

# Simulation Experiments with Bio-Inspired Algorithms for Odor Source Localization in Laminar Wind Flow

Thomas Lochmatter and Alcherio Martinoli  
École Polytechnique Fédérale de Lausanne (EPFL)  
Distributed Intelligent Systems and Algorithms Laboratory (DISAL)  
Building GR, Station 2, 1015 Lausanne, Switzerland  
thomas.lochmatter@epfl.ch, alcherio.martinoli@epfl.ch

## Abstract

We compare three bio-inspired odor source localization algorithms (*casting*, *surge-spiral* and *surge-cast*) for environments with a main wind flow in simulation. The wind flow is laminar and the simulation setup similar to the setup in the wind tunnel in which we have carried out similar experiments with real robots [13]. The algorithms are compared in terms of success rate and distance overhead when tracking the plume up to the source. We conclude that the algorithms based on upwind surge yield significantly better performance than pure casting.

## 1 Introduction

With the advances in robotics and chemicals sensor research in the last decade, odor sniffing robots have become an active research area. Notably the localization of odor sources would allow for very interesting robotic applications, such as search and rescue operations, safety and control operations in airports or industrial plants, and humanitarian demining [18] [5] [15] [8]. Many of these applications are time-critical, i. e. odor sources should be found as fast as possible. Moreover, as the structure of plumes in the air is intermittent in both time and space [19], tracking plumes is a challenging problem.

In recent work [14] [13], we have shown through experiments with real robots that the *surge-spiral* [6] [7] [2] [4] and the *surge-cast* [13] algorithms are faster and more reliable than pure *casting* [10] [9] [20] [12] [11] [1] in laminar wind flow. This result was insofar surprising, as the casting algorithm got much more attention by the research community up to date.

In this paper, we present simulation results for the same algorithms to support our findings with the real robots. Running the algorithms in simulation allowed us to carry out

many more experiments of a wider range of algorithmic parameter configurations. In particular, we carried out experiments with varying upwind angle (*casting*), spiral gap (*surge-spiral*), cast distance (*surge-cast*), wind sensor accuracy and plume lost distance.

The experiments were run in Webots [16], a realistic robotic simulator, enhanced with a plume simulation based on the filament propagation model proposed by Farrell et al. [3]. The setup and conditions are kept close to those in the wind tunnel used for the real-robot experiments [13], which allows us to compare the real-robot and simulation experiments.

Note that we only consider plume traversal (i. e. following the plume towards the source) and intentionally omit plume finding (i. e. randomized or systematic search until the plume is found) and source declaration (i. e. declaring that the source is in close vicinity), to prevent those two phases from interfering in the results.

The remainder of this paper is structured as follows. In Section 2 we formally present the three algorithms used in this paper. The simulator and the odor propagation model are introduced in Section 3. Finally, we discuss the results in Section 4 and conclude in Section 5.

## 2 Algorithms

All three algorithms used in this paper are bio-inspired and a combination of upwind surge, casting, and spiraling [17]. The algorithms use only binary odor information, that is, they either *perceive the odor* or *do not perceive any odor*, but ignore different concentrations levels. Commonly, the measured concentration is thresholded to obtain this binary value, but more elaborate processing could be used as well.

Finally, all three algorithms need a wind sensor to measure the wind direction. As molecules are mainly transported by advection, this piece of information is very valuable. The wind speed is ignored.

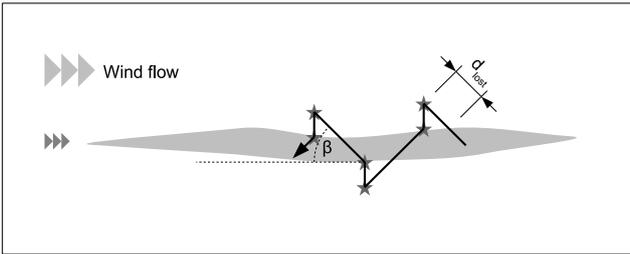
Since we are only interested in the plume traversal behavior, the robot starts in the plume, and declares failure if it gets too far away from it. This allows us to rule out arena geometry effects, which could greatly influence the results (e. g., high variance introduced by randomized search techniques).

