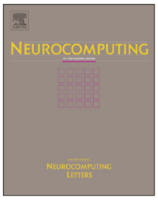
Neurocomputing 121 (2013) 481–497

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Evolutionary robotics approach to odor source localization

G.C.H.E. de Croon [a,b,*], L.M. O'Connor [a], C. Nicol [c], D. Izzo [a]

[a] *Advanced Concepts Team of the European Space Agency, ESTEC, The Netherlands*
[b] *Faculty of Aerospace Engineering, Delft University of Technology, The Netherlands*
[c] *Department of Mechanical and Aerospace Engineering of Carleton University, Canada*

ARTICLE INFO

Article history:
Received 27 January 2013
Received in revised form
23 April 2013
Accepted 13 May 2013
Communicated by H.R. Karimi
Available online 25 June 2013

Keywords:
Odor source localization
Neural controller
Evolutionary robotics
Continuous time recurrent neural network

ABSTRACT

An Evolutionary Robotics (ER) approach to the task of odor source localization is investigated. In particular, Continuous Time Recurrent Neural Networks (CTRNNs) are evolved for odor source localization in simulated turbulent odor plumes. In the experiments, the simulated robot is equipped with a single chemical sensor and a wind direction sensor. Three main contributions are made. First, it is shown that the ER approach can be successfully applied to odor source localization in both low-turbulent and high-turbulent conditions. Second, it is demonstrated that a small neural network is able to successfully perform all three sub-tasks of odor source localization: (i) finding the odor plume, (ii) moving toward the odor source, and (iii) identifying the odor source. Third, the analysis of the evolved behaviors reveals two novel odor source localization strategies. These strategies are successfully re-implemented as finite state machines, validating the insights from the analysis of the neural controllers.

1. Introduction

Odor source localization is a well-studied topic in the field of robotics. Robust strategies for autonomous odor source localization may serve many applications ranging from the localization of gas leakages to that of people trapped in collapsed buildings. For space agencies, the application of interest is the autonomous localization of odor sources on other planets such as Mars [12,23,35,26,21].

The odor source localization task is generally stated to consist of three sub-tasks: (i) finding the odor plume, (ii) following the odor plume up to the source, and (iii) recognizing the odor source.[1] The difficulty of odor source localization heavily depends on the context. For example, a factor that strongly influences the task's difficulty is the way in which odor is dispersed. At low Reynolds numbers, where viscosity dominates, diffusion results in smooth variations of the odor concentration. This allows the use of localization strategies that attempt to move up the concentration gradient [16,31,32]. At medium to high Reynolds numbers, advection and turbulence determine the way in which the concentration changes over time. As a consequence, the odor plume can become patchy, with pockets containing local maxima of odor concentration. Under such turbulent conditions, common for most robotic applications, gradient search strategies tend to fail. Other contextual factors determining the task's difficulty include the robot's sensory apparatus and computational capabilities.

Many odor localization strategies have been proposed for the different possible contexts (see the surveys in [22,17]). Coarsely, two main approaches can be discerned in the literature. The first is a probabilistic approach to odor source localization (cf. [9,39,46,34,33]). Probabilistic strategies employ a 'belief map': a spatial map that keeps track of the probabilities for source presence. The robot's movements can be determined such as to reduce the uncertainty on the source location. This strategy has been named 'infotaxis' [46]. It is successful in the difficult context of turbulent conditions and a sensor suite consisting of a single chemical and wind sensor, but the computational complexity of the algorithms can be considerable.

The second approach draws inspiration from biology. For example, odor source localization algorithms have been based on the strategies of bacteria (*Escherichia coli*) [10], dung beetles (*Geotrupes stercorarius*), and silkworm moths [24,25,4,32,42,29]. The algorithms generally can be interpreted as finite state machines with in each state simple reactive rules to the incoming inputs. Therefore, the algorithms are computationally efficient. A drawback of biological strategies is that suitable behavior for the animals on which they are inspired may not be a suitable behavior for the robot. Moreover, this approach may be hard to apply to an environment that is very different from the ones in which animals have been evolved. For extraterrestrial environments such as Mars, this may be the case.

An alternative approach to the above ones is to heuristically optimize controllers for the specific context of the odor source localization task. This 'Evolutionary Robotics' (ER) approach [38] to odor source localization is of interest for the following reasons.

[*] Corresponding author. Tel.: +31 648255416.
E-mail address: guido.de.croon@gmail.com (G.C.H.E. de Croon).

[1] Most studies exclusively focus on the second sub-task, of following the odor plume to the source.

0925-2312/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2013.05.028

First, it permits the development of computationally efficient solutions. Often in the field of ER small neural networks are used as controllers, such as feedforward neural networks or Continuous Time Recurrent Neural Networks (CTRNNs) [1]. Despite their limited computation, small neural network controllers have been shown to tackle various relatively complex tasks by means of sensorimotor coordination [37]. Second, having a heuristic optimization of the controllers will result in strategies that depend on the specific context and not so much on the designer's bias. This may lead to novel and robust strategies that have not been designed or observed before.

The present work is not the first application of the ER approach to odor source localization. It has first been studied in [3,24]. Later, in [18,19], the approach is used in order to shed more light on the neural circuitry underlying the klinotaxis behavior of *Caenorhabditis Elegans*, which performs a gradient ascent search under low Reynolds conditions. These previous studies employed simplified models of odor dispersion, with a smooth gradient to the source of either the concentration [3,18,19] or the probability of detecting odor particles [24]. In a very recent study [40] the generalization of the ER approach to the more difficult case of a turbulent odor plume is proposed. However, in [40] the focus lies on the experimental setup in which virtual odor plumes are combined with a real robot, and no experimental results are given. Therefore, it remains uncertain whether this more difficult setting can be tackled by the ER approach.

In this article, the ER approach is employed for performing a simulated odor source localization task under turbulent conditions. In the experiments, a small neural network will have to perform all three sub-tasks: plume finding, plume following, and odor source identification. In particular, the weights and time constants of a CTRNN are optimized with an evolutionary algorithm. The sensor suite of the simulated robot is rather minimal, consisting of one chemical sensor and a wind sensor. Experiments are performed for both low-turbulent and high-turbulent scenarios, leading to *three main contributions*. First, it is shown that the ER approach can be successfully applied to odor source localization in both low-turbulent and high-turbulent conditions. The evolved behaviors show robust performance under various conditions. Second, it is demonstrated for the first time that a small neural network is able to successfully perform all three sub-tasks of odor source localization. The simulated robot is able to find and follow the odor plume, while recognizing when it is close to the source. Third, both for the low-turbulent scenarios and high-turbulent scenarios, the analysis of the evolved behaviors reveals odor source localization strategies that have not been treated in the literature before.

The remainder of the article is organized as follows. In Section 2, the experimental setup is explained. Then, the experiments on low turbulence (Section 3) and high turbulence (Section 4) scenarios are presented. Finally, conclusions are drawn in Section 5.

2. Experimental setup

The odor source localization experiments are performed in simulation. In this section, first the simulated environment is explained, followed by the robot and its neural controller. Finally, the setup of the optimization with an evolutionary algorithm is discussed.

2.1. Simulated environment

For the behavior of the odor plume over time, a model is used that was made specifically for simulating methane spreading on Mars [36]. The model simulates the methane concentration at points in space and time taking into account reactions which release the substance, advection which carries the plume as a whole downwind, and molecular and turbulent diffusion responsible for vortices and small-scale turbulence. The methane source is modeled as a fixed location in the environment at which instantaneous methane release occurs at a given frequency ($\frac{1}{6}$ Hz) in the experiments.

The model was used to generate two sets of scenarios: low turbulence and high turbulence. Fig. 1 shows the methane concentration in an example low-turbulence scenario (left) and high-turbulence scenario (right), both with a simulated area of 80 m × 80 m. The most important observation to be made from the figure is that the methane model correctly captures the property of turbulent plumes that puffs of odor travel down the plume, with the concentration gradient not always leading toward the source. In addition, there are many areas in the plume with little to no odor concentration. This being said, the difference between the two scenarios is that the low-turbulence scenario involves a more diffuse plume than the high turbulence scenario. Robot controllers were evolved on both cases (Sections 3 and 4).

As mentioned in the introduction, the robot will have to learn how to search the plume. If the environment has only one source, it may be problematic to learn search behavior. If the robot does not start in a fixed position, it may perform a correct search behavior, but never find the methane plume. For example, when starting above the single plume shown in Fig. 1 while facing the wind, a good search strategy such as moving cross-wind to the

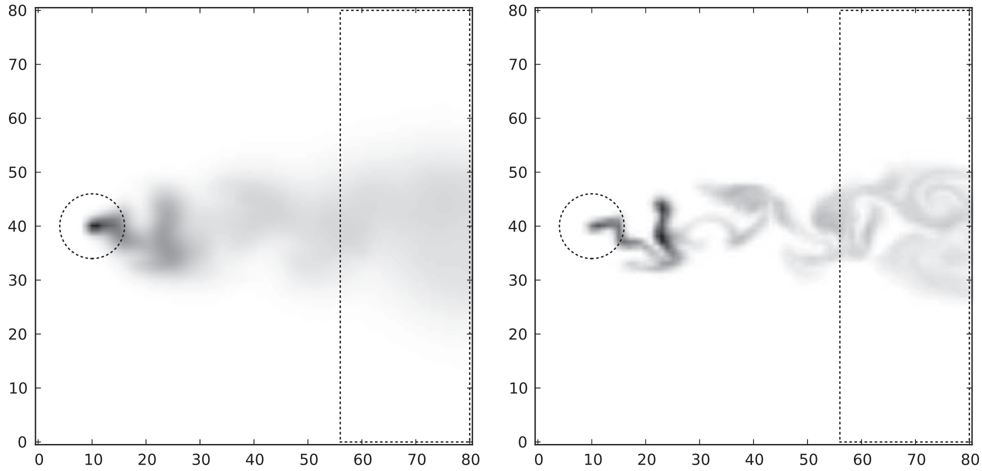


Fig. 1. *Left*: methane concentration in a low-turbulence scenario. High/low methane concentrations are represented by dark/bright pixels, respectively. The circle has a radius of 6 m and is centered on the methane source. The box in the right part of the environment is the area in which robots are initialized during evolution (see Section 2.4). *Right*: methane concentration in a high-turbulence scenario.

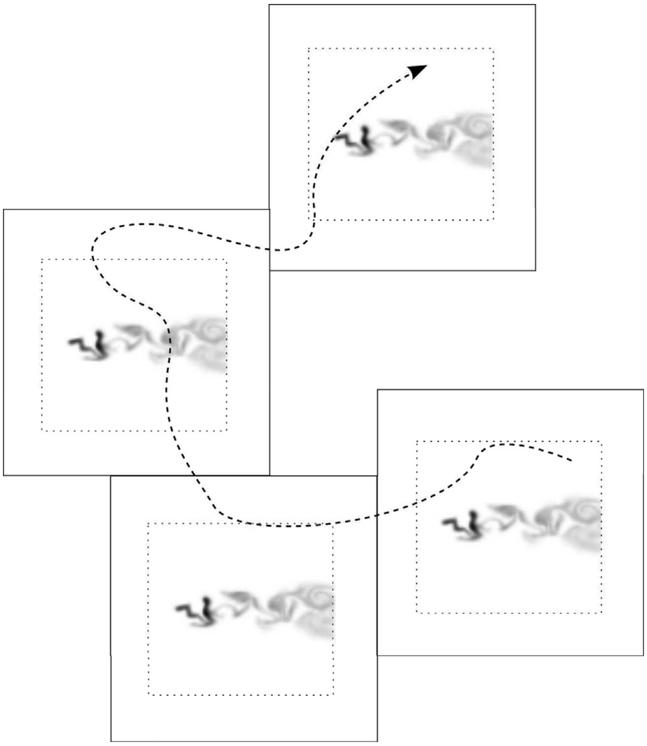


Fig. 2. Illustration of the random toroid concept. The simulated plume (area within the dotted box) is embedded in a larger area by putting a fixed size border (solid box) around it with no methane concentration. When the robot crosses a border (see the dashed path in the figure), it reappears at a random coordinate at the other side of the environment. Effectively, the robot is then searching in a much larger area with many unevenly spaced methane sources.

right will make the robot go further up. Although it will never find a plume, searching cross-wind is known to be optimal in the absence of other knowledge [42]. For this reason, the simulated environment contains not one source, but a large number of distinct methane sources at different locations. Because generating an enormous environment with multiple sources would be computationally prohibitive, a *random toroid* world is introduced. Fig. 2 illustrates the concept. The simulated methane plume is placed in a larger area, by creating a border around it. In the border area, the odor concentration is 0, while the wind is distributed similarly to the center area containing the plume. When the robot crosses a border such as the one on the top, it does not reappear exactly at the opposite side as would be the case for a normal toroid world. Instead, it reappears at a random x-position. This effectively implies that the robot moves in a large environment with many unevenly spaced methane sources.[2] The difficulty of the search task depends on the ratio of the plume area that can be sensed by the robot divided by the total area including the border. In the experiments, the border is set to $b = 20$ m on all sides.

2.2. Robot

The robot is modeled as a non-holonomic vehicle with two wheels, which can only move forward. The robot is located in two-dimensional space (x,y) with a heading, ψ, velocity, v, and angular velocity, $\dot{\psi}$. The velocity and angular velocity of the robot are both limited, with $v \in [0, v_{max}]$, $v_{max} = 0.4$ m/s and $|\dot{\psi}| \leq 0.5$ rad.

