2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

A Distributed Source Term Estimation Algorithm for Multi-Robot Systems

Faezeh Rahbar and Alcherio Martinoli

Abstract— **Finding sources of airborne chemicals with mobile sensing systems finds applications in safety, security, and emergency situations related to medical, domestic, and environmental domains. Given the often critical nature of all the applications, it is important to reduce the amount of time necessary to accomplish this task through intelligent systems and algorithms. In this paper, we extend a previously presented algorithm based on source term estimation for odor source localization for homogeneous multi-robot systems. By gradually increasing the level of coordination among multiple mobile robots, we study the benefits of a distributed system on reducing the amount of time and resources necessary to achieve the task at hand. The method has been evaluated systematically through high-fidelity simulations and in a wind tunnel emulating realistic and repeatable conditions in different coordination scenarios and with different number of robots.**

I. INTRODUCTION

Finding sources of chemical compounds released in the air using mobile sensing systems finds several applications in various critical situations. They range from finding explosives in security domains to search and rescue of survivors in case of natural hazards such as an earthquake.

Although odor source localization has been attracting a considerable attention from roboticists and environmental engineers in the past twenty years, it remains a challenging topic in realistic environments. The complex nature of the dispersion phenomenon is the main cause of the difficulty. According to [1], the dispersion is a combination of molecular diffusion that drives odor patches away from the source, as well as the advection which is caused by the airflow that carries the molecules in its direction. As a result of the dispersion, an odor plume is created that is shaped by the characteristics of the source as well as the airflow which, in turn, is affected by the environment.

A large variety of algorithms have already been developed for odor source localization. They can be classified into four often overlapping categories [2]: gradient-based, bio-inspired, formation-based, and probabilistic algorithms.

Gradient-based algorithms simply climb the concentration gradient to reach the source. This type of algorithms, while being the most intuitive and computationally light ones, they perform well only in presence a low-noise odor concentration signal which in turn requires a relatively long time to find the source [3].

Bio-inspired algorithms imitate the search strategies of living beings such as dogs, fish, moths, etc. [4]. Since they are very reactive algorithms, they do not need any a priori information about the environment and its atmospheric conditions, nor do they need to have a memory of the previous observations. 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 [5].

Formation-based algorithms have been recently designed for multi-robot systems [6]. They allow robots to sample multiple points at the same time, but since the coordination between agents is necessary, the method completely relies on inter-robot communication and relative positioning which could be challenging to maintain in a realistic environment.

Probabilistic algorithms use the observations made by the sensing system in the environment to create a belief on the source position in the form of a probability distribution function [2]. The belief is updated after every new observation using Bayesian estimation. The cycle continues until the probability distribution becomes a Dirac function. Infotaxis [7], particle-filter-based algorithms [8], and Source Term Estimation (STE) [9] are the most popular examples of these algorithms.

Despite being computationally more expensive than the four other categories, probabilistic algorithms have several advantages. Firstly, unlike the previously mentioned methods that provide only the source position, they are able to present a richer set of information about the environment (e.g., [10]) or the source characteristics [9]. Secondly, because of the probabilistic nature of the methods, the final result is associated with a measure of uncertainty, which shows how trustworthy the data are. Finally, 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).

Given the above mentioned advantages, we chose to leverage a STE algorithm in this work. 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 [11]). Neither is it limited to mobile sensor nodes, as it can be used with a static sensor network (e.g., [12]).

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

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.

978-1-7281-7395-5/20/$31.00 ©2020 IEEE

methods have already been used with STE in the literature. Partially Observable Markov Decision Processes (POMDP) [11] and mutual information maximization [13] are among the popular methods in this area.

Thanks to the flexibility of STE algorithm, it can also be used with a distributed sensing system. There could be several advantages in using a distributed sensing system for such task, for instance, reducing the necessary time to achieve the task, making use of diverse resources, or distributing sub-tasks among assets throughout the procedure. However, for a distributed sensing system to be advantageous in comparison to a single sensing asset, it must operate with an appropriate degree of coordination. Therefore it is important to understand how much the coordination helps the performance of the system.