Similarly, source declaration is done by a supervisor (ideal source declaration) and therefore does not affect the results. Experiments are considered successful if the robot has come in physical vicinity of the source.

## 2.1 The Casting Algorithm

The casting algorithm is very similar the one described by Li et al. [10]. As shown in Figure 1, a robot in the plume moves upwind with an angle  $\beta$  (relatively to the currently measured wind direction) until it is out of the plume for a certain distance, denoted  $d_{lost}$ . Once the plume is lost, the robot turns and moves cross-wind until it hits an odor packet, and then moves upwind with angle  $\beta$  again.

The wind direction is measured each time the robot switches to plume reacquisition, and when it encounters the plume again.



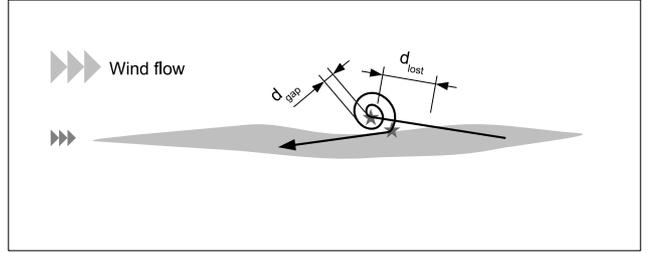
**Figure 1. Sketch of the casting algorithm. The stars indicate where the wind direction is measured.**

## 2.2 The Surge-Spiral Algorithm

The surge-spiral algorithm is similar to Hayes' algorithm presented in [6], except that here we focus exclusively on its use for plume tracking. Hence, we have a single spiral gap parameter.

A robot in the plume moves straight upwind until it loses the plume for a distance  $d_{lost}$ . It then tries to reacquire the plume by moving along an Archimedes spiral with gap size  $d_{gap}$ . Unlike [6], we start our spiral in upwind direction, as drawn in Figure 2.

The wind direction is measured when the robot switches from upwind surge to spiraling, and when it switches back to upwind surge.



**Figure 2. Sketch of the surge-spiral algorithm. The star indicates where the wind direction is measured.**

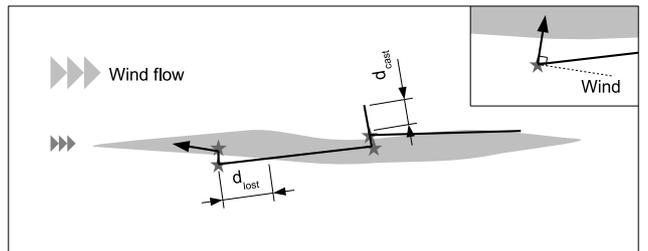
## 2.3 The Surge-Cast Algorithm

The *surge-cast* algorithm [13] is a combination of upwind surge and cross-wind casting. It is similar to the surge-spiral algorithm, with the spiral being replaced by cross-wind movement.

A robot in the plume moves straight upwind until it loses the plume for a distance  $d_{lost}$ . It then tries to reacquire the plume by moving cross-wind for a set distance ( $d_{cast}$ ), first on one side and then on the other. To maximize the chances of hitting the plume in the first cross-wind movement, the robot measures the wind direction to estimate from which side it left the plume.

If the robot did not reacquire the plume by casting, the run is considered unsuccessful. In a real application, the robot would probably switch back to plume finding behavior, or try to reacquire the plume with a larger cast distance or with spiraling.

The wind direction is measured when the robot switches from upwind surge to casting and when it switches back to upwind surge, as indicated in Figure 3.



**Figure 3. Sketch of the surge-cast algorithm. The stars indicate where the wind direction is measured.**

### 3 Simulation Experiments

We are using Webots [16] for the experiments. Webots is a commercial realistic robotic simulator, which ships with a calibrated model of the Khepera III robot that we used for the real-robot experiments [13]. The simulation environment (Figure 5) was augmented with a wind and odor propagation model, and the robot model was extended with the corresponding sensors to measure the odor concentration and a wind direction (Figure 4). The simulation time step,  $\Delta t$ , was set to 32 ms.

#### 3.1 Experimental Arena

The experimental arena is a rectangular area of 16 m length and 4 m width, which corresponds roughly to the dimensions of the wind tunnel. At 1 m from one end of the arena, a circular odor source of radius 12 cm is placed. The robot is placed at roughly 14.5 m downwind from that spot.

