

An Algorithm for Odor Source Localization based on Source Term Estimation

Faezeh Rahbar, Ali Marjovi, Alcherio Martinoli

Abstract—**Finding sources of airborne chemicals with mobile sensing systems finds applications across the security, safety, domestic, medical, and environmental domains. In this paper, we present an algorithm based on source term estimation for odor source localization that is coupled with a navigation method based on partially observable Markov decision processes. We propose an innovative strategy to balance exploration and exploitation in navigation. The method has been evaluated systematically through high-fidelity simulations and in a wind tunnel emulating realistic and repeatable conditions. The impact of multiple algorithmic and environmental parameters has been studied in the experiments.**

I. INTRODUCTION

In a large number of situations, finding the source of a gaseous chemical released in the air becomes vital. For instance, in the context of search and rescue, or safety and security operations, such mission would involve finding explosives or drugs in airports or industrial facilities, or search for survivors in case of natural hazards.

Despite the growing community of robotics and environmental engineers studying this crucial topic for more than two decades, odor source localization using mobile or static sensor nodes remains a challenging operation in realistic environments. The main difficulty arises from the dispersion phenomenon [1] which is a combination of molecular diffusion that drives odor patches away from the source, as well as the advection due to the airflow that carries the molecules in its direction [2]. As a result, the odor plume is shaped not only by the characteristics of the source, but also by the airflow which, in turn, is influenced by the environment.

To simplify this challenging problem, scientists usually divide it into three sub-tasks [3]: (i) *odor plume acquisition*, which refers to a search method that aims to find the plume in the environment; (ii) *odor plume tracking*, that is the phase where the robot leverages the samples obtained from the plume to localize the source; (iii) *odor source declaration*, is the final task, during which the agent validates and declares the location of the source.

The first and the last sub-tasks usually involve other modalities and depend on the application or the scenario in which the strategy is taking place. Therefore, most of the studies in the literature focus on the plume tracking sub-task, which is the core and the most challenging part of the

The authors are with the Distributed Intelligent Systems and Algorithms Laboratory, School of Architecture, Civil and Environmental Engineering, École Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland.

This work was funded by the Swiss National Science Foundation under grant 200020_175809/1.

mission. A large variety of strategies have already been implemented for this sub-task that can be classified into four often overlapping categories [4]: gradient-based, bio-inspired, formation-based, and probabilistic algorithms.

Gradient-based algorithms assume the plume to be a smooth concentration gradient. This type of algorithms, while being the most intuitive and computationally light ones, need relatively long time windows to find the source [5].

Bio-inspired algorithms define methods similar to existing behaviors in nature, such as search strategies of moths, dogs, bacteria, etc. [6]. The advantage of these algorithms is in their independence from any a priori information about the environment, its atmospheric conditions, or the historic observations, which makes them robust for unknown and dynamic areas. Nevertheless, due to low performance of current sensing and locomotion technologies compared to their biological counterparts, these algorithms are still far from being reliable in realistic environments [7].

In a relatively new approach, formation-based algorithms are designed for multi-robot systems [8]. Sampling multiple points at the same time is an advantage of this type of method, but since the coordination between agents is necessary, the entire method relies on inter-robot communication and relative positioning.

Probabilistic algorithms model a belief on the source location in the form of a probability distribution derived from the observations made by the agents in the environment [4]. After each observation, the belief is updated using Bayesian estimation. This process continues until the probability distribution reduces to a Dirac function. Infotaxis [9], particle-filter-based algorithms [10], and Source Term Estimation (STE) [11] are the main examples of this category.

While these algorithms can be computationally expensive, they have many advantages. Firstly, unlike the previously mentioned methods that provide only the source position, they generate a richer set of information about the environment (e.g., [12]) or the source characteristics [11]. Secondly, since the framework is probabilistic, the provided information is associated with an amount of uncertainty, which shows how trustworthy the data are. Lastly, compared to the other categories, probabilistic algorithms are more flexible in terms of the type of underlying hardware (e.g., static, mobile, single or multi-agent).

Considering the above mentioned advantages, we chose to focus on STE algorithms in this paper. STE algorithms rely on a plume model and aim to learn the parameters

of the model while the sensory system gathers data in the environment. Since the concept is very broad, the algorithm is not exclusively used for gas sources; depending on the model, it can be applied to any type of source (e.g., radiation [13]). Neither is it limited to mobile sensor nodes, as it can be used with a static sensor network (e.g., [14]).

