3-Dimensional Gas Source Localization via Chemotaxis using Uncrewed Aircraft Systems

Jiahao Li
Dept. of Computer Science
Louisiana Tech University
Ruston, Louisiana, USA
jli018@latech.edu

Lingxiao Wang
Dept. of Electrical Engineering
Louisiana Tech University
Ruston, Louisiana, USA
lwang@latech.edu

***Abstract*—Gas source localization (GSL) enables robotic agents to trace gas plumes and identify gas source locations. This paper presents a three-dimensional (3D) GSL navigation algorithm using an uncrewed aircraft system (UAS) to autonomously locate an active gas source. Two search modes, SCAN and SEARCH, are designed to guide the UAS motions. In the SCAN mode, the UAS performs a multi-layer lawn-mower trajectory to explore the environment and to detect initial plume signals. Once high gas concentrations are detected, the system switches to the SEARCH mode, in which the UAS follows local concentration gradients to approach the gas source location. The proposed approach is evaluated using a 3D Gaussian plume model and a quadrotor UAS simulator. Simulation results demonstrate that the proposed GSL navigation algorithm can reliably guide the UAS to the gas source location with varying gas source and initial UAS locations.**

***Index Terms*—gas source localization, uncrewed aircraft systems, chemotaxis.**

I. INTRODUCTION

Gas source localization (GSL) enables robotic agents to autonomously locate gas sources in unknown environments [1]–[4]. The core challenge in GSL lies in designing effective navigation algorithms that guide a robot to react to gas plume measurements and progressively approach the source location. By employing a robotic agent, GSL can replace human operators in hazardous environments to reduce safety risks and operational costs. Practical GSL applications include wildfire localization [5], smoke monitoring [6], [7], air quality monitoring [8], and hydrothermal vent localization [9].

Existing GSL navigation algorithms can be categorized into three classes: (i) chemotaxis-based methods, (ii) bio-inspired methods, and (iii) engineering-based (or probabilistic) methods. Chemotaxis-based methods [10] guide robots by following gas concentration gradients, directing the robot moving toward regions with higher gas concentrations. Bio-inspired methods mimic animal plume tracing strategies. A classic example is the moth-inspired navigation method [11], where the robot moves upwind with gas plume detection and performs crosswind casting when the plume detection is lost. Engineering-based methods rely on mathematical or physical models of gas dispersion to estimate possible source locations. Representative approaches include Gaussian inference [12], particle filter [13], source term estimation [14], and partially observable Markov decision processes (POMDPs) [15], [16].

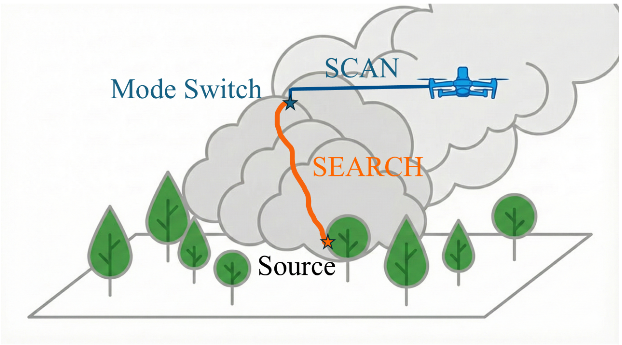


Fig. 1. Overview of the proposed chemotaxis-based GSL navigation algorithm for using on a UAS. The proposed framework consists of two search modes: SCAN and SEARCH. In the SCAN mode, the UAS follows a predefined lawn-mower trajectory to detect initial gas plume signals. Once high gas concentrations are detected, the system switches to the SEARCH mode, in w hich the UAS follows the estimated gas concentration gradient to approach the gas source.

While most existing GSL algorithms focus on ground robots in 2D environments [17], [18], and prior 3D aerial approaches often rely on computationally expensive probabilistic models [19], this work targets a lightweight and efficient strategy for 3D space. Compared to these previous approaches, the proposed "Scan-then-Search" strategy offers a computationally efficient solution for 3D GSL algorithm, as shown in Fig. 1. It combines SCAN mode, which performs structured multi-layer scanning, and SEARCH mode, which uses local gradient estimation to efficiently localize the source, making the method suitable for real-time UAS operations.