#### 3.2 Advection Model

A constant wind field of 1 m/s was used. This corresponds to a constant laminar flow which is comparable to the one used for the real-robot experiments in the wind tunnel [13]. In the coordinate system indicated in Figure 5, the wind vector at position  $\mathbf{u}$ ,  $\mathbf{a}(\mathbf{u})$ , can be written as

$$\mathbf{a}(\mathbf{u}) = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad (1)$$

#### 3.3 Odor Propagation Model

The odor propagation model closely resembles the filament-based model proposed by Farrell et al. [3]. This model is easy to implement and requires only a very limited amount of CPU power. Yet, it generates an intermittent plume which is similar to the real plume in the wind tunnel [13].

Odor is thereby simulated as a set of filaments ( $i = 0, \dots, N$ ), each containing a constant amount  $s = 8.3 \cdot 10^9$  of molecules or particles. Each filament is defined by its position,  $\mathbf{p}_{i,t}$ , and its width,  $w_{i,t}$ .

In each time step, the position of a filament is updated according to the wind flow and a stochastic process:

$$\mathbf{p}_{i,t+\Delta t} = \mathbf{p}_{i,t} + \mathbf{a}(\mathbf{p}_{i,t})\Delta t + \mathbf{v}_p \quad (2)$$

The stochastic component  $\mathbf{v}_p$  is a vector of three independent Gaussian random variables,  $N(0, \sigma_p^2)$ , with standard deviation  $\sigma_p = 0.1$  m.

To model molecular dispersion, filaments become wider with time while their peak concentration decreases. The width of a filament evolves as

$$w_{i,t+\Delta t} = w_{i,t} + \frac{\gamma}{2w_{i,t}} \quad \text{with } \gamma = 4 \cdot 10^{-7} \quad (3)$$

The odor source releases 100 such filaments per second with an initial width of  $w_{i,0} = 10$  cm and an initial position which is uniformly distributed over circular area of the source. This yields a plume which is comparable to the real plume in the wind tunnel.

#### 3.4 Odor Sensor Model

The odor concentration at time  $t$  and position  $\mathbf{u}$  is calculated as the sum over the concentration contribution of all filaments,

$$C_t(\mathbf{u}) = \sum_{i=0}^N c_{i,t}(\mathbf{u}) \quad (4)$$

and each filament  $i$  contributes

$$c_{i,t}(\mathbf{u}) = \frac{s}{w_{i,t}^3} \exp\left(-\frac{|\mathbf{u} - \mathbf{p}_{i,t}|}{w_{i,t}^2}\right) \quad (5)$$

to the concentration. Hence, the concentration decays exponentially with increasing distance from the center of a filament.

The virtual odor sensor reports this concentration  $C_t(\mathbf{u})$  without adding any additional noise, as the perceptual noise related to the chemical-to-electrical transduction is negligible even on the real platform. Furthermore, since the concentration is anyway thresholded and filtered through  $d_{\text{lost}}$  by the algorithms in use here, a precise calibration of the odor propagation and odor sensor model is not required.

#### 3.5 Wind Direction Sensor Model

The wind sensor reports a noisy wind measurement,

$$\mathbf{a}_s(\mathbf{u}) = \mathbf{a}(\mathbf{u}) + \mathbf{v}_a \quad (6)$$

where  $\mathbf{v}_a$  is a vector with samples of a zero-mean normal distribution ( $N(0, \sigma_a^2)$ ). Since the wind field is constant in all our simulations, the reported value in world coordinates is simply

$$\mathbf{a}_s(\mathbf{u}) = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} N(0, \sigma_a^2) \\ N(0, \sigma_a^2) \\ N(0, \sigma_a^2) \end{pmatrix} \quad (7)$$

This vector is rotated into the local reference system of the robot to account for the robot's pose.

