Reinforcement Learning for Bio-Inspired Target Seeking

James Gillespie[1], Iñaki Rañó[1], Nazmul Siddique[1], José Santos[1], and Mehdi Khamassi[2]

[1] Intelligent Systems Research Centre, Ulster University, Northern Ireland, UK
gillespie-j10@email.ulster.ac.uk
{i.rano, nh.siddique, ja.santos}@ulster.ac.uk
[2] Institute des Sistèmes Intelligents et de Robotique, University Pierre et Marie Curie, Paris, France
mehdi.khamassi@isir.upmc.fr

Abstract. Because animals are extremely effective at moving in their natural environments they represent an excellent model to implement robust robotic movement and navigation. Braitenberg vehicles are bio-inspired models of animal navigation widely used in robotics. Tuning the parameters of these vehicles to generate appropriate behaviour can be challenging and time consuming. In this paper we present a Reinforcement Learning methodology to learn the sensori-motor connection of Braitenberg vehicle 3a, a biological model of source seeking. We present simulations of different stimuli and reward functions to illustrate the feasibility of this approach.

Keywords: Braitenberg Vehicles, Reinforcement Learning, Source seeking.

1 Introduction

Everyday experience shows that animals are extremely effective at moving in their natural environments which makes them an excellent model to implement robust robotic movement and navigation [4]. Bio-inspiration has played an important role in robotics for the design of locomotive systems [11], locomotion control [8], and control of steering [20]. This paper focuses on learning animal-like steering behaviour using Braitenberg vehicles, a well known model of insect tropotaxis [3], i.e. movement of insects towards or away from a stimulus [5]. While previous works used experimentally tuned models or relied on time consuming evolutionary strategies [2, 12, 17], this work investigates how robots can learn these tropotaxis controllers from experience using Reinforcement Learning techniques.

How animals navigate in the presence of stimuli has been a subject of research for several decades [5, 7], yet the mechanisms generating these behaviours are not fully understood even for very simple animals. One natural mechanism to control movement is tropotaxis, which relies on sampling the stimulus in the

environment at two symmetrically placed sensing organs (eyes, ears...), and can be exploited to control robot movement. The values of the stimulus are used by the animal to steer or control its navigation, via some neural wiring between the sensing organs and the motor effectors. The concrete internal wiring defines the behaviour of the animal generating positive or negative taxis (movement towards or away from a stimulus), as captured in the seminal work of Valentino Braitenberg [3], where taxes are modelled through simple vehicles.

Braitenberg vehicles have been used in robotics for several decades from their most basic formulation [10], to extensions with fuzzy controllers [21] or neural networks [6]. Therefore, they are widely used to implement bio-inspired robotic behaviours, especially when the motion relies on unconventional sensors, for example sensors which are not providing distance readings [9,10,18]. Early research in the field aiming at enabling learning of animal-like movement relied on techniques like evolution strategies [17] and genetic algorithms [12]. These works obtained neural architectures of Braitenberg vehicles to navigate environments with no collisions through an iterative approach to optimise a fitness functions of the distance to obstacles and forward speed. A big drawback of these strategies is the amount of time necessary for the algorithms to find the optimal weights of the neural network to control the velocity of the wheels. Moreover, having such a neural controller makes the analysis of the sensori-motor connection extremely difficult. Another early approach to bio-inspired robotic steering used the Dynamical Systems Approach to behaviour generation for obstacle avoidance combined with a Braitenberg vehicle for target seeking [2]. Although the work presented is highly effective for robot navigation, the parameters of the control mechanism are selected experimentally, which means they might not generate the best possible behaviour.

Interestingly, recent works using Braitenberg vehicles have also developed models which do not use learning techniques to achieve their target seeking behaviour and can be sub-optimal. For instance, a basic implementation of Braitenberg vehicles to determine the speed of the wheels of a smelling robot [10] was used with a simple dynamical normalisation of the measured values of the sensors. All these works rely on experimental results, and while mathematical models of Braitenberg vehicles have been presented in [14,18], it can be challenging to tune the internal parameters of the vehicle, i.e. the controller, as formal proofs of stability are missing. Moreover, an outstanding open question in Braitenberg vehicles is how to define the relationship between sensors and motors in an optimal way. The main contribution of this paper is presenting a reinforcement learning based methodology to perform learning in bio-inspired steering, as a way to answer the question of optimal Braitenberg vehicle design. Obtaining an optimal controller depends on the selected definition of optimality, but also on the stimulus the vehicle perceives. We present results for two common families of stimuli, namely following the inverse square law, and the inverse distance law. As we will see, known theoretical results of Braitenberg vehicles help analysing and interpreting the results from the reinforcement learning process. The rest of the paper is organised as follows. Section 2 presents a brief introduction to the

model of Braitenberg vehicle 3a, and states the problem of tuning the sensorimotor connection as a reinforcement learning problem. Section 3 presents the experimental results obtained for the simulation of these vehicles using the types of stimuli mentioned and several reward functions defining optimality criteria. The paper ends presenting some conslusions and future work in section 4.