In recent works, such as [14] and [15], STE-related algorithms are used with a distributed system to achieve source localization. However, in [15] the algorithm is only evaluated in simulation using bodiless mobile sensors as well as an odor field data base. Additionally, the mobile sensors are supposed to be working in a synchronous fashion, which could be problematic in a real environment. In [14] the algorithm is evaluated using real robots using a humid-air plume, but the motion coordination was fully centralized and the advantage of the coordination was not sufficiently investigated and thoroughly benchmarked with single robot scenario.

Hence, the first contribution of the present work is the evaluation of an STE algorithm, which was presented and evaluated in our previous work [16], with three different coordination strategies using up to three robots. The evaluation was done in a high-fidelity simulator as well as in a wind tunnel. Additionally, the previously used navigation strategy is enhanced to reduce the travelled distance even for a single robot. The new navigation strategy is evaluated separately along with its key parameters.

In the remaining of this paper, we will present the global structure of the algorithm in the next section. More detailed explanation is available in our previously published works [17] [16]. Then we explain the evaluation process and present the obtained results. Lastly, in the conclusion, we will discuss the outcome and present the outlook for this work.

II. Method

As mentioned in the introduction, this work is based on STE which is a probabilistic framework that often regroups different strategies. The ultimate goal of an STE algorithm, independently of the strategy, is to estimate the characteristics of the plume which always includes the origin, i.e. the source position.

When deployed on robotic assets, the general structure of an STE algorithm consists of two main parts: estimation and navigation [16]. The estimation part relies on an analytical plume model whose parameters have to be estimated throughout the experiment using the data that the robot acquires. In the navigation part, the algorithm seeks to maximize the amount of obtained information by sending the robot to the most informative point in the arena. When a distributed team of mobile robots is involved, the navigation part of the algorithm needs to coordinate the moves with the other robots as well. The cycle between estimation and navigation continues until the uncertainty in the parameter estimation becomes negligible for at least one of the robots.

In this work, we have augmented the previous single-robot algorithm [16] with collaborative strategies for a homogeneous multi-robot system. In the remaining of this section, we first briefly present different components of the single-robot algorithm and then we explain the scenarios involving a multi-robot system.

A. Plume Model

For this work, the plume model on which the estimation relies is chosen to be the pseudo-Gaussian concentration plume model [18] which describes the time-averaged concentration model for a continuous point source in a laminar flow. It is presented in Eq. (1), where Q is the source release rate, $\bar{u}$ the average wind speed, (x_s, y_s) the source position, and σ_y the standard deviation of odor dispersion in the crosswind direction (Y-axis), which is simplified to be a linear function of the upwind distance from the source $(x - x_s)$.

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

Assuming the airflow to be aligned with the X-axis, this equation is applied to 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., $\forall x < x_s$).

Moreover, since we are leveraging a 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 varies along the X-axis, it can be modeled with the following affine function: $\sigma_y(x - x_s) = a(x - x_s) + b$. Thus, five parameters remain to be estimated, making the problem five-dimensional. The range and resolution of all the parameters are empirically chosen.

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. Also, we consider the prior $P(m)$ a uniform distribution in between the limits of each parameter. Therefore, the posterior $P(m|D)$ will be proportional to the likelihood $P(D|m)$ in the parameters limit, 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 [12] as follows:

$$P(D|m) \propto exp\Bigg(-\frac{1}{2}\sum_i \Bigg(\frac{(D_i - C_i(m))^2)}{\sigma_M^2 + \sigma_D^2} \Bigg) \Bigg) \quad (3)$$

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. (3) is applied on all samples D_i that the robot gathers during the experiment. $C_i(m)$ is the concentration determined by the plume model for a set of source parameters m for sample i.

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–Hasting 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, since the posterior probability function is also 5-D.

