et al. [21] applied it on an autonomous underwater vehicle (AUV) to find an underwater chemical source over a large search area (about 100×100 m^2). Above the ground, Luo et al. [22] designed a flying odor compass based on an unmanned aerial vehicle (UAV), which can identify the airflow direction based on plume detection events and trace plumes to find the odor source accordingly (i.e., fly upwind).

The conventional 'surge/casting' behavior in the moth-inspired method can be improved to obtain a better search performance on robotic systems. Farrell et al. [23] modified the conventional 'surge' behavior by adding customized 'track-in' and 'track-out' behaviors to increase the plume contact duration in the plume tracing process. Shigaki et al. [24] introduced a time varying moth-inspired method, where the duration of the 'surge' behavior is calculated by an equation obtained from analyzing the moth's muscle activity under plume stimulation. Recently, they also proposed a fuzzy controller [25] to adaptively control the transition among 'surge', 'stop', and 'casting' behaviors according to airflow changes in the search environment. Ferri et al. [26] proposed a 'spiral' search trajectory for directing the robot to re-detect plumes in the 'casting' behavior. Rahbar et al. [27] presented a 3-dimensional (3-D) version 'spiral' trajectory and validated it with experiments in a wind tunnel. Shunsuke et al. [28] found wind directions affect movement speeds of moths when tracing pheromone. They proposed a modified 'surge/casting' model: when the detected wind direction is upwind, the robot will decrease its upwind speed. Liberzon et al. [29] count the time interval (i.e., t_c) between two plume detection events. As long as the plumes are encountered at the rate of t_c, the robot performs the 'surge' behavior; otherwise, the robot performs the 'casting' behavior. Simulated results showed that the modified method has the average success rate around 80%.

Lobster-inspired methods are similar to chemotaxis but take inspiration of lobster's foraging strategies: the lobster turns toward the side of higher odor concentration or go straight forward if two antennules detect nearly the same concentration, and if neither of two antennules detects concentration, the lobster moves backward. Grasso et al. [30] implemented the lobster-inspired method on an AUV in turbulent flow underwater environment. Compared to the pure chemotaxis, the addition of the second strategy increased the success rate of finding the source from 33% to 66%. Recent studies on simulation verify the validity of lobster-inspired methods [31,32]: a lobster-inspired method that uses both temporal and spatial information achieves a higher success rate than the chemotaxis alone.

The dung beetle odor search strategy can be summarized as follow: when the beetle detects odors, it zigzags diagonally across the plume in an upwind direction turning back each time it leaves the edge of the plume. Inspired by dung beetle's odor search behaviors, the beetle-inspired methods can be summarized as following: when the robot detects odors, it zigzags diagonally across the plume in an upwind direction turning back each time it leaves the edge of the plume [16]. Macedo et al. [33] verified the validity of beetle-inspired methods in a simulation program. Hernadez et al. [34] applied this method to find gas sources in landfills. On-vehicle experiments show that the search performance of beetle-inspired in outdoor turbulent environments is not ideal due to the poor chemical sensing ability and the turbulent nature of outdoor environments that do not allow the formation of a steady odor plume.

Engineering-based methods, on the other hand, command the robot in the plume tracing process relying on the source estimations. The basic idea of source mapping is to calculate the probability of every region in the search area containing the odor source (i.e., constructing the source probability map) and loop improving the quality of map along the plume tracing history. Bayesian-inference method [35] is a typical engineering-based OSL algorithm. In this method, the plume movement is formulated by a Gaussian random process. Based on a series of plume detection and non-detection events, the robot can inversely calculate the probability of a region containing the odor source. Other algorithms to estimate possible odor source locations include particle filter [36], hidden Markov model (HMM) [13], occupancy grid mapping [37], source term estimation [38,39], and partially observable Markov decision process (POMDP) [40]. The core idea of these methods is to perceive the environment (especially airflow information) and estimate possible odor source locations accordingly.

After a source probability map is obtained, the robot is commanded to move toward the estimated source location via a path planning algorithm. Possible path planners include the artificial potential field (APF) [11] and A-star [41] algorithms. Besides, Vergassola et al. [42] proposed the 'infotaxis' algorithm, which uses the information entropy to guide the robot searching for the odor source. In this method, the robot selects a future movement that mostly reduces the information uncertainty of the odor source. As for multi-agent systems, a commonly-used planning algorithm is particle swarm optimization (PSO) [43], which optimizes the problem via iteratively improves a candidate solution concerning a given fitness function. When applied in an OSL task, measured odor concentrations (e.g., [44,45]) or the probability of an area containing the odor source, i.e., source probability map [46–48], can be utilized to define the fitness function. The best solution can be considered as the possible odor source location. At each iteration, the best solution is evaluated according to the given fitness function, and then, robots' positions are updated to approach the best solution.

By summarizing these works, it can be discovered that most research works focus on either bio-inspired or engineering-based methods, and few of them have considered the combination of these two methods. The research niche that our method fits is that we utilize the simple search behaviors inspired by bio-inspired methods to reduce the algorithm complexity while adaptively adjust the scale of search trajectories depending on the surrounding environment to improve the search efficiency. To achieve this mechanism, we design a fuzzy inference system to perceive the environment and understand the current search situation. Thus, when the robot encounters different search situations, it can efficiently adjust its search behaviors to trace odor plumes and find the odor source.

Implementing fuzzy inference systems in the OSL problem is prevalent in recent published works. Similar ideas can be found in [25,49], but the difference is that our method not only controls the transition between different search states (like [25,49]) but also dynamically adjusts robot search trajectories in different behaviors. [50] also modified a bio-inspired method (i.e., Levy Taxis) using fuzzy inference systems, but on-vehicle evaluations are needed to prove the validity of [50] in real-world environments. In [51], the adaptive neuro-fuzzy inference system (ANFIS) is employed as the machine learning model that controls the robot searching the odor source. In this method, training data was obtained by implementing traditional odor search algorithms in the simulation. After training, the ML model was verified in the simulation.

In recent years, we have witnessed a surge of implementing reinforcement learning (RL) algorithms in robotic problems (e.g., Atari games [52], robot navigation [53], autonomous driving vehicles [54]). The framework of RL algorithms is very suitable for modeling an odor source localization (OSL) problem, where the agent is the mobile robot with the goal of finding the odor source. By properly assigning reward functions, the agent (i.e., robot) is encouraged to choose actions that benefit in finding the odor source (e.g., detecting odor plumes). Despite the high intelligence of RL algorithms, implementing them in OSL problem is still a challenging topic. The primary hurdle is the training process. Most published RL-based OSL works is trained and evaluated in the simulation program, e.g., [55,56], and [57]. This is because training RL needs the agent to repeatedly perform a task and learn from errors. Conducting real OSL experiments is expensive and time-consuming, making it infeasible to train an RL agent in real-world environments. However, simulation cannot always accurately represent the real-world scenarios, while improvements in the simulation do not guarantee the same improvements in practical applications. Due to this concern, we choose to use a more practical method to solve the OSL problem, i.e., fuzzy inference systems. Compared to RL algorithms, the proposed fuzzy inference system does not include the training process. Besides, it can also intelligently adjust robot search behaviors with the change of search environments. The effectiveness of the proposed method is verified in both simulation and on-vehicle tests, where search results indicate that the proposed method outperforms traditional methods in both search time and success rate.

3. An overview of the proposed olfactory-based navigation algorithm

An OSL task can be separated into three search phases, namely plume finding, plume tracing, and source declaration [58]. The first search phase, i.e., plume finding, aims to verify the existence of plumes in the search area. After the robot detects plumes for the first time, the plume tracing phase is activated, in which the robot detects plumes as cues to approach the odor source. Once the robot gains enough information to confirm the source position, it declares the estimated source location in the source declaration phase, which is also considered as the end of an OSL task.

In this work, the design of search behaviors is based on moth-inspired methods presented in [21,23]. We add an intelligent decision-making mechanism, i.e., a fuzzy-inference system, to adjust parameters in robot search behaviors. Fig. 1 presents the flow diagram of these search behaviors in the proposed algorithm. Initially, the 'zigzag' search behavior is employed in the plume finding phase, which guides the robot to detect plumes for the first time. After the initial plume detection, the robot switches to the plume tracing phase, which consists of three search behaviors, namely 'track-in', 'track-out', and 'reacquire'. The designed fuzzy controller is activated at the beginning of the plume tracing phase.

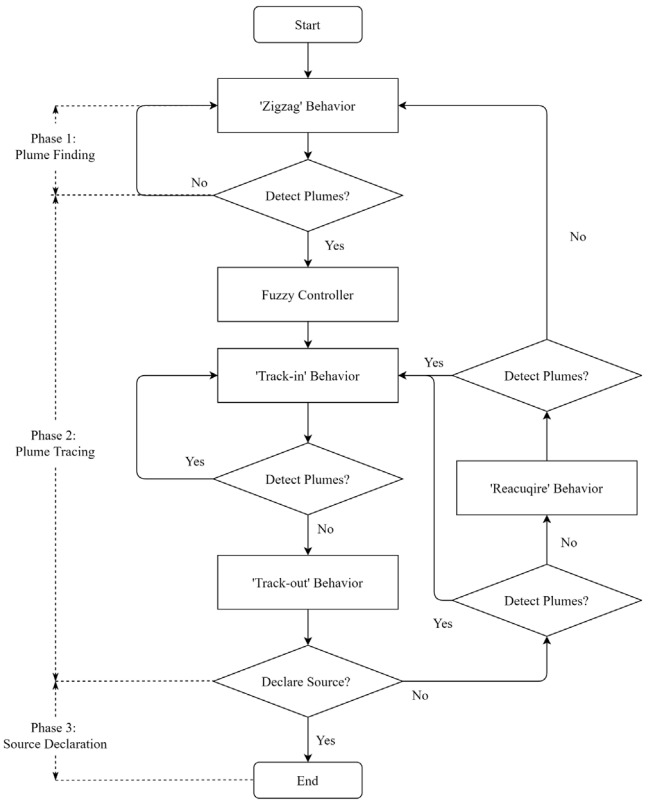

Fig. 1. The flow diagram of the proposed olfactory-based navigation algorithm.

In the plume tracing phase, the 'track-in' behavior is activated once the robot detects plumes. Similar to the 'surge' behavior of male moths, the 'track-in' behavior tries to make a rapid progress toward the odor source while plumes have been detected. When the robot moves out of plumes, the 'track-out' behavior is triggered to manipulate the robot to traverse the plume trajectory. The hope is that the robot can encounter plumes via the 'track-out' behavior, but if it does not, then the 'reacquire' behavior is activated. Like the 'casting' behavior of male moths, the robot in the 'reacquire' behavior performs crosswind excursions to detect plumes over a wide region. If the robot still cannot detect plumes, it will switch back to the plume finding phase and repeat the 'zigzag' behavior. Once the robot detects plumes, it turns back to the plume tracing phase and performs the 'track-in' behavior to trace plumes in the upwind direction.

In the source declaration phase, the robot exams whether it can confirm a source location after the 'track-out' behavior. If the robot collects enough information and is confident with the source estimation, it declares the estimated source location and completes the OSL task. The detailed source declaration procedures are presented in Section 4.3.

4. Behavior-based navigation algorithm

In this section, a bio-inspired navigation algorithm based on [21,23] is presented, which is designed inspired by mate-seeking behaviors of male moths. It should be mentioned that the OSL task defined in this work is a two-dimensional (2-D) problem since the aimed implementation platform is a ground robot.

4.1. Plume finding

The plume finding phase aims to verify the existence of plumes within the search area. Without any assumptions about the odor source location, the crosswind search is more efficient than the along-wind search to detect plumes since the robot is more likely to encounter plumes while moving in the crosswind direction [59]. Thus, the robot search trajectory is dominant with the crosswind movements and includes a smaller along-wind component to ensure the exploration.

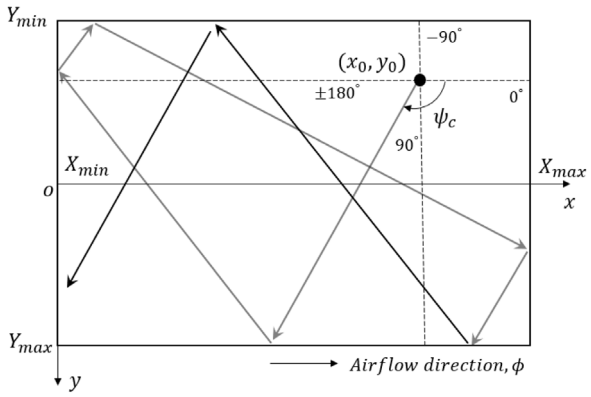

Fig. 2. The sample 'zigzag' search trajectory in the plume finding phase, where X_{min}, X_{max}, Y_{min}, and Y_{max} represent boundaries of the search area in the horizontal and vertical directions, respectively. The current robot location is at (x_0, y_0), and the commanded heading is ψ_c. The airflow direction presented in the diagram is just for the demonstration purpose only. When the airflow direction changes, the heading command ψ_c changes correspondingly to perform the crosswind movement.