The robot only has sensor readings expressed in a body frame. While the successful moth odor localization strategies use two chemical sensors and a wind sensor (cf. [24,25,32,42]), the experiments in this article focus on the more difficult case of having only a single chemical sensor and a wind sensor (cf. [15,46,14,27]), both placed at the center of the robot. Having a single chemical sensor is more difficult, since it does not allow to determine an instantaneous chemical gradient. Below, the measurements of the chemical and wind sensor are explained in more detail.

The experiments in this article are not intended for a specific robot setup. Instead of modeling the performance of a particular gas sensor, it is assumed in the robot simulation that the sensor can measure the concentration relatively accurately above a threshold concentration, and has a refresh rate of 2 Hz. These assumptions appear to be reasonable given the current state of gas sensor research [41,15,28,47]. Specifically, the odor concentrations of the methane model described in Section 2.1, are mapped to the set of integers $\mathcal{C} = \{0, 1, 2, \ldots, 255\}$. This was done to reduce memory requirements for storing the model matrices over time, and to limit the resolution of the concentration sensor. Furthermore, depending on the experiment, a threshold C_{thr} is used, below which all measurements are mapped to 0.

It is assumed that the robot is equipped with a wind sensor that is able to determine the wind direction. The sensor performs measurements at 2 Hz. This setup is closely matched by commercially available wind sensors (cf. [17]).

2.3. Neural network controller

In the interest of computational efficiency, the robot will be controlled by a small neural network. Since the controller has to explicitly identify the odor source, it will be advantageous if the network has a memory. This rules out feedforward neural networks, which instantaneously map sensory inputs to actions (including the source identification). There are different manners in which neural networks can be endowed with memory, e.g., by introducing recurrent connections. However, neural networks that only have recurrent connections as a means to establish a memory (such as Elman networks [8]) have more difficulties capturing slow memory dynamics than neural networks with additional neural inertia mechanisms[3] [6]. Since such slow memory dynamics may be useful for this task, the robot is controlled by a Continuous Time Recurrent Neural Network (CTRNN). CTRNNs have been applied successfully to many robotic tasks requiring a memory (cf. [13,2,11,45]).

Fig. 3 shows the connection structure of the CTRNN. Arrows between the boxes around the neural layers indicate that they are fully connected. The hidden layer is fully connected to itself via recurrent connections. The network used in the experiments receives 4 sensory inputs and consists of 10 hidden neurons and 4 output neurons. The number of hidden neurons was tuned with the help of preliminary experiments, where 10 gave better results than for instance 5 hidden neurons.

From the single chemical sensor and single wind sensor, four sensory inputs $\mathbf{s}$ are distilled to serve as input to the CTRNN:

$$s_1 = \frac{c - c_f}{c_f} \tag{1}$$

$$s_2 = 2\left(\frac{c_f}{c_{max}}\right) - 1 \tag{2}$$

[2] In a normal toroid world the methane sources would be evenly spaced, implying that they have a fixed spatial relation to each other. A robot could then easily scan the entire arena just by keeping a fixed angle with the wind.

[3] Neural inertia means that the neural activations are low-pass filtered over time, where the properties of the filters depend on the neurons' time constants.

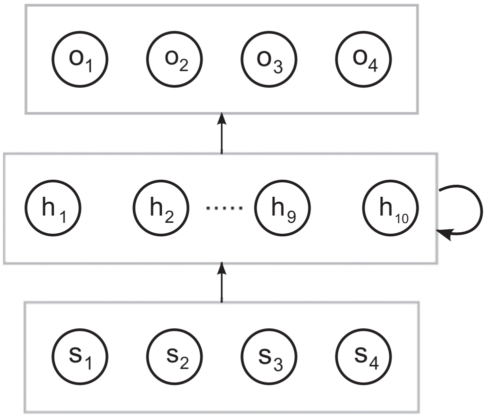


Fig. 3. Connection structure of the CTRNN transforming the sensory inputs **s** into the output activations **o**. Each layer of neurons is represented with a box. An arrow implies that the layer is fully connected. The hidden layer is fed back to itself.

$$s_3 = 2\left(\frac{|\text{atan }2(-u_y, -u_x) - \psi|}{\pi}\right) - 1 \tag{3}$$

$$s_4 = 2\left(\frac{\left|\text{atan }2(-u_y, -u_x) - \left(\psi + \frac{\pi}{2}\right)\right|}{\pi}\right) - 1 \tag{4}$$

where s_j is the jth input, c is the odor concentration sensed by the robot at it's current location, c_f is a low-pass filter of the odor concentration (see Eq. (5)), c_{max} is the maximum odor concentration that can be measured with the chemical sensor, and u_x and u_y are the strength of the wind in the x and y directions, respectively, at the robot's current location. We use the negative of the wind vector so that s_3 is equal to zero when the robot is heading directly into the wind. c_f is updated each time step (0.5 s) using

$$c_f \leftarrow 0.9c_f + 0.1c \tag{5}$$

and is initialized to 1.0. Since c_f is updated immediately after c, there can be no division by zero in the calculation of s_1. Moreover, the maximal possible s_1 occurs when c_f is (close to) zero and the robot encounters the maximal $c : \max(s_1) = 255/25.5 = 10$.

Inputs s_1 and s_2 convey information about the odor concentration and the concentration gradient. $s1$ indicates whether the robot is moving up or down the concentration gradient by subtracting c_f from the odor concentration sensed by the robot. This is divided by c_f to ensure that s_1 remains within a similar range regardless of the absolute concentration at the robot's location. s_2 is c_f normalized by the maximum possible odor concentration.

Inputs s_3 and s_4 convey information about the angle between the wind direction and the robot's heading (Fig. 4). We use the magnitude of the angle between the wind vector and the robot heading vector, and the magnitude of the angle between the wind vector and a vector perpendicular to the heading vector. This representation prevents discontinuities in input values as the robot rotates.

All hidden neurons have an activation that changes over time according to the equation [3,1]:

$$\tau_i \dot{y}_i = -y_i + \sum_{j=1}^{N} w_{ji}\sigma(y_j + \theta_j) + \sum_{k=1}^{M} w_{ki}s_k(t) \tag{6}$$

where y_i is the activation of the ith hidden neuron, τ_i is the neuron's time constant, N is the number of hidden neurons, M the number of sensory inputs, w_{ji} is the weight of the connection between the ith neuron and the jth neuron, σ is the activation function, θ_j is a bias term, w_{ki} is the weight between the kth sensory input and the ith neuron, and $s_k(t)$ is the value of the kth sensory input at time t. In our case, σ is the hyperbolic tangent function.

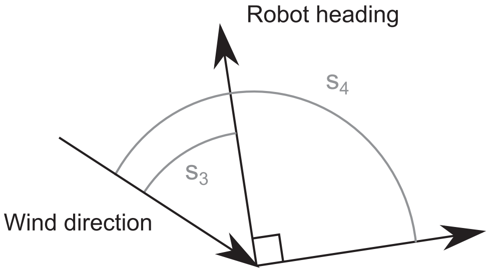


Fig. 4. Illustration showing how sensory inputs s_3 and s_4 are calculated from the wind direction and robot heading. s_3 and s_4 are normalized to the range $[-1, 1]$.

Each output is calculated as follows:

$$o_i = \sigma\left(\sum_{j=1}^{N} w_{ij}\sigma(y_j + \theta_j) + \theta_i\right) \tag{7}$$

where o_i is the ith output, σ is the hyperbolic tangent function and θ_i is a bias term applied to the output. The outputs are in the range [−1,1].

The velocity and heading of the robot are updated using the outputs as follows:

$$v = \left(\frac{o_3 + 1}{2}\right)v_{max}$$

$$\dot{\psi} = \left(\frac{o_2 - o_1}{2}\right)\dot{\psi}_{max}$$

where v_{max} and $\dot{\psi}_{max}$ are the maximum velocity and angular velocity of the robot respectively and positive $\dot{\psi}$ corresponds to the robot turning left. o_4 is used to identify whether the robot is at the odor source, as will be explained in the next subsection.

2.4. Evolutionary optimization

Evolutionary optimization is used for tuning the weights and time constants of the CTRNN to the odor source localization task. The choice for this heuristic optimization method is motivated by the following three reasons. The first and main motivation derives from the problem structure: it is a Partially Observable Markov Decision Process (POMDP). The decision process is such that the desired outputs of the neural network are unknown, ruling out optimization with for instance backpropagation. Moreover, the *partial* observability invalidates the Markovian assumption made by most online reinforcement learning methods such as Q-learning or Temporal-Difference methods [44]. Indeed, for non-Markovian tasks evolutionary optimization seems to provide better results [7,48]. Second, evolutionary optimization allows the combined optimization of different types of parameters (cf. [3,30]). In this case, the time constants of the CTRNN can easily be included in the optimization process. Third, evolutionary optimization is effective in avoiding local maxima when applied to the optimization of neural networks [49].

In particular, the weights and time constants of the CTRNN were optimized using the 'simple genetic algorithm' from the open-source PyGMO/PAGMO package [20]. The population size μ is 30 and the number of selected individuals for reproduction is $\lambda = 6$. Selection was performed according to a roulette scheme, where the probability of selection P of individual i is equal to

$$P(i) = \frac{\max(C) - C(i)}{\sum_{j=1}^{\mu}(\max(C) - C(j))}, \tag{8}$$

where $C(i)$ is the cost function evaluated for individual i, and $\max(C)$ is the worst performing individual of the generation. The best performing individual is copied once directly for insertion in the new generation (one elite individual). Its remaining offspring and the offspring of the other selected individuals are created under influence of exponential crossover (with a probability of $P_c = 0.1$ per offspring) and Gaussian mutation (with a probability of $P_m = 0.03$ per gene in the genome). The evolution carries on for 400 generations. All weights were restricted to the interval $[-2, 2]$. The time constants were encoded as $1/\tau \in [0, 1]$. The parameter settings of the evolutionary algorithm have been tuned in the preliminary experiments. Most importantly, the number of generations was increased to the point that the evolutionary algorithm could find a solution to all three phases of the task.

The evolutionary optimization was performed by evaluating the individuals on three different scenarios that all have the same amount of turbulence, but have been generated with different random seeds. During evolution, each individual performs two 'runs' per scenario, leading to a total of $R = 6$ runs. Each run commences with the initialization of the robot at a random location in the right-end of the field (with $x \in [56, 80]$ and $y \in [0, 80]$). As a consequence, the robot always starts relatively far from the source, while being above, in, or below the plume. The robot's heading is initialized randomly in $[-\pi, \pi]$. The robot then starts acting and sensing at 2 Hz until the simulation ends at $T = 1200$ s.

The cost function minimized by the evolutionary algorithm consists of multiple components. In particular, it is a weighted sum of

1. The distance from the robot to the source at the end of the run ($d_{t=T}(i, S)$, where S is the source).
2. The negative average concentration sensed by the robot $-\bar{c}$.
3. A negative bonus if the robot at any point on its trajectory passes at a distance smaller than 6 m to the source ($-B_1$).
4. A negative bonus if the robot succeeds in staying within 6 m of the source to the end of the run ($-B_2$).
5. An evaluation of how well the robot 'identifies' the source. The identification of the source is evaluated by determining the mutual information between a 'binarized' version of the identification output of the neural network ($O_4 = (o_4 \geq 0)$) and a binary variable Q representing whether the robot is inside or outside of a circle of 6 m around the source. The mutual information between these variables is indicated with $I(Q; O_4)$ (see Eq. (9)).

Component (1) in the list above can be considered as the main goal for searching the plume and following it to the source. Components (2)–(4) have been included in the fitness function in order to facilitate evolutionary learning. For example, component (2) is especially important at the start of evolution to reward individuals that find the plume. Later during evolution, the component stays of importance, as there is on average a higher concentration of odor close to the source. For the identification component (5), mutual information was chosen as a performance measure [43]:

$$I(Q; O_4) = H(Q) - H(Q|O_4) = \tag{9}$$

$$-\sum_{q \in Q} P(q) \log_2(P(q)) + \sum_{o \in O_4} P(o) \sum_{q \in Q} P(q|o) \log_2(P(q|o)) \tag{10}$$

The motivation behind mutual information is that it leaves the robot free in its manner to identify the source. It could equally well do so with a high as with a low output o_4. In addition, the mutual information $I(Q; O_4)$ is zero when the robot never passes close to the source. As a consequence, the optimization of the identification output only becomes important later during evolution, when the robots solve the task of finding the odor source.

More formally, given a number of runs R, the cost function C for an individual i is defined as follows:

$$C(i) = \frac{1}{R} \sum_{r=1}^{R} C_r(i), \tag{11}$$

$$C_r(i) = w_d d_{t=T}(i, S) - w_c \bar{c} - B_1 - B_2 - w_s \frac{I(Q; O_4)}{H(Q)} \tag{12}$$

Note how the mutual information in Eq. (12) is normalized by the entropy $H(Q)$, which maps the term to the interval $[0, 1]$ (assuming $0/0 = 0$). Concerning the weights of the components, preliminary experiments led to the following settings: $w_d = 1$, $w_c = 2$, $B_1 = 50$, $B_2 = 50$, and $w_s = 150$.

Experiments will be performed both for low turbulence scenarios (Section 3) and high turbulence scenarios (Section 4). The low turbulence scenarios already have patches of odor moving downwind, but still have a rather diffuse plume. In the high turbulence scenarios, the odor forms filaments of patches moving downwind, without a diffuse plume. The low turbulence scenarios are expected to be easier to solve.

For each type of scenario, 6 different evolutionary runs will be performed. During each evolutionary run, the best individual of each generation is stored in a file. After evolution, all these individuals are post-evaluated on the three training scenarios with $R = 30$ (ten runs per scenario). The individual with the lowest average cost is then selected as the best individual of that evolutionary run. These best individuals are tested on a separate, fourth test scenario. All test results mentioned in the following sections are obtained on the low-turbulence/high-turbulence test scenarios.

