2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

An Implementation of the Adaptive Neuro-Fuzzy Inference System (ANFIS) for Odor Source Localization

Lingxiao Wang[1] and Shuo Pang[1]

2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 978-1-7281-6212-6/20/$31.00 ©2020 IEEE | DOI: 10.1109/IROS45743.2020.9341688

Abstract— **In this paper, we investigate the viability of implementing machine learning (ML) algorithms to solve the odor source localization (OSL) problem. The primary objective is to obtain an ML model that guides and navigates a mobile robot to find an odor source without explicating searching algorithms. To achieve this goal, the model of an adaptive neuro-fuzzy inference system (ANFIS) is employed to generate the olfactory-based navigation strategy. To train the ANFIS model, multiple training data sets are acquired by applying two traditional olfactory-based navigation methods, namely moth-inspired and Bayesian-inference methods, in hundreds of simulated OSL tests with different environments. After training with the hybrid-learning algorithm, the ANFIS model is validated in multiple OSL tests with varying searching conditions. Experiment results show that the ANFIS model can imitate other olfactory-based navigation methods and correctly locate the odor source. Besides, by training it with the fused training data set, the ANFIS model is better than two traditional navigation methods in terms of the averaged searching time.**

I. INTRODUCTION

Olfaction, as an essential sensing ability, is widely used by animals to perform life-essential activities, such as homing, foraging, mate-seeking, and evading predators. Inspired by olfactory capabilities of animals, a mobile robot or an autonomous vehicle, equipped with odor-detection sensors, could locate an odor 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 marine surveys such as locating underwater hydrothermal vents [5].

Designing an effective olfactory-based navigation algorithm is crucial for an OSL problem. Similar to 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. Estimating plume locations is the main challenge of this navigation problem since plume propagation is not only related to the molecular diffusion that takes plumes away from the odor source but also the advection of airflow [6].

The simplest olfactory-based navigation approach is the chemotaxis [7], which commands the robot to move along the gradient of odor plume concentrations. However, this method is impractical in a turbulent flow environment because odor plumes are congregated into packets and the gradient of concentration is a patchy and intermittent signal [8]. Alternatively, bio-inspired methods, which direct the robot to mimic animal behaviors, were proposed. For instance, a male moth could successfully locate a female moth by tracking pheromones over a long distance [9]. 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. [10] generalized the 'surge/casting' model and implemented it on a wheeled vehicle. Li et al. [11] implemented this method on an autonomous underwater vehicle (AUV) to search for an underwater chemical source. Water test results [12] proved the validity of this method.

Another type of olfactory-based navigation method is the engineering-based method. Unlike a bio-inspired method, an engineering-based method does not command the robot to follow a fixed behavior pattern. Instead, it utilizes math and physics approaches to model odor plume distribution and estimate possible odor source locations. Various methods have been proposed to predict odor source locations, such as Bayesian-inference [13], particle filter [14], hidden Markov model (HMM) [15], and partially observable Markov decision process (POMDP) [16]. After source estimations are obtained, a path planner is employed to guide the robot moving toward the target. Artificial potential field (APF) [17] is a feasible algorithm to design the path planner. Besides, Vergassola et al. presented the 'infotaxis' method [18], which uses information entropy to guide the robot searching for an odor source, and the robot was commanded to select the movement that mostly reduces the information uncertainty of the odor source.

Comparing existing olfactory-based navigation strategies, the limitation of bio-inspired methods is the lack of odor plume estimations. Thus, when odor plumes are absent, the robot can only perform a time-consuming 'casting' behavior to find plumes. What is worse is that this method usually fails in a turbulent flow environment since odor plume trajectories change rapidly in this scenario. As for an engineering-based method, the requirement for the high computational capacity to online estimate odor source locations restrains its applications when the search area is large and complex. A desired olfactory-based navigation algorithm should be light-weighted, i.e., it does not have a high computational demand, while efficient and capable enough to locate odor sources in different flow conditions.

Derived from this consideration, we leverage the OSL problem with machine learning (ML) algorithms. The princi-

[1] Lingxiao Wang and Shuo Pang are with the Electrical Engineering and Computer Science Department, Embry-Riddle Aeronautical University, Daytona Beach, FL, 32114. lingxiaw@my.erau.edu and shuo.pang@erau.edu

pal idea of ML technologies is letting the computer recognize the pattern of performing a task by relying on data sets without providing explicit instructions [19]. A wide spectrum of ML-related applications [20]–[22] has demonstrated the powerful ability of ML algorithms to estimate uncertainties. The motivation of applying ML algorithms to solve an OSL problem is that we want an ML model to learn an odor source searching strategy from benefits of other successful olfactory-based navigation methods without explicating the specific searching algorithms and rules. Besides, an ML algorithm is light-weighted, which is suitable for working with autonomous vehicles. It should be mentioned that the challenging part of this application is to obtain training data sets since OSL experiments are expensive to repeatedly perform in different airflow conditions.