C. Navigation

The main goal of the navigation method is to feed the estimation part with worthwhile information that lead to a as precise and as quick as possible localization of the source. Therefore, it needs to predict which neighboring point is rich in terms of information to send the robot to. For this purpose, in our previous work [16] we defined the movement vector as a weighted sum of two vectors: the one leading towards the target point that provides more information, given by Kullback–Leibler divergence calculation, $\vec{V}_{KLD}$ and the vector that leads to the maximum a posteriori value of the source position $\vec{V}_{source}$.

$$\vec{V} = \alpha\vec{V}_{KLD} + (1-\alpha)\vec{V}_{source} \quad (4)$$

The optimal value for the coefficient $\alpha \in [0, 1]$ was deduced to be 0.5 according to experimental results.

In this work however, with the intention of reducing the travelled distance by the robot, we also take into account the distance of candidate points from the current robot's location. In fact, when the amount of uncertainty is still high because of lack of information, due to the stochastic nature of the MCMC algorithm, the maximum a posteriori value of the source position could change very quickly. This, indeed, promotes exploration, but makes the robot travel very far while a closer point could have a similar value.

Therefore, instead of leading the robot towards the maximum a posteriori value of the source position, we create a combined map of two density functions defined on every grid cell of the environment map: $P_M(x_s, y_s)$, defined in Eq. (5), is the marginal probability density function for the source position, and $D_I(x, y)$ represents the inverse of the distance of every point to the current position of the robot, as defined in Eq (6).

TABLE I
SUMMARY OF COORDINATION STRATEGIES

Strategy	Nb of robots	Coordination
Individualist	1, 2, 3	Source Declaration
Cooperative	2, 3	Source Declaration Sample Sharing
Collaborative	2, 3	Source Declaration Sample Sharing Movement Coordination

$$P_M(x_s, y_s) = \sum_Q \sum_{\sigma_a} \sum_{\sigma_b} P(Q, x_s, y_s, \sigma_a, \sigma_b) \quad (5)$$

where $P(Q, x_s, y_s, \sigma_a, \sigma_b)$ is the posterior density function for all the source parameters.

$$D_I(x, y) = 1/\sqrt{(x-x_r)^2 + (y-y_r)^2} \quad (6)$$

where (x_r, y_r) is the current position of the robot. D_I is normalized in such a way that the sum of all the values will be equal to 1. D_I is set to 0 for the cell that the robot is currently positioned on, in order to avoid division by 0.

In order to combine the two density functions we use a weighted sum defined in Eq.7 where $\beta \in [0, 1]$.

$$P_{combined}(x, y) = \beta P_M(x, y) + (1-\beta)D_I(x, y) \quad (7)$$

The combined density function contains higher values for points that have a high potential of containing the source and that are closer to the robot. Therefore, if two points have the same probability of containing the source according to the posterior, $P_{combined}$ attributes a higher value to the one that is close to the current position of the robot. Once the maximum point of $P_{combined}$ is found, $\vec{V}_{source}$ of eq. (4) will point to it.

The value of coefficient β is determined experimentally and will be discussed in Section III.

D. Coordination Strategies

There are multiple stages on which the robots can join forces in this framework: sharing samples, coordinating navigation, and source declaration. For a better understanding of the benefit of each collaboration technique, we suggest different strategies with different levels of collaboration. Then each strategy is evaluated with up to three robots. The characteristics of the strategies are explained in the following and a short summary is given in Table I.

1) Individualist Strategy: In this scenario, every robot performs the algorithm independently, i.e. they do not share any information with the other teammates. However, the source declaration of one robot is enough for the entire team to end the search. Indeed, although this strategy involves a minimalistic level of coordination, the robots still form a team, and if one of the teammates reaches the required level of certainty to declare the source, the other ones do not need to continue the search.

2) Cooperative Strategy: Cooperation means sharing information between teammates in support of each other's goals. Therefore, in this scenario, the robots share their acquired samples with the other robots. Since we suppose a reliable communication among the robots, they are all supposed to have the same data to calculate the posterior. However, since the estimation is done stochastically using the MCMC algortihm, the outcomes could be slightly different from one robot to another. In this scenario, no coordination is considered for navigation but the source declaration is done as a team like in the individualist scenario.

3) Collaborative Strategy: Collaboration is working together in support of a shared goal. In the collaborative scenario, the robots not only share their samples with each other as in the cooperative strategy, but also coordinate their movements together.

