

Adaptive Lévy Taxis for Odor Source Localization in Realistic Environmental Conditions

Romain Emery, Faezeh Rahbar, Ali Marjovi and Alcherio Martinoli

Abstract—Odor source localization with mobile robots has recently been subject to many research works, but remains a challenging task mainly due to the large number of environmental parameters that make it hard to describe gas concentration fields. We designed a new algorithm called Adaptive Lévy Taxis (ALT) to achieve odor plume tracking through a correlated random walk. In order to compare its performances with well-established solutions, we have implemented three moth-inspired algorithms on the same robotic platform. To improve the performance of the latter algorithms, we developed a rigorous way to determine one of their key parameters, the odor concentration threshold at which the robot considers to be inside or outside the plume. The methods have been systematically evaluated in a large wind tunnel under various environmental conditions. Experiments revealed that the performance of ALT is consistently good in all environmental conditions (in particular when compared to the three reference algorithms) in terms of both distance traveled to find the source and success rate.

I. INTRODUCTION

Finding sources of airborne chemicals with mobile robots has many applications in various critical situations. In case of natural hazards, such as avalanches, landslides or earthquakes, following blood odor cues may help to find and rescue wounded humans. Security operations in airports related to explosives or drugs also involve tracking of chemical traces, as well as the removal of land mines or the identification of pollutant leakages. All these examples reveal the vast fields where robotic odor source localization may play a part to efficiently save human lives or protect their environment.

Chemical compounds released by a source are mainly advected by the airflow, forming a three-dimensional odor plume. As the plume travels away from the source, it becomes more diluted due to molecular diffusion and turbulence that mixes the odor molecules with clean air [1]. Therefore, addressing the odor source localization problem in robotics consists of three consecutive phases [2]. First of all, the robot needs to search for the odor plume in the environment. As soon as it detects a gas concentration higher than a given threshold, the *plume acquisition* phase is completed. Then the second phase called *plume tracking* starts. During this phase, the robot is supposed to approach the source or obtain information about the source position by sensing the plume. Finally, in the *source identification* phase,

the position of the source should be declared, once the latter is surely localized. This paper, along with the majority of the works in this field, focuses on the second sub-problem.

Despite active studies addressing this problem, odor source localization remains a very challenging task. This high level of complexity comes from the large number of environmental parameters that take part in plume formation, often yielding concentration fields that can hardly be processed analytically. In order to understand the problem, systematic experiments are usually led in controlled environments involving a pseudo-laminar flow with a fixed velocity. Such a configuration can be modeled by the Advective-Diffusive Equations (ADE) [3] used in realistic simulations. In [4], Kowadlo and Russell provide an overview of a large number of robotic algorithms for odor source localization, which reflects very well the variety and richness of this research field. In a wide perspective, one can establish four general, partially overlapping categories, to classify odor plume tracking algorithms [5]: formation-based, probabilistic, map-based and bio-inspired algorithms.

Formation-based algorithms involve multiple robots which maintain a spatial formation. While the formation moves upwind, its scale can adapt to follow concentration's isoclines to approach the source [6], [7]. This category requires at least two robots and a mean of inter-robot relative positioning. *Probabilistic* algorithms model the source location as a probability distribution. Infotaxis [8] is an example of this category, the robot moves towards the position providing the highest gain of information. This requires to keep in memory all data collected along a run. In the case the plume varies strongly over time, further computations and complexity has to be involved in algorithms in order not to be misled using too old samples. *Map-based* algorithms do not necessarily focus on source detection, but usually aim to build a concentration map of the search field. In this category, the robots builds hypothesis based on naive physics airflow mapping [9] or on more complex models like hidden Markov plume model developed in [10] where probabilistic inferences are also performed. Here again, unsteady flows may generate contradictions among odor samples collected that may require further complexity in the algorithms. *Bio-inspired* algorithms take their inspiration from living beings and traditionally combine chemo- and anemo-taxis. A long list of species studied for this purpose is provided in [4]: among others dung beetle, moths and *Escherichia Coli* bacteria take an important place. Various strategies exist in this class: dung beetle- and moth-inspired algorithms rely on an odor concentration threshold to know whether they are in or

The authors are with the Distributed Intelligent Systems and Algorithms Laboratory, School of Architecture, Civil and Environmental Engineering, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland. E-mail: `firstname.lastname@epfl.ch`. This work was partially funded by the Swiss National Science Foundation under grant 200021_153310/1.

out of the plume and then behave accordingly [5], whereas E. Coli algorithm performs a biased random walk [11]. Based on the animal’s search strategy [12], random-walk models have been designed to reproduce similar trajectories, which yielded to Lévy Walk [13] and Lévy Taxis [14] algorithms. The algorithms of this category are typically designed for single robots and therefore not easily adaptable to multi-robot systems. They are mostly *reactive*, which means that the robot only processes the current (or very recent) state of the field. Thus they require only very limited memory and computational resources. The present paper works on algorithms belonging to this category.

Bio-inspired algorithms in odor source localization are among the most studied ones. A significant attention has been given to moth’s behavior [4] [15] [16] [17] where three main actions were identified: (i) surging upwind, (ii) casting and (iii) spiraling. In fact, a moth tracks odor plumes in the following way [18] [19]: when it senses high odor concentrations inside the odor plumes, (i) the moth moves upwind to get closer to the source; if the odor’s intensity decreases, (ii) it amplifies a casting movement yielding a sinusoidal trajectory; if the plume is lost, (iii) it starts spiraling to find it back. Based on this behavior, three algorithms have been proposed in the literature: a first one, we call Casting, but is also known as the dung beetle algorithm, relies only on the second behavior, while the two others, called Surge-Cast and Surge-Spiral, combine (i and ii) and (i and iii), respectively. These algorithms have been deeply studied both in simulations and real experiments in [5].