In our method, the model of an adaptive neuro-fuzzy inference system (ANFIS) [23] is adopted to control the robot in the odor plume tracing process. Unlike the traditional fuzzy inference system (FIS), an ANFIS does not require expert knowledge to define FIS parameters but learns the optimal FIS parameters via a neural network with training data. During the odor plume tracing process, the ANFIS model takes onboard sensor measurements as inputs and produces robot heading commands that lead the robot toward the odor source. A simulation program that emulates time-varying airflow fields and corresponding plume trajectories is selected as the platform to generated training data and validate the trained ANFIS model. Considering the difficulty of repeatedly conducting the actual OSL experiments and the requirement of the large quantity training data, collecting training data from a realistic simulation program is an acceptable option. Specifically, two traditional olfactory-based navigation methods, namely moth-inspired [11] and Bayesian-inference [13] methods, are implemented in hundreds of OSL tests with the simulation program to generate training data. After the training, the ANFIS model is expected to learn an olfactory-based navigation method via the training data and is validated in multiple OSL tests with various environments to verify the validity of this method.

II. Methodology

A. Overview of Modeling the ANFIS for the OSL Problem

As shown in Fig. 1, the scheme of modeling the ANFIS for the OSL problem can be separated into two parts, namely offline procedures and online procedures. In offline procedures, an ANFIS model is created and trained with records generated from OSL tests of implementing the moth-inspired and Bayesian-inference olfactory-based navigation methods. The training process is considered as complete if the number of training epoch reaches the pre-defined value. In online procedures, trained ANFIS models are validated by searching the odor source in the simulation with various airflow fields and different searching conditions. The testing is considered as complete if the robot can correctly find the odor source (by supervising); otherwise, parameters of the ANFIS model are adjusted, and the ANFIS model is trained again. It should be mentioned that the concentration of this work is to investigate an effective olfactory-based navigation method in the plume tracing procedures. In the real application, to determine whether or not the robot reaches the actual odor source location, a vision system can be employed to identify the odor source if the robot is close to it.

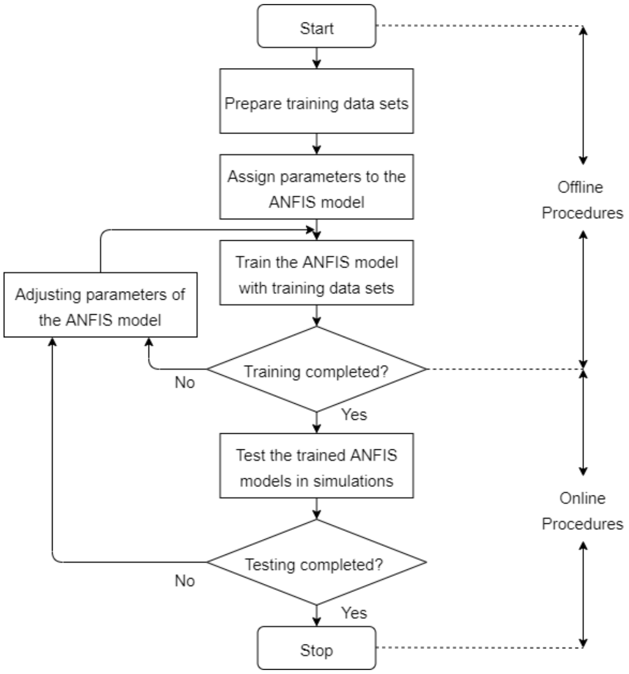

Start
Prepare training data sets
Assign parameters to the ANFIS model
Train the ANFIS model with training data sets
Adjusting parameters of the ANFIS model
Training completed?
No
Yes
Test the trained ANFIS models in simulations
Testing completed?
No
Yes
Stop
Offline Procedures
Online Procedures

Fig. 1. The framework of the proposed olfactory-based navigation method

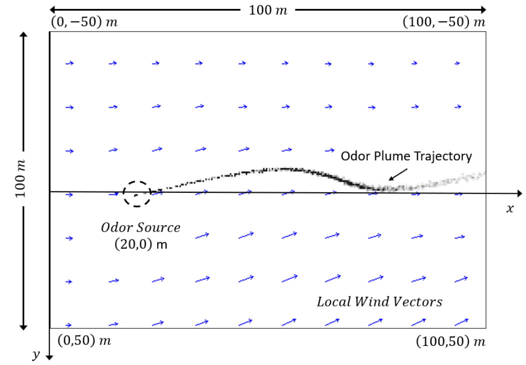


Fig. 2. The simulation environment

B. Prepare the Training Data Sets

1) The Simulated Searching Environment: In this work, the OSL is considered as a two-dimensional (2-D) problem since the aimed implementation robotic platform is a ground mobile robot. A realistic OSL simulation program [6] that emulates filament-based odor plume trajectories in time-varying airflow fields is employed as the platform to generate training data sets. Fig. 2 shows the simulated search area, where the size is 100×100 m^2. Over the search area, a coordinate system $(x - y)$ is constructed, and an odor source is located at $(20, 0)$ m and releases 10 plumes per second. Released plumes form a circular plume trajectory as