The movement coordination can be defined in different ways. Here, with the intention of saving time and resources for the robots, we chose to make use of a task allocation technique: considering that the current goal of each robot is a task, it can be achieved by any other robot, since any collected sample is shared with all the robots. Therefore, every time a robot decides a new goal position, it verifies if another way of distributing tasks between the robots would decrease the total travelled distance by the team. If so, it makes the necessary swaps and informs the involved teammates to change their goals. To achieve this, the robots need to share their current position, as well as their current goal position with each other at all times. The swap can only happen for a robot if it is moving towards its goal.

Additionally, in this scenario, since the robots have the information about their teammates' current goal positions, they will take them into account before deciding their next move. In the estimation part, they temporarily include a prediction for the outcome of the samples that the other robots are going to collect in their goal positions and in this way they reduce the risk of going to the same point.

Similarly to the cooperative scenario, the communication is assumed fully reliable and fast. In any case, the algorithm success does not depend on inter-robot communication. However, the advantages brought by coordination cannot be leveraged until the communication failure is resolved.

E. End of Algorithm

As mentioned above, the algorithm stops when the uncertainty on the source parameter estimation becomes very low. More concretely, the uncertainty indicator is the entropy of the posterior probability function. When it goes below a certain threshold the algorithm supposes to have all the information it needs to declare the source position.

When the source is declared by one robot, all the other robots stop the process as well. The entropy could be slightly different for different robots even when all the acquired data are shared because of the stochasticity of estimation caused by the MCMC algorithm. However, as previously explained, when one of the teammates is certain of its estimation, it is assumed to be enough for the entire team.

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. This timeout, while being the same for all the robots, does not necessarily happen for all of them at the same time because of the assynchronicity of the team. When one robot reaches its timeout, it does not force the others to stop as well, they can carry on hoping that one of them reaches the certainty threshold before the timeout.

III. Performance Evaluation

In order to validate the performance of the algorithm, we first implemented and evaluated the method in a high-fidelity simulation, and then in the physical reality with mobile robots.

The metrics used for the evaluations are as follows:

1) Number of iterations: it is the sum of the number of iterations of all the robots, which measures the total computational cost.
2) Experiment duration: it is the time measured from the start of the experiments until the first robot declares success and ends the experiment. This metric measures the response time of the method.
3) Travelled distance: it is the sum of the travelled distances by all the robots, which measures the total energy cost.
4) Estimation error: it is the Euclidean distance between the real source position and the estimated one. This metric measures the accuracy of the method.

In all four metrics, lower values are better. For single-robot experiments, the experiment duration is not shown because it is proportional to the travelled distance. In strategy comparisons, estimation error is not shown because they were all similarly very low.

In the remaining of this section, we explain the experimental procedure as well as the outcome.

A. Simulation Experiments

For the simulation experiments, we used Webots [21], which is an open-source high-fidelity robotics simulation software. Our simulation environment was extended with an odor dispersion plugin [22] which allows for a reasonably realistic simulation of wind and odor plume, based on the filament-based atmospheric dispersion model proposed in [23]. 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 IV robot, equipped with an olfaction and anemometer sensor. A view of the simulated environment can be seen in Fig. 1.

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.

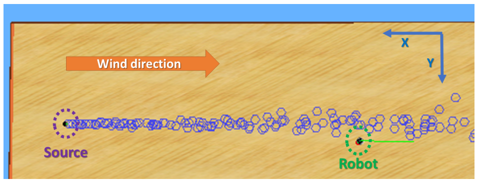

Fig. 1. Simulation environment in Webots, with the source upwind, the robot downwind, and the odor patches represented with blue hexagons.

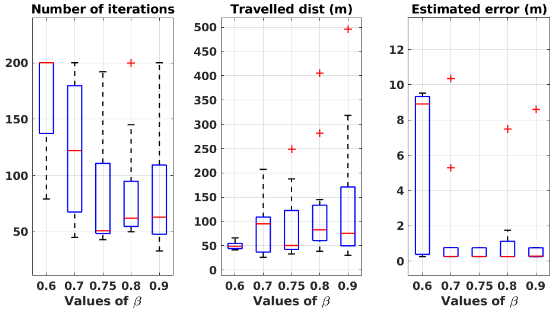

Fig. 2. Performance evaluation in simulation with different values for β in eq. (7).