In the SCAN mode, the UAS executes a lawn-mower trajectory at multiple altitudes to explore the 3D search space and detect initial plume signals. Upon detecting high gas concentrations, the system switches to the SEARCH mode, where the UAS motion is determined by a chemotaxis-based method. Specifically, gas concentrations measured by the UAS are collected to calculate the concentrate gradient, which is used to direct the UAS toward higher concentration regions. In experiments, the proposed approach is validated in a simulated gas plume environment generated using a Gaussian plume model. Simulation results demonstrate that the proposed navigation strategy can effectively guide the UAS to the gas source under varying source locations and initial UAS positions.

II. METHODOLOGY

A. The Overall Workflow

Algorithm 1 summarizes the proposed navigation strategy. At each time step within the time limit $T_{\max}$, the system queries the UAS state $\boldsymbol{\xi}_t$, including position $\mathbf{p}$, velocity $\mathbf{v}$, orientation $\mathbf{q}$, and angular velocity $\boldsymbol{\omega}$. A gas concentration measurement c is obtained at the current UAS location. Based on c, $\mathbf{p}$, and $\mathbf{v}$, the proposed navigation module (NAVIGATIONCONTROLLER) computes the next target way point $\mathbf{p}_{\text{next}}$, which is passed to the low-level controller (DRONECONTROL) to generate motor commands that steer the UAS toward the target. This process continues until the UAS reaches the source vicinity, defined by a distance threshold d_{tol}, or when the UAS moves out of the predefined search space.

Algorithm 1 The Overall Workflow

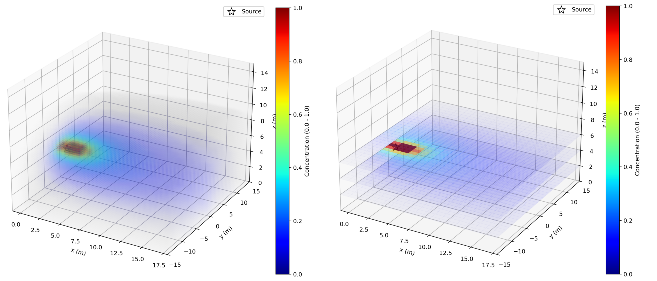
```
Require: Simulation Environment $env$, Max Simulation Time $T_{max}$, Time Step $\Delta t$, UAS State $\boldsymbol{\xi}_t$, Gas Concentration $c$, UAS Position $\mathbf{p}$, UAS Velocity $\mathbf{v}$, Gas Source Position $\mathbf{p}_{\text{source}}$, Source Declaration Distance $d_{\text{tol}}$.
Initialize: $t \leftarrow 0$
while $t < T_{max}$ do
    // 1. Get UAS State
    $\boldsymbol{\xi}_t \leftarrow$ GETDRONESTATUS($t$)
    $\mathbf{p} \leftarrow \boldsymbol{\xi}_t[0:3]$ // Get UAS position
    $\mathbf{v} \leftarrow \boldsymbol{\xi}_t[3:6]$ // Get UAS velocity
    // 2. Get Gas Concentration
    $c \leftarrow$ SENSORREADING($\mathbf{p}, t$)
    // 3. The Proposed GSL Navigation
    $\mathbf{p}_{\text{next}} \leftarrow$ NAVIGATIONCONTROLLER($c, \mathbf{v}, \mathbf{p}$)
    $u_{\text{motor}} \leftarrow$ DRONECONTROL($\mathbf{p}_{\text{next}}, \boldsymbol{\xi}_t, t$)
    $env$.STEP($u_{\text{motor}}$)
    // 4. Check Termination
    if $\|\mathbf{p} - \mathbf{p}_{\text{source}}\|_2 < d_{\text{tol}}$ then
        break // Success: Source located
    else if OUTOFBOUND($\mathbf{p}$) then
        break // Failure: Out Of Bound
    end if
    $t \leftarrow t + \Delta t$
end while
```

The following sections describe the gas plume model, the UAS simulator, and the proposed GSL navigation algorithm.