2 Reinforcement Learning in Braitenberg vehicles

Braitenberg vehicles are well known models of animal behaviour, used in robotics as a simple way of implementing avoidance and target seeking behaviours. The lack of a mathematical formalism for the vehicles did not hamper their usage but made their implementation a trial and error process, where intuition and experience played a big role. The development of a mathematical model of the closed-loop system of Braitenberg vehicle 3a [13] as a non-linear differential equation allowed using dynamical systems theory to analyse characteristics of the solution trajectories. Oscillatory [15] and unstable [13] trajectories can be found among the solutions of this non-linear dynamical system. Moreover, their formalisation and the model equations allow to get a deeper understanding of how these controllers work, and enable making better informed guesses on the parameters without the need for experimentation. However, as we already stated, one outstanding open question is finding under which conditions the behaviour of Braitenberg vehicle 3a is stable close to a given equilibrium point, since the linear stability test provides no information [15], while finding a Lyapunov function is a non-trivial problem. Therefore, our goal with this work is learning a stable non-linear controller that achieves target seeking for a Braitenberg vehicle 3a.

Figure 1 shows Braitenberg vehicle 3a in the proximity of a light source (although to derive its mathematical model the stimulus can be of any nature and does not need to come from a source). The wheels of the vehicle are used to abstract the locomotive systems of animals focusing on the steering level of motion [1]. In fact, similar steering models have been used to understand human motion. As shown in the figure, the vehicle has two sensors connected to the ipsilateral wheel (the same side wheel) in a decreasing (inhibitory) way, i.e. in the sense the higher the stimulus value, the slower the wheel turns. Intuitively the vehicle turns towards, and approaches, the light source, performing a 'hill-climbing' on the stimulus. The closer the vehicle gets to the source, or maximum, the lower its velocity, and it will eventually stop near the peak when the speed of both wheels is zero. Because intuitively the motion converges to the maximum, these vehicles were used in real robots and simulated artificial agents without mathematical formalisation of their behaviour. These works assumed the vehicle converged regardless of the stimulus provided, if the connection between the sensors and the wheels was appropriately tuned.

We will present the steps to model the vehicle shown in figure 1, with a wheelbase d and distance between the sensors δ. Assuming the vehicle is immersed in a stimulus that does not change over time, the stimulus itself can be modelled as a non negative function $S(\mathbf{x})$ of the position in the environment or the ve-

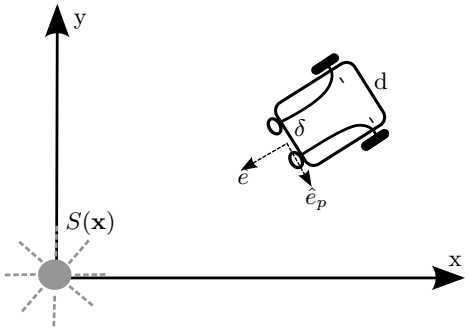


Fig. 1. Braitenberg vehicle 3a.

hicle workspace $\mathbf{x} = (x, y) \in \mathcal{W} \subseteq R^2$. The stimulus is considered to simply be a position dependant scalar function, like light or sound intensity, for instance. Furthermore, we will assume $S(\mathbf{x})$ is of type C^∞ in $\mathcal{W}$, i.e. its value changes smoothly in the environment. If the stimulus is originated by a point-like source located at $\mathbf{x}_S \in \mathcal{W}$ the function $S(\mathbf{x})$ will reach a maximum at this point as the stimulus intensity decays with distance, i.e. $S(\mathbf{x}_S) \geq S(\mathbf{x}) \forall \mathbf{x} \in \mathcal{W}$, which also implies $\nabla S(\mathbf{x}_S) = \mathbf{0}$ and $\mathbf{y}^T D^2 S(\mathbf{x}_S)\mathbf{y} < 0 \forall \mathbf{y} \in R^2$ because the function is smooth, i.e. the gradient of $S(\mathbf{x})$ vanishes and the Hessian is a negative definite matrix at $\mathbf{x}_S$. Without loss of generality, as figure 1 shows, we can assume the source is located at the origin of a reference system, i.e. $\mathbf{x}_S = \mathbf{0}$, and the position of the vehicle is referred to that coordinate system.

According to the qualitative model proposed by Braitenberg, the connection between the sensor readings 's' and the left and right wheel velocities $v_{L/R}$ is direct (ipsilateral) and inhibitory. We will model this connection as a smooth function $F(s)$, fulfilling the following criteria: (i) the motion of the vehicle is never backwards, i.e. $F : \Re^+ \cup \{0\} \rightarrow \Re^+ \cup \{0\}$; (ii) the vehicle stops at the source, i.e. $F(S(\mathbf{0})) = 0$; and (iii) an inhibitory connection implies that the derivative of $F(s)$ is negative for all stimulus values s, i.e. $F'(s) < 0$. In the literature of Braitenberg vehicles for source seeking the function is selected to be an experimentally tuned linear function, or an Artificial Neural Network, however it can be any function fulfilling these conditions. Under the above conditions, the behaviour of the vehicle can be approximated as:

$$\begin{aligned} \dot{x} &= F(S(\mathbf{x}))\cos\theta \\ \dot{y} &= F(S(\mathbf{x}))\sin\theta \\ \dot{\theta} &= -\frac{\delta}{d}\nabla_{\mathbf{x}} F(S(\mathbf{x})) \cdot \hat{e}_p \end{aligned} \tag{1}$$

where $\mathbf{x} = (x, y)$ is the midpoint between the sensors, $\nabla_{\mathbf{x}} F(S(\mathbf{x}))$ is the gradient of the function composition, '$\cdot$' represents the dot product, and $\hat{e}_p = [-\sin\theta, \cos\theta]$ is a unit vector perpendicular to the vehicle direction. While condition (ii) forces the dynamical system to have an equilibrium point, condition

(*iii*) is related to the stability of the equilibrium point. Therefore the functions sought using Reinforcement Learning should fulfil these conditions for the vehicle to perform positive taxis, i.e. it should have negative slope and it should be zero when the vehicle is at the source.

2.1 Simulations of Vehicle stimulus and connection

The mathematical model introduced above is the tool to obtain analytic results on the behaviour of the vehicle, like stability analysis, features of the trajectories, et cetera. However, because its application in a reinforcement learning set-up would require computing the derivative of the function to learn $F(s)$, we opted for a simpler statement of the learning problem. Hence, in our simulations we use the straightforward implementation of vehicle 3a, i.e. evaluating the stimulus at two points and computing the speeds for the left and right wheels. Given the position of the right and left sensors $\mathbf{x}_r$ and $\mathbf{x}_l$, respectively, we can compute the velocities for the right and left wheel as $v_r = F(S(\mathbf{x}_r))$ and $v_l = F(S(\mathbf{x}_l))$, where $F(s)$ is the sought function and $S(\mathbf{x})$ is a selected stimulus function. The left and right sensor positions can be easily computed from the vehicle pose, i.e. the midpoint between the sensors, $\mathbf{x}$ and the orientation of the vehicle, θ, as $\mathbf{x}_r = \mathbf{x} - \frac{\delta}{2}\hat{e}_p$ and $\mathbf{x}_l = \mathbf{x} + \frac{\delta}{2}\hat{e}_p$, where the $\hat{e}_p = [-\sin\theta, \cos\theta]$ is obtained from the vehicle orientation θ. The velocities of the wheels can be converted into forward speed v and turning rate ω of the vehicle as:

$$\begin{aligned} v &= \frac{F(S(\mathbf{x}_r)) + F(S(\mathbf{x}_l))}{2} \\ \omega &= \frac{F(S(\mathbf{x}_r)) - F(S(\mathbf{x}_l))}{d} \end{aligned} \tag{2}$$

where it is worth remembering that the positions of the sensors depend on the vehicle pose, i.e. $\mathbf{x}_r = \mathbf{x}_r(\mathbf{x}, \theta)$ and $\mathbf{x}_l = \mathbf{x}_l(\mathbf{x}, \theta)$, and, therefore, the movement of the vehicle depends on its pose ($\mathbf{x}$, θ) through the stimulus function and the sensori-motor connecting function $F(s)$. Given these velocities, the trajectory of the vehicle can be obtained as the solution of the system of non-linear differential equations:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(\mathbf{x}, \theta) \\ \omega(\mathbf{x}, \theta) \end{bmatrix} \tag{3}$$

which has to be integrated numerically, and in our experiments we used the Euler method with a fixed time step of 0.05. It is worth remembering that the connection function $F(s)$ must be decreasing for the vehicle to be of type 3a, otherwise instead of target seeking the vehicle will move away from the source (ipsilateral increasing connection corresponds to vehicle 3b, negative taxis). However, as we will see in the next section this constraint cannot be imposed in the learning process, but good solutions are found using appropriate reward functions.

2.2 The reinforcement learning problem

The problem of learning the target seeking behaviour for Braitenberg vehicle 3a given a stimulus function can be casted into finding a non-increasing function of the stimulus to compute the velocities. Given the stimulus $S(\mathbf{x})$ defined in some environment $\mathcal{W} \subseteq \Re^2$ and an initial pose of the vehicle $(x_0, \theta_0) \in \mathcal{W} \times S^1$ the trajectory followed depends on the connection function $F(s)$. Since the trajectory unfolds in time $(\mathbf{x}(t), \theta(t))$ but also depends on the initial pose and $F(s)$ we will write $(\mathbf{x}(t, \mathbf{x}_0, \theta_0, F), \theta(t, \mathbf{x}_0, \theta_0, F))$. For each pose of the vehicle we can define a scalar reward function $r(\mathbf{x}, \theta)$ measuring how good being at that state is, and because the trajectory depends on the initial pose and the connecting function, the reward along a given trajectory is therefore a function of the initial pose and the connecting function too, that is $r(\mathbf{x}, \theta) = r(t, \mathbf{x}_0, \theta_0, F)$. Theoretically, the function $F(s)$ can be approximated using any type of function approximation, however, for this work we opted to use a Radial Basis Function neural network of the form:

$$F(s) = \sum_{i=1}^{N} w_i \phi_i(s) \tag{4}$$

where N is the number of basis functions, w_i is the weight of the i-th neuron, and the radial basis functions $\phi_i(s)$ were Gaussian kernel functions centred at fixed equidistant positions s_i within the range of the stimulus (from $s = 0$ to $s = S(\mathbf{0})$). If we denote the weight vector of the network by $\Phi = (w_i)$, the reinforcement learning problem can be stated as maximising the following total return:

$$R[\Phi] = \int_{(x_0,\theta_0)\in\mathcal{W}\times S^1} dx_0 d\theta_0 \int_0^\infty r(t, \mathbf{x}_0, \theta_0, \Phi) dt \tag{5}$$

where we need to integrate for all the initial conditions, i.e. the points in the workspace $\mathbf{x}_0$ and all orientations in the unit circle $\theta_0 \in S^1$, but we also need to integrate over the whole trajectory. Because this total return function depends on the solution of a non-linear dynamical system it is impossible to evaluate these integrals. Moreover, integrating over the whole trajectory is obviously not feasible, so we defined a finite time t_f to run the simulations, changing the upper integration limit in equation (5). This does not solve the problem of evaluating the return, however we can use the sampling trick to estimate through simulations the value of the integral by randomly sampling the space of poses of the vehicle. We can use roll-outs from random initial poses to estimate the gradient $\nabla_\Phi R[\Phi]$ and perform a hill climbing on the return to find the optimal weight vector of the RBF network using:

$$\Phi_{k+1} = \Phi_k + \alpha_k \nabla_\Phi R[\Phi_k] \tag{6}$$

where $a_k = \frac{a_0}{k}$, a_0 is the initial learning rate, and, as already stated, the gradient was estimated using roll-outs through small random perturbations of the parameters Φ using central finite differences.

Although in our case the environment of the vehicle was the whole plane $\mathcal{W} = \Re^2$, to simplify things further the domain of the integral in equation (5), i.e. the domain in which the initial conditions are selected for the sampling process, was defined to be a square region around the origin. Moreover, the initial angular directions of the vehicle were selected to be pointing towards the source within a $\pm 90^\circ$ range, i.e. $\theta_0 \in [\theta_t - \pi/2, \theta_t + \pi/2]$, where $\theta_t = \arctan\left[\frac{y_0}{x_0}\right] - \pi$.

3 Experimental results

To learn the sensori-motor connection for a Braitenberg vehicle using Reinforcement Learning we defined and tested several reward functions. Because the goal of the vehicle is to reach the source of a stimulus, i.e. where it takes its highest value, an immediate candidate for the reward function would be the stimulus itself. Provided the vehicle has access to its pose, and knowing that the source is at the origin of the reference system, an alternative reward would be a function of the distance from the vehicle to the source and its relative heading. We simulated four different reward functions, two dependent on the stimulus and two on the pose of the vehicle, and to fulfil assumption (i) in section 2 the reward was given only when the movement of the vehicle was in the forward direction. The reward functions used are:

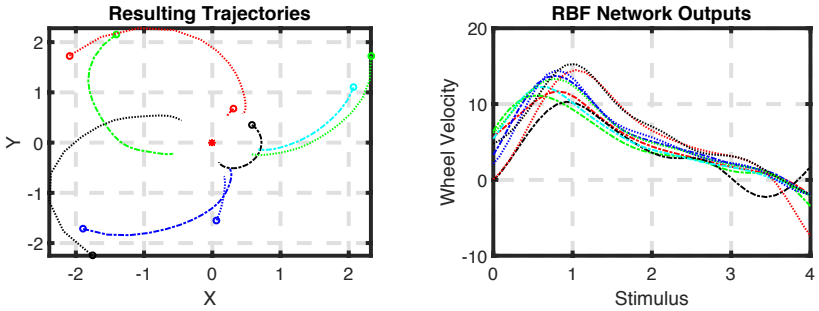
1. The stimulus itself, and because the vehicle obtains readings from both sensors, we use as reward their sum, i.e. $r = S_L + S_R$.
2. The previous reward tries to maximise the stimulus value, but for a stimulus source it is important that the vehicle heads in the right direction. We defined a second stimulus based reward that accounts for the heading direction by trying to make the value in both sensors identical. The reward function accounts for the value but penalises directions perpendicular to the source $r = \frac{S_L + S_R}{1+(S_L - S_R)^2}$.
3. In the simulations we have access to the pose of the vehicle, $(\mathbf{x}, \theta)$, which allows us to define additional reward functions. Using the pose this reward function consists of a linear combination of terms accounting for the proximity of the vehicle to the source and its heading. The selected function was $r = \frac{a}{1+\alpha(\theta,\theta_T)^2} + \frac{1}{1+||\mathbf{x}||^2}$, where $a = 3$ represents the relative importance of heading vs. the distance, $||\mathbf{x}||$ is the distance to the source, and $\alpha(\theta, \theta_T)$ is the angular distance in the range $[-\pi; \pi]$ between the robot heading (θ) and the desired heading $\theta_t = \arctan\left[\frac{y}{x}\right] - \pi$.
4. Our early experiments with the reward functions defined above showed the vehicle's trajectories stopped way before reaching the source. We thought it was due to the limited/bounded rate of growth of these reward functions close to the source. Therefore, we included as a reward function $r = \frac{1}{||\mathbf{x}||^2}$, but since it is singular at $\mathbf{x} = \mathbf{0}$, we defined an threshold distance ϵ such that if $||\mathbf{x}|| < \epsilon$, $r = \frac{1}{\epsilon^2}$.