1) Algorithmic Parameter: Before evaluating the algorithm in different scenarios, we studied the influence of the ratio β from Eq. (7). Using a single robot, we run ten experiments for different values of the parameter β. The results presented in Fig. 2 show that the best compromise value is 0.75, since it reduces the travelled distance, but lets the robot travel far enough to find the source with a low error.

Fig. 3 shows a sample trajectory of one robot on simulation. In this example, the robot started far from the source and outside the plume. At the beginning of the experiment, large steps were taken to explore the environment to find the plume. Once the uncertainty gradually decreased, 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.

2) Coordination Strategy Evaluation: Once the best value for the algorithmic parameter is found, we studied the performance of the algorithm using the different strategies that we defined in Section II-D. Each coordination strategy is evaluated with up to three robots, and each experiment is repeated ten times. Fig. 4 shows the results of this evaluation.

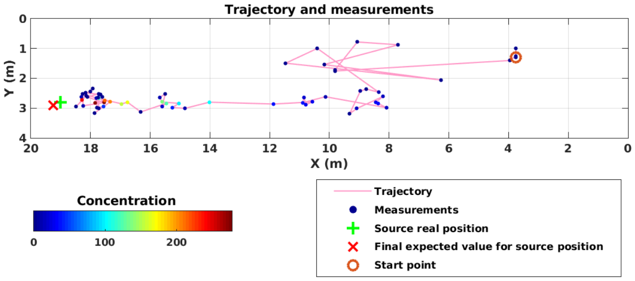

Fig. 3. An example of trajectory performed in simulation. The start point, the trajectory, the measurement points as well as the source position are shown on this figure. The search ended very close to the source.

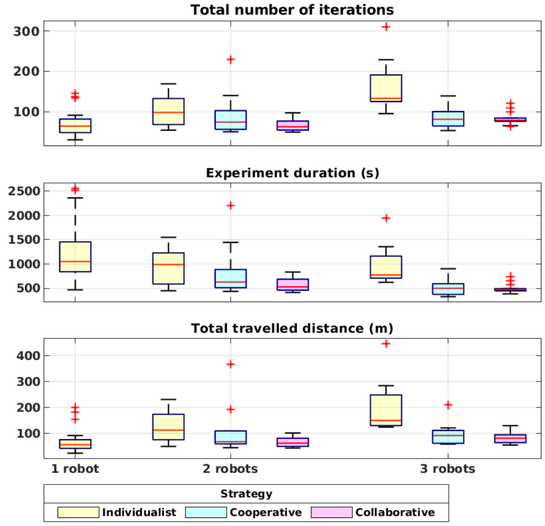

Fig. 4. Evaluation results of the algorithm in simulation with the strategies defined in Table I.

In the individualist strategy, the total number of iterations and the total travelled distance increase with the number of robots. The reason is that in the absence of coordination, more active robots travel more distance together compared to a single robot. The experiment duration, however, does not follow this trend. Its median value stays similar, while the variation drops with higher numbers of robots. This is due to the source declaration that is done as a team. With higher number of robots, and due to the stochasticity of the estimation process, there is a higher chance that a robot arrives at the required certainty level and declares the source position. Therefore, even with no coordination among the robots throughout the experiment, having more robots may reduce the necessary time to locate the source.

In the cooperative strategy, since the robots share the acquired data, they tend to reach the same conclusion about the estimated source position. The median of the number of iterations does not change much compared to the case of a single robot. This is due to the fact that the total number of iterations is equal to the number of acquired samples, and since all the collected samples are accessible for all robots, their number should be in the same range as the one of a single robot. For the same reason, the sum of all travelled distances follows the same trend. The experiment duration, however, decreases with higher number of robots since different points of the environment can be sampled simultaneously.