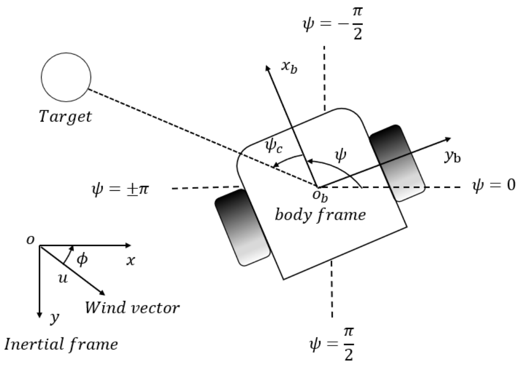


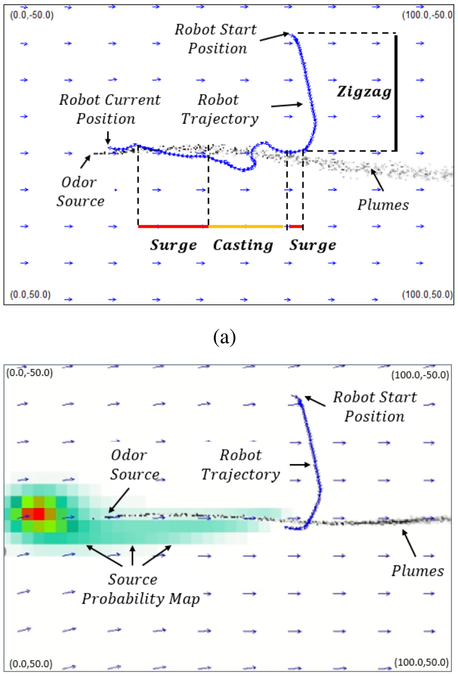
Fig. 3. The two-wheeled mobile robot used in the simulation program. Wind velocity u, wind direction ϕ, and robot positions (x, y) are measured from the onboard anemometer and the positioning sensor in the inertial frame. The robot heading ψ is monitored by the onboard compass. The heading command ψ_c is defined as the difference between the current heading and the target direction.

plotted by a grey-scale patchy trail. The shape and position of the plume trajectory are varying with local winds, which are indicated by arrows in the background. Wind vectors are calculated from time-varying boundary conditions that are generated by a mean flow ($\mathbf{U}_0 = (u_{0x}, u_{0y})$) plus Gaussian white noises (0 mean and ς variance). Varying airflow fields and the corresponding odor plume trajectories can be obtained by adjusting values of $\mathbf{U}_0$ and ς.

In the simulation program, a two-wheeled mobile robot, as shown in Fig. 3, is employed as the platform to implement the ANFIS model after the training. It is assumed that the robot is equipped with a chemical sensor, an anemometer, a positioning sensor, and a compass which measure odor concentrations ρ, wind speeds u and directions ϕ in the inertial frame, the robot position (x, y) in the inertial frame, and the robot heading ψ respectively. To control the mobile robot in a 2-D plane, only speed and heading commands are needed. To simplify the problem, the robot moves in a constant speed, i.e., 1 m/s, and the heading commands ψ_c are generated from the implementing olfactory-based navigation method. It should be mentioned that the range of heading commands is from $-\pi$ to π, which specifies the rotation direction (negative: anti-clockwise; positive: clockwise) and angle. 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.

2) Introduction of Two 'Instructors': Moth-inspired [11] and Bayesian-inference [13] olfactory-based navigation methods are selected as 'instructors' to generate training data sets, which are paradigms in bio-inspired and engineering-based methods, respectively.

The core idea of the moth-inspired method is to mimic the mate-seeking behavior of male moths, which can be summarized as a two-phase searching strategy: 'surge' and 'casting'. Specifically, the 'surge' searching phase is triggered when the robot detects odor plumes, where the robot stays inside plumes and moves upwind. When the robot is out of plumes, the 'casting' searching phase is activated, where the robot casts in circles to re-detect plumes. Fig. 4(a) demonstrates the robot searching trajectory generated by implementing this method. The robot first adopts a 'zigzag' searching trajectory to detect the existence of odor plumes in the search area, and after it detects plumes for the first time, the robot surges upwind until it is out of plumes. Then, the robot traverses the wind to recover plumes. The robot switches between 'surge' and 'casting' searching phases depending on whether or not the robot detecting plumes. Experiment results show that this method is valid in a laminar flow environment, where the plume trajectory is relatively stable and continuous.



(a)

(b)

Fig. 4. Searching trajectories generated by traditional olfactory-based navigation methods. (a) Moth-inspired method. Over the searching trajectory, different searching phases are highlighted with color bars. Black, red, and yellow represent 'zigzag', 'surge', and 'casting', respectively. (b) Bayesian-inference method. To visualize the source probability map, the searching area is painted with various colorful cells, where darker cells have higher probability of containing the odor source, and the lighter cells have less (red: highest, white: lowest).

For the Bayesian-inference method, it utilizes measured wind information and a Gaussian plume dispersion model to reversely deduce the odor source location. Fundamental procedures are twofold: source mapping and path planning. In the source mapping procedure, the search area is divided into multiple cells, and a Gaussian plume dispersion model is employed to calculate the probability of the robot detecting or not detecting plumes in an arbitrary cell. Then, a source probability map that estimates possible odor source locations over the search area is generated and iteratively updated with plume detection and non-detection events. In the path planning procedure, the robot first adopts a 'zigzag' search trajectory to detect the existence of odor plumes. After it senses odor plumes for the first time, a possible odor source location indicated by the source probability map, i.e., the cell with the highest probability of containing the odor source, is obtained. Then, a path planner is employed to guide the robot moving to the source estimations. Fig. 4(b) shows a snapshot of the source probability map that is generated after the robot detects odor plumes for the first time. Comparing to the moth-inspired method, experiment results indicate that the Bayesian-inference method carries a higher success rate of locating the odor source in a turbulent flow environment.

