Olfactory-Based Navigation via Model-Based Reinforcement Learning and Fuzzy Inference Methods

Lingxiao Wang, *Student Member, IEEE*, Shuo Pang, *Member, IEEE*, and Jinlong Li

***Abstract*—This article presents an olfactory-based navigation algorithm for using a mobile robot to locate an odor source in a turbulent flow environment. We analogize the odor source localization as a reinforcement learning problem. During the odor plume tracing process, the belief state in a partially observable Markov decision process model is adapted to generate a source probability map that estimates possible odor source locations, and a hidden Markov model is employed to produce a plume distribution map that premises plume propagation areas. Both source and plume estimations are fed to the robot, and a decision-making approach based on fuzzy inference is designed to dynamically fuse information from two maps and to balance the exploitation and exploration of the search. After assigning the fused information to reward functions, a value iteration based path planning algorithm is presented to solve for the optimal action policy. Comparing to other commonly used olfactory-based navigation algorithms, such as moth-inspired and Bayesian inference methods, simulation results show that the proposed method is more intelligent and efficient.**

***Index Terms*—Fuzzy theory, odor source localization (OSL), olfactory-based navigation, partially observable Markov decision process (POMDP).**

I. Introduction

OLFACTION, also known as the sense of smell, is an important sensing ability for animals to perform life-essential activities, such as homing, foraging, mate-seeking, and evading predators. Inspired by olfactory capabilities of animals, an autonomous vehicle or a mobile robot, equipped with odor-detection sensors (e.g., chemical sensors), could locate an odor (or volatile chemical) source in an unknown environment. The technology of employing a robot to find an odor source is referred to as odor source localization (OSL) [1]. Some practical OSL applications that are frequently quoted include monitoring air pollution [2], locating chemical gas leaks [3], locating unexploded mines and bombs [4], and detecting biological phenomena such as underwater hydrothermal vents [5].

Manuscript received February 8, 2020; revised May 29, 2020 and July 10, 2020; accepted July 17, 2020. Date of publication July 24, 2020; date of current version October 6, 2021. *(Corresponding author: Shuo Pang.)*

Lingxiao Wang and Shuo Pang are with the Department of Electrical Engineering and Computer Science, Embry-Riddle Aeronautical University, Daytona Beach, FL 32114 USA (e-mail: lingxiaw@my.erau.edu; shuo.pang@erau.edu).

Jinlong Li is with the Shanghai Jiaotong University, Shanghai 201101, China (e-mail: leoljl@126.com).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TFUZZ.2020.3011741

An effective navigation method is crucial for an OSL problem. Like image-based navigation methods, which use the information extracted from images as the reference to locate and navigate a robot, olfactory-based navigation methods detect odor plumes as cues to guide a robot toward an odor source. The challenging part of this navigation problem is to estimate plume locations, which are not only related to the molecular diffusion that takes plumes away from the odor source but also the advection of airflow [6].

The most straightforward olfactory-based navigation approach is the chemotaxis [7], which commands the robot to move along the gradient of odor plume concentrations. A common implementation of this method is to install a pair of chemical sensors on the left and right sides of a robot, and the robot is commanded to steer toward the side with the higher concentration [8]. Many experiments [9]–[12] have proved that the chemotaxis method is effective when the odor source is placed in an environment with laminar (i.e., low Reynolds numbers) flows. 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 is a patchy and intermittent signal [13].

Alternatively, two other categories of olfactory-based navigation strategies have been proposed, namely bio-inspired methods and engineering-based methods. A bio-inspired method directs the robot to mimic animal behaviors, such as mate-seeking behaviors of male moths, which could successfully locate a female moth by tracking pheromones over a long distance [14]. To complete this task, a male moth follows a "surge/casting" behavior pattern: it will fly upwind (surge) when detects pheromones and traverse the wind (casting) when pheromones are absent. Ryohei *et al.* [15] generalized the "surge/casting" model and implemented it on a ground wheeled vehicle to find an odor source in a laminar flow environment. Li *et al.* [16] implemented this method on an autonomous underwater vehicle to search for an underwater chemical source. Experiment results [17] proved the validity of this method.

By contrast, an engineering-based method does not follow a fixed behavior pattern. It utilizes math and physics approaches to model odor plume distribution and estimates possible odor source locations. Then, a path planner is employed to guide the robot moving toward the estimated source location. Methods that produce a source probability map, i.e., a map that indicates the

possibility of an area containing an odor source, are various, such as Bayesian inference [18], particle filter [19], hidden Markov model (HMM) [20], and partially observable Markov decision process (POMDP) [21]. In the planning procedure, artificial potential field [22], and deterministic policy gradient [23] are feasible algorithms to design path planners, which generate a search path that guides the robot to the estimated target. Besides, Vergassola *et al.* [24] presented the "infotaxis" method, which uses information entropy to guide the robot searching for an odor source. In this method, the robot was commanded to select the movement that reduces the information uncertainty of the odor source.

Comparing existing olfactory-based navigation strategies, the limitation of bio-inspired methods is that the robot lacks the capability of estimating odor plume locations. Thus, when odor plumes are not detected, the robot can only perform a time-consuming "casting" behavior to recover plumes. As for engineering-based methods, if the robot is source seeking oriented (e.g. [25]), the search efficiency is not ideal, since the source probability map is unreliable before the robot acquires enough odor source information. On the other hand, the search result is also not desired if the robot is plume seeking oriented (e.g., [23]), since it leans toward detecting plumes instead of locating the odor source. The research niche that our approach fits is to let the robot estimate both odor source and odor plume locations and fuse two estimations as the target to guide the robot. So that, not only does the robot search for the odor source location, but also it can quickly recover from plume nondetection events when it does not observe plumes.

Reinforcement learning (RL) algorithms are widely implemented in the field of artificial intelligence (AI). For instance, AlphaGo [26], an AI robot based on RL methods, defeated a couple of best professional human players in the game of Go. An RL algorithm models interactions between an agent and the environment: an agent receives rewards by performing actions, and the goal of the agent is to take the action that maximizes the cumulative reward [27]. The framework of an RL algorithm is similar to an OSL problem: an agent could be considered as a robot that aims to find an odor source in an unknown environment. By appropriately defining reward functions, the robot is driven to choose actions that are beneficial to locate an odor source. The optimal policy is adapted as a search path that leads the robot to the maximal reward location.

In this article, an olfactory-based navigation method for using on a ground mobile robot is presented. The proposed method contains two main procedures, i.e., modeling and planning. In the modeling procedure, odor source and plume estimations are obtained. Specifically, belief states in a POMDP model are adapted to represent a source probability map, from which the robot estimates possible odor source locations. Besides, a plume distribution map that predicts odor plume propagation areas is obtained from an HMM-based method. A fuzzy inference approach is designed to dynamically fuse the information from two maps, and the combined information is assigned to reward functions. In the planning procedure, a search route is determined based on the generated reward functions. The value iteration method is adopted to solve the RL problem and produces the optimal policy, which is a search route that leads the robot to the maximum reward location, i.e., the location that contains the most odor source information. Modeling and planning procedures are repeated until reward functions converge, which is considered as the complete of an OSL problem.

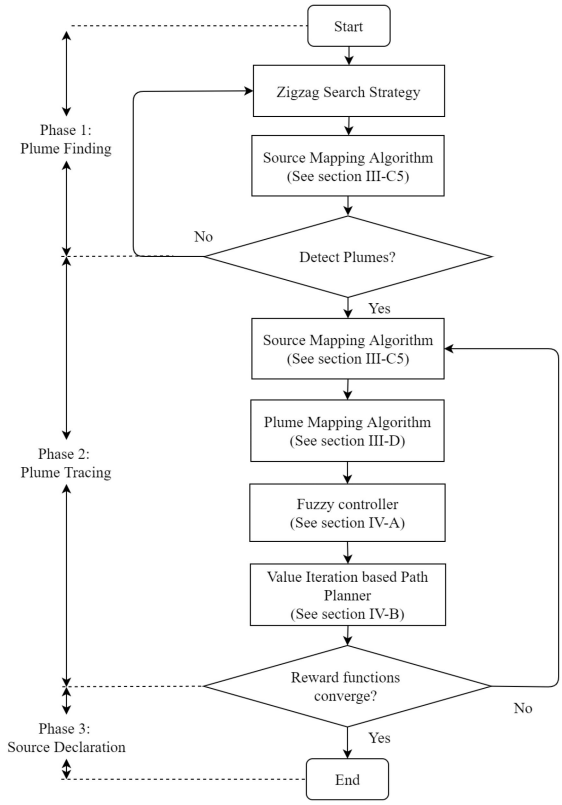


Fig. 1. Framework of the proposed olfactory-based navigation method.

II. Overview of the Proposed Olfactory-Based Navigation Method

It is commonly accepted that an OSL problem can be divided into three phases, namely plume finding, plume tracing, and source declaration [28]. In the first phase, the robot searches the presence of odor plumes. After the robot detects plumes, it switches to the plume tracing phase, which follows plumes as cues to find the odor source. In the source declaration phase, the robot recognizes the odor source and declares the odor source location.

The scheme of the proposed olfactory-based navigation method is presented in Fig. 1. In the plume finding phase, a "zigzag" search route presented in [16] is adopted to detect plumes for the first time, and the source mapping algorithm that estimates odor source locations (see Section III-C5) is activated simultaneously. After the robot detects plumes for the first time, the robot switches to the plume tracing phase, in which the source mapping algorithm sustains and the plume mapping

algorithm that predicts plume propagation areas (see Section III-D) is activated. Results from source and plume mappings are combined via a fuzzy controller to form reward functions (see Section IV-A). Then, a search route is generated by the value iteration based path planning algorithm (see Section IV-B). In the plume tracing phase, aforementioned algorithms keep updating until reward functions converge, which indicates that the robot finds the odor source location.

Specifically, the proposed method can be separated into two principle procedures, namely modeling and planning. In the modeling procedure, source estimates are obtained from belief states in a POMDP. The motivation of using belief states is that under the POMDP framework, belief states can estimate uncertainties of an environment in a stochastic fashion, i.e., the agent cannot directly observe states, but it can estimate the current state through the belief state, which is the probability of the agent being in a state. To adapt the POMDP framework in an OSL problem, the odor source location can be used to define hidden states, since it is unknown to the robot, actions can be considered as possible moving directions of the robot, observations can be adapted as plume detection and nondetection events, and the belief state can be interpreted as the probability of an area containing the odor source, i.e., a source probability map. Besides, to construct a plume distribution map, which estimates plume propagation areas, an HMM-based method is adopted.

