An Artificial Life Perspective on Olfactory Systems: Evolving Neural Coding, Developmental Symmetry and Odour Recognition in Agents

Nicolas Yvan Oros

April 8, 2010

A thesis submitted in partial fulfilment of the requirements of the University of Hertfordshire for the degree of Doctor of Philosophy

This research was carried out in the department of Computer Science, University of Hertfordshire

Abstract

This thesis addresses the problem of creating simulated agents controlled by neural networks that share features with biological olfactory systems. This work draws from the fields of Artificial Life, Artificial Intelligence and Neuroscience.

The techniques used in this work included simulated agents and chemicals situated in a 2D environment, spiking neural controllers in which neurons were placed on a 2D substrate and transmission delays depended on the length of the connections, a developmental model used with an indirect encoding that could map a genome onto a neural network, and a genetic algorithm used to evolve controllers. The findings of this program raised several interesting issues.

Results have shown that using a biologically plausible sigmoid function to map chemical concentration to the total input current of a leaky integrate-and-fire neuron, agents were able to detect the whole range of chemical concentration as well as small variations. The sensory neurons used in this work are able to encode the stimulus intensity into appropriate firing rates.

This research also reveals that two different neural coding strategies can be used by a simple neural network to control an agent . Both temporal coincidence (of spikes) and firing rate encoding strategies were important mechanisms used by the same neural network in different environmental conditions.

In addition, realistic model of neural noise were shown to improve the behaviour of an agent to perform a task like chemotaxis.

Models used to evolve developmental neural controllers for agents have been created and results have shown that evolved agents could perform a relatively realistic and difficult task, and their neural controllers could encode information in space and time. In this work, the use of symmetrical structures was shown to have major benefits for the evolution of neural controllers.

Finally, a detailed analysis of the neural dynamics was conducted on an evolved neural network and has shown that the model generates controllers that use rather sophisticated neural coding strategies involving detailed temporal information. This analysis revealed that single spikes sent at specific moments could modify the whole activity of a network and the behaviour of an agent.

List of Publications

(see Appendix B)

Journal Paper

1. *Evolution of Bilateral Symmetry in a Developmental Neural Controller.* Nicolas Oros, Volker Steuber, Neil Davey, Lola Cañamero and Rod Adams. Paper has been reviewed and I will resubmit it with corrections in January.

Selected Conferences

1. *Optimal receptor response functions for the detection of pheromones by agents driven by spiking neural networks.* Nicolas Oros, Volker Steuber, Neil Davey, Lola Cañamero and Rod Adams (2008). In R.Trappl (ed.): Cybernetics and Systems 2008. Proceedings of the 19th European Meeting on Cybernetics and Systems Research, Vienna.

2. *Adaptive Olfactory Encoding in Agents Controlled by Spiking Neural Networks.* Nicolas Oros, Volker Steuber, Neil Davey, Lola Cañamero and Rod Adams (2008). Lecture Notes in Computer Science: From Animals to Animats 10, pp. 148-158.

3. *Optimal noise in spiking neural networks for the detection of chemicals by simulated agents.* Nicolas Oros, Volker Steuber, Neil Davey, Lola Cañamero and Rod Adams (2008). In Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems. MIT Press, Cambridge, MA.

4. *Evolution of Bilateral Symmetry in Agents Controlled by Spiking Neural Networks.* Nicolas Oros, Volker Steuber, Neil Davey, Lola Cañamero and Rod Adams (2009). Proceedings of the 2009 IEEE Symposium on Artificial Life (CI-ALife 2009).

Contents

Chapter 1

Introduction

1.1 Motivation and Goals

This thesis addresses the problem of creating simulated agents controlled by neural networks that share features with biological olfactory systems. My motivation was to investigate the relationship between the morphology and the physiology of a neural network and the behaviour of an agent controlled by such a network. I was also interested to study how information is encoded in such a system when evolved. This work draws from the fields of Artificial Life, Artificial Intelligence and Neuroscience.

- I used an approach often found in Artificial Life using evolutionary computation and developmental models to generate neural controller for agents that had to react to simulated chemicals.
- When investigating the neural controller of agents, I used an approach inspired by neuroscience. I was then able to look for similarities with biological olfactory systems.