However, when used on mobile robots, STE algorithms can be coupled with a navigation strategy which makes the data collection more time-efficient. Different navigation methods have already been used with STE in the literature. Partially Observable Markov Decision Processes (POMDP) [13] and mutual information maximization [15] are among the popular methods in this area. However, most of the works involving mobile robots have been only evaluated in simplified simulations (e.g., [15], [16], [17]).

Hence, the first contribution of the present work is the evaluation of an STE algorithm in a high-fidelity robotics simulator as well as in the physical reality. Additionally, the evaluation was done in different environmental conditions to verify the robustness of the method to various configuration. Moreover, the influence of a few key algorithmic parameters of the method on the performance was also studied. Finally, an enhanced navigation strategy is presented and evaluated.

II. METHOD

As mentioned previously, this work is built on the STE algorithm. STE is a probabilistic framework in which different strategies can be used. The general idea is to rely on a plume concentration model whose parameters have to be estimated throughout the experiment. The nature of the parameters depends on the model of choice, but, in any case, they include the source position, which is the goal of the Odor Source Localization (OSL) problem.

STE methods are usually flexible enough to be performed on a large variety of sensing assets, ranging from static sensor networks to heterogeneous mobile robots. In the case of a controllable mobile system, the estimation algorithm is augmented with a navigation strategy to increase the amount of acquired information at each step.

Since in most of the applications of OSL the optimal method is the one with the lowest time of estimation, we believe that using mobility could significantly improve the performance. Moreover, assuming slowly changing plume conditions, in absence of mobility, a large number of static sensors would be required to capture spatially rich enough information, while much fewer mobile sensing assets would be sufficient to gather the same amount of information. Therefore, in this work, we use a single mobile robot equipped with a chemical sensor to sample the environment.

As a result, our method consists of two main parts: estimation and navigation. In the estimation part, the algorithm estimates the parameters using the data that the robot has acquired. In the navigation part, the goal is to send the robot to the best neighboring point to obtain as much information as possible. This cycle, represented in Figure 1, continues until the uncertainty on the parameter estimation becomes negligible.

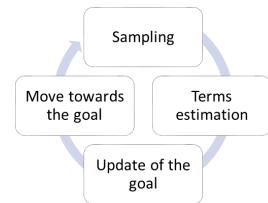


Fig. 1. The global structure of STE algorithm using a mobile robot

A. Plume Model

Probabilistic algorithms usually rely on a plume model for updating their belief about the source position. In this work, we chose to use the pseudo-Gaussian concentration plume model [18], presented in Eq. (1), which describes the time-averaged concentration model for a continuous point source in a laminar flow, where Q is the source release rate and \bar{u} the average wind speed.

$$C(Q, x_s, y_s, \sigma_y) = \frac{Q}{\sqrt{2\pi}\bar{u}\sigma_y} e^{-\frac{(y-y_s)^2}{2\sigma_y^2}} \quad (1)$$

In this equation, the X-axis is assumed to be aligned with the direction of the airflow and it is defined for all the x downwind of the source (i.e., $\forall x \geq x_s$). The concentration is simply 0 for all points upwind of the source (i.e., $\forall x < x_s$). The source is positioned at (x_s, y_s) and σ_y is the standard deviation of odor dispersion on the Y-axis, which is simplified to be a linear function of the upwind distance from the source $(x - x_s)$.

Moreover, since we are leveraging an STE method, the goal of the algorithm is to estimate the parameters of the model. Therefore, the set of parameters to be estimated for this model is $m = \{Q, x_s, y_s, \sigma_y\}$. As the standard deviation σ_y is of form $\sigma(x - x_s) = a(x - x_s) + b$, it has two parameters to be estimated. Thus, five parameters remain to be estimated, making the problem five-dimensional.

B. Parameter Estimation

The estimation is performed probabilistically using the Bayesian formulation presented in Eq. (2), where m represents the set of model's parameters and D the obtained data through sampling. The posterior $P(m|D)$ represents the probability distribution on the parameters values.

$$P(m|D) = \frac{P(m)P(D|m)}{P(D)} \quad (2)$$

The evidence $P(D)$ being a normalization factor, it can be neglected. Hence,

$$P(m|D) \propto P(m)P(D|m) \quad (3)$$

We consider the prior $P(m)$ a uniform distribution between the limits of each parameter. Therefore, the posterior $P(m|D)$ will be proportional to the likelihood $P(D|m)$ within the parameters limits, and equal to 0 outside.