2.5. Comparison with benchmark strategy

Of course, the results are only relevant if the performance of the evolved controllers is at a par with state-of-the-art strategies in the literature. Since the focus is on computationally efficient methods, a comparison will be made with a biologically inspired odor source localization algorithm. In [42] different biologically inspired strategies are clearly described as finite state machines, which led to the use of these models for benchmarking (cf. [10]). In this article, the silkworm moth strategy is chosen for comparison, since it has been well-studied [24,25,4,32,42,29] and can handle turbulent odor plumes. There are two main differences with the evolved neural controllers proposed in this article. First, the moth strategy receives inputs from two spatially separated chemical sensors instead of one chemical sensor. The two chemical sensors are placed symmetrically 30° away from the heading position, at a distance of 0.75 m of the body center. The access to two spatially separated chemical sensors can be considered an advantage for the moth strategy. Second, the moth strategy only covers the subtask of following the odor plume to the source, foregoing plume search and source identification. Of course, an ad hoc combination of strategies for all three sub-tasks could be made, but this would result in a new overall strategy and reduce the usefulness of the comparison. Therefore, the evolved controllers and the silkworm moth strategy are only compared on the plume following task, with final distance to the source as a performance measure.

Fig. 5 replicates the algorithm description as given in [42]. It receives as sensory inputs two odor concentrations and the wind angle. The algorithm has four parameters that determine its overall behavior: the movement duration t_m and velocity v (which together determine the movement distance d mentioned in Fig. 5), the detection threshold C_m (determining the outcome of the "Odor detected?" tests), and the turn rate ψ determining the size of the circles made. In the interest of a fair comparison, these variables are optimized with the evolutionary algorithm explained in

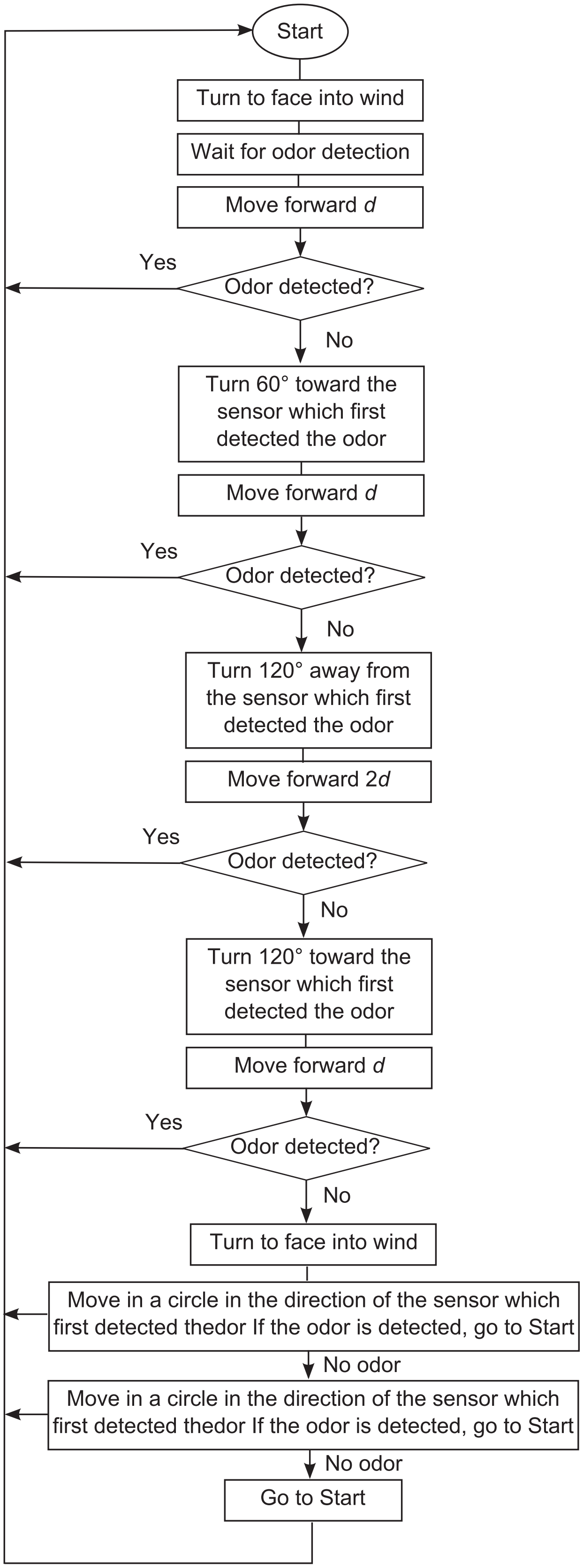

Start
Turn to face into wind
Wait for odor detection
Move forward d
Odor detected?
Yes
No
Turn 60° toward the sensor which first detected the odor
Move forward d
Odor detected?
Yes
No
Turn 120° away from the sensor which first detected the odor
Move forward 2d
Odor detected?
Yes
No
Turn 120° toward the sensor which first detected the odor
Move forward d
Odor detected?
Yes
No
Turn to face into wind
Move in a circle in the direction of the sensor which first detected thedor If the odor is detected, go to Start
No odor
Move in a circle in the direction of the sensor which first detected thedor If the odor is detected, go to Start
No odor
Go to Start

Fig. 5. Silkworm moth algorithm from [42].

the previous subsection for the same training scenarios as the neural network controllers. During optimization, the parameters are restricted to $t_m \in [0,\ 10]s$, $v \in [0,1]$ m/s, $C_m \in [0,\ 100]$, and $\dot{\psi} \in [0,72]^{\circ}/\mathrm{s}$.

3. Low turbulence scenarios

Experiments have been performed on the low turbulence scenario with a concentration measurement threshold $C_{\text{thr}} = 0$ and $C_{\text{thr}} = 20$. Both types of experiments provided successful solutions. However, in this section the focus will be on $C_{\text{thr}} = 0$, since it resulted in an interesting and novel strategy for odor source localization under relatively mild conditions. The setting of $C_{\text{thr}} = 20$ makes the plume as sensed by the robot much patchier and essentially results in the same type of strategy that will be discussed for the high turbulence scenario in Section 4. Note that with $C_{\text{thr}} = 0$, on average 22.0% of the environment contains odor that can be sensed by the robot.

3.1. Results

For the low turbulence scenarios with $C_{\text{thr}} = 0$, all six evolutionary runs resulted in successful odor source localization. Table 1 shows the performances of the best individuals on the test scenario, with $R = 1000$. The table shows the average fitness $\overline{C}$, but also the average distance to the source at the end of the run $\overline{d_{t=T}}$, the proportion of times that the robot is close to the source (≤ 6 m) at the end of the run $P(d_{t=T} \leq 6 \text{ m})$, and the normalized average mutual information $I(O_4; Q)/H(Q)$.

The best fitness is obtained by evolution number 2, with an average performance of $\overline{C} = -289$. This individual, referred to as robot 2, on average ends up at $d = 2.60$ m from the source, while being closer than 6 m to it in 97.4% of the cases. Although robot 1 scores slightly better on these two criteria ($d_{t=T}$, $P(d_{t=T} \leq 6 \text{ m})$), robot 2 is best at identifying the source. It has an average $I(O_4; Q) = 0.731$.

Fig. 6 shows a typical run of robot 2. The robot's trajectory is represented with circle markers, while the dashed circle represents a distance of 6 m to the source. White markers correspond to an output $o_4 \geq 0$, while black markers correspond to $o_4 < 0$. The underlying image is a snapshot of the plume at the end of the simulation. The robot starts the run at a location ‘above’ the plume. It then starts searching obliquely cross-wind toward the bottom left. At $t = 117$ s, the robot senses the plume and subsequently starts moving upwind to the source. The robot enters the 6 m distance circle at $t \sim 275$ s. It first circles around the source, to finally come to a stop while indicating with $o_4 < 0$ that it has arrived at the source (as shown by the black markers).[4]

The capability of robot 2's strategy to find the source while starting from other locations than the ones used in evolution is tested as follows. The robot is initialized at all locations of a 80×80 grid in the environment consisting of the simulated plume, excluding its border ($R = 6400$ runs). For each initialization location, the robot performs the odor localization task and the components of the cost function are stored in 80×80 matrices. The left part of Fig. 7 shows the final distance to the source from 0 m (white) to 13.8 m in the top image and 484 m in the bottom image (black). The top image was generated under normal conditions with the random toroid world and the bottom image was generated with a single odor plume in an infinite world. The main observation from the figure is that at most positions the robot succeeds in approaching the source quite closely, with many more bright than dark values. Furthermore, in the case where there is a single plume in an infinite world, the shape of the whitest area is related to the location of the averaged plume. The asymmetry of the coarse plume shape is caused by the robot's search strategy.

[4] For a video of the robot's behavior the reader is referred to http://www.youtube.com/watch?v=6IOOkIv3sX8&list=PL5B4AE2F9652E581C. Note that in the video, the robot starts from a different initial position.

Table 1
Performance on the low-turbulence scenario.

	C	$d_{t=T}$	$P(d_{t=T} \le 6$ m)	$I(O_4;Q)/H(Q)$
1	−262(± 60.2)	1.36(± 4.96)	0.985	0.558(± 0.166)
2	−289(± 75.8)	2.60(± 8.68)	0.974	0.731(± 0.158)
3	−198(± 96.6)	5.15(± 11.8)	0.889	0.577(± 0.253)
4	−272(± 42.0)	3.19(± 12.3)	0.953	0.693(± 0.198)
5	−262(± 83.2)	3.22(± 9.46)	0.958	0.631(± 0.180)
6	−283(± 80.8)	2.52(± 8.59)	0.968	0.674(± 0.159)

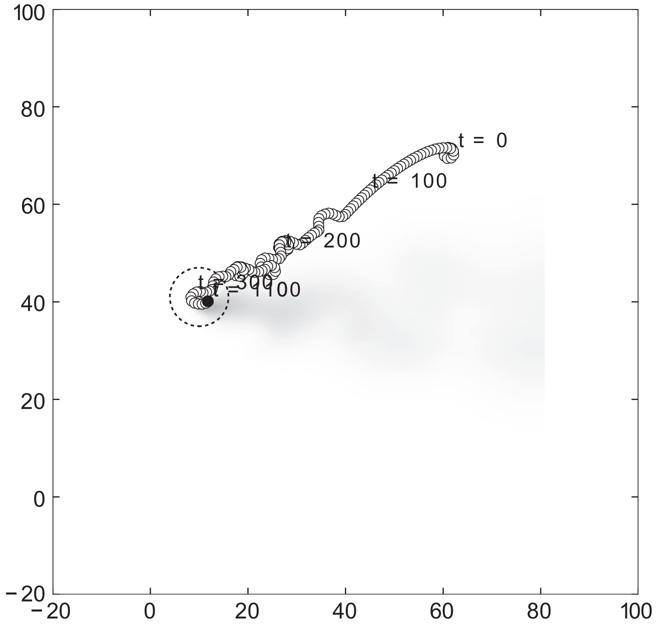


Fig. 6. Typical run of the best evolved individual in the low turbulent scenario.

Since the robot searches by moving to the bottom left on the map, it will find the plume if it starts above and to the right of it, but will miss the plume if it starts below or to the left of the plume.

The right part of Fig. 7 shows whether the individual ends up at the source (white), does not end up at the source but passes close to it at some point of its trajectory (gray), or does not pass close to the source (black), with the top image generated under normal conditions and the bottom image generated for a single plume in an infinite world. In the top image, the white area covers almost the entire area, with the robot locating the source from all but two of the starting locations.

Finally, robot 2's performance is compared with the moth strategy from [42]. The comparison only focuses on the plume following sub-task, so the robot is always initialized within the plume ($x \in [40,\ 60], y \in [25,\ 55]$). Furthermore, since the moth algorithm does not include source identification, only the distance to the source at the end of the trial is evaluated. The parameters for the moth algorithm optimized for the low-turbulent conditions are $v_m = 0.07$, $t_m = 5.01$, $C_m = 8.66$, and $\dot{\psi} = 10.94^\circ/\text{s}$. For the moth algorithm, the average distance of the robot to the source at the end of the trial is 6.9 (± 11.0). Furthermore, $P(d_{t=T} \le 6\ \text{m}) = 0.794$. When robot 2 starts in the plume, it has $d_{t=T} = 1.5(\pm 2.6)$ and $P(d_{t=T} \le 6\ \text{m}) = 0.998$. Because the final distance distributions are highly skewed toward low values and we do not want to make assumptions on their functional form, the significance of the difference in distance distribution is determined with a single-sided bootstrap test [5]). The test shows that robot 2 outperforms the optimized moth strategy on the high-turbulent test scenario with $p < 0.001$.

3.2. Analysis

The main goal of the analysis is to find out what strategy the evolved robots follow to find and recognize the odor source. The six different evolutions on the low turbulence case all resulted in similar strategies, and therefore the analysis focuses on the best performing solution, robot 2. The analysis is subdivided in the three sub-tasks of odor source localization: plume search, plume following, and source identification.

3.2.1. Plume search

The first phase of the odor source localization task is to search the plume. In order to investigate robot 2's searching strategy, it is placed in an environment without odor source and a constant wind going from left to right. $R=10$ runs are then performed, initializing the robot at random locations with random starting headings. For each run, the wind direction with respect to the robot's heading is stored for time steps $t > 200$. Fig. 8 shows a histogram of the wind angles during the search behavior. The mean wind direction is −51.9°, with a standard deviation of 5.0°. The robot follows a fixed heading with the wind when searching for a plume. This means that it always moves obliquely cross-wind. The other evolved controllers on the diffuse case also move with fixed, albeit different, angles to the wind. In an environment with randomly distributed odor plumes of finite extent, perpendicular cross-wind search (± ~90°) leads to the shortest search times [42]. The slight up-wind movement during search in the low turbulent scenario might be due to the initialization during evolution downwind of the source. Another possibility is that the search direction is a compromise between a good search behavior and a good behavior for when the robot 'looses' the plume during the plume following behavior.