Unlike moth-inspired algorithms, Lévy Taxis is not directly inspired from one particular species, but tries to reproduce what can be observed at many scales among a large variety of living beings [13]. Originally designed to address the plume finding phase, it is particularly interesting for its relatively low complexity and its potential to dynamically adapt to the environment to face discontinuous and peaky plumes, thanks to its intrinsic stochasticity. We based our work on this algorithm to push it further and extend its environmental adaptability by considering odor concentration gradients. We thus present a new algorithm called Adaptive Lévy Taxis (ALT), which is based on Lévy Taxis (LT) presented in [14]. ALT differs from LT as the movement decision at each step is based on measured gradient of the plume. This allows to continuously adapt the walk to the environment and thus better address the plume tracking phase. The advantages of such an algorithm are: its independence from absolute concentrations (which are usually affected by sensor’s drift in time), its computational simplicity and its suitability to address the plume tracking problem.

In order to evaluate this new algorithm, its performance had to be compared with algorithms of the same class, i.e. bio-inspired ones. As the three moth-inspired algorithms mentioned above have already been subject to deep studies, they have been chosen as reference for performance comparisons. Yet we pushed their development a step ahead and managed to test them under new challenging conditions.

Therefore, the contributions of the present paper are the

following: (i) the development of the new ALT algorithm for odor plume tracking, (2) the improvement of moth-inspired algorithms by rigorously determining the odor concentration threshold at which the robots considers itself inside or outside the plume, and (3) the systematic performance evaluation of ALT and three moth-inspired algorithms in various environmental conditions. Although moth-inspired algorithms are well-known, their performance has not been systematically evaluated in realistic environments under various plume conditions so far. We carried out experiments in a large wind tunnel, studying the impact on the performance of the algorithms of the two most important parameters, i.e. the wind speed and the release rate of the odor source. Particularly, the algorithms have been pushed to their limits and assessed in critical conditions, such as in low wind speed combined with high release rate, a scenario that generates important gradients and strong discontinuities in the plume.

In the present paper, after a description of the algorithms involved in the study, the evaluation setup, the metrics, and the sampling procedures applied to real world experiments are explained in detail. The obtained results are then presented and discussed.

II. ALGORITHMS

A. Lévy Taxis

In 2009, Pasternak et al. presented Lévy Taxis [14] as a novel algorithm for finding odor plume (i.e. *plume acquisition* phase). It is a search strategy of the same category as Correlated Random Walk (CRW) [20] and Lévy Walk (LW) [13]. These are random search algorithms that share the same global process: the agent starts from a random point and has to move in the space, according to a governing law, in order to find the odor plume. At each step, the robot should walk to a point determined by a move length (M_l) (AKA step length) and a turning angle (T_a), which are both calculated based on a fixed probability distribution. The main differences between these search strategies are the choice of the probability distributions determined by the value of their key parameters.

Lévy Taxis combines the M_l probability distribution of Lévy Walk and the T_a probability distribution of Correlated Random Walk. Hence M_l and T_a of each step are calculated using Eq. (1) and Eq. (2),

$$M_l = L_{min} \cdot r^{\frac{1}{1-\mu}} \quad (1)$$

with M_l the move length, r a random variable uniformly distributed $r \in [0, 1]$, L_{min} the minimum move length, and μ (Lévy-index) the move length’s key parameter ($1 < \mu \leq 3$).

$$T_a = \left[2 \cdot \arctan \left(\frac{1-\gamma}{1+\gamma} \cdot \tan(\pi \cdot (r - 0.5)) \right) \right] + bias \quad (2)$$

Where T_a is the turning angle, r a random variable uniformly distributed $r \in [0, 1]$ and γ the turning angle’s key parameter that shapes the distribution ($0 \leq \gamma \leq 1$). The *bias* is the upwind angle: this brings the center of the distribution at the upwind direction, to make the latter the most probably chosen value. Table I presents the governing laws of the

TABLE I
GOVERNING LAWS AND KEY PARAMETERS OF RANDOM WALKS
(ADAPTED FROM [14])

	Move Length (M_l)	μ	Turning Angle (T_a)	γ
Brownian Walk	Asymptotically Gaussian-like	3	Uniform	0
Lévy Walk	Power Law	$1 < \mu \leq 3$	Uniform	0
Correlated Random Walk	Asymptotically Gaussian-like	3	Wrapped Cauchy	$0 \leq \gamma \leq 1$
Lévy Taxis	Power Law	$1 < \mu \leq 3$	Wrapped Cauchy	$0 \leq \gamma \leq 1$

algorithms presented in [14] as well as the values of their key parameters.

The value of the key parameters μ and γ are chosen before the start of the algorithm and they are kept constant during a run. However, at each step, a random value r is picked in a uniform distribution $[0, 1]$, based on which M_l and T_a are calculated. Once a new point in the space has been chosen to be sampled, the agent walks towards it with a constant speed.

All these search algorithms have been evaluated in [14] using a chemical plume data set for different pairs of the key parameters. The performance of each setup has been assessed by measuring the detection success rate (i.e. percentage of successful runs over a total of 50 runs) as well as the path directness (i.e. the percentage ratio of the shortest distance between the start point and the plume detection point over the total traveled distance).

Comparing the performance of all these algorithms together shows that both LW and LT yield the best detection success rate ($\approx 85\%$) with $\mu = 2.8$ and both CRW and LT give the best path directness ($\approx 10\%$) using $\gamma = 0.05$ [14]. This implies that LW have the best success rate but tends to travel more to find the plume, compared to CRW. On the other hand, CRW finds a more direct path to the plume, but has a lower detection success rate. Yet, as LT cleverly combines their governing laws, it brings together the best success rate obtained from LW and the shortest traveled distance from CRW.