In the planning procedure, robot searching routes are determined. In an RL algorithm, reward functions determine behaviors of an agent and stipulate how we want the agent to accomplish its objective. In the proposed method, reward functions are expected to contain the information that not only conjectures odor source locations but also estimates plume propagation areas since source and plume estimations are instructive for the robot to either exploit or explore the odor source location. Thus, a source probability map and a plume distribution map are combined to form reward functions. Instead of combining them in a fixed proportional pattern (e.g., [29]), a fuzzy controller is designed to identify the current search condition and dynamically balance weights of two maps (exploitation or exploration). Then, a value iteration based path planning algorithm is adopted to solve for the optimal policy, i.e., the optimal search route that leads the robot toward the location containing the most odor source information.

III. Modeling

A. Search Area

In this work, the OSL is considered as a 2-D problem, since the aimed implementation platform of the proposed olfactory-based navigation method is a ground mobile robot. For computational feasibility, the search area is modeled as a rectangular grid with m cells in a row and n cells in a column as shown in Fig. 2. The size of a cell is defined as $L_x \times L_y$, where L_x and L_y are the length and width of a cell in the x and y directions, respectively. A vector $\mathbf{C} = [C_1, C_2, ..., C_M]$ is defined to store cell indexes, where $M = mn$. Besides, C_i can also represent the position of a cell, such that $C_i = (x_i, y_i)$ is the center point of a cell C_i,

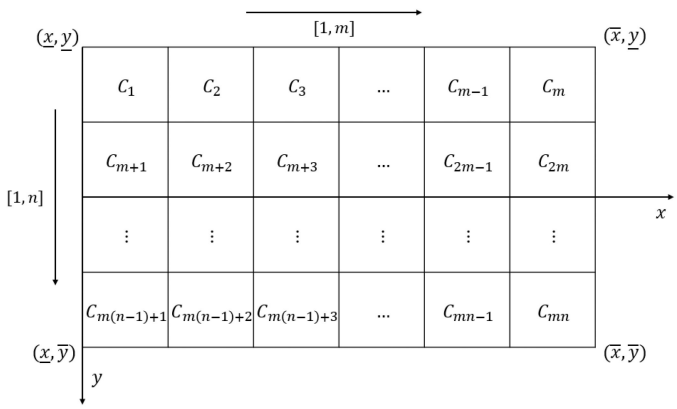


Fig. 2. Search area defined in the OSL problem.

$i \in [1, M]$. The odor source is placed in one of M cells, and its location is clouded to the robot.

Let $f \in [1, m]$ count over cells in the x direction and $g \in [1, n]$ count over cells in the y direction. Then, a cell index $i \in [1, M]$ can be represented as $i = f + (g-1)m$. Reversely, if i is given, f and g can be calculated as follows:

$$\begin{aligned} f(i) &= \text{rem}(i-1, m) + 1 \\ g(i) &= \text{int}(i-1, m) + 1 \end{aligned} \tag{1}$$

where $\text{rem}(n, m)$ and $\text{int}(n, m)$ are the remainder of n being divided by m and the greatest integer that is less than or equal to n/m, respectively. Thus, two cell representations, i.e., C_i and $C_{(f,g)}$, are equivalent.

B. Probabilities of Detecting and Not Detecting Plumes

1) Plume Model: In a turbulent flow environment, movements of odor plumes follow a random walk superimposed on the airflow advection, which can be expressed as [18]

$$\dot{\mathbf{X}}(t) = \mathbf{U}(\mathbf{X}, t) + \mathbf{N}(t) \tag{2}$$

where $\mathbf{X} = (x_p, y_p)$ is the odor plume location, $\mathbf{U} = (u_x, u_y)$ is the mean wind velocity, which transports a plume as the whole body (i.e., advection), and $\mathbf{N} = (n_x, n_y)$ denotes the random walk velocity, which stirs filaments inside a plume and changes the plume shape (i.e., diffusion). $\mathbf{N}$ can be modeled as a Gaussian random process with zero mean and $\boldsymbol{\sigma}^2 = (\sigma_x^2, \sigma_y^2)$ variance, where σ_x and σ_y are strengths of the random walk velocity in the x and y directions, respectively. It should be noted that the position of an odor plume is chiefly determined by the advection $\mathbf{U}$, since the strength of mean wind velocity is much higher than that of the random walk velocity [30].

If the odor source releases a single odor plume at time t_l, the location of the odor plume at time t_k ($t_l < t_k$) can be calculated by integrating (2)

$$\mathbf{X}(t_l, t_k) = \mathbf{X}_s(t_l) + \int_{t_l}^{t_k} \mathbf{U}(\mathbf{X}(\tau), \tau) d\tau + \int_{t_l}^{t_k} \mathbf{N}(\tau) d\tau \tag{3}$$

where $\mathbf{X}_s(t_l)$ is the odor source location. If the odor source continuously releases plumes in the time interval $[t_l, t_k]$, assuming that the release rate is G plumes per second, there are $G(t_k - t_l)$ plumes released, and positions of released plumes can be denoted by $\mathbf{P}(t_l, t_k) = [\mathbf{X}(t_l), \mathbf{X}(t_l + d\tau), \mathbf{X}(t_l + 2d\tau), ..., \mathbf{X}(t_k)]$ where $d\tau = 1/G$.

2) Single Released Odor Plume: This section presents the probability of detecting a plume in an arbitrary cell given that the source only releases a single plume.

Let $\mathbf{s}(t_l, t_k) = (s_x, s_y) = \int_{t_l}^{t_k} \mathbf{U}(\mathbf{X}(\tau), \tau) d\tau$, where s_x and s_y are plume advection distances in the x and y directions, respectively. Let $\mathbf{W}(t_l, t_k) = (w_x, w_y) = \int_{t_l}^{t_k} \mathbf{N}(\tau) d\tau$, which is a Gaussian random process with zero mean and $(t_k - t_l)\boldsymbol{\sigma}^2$ variance. Thus, (3) can be rewritten as follows:

$$\mathbf{X}(t_l, t_k) = \mathbf{X}_s(t_l) + \mathbf{s}(t_l, t_k) + \mathbf{W}(t_l, t_k). \tag{4}$$

It is worth mentioning that $\mathbf{s}(t_l, t_k)$ is approximated by integrating the sensed airflow velocities at the robot position from t_l to t_k since the global airflow information is absent: $\mathbf{s}(t_l, t_k) = (s_x, s_y) \approx \sum_{q=l}^{k-1} \mathbf{u}(\mathbf{X}_v(t_q), t_q) dt$, where $\mathbf{u}(\mathbf{X}_v(t), t)$ denotes airflow measurements at the location $\mathbf{X}_v(t)$ (i.e., the robot location at time t). This approximation will introduce additional errors, but, since the global airflow information is unavailable, this assumption is acceptable.

If the odor plume is propagated to a cell C_j at the time t_k, the estimated odor source location $\hat{\mathbf{X}}_s = (x_s, y_s)$ can be obtained by solving (4)

$$\hat{\mathbf{X}}_s(t_l, t_k) = \mathbf{X}(t_k) - \mathbf{s}(t_l, t_k) - \mathbf{W}(t_l, t_k) \tag{5}$$

where $\mathbf{X}(t_k) = (x_j, y_j)$ is the plume location at time t_k, which is inside the cell C_j. Note that, since $\mathbf{X}(t_k) - \mathbf{s}(t_l, t_k)$ is a constant and $\mathbf{W}(t_l, t_k)$ is a Gaussian random variable with zero mean and $(t_k - t_l)\boldsymbol{\sigma}^2$ variance, $\hat{\mathbf{X}}_s$ is also a Gaussian random variable with $\mathbf{X}(t_k) - \mathbf{s}(t_l, t_k)$ mean and $(t_k - t_l)\boldsymbol{\sigma}^2$ variance. Thus, the probability density function (PDF) of $\hat{\mathbf{X}}_s$ in the x and y directions are as follows:

$$\begin{aligned} f(x_s) &= \frac{e^{-\frac{(x_j - s_x - x_s)^2}{2(t_k - t_l)\sigma_x^2}}}{\sqrt{2\pi(t_k - t_l)\sigma_x^2}} \\ f(y_s) &= \frac{e^{-\frac{(y_j - s_y - y_s)^2}{2(t_k - t_l)\sigma_y^2}}}{\sqrt{2\pi(t_k - t_l)\sigma_y^2}}. \end{aligned} \tag{6}$$

Since x and y directions are orthogonal, the joint PDF of $\hat{\mathbf{X}}_s$ is $f(x_s) \times f(y_s)$. Then, the probability of the estimated odor source being located in an arbitrary cell can be calculated by integrating the PDF over positions in that cell.

Let $p_{ij}(t_l, t_k)$ donate the probability of there being an odor source in a cell C_i that released a single odor plume at time t_l given that the odor plume is in the cell C_j at time t_k. Thus, $p_{ij}(t_l, t_k)$ can be calculated as follows:

$$\begin{aligned} & p_{ij}(t_l, t_k) \\ &= \int_{x_s \in C_i} \int_{y_s \in C_i} \frac{e^{-\frac{(x_j - s_x - x_s)^2}{2(t_k - t_l)\sigma_x^2}}}{\sqrt{2\pi(t_k - t_l)\sigma_x^2}} \frac{e^{-\frac{(y_j - s_y - y_s)^2}{2(t_k - t_l)\sigma_y^2}}}{\sqrt{2\pi(t_k - t_l)\sigma_y^2}} dx_s dy_s \\ &= \int_{x_i - \frac{L_x}{2}}^{x_i + \frac{L_x}{2}} \int_{y_i - \frac{L_y}{2}}^{y_i + \frac{L_y}{2}} \frac{e^{-\frac{(x_j - s_x - x_s)^2}{2(t_k - t_l)\sigma_x^2}} e^{-\frac{(y_j - s_y - y_s)^2}{2(t_k - t_l)\sigma_y^2}}}{2\pi(t_k - t_l)\sigma_x\sigma_y} dx_s dy_s \\ &= \int_{-\frac{L_x}{2}}^{\frac{L_x}{2}} \int_{-\frac{L_y}{2}}^{\frac{L_y}{2}} \frac{e^{-\frac{(x_j - x_i - s_x - x_s)^2}{2(t_k - t_l)\sigma_x^2}} e^{-\frac{(y_j - y_i - s_y - y_s)^2}{2(t_k - t_l)\sigma_y^2}}}{2\pi(t_k - t_l)\sigma_x\sigma_y} dx_s dy_s \end{aligned} \tag{7}$$

which is a function of the relative positions of the cell C_j (i.e., plume position) and the cell C_i (i.e., possible odor source position), plume advection distances $\mathbf{s}(t_k, t_l)$, and the plume propagation time $t_k - t_l$. In the algorithm implementation, $p_{ij}(t_l, t_k)$ is approximated as

$$p_{ij}(t_l, t_k) = \frac{e^{-\frac{(x_j - x_i - s_x)^2}{2(t_k - t_l)\sigma_x^2}} e^{-\frac{(y_j - y_i - s_y)^2}{2(t_k - t_l)\sigma_y^2}}}{2\pi(t_k - t_l)\sigma_x\sigma_y} L_x L_y \tag{8}$$

for the calculation efficiency. This approximation will introduce additional errors when the cell size is large (i.e., $L_x >> \sigma_x$ and $L_y >> \sigma_y$), but in this work, the cell size is small (i.e., $L_x \approx \sigma_x$ and $L_y \approx \sigma_y$), thus, the produced errors are negligible.