C. Adaptive Neuro-Fuzzy Inference System

Jang [23] proposed the structure of the ANFIS, which is a feed-forward network that combines neural networks and fuzzy logic theory to map the relationships between inputs and outputs. The innovation of this method is that it does not require expert knowledge to assign parameters of a fuzzy inference system, but utilizes neural network learning algorithms to tune parameters.

Fig. 5 shows a five-layer ANFIS architecture with two inputs (x and y) and one output (f). Each layer contains one of two types of nodes, namely adaptive and fixed nodes. Adaptive nodes, represented by squares, have adjustable parameters, and fixed nodes, denoted by circles, contain fixed parameters. To present the ANFIS architecture, two fuzzy IF-THEN rules based on the first-order Sugeno fuzzy inference system are considered:

Rule 1:

IF x is A_1 AND y is B_1, **THEN** $f_1 = p_1 x + q_1 y + r_1$,

Rule 2:

IF x is A_2 AND y is B_2, **THEN** $f_2 = p_2 x + q_2 y + r_2$,

where A_i and B_i are fuzzy sets ($i = 1, 2$ is the index of fuzzy rules); f_i are outputs of fuzzy rules; p_i, q_i, and r_i are adaptive parameters determined during the training process. Each layer in the ANFIS architecture is illustrated as below.

The first layer is the fuzzification layer, where all nodes are adaptive nodes. Outputs θ_j^1 of this layer are the fuzzy membership grades of inputs, which can be represented as:

$$\begin{aligned} \theta_j^1 &= \mu_{A_j}(x), \quad j = 1, 2, \\ \theta_j^1 &= \mu_{B_{j-2}}(y), \quad j = 3, 4, \end{aligned} \tag{1}$$

where j is the index of nodes in this layer; μ_{A_j} and $\mu_{B_{j-2}}$ are membership functions associated with fuzzy sets A_j and B_{j-2}, respectively. For the Gaussian membership function, μ_{A_j} and $\mu_{B_{j-2}}$ can be presented as:

$$\begin{aligned} \mu_{A_j}(x) &= e^{-\left((x-b_j)^2/2a_j^2\right)}, \quad j = 1, 2, \\ \mu_{B_{j-2}}(y) &= e^{-\left((y-b_j)^2/2a_j^2\right)}, \quad j = 3, 4 \end{aligned} \tag{2}$$

where a_j and b_j are adjustable parameters of the membership functions. Parameters in this layer are referred to as premise parameters.

The second layer uses the AND operator (presented by a circle node labeled π) to fuzzify the incoming signals. All nodes in this layer are fixed, and each node computes the strength of rules (i.e., firing strength). The output of this layer w_j is the product of the incoming signals:

$$\theta_j^2 = w_j = \mu_{A_j}(x)\,\mu_{B_j}(y), \quad j = 1, 2. \tag{3}$$

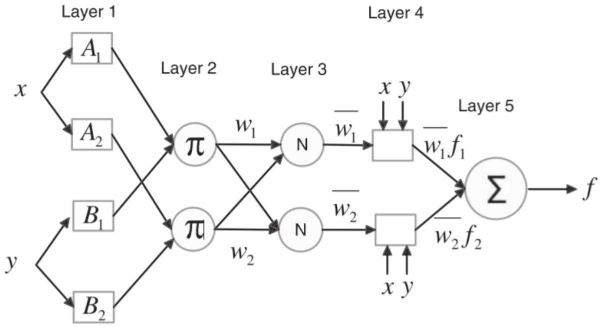


Fig. 5. The ANFIS architecture. Retrieved from [23].

The third layer is the normalization layer, where all nodes are fixed. The output of this layer $\bar{w}_j$ is the normalization of incoming signals:

$$\theta_j^3 = \bar{w}_j = \frac{w_j}{\sum_{k=1}^{2} w_k}, \quad j = 1, 2. \tag{4}$$

In the fourth layer, nodes are adaptive. The output of this layer is the the product of the normalized firing strength $\bar{w}_j$ and first order polynomials of inputs:

$$\theta_j^4 = \bar{w}_j f_j = \bar{w}_j \left(p_j x + q_j y + r_j\right), \quad j = 1, 2. \tag{5}$$

The last layer contains a single fixed node, which sums all incoming signals. The output of this node is represented by

$$\theta_j^5 = y = \sum_{k=1}^{2} \bar{w}_k f_k, \quad j = 1, 2. \tag{6}$$

The learning algorithm for ANFIS is the hybrid-learning algorithm, which is the combination of gradient descent and least squares methods. Each epoch of the hybrid learning algorithm is consisted of a forward pass and a backward pass. In the forward pass, premise parameters (i.e., b_j and a_j in the second layer) are kept constant and consequent parameters (i.e., p_j, q_j, and r_j in the fourth layer) are determined by the least squares method. In the backward pass, consequent parameters obtained from the previous step are kept constant and premise parameters are updated by the gradient descent method. The hybrid learning algorithm ensures a quicker convergence and avoids local minima issue comparing to the gradient descent method. The detailed description of the hybrid-learning algorithm can be found in [24].