The likelihood $P(D|m)$ defines the probability of obtaining a set of data, given a set of parameters. In other words, it returns the likelihood of a set of parameters given the data

that the robot collected up to the present time. It is defined in [14] as follows:

$$P(D|m) \propto \exp\left(-\frac{1}{2} \sum_i \left(\frac{(D_i - C_i(m))^2}{\sigma_M^2 + \sigma_D^2}\right)\right) \quad (4)$$

where σ_M and σ_D represent the standard deviations of model and measurement error, respectively. Both errors are assumed to be normally distributed, with mean on 0.

The sum in Eq. (4) is applied on all samples the robot gathered during the experiment. $C_i(m)$ is the concentration determined by the plume model for a set of source parameters m for sample i . It is defined using the pseudo-Gaussian concentration plume model presented in Section II-A. D_i is the actual measured data for the very same sample i . Since we use a time-averaged plume model, the sample needs to be represented by the mean of the sensed concentrations over a fixed time window. Therefore, to gather one sample, the robot stops at each target point for 5 seconds and then calculates the average of the sensed values gathered at 10 Hz (i.e. 50 concentration values in total).

Since there are five parameters to estimate, the posterior probability density function has to be a five-dimensional matrix, which is very time consuming to be entirely calculated. The solution to this problem would be to use an approximation algorithm such as the Markov Chain Monte Carlo (MCMC) [19] that allows evaluating the posterior probability function through efficient sampling.

In this work, we use the Metropolis–Hastings method [20]. The important factors in this method are the number of iterations and the proposal distribution, which is chosen as a 5-D Gaussian function, as the posterior probability function is also 5-D. Since this algorithm has already been used with STE algorithms in the literature (e.g. [21]), here we only present the concept from a general perspective.

At each iteration, a candidate point p in the proposal distribution is chosen, which is equivalent to selecting a set of parameters $p = \{Q_p, x_p, x_p, a_p, b_p\}$. Then, the likelihood of this point $P(p|D)$ is calculated and compared to the one of the previous point. The higher is the likelihood of the new point compared to the previous one, the higher is the chance of accepting it. In the case it is accepted, its likelihood is saved on the posterior probability distribution and replaces the previous point as the mean of the proposal distribution.

C. Navigation

The goal of the navigation method is to feed the estimation part with worthwhile information. Therefore, it needs to predict which point is the richest in terms of information to send the robot to. For this purpose, we propose to use POMDP [22] as a predictive navigation method to guide the robot in a profitable fashion.

POMDP requires three components for its operation: a posterior distribution, a set of possible actions, and a reward function. The posterior distribution is already given by the estimation part. As for the set of actions, to simplify the robot's motion, we limit the movement on one axis at a

time. Since the movement is possible in both positive and negative directions on the two axes, the set of possible actions becomes four-fold in a 2-D environment. The most essential component of this navigation method is the reward function, as it determines the behavior of the robot. In this work, it is chosen to be the relative entropy (also known as Kullback–Leibler divergence) [23] which represents the gain of information from one probability density function P to another Q , defined in Eq. (5). In our case, P would be the probability density function obtained through the estimation part, and Q the predicted probability density function that is calculated separately for each of the target points.

$$D_{KL}(P||Q) = \sum_j P(j) \log \frac{P(j)}{Q(j)} \quad (5)$$

Thus, at each step, once the four possible target points are determined, the robot will first predict the concentration that would be measured in each of them, using the current estimation of the model parameters. Then, it calculates the potential update of the posterior probability function using the Bayesian formulation, which would result in the predicted posterior probability function Q . Finally, it evaluates the gain of information at each point using Eq. (5).

The chosen target point leads the robot to the direction with the best gain of information. However, in the case where the robot does not have any information about the plume, i.e., no concentration is sensed, no direction would give more information than others. Hence, the robot tends to serially sample in all directions, and, as a result, it stays in the same region for a long time. To make the search more global, we need a component in the navigation strategy that promotes more exploration at the beginning, and leads the robot gradually towards the source when the estimation becomes more accurate. This indicating index can be seen in the evolution of the maximum a posteriori value of the source position.