The collaborative strategy follows the same trends as the cooperative strategy in all metrics, in comparison with the individualist strategy. A slight reduction in travelled distance and experiment duration can be seen when compared to the cooperative strategy which is due to the task allocation technique. Based on intuition, we believe that if the distance component was not introduced in the navigation, the steps would be longer (see our previous work [16]) and therefore, we could see a higher influence of the collaborative strategy when compared to the cooperative one.

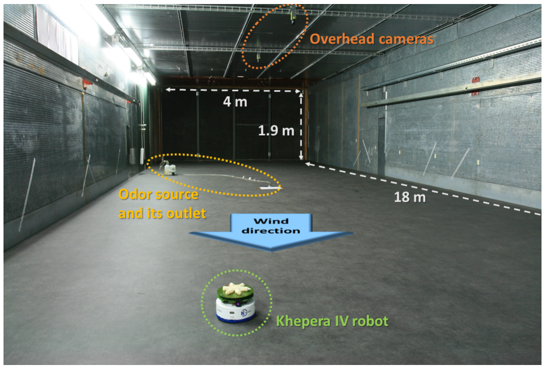

Fig. 5. A view of the wind tunnel environment, along with a Khepera IV robot, the odor source as well as the overhead cameras

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 $18 \times 4 \times 1.9\ \mathrm{m}^3$. The odor source was represented by an electric pump vaporizing ethanol. A team of Khepera IV robots, equipped with an olfaction sensor MiCS-5521 CO/VOC [24] as well as a wind sensor board [2], ran the algorithm autonomously in a fully distributed fashion. Overhead cameras in combination with the SwisTrack software [25] 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 communication between the robots was done via WiFi, using a UDP protocol. In order to avoid packet losses, we developed an acknowledgement-based reliable protocol on top of UDP. Fig. 5 shows a view of the wind tunnel with the deployed equipment.

Fig. 6 shows one sample trajectory for each strategy with two robots in the wind tunnel. Similarly to the simulation experiments, in all the strategies, first, the robots starts to explore the environment by taking large steps. Once they find the plume and have less uncertainty about the source parameters, the steps become smaller and the overall movement is oriented towards the source. The final expected source position is usually very close to the real one. It is worth noting that the more coordination is involved in the experiment, the smoother the trajectories become. In the individualist scenario, both robots are alone in performing the algorithm, therefore each of them needs to collect enough samples individually. In the cooperative scenario, the data are shared, therefore each robot needs to collect less samples and as a result, they progress faster toward the source. In the collaborative scenario, the trajectories rarely cross each other, which makes the robots to have less collision risk and less travelled distance.

Fig. 7 shows the performance of the algorithm in different strategies with different number of robots. Each set of experiment was repeated 5 times. Similarly to the simulation results, the experiment duration decreases with higher levels of coordination and higher number of robots. Travelled distance and number of iterations substantially increase in the individualist scenario, but by very little in the cooperative and collaborative scenarios.

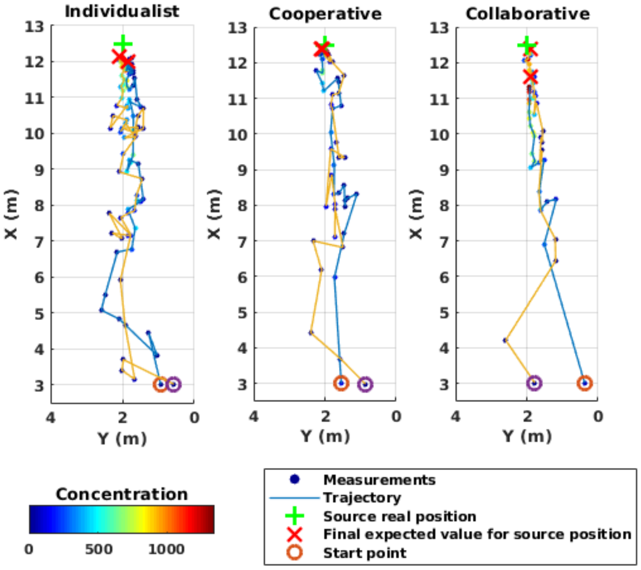

Fig. 6. An example of trajectory performed in the physical experiments. The start point, the trajectory, the measurement points as well as the source position are visible in this figure. The search ended very close to the source.