3.2.2. Plume following

When the robot has found the plume, it starts moving toward the source. A first question to be answered is whether the robot in doing so follows the concentration gradient. To test this, the robot is run while at each location of its trajectory over time the concentration gradient is determined. The direction up-gradient ψ'_c is then represented with respect to the robot's heading ψ, as $\psi_c = \psi'_c - \psi$. It can vary from −180° to 180°, with a heading of 0° implying that the robot is exactly following the gradient. Concentration gradient measurements are only retained if the robot is moving faster than 0.01 m/s.

Fig. 9 shows a histogram of the gradient direction over $R=30$ runs. The main observation is that the robot does not follow the concentration gradient. Surprisingly, it actually seems to move roughly perpendicular to the gradient, as there is a peak at ~80–110°.

Observation of the robot's behavior for different initial locations leads to the following explanation of its strategy. Since the robot searches downward, it will typically encounter the plume from the top. After finding the plume, the robot attempts to stay roughly perpendicular to the concentration gradient, which means that it follows an isoline of the plume. In particular, the robot seems to turn to the right when the concentration increases and to the left when it decreases. This results in isoline following on the top of the plume. However, if the robot keeps following the same isoline, it will go toward the source, around it, and then back again away from the source. The robot does not do this, and actually turns stronger to the left when it has wind in the back. As a result, the robot does not go away from the source, but steers toward it. As a consequence, the robot spirals in toward the source. The robot seems to use the high concentration close to the source for reducing its velocity and coming to a stop.

The rather fixed reactions to concentration increases and decreases lead to a less efficient strategy when starting in the bottom of the plume. The robot will actually first move to the top

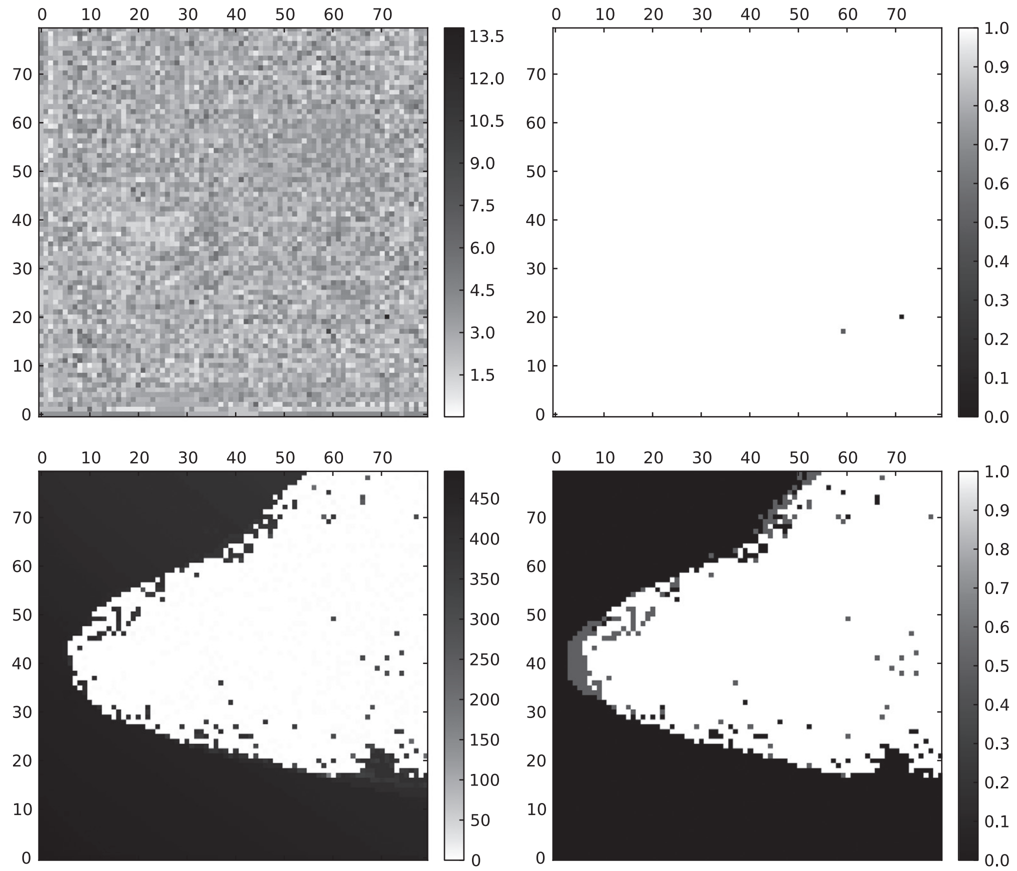


Fig. 7. *Left*: distances to the source at $t = T$ are shown at all initial positions of a 80 × 80 grid in the environment. The top image was generated under normal conditions with the random toroid world and the bottom image was generated with one single plume in an infinite world. In the top image the final distances range from 0 (white) to 13.8 (black). In the bottom image the final distances range from 0 (white) to 484 (black). *Right*: whether the robot ends up at the source (white), does not end up at the source but passes close to it at some point of its trajectory (gray), or does not pass close to the source (black). The top image was generated under normal conditions with the random toroid world and the bottom image was generated with one single plume in an infinite world.

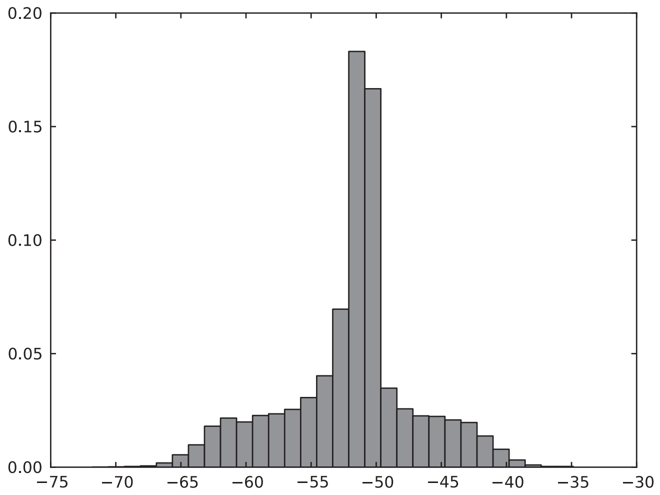


Fig. 8. Wind angles during robot 2's search behavior when it does not sense any odor.

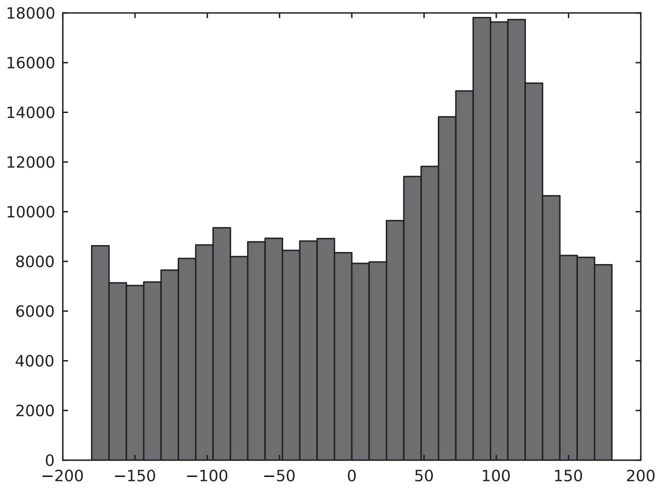


Fig. 9. Robot heading with respect to the concentration gradient. If the robot followed the concentration gradient, it would have a relative heading of 0°.

of the plume and then start following the isoline again.[5] Concentration decreases encountered in the bottom half of the plume will cause the robot to make full left-turning loops, because the leftward turn is generally not countered by a concentration increase.

The explanation above is corroborated by the correlation between sensory inputs and the motor actions. For example, the correlation coefficient[6] r between sensory input s_1 and o_1 at the

[5] The robot moves to the top of the plume without making rightward circles. Although an increasing concentration makes the robot turn more to the right, the wind direction experienced when moving to the top of the plume counteracts this rightward turning.

[6] r is defined as $r(a,b) = E[(a-\overline{a})(b-\overline{b})/\sqrt{\mathrm{var}(a)\mathrm{var}(b)}]$.

same time t is $r(s_1,\ o_1) = 0.23$ while $r(s_1,\ o_2) = -0.52$. This means that if s_1 increases, ω decreases, and the robot turns more to the right. The average concentration sensor s_2 is similarly correlated with o_1 and o_2, with $r(s_2, o_1) = 0.91$ and $r(s_2, o_2) = -0.17$. Wind direction sensor s_3, which has activation 1 when the wind comes from behind, has a stronger positive correlation with o_2 than with o_1: $r(s_3, o_1) = 0.17$ and $r(s_3, o_2) = 0.22$. This implies that wind from the back will lead to an increase of ω, which makes the robot turn more to the left.

The strategy can be nicely illustrated by runs during which the odor concentration is kept static. Fig. 10 shows two different environments, with corresponding concentration isolines. Per environment, two runs are shown, one starting high in the plume and one starting low in the plume. In the completely artificial plume on the top, one can see that the robot moves to and subsequently follows the isoline of concentration level ~20.0. The figure on the bottom is a ‘snap shot’ of one of the simulated low-turbulent scenarios. For both starting conditions, the robot finally ends up at isoline ~20.0, goes around the source, and then steers toward it. Note that in the static environment, the robot circles less frequently, since there are no puffs of odor moving past (with the associated concentration rises and drops). Also note that the run starting low in the plume makes three loops, when it moves down-gradient. This is less likely to happen when it starts high in the plume.

Finally, in order to verify that the above analysis captures the necessary components of a successful odor localization strategy,

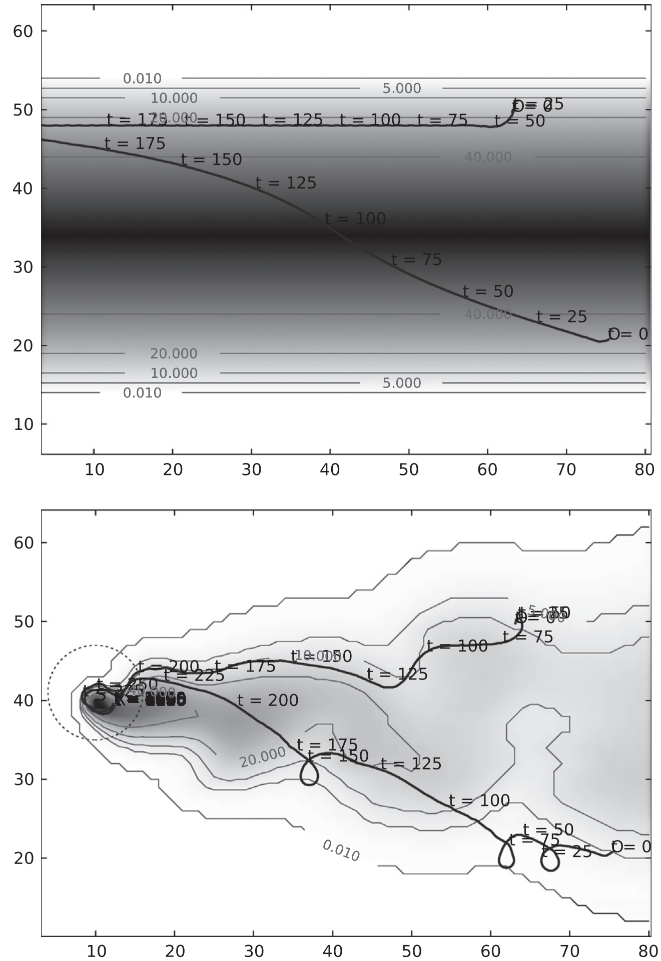


Fig. 10. Runs of robot 2 in environments with a static odor plume. *Top*: two runs in an environment with an artificial plume. *Bottom:* two runs in an environment with a static plume that is a ‘snapshot’ of on of the low turbulent scenarios.

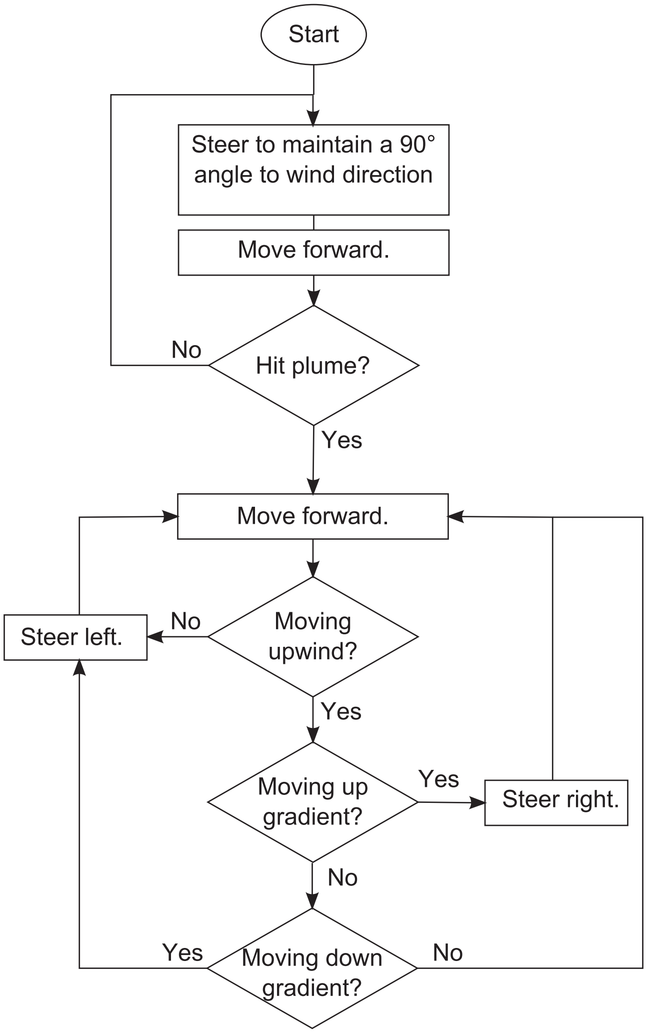


Fig. 11. Algorithm based on evolved behavior of robot in low turbulence scenario.