These four reward functions were used for the two general types of stimulus defined below. It is worth noting that in all the simulations the robot initial

heading was randomly selected within $\pm\pi/2$ radians in the direction of the stimulus.

3.1 Inverse-square law stimulus

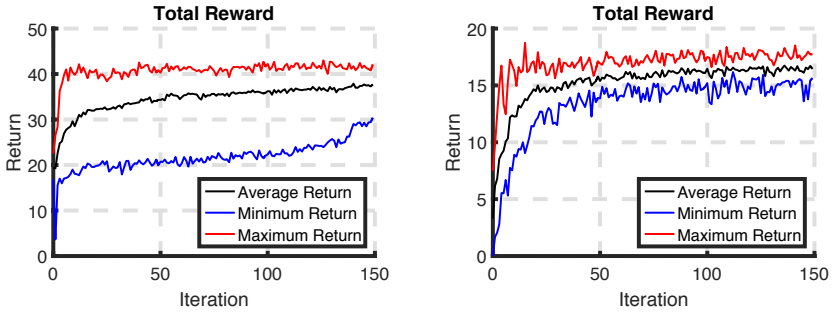
Probably the best known example of Braitenberg vehicles is the one implementing phototaxis using light sensors. With a light source placed at some height h_0 above the origin of a reference systems of a ground plane, and since light intensity follows the inverse-square law, the stimulus as a function of the position $\mathbf{x}$ will be $S(\mathbf{x}) = \frac{I_0}{h_0^2+||\mathbf{x}||^2}$, where I_0 is the intensity of the light at the source. This function can be rewritten as $S(\mathbf{x}) = \frac{g_0}{1+\eta||\mathbf{x}||^2}$ where $\eta = \frac{1}{h_0^2}$ and $g_0 = \frac{I_0}{h_0^2}$. We selected in all the light-like simulations this last functional form and used $g_0 = 4$ and $\eta = 0.25$.



(a) Simulated Trajectory (b) RBF Network Outputs

Fig. 2. Results from phototaxis simulations using reward function 1.

In our first set of simulations we used the stimulus sum reward function, and the results are shown in Figure 2. We ran 10 learning experiments per reward function with random initialisations of an RBF network. Only results of the successful simulations are shown in the figures: Figure 2 depicts simulated trajectories on the left and the corresponding $F(s)$ functions approximated by the RBF networks while Figure 3 shows an example of the evolution of the reward as a function of time. The successful simulations are those which fulfil the conditions stated above, i.e. taking positive values (the vehicle moves forward), having a negative slope (the behaviour is positive taxis), and they become zero close to the stimulus maximum (in our case $S(\mathbf{0}) = 4$). The trajectories are shown in the $x-y$ coordinates, with a red star at the origin (where the source is). While Figure 2 shows the vehicle successfully learns to reach the stimulus source in 9 out of the 10 trials, the degree to which it approaches the source changes across trials varies. All of the successful simulations stop relatively close to the source. Looking at the figure showing the RBF Network outputs we can see where the



(a) Evolution of stimulus based reward (b) Evolution of pose based reward

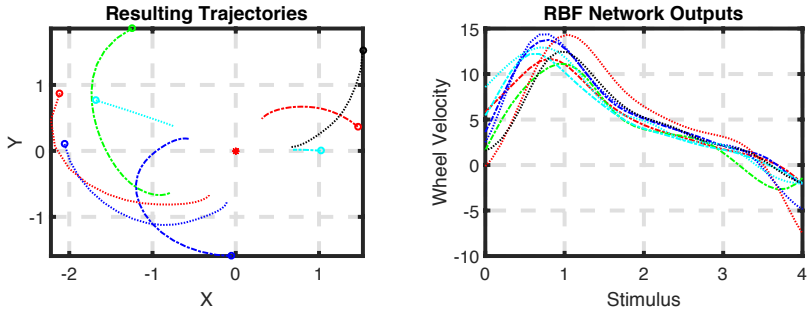
Fig. 3. On the left, an example from the phototaxis simulations of the evolution of a stimulus based reward (using reward function 1), and on the right, an example from the phonotaxis simulations of the evolution of a pose based reward (using reward function 3).

movement of the vehicle converges to, by looking at the zero crossings of the function. For instance the RBF network corresponding to the black dashed graph makes the vehicle stop when the stimulus $S(\mathbf{x})$ reaches a stimulus value of 3, and $F(s)$ is negative for larger values of the stimulus, which does not completely fulfil the conditions imposed for the sensori-motor connection function. The RBF network corresponding to the cyan graph, on the other hand, makes the vehicle stop when the stimulus $S(\mathbf{x})$ reaches 3.5, which is much closer to the source. On the other hand, as the figure on the right shows, most of the RBF networks approximate functions with a negative slope, and it is positive for points not experienced during the training. In general, the attractor of the Braitenberg vehicle on the workspace can be obtained by solving $S(\mathbf{x}) = s_0$, where s_0 is the value at the zero crossing of the RBF network.