In addition to its remarkable performance, LT has the advantage of being environmentally adaptive: since it uses the upwind angle as a bias for the navigation, the strategy is continuously adapted to the environment of the experiment. These two significant assets of the LT algorithm have been the main inspirations for this work: using Lévy Taxis to address the plume tracking phase by introducing chemo-taxis and thus designing the new algorithm we called Adaptive Lévy Taxis (ALT).

B. Adaptive Lévy Taxis

As mentioned earlier, Lévy Taxis has the advantage to adapt the search operation to the environment by orienting the agent to the upwind direction. However, the stochastic nature of this algorithm prevents it from being successful in the plume tracking phase, using only the wind direction as

bias. An agent performing LT in this phase would constantly go out of the plume just after finding it and thus, would have to restart the search for the plume all over again. Indeed, LT does not perform chemo-taxis as its movements are not influenced by odor cues. In order for the agent to stay in the plume once it is found, this type of algorithms needs to adapt the walk to the odor concentration as well, so it can find its path to the source. Thus, we propose to dynamically change the key parameters (i.e. μ and γ) of the ALT algorithm, according to the odor gradients measured. This modification enables ALT to address the odor plume tracking problem.

At each step, the odor concentration is measured and compared to the one of the previous step to calculate the local gradient ∇C between the current sampling position and the previous one. We then use the value of the local gradient to tune the key parameters of the governing laws before taking a new step – instead of keeping their value constant over the run which is the case in the original Lévy Taxis algorithm developed in [14].

In long-term exposures (i.e. in the case where the robot moves slowly and each sample is the average of many concentration measurements), when the average local gradient is strongly positive, it theoretically means that the robot is closer to the source compared to the previous step, so the next step can be a long move straight towards the upwind direction, i.e. exploitation. On the other hand, in the case where the gradient is around zero, as the agent does not get any strong clue about the variation of the plume, it takes a short move in a random direction in order to promote exploration. The former case corresponds to key parameters μ and γ tending to 1, while in the latter μ tends to its maximal value 3 and γ to its minimal value 0.

Eq. (3a) and Eq. (3b) show the relationship between the gradient of odor concentration ∇C and the key parameters μ and γ , while Fig. 1 and Fig. 2 present how the latter influences the probability distribution of M_l and T_a .

$$\mu = \begin{cases} \mu_{min}, & \text{if } |\nabla C| > \theta \\ \mu_{max} - \frac{|\nabla C|}{\theta}(\mu_{max} - \mu_{min}), & \text{otherwise} \end{cases} \quad (3a)$$

$$\gamma = \begin{cases} \gamma_{max}, & \text{if } |\nabla C| > \theta \\ \gamma_{min} + \frac{|\nabla C|}{\theta}(\gamma_{max} - \gamma_{min}), & \text{otherwise} \end{cases} \quad (3b)$$

Where ∇C is the gradient of odor concentration and θ the threshold beyond which the values of μ and γ remain at μ_{min} and γ_{max} respectively. These boundaries were set based on Pasternak's work [14].

Besides adapting key parameters during the run, the gradient serves another purpose: as it is calculated after taking one step, it brings information about the rightness of the last decision on Turning Angle (T_a). Therefore, the heading of the robot can be used as a complementary information to the wind direction for the bias used in the calculation of the T_a . For instance, if after taking one step the gradient is positive, which means that the last direction was the right decision, the angle for the next step should be biased not only by the local wind direction but also by the forward direction. Thus, in ALT, the bias for T_a is taken as an average of the heading

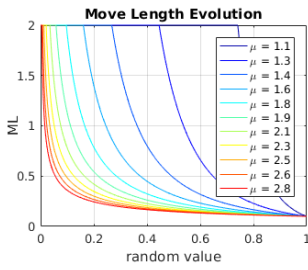


Fig. 1. M_l distribution according to its governing parameter $\mu \in [1.1; 2.9]$. M_l itself varies between 0.1 and 2.0.

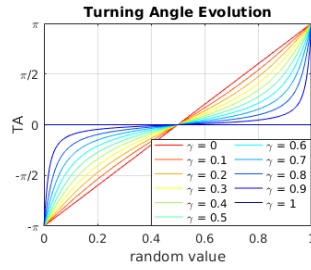


Fig. 2. T_a distribution in function of $\gamma \in [0; 1]$. $\gamma = 0$ yields a uniform distribution while $\gamma = 1$ yields $T_a = 0$ anyway.

of the robot and the upwind angle, unlike LT which only takes into account the local wind direction. This acts as a low-pass filter that prevents the heading from varying too much in one step.

However, in case the gradient is negative, which means the last chosen direction was not a right one (the robot is probably not heading towards the source), only the upwind angle is used as bias. Therefore the robot tends to move upwind to encounter a higher odor concentration.