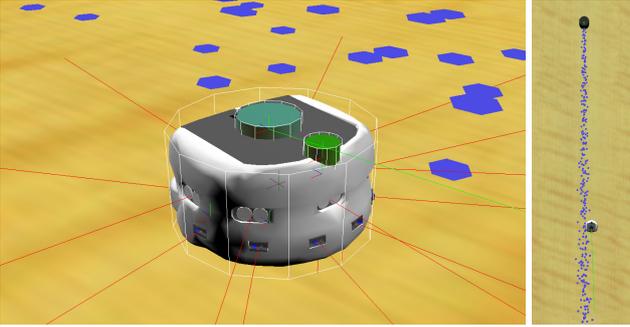


Figure 4. Simulated Khepera III robot equipped with an odor sensor (small cylinder on top of the robot) and a wind sensor (big cylinder). The hexagons in the air represent odor filaments.

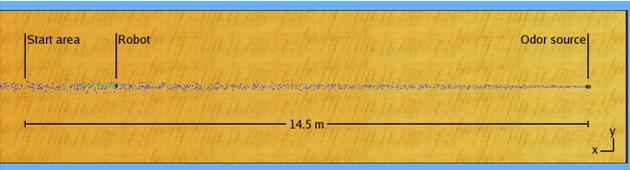


Figure 5. Simulated environment in Webots.

### 3.6 Experiments

We analyzed three parameters for each of the three algorithms, totaling to 9 sets of experiments:

	Algorithm	Specific parameter	$\sigma_a$	$d_{lost}$
A	Casting	variable	10 cm	61.4 cm
B	Casting	$\beta = 25^\circ$	variable	61.4 cm
C	Casting	$\beta = 25^\circ$	10 cm	variable
D	Surge-spiral	variable	10 cm	61.4 cm
E	Surge-spiral	$d_{gap} = 22.2$ cm	variable	61.4 cm
F	Surge-spiral	$d_{gap} = 22.2$ cm	10 cm	variable
G	Surge-cast	variable	10 cm	61.4 cm
H	Surge-cast	$d_{cast} = 27$ cm	variable	61.4 cm
I	Surge-cast	$d_{cast} = 27$ cm	10 cm	variable

Each set consists of 9 choices for the variable parameter with 50 independent runs each. In each run, the robot was released in the odor at a position about 14.5 m downwind from the target area, and the corresponding algorithm was launched. If the robot reached the odor source, the run was considered successful. If the robot touched an arena wall, the run was aborted and declared unsuccessful. Distance and upwind distance were derived from the trajectory, recorded during the run.

The forward speed of the robot (on straight lines) was 10.6 cm/s and therefore same as with the real-robot experiments in the wind tunnel. The plume threshold was set to  $c = 100$ .

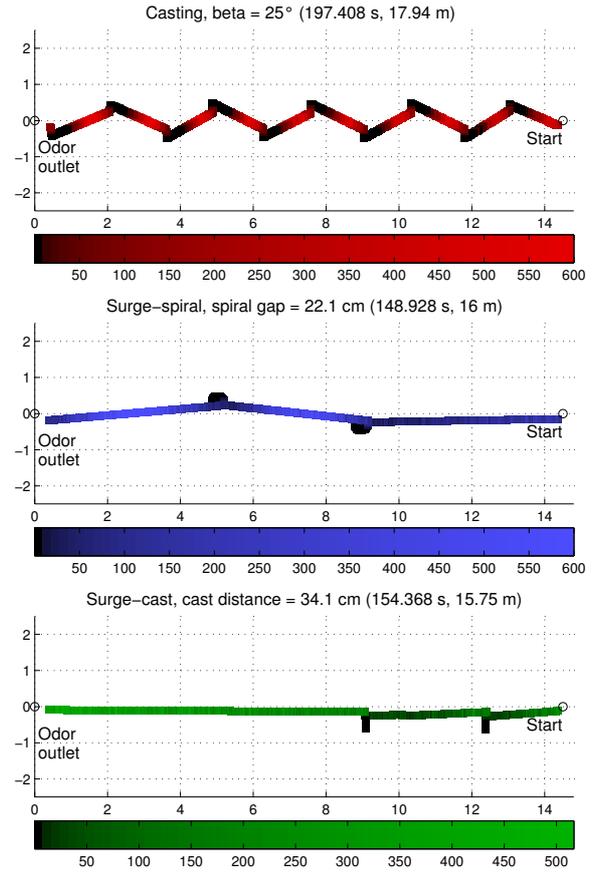


Figure 6. Successful sample runs of all three algorithms (configurations A, D and G). The bars below the plots indicate the concentration shading.