B. Gas Plume Model

The gas plume distribution is simulated using a Gaussian gas plume model [20]. In this framework, discrete gas puffs are released from the source at a constant rate and transported downwind by the ambient wind. For a location (x, y, z) at time t, the gas concentration c is computed as the sum of contributions from all active puffs in a set S_t:

$$c(x, y, z, t) = \sum_{k \in S_t} c_k(x, y, z, t), \tag{1}$$



(a) (b)

Fig. 2. Simulated gas plume field. (a) the 3D gas distribution within the search space; (b) Gas distribution slices at $z = \{2, 4, 6\}$ m.

where k denotes the index of an individual puff. The concentration contribution of the k-th puff, i.e., c_k, is given by:

$$\begin{aligned} c_k(x, y, z, t) = {} & \frac{m}{(2\pi)^{3/2}\sigma_y^2\sigma_z} \times \\ & \exp\left(-\frac{(x - x_k)^2 + (y - y_k)^2}{2\sigma_y^2}\right) \times \\ & \left[\exp\left(-\frac{(z - z_k)^2}{2\sigma_z^2}\right) + \exp\left(-\frac{(z + z_k)^2}{2\sigma_z^2}\right)\right], \end{aligned} \tag{2}$$

where m is the mass released per puff, (x_k, y_k, z_k) denotes the k-th puff position, and σ_y and σ_z are dispersion coefficients in the y and z directions, respectively.

Following [20], the puff advection is governed by winds. In this work, the wind is assumed to blow along the positive x-axis with the constant speed u. Each puff is initialized at the gas source location, and the puff position is updated as

$$\begin{aligned} x_k &\leftarrow x_k + u\Delta t, \\ y_k &\leftarrow y_k, \\ z_k &\leftarrow z_k + w_{\text{buoy}}\Delta t, \end{aligned} \tag{3}$$

where Δt is the simulation time step and w_{buoy} denotes the buoyancy velocity.

Fig. 2(a) presents the simulated gas distribution within a 3D search space. In this environment, the gas source is located at $(0, 0, 4)$ m. The wind speed u is set as a constant of 2 m/s, and the wind direction is in the positive x-direction. Fig. 2(b) shows the horizontal cross sections of the gas concentration field at different altitudes, with $z \in \{2, 4, 6\}$ m.

C. UAS Kinematics Simulation and Sensor Reading

In this work, `RotorPy` [21] is employed as the UAS kinematics simulator. At each simulation time step t, the simulator outputs the UAS state (i.e., GETDRONESTATUS) as

$$\boldsymbol{\xi}_t = [\mathbf{p}, \mathbf{v}, \mathbf{q}, \boldsymbol{\omega}]_t, \tag{4}$$

where $\mathbf{p} \in \mathbb{R}^{3\times1}$ denotes the position, $\mathbf{v} \in \mathbb{R}^{3\times1}$ is the linear velocity, $\mathbf{q} \in \mathbb{R}^{4\times1}$ represents the orientation in quaternion form, and $\boldsymbol{\omega} \in \mathbb{R}^{3\times1}$ is the angular velocity. `RotorPy` also provides a position-level control interface (i.e., DRONECONTROL): given a desired target way point, the current UAS state, and the simulation time t, the controller computes motor speed commands that drive the UAS toward the target.

The sensor reading module (i.e., SENSORREADING) takes the current UAS position and simulation time as inputs and computes the gas concentration at the UAS location using Eqn. (1). This calculation will consider all active puffs generated so far to emulate the gas sensor measurement at the UAS location.

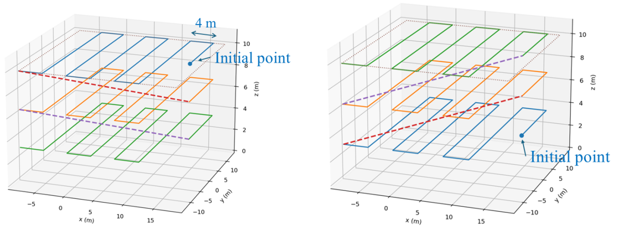
D. The Proposed GSL Navigation Algorithm

The proposed GSL navigation algorithm governs UAS flight through two operating modes, including the SCAN and SEARCH modes. The navigation initializes in the SCAN mode, in which the UAS follows a lawn-mower trajectory across multiple altitude layers to explore the search space. When the measured gas concentration exceeds a predefined threshold, the UAS permanently switches to the SEARCH mode. In this mode, recent gas measurements are stored in a buffer and used to estimate the local gas concentration gradient via the least-squares method [22], which determines the flight direction toward increasing concentration. Algorithm 2 summarizes the proposed GSL navigation framework.