Pseudo code of the Adaptive Lévy Taxis algorithm

```

while (step < stepLimit) do
  Sample odor ( $c_c$ ) and read current position ( $p_c$ )
  Calculate odor gradient  $\nabla C = c_c - c_p$ 
  Adapt parameters  $\mu$  and  $\gamma$  using Eq. (3a) and Eq. (3b)
  Calculate  $M_l$  and  $T_a$  using Eq. (1) and Eq. (2)
  Apply  $M_l$  and  $T_a$  to  $p_c$  to get next goal ( $G_x, G_y$ )
  previous odor ( $c_p$ )  $\leftarrow$  current odor ( $c_c$ )
  previous position ( $p_p$ )  $\leftarrow$  current position ( $p_c$ )
  while (( $G_x, G_y$ ) is not reached) do
    Move towards ( $G_x, G_y$ )
    if (Source is found) then
      Declare success and Terminate algorithm
    else if (Wall is encountered) then
      Declare failure and Terminate algorithm
    end if
  end while
  end while
  step  $\leftarrow$  step + 1
end while
Declare failure

```

The pseudo-code above provides an overview of the proposed ALT. In this algorithm, at the beginning of each step, the position of the agent as well as the local wind direction and the odor concentration are logged. Then the odor gradient is calculated using the last step's log. Using this information, M_l and T_a are calculated to determine the goal towards which the agent has to walk. Once the source is found, which is determined thanks to an external input, the agent declares success and terminates the algorithm. In case the robot reaches the limits of the experimentation area, it declares failure and terminates. On top of that, we used a limit for the number of steps, for the agent to avoid

spending too much experimentation time wandering far from the source. If this steps limit is reached, the algorithm also declares failure.

C. Moth-inspired Algorithms

Many studies analyzed how moths use odor source localization to find food. Three main algorithms have already been implemented and tested by Thomas Lochmatter in [5].

The algorithms, along with different parameters and expected trajectories, are presented in detail in this section. All of them are dependent at least on two parameters: the threshold O_{th} determining whether the robot is in or out of the plume and the distance D_{lost} that a robot travels sampling the field below the threshold O_{th} before considering being out of the plume. Additionally, each algorithm has its own set of parameters taken as the ones that came out as being the most promising in Lochmatter's work [5].

1) *Casting Algorithm*: The Casting algorithm performs casting only. It relies on the wind direction as it zigzags with a fixed angle $\pm\beta$ relative to the upwind direction. The agent first moves upwind with an angle $+\beta$. If the plume is lost for a distance greater than D_{lost} , the robot turns crosswind until it finds back the plume. Then, it continues upwind but this time heading $-\beta$. The same process continues until the source is found. The value used for β angle is 20° which proved to yield a good performance in [5].

2) *Surge-Cast Algorithm*: The Surge-Cast algorithm combines two behaviors identified in moths: casting and upwind surge. It slightly differs from the previous algorithm in the sense that the robot moves upwind with an angle β always equal to 0° . When the plume is lost, i.e. the robot has traveled more than D_{lost} out of the plume, it turns $+90^\circ$ and heads crosswind to find back the plume. If after traveling a distance greater than D_{cast} (43 cm based on [5]) the plume is still not found, the agent makes a $+180^\circ$ turn and heads crosswind in the other direction to find the plume. In case the plume is found while moving crosswind, the robot goes on a little bit deeper into the plume before heading upwind again.

3) *Surge-Spiral Algorithm*: The Surge-Spiral algorithm is based on spiraling and upwind surge behaviors. As the previous algorithm, the agent starts moving upwind but, if the plume is lost, it performs an Archimedean spiral to find the plume back. The spiral is made by adding a quantity D_{Speed} to the speed of one wheel and subtracting it from the other one. This quantity D_{Speed} is called *Differential Speed* and is calculated by Eq. (4),

$$D_{Speed} = \frac{d_{gap}}{\sqrt{step}} \quad (4)$$

with d_{gap} the coefficient that defines how tight the spiral is, and $step$ the current discretization step. At each step, the differential speed is thus calculated and summed to or subtracted from the constant forward speed. The d_{gap} parameter has been set to 58 cm as suggested in [5].

Defining the Odor Threshold

In moth-inspired algorithms, one of the most important parameter is the odor threshold O_{th} , on which they rely to

determine whether the agent is in or out of the plume. In previous studies [5], this threshold was determined intuitively by visualizing the odor distribution along the cross section of the wind tunnel where the operation starts. In our case, as we worked with various environmental conditions and needed to compare results of different setups, we derived a mathematical approach based on the Advective-Diffusive Equations (ADE) to determine this threshold in a rigorous way. ADE are expressed in (5) with C the odor concentration at a given location, t the time, D the diffusion coefficient, \vec{v} the wind speed and R the source or sink's rate.

$$\frac{\delta C}{\delta t} = \nabla \cdot (D \nabla C) - \nabla \cdot (\vec{v} C) + R \quad (5)$$

As demonstrated in [21], the profile at a downstream distance x of the source in an open space is a Gaussian function given by Eq. (6), where M is the mass input and D_t the turbulent coefficient of diffusivity.

$$C(x, t) = \frac{M}{\sqrt{4\pi D_t t}} \cdot \exp\left(\frac{-x^2}{4D_t t}\right) \quad (6)$$

The standard deviation σ of such a Gaussian is given in Eq. (7).

$$\sigma = \sqrt{2D_t t} \quad (7)$$

On this basis, we consider being out of the plume if the concentration measured is below concentrations theoretically present in the interval $[\bar{\mu} - \sigma, \bar{\mu} + \sigma]$, with $\bar{\mu}$ a position on the axis aligned with the mean direction of the wind passing by the source. Considering a statistically representative data set of samples taken along a long enough crosswind section of length L , the probability of occurrence of a sample marked as out is given by Eq. (8).

$$P_{th} = 1 - \frac{2\sigma}{L} \quad (8)$$

P_{th} is therefore considered as the probability threshold to which a concentration O_{th} can be matched based on the aforementioned data set collected.

A special caution has to be exercised as experiments are led in a bounded environment. In order for the plume shape not to be significantly affected by the walls separated by a distance d_W , Eq. (9) must be respected.

$$d_W > 6\sigma \quad (9)$$

As 99.73% of the plume is contained within the interval $[\bar{\mu} - 3\sigma, \bar{\mu} + 3\sigma]$, this condition implies that no more than 0.27% of the odor plume is stacked somewhere and neglected. This condition was respected in all our experiments.

main drawbacks of this approach is that D_t is dependent on many parameters and thus requires a special attention to be determined trustfully. That is why, for each setup, a constant threshold P_{th} was determined manually based on a visualization of the cumulative distribution function based on the samples taken across wind along the starting point of the robot in the arena. Another one is that this relies on the validity of the ADE, which is an approximation of the microscopic stochastic reality. Yet, this allowed similarity among all runs of a particular setup.