In our second set of simulations we used the stimulus sum reward function which penalises perpendicular directions, results of which are shown in Figure 4. For this experiment 8 out of the 10 trials successfully located the stimulus, again with varying degrees of success. However, it should be noted that still most of the RBF networks generate attractors around the maximum, not at the maximum.

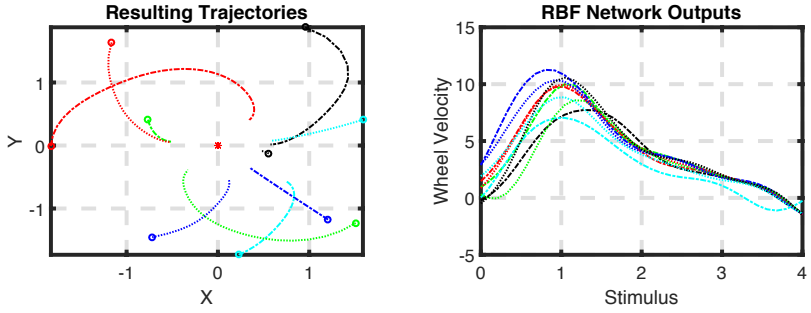
In our third set of simulations we used a pose based reward, consisting of the proximity of the vehicle to the source and its heading. Figure 5 shows the results of this simulation. As can be seen from the RBF network, the functions that have been approximated are more accurate, allowing the robot to get much closer to the source before stopping, i.e. before the attractor set of the dynamical system. Using the pose based reward function proved to be advantageous as shown by this set of simulations, with a 100% learning success rate across the 10 trajectories.

In our fourth and final set of simulations we used an altered pose based reward to try to get the vehicle to reach the stimulus source, detailed in Figure 6. As can be seen from the RBF network output, this is the first experiment whereby



(a) Simulated Trajectory (b) RBF Network Outputs

Fig. 4. Results from phototaxis simulations using reward function 2.



(a) Simulated Trajectory (b) RBF Network Outputs

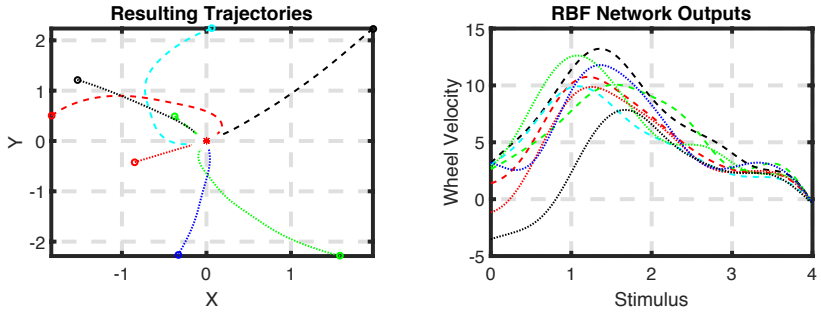
Fig. 5. Results from phototaxis simulations using reward function 3.

the RBF networks make the vehicle reach the maximum, i.e. the attractor is at the source. The results of this can be observed with the successful trajectories plotted, where each vehicle easily finds it's way to the stimulus source regardless of where it starts and gets very close to it in the simulated time.

3.2 Inverse distance stimulus

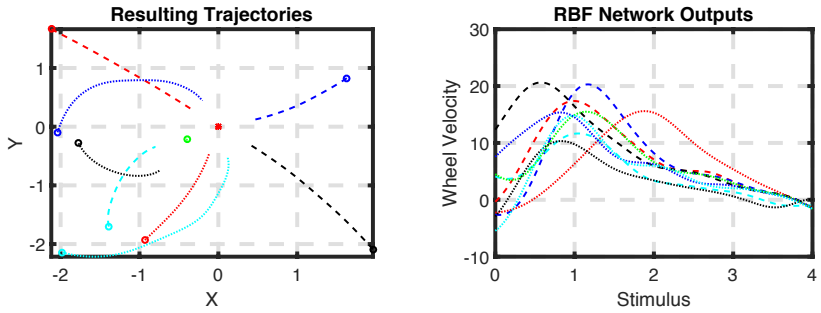
Another common example of Braitenberg vehicles is the one implementing phonotaxis through microphones [19], for instance with a sound source placed at some height h_0 above the ground. It can be seen that, according to the inverse-distance law, the sound intensity will fulfil the conditions for the stimulus function. Furthermore, if the emission pattern of the sound source is isotropic, the stimulus $S(\mathbf{x})$ will be such that $S(\mathbf{x}) \propto \frac{1}{\sqrt{h_0^2+x^2+y^2}}$.