I believe that this approach, based on Artificial Life research, may be useful to help neuro-scientists understand complex systems like olfactory systems. Finding an abstract model of development that can generate efficient neural networks is one the most promising goals of evolutionary computation (as described in the next paragraph), and its applications in robotics and in neural computation are multiple. Such a model could shed light on the evolution of natural systems like olfactory systems and also on the interaction between neural systems and behaviours. Neural computation taking place in olfactory systems has not yet been addressed by this approach.

It is usually difficult to create robust and adaptable neural controllers for agents that can perform a number of different actions. They often are optimized to perform one, or a small number, of simple tasks. A promising trend is to evolve neural networks using evolutionary computation, (see

[16, 23, 24, 57, 75, 67, 80, 83, 89, 106] for reviews). Evolutionary computation approaches allow researchers to design models or systems like neural networks or robots, with little human intervention [25, 23, 24]. With this optimization technique, it is possible to evolve different characteristics of a neural network, such as: its topology, the synaptic strengths for each connection, the learning rule or even evolve the body and the brain of an artificial organism. In the field of Artificial Life, certain researchers try to combine an evolutionary approach, with artificial development and learning, to generate efficient neural networks having a limited initial knowledge of the architecture needed [23, 24, 56, 5, 67, 75]. An artificial embryogenic (or ontogenic) approach is similar to a natural approach, in that an agent or robot, can learn and adapt to its surroundings over its lifetime, while the whole population of agents evolves over generations [5]. This approach allows evolved agents or robots to share certain features such as: robustness, flexibility, and modularity with certain biological systems [23, 24].

In nature, animals evolved for millions of years and scientists have now evidence that most of them can discriminate many different odours. Moths are a good example of such animals, having a very finely tuned olfactory system. Even with a relatively small brain, they can find a chemical source situated miles away and diffused at very small quantities (a few picograms per hour) [51, 105]. Insects like moths have a finely tuned sensory system linked to an olfactory system that can perform odour recognition in which timing of spikes, synaptic plasticity and neuromodulation all play an important role. Even though moth olfactory systems have been studied extensively, there are still many open questions on how olfactory computation works and how it generates and shapes behaviours. In order to understand olfactory systems, computational models have been created to complement experimental results [36, 51, 65, 64, 77]. In nature, olfactory systems encode odours as spatio-temporal neural activity patterns [64, 77]. This implies that olfactory information is coded both by the identity and the location of active neurons, and also by the temporal pattern of activity over the whole network. In fact, many real neural systems encode information by often using different types of encoding strategies based on the frequency of firing and/or on the time of firing.

During my PhD, I wanted to investigate how a neural network can encode information in order to discriminate odours and control an agent. I also wanted to investigate the evolution of a neural controller for agents that can encode temporal delays using spiking neurons and a developmental approach. Therefore, I wanted my model to be capable of temporal dynamics similar to those found in biological neural systems. I have used spiking neural networks with realistic transmission delays generated by the geometry of the model. This entailed placing the neurons on a 2D substrate, in which distances could be calculated. The time for a spike to be transmitted from one neuron

to another one depended on the length of the connection made between them. During these three years, I also wanted to investigate the importance of symmetry in the evolution of such neural controllers and how it could improve both the efficiency of the evolutionary process and the performance of an agent.

The work presented here has tried to answer the following questions:

- What computational strategies does a network of spiking neurons use to discriminate odours and control an agent?
- How such a neural architecture can encode information (as spatio-temporal patterns using different neural coding strategies) ?

My literature research, and source preliminary studies conducted, led me to related questions of methodology that could link the evolution of olfactory systems architecture and neural encoding, and the evolution of particular behaviour:

- How to evolve a neural controller for agents that can encode temporal delays using spiking neurons and a developmental approach?
- How can a spiking neural network encode information in order to control an agent that is attracted by a low level of concentration but repelled by a high level of the same chemical concentration? Is it necessary to have different types of sensory neuron that react to different concentration values in order to perform this task?

1.2 Contribution to Knowledge

The research conducted during my PhD have made the following contributions to the fields of Artificial Life, Artificial Intelligence and Neuroscience:

- I have created models that evolved developmental neural controllers for agents that could perform a relatively realistic and difficult task, and could encode information in space and time. I also have shown that in my models, the use of symmetrical structures had major benefits for the evolution of neural controllers. This work is important in that it addresses the problems of creating abstract models of development, of how information is encoded by a neural system and of generating agents that have realistic behaviours.
- I have conducted a detailed analysis of the neural dynamics on an evolved neural controller. This has shown that my model generates controllers that use rather sophisticated neural coding strategies involving detailed temporal information. This work is important in that

it shows the direct relationship between the activity of a neural controller and the resulting behaviour of an agent. It also shows how single spikes sent at specific moments can modify the whole activity of a network and the behaviour of an agent. To the best of my knowledge, this is the most detailed study of this type.