Indeed, while the value of entropy is around its maximum (i.e., very little information is available), the maximum a posteriori value remains very unstable, but it converges towards the ground truth, similarly to the expected value, when the estimation becomes more accurate.

Therefore, we propose to define the movement vector as a weighted sum of two vectors: one leading towards the target point that provides more information, given by Kullback–Leibler divergence calculation, \vec{V}_{KLD} , and one leading to the maximum a posteriori value of the source position \vec{V}_{source} .

$$\vec{V} = \alpha \vec{V}_{KLD} + (1 - \alpha) \vec{V}_{source} \quad (6)$$

Intuitively, one would suggest that the algorithms should lead the robot towards the expected source position, as opposed to the maximum a posteriori. However, the expected source position never promotes exploration and might entrap the robot in an equilibrium. The coefficient α in Eq. (6) determines how global or local the search would be. Its optimized value is studied in Section III.



Fig. 2. Simulation environment in Webots, with the source upwind, the robot downwind, and the odor patches represented with blue hexagons.

D. End of Algorithm

As mentioned above, the algorithm stops when the uncertainty on the source parameter estimation becomes very low. More concretely, we calculate the entropy on the posterior probability function, which indicates the amount of uncertainty, and, when it goes below a certain threshold, the algorithm supposes to have enough information to stop and start the source declaration procedure.

Additionally, there is also a timeout that forces the algorithm to stop when the estimation takes more time than expected which is represented by a pre-established total number of iterations.

III. PERFORMANCE EVALUATION

In order to validate the performance of the algorithm, we first implemented and evaluated the method in simulation in a 2-D environment, and then in the physical reality with a mobile robot. In the present section, we explain the experimental procedure as well as the outcome.

A. Simulation Experiments

For the simulation experiments, we used Webots [24], which is an open-source high-fidelity robotics simulation software. Our simulation environment was extended with an odor dispersion plugin [25] which allows for a reasonably realistic simulation of wind and odor plume, based on the filament-based atmospheric dispersion model proposed in [1]. The simulated wind flow is quasi-laminar and stationary in its intensity and direction (i.e., no meandering and wind gusts). We also used a simulated Khepera III robot, equipped with an olfaction and anemometer sensor. A view of the simulated environment can be seen in Figure 2.

To make the performance evaluation as fair as possible, we randomly set the initial position of the robot within the entire arena. The position of the source remains on extreme upwind direction, with 1 m distance from the wall on the X-axis, but is completely randomized on Y-axis. This choice ensures that the setup is as challenging as possible.

Figure 3 shows a sample trajectory of the robot. In this example, the robot started far from the source. At the beginning of the experiment, large steps were taken to explore the environment. When the uncertainty gradually decreased, the maximum a posteriori value converged to the expected value and as a result the steps became smaller. Finally, for the uncertainty to become negligible, the robot sampled several points around the source position area before stopping the algorithm.

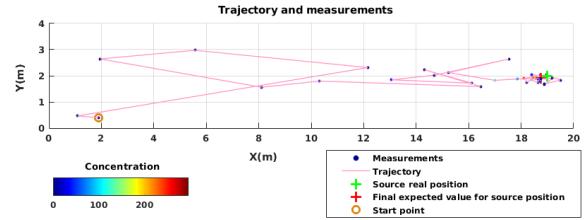


Fig. 3. An example of trajectory performed in simulation. The start point, the trajectory, the measurement points as well as the source position is visible on this figure. The search ended very close to the source.

TABLE I

ENVIRONMENTAL CONDITION SETUP

Label	A	B	C	D
Wind Speed	Low	Low	High	High
Release Rate	Low	High	Low	High

1) *Algorithmic Parameters:* The presented algorithm has three essential parameters that impact the performance: model and measurement errors considered in the likelihood, σ_M and σ_D in Eq. (4), that influence the uncertainty on the estimation and the ratio α between exploration and exploitation in the navigation part. In the simulation, we assumed to have no measurement error (i.e., $\sigma_D = 0$), firstly because in our simulation we could set the measurement noise to 0 and secondly because we wanted to calculate the model error σ_M in our setup. The value of measurement error σ_D was studied in physical experiments, as presented in Section III-B. In the simulation, we evaluated the performance of the algorithm using different values of the two remaining parameters to find the best set of values.