Algorithm 2 NavigationController Algorithm

```
Require: Gas Concentration $c$, Concentration Threshold $c_{thresh}$, Navigation
    Mode $M$, Sliding Window Buffer $\mathcal{D}$, Buffer Size $N$, UAS Position $\mathbf{p}$,
    UAS Velocity $\mathbf{v}$.
Initialize: $M \leftarrow SCAN$
if $c > c_{thresh}$ and $M = SCAN$ then
    $M \leftarrow SEARCH$    // Permanently Switch to SEARCH mode
end if
if $M = SCAN$ then
    // SCAN Mode
    // Check if a layer is complete and switch height
    if IsLayerComplete($\mathbf{p}$) then
        $\mathbf{p}_{next} \leftarrow$ GetNextLayer($\mathbf{p}$)
    else
        $\mathbf{p}_{next} \leftarrow$ NextLawnmowerPoint($\mathbf{p}$)
    end if
else
    // SEARCH Mode
    $\mathcal{D}$.append($\mathbf{p}, c$)
    if $\mathcal{D}$.size $< N$ then
        // Wait for buffer to fill
        $\mathbf{p}_{\text{next}} \leftarrow \mathbf{p} + \mathbf{v}$
    else
        // Least Squares
        $\hat{\mathbf{g}} \leftarrow$ LeastSquares($\mathcal{D}$)
        // Normalize
        $\hat{\mathbf{d}} \leftarrow \hat{\mathbf{g}} / \|\hat{\mathbf{g}}\|_2$
        $\mathbf{p}_{\text{next}} \leftarrow \mathbf{p} + \hat{\mathbf{d}}$
    end if
end if
return $\mathbf{p}_{\text{next}}$ // Output Next way point
```

1) SCAN Mode: The SCAN mode is designed to survey the 3D search space and detect the presence of gas plumes. In this mode, the UAS executes a lawn-mower pattern on horizontal planes at three distinct altitude layers, as shown in Fig.3. The selection of these altitude layers depends on the initial height of the UAS z_0 relative to a threshold of 6 m (i.e., mid-height): if the UAS starts in the upper airspace ($z_0 > 6$ m), it follows a descending sequence, stepping downward from z_0 in two equal increments to cover the lower region. In contrast, if the UAS starts in the lower airspace ($z_0 \leq 6$ m), it follows an ascending sequence, moving upward from z_0 toward the operational ceiling of 11 m and partitioning the remaining vertical span into two additional layers.



(a) (b)

Fig. 3. The lawn-mower scanning trajectory for the SCAN mode: (a) from high to low layers; (b) from low to high layers.

At each altitude layer, the lawn-mower path is oriented such that its longer sweep direction aligns with the y-axis. The lateral spacing between adjacent sweeps is fixed at 4 m. A horizontal scan is considered complete when the UAS reaches the left boundary of the search region. Upon completing a scan at the current layer, the UAS returns to the initial (x, y) position, transitions vertically to the next altitude layer, and repeats the lawn-mower pattern. This process continues across all three layers unless the measured gas concentration c exceeds the threshold c_{thresh}, at which point the UAS permanently transitions to the SEARCH mode.

2) SEARCH Mode: In SEARCH mode, we use a chemotaxis-based method to direct the UAS motion based on the local gas concentration gradient. To estimate this gradient, a sliding window buffer $\mathcal{D}$ of size N is maintained to store the most recent gas concentration measurements and their corresponding UAS positions.

If the number of samples in $\mathcal{D}$ is fewer than N, the UAS continues its motion along the current velocity direction:

$$\mathbf{p}_{\text{next}} \leftarrow \mathbf{p} + \mathbf{v} \tag{5}$$

where $\mathbf{p}_{\text{next}}$ denotes the next target way point. This process continues until sufficient samples are collected to estimate the concentration gradient.

Once the buffer stores enough gas concentrations, we estimate the gas concentration field via a linear model:

$$(\mathbf{p}_i - \bar{\mathbf{p}})^T \mathbf{g} = (c_i - \bar{c}), \tag{6}$$

where c_i is the gas concentration measured at position $\mathbf{p}_i$, $\bar{c}$ and $\bar{\mathbf{p}}$ denote the mean concentration and mean position over the buffer $\mathcal{D}$, respectively, and $\mathbf{g} \in \mathbb{R}^{3\times1}$ represents the local concentration gradient.