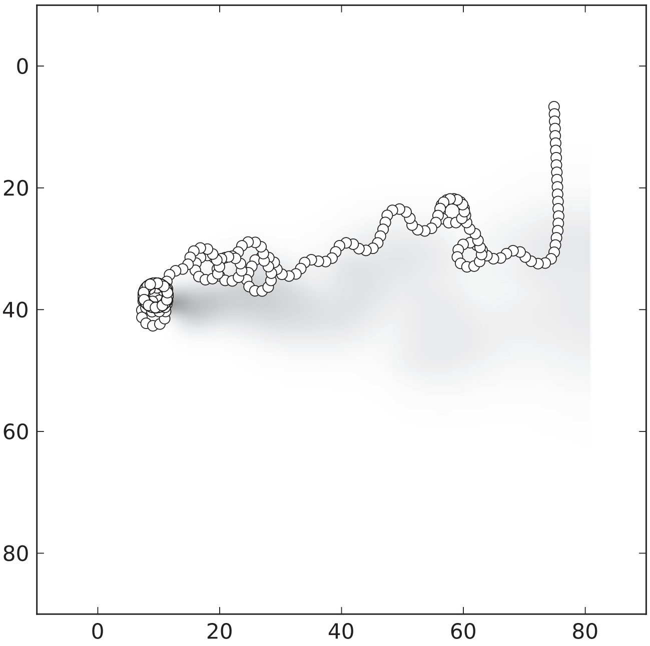


Fig. 12. A typical run of the finite-state-machine based on the behavior evolved for the low turbulence scenario.

a hand-coded version of the strategy was made. Fig. 11 shows a finite-state machine representation of the strategy. An additional advantage of developing such a representation of the behavior is that it allows other researchers to reproduce the strategy. The robot moves across wind until it encounters the odor plume, at which point it begins to exhibit a behavior that allows it to follow an isoline. If the robot is moving up gradient it will steer to the right, and if it is moving down gradient it will steer more severely to the left. By steering harder to the left than the right the algorithm more accurately reproduces the spiralling behavior of the evolved CTRNN. Additionally, if the robot begins to move downwind it will steer to the left until it is facing upwind again. When tested on $R = 1000$ runs, the percentage of the robot ending up within 6 m is 95.3%, closely matching the performance of the evolved CTRNN. An example run of the algorithm is shown in Fig. 12.

3.3. Source identification

When the robot gets close to the source, it slows down. Although it cannot be seen in a plot showing the entire environment, the robot starts circling downward extremely slowly. This behavior can in principle be detected automatically and used for source identification. However, the robot also explicitly identifies the source with its fourth output neuron, o_4. The average mutual information between o_4 and the presence within the circle, Q, has been determined as 0.731 (see Table 1).

In order to understand what a mutual information value entails, Fig. 15 shows the histogram of o_4 outside of the circle (white transparent) and inside of the circle (dark), for a run in which the robot obtains a mutual information value of 0.76. It is clear from the figure that o_4 is low close to the source and high far from the source.

The few high outputs close to the source are due to the initial part of the trajectory in the 6 m circle. To illustrate this, the trajectory of the final part of the run and the corresponding neural activations are shown. Fig. 13 contains the part of the run in which the robot gets close to the source ($t \approx 450$ s), follows the isoline around it ($t \in [480, 500]$), spirals in toward the source with two loops ($t \in [500, 550]$) and finally almost comes to a stop just downwind of the source $t > 550$ s. Fig. 14 is a plot of the involved neural activations of the output neurons (top, complemented with a graph of ω), the hidden neurons (center), and the sensory inputs (bottom).

While the robot is following the isoline and not really close to the source yet ($t \in [450, 500]$), the identification output o_4 still has

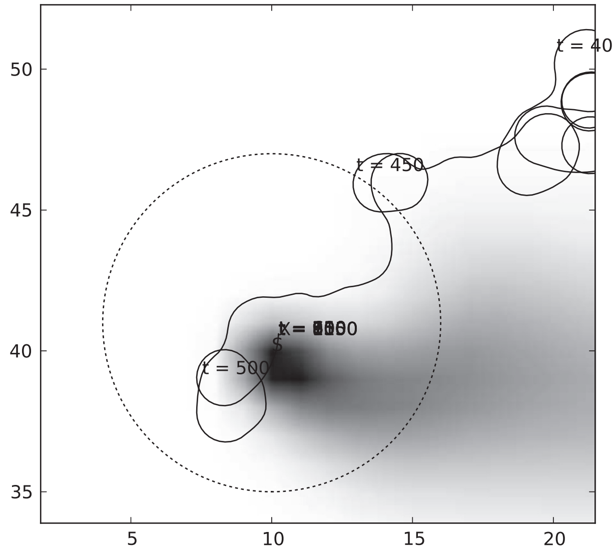


Fig. 13. Trajectory of the robot close to the odor source.

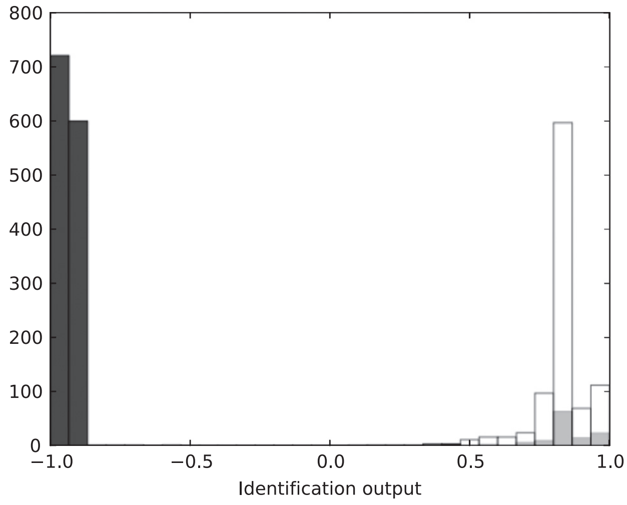


Fig. 15. Histograms of the identification output's activations within 6 m of the source (dark) and at larger distances (transparent white), for a single run.

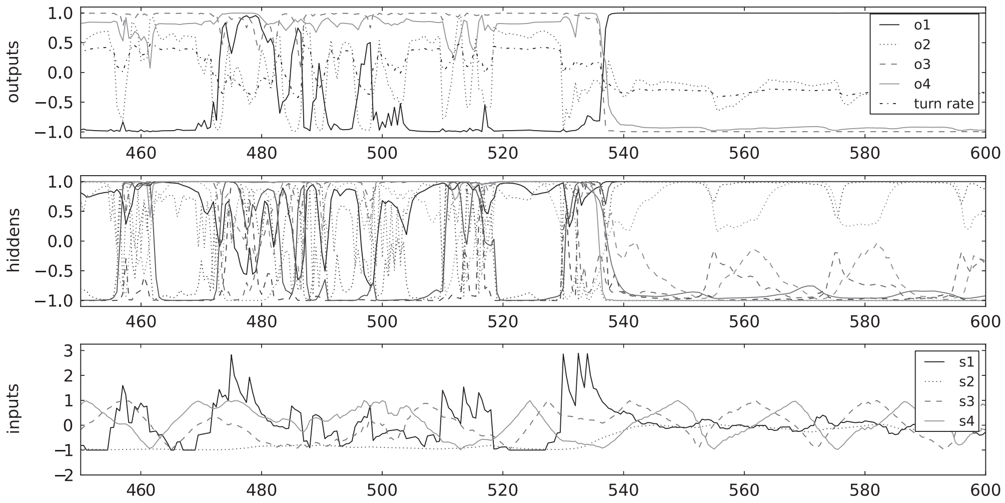


Fig. 14. The top, middle and bottom plots show the activations of the output neurons, hidden neurons, and sensory inputs respectively. The plot of the output neurons is complemented by a graph of the robot's turn rate, ω.

a high activity. Looking at the neural activations, one can notice that s_2 is still low ~−1.0, as is the base activation of s_1. They both increase around $t=537$, while o_4 decreases. To verify whether the higher activities of s_1 and s_2 cause stable source identification, the following test has been devised. The goal of the test is to influence the identification output, while keeping the robot's behavior identical to before. To this end, at each time step first the neural network was run on the normal sensory inputs. Then, a copy of the network was also run, but now with some of the sensory inputs replaced, and the identification output was calculated. In this way, in the copied network the hidden neurons activations and the identification output accumulate the manipulated sensor information over time, without influencing the robot's behavior. Two variants of the tests are performed: (1) a test in which s_1 is set to −1.0 if it is lower than 1.5 (the rationale being that this leaves intact the detection of a concentration increase but does influence the baseline activity), and (2) a test in which s_2 is set to −1.0. Both tests lead to high identification activation values, $o_4 \approx 1.0$. In terms of performance, while in the normal run a mutual information of 0.76 was obtained, in both tests the mutual information drops to 0.00. This means that both neural inputs s_1 and s_2 need to be coherent with sensor readings close to the source. Furthermore, replacing the sensory inputs only in a given time interval showed that any replacement longer than 2.5 s leads to high o_4 activation values.

4. High turbulence scenarios

As mentioned before, the experiments on the high turbulence scenarios have been performed with a concentration threshold $C_{thr}=20$. Fig. 16 shows the effects of this threshold on the area of the plume that can be sensed by the robot: black pixels indicate concentrations above the threshold, white pixels concentrations below the threshold. On average, the robot can sense the odor in 1.99% of the simulation environment.

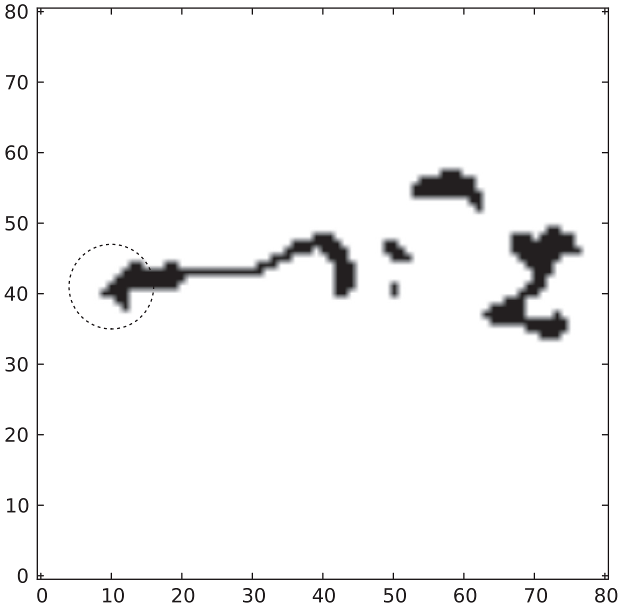


Fig. 16. Effect of a concentration threshold $C_{thr}=20$. Black pixels indicate concentrations above the threshold, white pixels indicate concentrations below the threshold.

4.1. Results

For the high turbulence scenarios, the six evolutionary runs resulted in different amounts of success on the odor source localization task. Table 2 shows the performances of the best individuals on the fourth test scenario, with $R=1000$. The table shows the average fitness $\overline{C}$, the average distance to the source at the end of the run $\overline{d_{t=T}}$, the proportion of times that the robot is close to the source (≤6 m) at the end of the run $P(d_{t=T} \leq 6\ m)$, and the normalized average mutual information $I(O_4;Q)/H(Q)$. All evolutionary runs lead to successful plume search and good plume following. However, only two out of six runs result in good odor source identification (evolutionary run 1 and 6).

Table 2
Performance on the high-turbulence scenario.

	C	$d_{t=T}$	$P(d_{t=T} \leq 6\ m)$	$I(O_4;Q)/H(Q)$
1	−137(± 103)	7.85(± 17.3)	0.843	0.439(± 0.268)
2	−69.0(± 66.1)	10.1(± 19.5)	0.774	0.0(± 0.0)
3	−35.8(± 62.3)	12.3(± 19.1)	0.706	0.057(± 0.079)
4	−1.31(± 60.0)	15.6(± 20.4)	0.407	0.016(± 0.030)
5	−30.8(± 64.6)	13.4(± 22.8)	0.746	0.05(± 0.073)
6	−71.5(± 124)	22.0(± 29.4)	0.523	0.308(± 0.333)

The best result is obtained by evolution number 1, with an average performance of $\overline{C}=-137$. This individual, referred to as robot 1, also scores best on all sub-criteria shown in Table 2. On average the individual ends up at $d=7.85$ m from the source, while being closer than 6 m to it in 84.3% of the cases. Furthermore, it has the highest average $I(O_4;Q)=0.439$.