One feature of the olfactory sensing device is that it has trivial false-alarm rates, but high missed-detection rates [18]. To model this mechanism, let β donate the probability of the robot successfully detecting plumes given that there are detectable plumes at the chemical sensor position. Thus, the probability of detecting a single released plume in C_j at t_k that was released from C_i at t_l is $\beta p_{ij}(t_l, t_k)$, and the probability of not detecting this plume is $1 - \beta p_{ij}(t_l, t_k)$.

3) Continuous Released Odor Plumes: If the odor source continuously releases odor plumes from t_l to t_k, what is the probability of detecting and not detecting odor plumes? To answer this question, the value of t_l should be clarified.

As the plume traveling time $t_k - t_l$ increases, the value of $\mathbf{s}(t_l, t_k)$ grows correspondingly, but $\mathbf{s}(t_l, t_k)$ should not exceed the size of the search area since the robot cannot detect odor plumes outside of it. The value of t_l is initialized as 0 because the airflow velocity is unavailable before the search, and as the search progresses (i.e., t_k increases), the distance between t_l and t_k should be constrained (i.e., $t_k - t_l < h$, where h is the maximum length of recorded flow history) to satisfy the aforementioned restraint. Therefore, t_l is defined as follows:

$$t_l = \max(0, t_k - h + 1). \tag{9}$$

In Section III-B2, the probability of not detecting a plume in a cell C_j at time t_k due to a single odor plume release from a cell C_i at time t_l is calculated as $1 - \beta p_{ij}(t_l, t_k)$. Thus, if all release times within $[t_l, t_k]$ are accounted, the probability of not

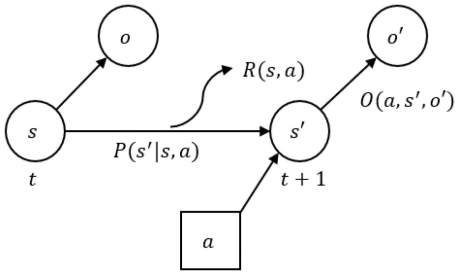


Fig. 3. Basic POMDP model.

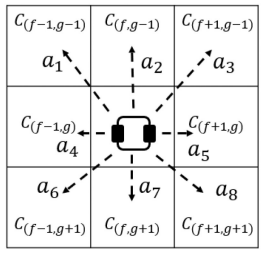


Fig. 4. Action space. The robot location is at the center cell. Arrows indicate possible actions that the robot can take.

detecting plumes in a cell C_j at time t_k due to the continuous plume release from a cell C_i is

$$\kappa_{ij}\left(t_l, t_k\right) = \prod_{t_l}^{t_{k-1}} \left[1 - \beta p_{ij}\left(t_l, t_k\right)\right]. \tag{10}$$

Since plume detection and nondetection events are complementary, the probability of detecting plumes under the same condition (i.e., continuous plume release) is $1 - \kappa_{ij}(t_l, t_k)$.

C. Source Mapping

Belief states in a POMDP model are adapted as source estimations, which are used to construct a source probability map. A basic POMDP model can be defined by a tuple ($\mathcal{S}$, $\mathcal{A}$, Ω, P, O, R, b_0) as shown follows [31]:

- $\mathcal{S}$ is a state space.
- $\mathcal{A}$ is an action space.
- Ω is an observation space.
- P are state transition probabilities between states.
- O are observation probabilities.
- R is the reward function defined on the transitions.
- b_0 is an initial probability distribution over states.

As shown in Fig. 3, at each time-step, the agent receives an observation (o, $o \in \Omega$) at the current state (s, $s \in \mathcal{S}$), and after performing an action (a, $a \in \mathcal{A}$), the agent is transferred to a new state (s', $s' \in \mathcal{S}$) according to the state transition probability $P(s'|s, a)$ and receives a new observation (o', $o' \in \Omega$) with the observation probability $O(a, s', o')$ and a reward $R(s, a)$. Since states are hidden to the agent, a probability distribution over states is defined as the belief state $b(s)$, which indicates the probability of the agent being in a particular state s, and the initial belief state is b_0.

To illustrate the proposed source mapping algorithm, the rest of section presents an approach that adapts elements in a basic POMDP model to the context of an OSL problem.

1) State Space: States in a basic POMDP model are hidden to the agent, i.e., the agent does not know which state it is in. In an OSL problem, the actual odor source location is unknown to the robot. Thus, we defined states as the actual odor source location, which can be represented by a length-M (M is the number of cells in the search area) vector of Boolean values indicating whether each cell contains the odor source. If the Boolean value is 1, then the corresponding cell contains the odor source; otherwise, this value is 0.

For instance, $s_1 = [1, 0, ..., 0]$ indicates that the odor source is in the cell C_1, $s_2 = [0, 1, ..., 0]$ represents that the odor source is in the cell C_2, etc. Since the odor source could be located in an arbitrary cell inside the search area, the state space is represented as $\mathcal{S} = \{s_1, s_2, ...s_M\}$.

2) Action Space: The action space defines possible actions that an agent could select. As shown in Fig. 4, at the center cell $C_{(f,g)}$, the robot could select one of eight actions and enter the corresponding cell around it.

In this work, an action is represented by the destination cell. For example, a_2 in Fig. 4 can be represented as $a_2 = C_{(f,g-1)}$ since the destination cell of this action is $C_{(f,g-1)}$. Thus, the action space can be represented as $\mathcal{A} = \{a_1 = C_{(f-1,g-1)}, a_2 = C_{(f,g-1)}, ..., a_8 = C_{(f+1,g+1)}\}$.

3) State Transition Probabilities: The location of the odor source is stationary in this work, i.e., the odor source cannot move. Thus, the state transition probability is 1 if the new state is the same as the old state; otherwise, this probability is 0

$$P(s' = s_i | s = s_j, a) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \tag{11}$$

where $i, j \in [1, M]$.

4) Observation Space and Probabilities: When the robot enters a cell C_j, it could or could not detect odor plumes. Thus, two observation states are defined in the observation space $\Omega = \{d, \bar{d}\}$, namely the plume detection event d and the plume nondetection event $\bar{d}$. A fixed plume concentration threshold is adopted to identify two events, i.e., a plume detection event is confirmed when the sensed concentration is higher than the threshold, otherwise, a plume nondetection event is confirmed.

When the robot enters a cell C_j, the probability of detecting continuous released plumes is $1 - \kappa_{ij}(t_l, t_k)$ and the probability of not detecting these plumes is $\kappa_{ij}(t_l, t_k)$ as defined in Section III-B3. Thus, the observation probability $O(a, s', o')$ is defined as follows:

$$O(a = C_j, s' = s_i, o') = \begin{cases} 1 - \kappa_{ij}(t_l, t_k) & o' = d \\ \kappa_{ij}(t_l, t_k) & o' = \bar{d} \end{cases} \tag{12}$$

where $i, j \in [1, M]$.

5) Belief States: In a basic POMDP model, after taking an action a and transferring to a new state s', the belief state of the new state $b(s')$ is updated based on the old belief state $b(s)$, the observation probability O, and the state transition probability

P, which is represented as [32]

$$b(s') = \frac{O(a, s', o') \sum_{s \in S} P(s'|s, a) b(s)}{\sum_{s \in S} \sum_{s' \in S} O(a, s', o') P(s'|s, a) b(s)}. \quad (13)$$

In an OSL problem, the belief state can be interpreted as the probability of the robot believing that there is an odor source in a cell. Based on the defined observation probability (12) and the state transition probability (11), (13) can be rewritten as follows:

$$b(s' = s_i) = \begin{cases} \frac{(1-\kappa_{ij}(t_l,t_k))b(s_i)}{\sum_{i=1}^{M}(1-\kappa_{ij}(t_l,t_k))b(s_i)} & o' = d \\ \\ \frac{\kappa_{ij}(t_l,t_k)b(s_i)}{\sum_{i=1}^{M}\kappa_{ij}(t_l,t_k)b(s_i)} & o' = \bar{d} \end{cases}. \quad (14)$$

The above equations iteratively update belief states depending on plume detection and nondetection events. The initial belief state b_0 is defined as $1/M$, since the prior information about the odor source location is unavailable to the robot before it starts the search. However, it could be exploited through an appropriate distribution of b_0 to reflect the prior knowledge known about the source if the information regarding the source location is available prior to the search.

In summary, by calculating belief states over all states, a source probability map $\boldsymbol{b}$ that estimates the source location is obtained

$$\boldsymbol{b} = [b(s_1), b(s_2), ..., b(s_M)]. \quad (15)$$

D. Plume Mapping

The plume mapping algorithm produces a plume distribution map, which indicates possible plume propagation areas. With the produced source probability map and the recorded airflow history, an HMM-based plume mapping algorithm [20] is presented in this section.

Let $\alpha_j(t_0, t_k)$ denote the probability that a cell C_j contains the detectable odor plume at time t_k due to the continuous plume release by the source starting at t_0, where t_0 is the initial time that the robot records airflow measurements. Denote

$$\boldsymbol{\alpha}(t_0, t_k) = [\alpha_1(t_0, t_k), \alpha_2(t_0, t_k), ..., \alpha_M(t_0, t_k)] \quad (16)$$

as the vector storing this variable for each cell, which is a plume distribution map at the current time t_k.