Derived from these considerations, we design and implement the 'zigzag' behavior in this search phase. Fig. 2 presents a sample 'zigzag' search trajectory. It can be observed that the main features of this behavior include: (1) the robot moves predominantly across the airflow direction; (2) the trajectory also contains an along-wind component to cause the robot to explore new regions; (3) when the robot reaches boundaries, it turns the heading toward the inside of the search area to continue the search. While the robot maneuvers, the plume tracing phase is activated at any time when the sensed odor concentration exceeds the detection threshold, which terminates the current 'zigzag' search behavior.

4.2. Plume tracing

The objective of the plume tracing phase is to command the robot to approach the odor source via tracing emitted plumes. Three search behaviors are designed and implemented in this search phase, including 'track-in', 'track-out', and 'reacquire'.

4.2.1. 'Track-in' behavior

Inspired by the 'surge' behavior of male moths, the robot in the 'track-in' behavior is commanded to move upwind when it detects odor plumes. Studies in [60] reveal that immediately following a plume detection, a good plume tracing performance is attained by driving at an offset angle $\beta \in [20^{\circ}, 70^{\circ}]$ relative to the upwind direction. The benefit of this design is that when the robot drives out of plumes with a nonzero offset angle β, it can predict which side of plumes it exited from and perform a counter-turn to re-contact plumes.

To realize this mechanism, the heading command ψ_c in the 'track-in' behavior, which is positively defined in clockwise rotations and 0° is at the positive x axis, is presented as:

$$\psi_c = \phi + 180^{\circ} + LHS \cdot \beta, \tag{1}$$

where ϕ is the sensed airflow direction at the robot position, and *LHS* (i.e., stands for the phrase: "*L*eft *H* and *S*ide") is an indicator that is either ± 1. The value of *LHS* reflects the side of plumes

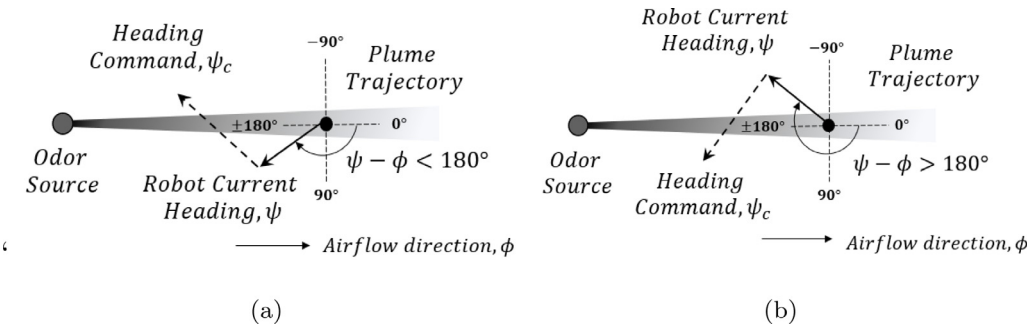

Fig. 3. Demonstrations for determining the value of *LHS* in the 'track-in' behavior, where (a) $LHS = 1$ and (b) $LHS = -1$.

Algorithm 1 'Track-in' Behavior

Set the speed command $v = v_c$
if sens.Odor $\geqslant$ threshold **then**
  Determine the value of *LHS*:
  if $(\psi$ - $\phi) < 180$ **then**
    $LHS = 1$
  else
    $LHS = -1$
  end if
  Update the heading command ψ_c:

$$\psi_c = \phi + 180 + LHS \cdot \beta$$

  Record the last detection time T_{last} and position $\mathbf{P}_{last}$:

$$T_{last} = t;\ \mathbf{P}_{last} = (x, y)$$

else
  if $(t - T_{last}) > \lambda$ **then**
    Save $\mathbf{P}_{last}$ into a list $\mathcal{LP}$:

$$\mathcal{LP}[i] = \mathbf{P}_{last};\ i++$$

    return 'Track-out' Behavior
  end if
end if
return 'Track-in' Behavior

that the robot would drive out, which is determined based on the difference between the robot heading angle ψ and the airflow direction ϕ, i.e., $\psi - \phi$. As presented in Fig. 3, if the robot drives out plumes from the left side of the plume trajectory (when looking upwind), i.e., $(\psi - \phi) < 180°$, the value of *LHS* is defined as 1; otherwise, the robot is expected to leave plumes from the right side, i.e., $(\psi - \phi) > 180°$, and the value of *LHS* is -1. In either case, the calculated heading command directs the robot to perform a counter-turn from the side that it drives out plumes and maintain inside the plume trajectory.

Algorithm 1 presents the pseudo-code for the 'track-in' behavior. When the robot detects plumes, i.e., the sensed odor concentration exceeds the threshold, the heading command ψ_c is updated via (1), and the plume detection time T_{last} and position $\mathbf{P}_{last}$ are recorded. When plumes are absent, the robot will hold the current heading for λ seconds to confirm the plume non-detection event. If the robot cannot detect plumes within λ seconds, i.e., $(t - T_{last}) > \lambda$ (t is the current time), the latest last plume detection location $\mathbf{P}_{last}$ will be added to a "last detection position" list, which is named as $\mathcal{LP}$. Then, the robot switches to the 'Track-out' behavior and inhibits the current 'track-in' behavior.

4.2.2. 'Track-out' behavior

The 'track-out' behavior attempts to make progress toward the odor source and quickly re-detect plumes in the vicinity of the last plume detection location. To achieve these two objectives, the robot is commanded to move toward a target point $\mathbf{P}_{target}$ that is L_u meters upwind and L_c meters crosswind from the most upwind position in the "last detection position" list, i.e., $\mathcal{LP}$.

Algorithm 2 presents procedures in the 'track-out' behavior. A critical step is to determine the correct crosswind direction: by following this direction, the robot is expected to traverse the plume trajectory and re-detect plumes. To achieve this mechanism, the indicator *LHS*, which predicts the side of plumes that the robot drives out, is employed to calculate $\mathbf{P}_{target}$. As presented in Fig. 4, if the robot leaves plumes from one side, the calculated target position will be located at the opposite side of plumes. Therefore, by proceeding to the target position, the robot is expected to move across the plume trajectory and re-detect plumes.

Algorithm 2 'Track-out' Behavior

Set the speed command, $v = v_c$
if sens.Odor >= threshold **then**
  if `SourceCheck` **then**
    return 'Source declaration' Behavior
  else
    return 'Track-in' Behavior
  end if
else
  Determine the target position, $\mathbf{P}_{target}$:
    (1) Find the upwind point $\mathbf{P}_{up}$ in $\mathcal{LP}$
    (2) Find the unit flow vector, $\mathbf{F} = (\cos\phi, \sin\phi)$
    (3) Rotate $\mathbf{F}$ clockwise by 90° to get $\mathbf{F}_p$
    (4) $\mathbf{P}_{target} = \mathbf{P}_{up} - L_u \cdot \mathbf{F} - L_c \cdot LHS \cdot \mathbf{F}_p$
  Calculate the heading command:

$$\psi_c = \arctan(y_{target} - y)/(x_{target} - x)$$

  if |sens.VehPosition - $\mathbf{P}_{target}$|< R **then**
    if `SourceCheck` **then**
      return 'Source declaration' Behavior
    else
      return 'Reacquire' Behavior
    end if
  end if
end if
return 'Track-out' Behavior

The 'track-out' behavior is terminated either when the robot detects plumes or reaches the target point. In either case, the robot checks whether it can declare the odor source location before determining the next behavior (i.e., `SourceCheck` in Algorithm 2): if a source location can be declared, the robot switches

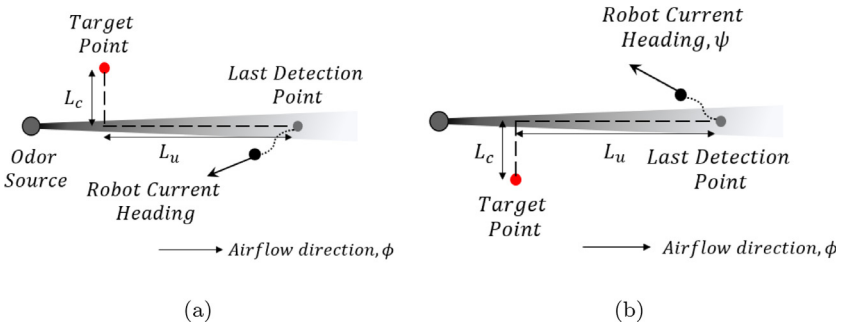

Fig. 4. Demonstrations for determining the target position $\mathbf{P}_{target}$ (painted in red) in the 'track-out' behavior, where (a) $LHS = 1$ and (b) $LHS = -1$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Algorithm 3 'Reacquire' Behavior

```
Set the speed command, v = v_c
if sens.Odor < threshold then
    Find the most upwind point in LP, P_up
    Find the unit flow vector, F = (cos φ, sin φ)
    P_ctr = P_up − D · F
    Perform the 'Bow-tie' trajectory at P_ctr and P_up
    if BowTie(P_ctr and P_up) == done then
        Remove the current P_up from LP
        if LP is empty then
            return 'Zigzag' Behavior
        end if
    end if
else
    return 'Track-in' Behavior
end if
return 'Reacquire' Behavior
```

to the source declaration behavior and completes the OSL task; if plumes are detected but the source cannot be declared, the robot turns to the 'track-in' behavior; if plumes are not detected and the source cannot be declared, the robot switches to the 'reacquire' behavior to search plumes over a larger scale.

4.2.3. 'Reacquire' behavior

The 'reacquire' behavior is designed to restore the plume contact in the situation where the robot fails to detect plumes in the 'track-out' behavior.

Biological studies [61,62] suggest that when a male moth loses contact with pheromone plumes, it ceases the upwind movement and progressively performs crosswind excursions. Inspired by this behavior, the Bow-tie trajectory is designed and implemented in the 'reacquire' behavior. This trajectory enables the robot to move across the airflow direction multiple times to increase the likelihood of re-detecting plumes. As shown in Fig. 5, a Bow-tie trajectory is constructed based on a center point $\mathbf{P}_{ctr}$, and by defining a Bow-tie radius K and an offset angle θ, four corner points ($\mathbf{P}_1$ to $\mathbf{P}_4$) are determined around $\mathbf{P}_{ctr}$. The robot will reach each corner point in ascending order to complete a Bow-tie trajectory. Besides, the robot turning radius is considered while designing the transition trajectory between two horizontal points.

If the robot fails to detect plumes in a Bow-tie, it will repeat this trajectory one more time to find plumes. Performing Bow-tie multiple times can improve the probability of detecting plume, but performing Bow-ties is also a time-consuming process. Here, the robot performs Bow-tie two times by following the design in [23]. As presented in Fig. 5, the initial Bow-tie trajectory is centered in $\mathbf{P}_{ctr}$, which is D meters away from the most upwind point $\mathbf{P}_{up}$ in the last detection position list, i.e., $\mathcal{LP}$. The next Bow-tie trajectory is centered at $\mathbf{P}_{up}$, and if the robot completes these two Bow-ties without plume detection, the current $\mathbf{P}_{up}$ will be removed from $\mathcal{LP}$. The robot then repeats the 'reacquire' behavior at the most upwind point on the remaining points in $\mathcal{LP}$. The 'reacquire' behavior is terminated when plumes are detected, which switches the robot to the 'track-in' behavior, or $\mathcal{LP}$ becomes empty, which reverts the robot to the plume finding phase, i.e., the 'zigzag' behavior.

4.3. Source declaration

The robot identifies and declares the odor source location in the source declaration phase. In this work, the spatial distribution of the last plume detection positions is utilized to declare the odor source location. As mentioned, the last plume detection positions are recorded at the end of 'track-in' behaviors. Since the odor source location is fixed, when the robot is far from the odor source, the last detection positions will be widely separated along the airflow direction; when the robot is close to the odor source, the last detection positions will be densely accumulated at the downflow area near the odor source location. This feature can be utilized to generate the source declaration algorithm, which includes two steps:

1. Recorded last detection positions are sorted with respect to the airflow direction, i.e., the most upwind point is placed at the beginning, then is the second upwind point, and so on;
2. When the first ϵ last detection positions differ in the airflow direction by less than γ meters, the most upwind point (i.e., the first point) is declared as the odor source location.

