In: 2024 IEEE International Smart Cities Conference (ISC2), 29 Oct - 1 Nov 2024, Pattaya, Thailand
DOI: 10.1109/isc260477.2024.11004185

Drone Pollution Tracking in Cities using Recurrent Proximal Policy Optimization Learning

Andreas Eliades[1], Elie Thellier[1], Haitao Tian[1], Shivam Mehta[1], Nouf Almesafri[1], Hongqian Chen[1], Zhuangkun Wei[1], Adolfo Perrusquía[1], Cunjia Liu[2], Weisi Guo[1*]

***Abstract*—Future smart cities will need to monitor polluters that periodically or burst emit illegal gases that is harmful to the environment. Tracking these sources in cities that have building obstacles and variation wind vector fields is challenging. Traditional methods using gradient kernels and partial-swarm-optimisation may not be suitable when the emissions are intermittent and pollution concentrations maybe trapped in local pockets. As such, step size tuning becomes difficult to generalise in these variational dynamic pollution environments. Here, in this paper, we have developed a simulated urban pollution propagation environment, whereby a drone is scanning the environment for gradients to search and localise the source. We consider both proximal policy optimisation (PPO)-based reinforcement learning and its recurrent PPO (R-PPO) alternative to achieve stable and reliable improvement of policy without the need to fine tune step sizes. We show localisation results across a range of wind, obstacle, and emission scenarios with success rate of 76-79% and high path efficiency of 95-96% in ideal conditions. When we examine alternative city structures and burst emissions, we can achieve success rate of 34% and path efficiency of 52%, showing that there is some generalisation in capability.**

***Index Terms*—Source localization; Proximal Policy Optimization; Navigation; Gas Emission; Static and Dynamic Environments**

I. Introduction

Smart cities will increasingly integrate sensors and drones for surveillance [1], [2]. In the field of urban environmental monitoring and response, the rapid and accurate localization of hazardous gas emissions, such as Chlorofluorocarbons (CFC), is crucial for minimizing detrimental impacts on human health and ecological systems [3], [4].

A. Review of Classical Localisation

Classical localisation has largely focused on smooth disperse problems (e.g., diffusion dominated at small scales) [5], [6] or obstacle free problems [7] using particle swarm optimisation (PSO). Other methods have used modified gradient search algorithms, such as the Rosenbrock gradient ascent algorithm adapted for underwater molecular signal tracking [8], where a robot uses an averaging space-time gradient kernel to reduce sensitivity to noise. Another approach is to attempt to inverse solve the problem using physics-informed neural networks (PINN) [9].

These algorithms have largely worked in relatively slow or uni-directional propagation environments (e.g., rivers and sewage pipes) and have offered some success but struggle against the unpredictable dynamics of atmospheric dispersion in urban areas, influenced significantly by interaction with obstacles, underscoring the pressing need for more sophisticated, reliable solutions. More recent approaches have used an information-based Bayesian search algorithm to work in turbulent environments [10], but this is largely for small-scale turbulence effects and do not consider large scale currents and obstacles in cities [11], [12].

The emergence of reinforcement learning (RL) has presented a new frontier in addressing these challenges by combining sensing and navigation together into a common learning agent's goal [13], [14]. RL enables a single or multiple learning agents to derive optimal behaviors from environmental interactions, optimizing for cumulative rewards—a fit for the sequential decision-making demands of odor source localization (OSL).

B. Deep Reinforcement Learning in Localisation

Unlike Bayesian or PSO methods, deep reinforcement learning (DRL) excels in handling complex, high-dimensional state spaces and offers real-time decision-making capabilities after substantial training through a single or multiple cooperative agents [15]. The challenge lies in configuring a robust DRL for the diverse turbulent/windy pollution propagation dynamics in tracing problems.

1) Proximal Policy Optimization (PPO): The PPO algorithm [16] represents a significant advancement in the field of DRL, particularly in policy gradient methods. PPO aims to address the key challenges associated with training policies, specifically ensuring stable and reliable improvement of the policy with each iteration, without the need for finely tuning the step sizes. This is crucial in applications such as 2D source localization in environments with discrete observation

[1]Authors are with the School of Aerospace, Transport, and Manufacturing, Cranfield University, Bedford, UK. [2] Author with the School of Aeronautical and Automotive Engineering, Loughborough University, UK. *Corresponding author: weisi.guo@cranfield.ac.uk. This work is partially supported by EPSRC TAS-S: Trustworthy Autonomous Systems: Security (EP/V026763/1).