- I have also investigated in a simple manner two different neural coding strategies used by a simple neural architecture. I showed that both temporal coincidence (of spikes) and firing rate encoding strategies were important mechanisms that can be used by the same neural network in different environmental conditions. This work is relatively important in that it addresses the problem of how a neural system can use different neural coding strategies depending on external conditions.

- I have shown that by using a biologically plausible sigmoid function in my model to map pheromone concentration to the total input current of a leaky integrate-and-fire neuron, I could produce agents able to detect the whole range of pheromone concentration as well as small variations. The sensory neurons used in my model are able to encode the stimulus intensity into appropriate firing rates. This work is relatively important in that it shows how to map a stimuli into the firing rate of a spiking neuron using a biologically realistic approach.

- I have used a realistic model of neural noise and showed that it improves the behaviour of an agent. This work is relatively important in that it addresses the problem of the effect of noise on the performance of neural systems.

1.3 Structure of the Thesis

The rest of the manuscript is structured as follows:

Chapter 2 gives a review on neural systems, different models of neurons used in the litterature and two main coding strategies to represent information about stimuli. One coding strategy is based on the firing rate of a neuron and the other, based on the actual time of firing (spiking neurons). I discuss that simple spiking models, like integrate-and-fire neurons, can run fast enough compared to the complex and computationally slow Hodgkin and Huxley model, and still have a more realistic behaviour than firing rates. The usage of spiking neurons allows information to be encoded in different manners using different strategies. For these reasons, more and more researchers are implementing spiking neurons in robots and simulated agents.

Chapter 3 first introduces the common features of olfactory systems and how they process chemical information. It then introduces the theories of how such systems compute. Then, it presents the work on olfaction that has been done in the field of Artificial Intelligence. Finally, it includes a discussion on the evolution of olfactory systems and indicates how this relates to the research questions that are the subject of my PhD.

Chapter 4 presents the model of neurons used in this thesis (leaky integrate-and-fire). It also presents the agent and its environment, and preliminary experiments conducted on the creation of the sensory neurons able to encode the stimulus intensity into appropriate firing rates. The main goal of these experiments was to create agents capable of finding and reacting to chemicals diffused uniformely from a point source. In order to achieve this goal, I had to find a model of spiking sensory neuron that could cope with small variations of pheromone concentration but could also react to the whole range of concentrations. It is already known that the mapping between the current and the firing rate of a leaky integrate-and-fire neuron is non-linear. Therefore, I tried many different functions to map the pheromone concentration onto the current of the sensory neuron in order to produce a reasonably linear relationship between the concentration and the firing rate of the sensor. After unsuccessful trials using linear currents, I derived a function that would necessarily give an exact linear relationship and used it as a model to help me find a similar function that is also used in biology. I concluded that by using a biologically plausible sigmoid function in my model to map pheromone concentration to current, I could produce agents able to detect the whole range of pheromone concentration as well as small variations. The sensory neurons used in my model are able to encode the stimulus intensity into appropriate firing rates. Moreover, using this model of sensory neurons, I managed to create an agent capable of chemotaxis.

Chapter 5 first presents experiments conducted using two different neural coding strategies in a neural controller of an agent. In this work, I used a simple neural architecture where temporal coincidence (of spikes) and firing rate encoding strategies were both important mechanisms used in different environmental conditions. In a low chemical concentration setting, synchronization of spikes sent by the sensors was essential to allow the agent to detect the blend of two chemicals. I changed the sensory delays and noticed that the agent was then not able to react to the chemicals anymore. However, in a high chemical concentration setting, the temporal coincidence between sensors firing was not a necessary condition and the agent was able to stay inside the chemical concentration using just the firing rate encoding strategy even in the presence of just one chemical. This model also showed much more

sensitivity to the presence of two chemicals than a single chemical. In principle, more than two chemicals can be detected and processed.