Introduce the variable $\bar{\alpha}_j(t_0, t_k)$ representing the probability that a cell C_j contains a detectable odor plume at t_k due to a single plume release at time t_0. Define

$$\bar{\boldsymbol{\alpha}}(t_0, t_k) = [\bar{\alpha}_1(t_0, t_k), \bar{\alpha}_2(t_0, t_k), ..., \bar{\alpha}_M(t_0, t_k)] \quad (17)$$

as the vector form of $\bar{\alpha}_j(t_0, t_k)$. At t_0, the plume propagation has not occurred yet and plumes are at the odor source location, therefore, $\bar{\boldsymbol{\alpha}}(t_0, t_0) = \boldsymbol{b}$ since the actual odor source location is unknown. To find $\bar{\alpha}_j(t_0, t_1)$, which is the probability of a cell C_j containing plumes after one time step, the plume transitions from all other cells to the cell C_j must be considered, i.e.,

$$\bar{\alpha}_j(t_0, t_1) = \sum_{k=1}^{M} \bar{\alpha}_k(t_0, t_0) a_{kj}(t_0) \quad (18)$$

where $a_{kj}(t_0)$ denotes the probability of the one step transition of odor plumes from a cell C_k at time t_0 to another cell C_j at time t_1, which can be obtained from the airflow history [20]. In addition, (18) can be rewritten in a vector notation

$$\bar{\boldsymbol{\alpha}}(t_0, t_1) = \bar{\boldsymbol{\alpha}}(t_0, t_0) \bar{\mathbf{A}}(t_0) \quad (19)$$

where $\bar{\mathbf{A}}(t_0) = [a_{kj}(t_0)] \in \mathcal{R}^{M \times M}$ is the matrix form of $a_{kj}(t_0)$. Moreover, for an arbitrary plume propagation period, i.e., $(t_k - t_0)$, $\bar{\boldsymbol{\alpha}}(t_0, t_k)$ can be calculated as follows:

$$\bar{\boldsymbol{\alpha}}(t_0, t_k) = \begin{cases} 0, & \text{for } t_k < t_0 \\ \boldsymbol{b}\mathbf{I}, & \text{for } t_k = t_0 \\ \boldsymbol{b}\Phi(t_0, t_k), & \text{for } t_k > t_0 \end{cases}. \quad (20)$$

where $\mathbf{I}$ is the identity matrix with the size of $M \times M$ and $\Phi(t_0, t_{k+1}) = \prod_{q=0}^{k} \bar{\mathbf{A}}(t_q)$.

For the continuous plume release scenario, $\boldsymbol{\alpha}(t_0, t_k)$ can be derived from the single release case $\bar{\boldsymbol{\alpha}}(t_0, t_k)$ by considering all release times from t_0 to t_k

$$\boldsymbol{\alpha}(t_0, t_k) = \frac{1}{k+1} \sum_{q=0}^{k} \bar{\boldsymbol{\alpha}}(t_q, t_k) \quad (21)$$

where $1/(k+1)$ is the normalization factor to maintain $||\boldsymbol{\alpha}(t_0, t_k)||_1 = 1$. With (20) and (21) can be further reduced as follows:

$$\begin{aligned} \boldsymbol{\alpha}(t_0, t_k) &= \frac{1}{k+1} \left[\sum_{q=0}^{k} \boldsymbol{b}\Phi(t_q, t_k) \right] \\ &= \frac{\boldsymbol{b}}{k+1} \left[\Phi(t_k, t_k) + \sum_{q=0}^{k-1} \Phi(t_q, t_k) \right] \\ &= \frac{\boldsymbol{b}}{k+1} \left[\mathbf{I} + \sum_{q=0}^{k-1} \Phi(t_q, t_k) \right] \\ &= \boldsymbol{b}\Psi(t_0, t_k), \end{aligned} \quad (22)$$

where $\Psi(t_0, t_k) = 1/(k+1)[\mathbf{I} + \sum_{q=0}^{k-1} \Phi(t_q, t_k)]$, which can be iteratively updated as follows:

$$\Psi(t_0, t_k) = \frac{1}{k+1} \left[\mathbf{I} + k\Psi(t_0, t_{k-1}) \bar{\mathbf{A}}(t_{k-1}) \right]. \quad (23)$$

Note that, since $\bar{\mathbf{A}}(t_{k-1})$ relates to the latest airflow measurement, (23) encapsulates the airflow history over the entire search time (i.e., from t_0 to t_k).

In summary, when the source probability map $\boldsymbol{b}$ is available at the current time t_k, a plume propagation map $\boldsymbol{\alpha}(t_0, t_k)$ can be obtained by (22), and if $\boldsymbol{b}$ is updated in the next time step, by updating Ψ with the latest airflow measurements, a renewed plume distribution map based on the new source probability map can be obtained.

IV. Planning

A. Generate Reward Functions With Fuzzy Inference

After the source probability map $\boldsymbol{b}$ and the plume distribution map $\boldsymbol{\alpha}(t_0, t_k)$ are obtained, information from two maps is fused and assigned to reward functions. The information provided by two maps is complementary for determining robot behaviors,

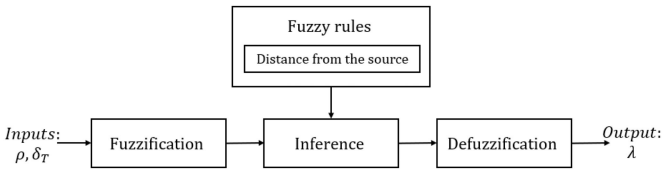


Fig. 5. Structure of the proposed fuzzy controller. ρ and δ_T are sensed plume concentration and plume non-detection period, respectively, and λ is the fusion coefficient.

i.e., the robot either moves to the estimated source location or to the possible plume areas. Thus, a weighted superposition pattern is adopted to combine two maps, and values from two maps are normalized with the min–max normalization before the combination.

Let define four constants, $b_{\max}$, $b_{\min}$, $\alpha_{\max}$, and $\alpha_{\min}$, as maximal and minimal values in $\boldsymbol{b}$ and $\boldsymbol{\alpha}(t_0, t_l)$, respectively. These constants can be determined before computing reward functions. For an action that moves the robot into a cell C_j, the reward function is

$$R(s, a = C_j) = \lambda \frac{b(s_j) - b_{\min}}{b_{\max} - b_{\min}} + (1 - \lambda) \frac{\alpha_j(t_0, t_k) - \alpha_{\min}}{\alpha_{\max} - \alpha_{\min}} \tag{24}$$

where $\lambda \in [0, 1]$ is the fusion coefficient that controls the balance of two maps. When $\lambda > 0.5$, the source probability map is chiefly dominated in reward functions, which results the robot surging to the estimated source location, i.e., the exploitation. Conversely, when $\lambda < 0.5$, the robot moves to possible plume areas since the plume distribution map outweighs the counterpart in reward functions, i.e., the exploration. An ideal value of λ should be adaptive for different search circumstances to generate the optimal search objective.

The primary hurdle of determining the value of λ is the vagueness of search circumstances. A critical question to ask is that under what conditions the robot should choose its search objective as the exploration or the exploitation. Attempts that use mathematical methods to quantitatively analyze differences of search conditions and assign a precise λ for different search circumstances are hard to implement due to uncertainties of the source location and the search environment.

Inspired by implementations of fuzzy theory in the field of decision making [33] and data classification [34], which successfully handles the problems with vagueness and uncertainties, a fuzzy inference approach is employed. In fuzzy theory, vague variables and environments can be handled in a deterministic manner via linguistic descriptions and rules. By analyzing sensor measurements, such as plume concentrations, search circumstances are expected to be identified. Then, the corresponding λ that dynamically balances the exploitation and exploration of the odor source information can be generated based on defined fuzzy rules.

As shown in Fig. 5, procedures of the proposed fuzzy controller include fuzzification, defining fuzzy rules, and defuzzification [35].

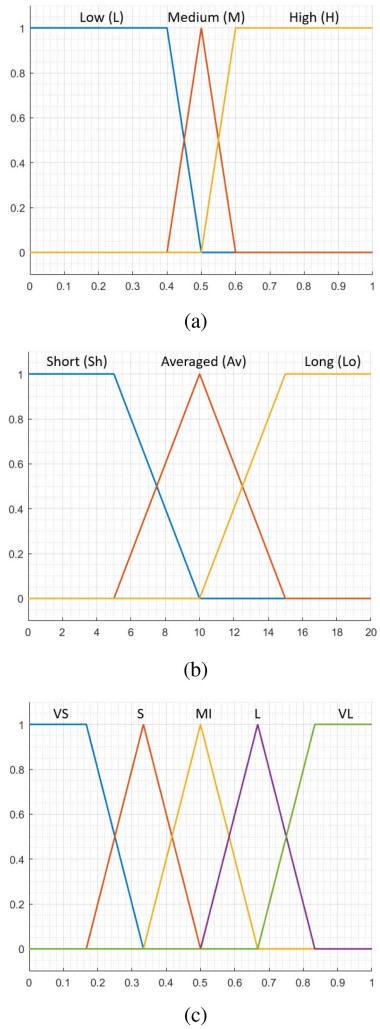


Fig. 6. Fuzzy sets and membership functions of antecedents and consequent. (a) Sensed plume concentration, ρ. (b) Plume nondetection period, δ_T. (c) Fusion coefficient, λ.

1) Fuzzification: Fuzzification is the process that changes real scalar values of antecedent and consequent (i.e., inputs and outputs) into fuzzy values, which are the degree of uncertainty that scalar values belong in a fuzzy set.

In this work, inputs are utilized to conjecture the distance from the robot to the odor source and the output is the value of λ. If the robot is close to the source, it surges toward the source ($\lambda > 0.5$); otherwise, it leans to detect plumes to gather more information ($\lambda < 0.5$). The sensed plume concentration at the robot location $\rho(\mathbf{X}_v(t), t)$ (for the simplification purpose, we use ρ as plume concentration measurements in the rest of this article) is set as an input, and due to the existence of local concentration maxima along the plume trajectory [36], the plume nondetection period δ_T, i.e., the time interval between two detection events, is added to inputs. Since the positions of local concentration maxima are time varying in a turbulent flow environment [37], if the robot senses consecutive high odor concentrations in a short period, it is very likely that the robot is near to the source.