Published by IEEE. This is the Author Accepted Manuscript issued with: Creative Commons Attribution License (CC:BY 4.0).
The final published version (version of record) is available online at DOI:10.1109/isc260477.2024.11004185. Please refer to any applicable publisher terms of use.

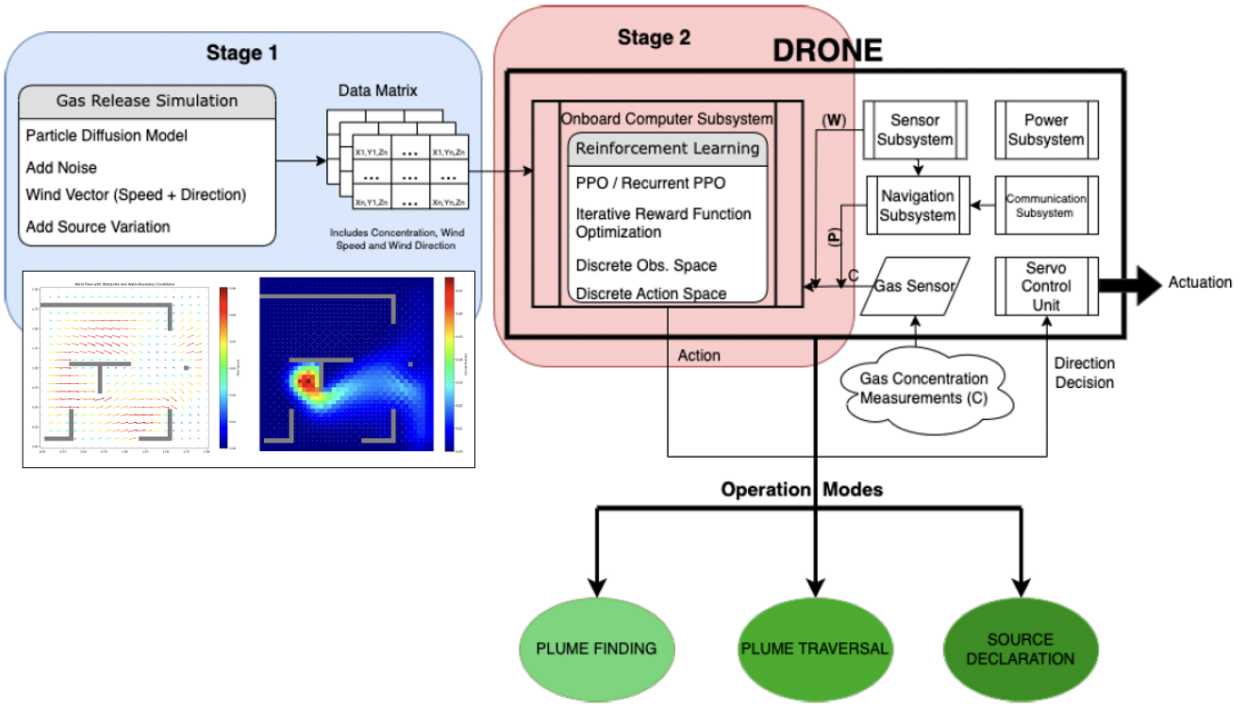

Fig. 1. System Diagram of Drone to Detect Illegal Pollution in Cities: including the gas release simulation (Stage 1) and the RL algorithm (Stage 2) onboard the drone.

and action spaces, where the agent must learn to navigate and make decisions based on limited and discretized sensory inputs to locate a target or source. PPO operates by optimizing a surrogate objective function that limits the amount of change imposed on the policy at each update, thereby preventing drastic policy updates that could lead to performance collapse. This is achieved through a clipped objective function, which restricts the ratio of the new policy to the old policy (a ratio termed the probability ratio) to stay within a neighbourhood defined by hyper-parameters. This mechanism ensures that the updates take small, safe steps, promoting gradual and stable learning progress.