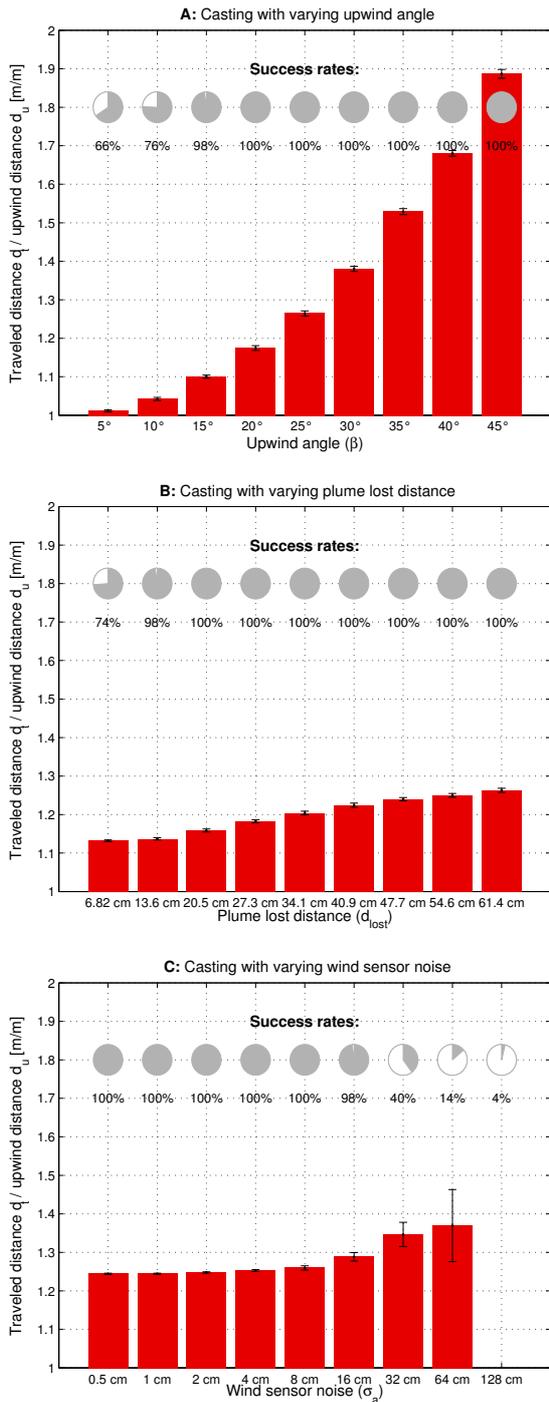
## 4 Results and Discussion

In the following paragraphs, we discuss the results for each of the three algorithms. Sample runs for three chosen parameter configurations are shown in Figure 6.