We considered two different metrics for the performance evaluation: first, the number of iterations (i.e. number of samples) that it takes the robot to reach the end of the algorithm; second, the estimation error on the source position on the X- and Y-axis once the source is declared. The algorithm estimates other parameters of the model as well, namely Q and σ_y , but their values are not of primary interest for us as we focused on the performance of the robot as opposed to the faithfulness of the model to the physical transport phenomenon. Both metrics deliver positive values, with the lower the value, the better the performance. For each set of values, we repeated the simulation 10 times with the same environmental conditions, where the wind speed was set to 0.9 m/s and the source release rate to 5%. This condition is labeled C in Table I and the actual values are mentioned in Table II.

The results are presented in Fig. 4. On each box, the central mark indicates the median, and the bottom and top

TABLE II

ENVIRONMENTAL PARAMETERS STUDIED IN SIMULATIONS AND WIND TUNNEL EXPERIMENTS

Parameter	Tested values
Wind speed (m/s)	Simulation: 0.2 (low), 0.9 (high) Wind tunnel: 0.2 (low), 1.0 (high)
Source release rate	Simulations: 5% (low), 10% (high) Wind tunnel: 9% (low) and 18% (high)

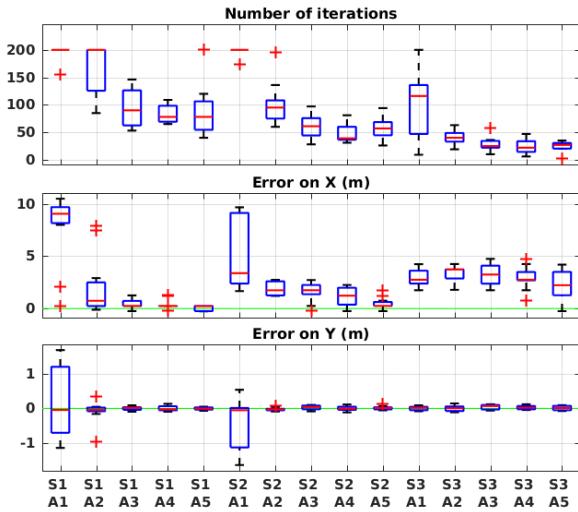


Fig. 4. Evaluation results in simulation with different algorithmic parameters. See Table III for the meaning of the labels.

TABLE III

LABELS ASSOCIATED WITH ALGORITHMIC PARAMETERS VALUE

Label	S1	S2	S3	Label	Sd1	Sd2	Sd3
σ_M value	50	30	10	σ_D value	200	250	300
Label	A1	A2	A3	A4	A5	α value	
	0.9	0.7	0.5	0.3	0.1		

edges of the box indicate the 25th and 75th percentiles of the results, respectively. The whiskers extend to the most extreme data points without considering the outliers, which are plotted individually using the “+” symbol. The parameter values corresponding to the labels are presented in Table III. The results show that the likelihood model error σ_M has a direct impact on the number of iterations. Naturally, the more uncertainty is considered, the more time the algorithm needs to converge. However, too low values also have negative impact on the estimation of the X value of the source position. The justification is that the pseudo-Gaussian plume model that we use in our algorithm does not entirely match the model used in the simulation odor dispersion plugin. Therefore, we need to allow for some values of model error. Given the results, $\sigma_M = 30$ seems to be the best compromise between the two metrics.

As for the exploration-exploitation ratio α , when at the highest value A1 (high exploration), it visibly perturbs the performance. On lower values (A2–A5), on the other hand, it appears to affect mostly the number of iterations, but very slightly. Therefore, we chose the middle value 0.5, which means 50% ratio between exploration and exploitation.

2) *Environmental Parameters*: Once the algorithmic parameters were selected as above, we studied the performance of the method in different environmental conditions. We considered two environmental parameters that have a high impact on the complexity of the problem: wind speed and source release rate.