2) Recurrent Proximal Policy Optimization (R-PPO): The innovative integration of Long Short-Term Memory (LSTM) networks into the R-PPO algorithm marks a significant advancement in training agents for complex tasks [17], such as locating a gas source in a 2D discrete environment. The inclusion of LSTM networks enables R-PPO to effectively handle environments where observations possess temporal dependencies, a critical feature for navigating through space where the dynamics can change over time or where the full effects of actions are only observable in a delayed manner. In environments characterized by partial observability, where an agent cannot fully see the entire environment at once, the role of LSTMs becomes even more pivotal. For instance, in the task of locating a gas source, the agent may only detect hints of the gas's presence in its immediate vicinity without a clear indication of the source's exact location. LSTMs empower R-PPO to leverage its memory component, allowing the agent to remember and integrate past observations with the current state. This capability is crucial for making informed decisions that guide the agent towards the goal, despite having only a fragmented view of the environment.

C. Novelty

In summary, traditional methods using gradient kernels and partial-swarm-optimisation may not be suitable when the emissions are intermittent and pollution concentrations maybe trapped in local pockets. Even for RL, step size tuning becomes difficult to generalise in these variational dynamic pollution environments. As such, we are motivated to consider both proximal policy optimisation (PPO)-based reinforcement learning and its recurrent PPO (R-PPO) alternative to achieve stable and reliable improvement of policy without the need to fine tune step sizes.

The overall system architecture shown in Fig.1 and is comprises two main stages and a physical component, representing the drone where the deployed algorithm is intended for use.

1) The first stage involves gas release simulation, aimed at generating data to train a custom model. The primary objectives of this stage are to produce reliable and semi-realistic physics-informed gas plumes and to export the data in a format suitable for the second stage.
2) The second stage centres around the Reinforcement Learning (RL) algorithm of the drone. The RL is designed to receive wind data from the Sensor subsystem, position data from the Navigation subsystem, and gas concentration data from the Gas Sensor onboard the drone. The trained RL algorithm then determines the next action for the drone. Within the source localization framework, the drone can operate in three different modes: plume finding and traversal, and source declaration.

We will show localisation results across a range of wind, obstacle, and emission scenarios for the KPIs of localisation success rate and journey path efficiency. We will examine alternative city structures and burst emissions, to challenge the out-of-sample generalisation capability of the algorithms.

II. City Gas Emission Simulation Environment

A discretized partial differential equation (PDE) Navier-Stokes model is utilized:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot)\nabla\mathbf{u} = -\frac{1}{\rho}\nabla p + \nu\nabla^2\mathbf{u}, \tag{1}$$

Where $\frac{\partial \mathbf{u}}{\partial t}$ denotes the rate of change of velocity vector $\mathbf{u}$ with respect to time t. The term $(\mathbf{u}\cdot)\nabla\mathbf{u}$ represents the convective acceleration, capturing how the velocity field advects itself. On the right-hand side, $-\frac{1}{\rho}\nabla p$ represents the pressure gradient term, where ρ is the pressure and p is the fluid density. Lastly, $\nu\nabla^2\mathbf{u}$ denotes the viscosity term, with ν being the kinematic viscosity and $\nabla^2\mathbf{u}$ representing the Laplacian of the velocity vector, accounting for the diffusion of momentum due to viscosity. The simulation is conducted in a two-dimensional environment with a uniform wind velocity and a fixed substance-emitting source. The simulation employs a finite difference method (FDM) to effectively solve the advection-diffusion equation -flowchart shown in Fig 2. This equation is discretized and then numerically solved over a discrete grid that accurately represents the spatial domain. The time evolution of the substance concentration is simulated through the utilization of the Runge-Kutta method. To ensure that the substance concentration is preserved at the boundaries, the team applied zero Dirichlet boundary conditions. Zero Dirichlet boundary conditions are imposed on all domain boundaries, assuming a substance concentration of zero at these boundaries, akin to impermeable barriers. This prevents substance escape and maintains computational stability. Post simulation, the solution is extended to the domain edges, ensuring a smooth transition and mitigating abrupt gradients that could induce numerical instability.