Values of ϵ and γ are determined such that the trade-off between the source declaration accuracy and the processing time is balanced. Generally, increasing the number of points used in the source declaration decision, i.e., ϵ, and decreasing the distance between consecutive points, i.e., γ, may increase the accuracy and reliability of the estimated source location but also increase the time required to satisfy the declaration criteria (sometimes even longer than the plume tracing process). In implementations, to obtain a well-algorithm performance and save the processing time, ϵ and γ are defined as 3 and 2, respectively. In practices, there are other ways to improve the localization accuracy without significantly increasing the search time, such as using cameras to determine the source location if the robot is close to it. Using visual sensors to assist the OSL task is also one of our future research directions.

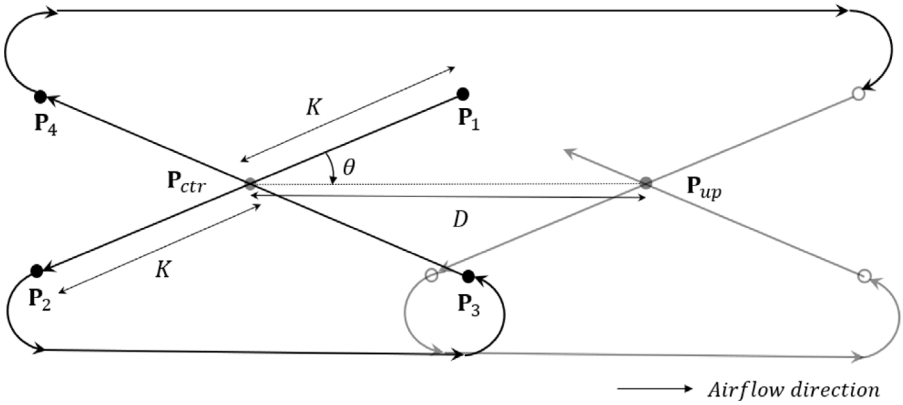

Fig. 5. The Bow-tie trajectory used in the 'reacquire' search behavior. The robot starts to perform the first Bow-tie centered in $\mathbf{P}_{ctr}$, and if the robot completes this Bow-tie without plume detection, it repeats the Bow-tie centered in $\mathbf{P}_{up}$. The distance between two Bow-ties is D meters.

5. Design of the fuzzy inference system

5.1. Design concept

Parameters in search behaviors, including β and λ in the 'track-in' behavior, L_u and L_c in the 'track-out' behavior, and K, θ, and D in the 'reacquire' behavior, determine the scale of robot trajectories and the transition between different search behaviors. Traditional bio-inspired methods, such as [21,23], assign values of these parameters by trial and error prior to the search. Once parameter values are settled, they are fixed during the entire search, which is not an ideal setting for real-world environment with varying airflow fields.

An improved design is to adjust search parameters according to the current search situation dynamically. For instance, when the airflow becomes turbulent, odor plumes are advected by strong winds and meander in a wide range. In this scenario, large values of β, L_c, K, and θ are preferred to generate an oscillating search trajectory covering a large search area, which is beneficial for detecting meandering plumes over a wide range. Conversely, when the airflow is laminar, values of λ, L_u, and D should be large to produce a smooth search trajectory, which is desired to keep the robot maintaining inside a stable plume trajectory. Besides the airflow characteristics, the distance between the robot and the odor source is also critical to affecting values of search parameters: if the robot is near the source, it should search locally to exploit the exact odor source location, i.e., values of parameters should be small to constrain the scale of search trajectories; otherwise, the robot should explore the search area over a broad region to collect the odor source information, i.e., values of parameters should be large to extend the scale of search trajectories.

The challenging part of designing such a controller is that it is required to analyze the current search situation and transfer the changes of search situations to a quantitative mechanism that adjusts search parameters. Considering uncertainties in the source location and search environment, mathematical methods to quantitatively analyze differences in search situations are challenging to implement. Inspired by the implementations of fuzzy theory in the field of decision-making [63] and data classification [64], which successfully handles the problems with vagueness and uncertainties, a fuzzy inference approach is employed to perceive the environment and adjust parameters in search behaviors. In the fuzzy theory, vague variables and environments can be handled in a deterministic manner via linguistic descriptions and rules. Therefore, by analyzing sensor data, such as airflow measurements and odor concentrations, the current search situation is expected to be identified by the fuzzy controller. Then, values of behavior parameters can be adapted based on the defined fuzzy rules to achieve an optimal search performance.

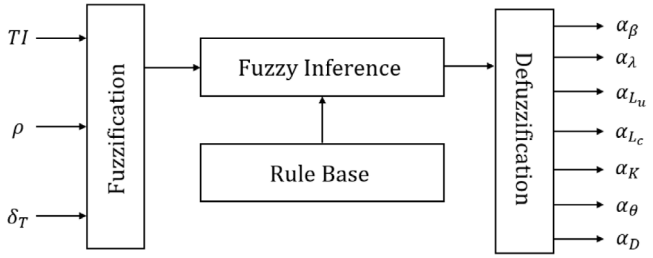

Fig. 6. Schematic diagram of the proposed fuzzy logic controller. TI is the turbulence intensity of the search environment, ρ is the sensed odor concentration at the robot position, and δ_T is the non-detection period. α_β, α_λ, ... , α_D are coefficients that adjust behavior parameters.

5.2. Define inputs and outputs of the Fuzzy controller

The block diagram of the proposed fuzzy controller is presented in Fig. 6. Inputs of the fuzzy controller are utilized to conjecture the current search situation, and outputs are coefficients that adjust values of behavior parameters. Search situations are categorized into four types with linguistic descriptions, namely laminar, turbulent, near the source, and far from the source. The first two search situations describe the search environment's airflow characteristics, and the last two indicate how close the robot is to the odor source.

There are three signals as the inputs of the fuzzy controller, namely turbulence intensity TI, sensed odor concentration at the robot position ρ, and the plume non-detection period δ_T, i.e., the period since the last detection event. Among inputs, turbulence intensity TI is a parameter that estimates airflow characteristics, which is defined as [65]:

$$TI = \frac{\sigma_u}{m_u}, \tag{2}$$

where σ_u and m_u are the standard deviation and the mean of airflow velocities u, respectively, and σ_u is calculated as [66]:

$$\sigma_u = \sqrt{\frac{1}{2}(\sigma_{u_x}^2 + \sigma_{u_y}^2)}, \tag{3}$$

where σ_{u_x} and σ_{u_y} are the standard deviation of airflow velocities on the x and y directions, i.e., u_x and u_y. Besides, m_u is computed from mean velocity components:

$$m_u = \sqrt{m_{u_x}^2 + m_{u_y}^2}, \tag{4}$$

where m_{u_x} and m_{u_y} are the mean of u_x and u_y. It can be observed that a larger TI indicates a higher level turbulence in the airflow environment.

In implementations, values of TI are computed based on the most recent airflow measurements within H seconds (the sampling frequency of the airflow sensor in implementations is 100 Hz). Various values of H have been evaluated (including $H = 20$, 50, 100, and 200), and test results show that when H is small, TI is more sensitive to instantaneous airflow changes in the environment, and when H is large, TI is sluggish to the instantaneous airflow changes and becomes the averaged airflow value over a period. In this work, TI is expected to reflect the instantaneous airflow changes in the robot surrounding environment; thus, based on test results, H is determined as 20 s in implementations.

The remaining two fuzzy inputs are the sensed odor concentration ρ and the plume non-detection period δ_T. Unlike traditional olfactory-based navigation algorithms, such as [23,35], which simplify the magnitude of ρ as a binary detector, this work utilizes the analog concentration signal to estimate the distance from the robot to the odor source. Notice that, due to the existence of local concentration maxima along the plume trajectory [67], a single high concentration detection is not enough to prove that the robot is near the odor source. Therefore, δ_T is added to the fuzzy inputs to assist the estimation. Since the positions of local concentration maxima are time-varying in turbulent flow environments [68], if the robot consecutively detects high odor concentrations in a short period, i.e., ρ is high and δ_T is short, the robot is very likely to be near the source.

Outputs of the fuzzy controller are a group of coefficients α that adjust search behavior parameters, including α_β, α_λ, α_{L_u}, α_{L_c}, α_K, α_θ, and α_D. Each coefficient is applied on the base values of behavior parameters, such as $\beta = \alpha_\beta \cdot \beta_{base}$, $\lambda = \alpha_\lambda \cdot \lambda_{base}$, etc. The output range of all coefficients is from 0 to 1. Therefore, by varying values of coefficients, behavior parameter values will be changed correspondingly.

5.3. Procedures of the fuzzy controller

5.3.1. Fuzzification

The fuzzification is a procedure that maps the crisp input values to linguistic fuzzy terms with a membership value between 0 and 1, which represents the degree of uncertainty that input values belong in a fuzzy set.

Fig. 7 presents plots of membership functions and fuzzy sets for fuzzy inputs and outputs, and the Gaussian membership function is selected to implement on all fuzzy sets. Since the number of potential fuzzy rules depends on the size of fuzzy sets of input variables, small numbers of fuzzy sets are defined for each input: three fuzzy sets are defined for the turbulence intensity TI, namely laminar (La), averaged (Av), and turbulent (Tu); two fuzzy sets are defined for the sensed odor concentration ρ, namely low (L) and high (H); two fuzzy sets are defined for the plume non-detection period δ_T, namely short (Sh) and long (Lo). For all fuzzy outputs (i.e., α_β, α_λ, ..., α_D), the fuzzification procedure is identical since the discourse of universe of each coefficient is equivalent, i.e., $\alpha \in [0, 1]$. Specifically, five fuzzy sets are defined over the discourse of universe of the output, namely very small (VS), small (S), middle (MI), big (B), and very big (VB).

5.3.2. Fuzzy rules

Fuzzy rules are responsible for the decision making in a fuzzy controller, which govern the input–output relationship. Each rule is given by a "IF-THEN" statement [69], where the "IF" part defines the combination of fuzzy inputs, and the "THEN" part specifies the consequent results of fuzzy outputs.

In the proposed fuzzy controller, fuzzy rules stipulate how the robot reacts to different search situations. Here, we design fuzzy rules based on our previous research experiences in both bio-inspired and engineering-based plume tracing algorithms. We discover that the search efficiency can be improved if the robot can extend its crosswind excursions to explore larger areas when it is in a turbulent environment and decrease the intensity of crosswind movements to save time if the robot is in a laminar flow environment. Borrowing this idea, we want the robot to explore if it is in a turbulent environment and exploit when the surrounding airflow environment is laminar.

Table 1
List of fuzzy rules. La: Laminar; Av: Averaged; Tu: Turbulent; L: Low; H: High; Sh: Short; Lo: Long; VS: Very Short; S: Short; MI: Middle; B: Big; VB: Very Big.

Rule No.	Inputs			Outputs						
	TI	ρ	δ_T	α_β	α_λ	α_{L_u}	α_{L_c}	α_K	α_θ	α_D
1	La	L	Sh	VS	B	B	VS	VS	VS	B
2	La	L	Lo	S	VB	VB	S	S	S	VB
3	La	H	Sh	VS	S	S	VS	VS	VS	S
4	La	H	Lo	VS	MI	MI	VS	VS	VS	MI
5	Av	L	Sh	B	MI	MI	B	MI	B	S
6	Av	L	Lo	B	MI	MI	B	B	B	S
7	Av	H	Sh	VS	VS	VS	S	VS	VS	VS
8	Av	H	Lo	S	VS	VS	S	VS	S	S
9	Tu	L	Sh	B	VS	VS	B	B	B	VS
10	Tu	L	Lo	VB	S	S	VB	VB	VB	S
11	Tu	H	Sh	S	VS	VS	S	S	S	VS
12	Tu	H	Lo	MI	VS	VS	MI	MI	MI	VS

To achieve this mechanism, the characteristics of the airflow environment is estimated. If the environment is turbulent, values of α_β, α_{L_c}, α_K, and α_θ are increased to generate oscillating search trajectories to improve the exploration; otherwise, these values are decreased and α_λ, α_{L_u}, and α_D are increased to generate smooth trajectories that emphasize upwind movements, which help the robot to approach the odor source in a laminar environment quickly. Additionally, values of α are fine-tuned by the estimated distance between the robot and the odor source: if the robot is far from the odor source, the robot inclines to find plumes over a large area, i.e., exploration; otherwise, the robot tends to search the odor source within a constrained area, i.e., exploitation. Specifically, in the stage of exploration, values of α are increased to generate the large scale search trajectories, and in the stage of exploitation, values of α are constrained to limit the scale of search trajectories.