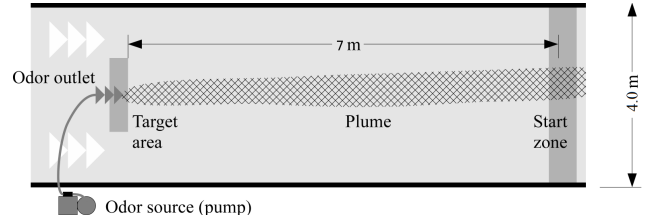


Fig. 3. Schema of the wind tunnel setup taken from [5].

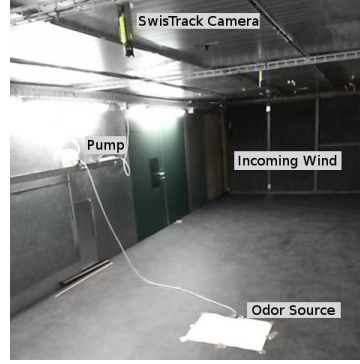


Fig. 4. A picture of the wind tunnel showing the pump, a camera and the odor source. The white sheet of paper surrounding the source was used to allow the robot know the source is reached when it rolls over it.



Fig. 5. A Khepera IV robot equipped with an odor sensing board and an anemometer board stacked on top.

III. EXPERIMENTAL EVALUATIONS

A. Environment

As the evaluation of ALT was led in various environmental conditions involving setups that were, as far as we know, not tested in other studies, it was necessary to repeat the same experiments with reference algorithms (i.e. the three moth-inspired algorithms) to assess the quality of performances obtained. No comparisons were made with LT, as the two algorithms do not address the same phase of the problem.

All algorithms were tested on Khepera IV robots [22] [23] in a wind tunnel of $20 \times 4 \times 2 \text{ m}^3$, setting the maximum speed to 1 m/s and focusing on the upper part of the channel, see Fig. 3-5. To enable Khepera IV robots to sense odor concentrations, we used the chemical sensing and wind sensor boards developed by Lochmatter [5].

ALT was evaluated in wind speeds ranging from 0.1 m/s to 1.0 m/s . As odor source, we used an electric pumping device vaporizing ethanol in the air through a thin pipe, which allowed us to control the release rate in percentage. The robot and the source were placed at the center of the wind tunnel, initially 7 m far from each other, see Fig. 3. Moreover, during the experiments, the position of the robot has been tracked and logged using the positioning system SwisTrack [24] which relies on 3 synchronized cameras covering an 8 m long section of the wind tunnel. This data has then been used to calculate performances. The latter is the result of 10 runs for every configuration and metrics used are presented below.

B. Metrics

Distance overhead and *success rate* are the two metrics used to estimate algorithmic performances. Distance overhead is the ratio of the distance traveled by the robot (d_t) over the length of the shortest path linking the start point and the point where source is declared (d_u). This is easily expressed mathematically in Eq. (10).

$$d_0 = \frac{d_t}{d_u} \quad (10)$$

A lower d_0 means a better performance and the best achievable result is $d_0 = 1$. One of the advantages of considering the distance overhead is that it does only indirectly reflect the time spent in the experiment, but it does not penalize an algorithm requiring lots of turns - which is time consuming for a non-holonomic vehicle. Furthermore, this takes out the actual robot's speed profile over the path from the metric.

The second metric, the success rate s_r , is the ratio of the number of successes (i.e. finding the source within a specific time window without hurting a wall) over the total number of experiments, as expressed in Eq. (11).

$$s_r = \frac{\#success}{\#runs} \quad (11)$$

C. Odor Sampling

Each time the robot needs to sample the odor concentration, it takes 5 samples and then retains the highest value in order to filter noise. We chose a sampling rate of 10 Hz in order to respect the response time of MiCS-5521 CO/VOC sensors [25] in our active sniffing setup. Additionally, for the results to be comparable in terms of measured absolute odor concentration in all the configurations, a baseline (B_0) was sampled out of the plume before each run. This cancels out the influence of sensor drift as well as temperature and humidity variations on the data. The procedure is shown by Eq. (12) which implies discarding the first 200 samples and then subtracting 500 (arbitrary value) from the average of 800 other samples to work with lower yet positive values.

$$B_0 = \sum_{n=201}^{1000} \frac{S_n}{800} - 500 \quad (12)$$

with S_n the n^{th} raw sample. The obtained odor baseline B_0 is then subtracted from all further samples. The relationship between measurements and ppm units was not established.

D. Results

Fig. 6 provides typical trajectories for all four algorithms. Figures 7, 8 and 9 show the performance of moth-inspired algorithms in different setups. They were used to explore environmental parameters ranges yielding to challenging operational conditions for the algorithms. Box-plots represent the distance overhead which is, in most cases, below a factor 2. The success rate for each setup is presented using bars at the top.