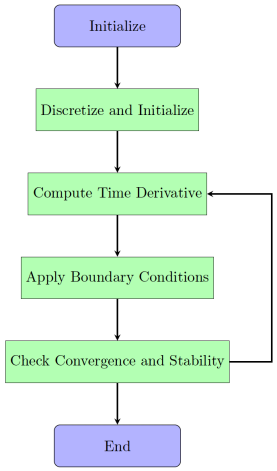

Fig. 2. Generation Algorithm for Pollution Variational Flow in City Obstacle Environment

To investigate the combined effects of varying wind speeds and obstacles on the dispersion of a substance, we extend the modified PDE model used in the variant wind speed simulation. This extension incorporates obstacles within the two-dimensional environment, altering the substance's dispersion characteristics in the presence of barriers. Three pollution emission models are considered: (1) continuous emission (e.g., industrial chimney), (2) periodic emission (e.g., exhaust from traffic), and (3) one burst (e.g., event/disaster based). An example of continuous emission with obstacles and variational wind is shown in Fig.1.

III. Pollution Source Localisation and Navigation

The drone's objective is to approach the source location iteration by iteration within minimum steps while solely relying on its sensors' measurements and by making the next move decision based on this information. In the first instance, we can model this process as a Markov Decision Process in which the agent's action only relies on the current observation and does not consider history. Inside the RL environment, the model contains an action space and a state (or observation) space. It also contains a reward function and the problem can be solved using a variety of algorithms with different hyper-parameters that we fine-tuned in order to have the best performing model.

A. DRL Design Choices

Proximal Policy Optimization (PPO) and its variant, Recurrent PPO (RPPO), are compared based on their performance

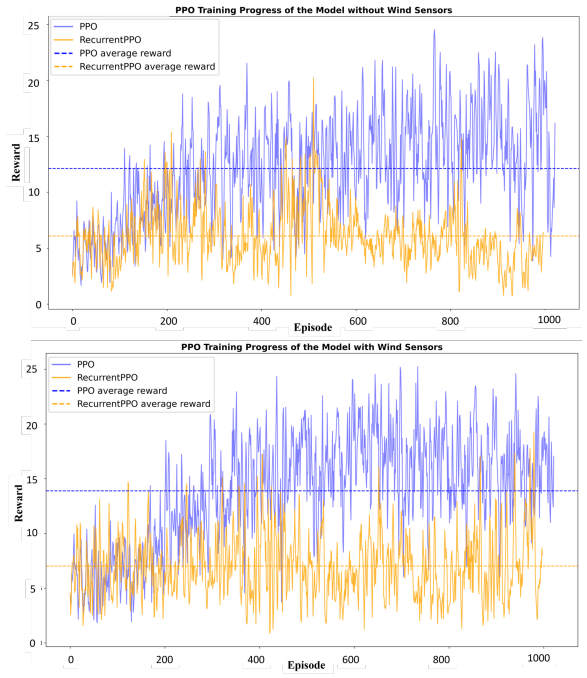

Fig. 3. Training Process for the PPO and Recurrent PPO Algorithms (top) without Wind Sensors, and (bottom) with Wind Sensors.

across various environments, focusing on the mean episodic rewards achieved during both training and evaluation phases. One critical distinction between these two algorithms lies in their handling of temporal dependencies. While PPO operates without the use of frame-stacking, suggesting a straightforward, moment-to-moment decision making process, RPPO incorporates a recurrent neural network layer (typically an LSTM or GRU), enabling it to effectively manage and leverage temporal sequences for decision-making. This difference fundamentally alters their approach to processing environmental states, with RPPO being potentially better suited for environments where understanding the sequence of events is crucial.

The key configurations of the RL are as follows:

- Action Space: We the simple four cardinal directions (up, down, left, and right).
- Observation States: Four concentration measurements in a cross configuration (four cardinal directions) around the drone. This enables to measure the concentration in the drone's vicinity.

B. Reward Function Design

We are first motivated by the concentration change: $R_1 = C_t - C_{t-1}$. Two versions of this base reward were studied: giving absolute or relative reward values. The advantage of using an absolute reward value is that we can easily regularize the total reward to facilitate the training process. However, a relative reward value that depends on the norm of the gradient was chosen to encourage the agent to take the steepest paths. This base reward is upgraded by adding a bonus reward of 10% if the gradient is positive, meaning that the drone is moving toward the source.