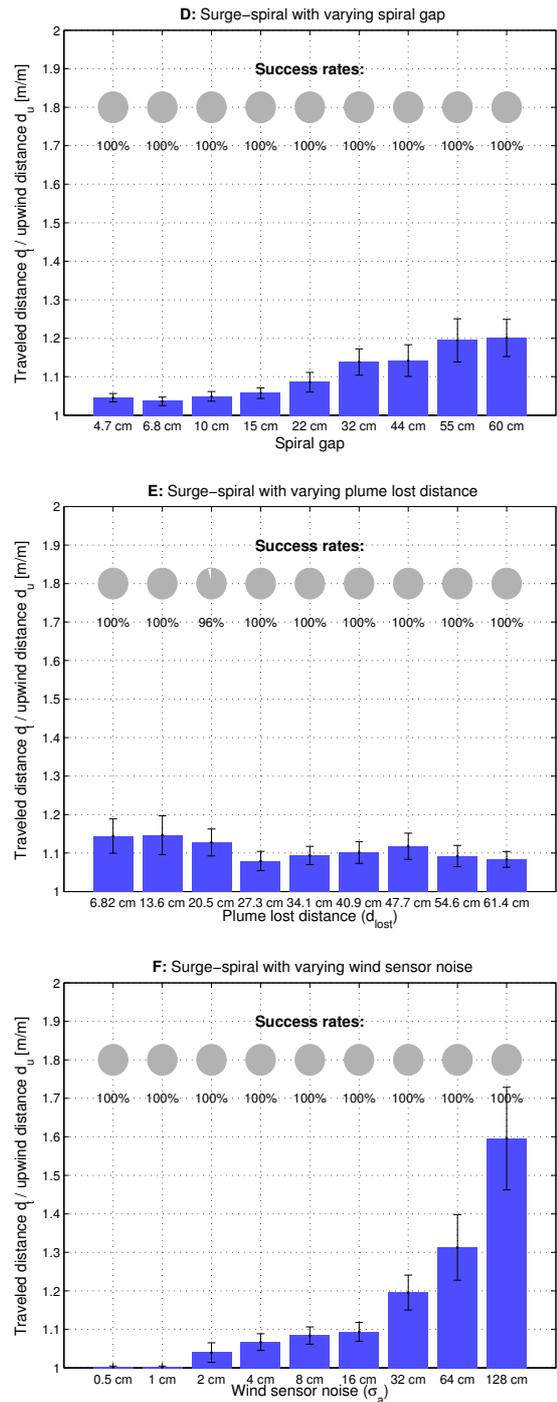
We compare the success ratio and the distance overhead of the runs. The latter is calculated as the traveled distance divided by the upwind distance ( $\frac{d_t}{d_u}$ ) and represents an excellent measure for the performance of a plume traversal algorithm.

### 4.1 Casting

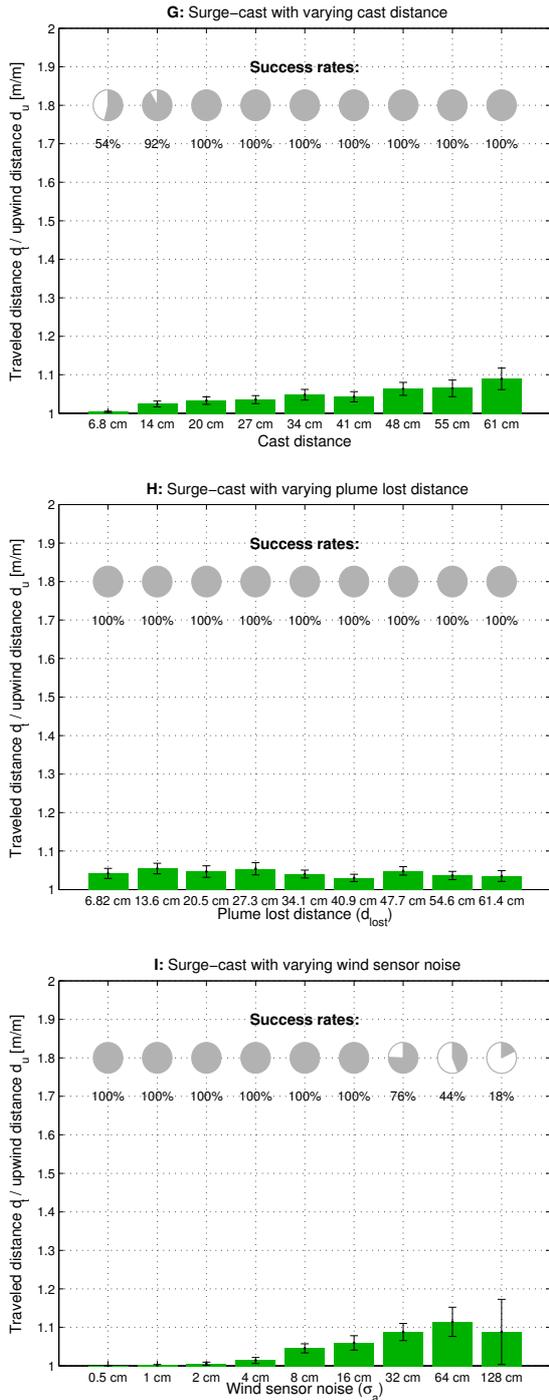
The results for the *casting* algorithm are displayed in Figure 7. The upwind angle has a major influence on the performance. Small angles yield a low distance overhead, but also a low success rate. In our setup, only configurations with  $\beta > 20^\circ$  resulted in acceptable success rates.