An example fuzzy rule is demonstrated as follow: when TI is laminar, ρ is high, and δ_T is short, the robot is expected to be close to the odor source in a laminar environment; thus, α_λ, α_{L_u}, and α_D are increased and α_β, α_{L_c}, α_K, and α_θ are decreased to render a smooth trajectory in the current laminar environment, and considering the close distance to the odor source, values of all coefficients should be constrained in a small scale to generate local search trajectories within a limited area. In the "IF-THEN" format, the above rule can be presented as:

$\mathcal{F}^1$ = {IF TI is La AND ρ is H AND δ_T is Sh, THEN α_β is VS AND α_λ is S AND α_{L_u} is S AND α_{L_c} is VS AND α_K is VS AND α_θ is VS AND α_D is S.}

Enumerate all possible combinations of inputs and outputs, Table 1 presents twelve fuzzy rules in the proposed fuzzy controller.

5.3.3. Defuzzification

Defuzzification is a procedure that maps the fuzzy output to a crisp signal. In this work, the centroid method [69] is selected as the defuzzification algorithm, which can be expressed as follow:

$$\alpha = \frac{\sum_{i=1}^{n} Q_i \cdot \mu(Q_i)}{\sum_{i=1}^{n} \mu(Q_i)}, \tag{5}$$

where α is the coefficient value, which could be α_β, α_λ, ... ; i is the index of fuzzy rules, i.e., $i \in [1, 12]$; Q_i denotes the center of

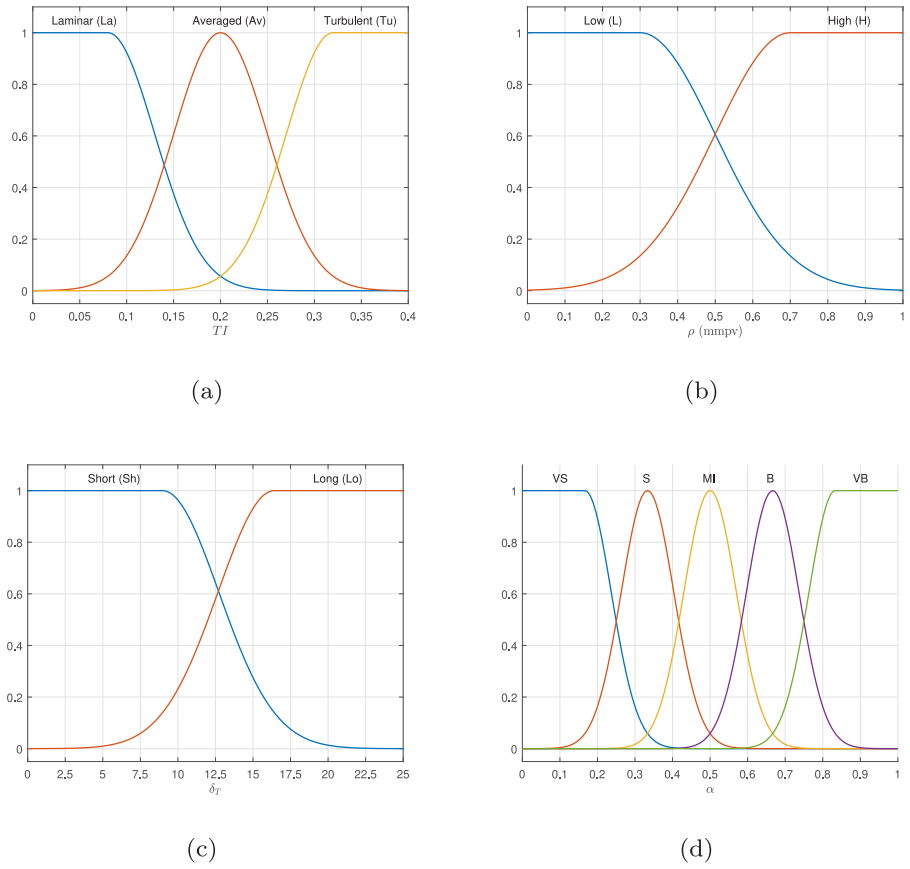

Fig. 7. Membership functions and fuzzy sets for inputs and output of the fuzzy controller. Inputs include: (a) turbulence intensity TI, (b) sensed odor concentration ρ, and (c) plume non-detection period δ_T. The output is (d), which represents coefficients of behavior parameters, i.e., α_β, α_λ, ..., α_D.

the fired membership function of the output variable provided by the ith rule; $\mu(Q_i)$ is the output of the conjunction degree of the IF part of the ith rule.

6. Simulations and results

In this section, we evaluate the effectiveness of the proposed olfactory-based navigation algorithm and compare the results with traditional bio-inspired [23] and engineering-based [35] methods. The simulation program used in this work is constructed based on [6]. Based on our previous on-vehicle experiment results [23,70], the simulation based evaluation tool allows graphical and batch statistical analysis of plume tracing performance. It allows the user to analyze performance using either a simulated plume or data obtained from field or plume experiments. The simulation includes an evolving flow field, a coherent chemical plume with realistic short term chemical signatures and long term exposure, and a full six degree of freedom autonomous vehicle dynamics. Other research works, such as [71–75], also utilized this simulator as the evaluation platform.

6.1. Simulation setup

6.1.1. Simulated environment

Fig. 8 presents the simulated search area, where a coordinate (x–y) is constructed to represent positions. An odor source is located at (20, 0) m and releases 10 filament packages (i.e., plumes) per second. Released plumes form a circular plume trajectory as plotted by a gray-scale patchy trail. Arrows in the background represent airflow vectors, where the tail of an arrow points to the airflow direction and the length of an arrow indicates the strength of airflow velocity. In this simulator, airflow vectors are calculated from time-varying boundary conditions that are generated by a mean flow ($\mathbf{U}_0$) and Gaussian white noise (zero mean and ς variance). By changing values of boundaries condition variables (i.e., $\mathbf{U}_0$ and ς), different amplitudes of airflow fields can be obtained. Fig. 9 shows snapshots of two simulated airflow fields. In the left diagram, a laminar airflow environment is created with $\mathbf{U}_0 = (1, 0)$ m/s and $\varsigma = 0$, and in the right diagram, a turbulent airflow environment is generated with $\mathbf{U}_0 = (2, 0)$ m/s and $\varsigma = 20$.

6.1.2. Vehicle assumptions

In the simulation program, a two-wheeled mobile robot is employed to implement the proposed navigation algorithm. Comparing to the large scale of the search area, the size of the robot is negligible. Therefore, the robot is approximated as a single point in the simulation program. It is assumed that the robot is equipped with a chemical sensor, an anemometer, and a positioning sensor, which measure odor concentrations, wind speeds and directions in the inertial frame, and the robot position in the inertial frame, respectively. All sensors' measurements are corrupted with Gaussian white noises to imitate real-world applications, where noise parameters are listed in Table 2. The proposed navigation algorithm is operated on an onboard computer to process sensor readings and calculate the heading and speed commands, which are limited in ranges of $\psi_c = [-180°, 180°]$ and $v_c = [0.6, 1]$ m/s, respectively.

The sampling frequency of all sensors is 100 Hz, while the implemented navigation algorithm produces a command per second. Considering olfactory sensing is characterized by very low false alarm rates but potentially high missed detection rates [13],

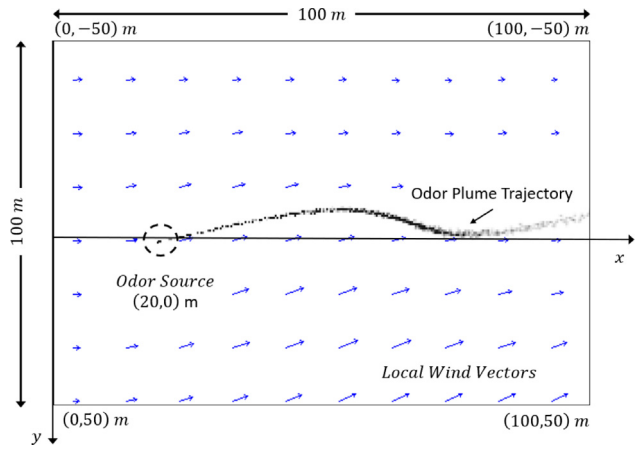

Fig. 8. The simulated search area. The size of the search area is 100 × 100 m^2, and a fixed location odor source is placed at (20, 0) m, which emits 10 plumes per second. Emitted plumes form a curvy trajectory as plotted by the gray-scale patchy trail. Airflow vectors, which are represented by blue arrows in the background, are calculated from a mean flow $\mathbf{U}_0$ and a variance ς. By changing these two variables, different airflow fields can be obtained.

Table 2
Values of parameters in Gaussian noises in sensor measurements.

	Chemical sensor	Anemometer	Positioning sensor
Mean	0	0	0
Standard deviation	0.05 mmpv[a]	0.1 m/s and 1°	0.1 m

[a]mmpv: million molecules per cm^3.

we pick the highest chemical sensor reading during one decision-making period (i.e., 1 s) to identify whether the robot detects odor plumes. Here, a concentration threshold is employed to determine a detection event: if the sensed concentration exceeds the threshold, the detection event is confirmed; otherwise, the robot does not detect odor plumes. For other sensors, the averaged value of measurements among one decision-making period is fed to the navigation algorithm.

6.1.3. Experiment designs

Around 400 tests have been conducted in the simulation to evaluate the performance of the proposed olfactory-based navigation algorithm, which can be separated into three groups.

In the first group of tests, the effectiveness of the proposed navigation algorithm in a laminar flow environment is evaluated. Besides, robot trajectories in different search behaviors are demonstrated and compared with those generated without the designed fuzzy controller. Tests in the second group are carried out to investigate the validity of implementing the proposed navigation algorithm in a turbulent flow environment. Snapshots of robot trajectories at different time steps and the plots of fuzzy inputs and outputs are also presented. Tests in the last group are designed for evaluating the robustness of the proposed navigation algorithm, where various search conditions, including varying robot initial positions and airflow environments, are defined. Additionally, results of the proposed navigation algorithm in these tests are compared with traditional olfactory-based navigation algorithms.

6.2. Group 1: Implementation in a laminar flow environment

In this group of tests, the robot is placed in a laminar flow environment, where $\mathbf{U}_0 = (1, 0)$ m/s and $\varsigma = 3$. The robot initial position is at (80, −40) m and moves at a constant speed 1 m/s. Fig. 10 presents robot trajectories in different search behaviors with and without the designed fuzzy controller.

At the initial phase of the search, the 'zigzag' trajectory is employed to guide the robot in detecting plumes. After the first plume detection event, the robot switches to the 'track-in' behavior. Comparing Figs. 10(a) and 10(d), it can be observed that with the proposed navigation algorithm, the trajectory length is longer than the counterpart in the 'track-in' behavior (23 m vs. 17 m). This is because the proposed fuzzy controller increases the value of λ in the 'track-in' behavior due to the laminar flow environment. This operation elongates the upwind movement, which helps the robot quickly approach the odor source.

Then, as shown in Figs. 10(b) and 10(e), the 'track-in' and 'track-out' behaviors alternate to command the robot proceeding toward the odor source location. When the robot is close to the odor source, the fuzzy controller decreases the value of behavior parameters to constrain the scale of trajectories; therefore, the robot can perform a local search within a small area to exploit the exact odor source location. Comparing Figs. 10(c) and 10(f), which present robot trajectories in the 'reacquire' behavior, the trajectory with the fuzzy controller is more favorable since the robot circulates near the odor source location; by contrast, the trajectory without the fuzzy controller has a large scale, which is not efficient for saving the search time.

At the end of the search, the robot declares the odor source once the source declaration conditions are satisfied. The search time and the declared odor source location for the robot with the designed fuzzy controller is 175 s and (21.7, 0.1) m, respectively. The distance to the real odor source location, i.e., (20, 0) m, is 1.7 m. Considering the large size of the search area, i.e., 100 × 100 m^2, the performance of the proposed navigation algorithm is satisfied (within 5 m). Compared to the search results of the traditional bio-inspired method, where the robot uses 351 s and declares the odor source at (21.2, 0.2) m, the proposed navigation algorithm is more efficient in Group 1 tests.