The results of these evaluations are presented in Figure 5. Apart from a few outliers, the error is mostly very low on the source position and the number of iterations seem to

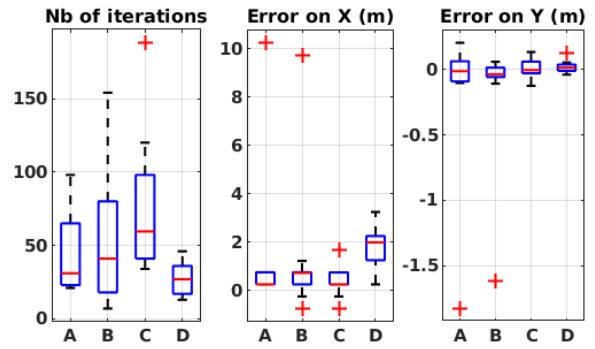


Fig. 5. Evaluation results in different environmental conditions in simulation. Refer to Table I for the meaning of the labels.

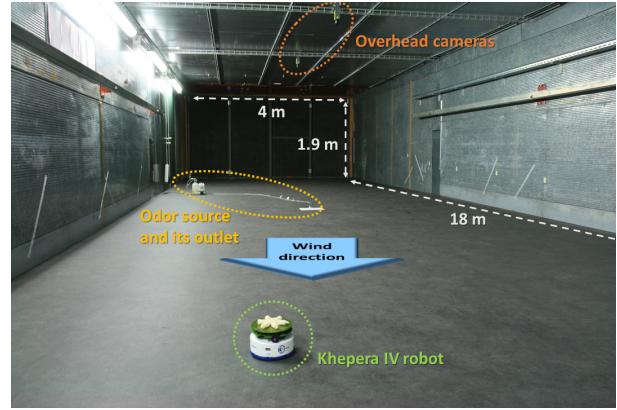


Fig. 6. A view of the wind tunnel environment, along with a Khepera IV robot, the emulated odor source as well as the overhead cameras

be acceptable in all environmental conditions. The error is around 15 cm on the Y-axis. On the X-axis, on the other hand, the error is usually around 1 m. We suspect that the issue could come from the granularity for estimating the posterior probability density function, which is 0.5 m on X, while 0.2 m on Y. From a general point of view, however, we can see that the performance of the algorithm is very similar in all the tested environmental conditions. This result was expected, since the underlying model of the experiment adapts itself to both wind speed, which is captured using an anemometer, and release rate, which is estimated during the experiment by the algorithm.

B. Wind Tunnel Experiments

To evaluate the performance of the algorithm in a repeatable fashion, our experiments were carried out in a wind tunnel of volume $18 \times 4 \times 1.9 \text{ m}^3$, which provides a controllable wind flow. A Khepera IV robot, equipped with an olfaction sensor MiCS-5521 CO/VOC [26] as well as a wind sensor board [4], ran the algorithm autonomously. Overhead cameras in combination with the SwisTrack software [27] were used for tracking the robot pose in the wind tunnel, which is sent back to the robot to be used in the algorithm. The odor source was emulated using an electric pump vaporizing ethanol. An illustration of the wind tunnel with the deployed equipment is presented in Fig. 6.

Fig. 7 shows a sample trajectory of the robot in the wind

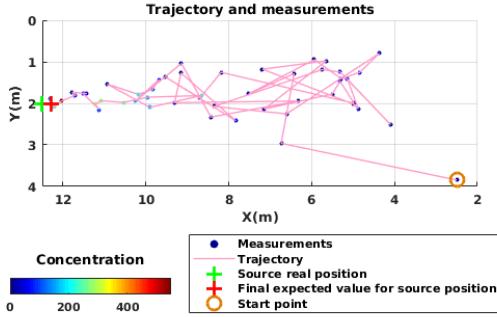


Fig. 7. An example of trajectory performed in the physical experiments. The start point, the trajectory, the measurement points as well as the source position are visible in this figure. The search ended very close to the source.

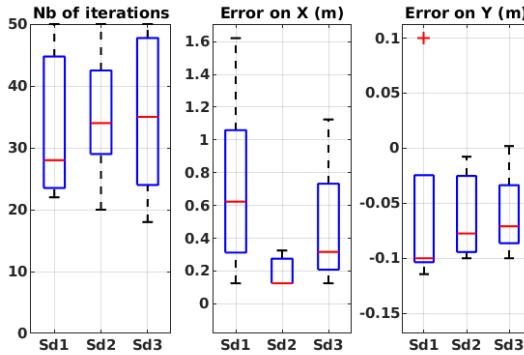


Fig. 8. Evaluation results with different values of σ_D . See Table III for the meaning of the labels.

tunnel. Similar to the simulation experiments, first, the robot started to explore the environment by taking large steps. Once it found the plume and had less uncertainty about the source parameters, the steps became smaller and the overall movement was oriented towards the source. The final expected source position was only a few centimeter away from the real source position.