All three moth-inspired algorithms have remarkable performances in setups with high release rate (50%) and high

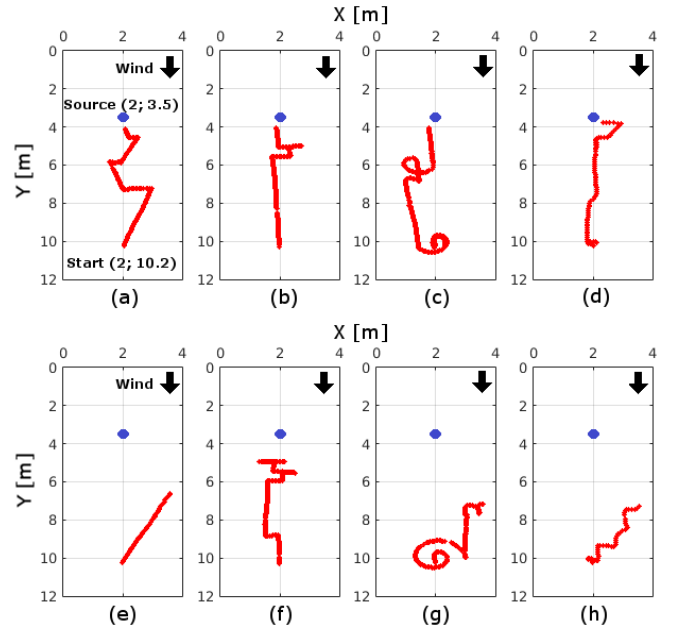


Fig. 6. Trajectories obtained in the wind tunnel with a wind speed of 1.0 m/s and a release rate of 10%: (a, e) Casting, (b, f) Surge-Cast, (c, g) Surge-Spiral and (d, h) Adaptive Lévy Taxis algorithm. (a-d) show successful runs (not necessarily the best one in terms of performance) while (e-h) show failed runs.

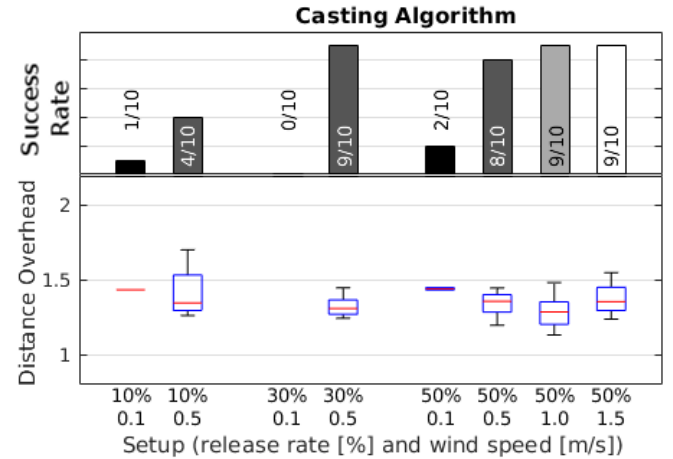


Fig. 7. Performances of the Casting algorithm in the wind tunnel.

wind speed ($\geq 0.5 m/s$). This is due to the shape of the plume in such conditions where the source emits a fair amount of odor patches and the wind folds the plume in the center. Thus, it is easily tracked in these conditions. This is the common point of all the three algorithms; however, each of them reacts differently to more critical conditions.

Fig. 7 shows that Casting is very sensitive to extreme environmental conditions. In our experiments, it has been unsuccessful when the wind speed was low (0.1 m/s) or when the release rate of the source was set as low as 10%. On the other hand, in case the conditions are suitable for this algorithm to work, it yields remarkable results, both in terms of success rate and distance overhead, as we can see for release rate $\geq 30\%$ and wind speed $\geq 0.5 m/s$. Fig. 6 (a)

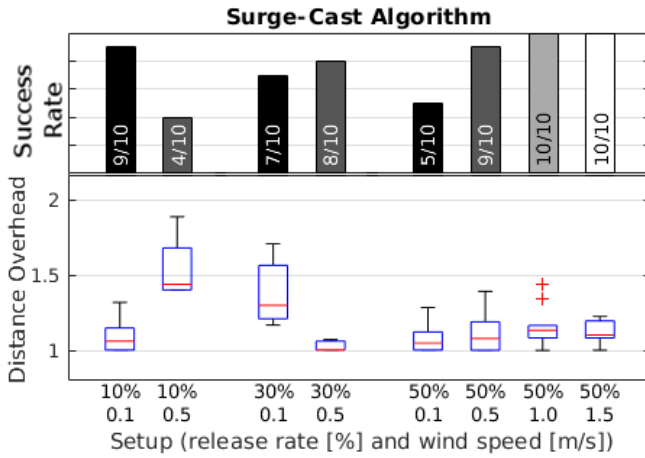


Fig. 8. Performances of Surge-Cast algorithm in the wind tunnel.

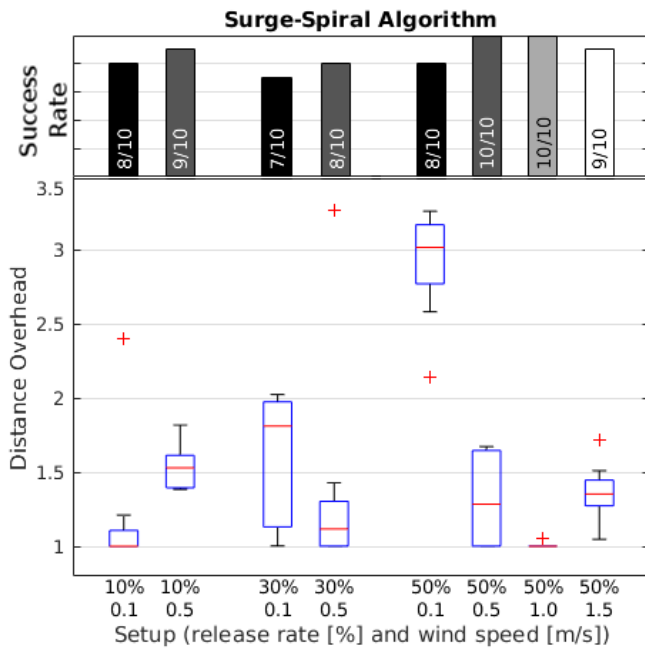


Fig. 9. Performances of Surge-Spiral algorithm in the wind tunnel.

and (e) refer to trajectories obtained with this algorithm. In case of a rich odor plume, the agent needs to make very little maneuvering to reach the source. But in case of a sparse plume, it might lose the plume very easily and do not find it again, which leads to failure (the case in Fig. 6 (e)).