Fig. 17 shows a typical run of robot 1. The robot starts the run at a location under the plume. It then starts searching obliquely cross-wind toward the bottom left, at a relatively high speed. It leaves the simulation area on the bottom, and reappears at a random (more distant) location on the top. At $t=400$ s, the robot senses a puff of the odor plume and subsequently starts moving (more slowly) upwind to the source. Arrived at the source, the robot first makes a loop around it. Slightly downwind of the source, at $t=1024$ s, the robot starts to move so slowly that it almost stops. The robot identifies the source with a low activation of the identification output (black circle markers).

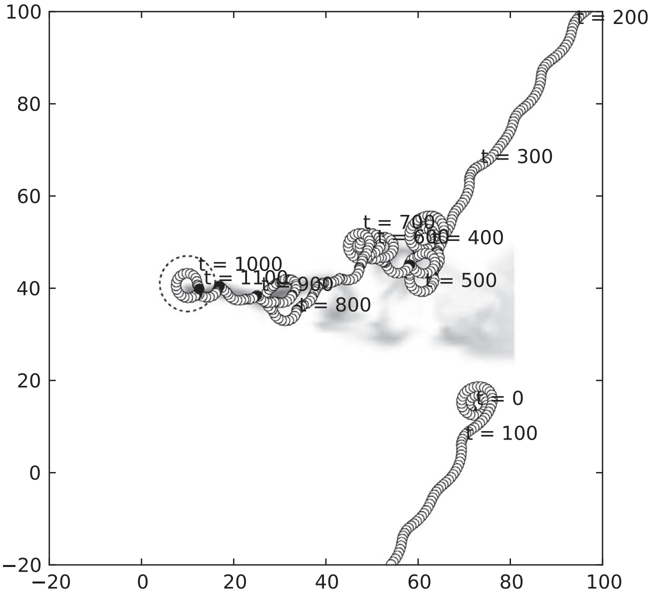


Fig. 17. Typical run of the best evolved individual on the high turbulence scenario.

The generalization of robot 1's strategy to other locations than the ones used in evolution is tested in the same manner as for the

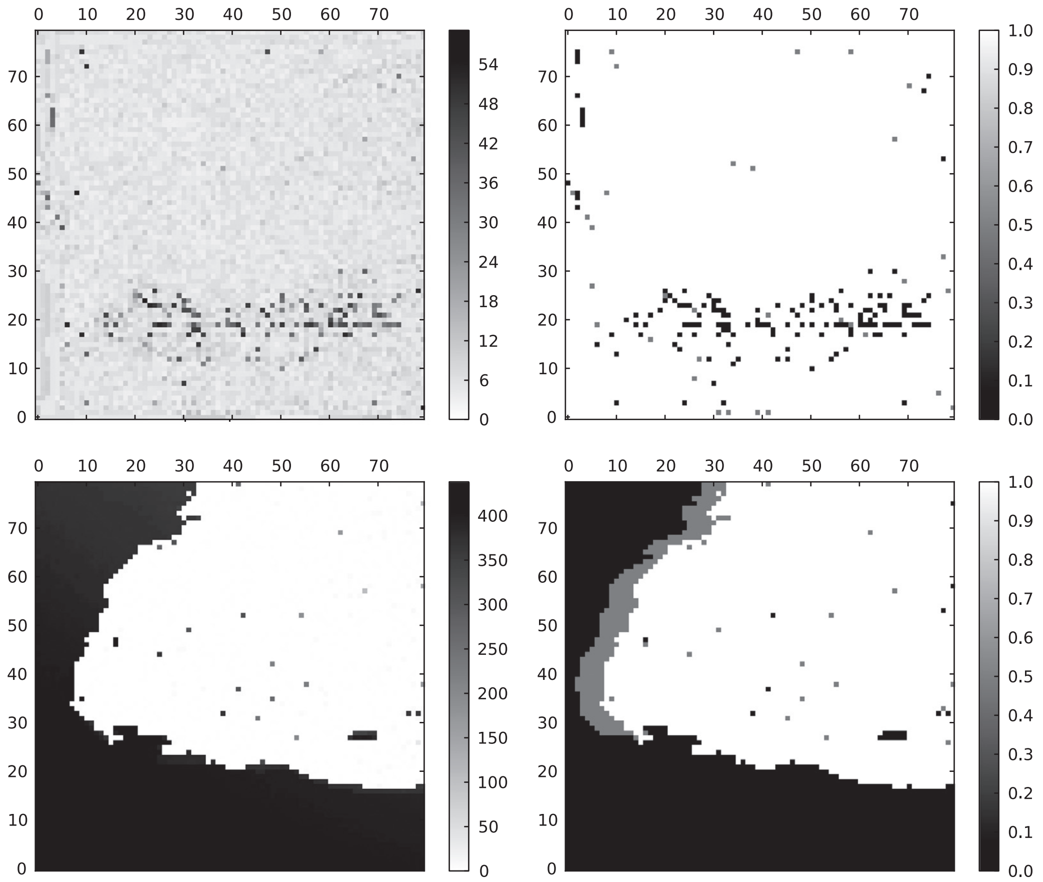


Fig. 18. *Left*: distances to the source at $t = T$ are shown at all initial positions of a 80 × 80 grid in the environment. The final distances range from 0 (white) to 52 (black). *Right*: whether the robot ends up at the source (white), does not end up at the source but passes close to it at some point of its trajectory (gray), or does not pass close to the source (black).

low-turbulent case. The results are shown in Fig. 18. The images are similar to those generated for the low turbulence scenario. However, below the plume is a region of dark 'noise' in the figures located in the left and in the bottom of the figure. Due to the search direction and the location of the plume, these areas lead to the robot not sensing the plume, leaving the simulated area and reappearing at a random location on the right or top, respectively. With a bit of bad luck, the robot misses the plume again, and finally either does not sense any plume at all, or arrives too late at a plume. This also explains why the utter bottom is bright/white again: starting at such a place effectively means that the robot is closer to a plume in its search direction.

The bottom images of Fig. 18 show the performance of the robot starting from every location if there is only one plume in an infinite world. Again, the performance is similar to that of the robot in the low turbulence scenario in that it is successful from all locations within or near to the plume and from an area above the plume. However, the white areas cover a greater portion of the world above the plume than in the low turbulence scenario because the robot evolved under turbulent conditions has a more crosswind search strategy, allowing it to locate the plume from more starting positions.

Also on the high-turbulent case, the performance is compared with the moth strategy from [42]. The parameters for the moth algorithm optimized for the high-turbulent conditions are $v_m = 0.83$, $t_m = 0.07$, $C_m = 20.0$, and $\dot{\psi} = 47.4\ ^{\circ}/\text{s}$. For the moth algorithm, the average distance of the robot to the source at the end of the trial is 13.6(±16.6). Furthermore, $P(d_{t=T} \leq 6\ \text{m}) = 0.574$. When robot 1 starts in the plume, it has $d_{t=T} = 4.3(\pm 12.7)$ and $P(d_{t=T} \leq 6\ \text{m}) = 0.933$. A single-sided bootstrap test [5] shows that robot 1 outperforms the optimized moth strategy on the high-turbulent test scenario with $p < 0.001$.

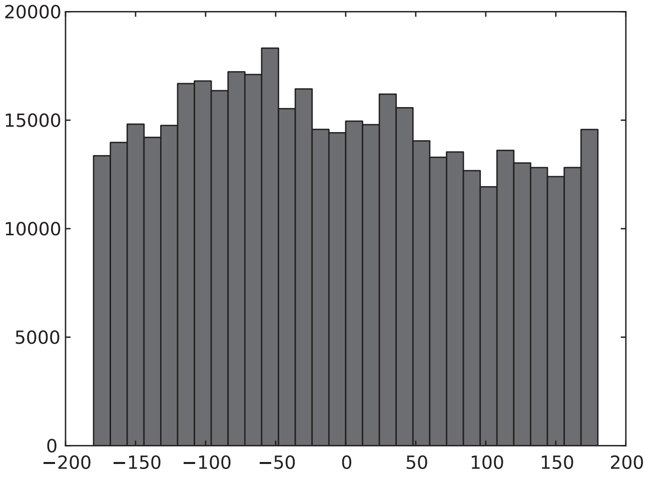


Fig. 19. Robot heading with respect to the concentration gradient. If the robot followed the concentration gradient, it would have a relative heading of 0°.

4.2. Analysis

In this section, the odor source localization strategies are analyzed of robots evolved on the high turbulence scenarios. The six different evolutions all resulted in similar strategies, and therefore the analysis focuses on the best performing solution, robot 1. The analysis is again subdivided in the three sub-tasks of odor source localization: plume search, plume following, and source identification.

4.2.1. Plume search

The same test as for the low-turbulence scenario is performed to find the angles to the wind that the robot follows when there is no odor concentration. The mean direction is −59.0°(±15.7). Again, the robot follows a rather fixed heading with the wind when searching for a plume. It always moves obliquely cross-wind, but oscillates more from left to right than in the diffuse scenario. This is the reason for the larger spread in directions. The oscillating behavior can be seen in Fig. 17.

4.2.2. Plume following

When the robot has found the plume, it starts moving toward the source. Although it is well-known that under highly turbulent conditions, gradient search is not a good strategy, it is interesting to repeat the analysis on the gradient direction with respect to the robot's heading. Fig. 19 shows a histogram of the gradient direction over the entire run. As in the low-turbulent scenario, the robot does not follow the concentration gradient. However, in this case, the histogram is close to uniform. Although the negative gradient directions are slightly more likely, the histogram suggests that the robot is not keeping any fixed heading with respect to the concentration gradient.

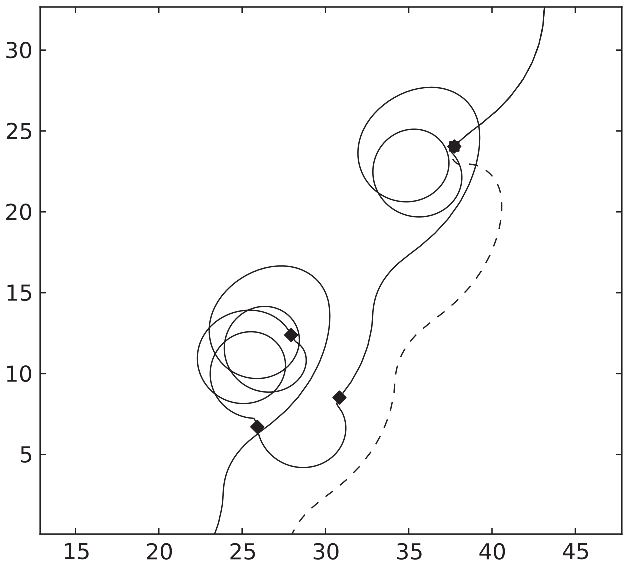


Fig. 20. Behavior of the robot when it encounters four (artificial) puffs at different times, but all lasting 5 s.

Observation of the robot's behavior leads to an alternative hypothesis. When the robot senses the odor, it performs two loops that move up-wind and end closely downwind of the position of the odor detection. If the robot does not sense the odor after some time, it seems to start searching for the plume again. Since there is a reasonable probability that the robot senses the odor again while looping upwind or moving cross-wind, it has a reasonable chance of making a new loop closer to the source. The closer the robot gets to the source, the larger the probability of sensing the odor, and hence of the robot making another loop toward the source. This behavior will finally bring it to a position just down-wind of the source. At that location it will make a full loop, since it will not detect any odor upwind of the source. In addition, it is highly likely to sense the odor downwind of the source again, resulting in a new loop around the source.

In order to investigate the behavior over time of the robot when it hits puffs of odor, the robot is placed in the environment without the odor plume and first run normally for $t=200$ s. Then, in the time interval $t=[200, 205]$ s, the odor concentration measurement is artificially set to a concentration $c=30$, i.e., above the detection threshold of 20. This is repeated for $t\in[350, 355]$ s, $t\in[380, 385]$ s, and $t\in[456, 461]$ s. Fig. 20 shows the behavior of the robot when it hits the artificial puffs of $c=30$. The trajectory is represented with a solid line, while the onset of each puff is indicated with a diamond. When hitting the first puff, the robot first steers slightly to its left ($\omega=0.50$ rad). Then it makes two loops. After $t=325$ s, the normal search behavior is resumed. The angular velocity of the robot during the loops smoothly goes from −0.22 at the onset of the first loop to 0 when exiting the second loop. The 5 s-long detection of a puff has an influence on the behavior for 125 s. This could be due to the neural network dynamics being triggered by the peak of $s_1=c-c_f/c_f$. However, it turns out that this peak only causes a brief 'noise' on the turning rate and velocity: artificially setting the sensory input c_f to zero completely removes the circling behavior of the robot, but leaves intact the initial steering to the left (dashed line in the figure). Note that c_f influences both s_1 and s_2. It is sensory input s_1 that causes the looping: leaving c_f intact, but clamping s_2 to zero has a negligible effect on the looping behavior.