Unlike in simulation, in the physical experiments, it is inevitable to have measurement errors. Therefore, in our wind tunnel experiments, we studied the impact of the parameter σ_D of Eq. (4), the standard deviation of the measurement error. In these experiments, the parameters σ_M and α were fixed to the values selected from the simulation results presented in Section III-A.1.

Figure 8 shows the performance of the algorithm with different values of σ_D . In these experiments, the maximum number of iterations was fixed to 50. These results show a subtle trend: with increasing σ_D value, the error on the X- and Y-axis decrease, but the average number of iterations increases. We chose to continue the experiments with the middle value $\sigma_D = 250$ which shows a very low error value on the X-axis and seems to be a good compromise between error and number of iterations.

Similar to the simulation, the algorithm was also evaluated in the environmental conditions listed in Table II. Each set of experiment was carried out five times.

Figure 9 shows the results of the experiments in different

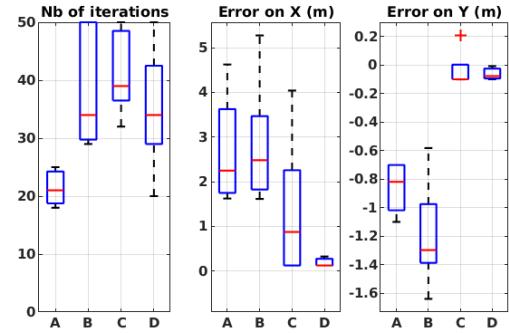


Fig. 9. Results of evaluation in wind tunnel experiments in different environmental conditions. Refer to Table I for the meaning of the labels.

environmental conditions. In most of the experiments, there is a larger error on the X-axis compared to Y-axis, similar to the simulation results. In these experiments, the amount of error is higher compared to the simulation, mainly because the maximum number of iterations was arbitrarily fixed to 50, but also because of different irregularities that happen in physical experiments. In all cases, the mean error on the X-axis is less than 2.5 m and on the Y-axis less than 1.5 m. Although the algorithm seems successful in all the tested configurations, we can see that in high wind speed (conditions C and D) the average error is much less compared to low wind speed conditions. One main reason could be the fact that the algorithmic parameters were optimized in high wind speed conditions. Further evaluations in more extreme conditions would be necessary to claim that the algorithm is robust to all environmental conditions.

IV. CONCLUSION

We successfully developed an odor source localization algorithm based on the STE method. The algorithm was evaluated both in simulation and physical reality. The impact of different algorithmic parameters were studied and the sensitivity of the method to different environmental configurations, namely wind speed and source release rate, was evaluated.

One of the advantages of this algorithm was its ability to cover the three sub-tasks of odor source localization problem. The enhanced navigation methods proposed in this paper allowed for an efficient search for the plume acquisition part. The amount of uncertainty associated with the estimation ensured the source validation sub-task as well.

Thanks to the MCMC method, the high computational cost related to probabilistic algorithms was remarkably reduced and the algorithm could be run entirely on a Khepera IV robot.

Further results, as well as the adaptation of this algorithm to an environment without a global positioning system can be found in [28].

In the future, we plan to evaluate and adapt this algorithm to other airflow profiles, for instance when no wind flow is present or when it is turbulent. The method has also the potential to be deployed on a static sensor network or on a multi-robot system.