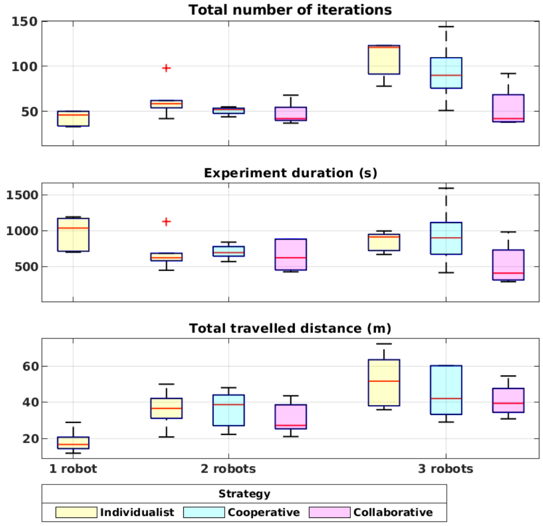

Fig. 7. Evaluation results of the algorithm in the physical experiments with the scenarios defined in Table I.

IV. Conclusion

We successfully evaluated a distributed implementation of a STE algorithm using a multi-robot system. The system was evaluated both in simulation and physical reality. The enhanced navigation strategy allowed for a smoother trajectory compared to our previous work [16]. Different coordination strategies were proposed in order to verify the advantage and constraints of using a distributed sensing system and different coordination strategies.

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, and the coordination of a larger number of possibly heterogeneous sensing assets.

References

[1] B. L. Villarreal, G. Olague, and J. L. Gordillo, “Synthesis of odor tracking algorithms with genetic programming,” *Neurocomputing*, pp. 1019–1032.

[2] T. Lochmatter, “Bio-inspired and probabilistic algorithms for distributed odor source localization using mobile robots,” PhD Thesis No. 4628, EPFL, 2010.

[3] V. Genovese, P. Dario, R. Magni, L. Odetti, and S. Anna, “Self-organizing behavior and swarm intelligence in a pack ofmobile miniature robots in search of pollutants,” in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 1992, pp. 243–248.

[4] 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.

[5] 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.

[6] 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.

[7] M. Vergassola, E. Villermaux, and B. I. Shraiman, “ Infotaxis ’ as a strategy for searching without gradients,” *Nature*, vol. 445, pp. 406–409, 2007.

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

[9] 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.

[10] 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,” in *IEEE SENSORS*, 2010, pp. 999–1004.

[11] B. Ristic, M. Morelande, and A. Gunatilaka, “Information driven search for point sources of gamma radiation,” *Signal Processing*, no. 4, pp. 1225–1239.

[12] 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.

[13] R. Madankan, 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.

[14] J. R. Bourne, E. R. Pardyjak, and K. K. Leang, “Coordinated bayesian-based bioinspired plume source term estimation and source seeking for mobile robots,” *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 967–986, Aug 2019.

[15] M. Park and H. Oh, “Cooperative information-driven source search and estimation for multiple agents,” *Information Fusion*, vol. 54, pp. 72–84, 2020.

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

[17] 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, p. 656, Feb 2019. [Online]. Available: http://dx.doi.org/10.3390/s19030656

[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] O. Michel, “Webots TM : Professional Mobile Robot Simulation,” *International Journal of Advanced Robotic Systems*, vol. 1, no. 1, pp. 39–42, 2004.

[22] 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}

[23] J. A. Farrell, J. Murlis, X. Long, W. Li, and R. T. Cardé, “Filament-Based Atmospheric Dispersion Model to Achieve Short Time-Scale Structure of Odor Plumes,” *Environmental fluid mechanics*, vol. 2, no. 1-2, pp. 143–169, 2002.

[24] SGX Sensortech technologies, “MiCS-5521 CO/VOC sensor,” http://www.sgxsensortech.com.

[25] 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.