D. Adapt the ANFIS Architecture for OSL Problems

In this work, the ANFIS model is selected to generate the searching strategy that controls the robot searching for the odor source. Steps to adapt the ANFIS model include: define input and output variables; define fuzzy sets for input variables; define fuzzy rules; create and train the ANFIS model.

1) Select Input and Output Variables: In an OSL problem, tens of variables are related with the robot searching behavior. For instance, in the moth-inspired method, sensed odor concentration and wind direction determine whether the robot is in the 'surge' or 'casting' searching phase, and in the

TABLE I

FUZZY SETS OF INPUTS TO THE ANFIS MODEL

Inputs	u_x	u_y	ρ	x	y	ψ
Fuzzy Sets	Slow, Fast	Slow, Fast	Low, High	Small, Large	Small, Large	Little, Big

Bayesian-inference method, robot positions and wind history records are essential to estimate odor source locations. Since these two methods are employed to generate training data in our method, all mentioned variables are included into inputs of the ANFIS model.

In summary, inputs of the ANFIS model are defined as wind speeds in x and y directions (u_x and u_y), sensed odor concentrations (ρ), robot positions (x and y), and heading measurements (ψ). As mentioned, the robot moving velocity is fixed in this work, thus, the output of the ANFIS model is set as the heading command (ψ_c, see Fig. 3). It is noted that wind speeds in x and y directions are calculated from wind speed (u) and direction (ϕ) measurements:

$$\begin{aligned} u_x &= u \times cos\phi, \\ u_y &= u \times sin\phi. \end{aligned} \tag{7}$$

The reason of using u_x and u_y as inputs instead u and ϕ is that the Bayesian-inference method uses u_x and u_y to compute wind advection distances on x and y directions.

2) Define Fuzzy Sets and Fuzzy Rules: To determine the number of fuzzy sets for each input, the complexity of the ANFIS model (i.e., the number of fuzzy rules) needs to be considered to prevent the overfitting (i.e., the complexity of the ANFIS model is greater than the size of training data) problem. Denote the number of inputs to the ANFIS model as I and the number of fuzzy sets of a single input is $M_i, i \in [1, I]$. Then, the total number of fuzzy rules N can be calculated as:

$$N = \prod_{i=1}^{I} M_i. \tag{8}$$

In this work, the number of inputs is determined as 6. If three fuzzy sets were defined for a single input, the total number of fuzzy rules would be 729, and if two fuzzy sets were defined for a single input, the number of fuzzy rules drops significantly to 64. It should be mentioned that increasing the number of membership functions per input does not necessarily improve the ANFIS model performance, but usually leads to model overfitting [25]. Thus, two fuzzy sets are defined for each input in the proposed ANFIS model, and the Gaussian function is selected as fuzzy membership functions of all fuzzy sets. Table I presents labels of fuzzy sets for each input.

For the first-order Sugeno fuzzy inference system, the consequence of a fuzzy rule f_1 can be expressed as:

Rule 1:

IF u_x is Slow AND u_y is Slow AND ρ is Low AND x is Small AND y is Small AND ψ is Little,

THEN $f_1 = \alpha_{1,1}u_x + \alpha_{2,1}u_y + \alpha_{3,1}\rho + \alpha_{4,1}x + \alpha_{5,1}y + \alpha_{6,1}\psi + \alpha_{7,1}$,

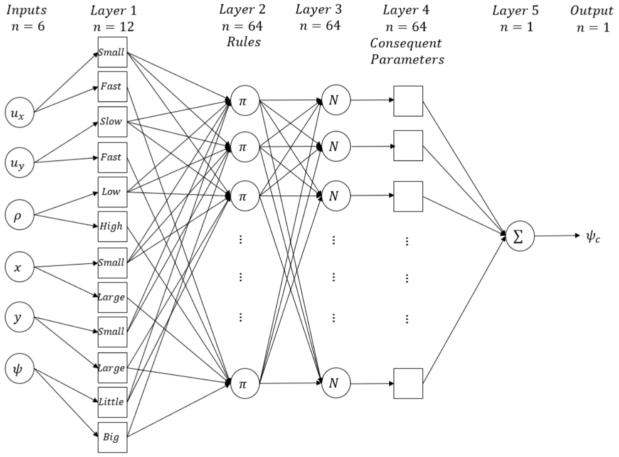


Fig. 6. The adapted ANFIS architecture for OSL problems. The number of nodes in a layer is indicated by n as shown at the top of the diagram.

where $\alpha_{i,j}, i \in [1, I], j \in [1, N]$ are the linear parameters in the consequent part of the Sugeno fuzzy inference system.

3) Create and Train the ANFIS Model: In this study, the MATLAB Fuzzy Logic Toolbox is used as the development tool to construct and train the ANFIS model. With the Fuzzy logic toolbox, an ANFIS model can be created and trained with few lines of codes, while training options such as epoch numbers, membership function types and numbers, training algorithms are adjustable.