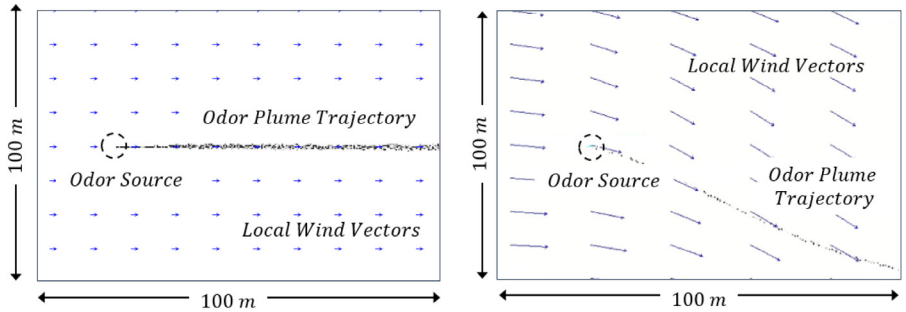

Fig. 9. Airflow fields and corresponding odor plume trajectories in the simulation. (a) Laminar Flows. (b) Turbulent Flows.

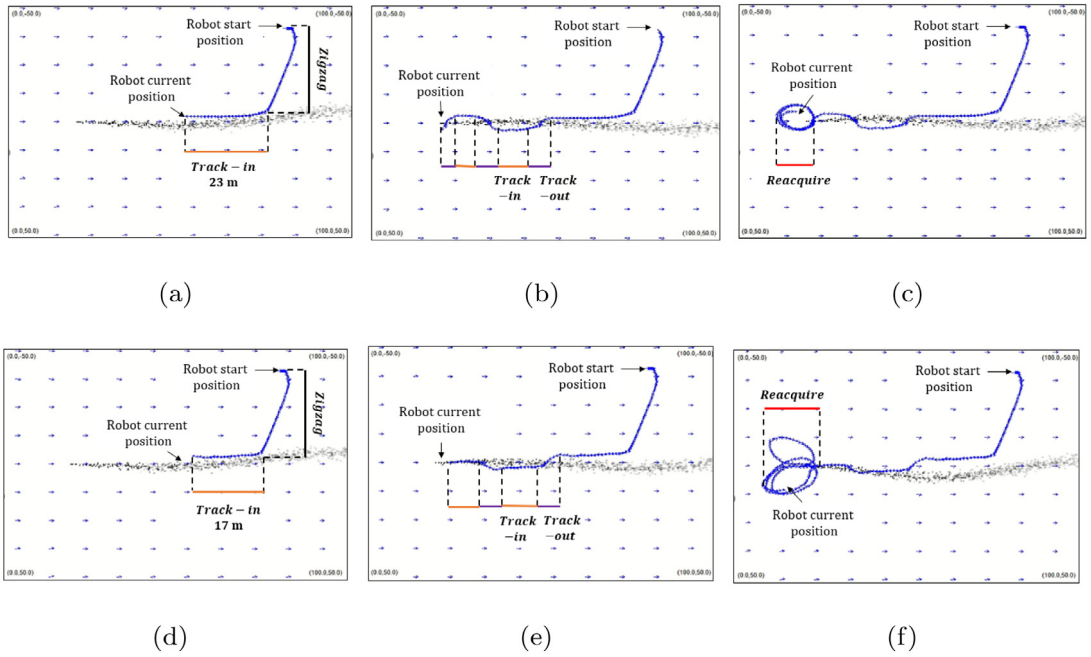

Fig. 10. Snapshots of robot trajectories in the Group 1 tests. Among these diagrams, three top diagrams, i.e., (a), (b), (c), are robot trajectories generated with the proposed method, while the remaining diagrams, i.e., (d), (e), (f), are trajectories generated by the original bio-inspired method. Over these robot trajectories, durations of search behaviors are labeled by different color bars, where black is 'zigzag'; orange is 'track-in'; purple is 'track-out'; red is 'reacquire'. In Group 1 tests, the robot moves in a constant speed at 1 m/s. With the proposed method, the robot correctly locates the odor source with 175 s, which is much shorter compared to 351 s for the traditional bio-inspired counterpart. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

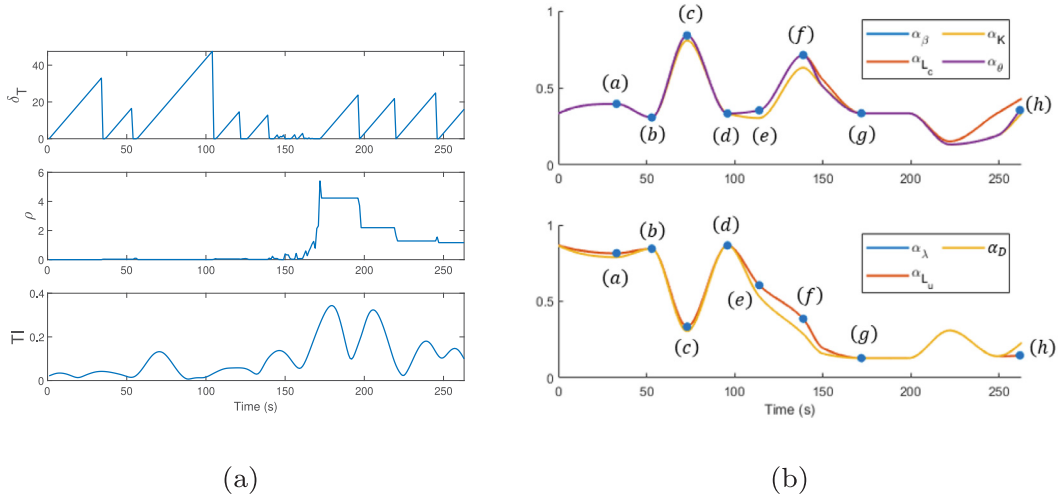

Fig. 11. Plots of fuzzy inputs and outputs generated in the Group 2 test, where (a) presents fuzzy inputs, and (b) shows fuzzy outputs. Labels on the fuzzy output plots, i.e., (a), (b), ... (h), are time steps that mentioned in Fig. 12.

6.3. Group 2: Implementation in a turbulent flow environment

In this group of tests, the proposed navigation algorithm is implemented in a turbulent flow environment, where $\mathbf{U}_0 = (1, 0)$ m/s and $\varsigma = 10$.

In Fig. 11, the change of fuzzy inputs and the corresponding fuzzy outputs, i.e., coefficient values, can be inspected and compared. When the airflow environment around the robot position is laminar, i.e., *TI* is at a low level, values of α_β, α_{L_c}, α_K, and α_θ, are low either, and values of α_λ, α_{L_u}, and α_A are high. Conversely, when the airflow environment becomes turbulent, i.e., *TI* grows to a high level, values of α_β, α_{L_c}, α_K, and α_θ increase and α_λ, α_{L_u}, and α_D decrease correspondingly. Besides, the shift of the robot from exploration to exploitation can also be observed. When the robot is near the odor source location, i.e., ρ is high and δ_T is short, values of all coefficients are constrained to produce a limited scale trajectory, which commands the robot to exploit the exact odor source location. Note that, the value of ρ will not be updated until the next above-threshold concentration measurement, where the threshold is defined to filter out the background concentration noises. Since the chemical sensor has a low false-alarm rate but a high miss-detect rate [13], this design enables the robot to memorize the most recent above-threshold concentration measurements, which helps the robot estimate the distance to the source.

Search details are demonstrated in Fig. 12. At $t = 33$ s, the robot encounters plumes for the first time and switches to the 'track-in' behavior. In the period from $t = 50$ s to $t = 75$ s,

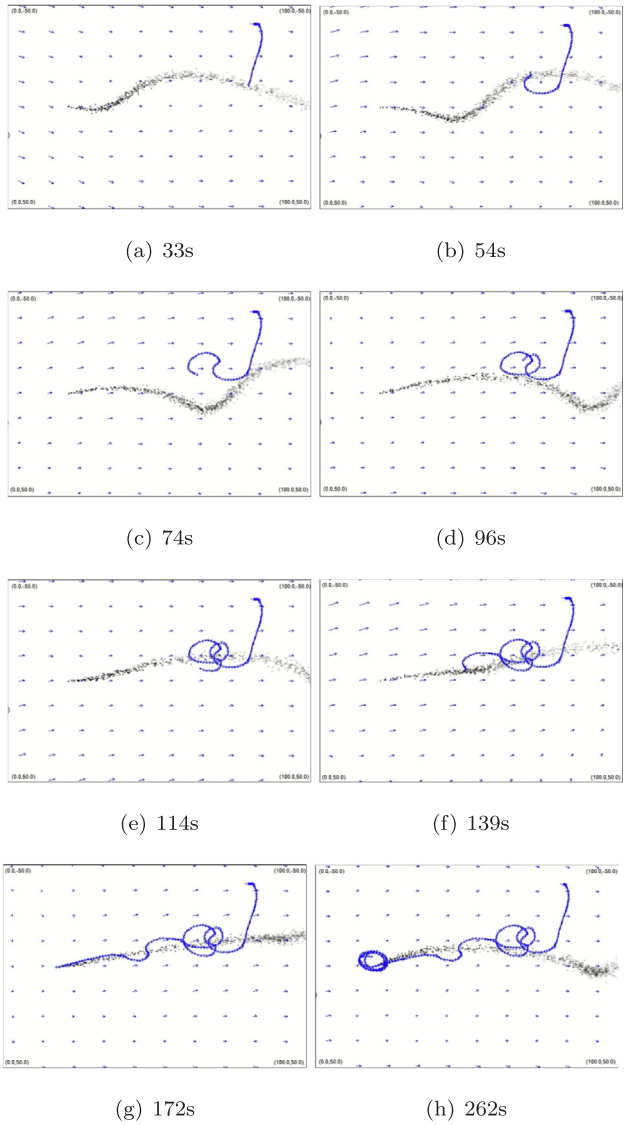

Fig. 12. Snapshots of robot trajectories at different time steps in the Group 2 test. The robot is placed in a turbulent airflow environment with the environmental settings $\mathbf{U}_0 = (1, 0)$ m/s and $\varsigma = 10$. The robot starts at $(80, -40)$ m and declares the odor source location at 262 s. The declared odor source location is at $(20.2, 0.1)$ m, which is 0.22 m to the actual odor source location. It should be noted that the airflow field displayed at the background is at the current time while the shown plume trajectory is developed over a period.

due to the turbulent airflows, the robot loses the plume contact and performs the 'track-out' and 'reacquire' behaviors to recover plumes. As presented in Fig. 12(c), the robot moves in circles to find plumes. Meanwhile, it can be observed in Fig. 11(b) that the fuzzy controller increases values of α_β, α_{L_c}, α_K, and α_θ and decreases α_λ, α_{L_u}, and α_D to generate the large scale trajectories for fostering the plume search over a large area. At around $t = 100$ s, the robot re-detects plumes and switches back to the 'track-in' behavior; however, this behavior does not last long due to the circulating plume trajectory produced by turbulent airflows. At $t = 139$, after performing a 'track-out' behavior to traverse the plume trajectory, the robot maintains inside plumes until $t = 172$ s. During this period, the robot quickly approaches the odor source location. In the plots of coefficients, i.e., Fig. 11(b), coefficient values are constrained to a low level after $t = 172$ s, which results in the small scale trajectories. It can be seen in Fig. 12(h) that the robot circulates around the odor source location with a small scale trajectory. At $t = 262$ s, the robot declares the odor source location, which is located at $(20.2, 0.1)$ m. The distance error to the actual odor source location is 0.22 m.

6.4. Group3: Comparisons to traditional methods

In this group of tests, the robustness of the proposed navigation algorithm in varying search conditions is evaluated. Two test scenarios have been designed as follows:

- Scenario 1: the robot starts the OSL task at different initial positions in a turbulent flow environment, where $\mathbf{U}_0 = (1, 0)$ m/s and $\varsigma = 8$.
- Scenario 2: the robot starts at the same initial position, i.e., $(80, -40)$ m, but airflow environments vary.

Additionally, to evaluate the performance of the proposed navigation algorithm, a traditional bio-inspired method [23] and an engineering-based method [12] are also implemented and compared in this group of tests. It should be mentioned that for each navigation algorithm, the robot operates at the same speed, i.e., 1 m/s, and with the same source declaration algorithm (presented in Section 4.3).

6.4.1. Results of scenario 1

Table 3 (at Page 42) presents search results in Scenario 1 tests, where six different robot initial positions over the search area are evaluated.

For the proposed navigation algorithm, it can be observed in Table 3 that no matter how the robot initial position changes, the robot can correctly declare the odor source within 5 m to the actual source location. Compared to other two methods, the proposed method achieves shorter search time in Test 3–6, while for Test 1–2, the search time of the proposed method is comparable to the engineering-based counterpart. However, the time complexity of the implemented engineering-based method grows significantly with respect to the size of the search area, i.e., when the search area is large, the robot needs more time to generate a source probability map. The proposed algorithm, on the other hand, does not relate with the size of search area, where robot commands are calculated directly based on the sensors' measurements. Considering the long processing time of the implemented engineering-based algorithm, the proposed method is more preferable for implementations. Besides, we can select the task time when the airflow is slowly time-varying to facilitate the plume tracing process. Test results in Scenario 1 demonstrate the validity of the proposed navigation algorithm with varying robot initial positions.