This section presents the results obtained for a stimulus source following the inverse distance law, i.e a stimulus of the form $S(\mathbf{x}) = \frac{g_0}{\sqrt{1+\eta||\mathbf{x}||^2}}$. We selected



(a) Simulated Trajectory (b) RBF Network Outputs

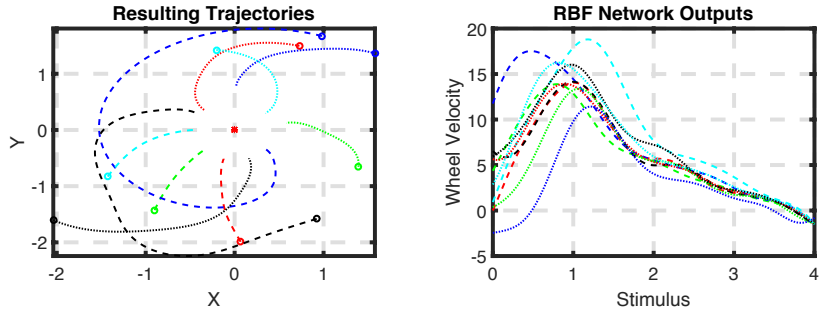
Fig. 6. Results from phototaxis simulations using reward function 4.



(a) Simulated Trajectory (b) RBF Network Outputs

Fig. 7. Results from phonotaxis simulations using reward function 1.

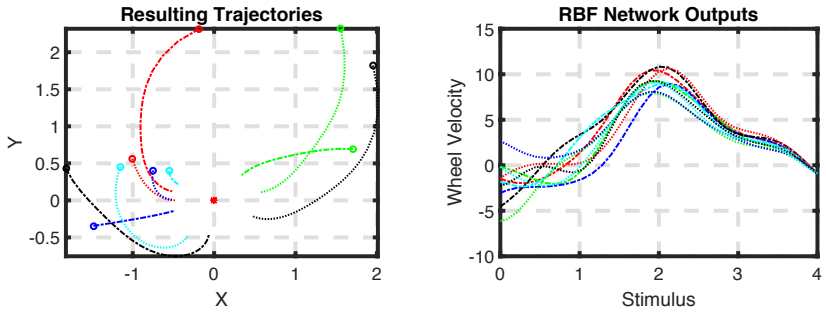
in all the sound-like simulations this last functional form and used $g_0 = 4$ and $\eta = 0.25$. The same experiments were repeated to test whether changing the stimulus affects the performance of the learned controller, i.e. whether some stimulus might be more difficult to follow. As can be seen from the figures below, generally the performance was similar in these sets of simulations compared to the ones above using the inverse-square law stimulus. Like in the phototaxis experiments the RBF networks approximate a function which crosses the zero velocity before reaching the maximum stimulus, meaning the vehicles stop before they reach the stimulus source. The stimulus sum reward function (Figure 7) has the same number of successful trials while the stimulus sum reward function which penalises perpendicular directions (Figure 8) has 10 successful trials, 2 more than the phototaxis counterpart. This proves that, when a sound-like stimulus is used, including a penalty for perpendicular directions helps the Reinforcement Learning algorithm.



(a) Simulated Trajectory (b) RBF Network Outputs

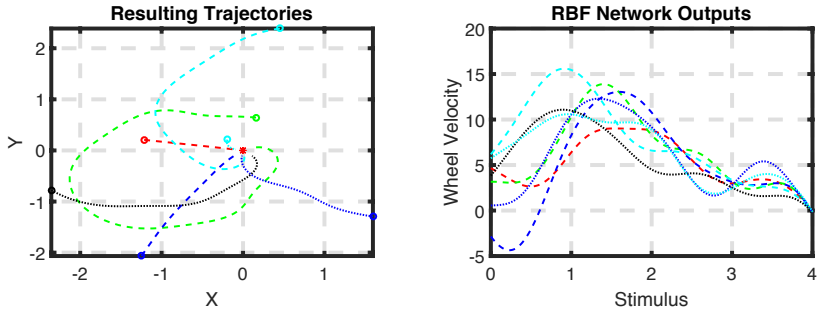
Fig. 8. Results from phonotaxis simulations using reward function 2.

Interestingly, although the learning functions display a shape different from the previous results (and seem to have a smaller basin of attraction) the connection function $F(s)$ for reward function 3 and 4 generate a point attractor at the stimulus source. Figure 9 shows 10 successful trials using the first pose based reward, based on the proximity of the vehicle to the source and its heading (see Figure 3 for the evolution of this reward), while Figure 10 shows 7 successful trials of the altered pose reward function. Although Figure 10 has less successful trials, the function it approximates is clearly more accurate as the vehicle trajectory is closer to a shortest path trajectory compared to the results that can be seen in Figure 9.



(a) Simulated Trajectory (b) RBF Network Outputs

Fig. 9. Results from phonotaxis simulations using reward function 3.