After the training data set is loaded into the MATLAB $workspace$, the initial ANFIS model is created with the $genfis$ MATLAB function. This function finds the lower and upper bounds of input training data and creates membership functions to evenly cover the discourse of universe of each input. Once the initial ANFIS model is created, the training process begins. In MATLAB, the training command is $anfis$, which reads the training input/output data and generates an ANFIS model that maps inputs to the desired output. Specifically, all training data is fed to a neural network (see Fig. 5), in which input parameters are adjusted to minimize training errors. Root mean square error (RMSE) is the function that used to compute the training errors, which is defined as:

$$RMSE = \sqrt{\frac{1}{G}\sum_{j=1}^{G}\left(Y_j - \hat{Y}_j\right)^2}, \tag{9}$$

where G is the size of the training data; $\hat{Y}$ is the predicted output generated from ANFIS model; Y is the original output from the training data set, i.e., ψ_c. In summary, the proposed ANFIS architecture is presented in Fig 6.

III. EXPERIMENTS AND RESULTS

To verify the validity of the proposed method, various experiments were conducted. Multiple training data sets are acquired by implementing different olfactory-based navigation methods. As mentioned in the previous section, the moth-inspired and Bayesian-inference methods are selected

TABLE II
SPECIFICATIONS OF TRAINING DATA SETS

Training Data Set Name		Moth-inspired	Bayesian-inference	Fused
Dimensions (rows / columns)		7932 / 7	8891 / 7	16823 /7
ANFIS Model Name		MO-ANFIS	BA-ANFIS	FU-ANFIS
Discourse of Universe of Inputs	u_x, (m/s)	[0.12, 1.63]	[0.07, 1.13]	[0.07, 1.63]
	u_y, (m/s)	[-0.29, 0.19]	[-0.08, 0.19]	[-0.08, 0.19]
	ρ, (mmpv)	[0, 6.16]	[0, 3.37]	[0, 6.16]
	x, (m)	[15.63, 69.89]	[12.83, 69.87]	[12.83, 69.89]
	y, (m)	[-39.67, 10.86]	[-39.67, 10.8]	[-39.67, 10.86]
	ψ, (rad)	[-π, π]	[-π,π]	[-π,π]
Discourse of Universe of Output ψ_c, (rad)		[-π, 3.03]	[-2.69, 2.47]	[-π, 3.03]

mmpv: million molecules per cm^3

to generate training data sets. According to the type of the implementing algorithm, three training data sets (moth-inspired, Bayesian-inference, and Fused) and corresponding ANFIS models (MO-ANFIS, BA-ANFIS, and FU-ANFIS) were created. Note that, architectures of three ANFIS models are the same, but their parameters after training with the corresponding training set are varying. Table II shows specifications of training data sets.

A. Train ANFIS Models with Training Data Sets

Fig. 7 shows plots of RMSE after training ANFIS models with different training data sets. It can be seen that all RMSE plots converge to a constant after around 100 epochs. Specifically, the RMSE of training with the moth-inspired training data set is 0.572 after 100 epochs and slightly drops to 0.571 after 500 epochs. Similar RMSE values can be found for the Bayesian-inference training data set: 0.489 after 100 epochs and 0.488 after 500 epochs. As for the fused training data set, the RMSE is 0.523 when training epoch is 100 and vaguely decreases to 0.522 after 500 epochs. In general, for three training data sets, values of RMSE decrease with the increase of epoch numbers, but after 100 epochs, the change of RMSE is minor. After the training process is complete, trained ANFIS models (MO-ANFIS, BA-ANFIS, and FU-ANFIS) obtained after 500 training epochs are selected and deployed in the simulation with environmental settings $\mathbf{U}_0 = (0.4, 0)$ m/s and $\varsigma = 5$ to verify the validity of the proposed ANFIS method in an OSL problem.

Fig. 8 presents searching trajectories generated from implementing three ANFIS models in OSL tests. In these tests, the robot starts at the same position $(60, -40)$ m and adopts a 'zigzag' searching strategy to detect the existence of odor plumes. After the robot detects plumes for the first time, the ANFIS model is activated and controls the robot to search the odor source. As shown in Fig. 8, all ANFIS models can correctly locate the odor source in this environment, which demonstrates the capability of ANFIS models to learn other olfactory-based navigation methods.

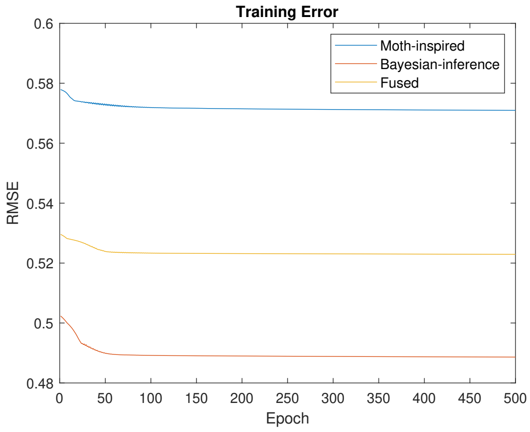


Fig. 7. Plots of RMSE in each epoch with different training sets

Specifically, it can be seen in Fig. 8(a) that the robot searching trajectory of MO-ANFIS is resembling to the moth-inspired olfactory-based navigation method. Boundaries between 'surge' and 'casting' searching phases are distinct, and these two phases are triggered with the plume detection and non-detection events, which is the identical searching logic to the moth-inspired method. The searching trajectory of BA-ANFIS is presented in Fig. 8(b). The robot acts like it could estimate the odor source location: after $t = 70$ s, the robot moves in parallel to the plume trajectory, and when it is close to the odor source, the robot correctly turns to the source at $t = 98$. The searching mechanism presented by BA-ANFIS is reminiscent to the Bayesian-inference olfactory-based navigation method, which estimates and surges toward the possible odor source location. For the ANFIS model trained by fused data set, i.e., FU-ANFIS, the searching trajectory is presented in Fig. 8(c). In this diagram, the presented searching trajectory is also similar to the moth-inspired method, but the robot moves in a less oscillating way comparing to the MO-ANFIS trajectory. Comparing to the BA-ANFIS, the robot with FU-ANFIS moves less aggressively in the upwind direction and takes a slightly longer time (i.e., 100 s) to find the odor source.