Additionally, a fixed reward value $R_2 = C_t$ if $C_t > C_{max}$ is given for every time the agent finds a new maximum concentration because it incentivizes the agent to actively seek out and discover areas with higher concentrations, thereby promoting efficient exploration and exploitation of the environment. The maximum found concentration is updated every time this reward is obtained. This fixed reward serves as positive reinforcement for the behavior of finding the source of the pollutant, encouraging the agent to continually strive for better results instead of abusing the bonus of the base reward.

The final reward value is inspired by anemotaxis, a term that refers to the ability of certain organisms, particularly insects, to navigate or orient themselves by detecting and moving towards odor cues carried by the wind. In our scenario, the agent should only go up the wind vector if it is inside the plume, thus the reward $R_3 = 0.1C_t$ is given only if the measured concentration is higher than a fixed threshold.

The total reward is: $R = \sum_t^T R_1(t) + R_2(t) + R_3(t)$, where T is the maximum tolerated step sizes for localisation which is set to 50.

TABLE I
HYPERPARAMETERS FOR THE PPO AND RECURRENT PPO MODELS.

Hyperparameter	Value	Description
Number of training steps	50,000	Number of training steps, making the number of episodes to 1,000
Minibatch size	52	The number of samples in each batch during training
N-epochs	11	The number of times the entire dataset is passed forward and backward through the neural network
Discount factor	0.98	Discount factor for future rewards in reinforcement learning
Policy entropy coefficient	0.04	Coefficient for entropy regularization in reinforcement learning
Learning rate	0.0059	Rate at which the model's parameters are updated during training
GAE parameter	0.92	Generalized advantage estimation coefficient λ
PPO clip range	0.11	Range for clipping the gradient updates in reinforcement learning

C. DRL Training

IV. RESULTS

A. Static Environment

Fig.4(top) compares the success rates and model efficiencies of our four models. Recurrent PPO models do not yield good results as the training was not successful. Failures include agents stuck on local maxima created by the obstacles and

Model	Success Rate	Average Path Efficiency
PPO without wind sensor	76%	0.954
PPO with wind sensor	79%	0.965
R-PPO without wind sensor	39%	0.629
R-PPO with wind sensor	44%	0.643

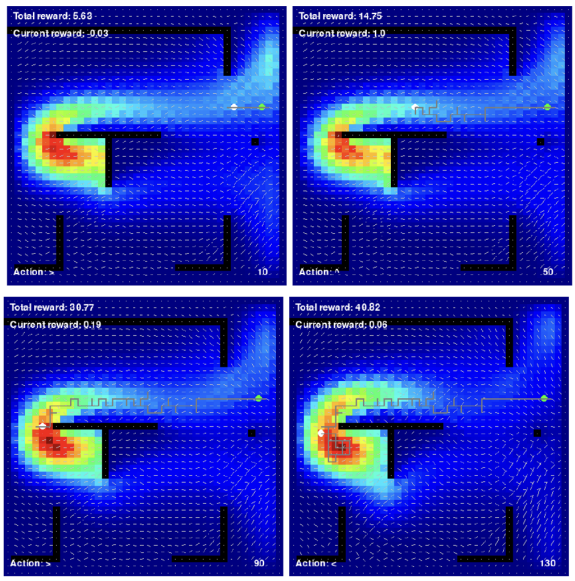

Fig. 4. The search process in different environment setups: (top) statistical results of the static environment, and (bottom) illustration of dynamic environment search processes at different time steps.

scenarios in which the agent loses the plume and thus moves randomly in the area with no concentration. However, the results of the PPO models are encouraging as the agent is able to consistently move to the area with the maximum concentration and find the source location most of the time. Additionally, PPO models take very efficient paths as shown in the path efficiency values. Finally, adding the wind vector to the reward function and observation space seems profitable for the agent to wind up toward the source location. It is worth noting that stochastic behavior improves the success rate, at the expense of a lower path efficiency.

B. Dynamic Environment

The best PPO model with wind direction sensor is evaluated in a customized environment where the plume is in a steady state that dynamically changes due to variations in the wind field. Fig.4(bottom) shows the search process at simulation time steps 10, 50, 90, and 130. The agent is heading in the right direction from the beginning, then the varying concentration value makes the gradient reward less representative, while the reward based on the wind direction helps the drone to wind up the plume. The agent successfully finds the zone with the source location at time step 93, then hovers in this area. The PPO with wind sensor had 49% success chance and 0.463 path efficiency.