6.4.2. Results of scenario 2

In Scenario 2, the proposed navigation algorithm is implemented in five different airflow environments. For an airflow environment, each navigation algorithm was repeatedly performed 20 times. In total, each navigation algorithm was performed 100 times in this group of tests. Search results of each navigation methods in different airflow environments are presented in Table 4, and statistical results are presented in Table 5.

Table 4 demonstrates that the proposed navigation algorithm outperforms the traditional bio-inspired method in all environments concerning the mean search time. Compared to the engineering-based method, the proposed algorithm achieves a shorter search time all environments except Env. 2, where the engineering-based method is slightly better than the proposed algorithm in terms of the mean search time (i.e., 227.2 s vs.

Table 3
Different robot initial positions and search results in scenario 1 tests.

Tests	Robot initial positions (m)	Traditional Bio-inspired method [23]			Traditional engineering-based method [12]			The proposed navigation algorithm		
		Search time (s)	Declared source location (m)	Error to the actual source (m)	Search time (s)	Declared source location (m)	Error to the actual source (m)	Search time (s)	Declared source location (m)	Error to the actual source (m)
1	(35, 20)	250	(20.1, −0.2)	**0.2**	**138**	(21.5, −0.1)	1.5	151	(23.9, 1.0)	4.0
2	(50, 40)	279	(20.4, 0.2)	**0.5**	**140**	(21.1, −0.3)	1.1	215	(20.7, 0.2)	0.7
3	(60, −40)	343	(22.1, −1.2)	2.4	193	(20.5, −0.1)	**0.5**	**189**	(20.7, −0.3)	0.8
4	(70, 30)	345	(20.5, −0.5)	0.7	164	(20.9, −0.1)	0.9	**157**	(20.3, −0.3)	**0.4**
5	(90, 40)	313	(20, −0.1)	**0.1**	323	(19.9, 0)	0.1	**195**	(21.6, −0.7)	1.8
6	(90, −30)	399	(19.9, 0)	**0.1**	467	(27.7, −0.8)	7.7	**188**	(21.1, −0.6)	1.3

Table 4
Search results of three navigation methods in different airflow environments. f: successful/total tests; μ: averaged search time; σ: standard deviation of search time.

Environmental settings		Traditional bio-inspired method [23]			Traditional engineering-based method [12]			The proposed navigation method		
$\mathbf{U}_0$ (m/s)	ς	f	μ (s)	σ	f	μ (s)	σ	f	μ (s)	σ
(1, 0)	8	20/20	336.5	57.0	18/20	297.1	65.7	**20/20**	**251.8**	**42.7**
(1, 0.5)	5	20/20	322.0	45.0	**20/20**	**227.2**	**36.0**	20/20	245.4	30.0
(1, 0)	10	20/20	326.7	32.8	20/20	300.9	78.1	**20/20**	**243.2**	**56.4**
(2, 0)	10	20/20	331.2	24.0	20/20	304.4	68.6	**20/20**	**252.8**	**36.8**
(2, 0.3)	15	20/20	349.5	40.8	20/20	334.8	63.2	**20/20**	**313.3**	**68.4**

Table 5
Statistical results of repeated tests.

	Successful/Total tests	Success rate	Standard deviation of search time	Averaged search time (s)
Traditional bio-inspired method [23]	100/100	100%	42.5	333.2
Traditional engineering-based method [12]	98/100	98%	78.8	292.9
The proposed navigation algorithm	100/100	**100%**	**55.6**	**260.7**

245.4 s). Table 5 presents the statistic results of three navigation algorithms. It can be seen that the proposed navigation algorithm achieves the shortest averaged search time among three navigation algorithms, which indicates that the proposed algorithm is more efficient than the other two methods in these tests. As for the standard deviation of search time, the proposed method has a smaller value than the engineering-based method and comparable to the bio-inspired method, indicating that the proposed method is reliable in search performance. This result verifies the effectiveness of the proposed method, which enables the robot to exploit the exact odor source location when it is near the source. Considering the great computational demand of the engineering-based method, the proposed algorithm is more favorable to be implemented on mobile robots due to the low computation load and acceptable localization accuracy.

In general, the main advantage of the proposed navigation algorithm is that the computation complexity is much less, but the search performance is comparable to the engineering-based method [12]. The navigation method proposed in [12] contains a cell-based source mapping algorithm and a cell-based plume mapping algorithm, generating a source probability map and a plume likelihood map, respectively. In cell-based mapping algorithms, the search area is divided into multiple small cells, and the computational cost relates to the number of cells defined in the search area [13]. According to [13], the calculation of plume likelihood map needs $N^2 \cdot T$ operations per time step, where N is the number of cells in the search area and T is the plume release period. Notice that the plume release period increases with the progress of the search time. In comparison, the proposed navigation algorithm's computational complexity is fixed (i.e., it does not need to compute the cell-based probability). In the proposed method, the robot is commanded based on sensor readings at the current time step, which is more reliable and much faster than the engineering-based counterpart in varying search environments.

7. On-vehicle implementations

7.1. Experiment setup

To verify the effectiveness of the proposed algorithm, we implemented the proposed method on a mobile robot to find the odor source in real airflow environments. As presented in Fig. 13(a), a mobile robot was constructed and employed as the robotic agent, which carries a comprehensive sensor suite for perceiving the environment. The search area, as presented in Fig. 13(b), contains an odor source and an electrical fan. In an OSL test, the odor source location is hidden to the robot, and the implemented olfactory-based navigation method directs the robot to find the odor source.

Experiments were conducted in the indoor autonomous robots testing lab at the Embry-Riddle Aeronautical university. We divided the lab into two areas, including a search area where the robot can move and an operation area for accommodating the ground station. As shown in Fig. 14(a), the size of the search area is 9×4 m^2, containing an odor source and an electrical fan. We defined the size of search area according to the longest sensing distance of the employed chemical sensor. In addition, the lab contains a localization system (i.e., Vicon), which provides accurate positions and orientations of the robot. The main airflow direction is created by the electrical fan, but the overall airflow field is turbulent since there are four vents mounted on the wall that occasionally blow winds to maintain the room temperature. The ethanol vapor was employed as the odor source since it is minimally toxic and commonly implemented in OSL research [76].

Fig. 14(b) shows the configuration of the implemented system. There are three main components, including a mobile robot, a ground station, and an indoor localization system. The robot is equipped with a chemical sensor (MQ-3, Waveshare) and an anemometer (WindSonic, Gill Instruments) for measuring odor concentration and airflow speed and direction at the robot position. Both sensors are connected to a micro-controller (Arudino Mega, Arduino) for fetching sensor measurements. The second

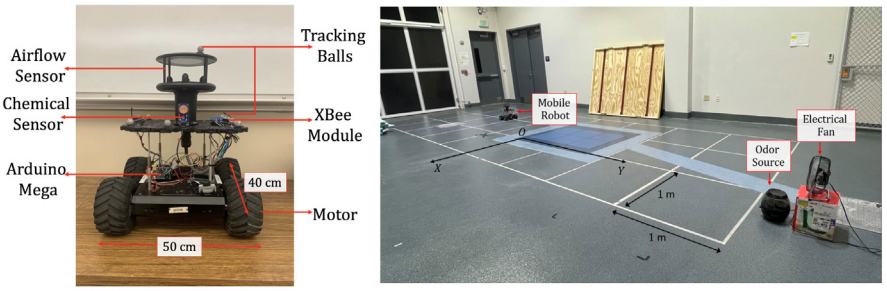

Fig. 13. (a) The mobile robot used in this work. This robot is equipped with an airflow sensor for measuring wind speeds and directions; a chemical sensor for detecting odor plumes; Xbee modules for wireless communication; an onboard computer for processing sensor data. (b) The experiment setup.

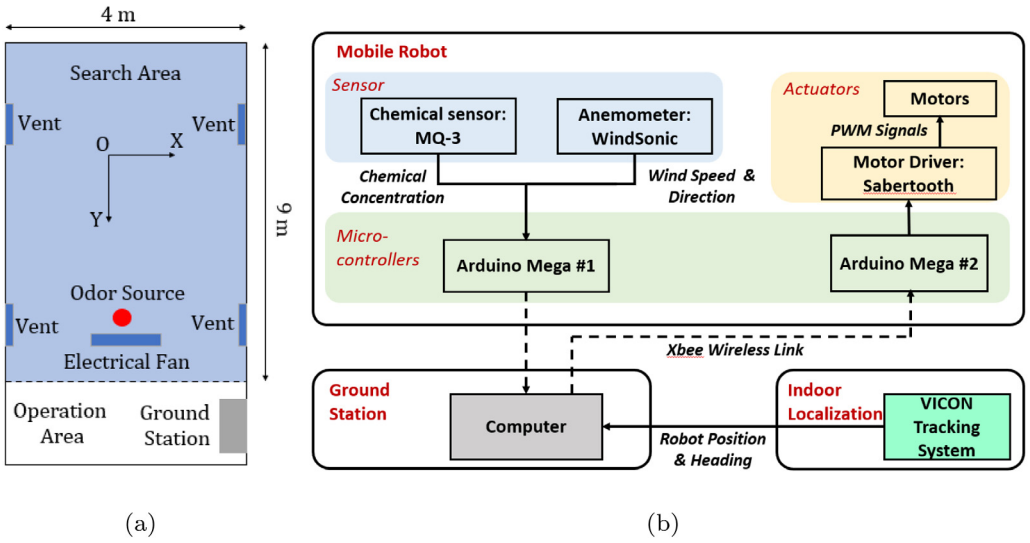

Fig. 14. (a) Search area (b) System configuration. This system contains three main components, including mobile robot, ground station, and indoor localization system. The solid connection line represents physical cables, and the dotted connection line represents wireless link.

onboard micro-controller controls robot motors via a motor driver (Sabertooth, Dimension Engineering). Two micro-controllers can communicate with the ground station via a wireless communication network (Xbee, Digi international). The Vicon tracking system (Vicon Inc.) is employed to determine robot positions and orientations, which sends to the ground station via an Ethernet cable. The response time of all sensors were set to 0.25 s.

The robot starts an OSL test in downwind areas. During the plume tracing process, the robot sends sensor measurements to the ground station. Since the chemical sensor has a long recovery time, an adaptive concentration threshold [36] is employed to determine the odor detection and non-detection events. The navigation algorithms are implemented in the ground station to calculate robot's heading commands (the robot moves in a constant speed), which will be transmitted to the mobile robot via the wireless communication network. Then, the robot moves toward the target heading and collects information at the new location. These processes repeat until the robot reaches the source location, i.e., the distance between the odor source and robot is less than 0.5 m.

7.2. Experiment results and analysis

Hundreds of OSL tests were conducted with three navigation methods, including the traditional bio-inspired method [23], the traditional engineering-based method [12], and the proposed method. In these tests, the robot was placed at different initial positions to find the odor source. To compare search performance, Table 6 lists search time of three navigation methods, collected from tests with the same initial positions $(-1.4, -0.2)$ m. The robot travels at a constant speed 0.15 m/s.

It can be observed that all navigation methods successfully found the odor source in repeated tests. Specifically, the averaged search time of the proposed method is shorter than the bio-inspired method (42.9 s v.s. 46.4 s), and the standard deviation of the proposed method is also lower than the bio-inspired counterpart (3.8 v.s. 6.0). This result indicates that the proposed method improves the search performance compared to the traditional bio-inspired method in terms of search time. It also shows the effectiveness of the implemented fuzzy controller, which dynamically adjusts behavior parameters to improve search efficiency. Besides, the proposed method outperforms the engineering-based method by 28% in terms of the averaged search time. As for the standard deviation, the proposed method is also lower than the engineering-based counterpart (3.8 v.s. 5.9). We found similar results from OSL tests starting from other initial positions, showing that the proposed method outperforms traditional methods in search time.

We present search trajectories in Test 1 as an example to demonstrate the plume tracing process of different navigation methods. As shown in Fig. 15, all navigation methods can correctly find the odor source, i.e., the robot reaches the odor source location. Compared to traditional methods, the proposed method achieves the shortest search time (45 s) and travel distance (5.9

Table 6
Search time of three navigation methods in repeated tests. μ: mean search time; σ: standard deviation of search time; f: success rate.