Comparing searching results of three ANFIS models, the performance of the FU-ANFIS model is better than the others. Although the searching time of FU-ANFIS is marginally longer than BA-ANFIS, the searching trajectory of the FU-ANFIS model is more effective. It is because source estimations based on Bayesian-inference method usually are not reliable before the robot gains enough information about the odor source. It can be seen that the robot can always detect plumes by following the FU-ANFIS trajectory, which is an essential behavior for the robot to acquire odor source information. The robustness of the FU-ANFIS model in OSL tests is further investigated in the next section.

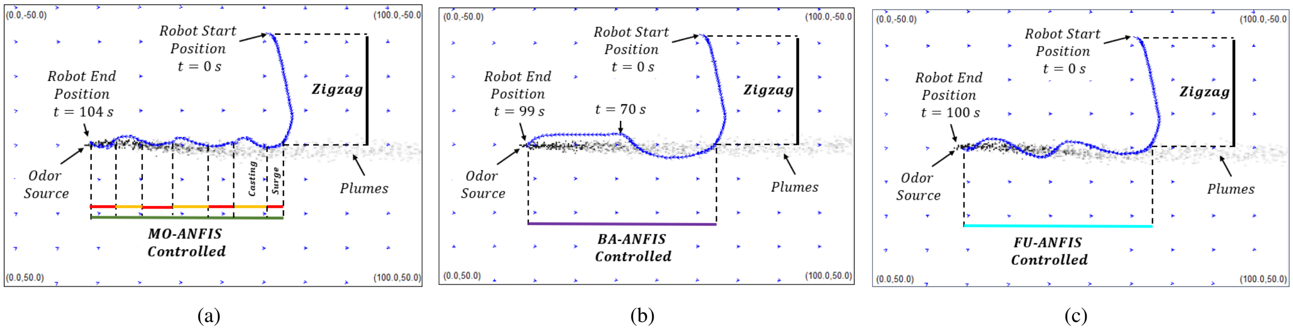


Fig. 8. Search trajectories of implementing different ANFIS models in the simulation with environmental settings $\mathbf{U}_0 = (0.4, 0)$ m/s and $\varsigma = 5$. Search phases are indicated by different colors, where black represents the 'zigzag' search phase; green, purple, and cyan indicate MO-ANFIS, BA-ANFIS, and FU-ANFIS controlled phases, respectively; red and yellow represent 'surge' and 'casting' searching phases, respectively. (a) MO-ANFIS. (b) BA-ANFIS. (c) FU-ANFIS.

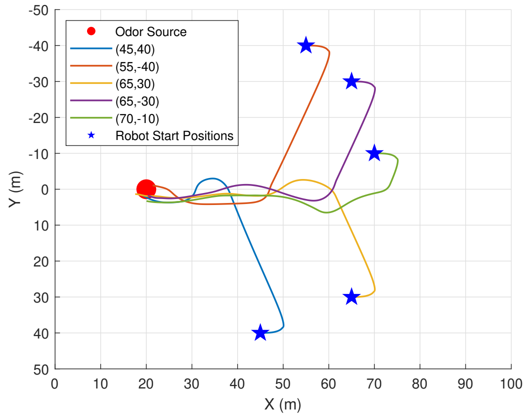


Fig. 9. Searching trajectories generated from different initial robot positions with the FU-ANFIS model. The initial start position of each trail is indicated in the diagram legend.

B. Test ANFIS Models with Various Searching Conditions

Experiment results of implementing the FU-ANFIS with different searching conditions, including varying initial searching positions and environmental settings, are presented in this section.

1) Varying Robot Starting Positions: In this group of tests, the robot starts the OSL task at different initial positions with the same environmental settings $\mathbf{U}_0 = (0.4, 0)$ m/s and $\varsigma = 5$. Fig. 9 shows the searching trajectories of implementing the FU-ANFIS model with varying initial robot positions. As shown in the diagram, all trails terminate at the actual odor source location, i.e., the robot correctly locate the odor source location, which indicates the validity of the proposed ANFIS model with varying initial start positions.

2) Varying Environmental Settings: To evaluate the performance of the FU-ANFIS model, the moth-inspired and Bayesian-inference olfactory-based navigation methods are also implemented and compared. Each navigation method is implemented in five different environments, and the corresponding searching time of each method is presented in Table III.