Three aspects of the loop setup are particularly important. First, if the robot hits a puff on the upper side of its circle, the new circle should be located higher than the previous one. If this were not the case, the robot would move downwards through the plume

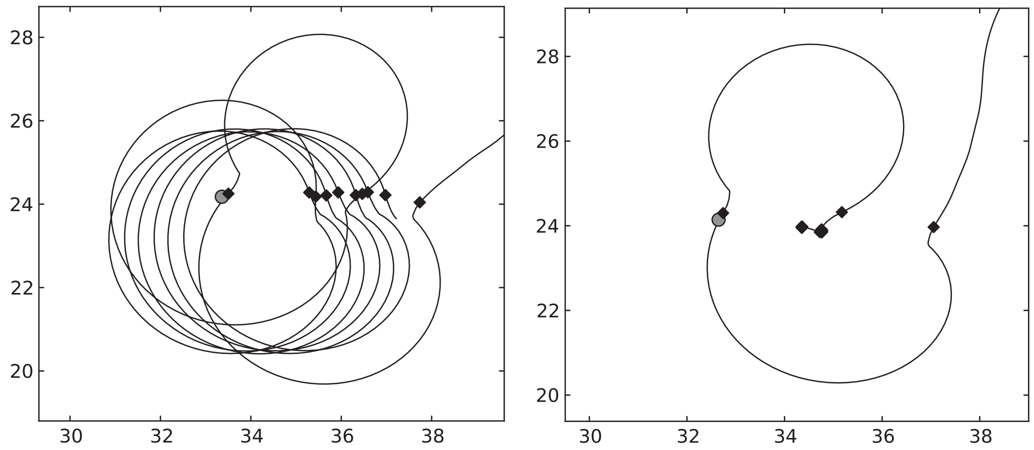


Fig. 21. Behavior of the robot when an artificial source is placed upwind (grey circle marker). If the robot is downwind, it receives artificial puffs lasting 5 s. *Left*: puffs have a concentration of $c=30$. *Right*: puffs have a concentration of $c=75$.

and likely exit it from the bottom before arriving at the source. One can see the robot moving up in Fig. 20, after the third and fourth puff.

Second, hitting the puff close to the original detection location should lead to a new, almost identical loop. The reader can notice in Fig. 20 that for the single puff detection, the detection location is enclosed by the second loop (the location of the fourth puff is even enclosed by both loops). Because of this setup, the robot will keep on circling around the source, if it will hit a puff once every two loops.

Third, not all puff reactions are equal. For example, when the robot receives a new artificial puff while still circling (see bottom part of Fig. 20), the shape of the robot's trajectory is not equal to that of the first puff: one can see that after the fourth puff, the robot moves less strongly to its left (in fact it steers to the left for a shorter duration and subsequently steers stronger to the right). The different responses are caused by the different inputs and network state at subsequent detections (at the fourth puff, the robot is going downwind). Also the strength and duration of the puff have an influence on the resulting behavior.

To illustrate the second and third aspect of the loop setup, another test is performed with artificial puffs. This time, when the robot receives the first puff at $t=200$, an 'artificial source' is placed at the same y-coordinate, but 4.5 m downwind of the robot, i.e., $(x_s, y_s)=(x-4.5, y)$. Every time the robot passes in a band $y \in [y_s-0.3, y_s+0.3]$ and downwind of the source $x > x_s$, the robot receives a new puff of 5 s duration. Fig. 21 shows the resulting behavior for a 'weak' puff ($c=30$, left part of the figure) and a strong puff ($c=75$, right part of the figure). The trajectories are especially interesting, because they show that the robot exploits two different ways of staying close to the source. First, it exploits the source's emission frequency. The frequency should be high enough so that the robot always encounters the plume at least once during its two loops (of which at least one passes behind the initial detection location). Second, it makes use of the higher odor concentration levels just downwind of the source: this makes the robot slow down and hence experience strong odor puffs more frequently. These two ways add robustness to the behavior. For instance, if an odor source releases less strong odor puffs, the robot will still circle around it if the emission frequency is high enough.

In order to test the validity of the hypothesis advanced in the analysis above, a simplified, hand-coded version of the strategy has been devised. It can be represented as a finite state machine, which is shown in Fig. 22. The robot moves across wind until it encounters the odor. Upon encountering the odor plume the robot

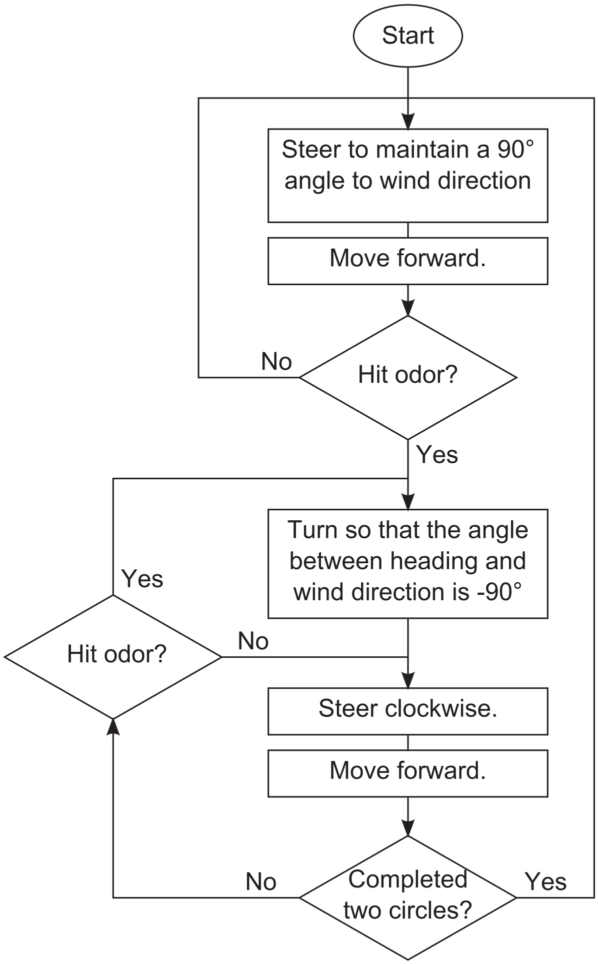

Start
Steer to maintain a 90° angle to wind direction
Move forward.
Hit odor?
No
Yes
Turn so that the angle between heading and wind direction is -90°
Hit odor?
Yes
No
Steer clockwise.
Move forward.
Completed two circles?
No
Yes

Fig. 22. Algorithm based on evolved behavior of robot in high turbulence scenario.

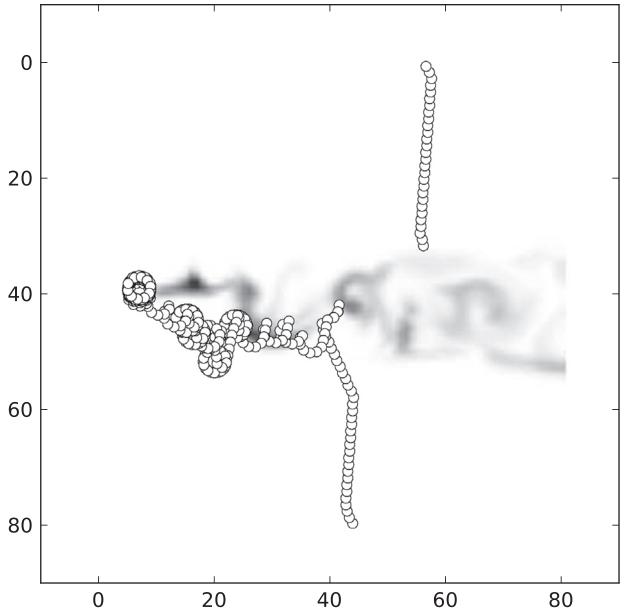


Fig. 23. A typical run of the algorithm based on the behavior evolved for the high turbulence scenario.

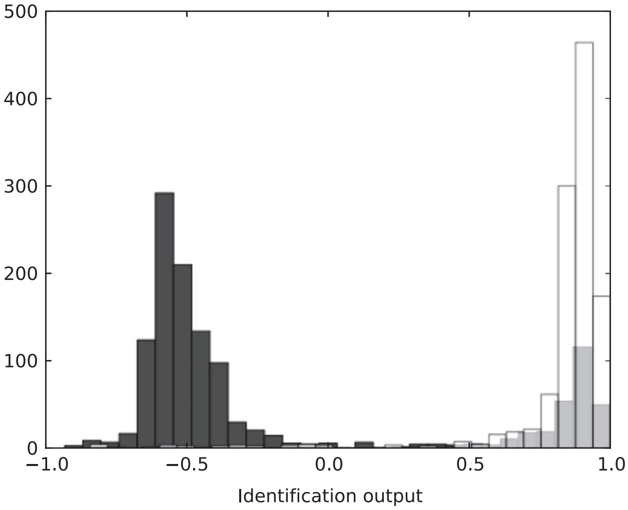


Fig. 24. Histograms of the identification output's activations within 6 m of the source (dark) and at larger distances (transparent white), for a single run.

will turn to head across wind in the opposite direction, before moving in two clockwise circles. If during this time the plume is detected again the circular motion is restarted. If however the plume is not detected during the circling behavior the robot will revert to searching across wind. The hand-coded algorithm abstracts away from some of the aspects of the CTRNN, but it captures the essential properties. After manually tuning the parameters of the algorithm, it can indeed successfully perform the odor source localization task. When tested on $R=1000$ runs, the percentage of the robot ending up within 6 m is 69.9%. This is worse than the evolved agent by 14.4%, but it shows that the algorithm covers at least the basic principles behind the robot's strategy (Fig. 23).

4.2.3. Source identification

The average mutual information between o_4 and the presence within the circle, Q, has been determined as 0.439 (see Table 2). This average is slightly lower than in the low-turbulence scenario for two reasons. First, the average is influenced by the number of times that the robot reaches the source, as the mutual information is zero otherwise. In the high-turbulence scenario, the probability of ending up at the source is 13.1% lower than in the low-turbulence case. Second, the higher turbulence leads to a more difficult identification problem, with the instantaneous concentration level being less indicative of source proximity. To illustrate the source identification performance, Fig. 24 shows the histogram of o_4 outside of the circle (white, transparent) and inside of the circle (dark), for a run in which the robot finds the source.

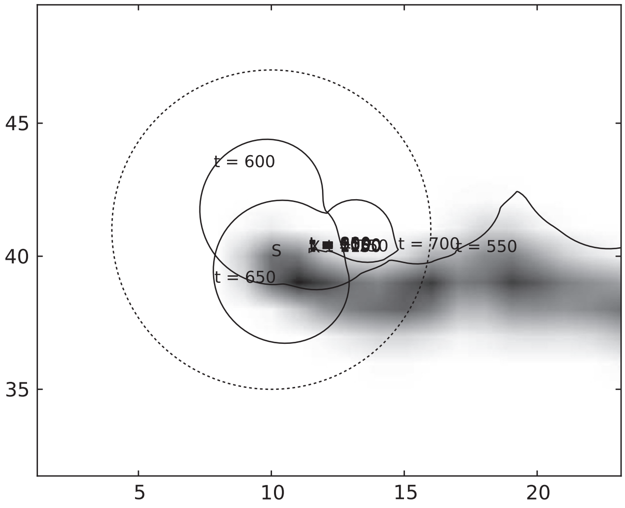


Fig. 25. Trajectory of the robot when it approaches the source (t=560 s), loops around it ($t\in[580,700]$), and finally almost comes to a stop just downwind of the source $t>775$ s.

It is clear from the figure that o_4 generally is low close to the source and high far from the source. However, the robot's output is also sometimes low far away from the source (just after detecting a new odor puff) and high close to the source (when it is upwind of the source for example). Fig. 25 shows the robot's trajectory when it approaches the source ($t=560$ s), loops around it ($t\in[580,\ 700]$), and finally almost comes to a stop just downwind of the source $t>775$ s. Fig. 26 is a plot of the involved neural activations of the output neurons (top, complemented with a graph of ω), the hidden neurons (center), and the sensory inputs (bottom).

Looking at the sensory inputs and neural activations, one can see that o_4 has a lower activity when the robot has a high $s1$. This causes the few faulty identifications far away from the source. However, encountering a single odor puff does not always result in a low o_4 and does not lead to continuous source identification. This only happens when the robot gets really close to the source. Odor puffs are sensed more frequently, and although this leads to smaller peaks in s_1's activity (around 2 instead of 8), it does lead to a higher baseline activity of both s_1 and s_2. To verify that the higher activities of s_1 and s_2 cause stable source identification, the same two tests as for the low turbulent case are performed. While in the normal run a mutual information of 0.46 was obtained, in both tests the mutual information dropped to 0.00. This means that both neural inputs s_1 and s_2 need to be coherent with sensor readings close to the source. Furthermore, replacing the sensory inputs only in a given time interval showed that any replacement longer than 2.5 s leads to high o_4 activation values.

With the same kind of test, a more remarkable observation was made. When looking at the activations in Fig. 26, one can see that s_3 and s_4, the wind direction inputs, are much more variable before $t=700$ s than after. When the robot circles, these inputs go up and down. When the robot slows down close to the source, it moves straight down with the wind coming from its right. The wind direction inputs s_3 and s_4 then are rather constant. To investigate whether this also influences the source identification, the activations of s_3 and s_4 were replaced after $t=700$ with those 500 s before. This led to a significant increase in identification activation, with o_4 ranging in between −0.5 (as in the normal run) and 0.5

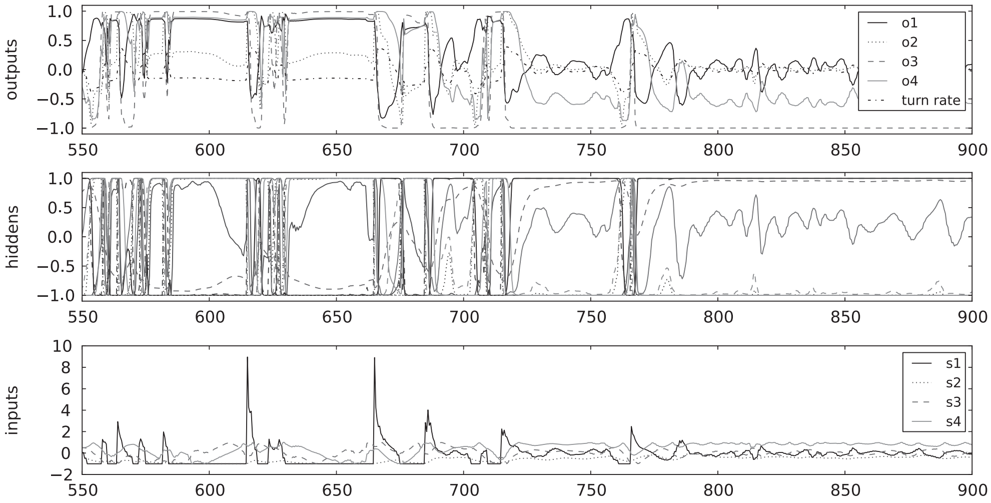


Fig. 26. Neural activations in the same time interval. The top, middle and bottom plots show the activations of the output neurons, hidden neurons, and sensory inputs respectively. The plot of the output neurons is complemented by a graph of the robot's turn rate, ω.