Fig. 6 shows plots of membership functions of inputs and the output, which are determined based on the distribution

TABLE I
LIST OF FUZZY RULES

Rule No.	Rule
$\mathcal{F}^1$	IF ρ is L AND δ_T is Sh, THEN λ is MI
$\mathcal{F}^2$	IF ρ is L AND δ_T is Av, THEN λ is S
$\mathcal{F}^3$	IF ρ is L AND δ_T is Lo, THEN λ is VS
$\mathcal{F}^4$	IF ρ is M AND δ_T is Sh, THEN λ is L
$\mathcal{F}^5$	IF ρ is M AND δ_T is Av, THEN λ is MI
$\mathcal{F}^6$	IF ρ is M AND δ_T is Lo, THEN λ is S
$\mathcal{F}^7$	IF ρ is H AND δ_T is Sh, THEN λ is VL
$\mathcal{F}^8$	IF ρ is H AND δ_T is Av, THEN λ is VL
$\mathcal{F}^9$	IF ρ is H AND δ_T is Lo, THEN λ is L

of measured data from experiments. As shown in Fig. 6, all membership functions are triangular. Three fuzzy sets have been defined to cover the discourse of universe of the sensed plume concentration ρ, namely low (L), medium (M), and high (H). The discourse of universe of the plume nondetection period δ_T is also covered by three fuzzy sets, namely short (Sh), averaged (Av), and long (Lo). For the output λ, five fuzzy sets, namely very small (VS), small (S), middle (MI), large (L), and very large (VL), are defined to cover its discourse of universe.

2) Fuzzy Rules: Fuzzy rules in the fuzzy inference theory are presented in a "IF–THEN" format, which determine search strategies of the robot. In this work, fuzzy rules are designed based on moth odor searching behaviors [38]. As mentioned, previous researchers [16], [17], [39] have summarized these behaviors into a "surge/casting" model and demonstrated the validity of implementing this model on robots in OSL problems. Borrowing this idea, we want the robot to explore if plumes are absent (like the moth's casting behavior) and to exploit when the robot is in plumes (like the moth's surge behavior). To achieve this mechanism, the distance from the robot to the odor source is estimated and monitored: if the robot is far from the source, the robot inclines to find plumes, i.e., exploration; otherwise, the robot tends to search the source, i.e., exploitation.

The inclination of changing λ is that: when ρ is high and δ_T is short, the robot is very likely being close to the odor source; thus, λ is large. On the other hand, when ρ is low and δ_T is long, the robot is probably far from the source; thus, λ is small. In a "IF–THEN" format, the above rules are represented as follows:

$\mathcal{F}^1 = \{$IF ρ is H AND δ_T is Sh, THEN λ is VL $\}$

$\mathcal{F}^2 = \{$IF ρ is L AND δ_T is Lo, THEN λ is VS $\}$.

Enumerate all possible combinations of antecedents and the corresponding consequent, a rule table (see Table I) can be obtained.

3) Defuzzification: The centroid method [40] is chosen as the defuzzification algorithm, which can be expressed as follows:

$$U_0 = \frac{\sum_{i=1}^{n} U_i \cdot \mu(U_i)}{\sum_{i=1}^{n} \mu(U_i)} \tag{25}$$

where U_0 is the output (i.e., the value of λ), i is the index of rules $i \in [1, 9]$, $\mu(U_i)$ is the truth value of result membership function for the ith rule, and U_i is the value where the result membership function is maximum over the output variable fuzzy set range.

Fig. 7 presents the result of the proposed fuzzy controller. In general, a small ρ and a long δ_T produce a trivial λ value that

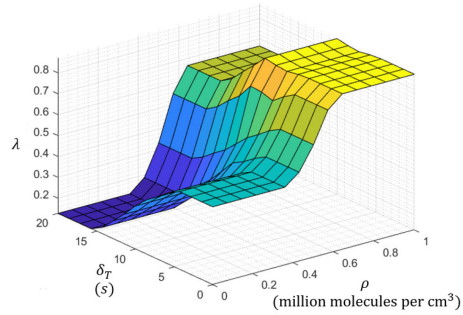


Fig. 7. Result of the proposed fuzzy controller. In the plot, the horizontal axes are two inputs, the sensed odor concentration ρ and the plume non-detection period δ_T, and the vertical axis is the output, the fusion coefficient λ.

Algorithm 1: Value Iteration Based Planning Algorithm.

```
Initialize Value Functions V(C_i) = 0, i ∈ [1, M]
Calculate Reward Functions R(s, a) for all cells based on (24)
Set the convergence tolerance ϵ
while Δ ≥ ϵ do
  δ = 0
  for i ∈ [1, M] do
    v = V(C_i)
    V(C_i) = max_{a∈A}(R(s, a) + γV(a))
    Δ = max(Δ, |v − V(C_i)|)
  end for
end while
Generate the optimal policy π* = argmax V(C_i)
```

emphasizes the plume mapping information in reward functions, and the opposite combination of ρ and δ_T (i.e., a large ρ and a short δ_T) provides a large λ that prioritizes the source mapping information in reward functions.

B. Solve for the Optimal Policy

After reward functions are determined, a search route is generated in the planning procedure. Given the current reward functions, we adopt a value iteration method (see Algorithm 1) to fast determine the optimal policy, i.e., the search route. The motivation of using this method is to reduce the processing time, which allows the robot to timely respond to new plume observations. By contrast, solving the POMDP [41] is also feasible to obtain the search route, but, considering the large size of the hidden state space defined in our POMDP, this approach becomes time-consuming and intractable. The ability of fast solving the searching path is one of our main concerns, since the ultimate goal of this article is implementing this algorithm on a mobile robot, which has limited onboard computational resources.

As shown in Algorithm 1, value functions of all cells are initialized as 0. If the robot is currently located in a cell C_i, it could choose one of eight actions (see Fig. 4) and enter the corresponding cell. Note that, based on the reliable maneuverability of the ground robot, it is assumed that the transition of

robot positions is deterministic, i.e., after taking an action, the robot can correctly enter the corresponding cell. Based on the Bellman optimality equation [27], the value function of a cell C_i can be calculated by

$$V(C_i) = \max_{a \in \mathcal{A}} \left(R(s,a) + \gamma V(a) \right) \tag{26}$$

where $\gamma \in [0,1]$ is the discounting factor that penalizes future rewards. In this work, the larger the γ is, the broader region the robot considers in planning the search route. In implementations, we set γ to be 0.9.

A convergence tolerance ϵ is defined to check whether or not value functions converge. The maximal update of value functions (i.e., Δ in Algorithm 1) is compared with ϵ, and value functions are considered as converged if $\Delta < \epsilon$, otherwise value functions keep updating. To balance the tradeoff between the navigation performance and the processing time, we set ϵ as 10^{-6} in experiments to obtain a well-algorithm performance and to save the processing time. After value functions converge, the optimal policy is obtained by selecting the optimal action with the maximal value function. Thus, a series of optimal actions can be obtained between the robot current position and the maximal reward location, which is a search route that guides the robot toward the position with the most odor source information.

It should be noted that this optimal policy is obtained based on the current reward functions and is not permanent. In every time step, new observations, i.e., plume detection and nondetection events, update reward functions via (24), and a new optimal policy will be determined via Algorithm 1. The new policy overwrites the old one, which allows the robot to timely adjust its searching targets (exploration or exploitation) and trajectories to intelligently fit new observations. In general, the overall searching trajectory is a combination of a sequence of optimal policies generated from different reward functions.

V. Experiments

A. Simulation Setup

1) Simulated Environment: The proposed olfactory-based navigation algorithm is evaluated in a simulation program that was designed based on [6]. In this simulation program, time-varying plume trajectories in different flow environments can be customized and created. Some other researchers, such as [18], [20], and [42], also employed this simulator as the evaluation tool for their works.

Fig. 8 shows the simulated search area, where the size is $100 \times 100\,\mathrm{m}^2$. Over the search area, a coordinate is constructed with 40×25 cells in the x and y directions, respectively. An odor source is located at (20, 0) m (at the cell C_{332}) and releases 10 plumes per second. Released plumes form a circular plume trajectory as plotted by a grey-scale patchy trail. Local airflow vectors are presented by arrows in the background, where the tail of the arrow points to the airflow direction and the length of an arrow indicates the strength of airflow velocity. In the simulator, airflows are calculated from time-varying boundary conditions that are generated by a mean flow ($\mathbf{U}_0$) and Gaussian white noises (zero mean and ς variance). Thus, by altering values of boundary condition variables (i.e., $\mathbf{U}_0$ and ς), varying amplitudes of airflow fields can be created.

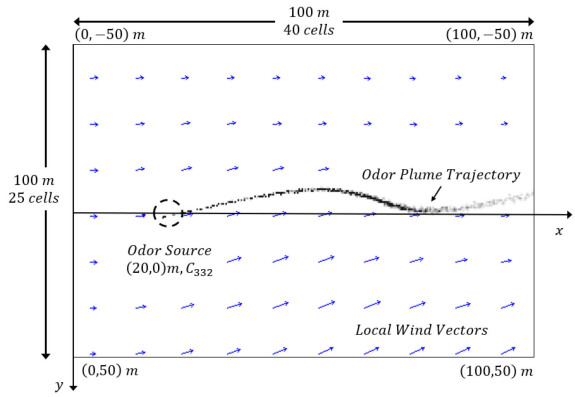


Fig. 8. Search area in the simulation program.

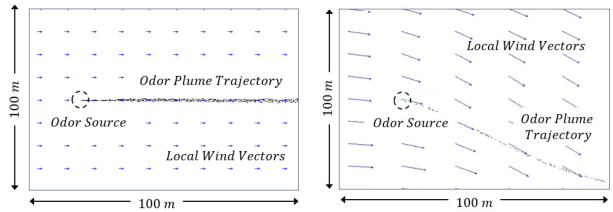


Fig. 9. Airflow fields and corresponding odor plume trajectories in the simulation with different environmental settings. (a) Laminar flows, $\mathbf{U}_0 = (1, 0)$ m/s and $\varsigma = 0$. (b) Turbulent flows, $\mathbf{U}_0 = (3, 0.5)$ m/s and $\varsigma = 30$.

TABLE II
Values of Parameters in Gaussian Noises in Sensor Measurements

	Chemical Sensor	Anemometer	Positioning Sensor
Mean	0	0	0
Standard deviation	0.05 mmpv	0.1 m/s and 1°	0.1 m

mmpv: million molecules per cm^3.

Fig. 9 shows snapshots of two simulated airflow fields. In the left diagram [see Fig. 9(a)], $\mathbf{U}_0 = (1, 0)$ m/s and $\varsigma = 0$, a laminar airflow filed is created, and in the right diagram [see Fig. 9(b)], $\mathbf{U}_0 = (3, 0.5)$ m/s and $\varsigma = 30$, a turbulent airflow field is produced.