**Figure 7. Results obtained with the casting algorithm.** The error bars indicate the 95% confidence interval for the mean (assuming normally distributed data). **A:** With varying upwind angle ( $\beta$ ). **B:** With varying plume lost distance ( $d_{\text{lost}}$ ). **C:** With varying noise on the wind sensor reading ( $\sigma_a$ ). The last bar is omitted because of the small number of successful runs.



**Figure 8. Results obtained with the surge-spiral algorithm.** The error bars indicate the 95% confidence interval for the mean (assuming normally distributed data). **D:** With varying spiral gap ( $d_{\text{gap}}$ ). **E:** With varying plume lost distance ( $d_{\text{lost}}$ ). **F:** With varying noise on the wind sensor reading ( $\sigma_a$ ).



**Figure 9. Results obtained with the *surge-cast* algorithm. The error bars indicate the 95% confidence interval for the mean (assuming normally distributed data). G: With varying cast distance ( $d_{cast}$ ). H: With varying plume lost distance ( $d_{lost}$ ). I: With varying noise on the wind sensor reading ( $\sigma_a$ ).**

A similar behavior can be observed for the plume lost distance: on one end of the scale, the success rate drops significantly, whereas on the other end, the performance gets worse. Hence, choosing parameters for this algorithm is ultimately a trade-off between performance and robustness.

The accuracy of the wind sensor only has a marginal impact on the performance, and no visible influence on the robustness as long as the accuracy is good enough. If the noise is too high, however, the algorithm does not work at all.

Quantitatively, the simulation results are much better than the results obtained with the real robots [13]. Qualitatively, the results are very similar, though, and the *casting* algorithm remains the worst among the three algorithm in terms of distance overhead and success ratio.

## 4.2 Surge-Spiral

The *surge-spiral* algorithm is extremely robust and virtually all 1350 runs succeeded. With the simple plume used in this setup, a spiral of increasing radius will always reacquire the plume. In addition, the performance is fairly good over a wide parameter range. As expected, a small spiral gap is advantageous, at least as long as the robot reacquires the plume within one turn. Figure 8 (E) also suggests that higher  $d_{lost}$  yield slightly better performance. This, however, is simply due to the fact that the upwind steps get larger, and could have a negative influence in non-laminar flow conditions. In contrast to the casting algorithm, wind sensor accuracy only affects the distance overhead of *surge-spiral*, and not its success rate. For high noise values, the distance overhead becomes significantly larger, as the algorithm more often fails guessing on which side the plume is.

Interestingly, the simulation results seem to be slightly worse than the real-robot results [13]. This is presumably due to differences in the distribution of the wind direction measurement error.

## 4.3 Surge-Cast

The results obtained with the *surge-cast* algorithm are comparable to those of the *surge-spiral* algorithm. As expected, the distance overhead grows almost linearly with the cast distance, but at a fairly low rate. Furthermore, for very low cast distances, the algorithm fails to work reliably - the robot simply does not get back to the plume. These results for this algorithm match closely those obtained with the real robots [13].

Furthermore, the wind sensor noise seems to mainly affect the success rate, which we have observed with *casting* algorithm as well.

## 5 Conclusion

We carried out odor source localization experiments in simulation with three different bio-inspired single-robot algorithms. The setup was similar to the real-robot experiments carried out in the wind tunnel [13].

While there are some quantitative differences between the simulation results and those obtained with the real robots, our previous observations have been confirmed. Namely, pure casting is inefficient in laminar flow, and upwind surge techniques yield much better performance in terms of success rate and distance overhead.

In addition, we could show that the plume lost distance,  $d_{\text{lost}}$ , does not have a significant influence on the performance of the algorithms. Well-chosen upwind angles, spiral gap or cast distances are much more important. In addition, improving the wind direction sensor helps mainly the *surge-cast* algorithm, while large errors in the wind direction measurement impair all three algorithms.

In future work, we will test the algorithms in turbulent flow and/or meandering plume conditions.

## Acknowledgments

This work was supported by the National Competence Center in Research on Mobile Information and Communication Systems NCCR-MICS, a center supported by the Swiss NSF under grant number 5005-67322.