To solve for $\mathbf{g}$, the position offsets and concentration deviations of all buffered samples are stacked to form $A = [(\mathbf{p}_i - \bar{\mathbf{p}})^T]_{i=1}^N$ and $\mathbf{b} = [(c_i - \bar{c})]_{i=1}^N$. Then, Eqn. (6) can then be expressed as $A\mathbf{g} = \mathbf{b}$, and the gradient is estimated via a least-squares solution [22]:

$$\hat{\mathbf{g}} = (A^T A)^{-1} A^T \mathbf{b}. \tag{7}$$

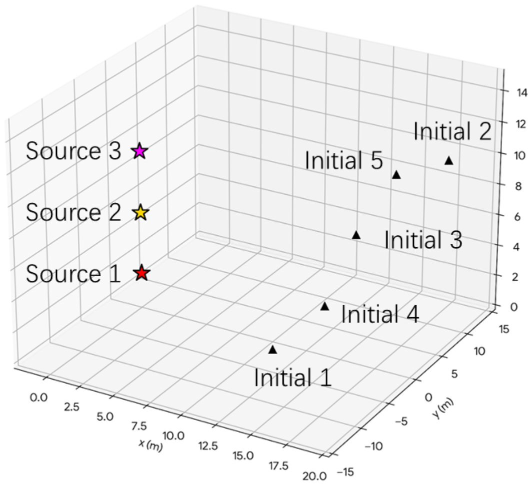


Fig. 4. Demonstration of varying source locations and initial UAS locations. Three possible source locations are defined at (0, 0, 2) m, (0, 0, 6) m, and (0, 0, 10) m, denoted as Source 1, 2, and 3, respectively. Five initial UAS locations are defined as (15, -12, 4) m, (18, 12, 10) m, (16, 0, 8) m, (17, -8, 6) m, and (15, 10, 9) m, denoted as Initial 1, 2, 3, 4, and 5, respectively.

The estimated gradient vector $\hat{\mathbf{g}}$ points in the direction of the steepest local increase in gas concentration. The next target way point is then computed as

$$\mathbf{p}_{\text{next}} = \mathbf{p} + \frac{\hat{\mathbf{g}}}{\|\hat{\mathbf{g}}\|_2}. \tag{8}$$

Once the navigation module computes the next target waypoint, it is passed to the low-level controller to generate motor commands that drive the UAS toward the target.

III. EXPERIMENTS AND RESULTS

A. *Experiment Design*

In our simulation, the maximum search duration is set as $T_{\max} = 600$ s. The UAS navigates within a defined 3D boundary of $x \in [-10, 20]$ m, $y \in [-15, 15]$ m, and $z \in [1, 11]$ m. The gas environment simulates a continuous plume carried by a constant horizontal wind field of 2 m/s and a vertical buoyancy velocity of $w_{\text{buoy}} = 0.15$ m/s. The concentration threshold for switching from the SCAN mode to the SEARCH mode is set as $c_{thresh} = 0.05$. For the gradient estimation, the buffer size is set as $N = 20$ samples. The source declaration distance is set as $d_{tol} = 2$ m.

Fig. 4 illustrates the layout of the gas source locations and UAS initial positions. Three gas source locations are considered: Source 1 at $(0, 0, 2)$ m, Source 2 at $(0, 0, 6)$ m, and Source 3 at $(0, 0, 10)$ m. Five different UAS initial locations are evaluated: $(15, -12, 4)$ m, $(18, 12, 10)$ m, $(16, 0, 8)$ m, $(17, -8, 6)$ m, and $(15, 10, 9)$ m. In each experiment, one gas source location and one UAS initial location are selected, resulting in a total of 15 test cases.

TABLE I
FLIGHT TIME FOR DIFFERENT INITIALIZATION POINTS ACROSS DIFFERENT GAS SOURCE.

	Flight Time (s)				
	Initial 1	Initial 2	Initial 3	Initial 4	Initial 5
Source 1	13.10	91.95	33.25	22.70	82.70
Source 2	19.65	23.45	14.75	17.30	19.50
Source 3	88.50	19.35	15.85	24.35	18.15

B. *Sample Trials*

Five representative trajectories are selected for detailed analysis, as shown in Fig. 5 and Fig. 6. Specifically, Fig. 5 illustrates the UAS trajectories segmented by navigation mode, i.e., SCAN and SEARCH, with each segment colored according to its corresponding mode. Fig. 6 shows the same trajectories colored by the measured gas concentration (red indicating high concentration and blue indicating low concentration).