The second part of this chapter discusses the effect of noise on the agent's behaviour using the neural architecture from the previous experiments. I constructed a more complex environment using chemical gradients and a realistic model of neural noise. I found that the overall fitness of the agent was better when a certain amount of noise was added in the neural network. These results suggest that a realistic model of noise can improve an agent's behaviour. This is further evidence that adding biologically realistic features can be beneficial for certain engineering tasks, and suggests a potential function of noise in real biological systems. The effect of biologically realistic noise should be an interesting topic of research in other artificial life scenarios. I need to emphasize the fact that I might have the same results by simulating environmental noise or sensory noise instead. I think it would be interresting to add neural noise in real robotic experiment to study its effects.

Chapter 6 summarises work undertaken using an evolutionary approach and three novel developmental models allowing information to be encoded in space and time using spiking neurons placed on a 2D substrate. In two of these models the neural developmental model can use bilateral symmetry. I show that these models created neural controllers that permit agents to perform chemotaxis, and do so much better than controllers that were evolved from models that made no intrinsic use of symmetry. I also show that with the model using evolvable symmetry, neural bilateral symmetry is often evolved and was found to be beneficial for the agents. This model, together with the model that coded for symmetrical neurons at all times (that is, it had enforced symmetry) show that the use of symmetry is clearly advantageous allowing faster evolution. Using a model with no explicit symmetry, no correct solutions were found during the allocated time; however, this model should in theory find a correct solution if the genetic algorithm would run longer. Also, there were no restrictions about the number of neurons that each of the three models could create. All the networks evolved with the three models could have the same number of neurons and connections. It is important to note that complexification, targeting and neural selection are important concepts in the model. I used a 2D neural substrate where spiking neurons are placed and can grow connections to target locations. Therefore, the geometric configurations of the neural network significantly matter. Since I used spiking neurons with tranmission delays, distances separating connected neurons result in time delays between the points in time when spikes are sent by a neuron, and the times they are received by another neuron. A neural network generated by my developmental

models can encode information not only using firing rate encoding but also using the time of spikes. Evolution can therefore generate neural networks able to encode external information as spatio-temporal patterns. In a system as complex as a spiking neural network placed on a 2D substrate in which both neural position and connectivity are evolved, the exploitation of the physical symmetry of an agent has significant advantages allowing a more compact genetic representation leading to a relatively fast evolution of efficient neural networks. This work was the first, as far as I know, to present developmental models where spiking neurons are generated in space and where bilateral symmetry can be evolved and proved to be beneficial in this context. I think that studying how evolutionary processes can be affected by symmetrical structures in neural networks is of major importance and will have beneficial repercussions on Artificial Life research. One aspect of Artificial Life investigates major transitions in artificial and real evolution and symmetry surely plays an important role in this process. I also emphasize that the creation of neural controllers having the possibility to use different neural coding strategies, using spiking neurons, is a very interesting and promising approach.

Chapter 7 presents experiments that shows that using my developmental model, a neural controller exhibiting complex dynamics where timing of spikes is a key element, can be evolved. In the experiments presented in this chapter, agents were evolved to be attracted by a low level of concentration but repelled by a high level of concentration. The agents were evolved to maximize their energy value by moving as close as possible to a concentration of 150 and avoiding higher values. The artificial evolution managed to create an efficient neural controller that uses complex neural coding based on firing rates and spike timing, and where bilateral inhibition played a major role. Even though the evolved neural controller has a small number of neurons and connections, it exhibits an emergent coding strategy that is relatively complex. In this experiment, I also showed that the agent could adapt quite well to different environments such as when the chemical covered different extents of the environment. In this chapter, I present a detailed study on the neural dynamics of the fittest evolved agent and show that it used specific encoding based on temporal coincidence and firing rate. This analysis is the first, as far as I know, to have been done on an evolved neural network. These results shows that an Artificial Life approach to the study of natural processes like olfaction is feasible, and can give a better understanding of different encoding strategies used by neural systems.

Chapter 8 summarises the conclusions of each chapter as a conclusion of this study. In addition,

a number of further research directions are suggested which would extend the contributions of research reported here.

Chapter 2

Neural Systems

2.1 Introduction

The nervous system of animals allows them to interact with the world around them by processing external signals coming from the environment, and by controlling their movements. The main components of nervous systems are neurons and glia cells. Neurons are cells responsible for signal transmission inside the body. There are many different types of neurons but we can categorize them in three main classes: sensory neurons, interneurons and motorneurons. Sensory neurons map information from the environment onto electrical signals. Inter-neurons process information coming from sensory neurons or other inter-neurons and forward it to other parts of the nervous system. Motorneurons, when stimulated by other neurons, control body parts by mapping electrical signals to movements. Glia cells are cells that support neurons in different ways. They keep neurons in place by surrounding them, provide them nutrients and oxygen, destroy pathogens, insulate them and remove dead neurons. One of the main differences between the two is that glia cells do not contribute actively to information processing.