2) Vehicle Assumptions: Comparing to the large scale of the search area, the size of the robot is negligible. Thus, the robot is approximated as a single point in the simulation. It is assumed that the robot is equipped with a chemical sensor, an anemometer, and a positioning sensor, which measure plume concentrations, wind speeds and directions in the inertial frame, and the robot position in the inertial frame, respectively. Measurements of all sensors are corrupted with Gaussian white noises to imitate the real-world applications, where parameters of noises are listed in Table II. The proposed olfactory-based navigation algorithm is operated on an onboard computer to process sensor readings and produce the target heading θ and

velocity v, which are limited in ranges of $[-180°, 180°]$ and $[0.6, 1]$ m/s, respectively. During the search, the robot follows heading and velocity commands to find the odor source.

3) Experiment Designs: To evaluate the performance of the proposed navigation method, around 60 tests have been conducted on the simulation program.

These tests can be separated into three groups. In the first group of tests, the proposed source mapping, plume mapping, and fusion algorithms are implemented in a laminar flow environment to verify their validities. Source and plume estimations are presented and compared with actual source and plume locations. Tests in the second group are carried out to investigate the effectiveness of implementing the proposed navigation method in a turbulent flow environment. Results are compared with a moth-inspired method [16] and a Bayesian inference method [18], which are typical plume tracing approaches in categories of bio-inspired and engineering-based methods, respectively. In the last group of tests, the robustness of the proposed navigation method is evaluated. Various search conditions, including varying initial search positions and airflow fields, are designed. Similar to the second group of tests, implementation results of the proposed navigation method are compared with two typical plume tracing methods.

B. Group 1: Implementation in a Laminar Flow Environment

This section demonstrates the results of implementing the proposed source mapping (see Section III-C5), plume mapping (see Section III-D), and fusion algorithms (see Section IV) in the simulation. The robot starts the OSL task at $(60, -40)$ m (or in the cell C_{64}) in a laminar flow environment, where $\mathbf{U}_0 = (2, 0)$ m/s and $\varsigma = 5$.

Fig. 10 shows results of the above algorithms after the robot detecting odor plumes for the first time. To visualize algorithm results, cells in the search area are painted with various colors, where darker cells have the higher values (red: largest, white: lowest). Depending on the implementing algorithm, the value of a cell could be the probability of containing the source (i.e., the result of source mapping), the probability of carrying plumes (i.e., the result of plume mapping), and the reward function (i.e., the result of fusion algorithm).

For the source mapping algorithm, it can be observed in Fig. 10(a) that possible source locations are narrowed to upflow areas of the plume detection location, and as shown in Fig. 10(b), the cell with the maximal probability of containing the odor source is close to the actual odor source location. For the plume mapping algorithm, Fig. 10(a) and (d) illustrates plume estimations and the plume distribution map, respectively. As shown in these two diagrams, plume estimations correctly overlap with the actual plume trajectory, and the cell with the peak probability of containing plumes is located at the upflow area, which is a reasonable estimation of plume propagation areas.

For the fusion algorithm, due to the low sensed plume concentration ρ and the short plume nondetection period δ_T, the fusion coefficient λ is middle according to the defined fuzzy rules. Fig. 10(e) presents the distribution of reward functions over the search area. It can be seen that reward functions cover

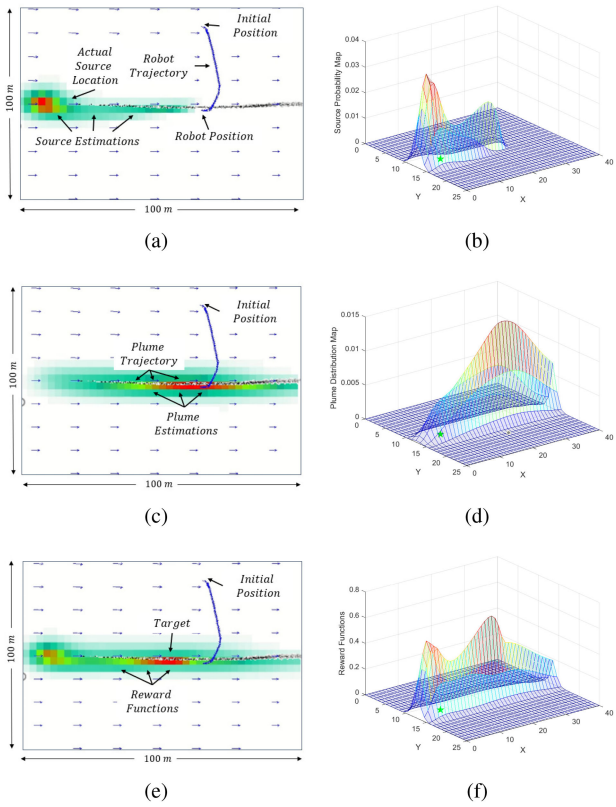


Fig. 10. Results of the source mapping, plume mapping, and fusion algorithms after the robot detecting plumes for the first time. In the left column of diagrams [i.e., (a), (c), and (e)], cells in the search area are painted with different colors to reflect various values of algorithm results. In the right column of diagrams [i.e., (b), (d), and (f)], the horizontal plane is the grid that covers the search area, and the star mark in the horizontal plane indicates the actual odor source location. (a) Source estimations. (b) Source probability map. (c) Plume estimations. (d) Plume distribution map. (e) Reward functions. (f) The plot of reward functions.

the plume trajectory, and the maximal reward location (i.e., the target of the path planning algorithm), as shown in Fig. 10(f), is at the upflow area of the last plume detection location. Test results reveal that reward functions are instructive for the robot to collect more odor source information, since the robot is expected to detect more plumes at the maximal reward location.

C. Group 2: Implementation in a Turbulent Flow Environment

In this test, the proposed navigation method is implemented in an environment with turbulent flows ($\mathbf{U}_0 = (3, 0.5)$ m/s and $\varsigma = 30$). Fig. 11 shows reward functions over the search area at different times in an OSL task and the plot of the fusion coefficient λ. Similar to the first group of tests, cells in the search area are painted with different colors, where the darker color indicates the higher value in reward functions and vice versa.

The robot starts at the cell C_{64} and adopts a "zigzag" search trajectory at the beginning of the OSL task, where the source mapping algorithm is activated simultaneously. As shown in Fig. 11(a), the source mapping algorithm excludes possible source locations from the upflow areas of the robot (white areas), since the robot does not detect plumes. At 55 s, the robot detects

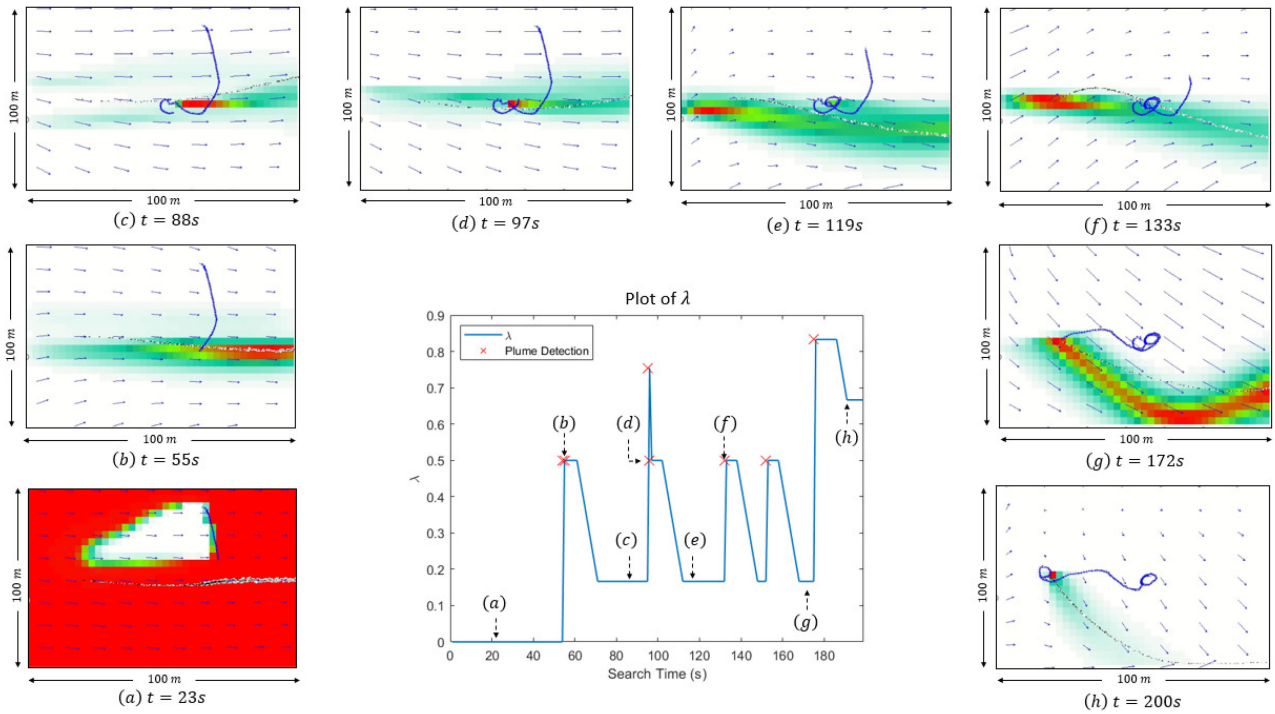


Fig. 11. Robot search trajectories and reward functions at different time steps in an environment with turbulent flows. The plot of the fusion coefficient λ versus the search time t is presented at the center of the diagram, where cross marks indicates plume detection events. Diagrams around the center plot are robot trajectories and reward functions at different time steps. For each of these diagrams, the robot trajectory is represented by the trail of dark arrows; the grey-scale patchy trail in the middle of the background indicates the simulated plume trajectory; cells are painted with colors according to their reward values, where darker cells have higher reward values (red: largest, white: smallest).

plumes for the first time, and the plume mapping and fusion algorithms are activated to generate reward functions.