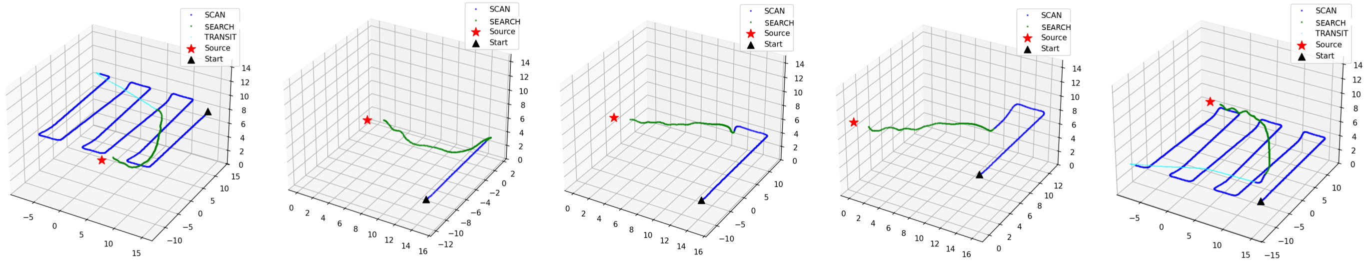
Case 1 (Fig. 5(a) and Fig. 6(a)) demonstrates the algorithm's ability to handle large vertical offsets, where the UAS is initialized at $(15, 10, 9)$ m and the gas source is located near the ground at $(0, 0, 2)$ m. Since no valid concentration signal is detected during the initial lawn-mower sweep, the UAS descends and repeats the lawn-mower pattern at a lower altitude. During this descent, high gas concentrations are detected, activating the transition to the SEARCH mode. Then, the UAS follows the concentration gradient and successfully localizes the source at $t = 82.7$ s.

Cases 2–4 (Fig. 5(b)–5(d) and Fig. 6(b)–6(d)) demonstrate the algorithm's ability to correct both lateral and vertical offsets under different initial UAS positions. In Case 2, the UAS starts at $(15, -12, 4)$ m and successfully corrects a 12 m lateral displacement, reaching the source at $t = 13.1$ s. Case 3 introduces an additional vertical offset, with the source located at $(0, 0, 6)$ m. The UAS climbs after the initial plume detection and reaches the source. In Case 4, the UAS is initialized at $(16, 0, 8)$ m, already aligned with the plume centerline, while the source is positioned higher at $(0, 0, 10)$ m. Trajectory shows that the UAS primarily climbs from 8 m to 10 m and successfully completes the trial at $t = 15.85$ s.

In Case 5 (Fig. 5(e) and Fig. 6(e)), the UAS begins at $(15, -12, 4)$ m, whereas the source is located significantly higher at $(0, 0, 10)$ m. No valid concentration signal is detected at the 4 m layer, making the UAS to ascend to 6.33 m. The plume is detected at $t = 77.25$ s, when the UAS switches to the SEARCH mode, and the source is successfully localized at $t = 88.50$ s. Across all five scenarios, the results demonstrate that the proposed navigation strategy is robust in the 3D environment.

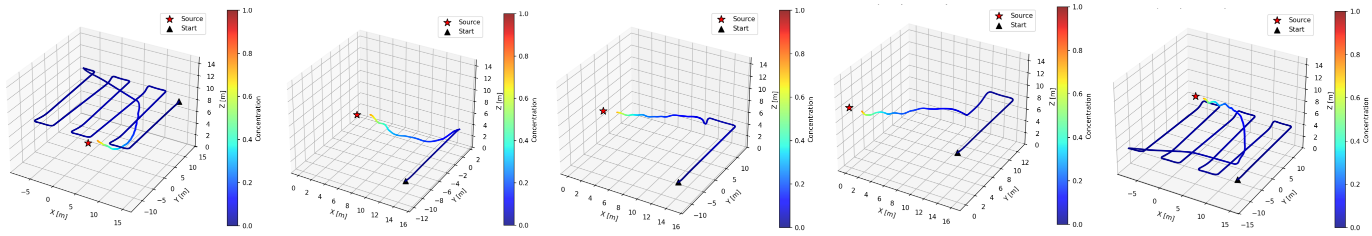
C. *Statistic Results*

Table I summarizes the flight time for all 15 experimental scenarios. The corresponding success rates are presented in Table II. In particular, the proposed GSL navigation strategy achieved a 100% success rate across all trials.



(a) Case 1 (b) Case 2 (c) Case 3 (d) Case 4 (e) Case 5

Fig. 5. UAS Trajectory State.



(a) Case 1 (b) Case 2 (c) Case 3 (d) Case 4 (e) Case 5

Fig. 6. UAS Trajectory Concentration.