	Bio-inspired method [23] (s)	Engineering-based method [12] (s)	The proposed method (s)
Test 1	54	63	45
Test 2	56	68	42
Test 3	45	64	41
Test 4	40	63	40
Test 5	48	52	41
Test 6	42	65	41
Test 7	40	55	40
Test 8	44	52	46
Test 9	42	59	52
Test 10	53	54	41
μ (s)	46.4	59.5	**42.9**
σ	6.0	5.9	**3.8**
f	100%	100%	**100%**

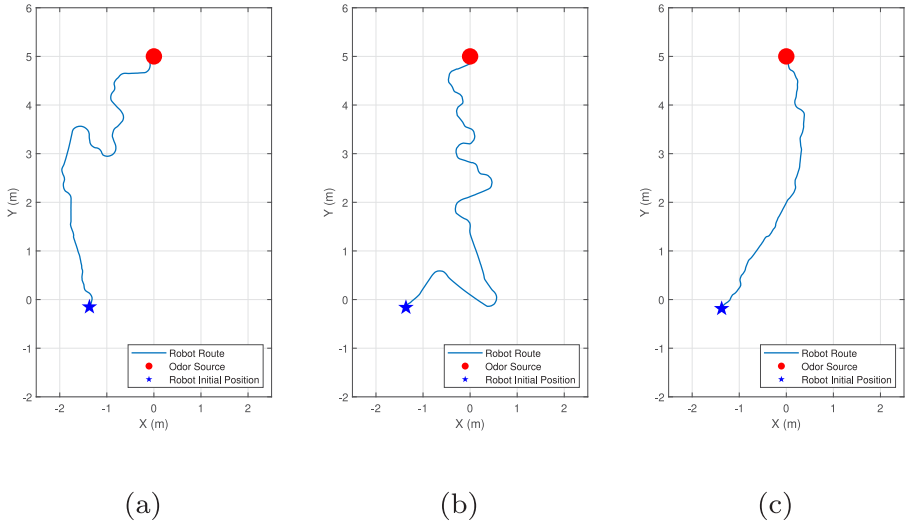

Fig. 15. Sample runs of three algorithms in Group 1 tests. The blue star indicates the robot initial position $(-1.4, -0.2)$ m, and red dot is the odor source location $(0, 5)$ m. (a) Traditional bio-inspired method. Search time: 54 s; travel distance: 7.9 m. (b) Traditional engineering-based method. Search time: 63 s; travel distance: 9.6 m. (c) The proposed method. Search time: 45 s; travel distance: 5.9 m.

m). Besides, the search trajectory of the proposed method is more smooth compared to others. Although the robot was not initially pointing directly to the odor source, the proposed method can adjust robot headings during the plume tracing process and find the odor source. For the traditional bio-inspired and engineering-based methods, search times are 54 s and 63 s, respectively, and travel distances are 7.9 m and 9.6 m, respectively.

8. Conclusion

In this article, an olfactory-based navigation algorithm for using on a mobile robot to find an odor source in an unknown environment is presented. Inspired by the mate-seeking behaviors of male moths, a behavior-based search framework is designed and constructed. To enable the robot perceive the environment, a fuzzy controller is designed and implemented to adapt parameters in search behaviors. As a result, when the airflow environment is turbulent, the robot trajectories are extended to allow the robot to find plumes over a large area, i.e., exploration. When the robot is near the odor source, search trajectories are limited to constrain the robot in a local search, i.e., exploitation. Experiment results show that the proposed navigation algorithm is valid in both laminar and turbulent flow environments. Compared to two traditional bio-inspired and engineering-based methods, the proposed algorithm is more effective and efficient in terms of the averaged search time and success rates.

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.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.robot.2021.103914.

References

[1] U. Montanaro, S. Dixit, S. Fallah, M. Dianati, A. Stevens, D. Oxtoby, A. Mouzakitis, Towards connected autonomous driving: review of use-cases, Veh. Syst. Dyn. 57 (6) (2019) 779–814, http://dx.doi.org/10.1080/00423114.2018.1492142.

[2] G. Kowadlo, R.A. Russell, Robot odor localization: a taxonomy and survey, Int. J. Robot. Res. 27 (8) (2008) 869–894, http://dx.doi.org/10.1177/0278364908095118.

[3] M. Dunbabin, L. Marques, Robots for environmental monitoring: Significant advancements and applications, IEEE Robot. Autom. Mag. 19 (1) (2012) 24–39, http://dx.doi.org/10.1109/MRA.2011.2181683.

[4] S. Soldan, G. Bonow, A. Kroll, Robogasinspector-a mobile robotic system for remote leak sensing and localization in large industrial environments: Overview and first results, IFAC Proc. Vol. 45 (8) (2012) 33–38, http://dx.doi.org/10.3182/20120531-2-NO-4020.00005.

[5] G. Ferri, M.V. Jakuba, D.R. Yoerger, A novel method for hydrothermal vents prospecting using an autonomous underwater robot, in: 2008 IEEE International Conference on Robotics and Automation, IEEE, 2008, pp. 1055–1060, http://dx.doi.org/10.1109/ROBOT.2008.4543344.

[6] J.A. Farrell, J. Murlis, X. Long, W. Li, R.T. Cardé, Filament-based atmospheric dispersion model to achieve short time-scale structure of odor plumes, Environ. Fluid Mech. 2 (1–2) (2002) 143–169, http://dx.doi.org/10.1023/A:1016283702837.

[7] H. Ishida, K.-i. Suetsugu, T. Nakamoto, T. Moriizumi, Study of autonomous mobile sensing system for localization of odor source using gas sensors and anemometric sensors, Sensors Actuators A 45 (2) (1994) 153–157, http://dx.doi.org/10.1016/0924-4247(94)00829-9.

[8] X.-x. Chen, J. Huang, Odor source localization algorithms on mobile robots: A review and future outlook, Robot. Auton. Syst. 112 (2019) 123–136, http://dx.doi.org/10.1016/j.robot.2018.11.014.

[9] R.T. Cardé, A. Mafra-Neto, Mechanisms of flight of male moths to pheromone, in: Insect Pheromone Research, Springer, 1997, pp. 275–290, http://dx.doi.org/10.1007/978-1-4615-6371-6_25.

[10] L.L. López, V. Vouloutsi, A.E. Chimeno, E. Marcos, S.B.d. i Badia, Z. Mathews, P.F. Verschure, A. Ziyatdinov, A.P. i Lluna, Moth-like chemo-source localization and classification on an indoor autonomous robot, in: On Biomimetics, IntechOpen, 2011, http://dx.doi.org/10.5772/19695.

[11] S. Pang, F. Zhu, Reactive planning for olfactory-based mobile robots, in: 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2009, pp. 4375–4380, http://dx.doi.org/10.1109/IROS.2009.5353993.

[12] L. Wang, S. Pang, J. Li, Olfactory-based navigation via model-based reinforcement learning and fuzzy inference methods, IEEE Trans. Fuzzy Syst. (2020) http://dx.doi.org/10.1109/TFUZZ.2020.3011741.

[13] J.A. Farrell, S. Pang, W. Li, Plume mapping via hidden Markov methods, IEEE Trans. Syst. Man Cybern. B 33 (6) (2003) 850–863, http://dx.doi.org/10.1109/TSMCB.2003.810873.

[14] G. Sandini, G. Lucarini, M. Varoli, Gradient driven self-organizing systems, in: Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vol. 1, IROS'93, IEEE, 1993, pp. 429–432, http://dx.doi.org/10.1109/IROS.1993.583132.

[15] F.W. Grasso, T.R. Consi, D.C. Mountain, J. Atema, Biomimetic robot lobster performs chemo-orientation in turbulence using a pair of spatially separated sensors: Progress and challenges, Robot. Auton. Syst. 30 (1–2) (2000) 115–131, http://dx.doi.org/10.1016/S0921-8890(99)00068-8.

[16] R.A. Russell, A. Bab-Hadiashar, R.L. Shepherd, G.G. Wallace, A comparison of reactive robot chemotaxis algorithms, Robot. Auton. Syst. 45 (2) (2003) 83–97, http://dx.doi.org/10.1016/S0921-8890(03)00120-9.

[17] A. Lilienthal, T. Duckett, Experimental analysis of gas-sensitive Braitenberg vehicles, Adv. Robot. 18 (8) (2004) 817–834, http://dx.doi.org/10.1163/1568553041738103.

[18] H. Ishida, G. Nakayama, T. Nakamoto, T. Moriizumi, Controlling a gas/odor plume-tracking robot based on transient responses of gas sensors, IEEE Sens. J. 5 (3) (2005) 537–545, http://dx.doi.org/10.1109/ICSENS.2002.1037374.

[19] J. Murlis, C. Jones, Fine-scale structure of odour plumes in relation to insect orientation to distant pheromone and other attractant sources, Physiol. Entomol. 6 (1) (1981) 71–86, http://dx.doi.org/10.1111/j.1365-3032.1981.tb00262.x.

[20] 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: 2008 IEEE International Conference on Robotics and Automation, IEEE, 2008, pp. 1138–1143, http://dx.doi.org/10.1109/ROBOT.2008.4543357.

[21] W. Li, J.A. Farrell, S. Pang, R.M. Arrieta, Moth-inspired chemical plume tracing on an autonomous underwater vehicle, IEEE Trans. Robot. 22 (2) (2006) 292–307, http://dx.doi.org/10.1109/TRO.2006.870627.

[22] B. Luo, Q.-H. Meng, J.-Y. Wang, M. Zeng, A flying odor compass to autonomously locate the gas source, IEEE Trans. Instrum. Meas. 67 (1) (2017) 137–149, http://dx.doi.org/10.1109/TIM.2017.2759378.

[23] J.A. Farrell, S. Pang, W. Li, Chemical plume tracing via an autonomous underwater vehicle, IEEE J. Ocean. Eng. 30 (2) (2005) 428–442, http://dx.doi.org/10.1109/JOE.2004.838066.

[24] S. Shigaki, T. Sakurai, N. Ando, D. Kurabayashi, R. Kanzaki, Time-varying moth-inspired algorithm for chemical plume tracing in turbulent environment, IEEE Robot. Autom. Lett. 3 (1) (2017) 76–83, http://dx.doi.org/10.1109/LRA.2017.2730361.

[25] S. Shigaki, Y. Shiota, D. Kurabayashi, R. Kanzaki, Modeling of the adaptive chemical plume tracing algorithm of an insect using fuzzy inference, IEEE Trans. Fuzzy Syst. 28 (1) (2019) 72–84, http://dx.doi.org/10.1109/tfuzz.2019.2915187.

[26] G. Ferri, E. Caselli, V. Mattoli, A. Mondini, B. Mazzolai, P. Dario, SPIRAL: A novel biologically-inspired algorithm for gas/odor source localization in an indoor environment with no strong airflow, Robot. Auton. Syst. 57 (4) (2009) 393–402, http://dx.doi.org/10.1016/j.robot.2008.07.004.

[27] F. Rahbar, A. Marjovi, P. Kibleur, A. Martinoli, A 3-D bio-inspired odor source localization and its validation in realistic environmental conditions, in: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2017, pp. 3983–3989, http://dx.doi.org/10.1109/IROS.2017.8206252.

[28] S. Shigaki, S. Haigo, C.H. Reyes, T. Sakurai, R. Kanzaki, D. Kurabayashi, H. Sezutsu, Analysis of the role of wind information for efficient chemical plume tracing based on optogenetic silkworm moth behavior, Bioinspiration Biomim. 14 (4) (2019) 046006, http://dx.doi.org/10.1088/1748-3190/ab1d34.

[29] A. Liberzon, K. Harrington, N. Daniel, R. Gurka, A. Harari, G. Zilman, Moth-inspired navigation algorithm in a turbulent odor plume from a pulsating source, PLoS One 13 (6) (2018) e0198422, http://dx.doi.org/10.1371/journal.pone.0198422.

[30] F.W. Grasso, J.A. Basil, J. Atema, Toward the convergence: robot and lobster perspectives of tracking odors to their source in the turbulent marine environment, in: Proceedings of the 1998 IEEE International Symposium on Intelligent Control (ISIC) Held Jointly with IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA) Intell, IEEE, 1998, pp. 259–264, http://dx.doi.org/10.1109/ISIC.1998.713671.

[31] B.T. Michaelis, K.W. Leathers, Y.V. Bobkov, B.W. Ache, J.C. Principe, R. Baharloo, I.M. Park, M.A. Reidenbach, Odor tracking in aquatic organisms: the importance of temporal and spatial intermittency of the turbulent plume, Sci. Rep. 10 (1) (2020) 1–11, http://dx.doi.org/10.1038/s41598-020-64766-y.