As neurons are the key units for signal transmission in nervous systems, this chapter will focus on neural systems neglecting the role of glia cells. In the following, I will first describe the general characteristics of a typical neuron, give an introduction to the different manners neurons can encode information and I will present different models of neurons and conclude.

2.2 Neurons

A neuron is a nerve cell that can transmit electrochemical signals. It generally receives input signals sent by other neurons onto its dendrites and cell body (or soma) (see Figure (2.1)). If a neuron is stimulated enough by its inputs, so that its membrane potential reaches a firing threshold,

an electrical pulse (spike or action potential) is generated and sent to other neurons through its axon. An action potential usually has an amplitude of approximately 100mV and a duration of approximately 1ms (see Figure (2.3)) [49]. Typically, once a neuron has sent a spike, it cannot emit another one during a certain time (refractory period). The link between an axon terminal bouton and a dendrite is called a synapse. Chemical transmitters (neurotransmitters) are released when a pulse arrives at a synapse. Usually, the amount of transmitter released depends on the number and the frequency of the action potentials. These transmitters will bind to postsynaptic receptors on another neuron that will increase or decrease the post-synaptic neuron's membrane potential. As any other cell, neurons carry genetic information. Their genes are stored in the nucleus. It has been observed that the human brain contains around 10^{11} neurons of at least a thousand different types, with each one of them receiving input signals from about $10^3 - 10^5$ other neurons.

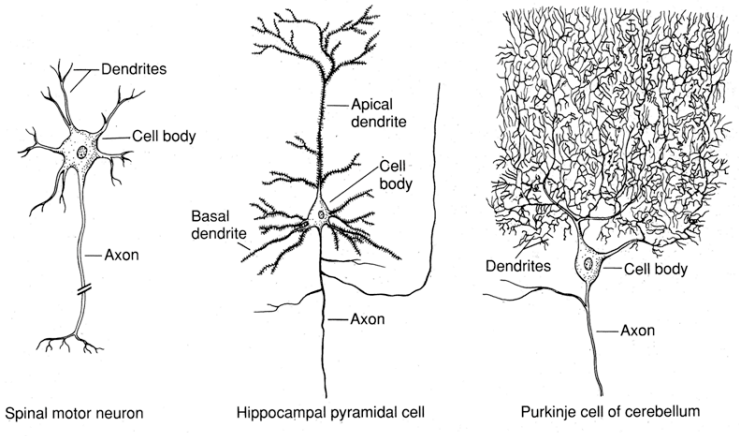


Figure 2.1: Example of three neurons with different structures.

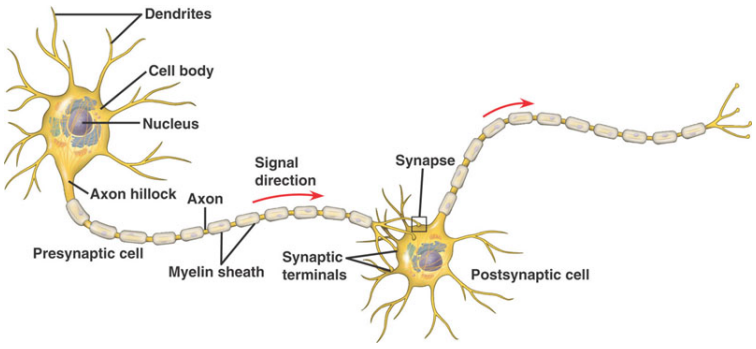


Figure 2.2: Drawing of two motor neurons and their connections. Figure from http://kvhs.nbed.nb.ca/gallant/ biology/ neuron_structure.html.