## References

- [1] E. Balkovsky and B. I. Shraiman. Olfactory search at high reynolds number. *PNAS*, 99(20):12589–12593, October 2002.
- [2] J. H. Berlinger and M. A. Willis. Adaptive control of odor-guided locomotion: behavioral flexibility as an antidote to environmental unpredictability. *Adaptive Behavior*, 4(3-4):217–253, August 1996.
- [3] 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*, 2:143–169, 2002.
- [4] G. Ferri, E. Caselli, V. Mattoli, A. Mondini, B. Mazzolai, and P. Dario. A biologically-inspired algorithm implemented on a new highly flexible multi-agent platform for gas source localization. In *Proceedings of the First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechanics (BIROB 2006)*, February 2006.
- [5] D. W. Gage. Many-robot MCM search systems. In *Proceedings of the Autonomous Vehicles in Mine Countermeasures Symposium*, pages 9.56–9.64, April 1995.
- [6] A. T. Hayes, A. Martinoli, and R. M. Goodman. Distributed odor source localization. *IEEE Sensors Journal*, 2(3):260–271, June 2002.
- [7] A. T. Hayes, A. Martinoli, and R. M. Goodman. Swarm robotic odor localization: Off-line optimization and validation with real robots. *Robotica*, 21:427–441, 2003.
- [8] H. Ishida, T. Nakamoto, T. Moriizumi, T. Kikas, and J. Janata. Plume-tracking robots: A new application of chemical sensors. *Biological Bulletin*, (200):222–226, April 2001.
- [9] 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*, 31(2):103, June 2006.
- [10] W. Li, J. A. Farrell, and R. T. Cardé. Tracking of fluid-advected odor plumes: Strategies inspired by insect orientation to pheromone. *Adaptive Behavior*, 9(3-4):143–170, 2001.
- [11] W. Li, J. A. Farrell, S. Pang, and R. M. Arrieta. Moth-inspired chemical plume tracing on an autonomous underwater vehicle. *IEEE Transactions on Robotics*, 22(2):292–307, April 2006.
- [12] A. J. Lilienthal, D. Reiman, and A. Zell. Gas source tracing with a mobile robot using an adapted moth strategy. In *Autonome Mobile Systeme (AMS), 18. Fachgespräch*, pages 150–160. GDI, December 2003.
- [13] T. Lochmatter and A. Martinoli. Tracking odor plumes in a laminar wind field with bio-inspired algorithms. In *Proceedings of the 11th International Symposium on Experimental Robotics 2008 (ISER 2008)*, Springer Tracts in Advanced Robotics, Athens, Greece, July 2008. to appear.
- [14] T. Lochmatter, X. Raemy, L. Matthey, S. Indra, and A. Martinoli. A comparison of casting and spiraling algorithms for odor source localization in laminar flow. In *Proceedings of the 2008 IEEE International Conference on Robotics and Automation (ICRA 2008)*, pages 1138–1143, May 2008.
- [15] M. Long, A. Gage, R. Murphy, and K. Valavanis. Application of the distributed field robot architecture to a simulated demining task. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA 2005)*, pages 3193–3200. IEEE, April 2005.
- [16] O. Michel. Webots: Professional mobile robot simulation. *International Journal of Advanced Robotic Systems*, 1(1):39–42, 2004.
- [17] R. A. Russell. *Odour Detection by Mobile Robots*, volume 22 of *World Scientific Series in Robotics and Intelligent Systems*. World Scientific Publishing Company, 1999.
- [18] G. S. Settles. Sniffers: Fluid-dynamic sampling for olfactory trace detection in nature and homeland security—the 2004 freeman scholar lecture. In *Journal of Fluids Engineering*, volume 127 of *Transactions of the ASME*, pages 189–218, 2005.
- [19] M. Vergassola, E. Villermaux, and B. I. Shraiman. ‘info-taxis’ as a strategy for searching without gradients. *Nature*, 445:406–409, January 2007.
- [20] B. Webb, R. R. Harrison, and M. A. Willis. Sensorimotor control of navigation in arthropod and artificial systems. *Arthropod Structure and Development*, 33:301–329, May 2004.