TABLE II
AVERAGE FLIGHT TIME AND SUCCESS RATE OF TRIALS FOR DIFFERENT GAS SOURCE

	Average Flight Time (s)	Success Rate
Source 1	48.74	5/5
Source 2	18.93	5/5
Source 3	33.24	5/5

TABLE III
SYSTEM PERFORMANCE UNDER VARYING ENVIRONMENTAL AND ALGORITHMIC PARAMETERS (SOURCE 1).

Parameter Settings	Flight Time (s)				
	Initial 1	Initial 2	Initial 3	Initial 4	Initial 5
Default Settings	13.10	**91.95**	**33.25**	**22.70**	82.70
$u = 3.0$	**12.40**	93.45	75.55	23.65	86.15
$c_{thresh} = 0.08$	20.70	93.15	74.40	24.25	88.25
$N = 10$	13.90	107.30	**33.25**	24.15	84.45
$N = 40$	13.00	92.50	33.30	22.90	**82.55**

As shown in the average performance metrics (i.e., Table II), Source 2 proved to be the most accessible target, recording the lowest average flight time of 18.93 s. This efficiency is attributed to its central position within the vertical search space, allowing the UAS to intercept the plume quickly regardless of whether it started from a high or low altitude. In contrast, scenarios with significant vertical disparities required longer search durations.

Source 1 yielded the highest average time (48.74 s), a figure heavily skewed by high-altitude initializations (e.g., Initial 2 and 5), where the UAS had to traverse multiple layers downwards. Similarly, for Source 3, the low-altitude start (Initial 1) resulted in a prolonged duration of 88.50 s due to the necessary upward layer transition. However, even in these extreme cases, the algorithm successfully converged to the source, confirming that the strategy remains effective regardless of the large initial gap.

D. Parameter Analysis

To validate the proposed framework, we conducted a parameter analysis. All experiments were carried out under a low-altitude source scenario (Source 1). The default parameter settings were defined as wind speed $u = 2.0$ m/s, concentration threshold $c_{thresh} = 0.05$, and buffer size $N = 20$. The results are summarized in Table III. As highlighted in bold, the Default Settings yielded the shortest flight times for most initial conditions (Initial 2, 3 and 4). In addition, we change several parameter values to exam their impacts on the search performance. Under a higher wind speed environment ($u = 3.0$ m/s), the method achieved the fastest time for Initial 1 (12.40 s). The stronger wind advected the plume downstream more quickly, and because the initial 1 ($z = 4$ m) is close to the source height ($z = 2$ m), the UAS intercepted the plume earlier. Similarly, when we increase the buffer size, i.e., $N = 40$, the UAS can find the gas source slightly faster for the initial position 5 (82.55 s). For this high-altitude start that required a longer descent, the larger buffer smoothed the gradient estimates.

IV. CONCLUSION AND FUTURE WORKS

This paper presents a 3D GSL navigation algorithm that enables a UAS to autonomously localize a gas source. The proposed framework consists of two search modes, SCAN and SEARCH. In the SCAN mode, the UAS executes lawnmower trajectories at multiple altitude layers to detect initial gas plume signals. Upon detecting high gas concentrations, the UAS transitions to the SEARCH mode and traces the plume by following local gas concentration gradients. Simulation results demonstrate that the proposed navigation algorithm is effective across varying gas source locations and UAS initial conditions.

In this work, we used the `RotorPy` simulation and and the Gaussian-based gas dispersion model to support a controlled and repeatable evaluation. However, we acknowledge the gap between the simulated environment and real-world conditions. Real wind fields are uncertain and time-varying, and atmospheric turbulence can cause strong plume intermittency. In addition, sensor noise, response latency, and onboard computational constraints can further affect the stability of the proposed method. Future work will focus on evaluating the proposed approach under more turbulent and time-varying wind conditions. We also plan to incorporate more realistic sensor models (e.g., gas sensor noises) and to evaluate the approach under practical deployment constraints (e.g., drone battery constraints). Moreover, extending the framework to incorporate multiple cooperative agents for plume tracing is also a promising direction to further improve source localization efficiency.

ACKNOWLEDGMENT

This work is supported by the National Science Foundation (Award: 2502074) and Louisiana State University Sub-award (Award: PO-0000307044).

REFERENCES

[1] H. Ishida, Y. Wada, and H. Matsukura, "Chemical Sensing in Robotic Applications: A Review," *IEEE Sensors Journal*, vol. 12, no. 11, pp. 3163–3173, Nov. 2012.