TABLE III
ENVIRONMENTAL SETTINGS AND SEARCHING TIME OF DIFFERENT NAVIGATION METHODS

Test	Mean Wind Velocity $\mathbf{U}_0$ (m/s)	Gaussian Noise Variance ς	FU-ANFIS Model-Based Method (s)	Moth-Inspired Method (s)	Bayesian-Inference Method (s)
Test 1	[0.1, 0]	3	92	93	132
Test 2	[1,0]	3	96	95	121
Test 3	[0.8, 0.1]	3	99	93	139
Test 4	[1.5, 0.1]	3	95	141	153
Test 5	[2, 0.3]	3	Failed	Failed	137

Performances of FU-ANFIS and moth-inspired methods are similar in terms of the searching time in Test 1, 2, and 3, where the robot is in laminar flow environments. In a more turbulent environment, e.g., Test 4, the moth-inspired method requires a longer searching time (141 s) than the FU-ANFIS method (95 s). Both FU-ANFIS and moth-inspired methods failed to locate the odor source in a highly turbulent environment (Test 5). The averaged searching time of FU-ANFIS method for Test 1-4 is 95.5 s. Comparing to moth-inspired method, which achieves the averaged searching time as 105.5 s for same tests, the FU-ANFIS method is more efficient. Comparing to the Bayesian-inference method, it can be concluded that FU-ANFIS method is more efficient in laminar and slightly turbulent environments (Test 1-4), but the Bayesian-inference method is more effective in highly turbulent environments.

C. Discussions

Although experiment results of implementing the ANFIS model in OSL tests are satisfied, we should also see the limitation of this method: the performance of the trained ANFIS model is highly depending on the quality and quantity of the training data set. To obtain a versatile ANFIS model that can succeed to locate odor sources in more extreme environments (e.g., Test 5) needs more training data sets. A preferred training data set should cover more searching situations, such as searching in various environments (both laminar and turbulent flow fields) and different initial positions. In addition, besides moth-inspired and Bayesian-inference olfactory-based methods, more odor searching strategies could be considered to generate training data sets.

IV. Conclusions

In this work, we investigate the possibility of implementing machine learning algorithms to solve OSL problems. The ANFIS model is trained with traditional olfactory-based navigation methods, and the trained ANFIS models are tested in various environments. Experiment results show that the proposed ANFIS models are feasible in locating odor source with different searching conditions. Besides, to improve the searching efficiency of the ANFIS model, more training data that covers versatile searching situations is required.

References

[1] G. Kowadlo and R. A. Russell, "Robot odor localization: a taxonomy and survey," *The International Journal of Robotics Research*, vol. 27, no. 8, pp. 869–894, 2008.

[2] M. Dunbabin and L. Marques, "Robots for environmental monitoring: Significant advancements and applications," *IEEE Robotics & Automation Magazine*, vol. 19, no. 1, pp. 24–39, 2012.

[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 Proceedings Volumes*, 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 *2008 IEEE International Conference on Robotics and Automation*. IEEE, 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," *Environmental fluid mechanics*, 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 and Actuators A: Physical*, vol. 45, no. 2, pp. 153–157, 1994.

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

[9] R. T. Cardé and A. Mafra-Neto, "Mechanisms of flight of male moths to pheromone," in *Insect pheromone research*. Springer, 1997, pp. 275–290.

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

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

[12] J. A. Farrell, S. Pang, and W. Li, "Chemical plume tracing via an autonomous underwater vehicle," *IEEE Journal of Oceanic Engineering*, vol. 30, no. 2, pp. 428–442, 2005.

[13] S. Pang and J. A. Farrell, "Chemical plume source localization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 36, no. 5, pp. 1068–1080, 2006.

[14] 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," *Autonomous Robots*, vol. 30, no. 3, pp. 281–292, 2011.

[15] J. A. Farrell, S. Pang, and W. Li, "Plume mapping via hidden Markov methods," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 33, no. 6, pp. 850–863, 2003.

[16] Z. A. Saigol, R. W. Dearden, J. L. Wyatt, and B. J. Murton, "Information-lookahead planning for auv mapping," in *Twenty-First International Joint Conference on Artificial Intelligence*, 2009.

[17] S. Pang and 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.

[18] M. Vergassola, E. Villermaux, and B. I. Shraiman, "'infotaxis' as a strategy for searching without gradients," *Nature*, vol. 445, no. 7126, p. 406, 2007.

[19] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[20] G. Cheng and J. Han, "A survey on object detection in optical remote sensing images," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 117, pp. 11–28, 2016.

[21] K. Kourou, T. P. Exarchos, K. P. Exarchos, M. V. Karamouzis, and D. I. Fotiadis, "Machine learning applications in cancer prognosis and prediction," *Computational and structural biotechnology journal*, vol. 13, pp. 8–17, 2015.

[22] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on intelligent vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

[23] J.-S. Jang, "Anfis: adaptive-network-based fuzzy inference system," *IEEE transactions on systems, man, and cybernetics*, vol. 23, no. 3, pp. 665–685, 1993.

[24] J.-S. Jang and C.-T. Sun, "Neuro-fuzzy modeling and control," *Proceedings of the IEEE*, vol. 83, no. 3, pp. 378–406, 1995.

[25] A. Al-Hmouz, J. Shen, R. Al-Hmouz, and J. Yan, "Modeling and simulation of an adaptive neuro-fuzzy inference system (anfis) for mobile learning," *IEEE Transactions on Learning Technologies*, vol. 5, no. 3, pp. 226–237, 2011.