The results of Surge-Cast algorithm are also affected by low release rate and low wind speed, according to Fig. 8. However, as this algorithm makes the agent follow the wind speed without any shift, it has more chances in extreme setups, compared to Casting, as it starts downstream of the source. Surge-Cast has the best distance overhead of the three moth-inspired algorithms and performed particularly well in lowest release rate (10%) and lowest wind speed (0.1 m/s). Fig. 6 (b) and (f) show two exemplar trajectories of this algorithm.

The Surge-Spiral algorithm, unlike the two others,

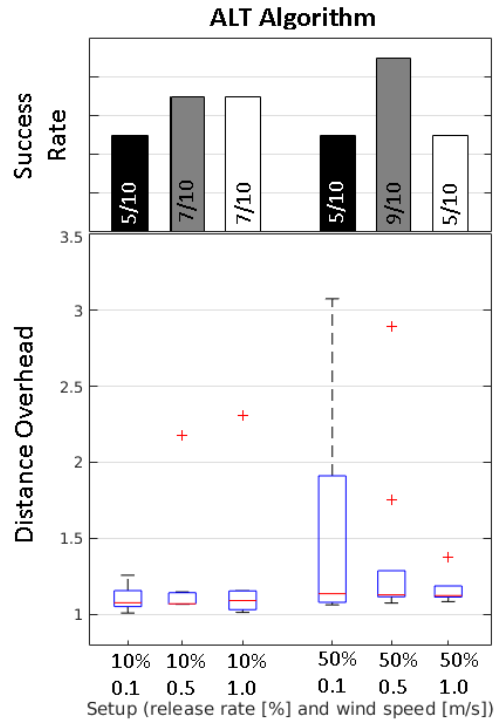


Fig. 10. Performances of ALT algorithm in the wind tunnel.

yields outstanding success rates in all the setups, according to Fig. 9. In case of difficult setups (e.g., wind speed = 0.1 m/s and release rate = 50%), in which Casting and Surge-Cast fail in the majority of the experiments, spiraling has a satisfactory success rate ($\geq 7/10$), but a relatively high distance overhead. In fact, this spiraling strategy helps the robot to eventually re-acquire the plume after losing it. However, it also implies taking more steps, compared to the two previous algorithms, which lead to a higher distance overhead. Fig. 6 (c) shows how Surge-Spiral manages to find the plume back using a simple spiral trajectory, while Fig. 6 (g) shows how it sometimes fails to do so.

Fig. 6 (d) and (h) show two trajectories obtained in two experiments (a success and a failure) by our newly developed ALT algorithm. The numerical results are presented in Fig. 10. Based on the information obtained from the reference methods, different parameters combinations were tested, giving priority to extreme release rates. As it is shown, the average distance overhead in all setups is lower than 1.2 (i.e. close to 1.0 that is the best possible value). In addition, ALT shows satisfactory results ($\geq 5/10$) in terms of success rate in all configurations. The consistency of these good results in all the setups is a remarkable performance, not achieved by moth-inspired algorithms.

IV. DISCUSSION

As mentioned while reporting the results of experiments, ALT yielded remarkable performance in all setups, which is an advantage over moth-inspired algorithms that are sensitive to environmental conditions. Moreover, this new algorithm presents another asset: if encountering walls was not con-

sidered as a failure, then the only limit of this algorithm would be the one applied to the number of steps. In case of a wall free arena, a robot performing ALT, could eventually be able to reacquire the plume after losing it (yielding to 100% success rate), thanks to the stochastic search properties of this algorithm. Indeed, convergence in 3D was demonstrated in [26] for a Lévy Flight. In contrast, moth-inspired algorithms, such as Casting or Surge-Cast, once they lose the plume, have severe difficulties to re-acquire it. The only moth-inspired competitive algorithm from this perspective is the Surge-Spiral, an algorithm which would be also 100% successful in case of a wall-free arena at the price of a very high distance overhead. Probably, only a smart combination of casting, surging, and spiraling, as it actually happens in real moths, would be competitive with our ALT algorithm in all different environmental conditions.

V. CONCLUSION

A new algorithm called Adaptive Lévy Taxis (ALT) has been developed to address the plume tracking in odor source localization. It has the novelty to yield consistent performance in different environmental conditions, unlike previous algorithms of the bio-inspired category, such as moth-inspired algorithms. The latter have also been reviewed in this work by improving their odor threshold determination process. The new ALT and the well-known moth-inspired algorithms have all been examined in extreme environmental conditions and their results have been thoroughly reported and discussed.

Thanks to its consistency, this new ALT algorithm can find its application in environments where atmospheric conditions are either unknown or too complex for other algorithms.

Finally, to go further with this new approach of ALT, one can try to calculate the odor gradient using more than two samples. This may increase the performance of the algorithm. Furthermore, a collaborative strategy based on ALT could be developed, involving multiple agents sharing odor samples. This would allow to calculate the odor gradient every single step, replacing the need of memory by communication. Finally, the same approach can be implemented for a drone in order to perform odor source localization in 3D.

ACKNOWLEDGMENT

We sincerely thank Dr. Thomas Lochmatter for constructive discussions and his work on moth-inspired algorithms and Khepera robots.