[2] T. Jing, Q.-H. Meng, and H. Ishida, "Recent Progress and Trend of Robot Odor Source Localization," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 16, no. 7, pp. 938–953, Jul. 2021.

[3] X.-x. Chen and J. Huang, "Odor Source Localization Algorithms on Mobile Robots: A Review and Future Outlook," *Robotics and Autonomous Systems*, vol. 112, pp. 123–136, Feb. 2019.

[4] J. Wang, Y. Lin, R. Liu, and J. Fu, "Odor source localization of multi-robots with swarm intelligence algorithms: A review," *Frontiers in Neurorobotics*, vol. 16, Nov. 2022.

[5] D. Rjoub, A. Alsharoa, and A. Masadeh, "Unmanned-Aircraft-System-Assisted Early Wildfire Detection with Air Quality Sensors," *Electronics*, vol. 12, no. 5, p. 1239, Mar. 2023.

[6] S. K. Pal, S. Sharma, N. Krishnakumar, and J. Hong, "Autonomous Drone for Dynamic Smoke Plume Tracking," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, May 2025, pp. 303–309.

[7] L. Wang, S. Pang, M. Noyela, K. Adkins, L. Sun, and M. El-Sayed, "Vision and Olfactory-Based Wildfire Monitoring with Uncrewed Aircraft Systems," in *2023 20th International Conference on Ubiquitous Robots (UR)*, Jun. 2023, pp. 716–723.

[8] S. Shigaki, Y. Yoshimura, D. Kurabayashi, and K. Hosoda, "Palm-Sized Quadcopter for Three-Dimensional Chemical Plume Tracking," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–12, 2022.

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

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

[11] L. Wang and S. Pang, "Robotic odor source localization via adaptive bio-inspired navigation using fuzzy inference methods," *Robotics and Autonomous Systems*, vol. 147, p. 103914, 2022.

[12] S. Pang and J. Farrell, "Chemical Plume Source Localization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 36, no. 5, pp. 1068–1080, Oct. 2006.

[13] J.-G. Li, Q.-H. Meng, F. Li, M. Zeng, and D. Popescu, "Mobile robot based odor source localization via particle filter," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) Held Jointly with 2009 28th Chinese Control Conference*, Dec. 2009, pp. 2984–2989.

[14] M. Park, P. Ladosz, and H. Oh, "Source Term Estimation Using Deep Reinforcement Learning With Gaussian Mixture Model Feature Extraction for Mobile Sensors," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8323–8330, Jul. 2022.

[15] J. Hai-Feng, C. Yu, D. Wei, and P. Shuo, "Underwater chemical plume tracing based on partially observable Markov decision process," *International Journal of Advanced Robotic Systems*, vol. 16, no. 2, p. 1729881419831874, Mar. 2019.

[16] L. Wang, S. Pang, and J. Li, "Olfactory-Based Navigation via Model-Based Reinforcement Learning and Fuzzy Inference Methods," *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 10, pp. 3014–3027, Oct. 2021.

[17] J. Monroy, J.-R. Ruiz-Sarmiento, F.-A. Moreno, F. Melendez-Fernandez, C. Galindo, and J. Gonzalez-Jimenez, "A semantic-based gas source localization with a mobile robot combining vision and chemical sensing," *Sensors*, vol. 18, no. 12, p. 4174, 2018.

[18] F. Rahbar, A. Marjovi, and A. Martinoli, "An algorithm for odor source localization based on source term estimation," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 973–979.

[19] C. Ercolani, L. Tang, and A. Martinoli, "Gaslam: An algorithm for simultaneous gas source localization and gas distribution mapping in 3d," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 333–340.

[20] M. Jia, R. Fish, W. S. Daniels, B. Sprinkle, and D. Hammerling, "A fast and lightweight implementation of the gaussian puff model for near-field atmospheric transport of trace gasses," *Scientific Reports*, vol. 15, no. 1, p. 18710, 2025.

[21] C. A. Dimmig, G. Silano, K. McGuire, C. Gabellieri, W. Hönig, J. Moore, and M. Kobilarov, "Survey of simulators for aerial robots: An overview and in-depth systematic comparisons [survey]," *IEEE Robotics & Automation Magazine*, vol. 32, no. 2, pp. 153–166, 2024.

[22] S. J. Miller, "The method of least squares," *Mathematics Department Brown University*, vol. 8, no. 1, pp. 5–11, 2006.