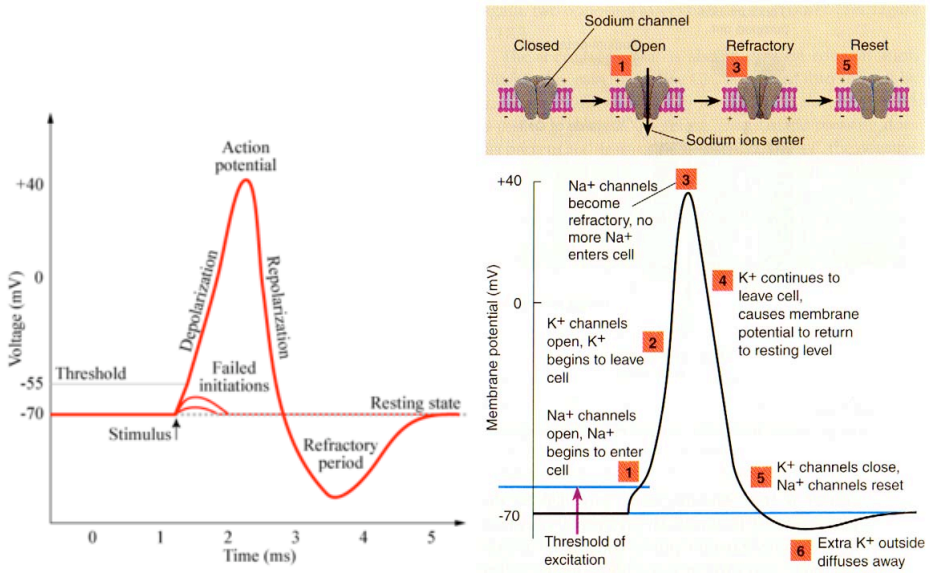


Figure 2.3: Typical action potential (electrical spike) produced by a neuron. When the neuron receives a stimulus, its potential increases. If a certain threshold is reached, the neuron's ion channels (Sodium, Na+, and Potassium, K+) open leading to a fast depolarization until the potential reaches a maximum value. The membrane is then repolarized decreasing the potential until it reaches a minimum value during the refractory period. During this period of time, the neuron cannot generate another spike. After stimulation, the neuron's membrane potential settles back into its resting state. Left figure from: http://www.chm.bris.ac.uk/webprojects2006/Cowlishaw/300px-Action-potential.png. Right figure from: http://www.gregalo.com/neuralnets.html

Neurons process sensory information to control muscles in the body. We can try to understand how the nervous system produces behaviour by focussing on its main features [49]:

1. The mechanisms by which neurons produce signals.
2. The patterns of connections between neurons.
3. The relationship between different patterns of interconnection and different types of behaviour.
4. The way in which neurons and their connections are modified by experience.

2.3 Development of Neural Systems

The development of the nervous system involves four main stages of different processes [104]. These processes are similar to those found in other developmental systems (for more details on

development, see Appendix A).

First, the neural tissue is specified early in development. During this stage, neural cells will specialize to become glia cells, motor neurons, sensory neurons or inter neurons, and each of these categories can have many different types of neurons. The main processes generally involved in specialization are lateral inhibition, asymmetric cell divisions and cell-cell signalling (see Appendix A for more information on these processes).

The second main stage involves the migration of neurons and the growth of axons to the targets cells. During this stage, neurons usually move to a specific location depending on the pattern of concentration of certain chemicals. Then, the growth of their axons is also guided by attractive or repulsive chemical signals.

Once axons find their targets, synapses can be formed with the target cells. The specificity of neurons often involves a competition between many neurons to create a connection to a particular target. A large number of neurons are created in early development, however many will die during this competition. For example, 50% of the initial population of motor neurons of vertebrates' limb die during this process [104] . The survivors are the ones that managed to create a connection to a target. In many cases, it seems that cells are programmed to die unless they receive a particular signal [104] (see also cell death in Appendix A).

Finally, synaptic connection can be refined through elimination of axon branches and cell death. Neural activity plays a major role in refinement of synaptic connections.

2.4 Neural Coding Strategies

The brain has the ability to process information in many ways using different coding strategies, mainly based on firing rates of neurons or specific times of spikes. One of the first main neural coding strategy maps relevant information onto the firing rate of neurons (rate coding) (see Figure 2.4.a). Experimental results have shown, for example, that the relationship between increases in frequency of neural firing and pressure onto the skin is linear [49]. However, if the brain was using solely this encoding strategy, it would be extremely slow and animals would not be able to react fast enough to certain stimuli. Computing an average firing rate requires a relatively long time which can be disadvantageous for rapid reactions.