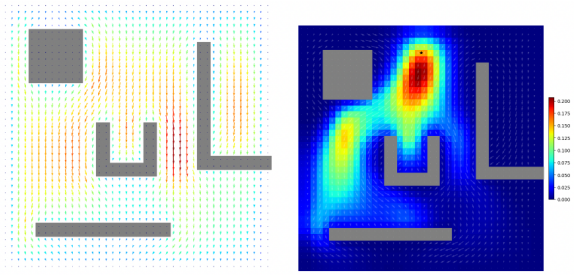

Fig. 5. Out of Sample Validation in Different City.

Model	Success Rate	Average Path Efficiency
PPO with wind sensor	34%	0.520

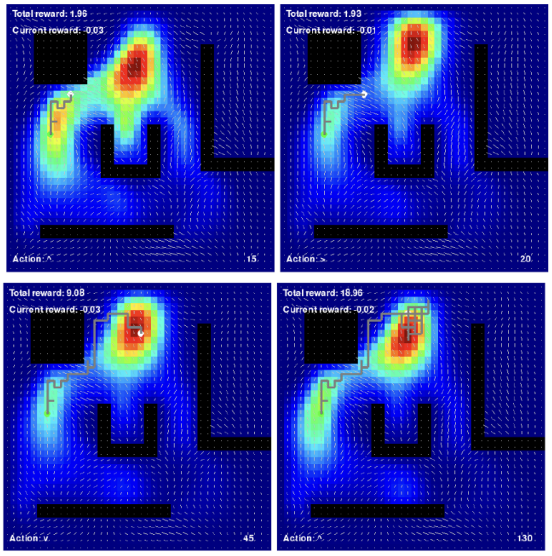

Fig. 6. Out of Sample Validation in Different City with Burst Emissions.

C. Out of Sample Alternative City Structures

The PPO model is evaluated in an environment with a different obstacle structure thus creating a different wind field as shown in Fig.5. The model was not trained in this environment. The agent position is initialised near a local concentration maxima. The drone is stuck until it leaves this area at time step 60 thanks to the policy that encourages the drone to move against the wind. The agent successfully finds the source of pollution at $t = 75$ then hovers into the final area. Across out of sample urban areas, we had a lower success rate of 39% and average path efficiency of 0.570.

D. Out of Sample Alternative City Structures with Periodic Burst Emission

The PPO model is evaluated with a periodic emission (defined by a burst emission every 18 time steps). The model

was not trained in this kind of environment but we expect the agent to use the hover action to gather information averaged over time. Fig.6 shows the search process at time steps 15, 20, 45, and 130 in the new dynamic environment. The agent position is initialised near a local concentration maxima. The drone takes advantage of the information obtained during the burst before $t = 18$ to head for the area with the highest concentration, then finds the source at $t = 45$ and hovers around the source. Due to the variations in out of sample cases, we had a lower success rate of 34% and average path efficiency of 0.520.

V. Conclusions

Smart cities will increasingly integrate drones for surveillance. In the field of urban environmental monitoring and response, the rapid and accurate localization of hazardous gas emissions is crucial. Our research into the application of novel deep reinforcement learning (DRL) for the task of pollution gas source localization within a custom-made 2D urban environment. We show localisation results across a range of wind, obstacle, and emission scenarios with success rate of 76-79% and high path efficiency of 95-96% in ideal conditions. When we examine alternative city structures and burst emissions, we can achieve success rate of 34% and path efficiency of 52%, showing that there is some generalisation in capability.

Moving forward, designing a drone equipped with the capability to detect a gas plume source location using a RL algorithm presents an innovative approach to environmental monitoring, industrial safety, and emergency response. However, integrating sophisticated algorithms with real-world hardware like sensors and microprocessors on a drone platform introduces several practical implementation challenges including sensor sensitivity and drone disturbance [18].

References

[1] M. Fadda, M. Anedda, R. Girau, G. Pau, and D. D. Giusto, "A social internet of things smart city solution for traffic and pollution monitoring in cagliari," *IEEE Internet of Things Journal*, vol. 10, no. 3, pp. 2373–2390, 2023.