[32] K.W. Leathers, B.T. Michaelis, M.A. Reidenbach, Interpreting the spatial-temporal structure of turbulent chemical plumes utilized in odor tracking by lobsters, Fluids 5 (2) (2020) 82, http://dx.doi.org/10.3390/fluids5020082.

[33] J. Macedo, L. Marques, E. Costa, A comparative study of bio-inspired odour source localisation strategies from the state-action perspective, Sensors 19 (10) (2019) 2231, http://dx.doi.org/10.3390/s19102231.

[34] V. Hernandez Bennetts, A.J. Lilienthal, P. Neumann, M. Trincavelli, Mobile robots for localizing gas emission sources on landfill sites: is bio-inspiration the way to go? Frontiers in Neuroengineering 4 (2012) 20, http://dx.doi.org/10.3389/fneng.2011.00020.

[35] S. Pang, J.A. Farrell, Chemical plume source localization, IEEE Trans. Syst. Man Cybern. B 36 (5) (2006) 1068–1080, http://dx.doi.org/10.1109/TSMCB.2006.874689.

[36] J.-G. Li, Q.-H. Meng, Y. Wang, M. Zeng, Odor source localization using a mobile robot in outdoor airflow environments with a particle filter algorithm, Auton. Robots 30 (3) (2011) 281–292, http://dx.doi.org/10.1007/s10514-011-9219-2.

[37] M.V. Jakuba, Stochastic Mapping for Chemical Plume Source Localization with Application to Autonomous Hydrothermal Vent Discovery (Ph.D. thesis), Massachusetts Institute of Technology, 2007, http://dx.doi.org/10.1575/1912/1583.

[38] F. Rahbar, A. Marjovi, A. Martinoli, An algorithm for odor source localization based on source term estimation, in: 2019 International Conference on Robotics and Automation, ICRA, IEEE, 2019, pp. 973–979, http://dx.doi.org/10.1109/ICRA.2019.8793784.

[39] M. Hutchinson, C. Liu, W.-H. Chen, Information-based search for an atmospheric release using a mobile robot: Algorithm and experiments, IEEE Trans. Control Syst. Technol. 27 (6) (2018) 2388–2402, http://dx.doi.org/10.1109/TCST.2018.2860548.

[40] H. Jiu, Y. Chen, W. Deng, S. Pang, Underwater chemical plume tracing based on partially observable Markov decision process, Int. J. Adv. Robot. Syst. 16 (2) (2019) 1729881419831874, http://dx.doi.org/10.1177/1729881419831874.

[41] L. Wang, S. Pang, Chemical plume tracing using an AUV based on POMDP source mapping and A-star path planning, in: OCEANS 2019 MTS/IEEE SEATTLE, IEEE, 2019, pp. 1–7, http://dx.doi.org/10.23919/OCEANS40490.2019.8962795.

[42] M. Vergassola, E. Villermaux, B.I. Shraiman, 'Infotaxis' as a strategy for searching without gradients, Nature 445 (7126) (2007) 406, http://dx.doi.org/10.1038/nature05464.

[43] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95-International Conference on Neural Networks, Vol. 4, IEEE, 1995, pp. 1942–1948, http://dx.doi.org/10.1109/ICNN.1995.488968.

[44] L. Marques, U. Nunes, A.T. de Almeida, Particle swarm-based olfactory guided search, Auton. Robots 20 (3) (2006) 277–287, http://dx.doi.org/10.1007/s10514-006-7567-0.

[45] Z. Fu, Y. Chen, Y. Ding, D. He, Pollution source localization based on multi-UAV cooperative communication, IEEE Access 7 (2019) 29304–29312, http://dx.doi.org/10.1109/ACCESS.2019.2900475.

[46] Q.-H. Meng, W.-X. Yang, Y. Wang, M. Zeng, Collective odor source estimation and search in time-variant airflow environments using mobile robots, Sensors 11 (11) (2011) 10415–10443, http://dx.doi.org/10.3390/s111110415.

[47] Q. Lu, P. Luo, A learning particle swarm optimization algorithm for odor source localization, Int. J. Autom. Comput. 8 (3) (2011) 371–380, http://dx.doi.org/10.1007/s11633-011-0594-0.

[48] X. Huang, Improved 'infotaxis' algorithm-based cooperative multi-USV pollution source search approach in lake water environment, Symmetry 12 (4) (2020) 549, http://dx.doi.org/10.3390/sym12040549.
[49] P. Jiang, Y. Wang, A. Ge, Multivariable fuzzy control based mobile robot odor source localization via semitensor product, Math. Probl. Eng. 2015 (2015) http://dx.doi.org/10.1155/2015/736720.
[50] X. Chen, J. Huang, Towards environmentally adaptive odor source localization: Fuzzy Lévy taxis algorithm and its validation in dynamic odor plumes, in: 2020 5th International Conference on Advanced Robotics and Mechatronics, ICARM, IEEE, 2020, pp. 282–287, http://dx.doi.org/10.1109/ICARM49381.2020.9195363.
[51] L. Wang, S. Pang, An implementation of the adaptive neuro-fuzzy inference system (ANFIS) for odor source localization, in: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2020, pp. 4551–4558, http://dx.doi.org/10.1109/IROS45743.2020.9341688.
[52] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, A. Graves, M. Riedmiller, A.K. Fidjeland, G. Ostrovski, et al., Human-level control through deep reinforcement learning, Nature 518 (7540) (2015) 529–533, http://dx.doi.org/10.1038/nature14236.
[53] H. Li, Q. Zhang, D. Zhao, Deep reinforcement learning-based automatic exploration for navigation in unknown environment, IEEE Trans. Neural Netw. Learn. Syst. 31 (6) (2019) 2064–2076, http://dx.doi.org/10.1109/TNNLS.2019.2927869.
[54] B.R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A.A. Al Sallab, S. Yogamani, P. Pérez, Deep reinforcement learning for autonomous driving: A survey, IEEE Trans. Intell. Transp. Syst. (2021) http://dx.doi.org/10.1109/TITS.2021.3054625.
[55] H. Hu, S. Song, C.P. Chen, Plume tracing via model-free reinforcement learning method, IEEE Trans. Neural Netw. Learn. Syst. (2019) http://dx.doi.org/10.1109/TNNLS.2018.2885374.
[56] X. Chen, C. Fu, J. Huang, A Deep Q-Network for robotic odor/gas source localization: Modeling, measurement and comparative study, Measurement 183 (2021) 109725, http://dx.doi.org/10.1016/j.measurement.2021.109725.
[57] T. Wiedemann, C. Vlaicu, J. Josifovski, A. Viseras, Robotic information gathering with reinforcement learning assisted by domain knowledge: An application to gas source localization, IEEE Access 9 (2021) 13159–13172, http://dx.doi.org/10.1109/ACCESS.2021.3052024.
[58] W. Naeem, R. Sutton, J. Chudley, Chemical plume tracing and odour source localisation by autonomous vehicles, J. Navig. 60 (2) (2007) 173–190, http://dx.doi.org/10.1017/S0373463307004183.
[59] M. Sabelis, P. Schippers, Variable wind directions and anemotactic strategies of searching for an odour plume, Oecologia 63 (2) (1984) 225–228, http://dx.doi.org/10.1007/BF00379881.
[60] W. Li, J.A. Farrell, R.T. Card, Tracking of fluid-advected odor plumes: strategies inspired by insect orientation to pheromone, Adapt. Behav. 9 (3–4) (2001) 143–170, http://dx.doi.org/10.1177/10597123010093003.
[61] L. Kuenen, R.T. Carde, Strategies for recontacting a lost pheromone plume: casting and upwind flight in the male gypsy moth, Physiol. Entomol. 19 (1) (1994) 15–29, http://dx.doi.org/10.1111/j.1365-3032.1994.tb01069.x.
[62] J. Murlis, J.S. Elkinton, R.T. Carde, Odor plumes and how insects use them, Ann. Rev. Entomol. 37 (1) (1992) 505–532, http://dx.doi.org/10.1146/annurev.en.37.010192.002445.
[63] G.C. Sousa, B.K. Bose, A fuzzy set theory based control of a phase-controlled converter DC machine drive, IEEE Trans. Ind. Appl. 30 (1) (1994) 34–44, http://dx.doi.org/10.1109/28.273619.
[64] M. Pratama, J. Lu, G. Zhang, Evolving type-2 fuzzy classifier, IEEE Trans. Fuzzy Syst. 24 (3) (2015) 574–589, http://dx.doi.org/10.1109/TFUZZ.2015.2463732.
[65] N. Carpman, Turbulence intensity in complex environments and its influence on small wind turbines, 2011.
[66] F. Russo, N.T. Basse, Scaling of turbulence intensity for low-speed flow in smooth pipes, Flow Meas. Instrum. 52 (2016) 101–114, http://dx.doi.org/10.1016/j.flowmeasinst.2016.09.012.
[67] J.P. Crimaldi, M.B. Wiley, J.R. Koseff, The relationship between mean and instantaneous structure in turbulent passive scalar plumes, J. Turbul. 3 (14) (2002) 1–24, http://dx.doi.org/10.1088/1468-5248/3/1/014.
[68] J. Elkinton, R. Cardé, C. Mason, Evaluation of time-average dispersion models for estimating pheromone concentration in a deciduous forest, J. Chem. Ecol. 10 (7) (1984) 1081–1108, http://dx.doi.org/10.1007/BF00987515.
[69] B.K. Bose, Expert system, fuzzy logic, and neural network applications in power electronics and motion control, Proc. IEEE 82 (8) (1994) 1303–1323, http://dx.doi.org/10.1109/5.301690.
[70] J.A. Farrell, S. Pang, W. Li, 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), Vol. 2, IEEE, 2003, pp. 962–968, http://dx.doi.org/10.1109/OCEANS.2003.178458.
[71] W. Jatmiko, K. Sekiyama, T. Fukuda, A pso-based mobile robot for odor source localization in dynamic advection-diffusion with obstacles environment: theory, simulation and measurement, IEEE Comput. Intell. Maga. 2 (2) (2007) 37–51, http://dx.doi.org/10.1109/MCI.2007.353419.
[72] Q. Lu, Q.-L. Han, X. Xie, S. Liu, A finite-time motion control strategy for odor source localization, IEEE Trans. Ind. Electron. 61 (10) (2014) 5419–5430, http://dx.doi.org/10.1109/TIE.2014.2301751.
[73] Y. Tian, A. Zhang, Simulation environment and guidance system for AUV tracing chemical plume in 3-dimensions, in: 2010 2nd International Asia Conference on Informatics in Control, Automation and Robotics, Vol. 1, CAR 2010, IEEE, 2010, pp. 407–411, http://dx.doi.org/10.1109/CAR.2010.5456812.
[74] Q. Lu, Q.-L. Han, S. Liu, A cooperative control framework for a collective decision on movement behaviors of particles, IEEE Trans. Evol. Comput. 20 (6) (2016) 859–873, http://dx.doi.org/10.1109/TEVC.2016.2526656.
[75] J.-y. Zhou, J.-g. Li, S.-g. Cui, A bionic plume tracing method with a mobile robot in outdoor time-varying airflow environment, in: 2015 IEEE International Conference on Information and Automation, IEEE, 2015, pp. 2351–2355, http://dx.doi.org/10.1109/ICInfA.2015.7279679.
[76] Q. Feng, H. Cai, Z. Chen, Y. Yang, J. Lu, F. Li, J. Xu, X. Li, Experimental study on a comprehensive particle swarm optimization method for locating contaminant sources in dynamic indoor environments with mechanical ventilation, Energy Build. 196 (2019) 145–156, http://dx.doi.org/10.1016/j.enbuild.2019.03.032.

Lingxiao Wang received the B.S. degree in electrical engineering from Civil Aviation University of China, Tianjin, China, in 2015, and the M.S. degree in electrical and computer engineering from Embry-Riddle Aeronautical University, Daytona Beach, FL, in 2017.

He is a Ph.D. candidate with the Department of Electrical Engineering and Computer Science at Embry-Riddle Aeronautical University, and his research interests include autonomous systems, olfactory-based navigation methods, and artificial intelligence.

Shuo Pang received his B.S. degree in electrical engineering from Harbin Engineering University, China, in 1997, and the M.S. and Ph.D. degrees in electrical engineering from University of California, Riverside, in 2001 and 2004, respectively. Currently, he is an Associate Professor in the Department of Electrical Engineering and Computer Science, Embry-Riddle Aeronautical University, Daytona Beach, Florida.

His research spans theoretical-algorithm development and application-driven intelligent systems. His current research interests include embedded systems, robotics, and artificial intelligence techniques for autonomous vehicles, i.e., autonomous vehicle chemical plume tracing and autonomous vehicle online mapping and planning.