For this reason, the brain also uses the timing of individual spikes. Certain neurons do not fire many spikes per second but respond to a particular stimulus by sending only a few spikes [60]. Information can be encoded by the specific time of spikes (temporal code), by the synchrony of spikes sent by different neurons (correlation code or temporal coincidence encoding) (see Figure

2.4.b) and also by the delay of firing (delay coding) (see Figure 2.4.c). Temporal coding, using both temporal coincidences and delays, is used by bats to find prey in the dark using a remarkable navigation system called echolocation [28]. Delay coding is a general feature of neurons as the more a neuron is stimulated, the faster it will fire. However, this encoding strategy is very sensitive to noise compared to the firing rate encoding. Correlation coding is used by olfactory systems in insects (see Chapter 3).

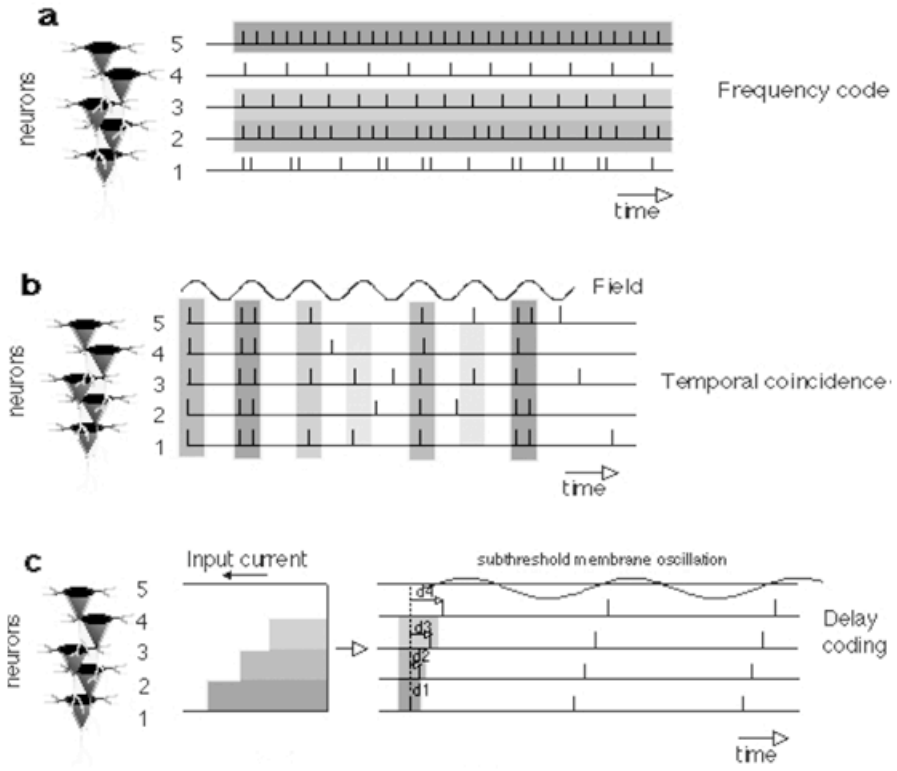


Figure 2.4: Three main neural coding strategies. Fictive spike trains recorded from five neurons. The different stimulus intensities (represented by gray scale) are converted to different spike sequences. (a) Neurons generate different frequency of spikes as a response to different stimulus intensities (frequency code or rate coding). (b) Tighter coincidence of spikes recorded from different neurons represent higher stimulus intensity (temporal coincidence encoding). Spikes occurrences are correlated with local field oscillation. (c) In the delay coding hypothesis, the input current is converted to the spike delay. The stronger a neuron is stimulated, the faster this neuron will fire. Different delays of the spikes (d2-d4) represent relative intensities of the different stimulus. Figure from Floreano and Mattiussi [25]

2.5 Models

Many scientists have tried to produce models of nervous systems composed of models of neurons with different degrees of realism. More complete reviews of these models can be found in [26, 30, 53]. An example of a basic artificial neural network is shown Figure 2.5.