(a) Simulated Trajectory (b) RBF Network Outputs

Fig. 10. Results from phonotaxis simulations using reward function 4.

4 Conclusions and Future Work

This paper presents the first application of Reinforcement Learning to Braitenberg vehicles, a model of insect navigation. Previous robotic works implementing this taxis strategy used hand tuned parameters, or evolutionary strategies [2, 12, 17] to optimise some cost function dependent on the parameters of a neural controller. We used a reinforcement learning framework to adjust the weights of a radial basis function network to control the vehicle. The presented approach has two main advantages over existing techniques. First, it allows us to define an objective function instead of relying on the perception of the resulting behaviour by the developer. Second, the optimal solution is reached several orders of magnitude faster than the evolutionary approach. On the other hand, the analytic model of Braitenberg vehicles allows to draw conclusions about the stability of the learnt RBF controller.

The results obtained here are limited to computer simulations of the vehicles. In the future we plan to test this framework in real robots performing different types of target reaching behaviours. As an intermediate step, however, we plan to introduce noise in the simulated sensors, i.e. add noise to the stimulus function. This will turn the motion equation of the vehicle into a stochastic differential equation, and theoretical results on deterministic systems might not be applicable. Some early results on the behaviour of vehicle 3a under noise conditions exist [16] that can help interpreting the results of (or simplifying) the learning problem.

Acknowledgements

This work was partially supported by the Royal Society International Exchange Scheme under grant IE151293.

References

1. Arechavaleta, G., Laumond, J.P., Hicheur, H., Berthoz, A.: An optimality principle governing human walking. IEEE Transactions on Robotics 24(1), 5–14 (2008)
2. Bicho, E., Schöner, G.: The dynamic approach to autonomous robotics demonstrated on a low-level vehicle platform. Robotics and Autonomous Systems 21, 23–35 (1997)
3. Braitenberg, V.: Vehicles. Experiments in synthetic psycology. The MIT Press (1984)
4. Floreano, D., Ijspeert, A.J., Schaal, S.: Robotics and neuroscience. Current Biology 24 (2014)
5. Fraenkel, G.S., Gunn, D.L.: The orientation of animals. Kineses, taxes and compass reactions. Dover publications (1961)
6. French, R., Cañamero, L.: Introducing neuromodulation to a braitenberg vehicle. In: Proceedings of the 2005 IEEE International Conference on Robotics and Automation. pp. 4188–4193 (2005)
7. Gallistel, C.: Navigation: Whence our sense of direction? Current Biology 27(3), 108–110 (2017)
8. Ijspeert, A., Crespi, A., Ryczko, D., Cabelguen, J.: From swimming to walking with a salamander robot driven by a spinal cord model. Science 315(5817), 1416–1420 (2007)
9. Lebastard, V., Boyer, F., Chevallereau, C., Servagent, N.: Underwater electronavigation in the dark. In: Proceedings of the International Conference on Robotics and Automation (ICRA). pp. 1155–1160 (2012)
10. Lilienthal, A.J., Duckett, T.: Experimental analysis of smelling braitenberg vehicles. In: Proceedings of the IEEE International Conference on Advanced Robotics (ICAR 2003). IEEE (2003)
11. Menciassi, A., Dario, P.: Bio-inspired solutions for locomotion in the gastrointestinal tract: background and perspectives. The Royal Society - Biologically inspired robots (2003)
12. Mondada, F., Floreano, D.: Evolution of neural control structures: some experiments on mobile robots. Robotics and Autonomous Systems pp. 183–195 (1995)
13. Rañó, I.: A steering taxis model and the qualitative analysis of its trajectories. Adaptive Behavior 17(3), 197–211 (2009)
14. Rañó, I.: A model and formal analysis of braitenberg vehicles 2 and 3. International Conference on Robotics and Automation, IEEE (2012)
15. Rañó, I.: Results on the convergence of braitenberg vehicle 3a. Artificial Life 20(2), 223–235 (2014)
16. Rañó, I., Wong-Lin, K., Khamassi, M.: A drift diffusion model of biological source seeking for mobile robots. In: Proceedings of the IEEE International Conference on Robotics and Automation (2017)
17. Salomon, R.: Evolving and optimising braitenberg vehicles by means of evolution strategies. International Journal of Smart Engineering Systems Design pp. 1–13 (1999)
18. Salumäe, T., Rañó, I., Akanyeti, O., Kruusmaa, M.: Against the flow: A braitenberg controller for a fish robot. International Conference on Robotics and Automation, IEEE (2012)
19. Shaikh, D., Hallam, J., Christensen-Dalsgaard, J.: From ear to there: a review of biorobotic models of auditory processing in lizards. Biological Cybernetics 110(4), 303–317 (2016)

20. Webb, B.: A Spiking Neuron Controller for Robot Phonotaxis, pp. 3–20. The MIT/AAAI Press (2001)
21. Yang, X., Patel, R., Moallem, M.: A fuzzy-braitenberg navigation strategy for differential drive mobile robots. Journal of Intelligent Robotic Systems 47, 101–124 (2006)