REFERENCES

- [1] P. J. Roberts and D. R. Webster, *Turbulent diffusion*. ASCE Press, Reston, Virginia, 2002.
- [2] R. A. Russell, D. Thiel, R. Deveza, and A. Mackay-Sim, "A robotic system to locate hazardous chemical leaks," in *IEEE Int. Conf. on Robotics and Automation*, pp. 556–561, 1995.
- [3] H. Hattori, S. Yamada, M. Tanaka, T. Houra, and Y. Nagano, "DNS, LES and RANS of turbulent heat transfer in boundary layer with suddenly changing wall thermal conditions," *Int. Journal of Heat and Fluid Flow*, vol. 41, pp. 34–44, 2013.
- [4] G. Kowadlo and R. A. Russell, "Robot odor localization: A taxonomy and survey," *The Int. Journal of Robotics Research*, vol. 27, no. 8, pp. 869–894, 2008.

- [5] T. Lochmatter, "Bio-inspired and probabilistic algorithms for distributed odor source localization using mobile robots," *PhD Thesis No. 4628*. EPFL, 2010.
- [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 Int. Conf. on Robotics and Automation*, pp. 1830–1836, 2015.
- [7] J. M. Soares, A. P. Aguiar, A. M. Pascoal, and A. Martinoli, "A graph-based formation algorithm for odor plume tracing," in *Proc. Int. Symp. on Distributed Autonomous Robotic Systems, 2014; Springer Tracts in Advanced Robotics*, vol. 112, pp. 255–269, 2016.
- [8] M. Vergassola, E. Villermaux, and B. I. Shraiman, "infotaxis as a strategy for searching without gradients," *Nature*, vol. 445, no. 7126, pp. 406–409, 2007.
- [9] G. Kowadlo and R. A. Russell, "Improving the robustness of naive physics airflow mapping, using Bayesian reasoning on a multiple hypothesis tree," *Robotics and Autonomous Systems*, pp. 672–677, 2006.
- [10] 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, pp. 143–169, 2002.
- [11] L. Marques, U. Nunes, and A. T. de Almeida, "Olfaction-based mobile robot navigation," *Thin Solid Films*, vol. 418, no. 1, pp. 51–58, 2002.
- [12] F. Bartumeus, M. G. E. da Luz, G. M. Viswanathan, and J. Catalan, "Animal search strategies: A quantitative random-walk analysis," *Ecology*, vol. 86, no. 11, pp. 3078–3087, 2005.
- [13] V. Zaburdaev, S. Denisov, and J. Klafter, "Lévy walks," *Reviews of Modern Physics*, vol. 87, no. 2, pp. 483–530, 2015.
- [14] Z. Pasternak, F. Bartumeus, and F. W. Grasso, "Lévy-taxis: a novel search strategy for finding odor plumes in turbulent flow-dominated environments," *Journal of Physics A: Mathematical and Theoretical*, vol. 42, no. 43, p. 434010, 2009.
- [15] N. J. Vickers, "Mechanisms of animal navigation in odor plumes," *The Biological Bulletin*, vol. 198, no. 2, pp. 203–212, 2000.
- [16] J. H. Belanger and M. A. Willis, "Biologically-inspired search algorithms for locating unseen odor sources," in *Intelligent Control (ISIC)*, pp. 265–270, 1998.
- [17] H. Ishida, K. Hayashi, M. Takakusaki, T. Nakamoto, T. Moriizumi, and R. Kanzaki, "Odour-source localization system mimicking behaviour of silkworm moth," *Sensors and Actuators A: Physical*, vol. 51, no. 2, pp. 225–230, 1995.
- [18] L. P. S. Kuenen and H. C. Rowe, "Cowpea weevil flights to a point source of female sex pheromone: analyses of flight tracks at three wind speeds," *Physiological Entomology*, vol. 31, no. 2, pp. 103–109, 2006.
- [19] K. A. Justus, S. W. Schofield, J. Murlis, and R. T. Cardé, "Flight behaviour of *cadra cautella* males in rapidly pulsed pheromone plumes," *Physiological Entomology*, vol. 27, no. 1, pp. 58–66, 2002.
- [20] M.-L. Cao, Q.-H. Meng, B. Luo, and M. Zeng, "Experimental comparison of random search strategies for multi-robot based odour finding without wind information," *Austrian Contributions to Veterinary Epidemiology*, vol. 8, pp. 43–50, 2015.
- [21] S. A. Socolofsky and G. H. Jirka, *Special Topics in Mixing and Transport Processes in the Environment*. Coastal and Ocean Engineering Division, 5th edition ed., 2005.
- [22] J. M. Soares, I. Navarro, and A. Martinoli, "The Khepera IV mobile robot: performance evaluation, sensory data and software toolbox," in *Robot 2015: Second Iberian Robotics Conference*, pp. 767–781, 2015.
- [23] J. M. Soares, A. Marjovi, J. Giezendanner, A. A. Kodyan, A. P. Aguiar, A. Pascoal, and A. Martinoli, "Towards 3-D distributed odor source localization: an extended graph-based formation control algorithm for plume tracking," *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 1729–1736, 2016.
- [24] T. Lochmatter, P. Roduit, C. Cianci, N. Correll, J. Jacot, and A. Martinoli, "SwisTrack - a flexible open source tracking software for multi-agent systems," *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 4004–4010, 2008.
- [25] SGX Sensortech technologies, "MiCS-5521 CO/VOC sensor." url: <http://www.sgxsensortech.com>.
- [26] V. Nazarzehi and A. Baranzadeh, "A distributed bio-inspired algorithm for search of moving targets in three dimensional spaces," in *IEEE Int. Conf. on Robotics and Biomimetics*, pp. 2507–2512, 2015.