(higher than in the normal run). The mutual information more than halved to 0.19. In the case of the wind direction inputs, already a sensory replacement lasting 1 s has an effect on o_4.

The finding that the wind sensors influence source identification is interesting, because it implies that the robot does not only use environmental cues to identify the source, but also the properties of its own behavior. Namely, in general there is no correlation between wind direction and proximity to the source. However, given the behavior of the robot there is such a correlation.

5. Conclusions

The main conclusion is that the ER approach can be successfully employed to solve the odor source localization task under turbulent conditions. The evolved strategies are both robust and computationally efficient, and significantly outperform the silkworm moth strategy from [42]. Moreover, both the evolution under low-turbulent conditions and the one under high-turbulent conditions resulted in novel odor source localization strategies, using only one chemical sensor and one wind direction sensor. The analysis revealed the underlying characteristics of the strategies.

The robot evolved for the low-turbulent scenarios approximately follows an isoline of the concentration gradient until it starts to move downwind when passing the source. Having the wind in the back makes the robot spiral in toward the source. This strategy exploits the fact that in the low-turbulence scenario, diffusion plays a considerable role in the cross-wind direction. While the concentration in the center of the plume has a high variance because of the passing odor puffs, the border of the plume has a rather low variance. This implies that the lower concentration isolines generally lead toward the source without interruption.

The robot evolved for the high-turbulent scenarios makes two upwind loops upon the detection of an odor puff. The loops pass close to (and behind) of the original detection location. As a consequence, the robot will keep looping around the source. This strategy exploits the spatial probability distribution for sensing an odor puff, and in particular the fact that this probability is practically zero up-wind of the source, while it is considerable just downwind of the source.

The strategies as revealed by the analysis have been validated by means of hand-coded finite-state-machines. Finally, in both scenarios, source identification is performed on the basis of the frequency of odor puffs and the higher odor concentration. In the high-turbulence scenarios, the wind direction also aids identification. Although the wind direction in itself does not carry any information on source proximity, it does given the behavior of the robot: close to the source the robot starts moving downward extremely slowly, while far from the source it goes downward more rarely.

References

[1] R.D. Beer, A dynamical systems perspective on agent–environment interaction, Artif. Intell. 72 (1995) 173–215.

[2] R.D. Beer, The dynamics of active categorical perception in an evolved model agent, Adaptive Behav. 11 (4) (2003) 209–243.

[3] R.D. Beer, J.C. Gallagher, Evolving dynamical neural networks for adaptive behavior, Adaptive Behav. 1 (1) (1992) 91–122.

[4] J.H. Belanger, M.A. Willis, Biologically-inspired search algorithms for locating unseen odor sources, in: ISIC/CIRA/ISAS Joint Conference, 1998, pp. 265–270.

[5] P. Cohen, Empirical Methods for Artificial Intelligence, MIT Press, Cambridge, MA, 1995.

[6] G.C.H.E. de Croon, S. Nolfi, E.O. Postma, Towards pro-active embodied agents: on the importance of neural mechanisms suitable to process time information, in: D. Braha, A. Minai, Y. Bar-Yam (Eds.), Complex Engineered Systems, Springer-Verlag, New York, NY, 2006, pp. 338–363.

[7] G.C.H.E. de Croon, M.F. van Dartel, E.O. Postma, Evolutionary learning outperforms reinforcement learning on non-Markovian tasks, in: Learning and Memory Workshop of the European Conference on Artificial Life (ECAL 2005), Canterbury, UK, 2005.

[8] J.L. Elman, Finding structure in time, Cognit. Sci. 14 (2) (1990) 179–211.

[9] J. Farrell, W. Li, S. Pang, Chemical plume tracing experimental results with a Remus AUV, in: OCEANS, vol. 2, 2003, pp. 962–968.

[10] G. Ferri, E. Caselli, V. Mattoli, A. Mondini, B. Mazzolai, P. Dario, Spiral a novel biologically inspired algorithm for gas/odor source localization in an indoor environment with no strong airflow, Robot. Auton. Syst. 57 (2009) 393–402.

[11] D. Floreano, P. Dürr, C. Mattiussi, Neuroevolution: from architectures to learning, Evol. Intell. 1 (2008) 47–62.

[12] V. Formisano, S. Atreya, Therese Encrenaz, Nikolai Ignatiev, M. Giuranna, Detection of methane in the atmosphere of mars, Science 306 (5702) (2004) 1758–1761.

[13] K. Funahashi, Y. Nakamura, Approximation of dynamical systems by continuous time recurrent neural networks, Neural Networks 6 (6) (1993) 801–806.

[14] D.J. Harvey, T.-F. Lu, M.A. Keller, Comparing insect-inspired chemical plume tracking algorithms using a mobile robot, IEEE Trans. Robot. 24 (2) (2008) 307–317.

[15] A.T. Hayes, A. Martinoli, R.M. Goodman, Distributed odor source localization, IEEE Sensors 2 (2002) 260–271.

[16] O. Holland, C. Melhuish, Some adaptive movements of animats with single symmetrical sensors, in: From Animals to Animats, Proceedings of the 4th International Conference on Simulation of Adaptive Behaviour, vol. 4, 1996, pp. 55–64.

[17] H. Ishida, Y. Wada, H. Matsukura, Chemical sensing in robotic applications: a review, IEEE Sensors J. 12 (11) (2012).

[18] E.J. Izquierdo, T. Buhrmann, Analysis of a dynamical recurrent neural network evolved for two qualitatively different tasks: walking and chemotaxis, in: 11th International Conference on Artificial Life, 2008, pp. 257–264.

[19] E.J. Izquierdo, S.R. Lockery, Evolution and analysis of minimal neural circuits for klinotaxis in Caenorhabditis elegans, J. Neurosci. 30 (2010) 12908–12917.

[20] D. Izzo, M. Rucinski, F. Biscani, The generalized island model, in: Parallel Architectures and Bioinspired Algorithms, Studies in Computational Intelligence, vol. 415/2012, 2012, pp. 151–169.

[21] R.S. Freedman, K. Zahnle, D.C. Catling, Is there methane on mars? Icarus 212 (2) (2011) 493–503.

[22] G. Kowadlo, R.A. Russell, Robot odor localization: a taxonomy and survey, Int. J. Robot. Res. IJRR 27 (2008) 869–894.

[23] V.A. Krasnopolsky, J.P. Maillard, T.C. Owen, Detection of methane in the martian atmosphere: evidence for life? Icarus 172 (2) (2004) 537–547.

[24] Y. Kuwana, I. Shimoyama, Y. Sayama, H. Miura, Synthesis of pheromone-oriented emergent behavior of a silkworm moth, in: 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 96), vol. 3, 1996, pp. 1722–1729.

[25] Y. Kuwana, I. Shimoyana, Pheromone-guided mobile robot that behaves like a silkworm moth with living antennae as pheromone sensors, Int. J. Robot. Res. 17 (9) (1998) 924–933.

[26] F. Lefevre, F. Forget, Observed variations of methane on mars unexplained by known atmospheric chemistry and physics, Nature 460 (2009) 720–723.

[27] T. Lochmatter, A. Martinoli, Theoretical analysis of three bio-inspired plume tracking algorithms, in: IEEE Conference on Robotics and Automation (ICRA), 2009, pp. 2661–2668.

[28] T. Lochmatter, X. Raemy, L. Matthey, S. Indra, A. Martinoli, A comparison of casting and spiraling algorithms for odor source localization, in: IEEE Conference on Robotics and Automation (ICRA), 2008, pp. 1138–1143.

[29] L.L. López, V. Vouloutsi, A.E. Chimeno, E. Marcos, S. Bermúdez i Badia, Z. Mathews, P.F.M.J. Verschure, A. Ziyatdinov, A. Perera i Lluna, Moth-like chemo-source localization and classification on an indoor autonomous robot, in: L.Pramatarova (Ed.), On Biomimetics, InTech, 2011.

[30] I. Macinnes, E.A. Di Paolo, The advantages of evolving perceptual cues, Adaptive Behav. 14 (2) (2006) 147–156.

[31] L. Marques, A. De Almeida, Electronic nose-based odour source localization, in: Proceedings of the 6th International Workshop on Advanced Motion Control, Nagoya, Japan, 2000, pp. 36–40.

[32] L. Marques, U. Nunes, A. de Almeida, Olfaction-based mobile robot navigation, Thin Solid Films 418 (1) (2002) 51–58.

[33] E.M. Moraud, E. Chicca, Toward neuromorphic odor tracking: perspectives for space exploration, Acta Futura 4 (2011) 9–19.

[34] E.M. Moraud, D. Martinez, Effectiveness and robustness of robot infotaxis for searching in dilute conditions, Front. Neurorobot. 4 (2010) 1–8.

[35] M. Mumma, G. Villanueva, R. Novak, T. Hewagama, B. Bonev, M. DiSanti, A. Mandell, M. Smith, Strong release of methane on mars in northern summer 2003, Science 323 (2009) 1041–1045.

[36] C. Nicol, A.A. Ellery, E. Cloutis, B. Lynch, G.C.H.E. de Croon. A martian methane plume modeling scheme for testing robotic search algorithms, submitted for publication.

[37] S. Nolfi, Power and the limits of reactive agents, Neurocomputing 42 (1–4) (2002) 119–145.

[38] S. Nolfi, D. Floreano, Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines, MIT Press/Bradford Books, Cambridge, MA, 2000.

[39] S. Pang, J.A. Farrell, Chemical plume source localization, IEEE Trans. Syst. Man Cybern. 36 (5) (2006) 1068–1080.

[40] P.A. Rhodes, T.O. Anderson, Evolving a neural olfactorimotor system in virtual and real olfactory environments, Front. NeuroEng. 5 (2012) 1–14.

[41] R.A. Russell, D. Thiel, R. Deveza, A. Mackay-Sim. A robotic system to locate hazardous chemical leaks, in: IEEE Conference on Robotics and Automation (ICRA), 1995, pp. 556–561.

[42] R.A. Russell, A. Bab-Hadiashar, R.L. Shepherd, G.G. Wallace, A comparison of reactive robot chemotaxis algorithms, Robot. Auton. Syst. 45 (2003) 83–97.

[43] C.E. Shannon, A mathematical theory of communication, Bell Syst. Tech. J. 27 (1948) 379–423 623–656.

[44] R.S. Sutton, A.G. Barto, Reinforcement Learning: an Introduction, MIT Press, A Bradford Book, Cambridge, MA, 1998.

[45] E. Tuci, G. Massera, S. Nolfi, Active categorical perception of object shapes in a simulated anthropomorphic robotic arm, Trans. Evol. Comput. J. 14 (6) (2010) 885–899.

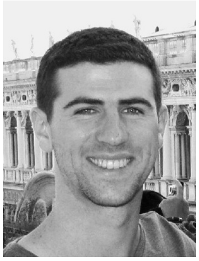
[46] M. Vergassola, E. Villermaux, B.I. Shraiman, 'Infotaxis' as a strategy for searching without gradients, Nature 445 (2007) 406–409.

[47] C. Wang, L. Yin, L. Zhang, D. Xiang, R. Gao, Metal oxide gas sensors: sensitivity and influencing factors, Sensors 10 (3) (2010) 2088–2106.

[48] S. Whiteson, Evolutionary Computation for Reinforcement Learning, Springer326–355.

[49] X. Yao, Evolving artificial neural networks, Proc. IEEE 87 (9) (1999) 1423–1447.

G.C.H.E. de Croon received his M.Sc. and Ph.D. in the field of Artificial Intelligence at Maastricht University, the Netherlands. His research interest lies with computationally efficient algorithms for robot autonomy, with a particular focus on evolutionary robotics and computer vision. He recently worked as a Research Fellow in Artificial Intelligence at the European Space Agency. Currently, he is an assistant-professor at the Micro Air Vehicle lab of Delft University of Technology.



L.M. O'Connor obtained an M.Eng. in Mechanical Engineering from the University of Cambridge in 2009, and an M.Sc. in Artificial Intelligence from Imperial College London in 2011. He has recently completed a year as a Young Graduate Trainee at the European Space Agency where he carried out research as part of the Advanced Concepts Team. His research interests include neural networks, evolutionary robotics and swarm intelligence.

C. Nicol received his B.Eng. in Aerospace Engineering concentrating on Space Systems Design from Carleton University in 2011. His numerous research interests include autonomous science for planetary rovers including intelligent systems for target identification, and planetary penetrators as a lowcost solution for space exploration. He is currently working on his Ph.D. in the field of Astrobiology and Autonomous Robotics at Carleton University as a part of the Canadian Astrobiology Training Program.

D. Izzo has a background in aerospace engineering (M.Sc. at the Sapienza in Rome and at Cranfield University, UK). After obtaining his Ph.D. and serving as an assistant-professor at "la Sapienza" in Rome, he moved to the European Space Agency in 2004. His current research focuses on global optimization, swarm intelligence for space systems, interplanetary trajectories optimization, massively parallel computing, guidance and navigation, and optimal control theory.