From 55 to 172 s, the robot is in the exploration, where the robot is encouraged to detect plumes and gather odor source information. Specifically, it can be observed in the plot of λ that the value of λ fluctuates between 0 and 0.5 due to the low sensed odor concentration ρ and the long plume nondetection period δ_T. As the result, plume estimations outweigh source estimations in reward functions, which drives the robot to seek plumes. Note that, at 96 and 97 s, λ rises to 0.75 but quickly falls to 0.5. It is because the robot senses a local concentration maximum at 96 s, but the successive sensed concentration is not as high as the previous, which contributes the drop of λ. After successively detecting high concentration plumes at 175 s, the robot is in the exploitation state. At 175 s, λ rises to 0.83, which indicates that the robot is near the source, and the robot surges toward the estimated source location. At 200 s, reward functions converge to a single cell C_{292} [i.e., the red cell in Fig. 11(h)], which overlaps the actual odor source location, and the robot successfully finds the odor source location.

A moth-inspired method [16] and a Bayesian inference method [18] are implemented in the same environment to compare with the proposed method. Fig. 12 shows search trajectories, and Table III compares search times and travel distances of three navigation methods. As shown in Fig. 12(a), the moth-inspired method fails to find the odor source within the time limit (500 s). The primary reason is that the plume trajectory alters rapidly in a turbulent flow environment, thus, the robot can barely stay in plumes and surge upflow to seek the odor source. For the Bayesian inference method, it can be observed in Fig. 12(b) that due to the lack of plume estimations, the robot constantly circulates and tries to detect new plumes after the second plume detection event (i.e., the middle cross on the searching trajectory). As the result, the robot spends a lot of time to recover from plume nondetection events. By contrast, the proposed navigation method achieves the best performance with the shortest search time and travel distance. Compared to the search trajectory of the Bayesian inference method, the robot implemented with the proposed navigation method detects more plumes, and after several plume detection events, the robot surges toward estimated source location and correctly finds the odor source eventually. Results in this test verify the validity of implementing the proposed method in a turbulent flow environment.

TABLE III
SEARCH TIME AND TRAVEL DISTANCE OF THREE NAVIGATION METHODS IN A TURBULENT FLOW ENVIRONMENT

	Moth-inspired method	Bayesian inference method	Proposed method
Search time (s)	-	251	200
Travel distance (m)	-	223.96	172.07

-:Fail to locate the source within 500 s.

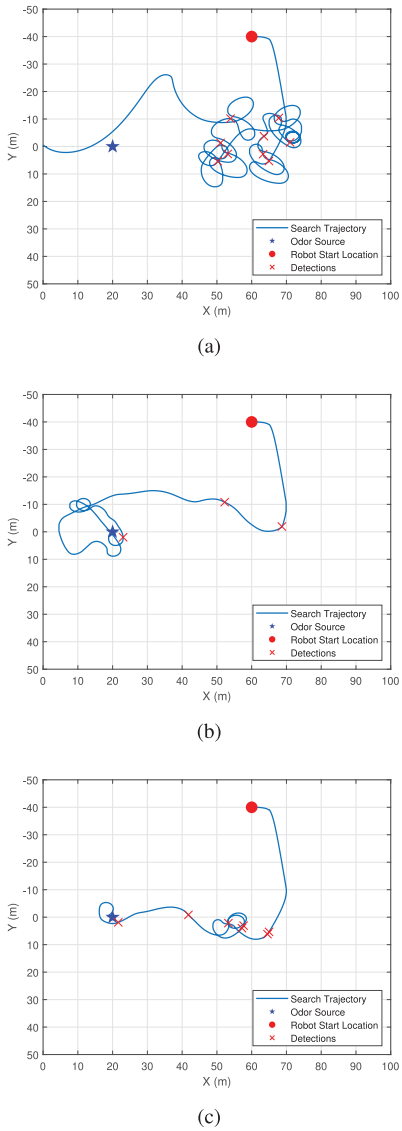


Fig. 12. Search trajectories of three navigation methods in a turbulent flow environment. (a) Moth-inspired method. (b) Bayesian-based method. (c) Proposed method.

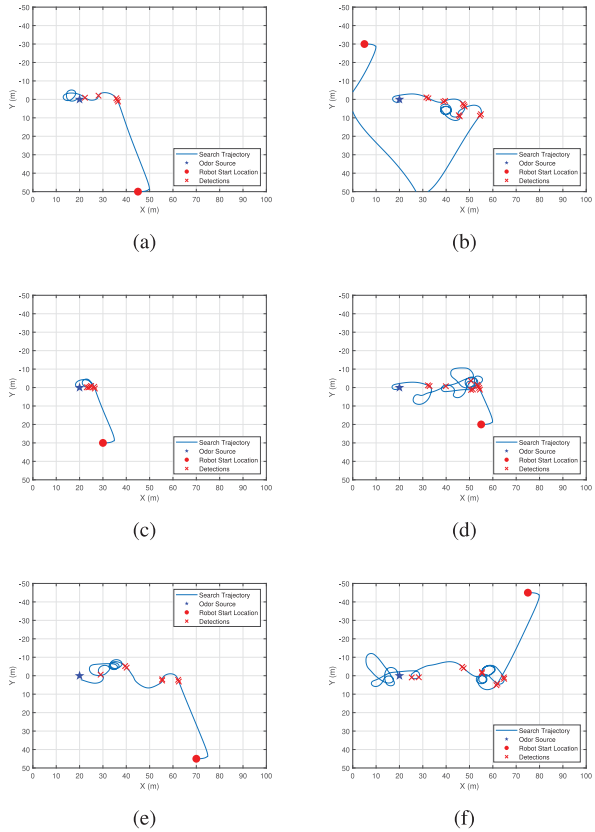


Fig. 13. Search trajectories of the proposed navigation method at different initial positions. The robot starts an OSL task from (a) $(45, 50)$ m. (b) $(5, -30)$ m. (c) $(30, 30)$ m. (d) $(55, 20)$ m. (e) $(70, 45)$ m. (f) $(70, -45)$ m.

D. Group 3: Varying Search Conditions

In this test, the robustness of the proposed navigation method in varying search conditions is investigated. Two scenarios have been designed in this test, which are listed in the following.

1) Scenario 1: the robot starts an OSL task at different initial positions in a turbulent flow environment ($\mathbf{U}_0 = (2, 0)$ m/s and $\varsigma = 15$).
2) Scenario 2: the robot starts at the same initial position, but airflow fields are varying ($u_x \in [1, 3]$ m/s, $u_y \in [-0.5, 0.5]$ m/s, and $\varsigma \in [10, 30]$).

1) Results of Scenario 1: Six tests have been conducted in Scenario 1, and Fig. 13 presents search trajectories of all tests. It can be observed in Fig. 13 that all search trajectories terminates at the actual source location, which is marked by a star, i.e., the robot correctly finds the odor source in all tests, which demonstrates the validity of the proposed navigation method with varying initial searching positions.

2) Results of Scenario 2: In order to evaluate the performance of the proposed navigation method, the moth-inspired method and the Bayesian-based method are also implemented and compared in this test. For each navigation method, the test is repeated 15 times in environments with varying airflow conditions. Table IV presents airflow conditions of 15 tests and search times of three navigation methods in the corresponding environment, and Table V lists statistical results of all tests. It can be seen from Table IV that the moth-inspired method barely succeeds to find the odor source in turbulent flow environments. Comparing to the Bayesian inference method, the robot with the proposed method achieves a higher success rate (100% vs. 80%) and a shorter averaged search time (199.5 s vs. 299.1 s), which illustrates the effectiveness of the proposed method in varying airflow environments.

E. Discussion

Results in the above tests reveal the effectiveness of the proposed olfactory-based navigation method. However, it is worth mentioning that the limitation of our method is the lack of global wind measurements. Due to this reason, as mentioned in Section III-B2, wind measurements at robot positions are

TABLE IV
ENVIRONMENTAL SETTINGS AND SEARCH TIMES OF DIFFERENT NAVIGATION METHODS

Tests	Mean Wind Velocity, $\mathbf{U}_0$ (m/s)	Gaussian Noise Variance, ς	Moth-inspired Method (s)	Bayesian inference Method (s)	Proposed Method (s)
Test 1	[1, -0.5]	10	-	245	171
Test 2	[1.2, 0.5]	11	-	160	177
Test 3	[2, 0]	15	-	223	150
Test 4	[2, 0.3]	18	275	367	229
Test 5	[3, 0.5]	12	-	489	214
Test 6	[3, -0.3]	20	-	418	194
Test 7	[2, 0.4]	30	-	-	282
Test 8	[3, -0.5]	25	-	248	152
Test 9	[2, -0.2]	24	-	303	231
Test 10	[1, 0.2]	28	-	-	152
Test 11	[2.4, 0.2]	28	-	287	283
Test 12	[2.1, 0.5]	13	-	243	224
Test 13	[1.2, -0.3]	12	-	-	213
Test 14	[1.1, -0.1]	28	-	375	194
Test 15	[2, -0.4]	10	-	231	126

-:Fail to locate the source within 500 s.

TABLE V
STATISTICAL RESULTS OF REPEATED TESTS AND THE COMPARISON OF THREE NAVIGATION METHODS

	Total Test	Successful Test	Success Rate	Averaged Time (s)
Moth-inspired Method	15	1	6.7%	275
Bayesian-based Method	15	12	80%	299.1
Proposed Method	15	15	100%	199.5

utilized to estimate plume advection distances. This approximation introduces additional errors if the robot is in a highly turbulent flow environment, i.e., wind directions and velocities have a huge variance in space. It can be observed in Table IV that the searching time grows significantly if wind fields are highly turbulent (e.g., Test 4, 7, and 11). This issue could be alleviated with the multiagent searching algorithm. By employing multiple robots, wind information at different locations are obtained, and a comprehensive wind map over the searching area could be derived. The design and implementation of the multiagent searching algorithm is one of our prospective research directions.

VI. CONCLUSION

An olfactory-based navigation algorithm based on model-based RL and fuzzy inference methods was presented in this article. The OSL problem was modeled as a model-based RL problem, in which belief states in a POMDP model are adapted to generate a source probability map, and a plume distribution map is constructed via an HMM-based method. The information from both maps was fused by a fuzzy inference based fuzzy controller and assigned to reward functions, and the value iteration method was adopted to solve for the optimal policy. Experiment results showed that the proposed navigation method was valid in turbulent flow environments. Besides, compared to the moth-inspired method and the Bayesian-based method, the proposed method is more effective and intelligent in turbulent flow environments.

REFERENCES

[1] G. Kowadlo and R. A. Russell, "Robot odor localization: A taxonomy and survey," *Int. J. Robot. Res.*, vol. 27, no. 8, pp. 869–894, 2008.

[2] M. Dunbabin and L. Marques, "Robots for environmental monitoring: Significant advancements and applications," *IEEE Robot. Autom. Mag.*, vol. 19, no. 1, pp. 24–39, Mar. 012.