[2] M. Khan, W. Gueaieb, A. E. Saddik, G. De Masi, and F. Karray, "An efficient violence detection approach for smart cities surveillance system," in *2023 IEEE International Smart Cities Conference (ISC2)*, 2023, pp. 1–5.

[3] Z. Fu, Y. Chen, Y. Ding, and D. He, "Pollution source localization based on multi-uav cooperative communication," *IEEE Access*, vol. 7, pp. 29 304–29 312, 2019.

[4] S. Hassan, L. Wang, and K. R. Mahmud, "Multi-modal robotic platform development for odor source localization," in *IEEE International Conference on Robotic Computing (IRC)*, 2023.

[5] Y. Wang, Y. Hu, J. Wang, and J. Tu, "Pollution source localization in complex environments using a swarm of mobile agents," *IEEE Sensors Letters*, vol. 6, no. 1, pp. 1–4, 2022.

[6] S. Kumar, "Nanomachine localization in a diffusive molecular communication system," *IEEE Systems Journal*, vol. 14, no. 2, pp. 3011–3014, 2020.

[7] H. Saadaoui and F. El Bouanani, "A local pso-based algorithm for cooperative multi-uav pollution source localization," *IEEE Access*, vol. 10, pp. 106 436–106 450, 2022.

[8] S. Qiu, W. Guo, B. Li, Y. Wu, X. Chu, S. Wang, and Y. Y. Dong, "Long range and long duration underwater localization using molecular messaging," *IEEE Transactions on Molecular, Biological, and Multi-Scale Communications*, vol. 1, no. 4, pp. 363–370, 2015.

[9] G. Kim and G. Heo, "Solving partial differential equation for atmospheric dispersion of radioactive material using physics-informed neural network," *Nuclear Engineering and Technology*, vol. 55, no. 6, 2023.

[10] M. Hutchinson, C. Liu, and W.-H. Chen, "Information-based search for an atmospheric release using a mobile robot: Algorithm and experiments," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 6, pp. 2388–2402, 2019.

[11] C. Rhodes, C. Liu, P. Westoby, and W.-H. Chen, "Autonomous search of an airborne release in urban environments using informed tree planning," *Autonomous Robots*, vol. 47, no. 1, pp. 1–18, 2023.

[12] Y. Zhao, B. Chen, Z. Zhu, F. Chen, Y. Wang, and D. Ma, "Entrotaxis-jump as a hybrid search algorithm for seeking an unknown emission source in a large-scale area with road network constraint," *Expert Systems with Applications*, vol. 157, p. 113484, 2020.

[13] C. Lin, G. Han, T. Zhang, S. B. H. Shah, and Y. Peng, "Smart underwater pollution detection based on graph-based multi-agent reinforcement learning towards auv-based network its," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 7, pp. 7494–7505, 2023.

[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, 2022.

[15] Z. Hu, T. Song, K. Bian, and L. Song, "Deep reinforcement learning based indoor air quality sensing by cooperative mobile robots," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, 2020, pp. 1–6.

[16] Y. Gu, Y. Cheng, C. L. P. Chen, and X. Wang, "Proximal policy optimization with policy feedback," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 7, pp. 4600–4610, 2022.

[17] M. Pleines, M. Pallasch, F. Zimmer, and M. Preuss, "Generalization, mayhems and limits in recurrent proximal policy optimization," *arXiv*, 2022.

[18] R. Baptista, H. Magalhães, and L. Marques, "Impact of uav propellers on gas plume tracking," in *2023 European Conference on Mobile Robots (ECMR)*, 2023, pp. 1–7.

https://dspace.org/

Drone pollution tracking in cities using recurrent proximal policy optimization learning

Eliades, Andreas; Thellier, Elie; Tian, Haitao; Mehta, Shivam; Almesafri, Nouf ...

2024-10-29
Institute of Electrical and Electronics Engineers (IEEE)
https://doi.org/10.1109/isc260477.2024.11004185
Eliades A, Thellier E, Tian H, et al., (2024) Drone pollution tracking in cities using recurrent proximal policy optimization learning. In: 2024 IEEE International Smart Cities Conference (ISC2), 29 Oct - 1 Nov 2024, Pattaya, Thailand

Downloaded from a DSpace repository. https://dspace.org/