REFERENCES

- [1] J. A. Farrell, J. M.“Filament-Based Atmospheric Dispersion Model to Achieve Short Time-Scale Structure of Odor Plumes,” *Environmental Fluid Mechanics*, no. 1, pp. 143–169.
- [2] B. L. Villarreal, G. Olague, and J. L. Gordillo, “Synthesis of odor tracking algorithms with genetic programming,” *Neurocomputing*, pp. 1019–1032.
- [3] A. Hayes, A. Martinoli, and R. M. Goodman, “Distributed Odor Source Localization,” *IEEE Sensors Journal*, vol. 2, no. 3, pp. 260–271, 2002.
- [4] T. Lochmatter, “Bio-inspired and probabilistic algorithms for distributed odor source localization using mobile robots,” PhD Thesis No. 4628, EPFL, 2010.
- [5] V. Genovese, P. Dario, R. Magni, L. Odetti, and S. Anna, “Self-organizing behavior and swarm intelligence in a pack of mobile miniature robots in search of pollutants,” in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 1992, pp. 243–248.
- [6] 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.
- [7] V. H. Bennetts, A. J. Lilienthal, P. Patrick, and M. Trincavelli, “Mobile robots for localizing gas emission sources on landfill sites: is bio-inspiration the way to go?” *Bioinspired solutions to the challenges of chemical sensing*, vol. 4, p. 164, 2012.
- [8] J. M. Soares, A. P. Aguiar, A. M. Pascoal, and A. Martinoli, “A distributed formation-based odor source localization algorithm-design, implementation, and wind tunnel evaluation,” in *IEEE International Conference on Robotics and Automation*, 2015, pp. 1830–1836.
- [9] M. Vergassola, E. Villermaux, and B. I. Shraiman, “Infotaxis’ as a strategy for searching without gradients,” *Nature*, vol. 445, pp. 406–409, 2007.
- [10] 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*, no. 3, pp. 281–292.
- [11] M. Hutchinson, H. Oh, and W. H. Chen, “A review of source term estimation methods for atmospheric dispersion events using static or mobile sensors,” *Information Fusion*, pp. 130–148.
- [12] M. Reggente and A. J. Lilienthal, “The 3D-Kernel DM + V / W Algorithm : Using Wind Information in Three Dimensional Gas Distribution Modelling with a Mobile Robot,” pp. 999–1004, 2010.
- [13] B. Ristic, M. Morelande, and A. Gunatilaka, “Information driven search for point sources of gamma radiation,” *Signal Processing*, no. 4, pp. 1225–1239.
- [14] A. Keats, E. Yee, and F. S. Lien, “Bayesian inference for source determination with applications to a complex urban environment,” *Atmospheric Environment*, vol. 41, no. 3, pp. 465–479, 2007.
- [15] R. Madankar, P. Singla, and T. Singh, “Optimal information collection for source parameter estimation of atmospheric release phenomenon,” *Proceedings of the American Control Conference*, pp. 604–609, 2014.
- [16] B. Ristic, A. Skvortsov, and A. Gunatilaka, “A study of cognitive strategies for an autonomous search,” *Information Fusion*, pp. 1–9.
- [17] C. Huang, T. Hsing, N. Cressie, A. R. Ganguly, V. A. Protopopescu, and N. S. Rao, “Bayesian source detection and parameter estimation of a plume model based on sensor network measurements,” *Applied Stochastic Models in Business and Industry*, vol. 26, no. January, pp. 331–348, 2010.
- [18] S. P. Arya, *Air pollution meteorology and dispersion*. Oxford University Press, 1999.
- [19] W. R. Gilks, S. Richardson, and D. Spiegelhalter, *Markov chain Monte Carlo in practice*. CRC press, 1995.
- [20] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, “Equation of state calculations by fast computing machines,” *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.
- [21] M. Hutchinson, H. Oh, and W.-H. Chen, “Adaptive bayesian sensor motion planning for hazardous source term reconstruction,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 2812–2817, 2017.
- [22] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, “Planning and acting in partially observable stochastic domains,” *Artificial Intelligence*, vol. 101, no. 1-2, pp. 99–134, 1998.
- [23] S. Kullback and R. A. Leibler, “On information and sufficiency,” *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.
- [24] O. Michel, “Webots TM : Professional Mobile Robot Simulation,” *International Journal of Advanced Robotic Systems*, vol. 1, no. 1, pp. 39–42, 2004.
- [25] Wikibooks, “Webots odor simulation — wikibooks, the free textbook project,” 2010, [Online; accessed 7-November-2016]. [Online]. Available: https://en.wikibooks.org/w/index.php?title=Webots_Odor-Simulation&oldid=1966420
- [26] SGX SensorTech technologies, “MiCS-5521 CO/VOC sensor,” <http://www.sgxsensortech.com>.
- [27] T. Lochmatter, P. Roduit, C. Cianci, N. Correll, J. Jacot, and A. Martinoli, “Swistrack - A flexible open source tracking software for multi-agent systems,” in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 4004–4010.
- [28] F. Rahbar, A. Marjovi, and A. Martinoli, “Design and performance evaluation of an algorithm based on source term estimation for odor source localization,” *Sensors*, vol. 19, no. 3, 2019.