[3] S. Soldan, G. Bonow, and A. Kroll, "Robogasinspector-a mobile robotic system for remote leak sensing and localization in large industrial environments: Overview and first results," *IFAC Proc. Vol.*, vol. 45, no. 8, pp. 33–38, 2012.

[4] R. A. Russell, "Robotic location of underground chemical sources," *Robotica*, vol. 22, no. 1, pp. 109–115, 2004.

[5] G. Ferri, M. V. Jakuba, and D. R. Yoerger, "A novel method for hydrothermal vents prospecting using an autonomous underwater robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2008, pp. 1055–1060.

[6] J. A. Farrell, J. Murlis, X. Long, W. Li, and R. T. Cardé, "Filament-based atmospheric dispersion model to achieve short time-scale structure of odor plumes," *Environ. Fluid Mech.*, vol. 2, no. 1–2, pp. 143–169, 2002.

[7] H. Ishida, K.-I. Suetsugu, T. Nakamoto, and T. Moriizumi, "Study of autonomous mobile sensing system for localization of odor source using gas sensors and anemometric sensors," *Sensors Actuators A, Phys.*, vol. 45, no. 2, pp. 153–157, 1994.

[8] G. Sandini, G. Lucarini, and M. Varoli, "Gradient driven self-organizing systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 1993, vol. 1, pp. 429–432.

[9] F. W. Grasso, T. R. Consi, D. C. Mountain, and J. Atema, "Biomimetic robot lobster performs chemo-orientation in turbulence using a pair of spatially separated sensors: Progress and challenges," *Robot. Auton. Syst.*, vol. 30, no. 1–2, pp. 115–131, 2000.

[10] R. A. Russell, A. Bab-Hadiashar, R. L. Shepherd, and G. G. Wallace, "A comparison of reactive robot chemotaxis algorithms," *Robot. Auton. Syst.*, vol. 45, no. 2, pp. 83–97, 2003.

[11] A. Lilienthal and T. Duckett, "Experimental analysis of gas-sensitive Braitenberg vehicles," *Adv. Robot.*, vol. 18, no. 8, pp. 817–834, 2004.

[12] H. Ishida, G. Nakayama, T. Nakamoto, and T. Moriizumi, "Controlling a gas/odor plume-tracking robot based on transient responses of gas sensors," *IEEE Sensors J.*, vol. 5, no. 3, pp. 537–545, Jun. 2005.

[13] J. Murlis and C. Jones, "Fine-scale structure of odour plumes in relation to insect orientation to distant pheromone and other attractant sources," *Physiol. Entomol.*, vol. 6, no. 1, pp. 71–86, 1981.

[14] R. T. Cardé and A. Mafra-Neto, "Mechanisms of flight of male moths to pheromone," in *Insect Pheromone Research*. Berlin, Germany: Springer, 1997, pp. 275–290.

[15] R. Kanzaki, N. Sugi, and T. Shibuya, "Self-generated zigzag turning of Bombyx mori males during pheromone-mediated upwind walking (Physiology)," *Zool. Sci.*, vol. 9, no. 3, pp. 515–527, 1992.

[16] W. Li, J. A. Farrell, S. Pang, and R. M. Arrieta, "Moth-inspired chemical plume tracing on an autonomous underwater vehicle," *IEEE Trans. Robot.*, vol. 22, no. 2, pp. 292–307, Aug. 2006.

[17] J. A. Farrell, S. Pang, and W. Li, "Chemical plume tracing via an autonomous underwater vehicle," *IEEE J. Ocean. Eng.*, vol. 30, no. 2, pp. 428–442, Apr. 2005.

[18] S. Pang and J. A. Farrell, "Chemical plume source localization," *IEEE Trans. Syst., Man, Cybern., B*, vol. 36, no. 5, pp. 1068–1080, Oct. 2006.

[19] J.-G. Li, Q.-H. Meng, Y. Wang, and M. Zeng, "Odor source localization using a mobile robot in outdoor airflow environments with a particle filter algorithm," *Auton. Robots*, vol. 30, no. 3, pp. 281–292, 2011.

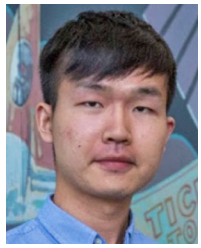
[20] J. A. Farrell, S. Pang, and W. Li, "Plume mapping via hidden Markov methods," *IEEE Trans. Syst., Man, Cybern., Part B*, vol. 33, no. 6, pp. 850–863, Dec. 2003.

[21] H. Jiu, Y. Chen, W. Deng, and S. Pang, "Underwater chemical plume tracing based on partially observable Markov decision process," *Int. J. Adv. Robot. Syst.*, vol. 16, no. 2, 2019. [Online]. Available: https://doi.org/10.1177/1729881419831874

[22] H.-F. Jiu, S. Pang, J.-L. Li, and B. Han, "Odor plume source localization with a pioneer 3 mobile robot in an indoor airflow environment," in *Proc. IEEE Southeastcon*, 2014, pp. 1–6.

[23] H. Hu, S. Song, and C. P. Chen, "Plume tracing via model-free reinforcement learning method," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 8, pp. 2515–2527, Aug. 2019.

[24] M. Vergassola, E. Villermaux, and B. I. Shraiman, "'Infotaxis' as a strategy for searching without gradients," *Nature*, vol. 445, no. 7126, 2007, Art. no. 406.
[25] S. Pang and F. Zhu, "Reactive planning for olfactory-based mobile robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2009, pp. 4375–4380.
[26] D. Silver *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, 2016, Art. no. 484.
[27] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.
[28] W. Naeem, R. Sutton, and J. Chudley, "Chemical plume tracing and odour source localisation by autonomous vehicles," *J. Navig.*, vol. 60, no. 2, pp. 173–190, 2007.
[29] F. Rahbar, A. Marjovi, and A. Martinoli, "An algorithm for odor source localization based on source term estimation," in *Proc. Int. Conf. Robot. Autom.*, 2019, pp. 973–979.
[30] B. Luo, Q.-H. Meng, J.-Y. Wang, and M. Zeng, "A flying odor compass to autonomously locate the gas source," *IEEE Trans. Instrum. Meas.*, vol. 67, no. 1, pp. 137–149, Jan. 2018.
[31] O. Sigaud and O. Buffet, *Markov Decision Processes in Artificial Intelligence*. New York, NY, USA: Wiley, 2013.
[32] M. L. Littman, "A tutorial on partially observable Markov decision processes," *J. Math. Psychol.*, vol. 53, no. 3, pp. 119–125, 2009.
[33] G. C. Sousa and B. K. Bose, "A fuzzy set theory based control of a phase-controlled converter dc machine drive," *IEEE Trans. Industry Appl.*, vol. 30, no. 1, pp. 34–44, Jan./Feb. 1994.
[34] M. Pratama, J. Lu, and G. Zhang, "Evolving type-2 fuzzy classifier," *IEEE Trans. Fuzzy Syst.*, vol. 24, no. 3, pp. 574–589, Jun. 2016.
[35] L. A. Zadeh, "Fuzzy sets," *Inf. Control*, vol. 8, no. 3, pp. 338–353, 1965.
[36] J. P. Crimaldi, M. B. Wiley, and J. R. Koseff, "The relationship between mean and instantaneous structure in turbulent passive scalar plumes," *J. Turbulence*, vol. 3, no. 14, pp. 1–24, 2002.
[37] J. Elkinton, R. Cardé, and C. Mason, "Evaluation of time-average dispersion models for estimating pheromone concentration in a deciduous forest," *J. Chem. Ecol.*, vol. 10, no. 7, pp. 1081–1108, 1984.
[38] R. T. Cardé and M. A. Willis, "Navigational strategies used by insects to find distant, wind-borne sources of odor," *J. Chem. Ecol.*, vol. 34, no. 7, pp. 854–866, 2008.
[39] S. Shigaki, T. Sakurai, N. Ando, D. Kurabayashi, and R. Kanzaki, "Time-varying moth-inspired algorithm for chemical plume tracing in turbulent environment," *IEEE Robot. Autom. Lett.*, vol. 3, no. 1, pp. 76–83, Jan. 2018.
[40] B. K. Bose, "Expert system, fuzzy logic, and neural network applications in power electronics and motion control," *Proc. IEEE*, vol. 82, no. 8, pp. 1303–1323, 1994.
[41] S. Ross, J. Pineau, S. Paquet, and B. Chaib-Draa, "Online planning algorithms for POMDPS," *J. Artif. Intell. Res.*, vol. 32, pp. 663–704, 2008.
[42] Q. Lu, Q.-L. Han, and S. Liu, "A cooperative control framework for a collective decision on movement behaviors of particles," *IEEE Trans. Evol. Comput.*, vol. 20, no. 6, pp. 859–873, Dec. 2016.

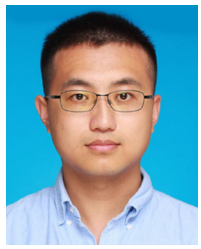


Lingxiao Wang (Student Member, IEEE) received the B.S. degree in electrical engineering from the Civil Aviation University of China, Tianjin, China, in 2015 and the M.S. degree in electrical and computer engineering from Embry-Riddle Aeronautical University, Daytona Beach, FL, USA, in 2017. He is currently working toward the Ph.D. degree with the Department of Electrical Engineering and Computer Science, Embry-Riddle Aeronautical University.

His current research interests include autonomous systems, olfactory-based navigation methods, and artificial intelligence.

Shuo Pang (Member, IEEE) received the B.S. degree in electrical engineering from Harbin Engineering University, Harbin, China in 1997, and the M.S. and Ph.D. degrees in electrical engineering from the University of California, Riverside, CA, USA in 2001 and 2004, respectively.

He is currently an Associate Professor with the Department of Electrical Engineering and Computer Science, Embry-Riddle Aeronautical University, Daytona Beach, FL, USA. His research interests include spans theoretical-algorithm development and application-driven intelligent systems. His current research interests also include embedded systems, robotics, and artificial intelligence techniques for autonomous vehicles, i.e., autonomous vehicle chemical plume tracing, autonomous vehicle online mapping, and planning.



Jinlong Li is currently working toward the Ph.D. degree in naval architecture and ocean engineering with Shanghai Jiao Tong University, Shanghai, China.

His current research interests include online mapping and planning in chemical plume tracing via an autonomous vehicle and computational fluid dynamics.