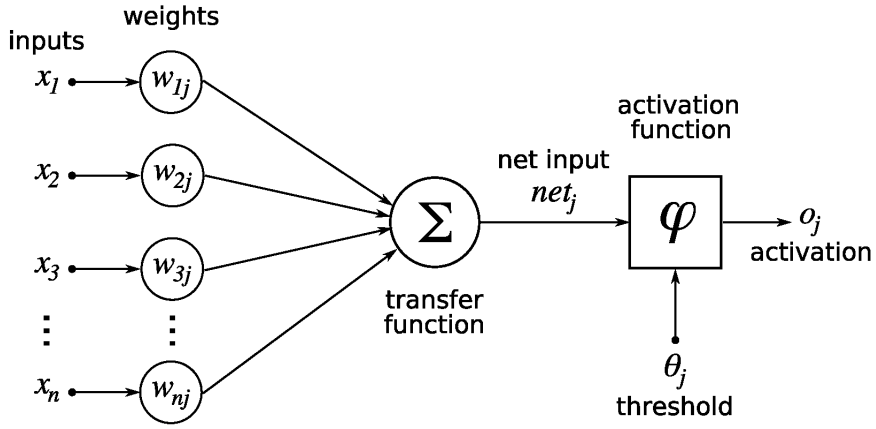


Figure 2.5: Example of a basic artificial neural network composed of one neuron receiving n inputs. An artificial neuron is defined by a transfer function and an activation function. The transfer function usually sums all the input values X_i multiplied by the weights w_{ij}. There are different types of activation functions that can modify the dynamics of the neuron. Most of them use a modification of the principle that a neuron is activated when the net input reaches a certain threshold.

Figure from: http://en.wikibooks.org/wiki/Artificial_Neural_Networks/Print_Version

2.5.1 Hodgkin and Huxley model

This model was first published in 1952 based on observations made on the squid axon [40, 43, 42, 39, 41]. A Hodgkin and Huxley type neuron has a membrane with a certain capacitance. The ionic currents coming from the different inputs and the intrinsic ion channels charge this capacitance:

$$C\frac{dV}{dt} = -\sum_k I_k + I_{ext} \tag{2.1}$$

where:

- V is the membrane potential.
- I_{ext} is an external driving current.
- $\sum_k I_k$ is the sum of the intrinsic ionic currents in the membrane.

There are three different ionic currents in the original Hodgkin-Huxley model(2.2): Sodium (Na), Potassium (K) and a leakage current (L).

$$\sum_k I_k = g_{Na} m^3 h \left(V - E_{Na}\right) + g_k n^4 \left(V - E_k\right) + g_L \left(V - E_L\right) \tag{2.2}$$

- g_{Na}, g_k, g_L are constant conductances.
- $E_{Na}, E_k, E_L$are reversal potentials.
- m, n, h are activation and inactivation variables that change according to the following equations (2.3):

$$\frac{dm}{dt} = \alpha_m \left(V\right)\left(1 - m\right) - \beta_m \left(V\right) m \tag{2.3}$$

$$\frac{dn}{dt} = \alpha_n \left(V\right)\left(1 - n\right) - \beta_n \left(V\right) n \tag{2.4}$$

$$\frac{dh}{dt} = \alpha_h \left(V\right)\left(1 - h\right) - \beta_h \left(V\right) h \tag{2.5}$$

where a and β are voltage dependent rate constants. This model can be fitted easily to experimental data so that the behaviour of real neurons can be approached. Unfortunately, solving the equations of this model requires a lot of computational power so only a limited number of neurons can be simulated on a basic computer.

2.5.2 Simpler spiking models

Simpler spiking neuron models have been created in order to simulate neurons with similar behaviours to real neurons but running faster than Hodgkin and Huxley models. These models do not contain explicit representation of ion channels but they are still able to simulate spike emission and integration of the input currents. For each class of models, many different versions have been created but only the main ones are presented here.

2.5.2.1 Leaky integrate-and-fire model

This model is the most common and one of the easiest to simulate. A neuron acts as a leaky capacitor (capacitor in parallel with a resistor)[26, 30, 53]. The spatial structure and different ionic currents of real neurons are neglected. The input currents charge the capacitance and change the membrane potential V according to the following equation:

$$\frac{dV}{dt} = -\frac{V}{\tau_m} + \sum I \tag{2.6}$$

- V is the membrane potential.
- τ_m is the neuronal time constant.
- $\sum I$ is the sum of input currents densities (in Amperes per Farad).

When the potential reaches a certain threshold ϑ, the neuron emits a spike and its potential is set to a reset value. A refractory period can be simulated so that the neuron cannot fire for a certain time after having emitted a spike. A spike sent by a presynaptic neuron will take some time to arrive at a postsynaptic neuron. This time delay depends on the distance between the sender and the receiver. All the spikes arriving at a neuron are summed to calculate the neuron's membrane potential. A possible equation for the input current *I(t)* is:

$$I(t) = \left(\frac{t - (t_{spike} + delay)}{\tau_s}\right) exp\left(\frac{1 - (t - (t_{spike} + delay))}{\tau_s}\right) \tag{2.7}$$

- t is the actual time of the simulation
- t_{spike} is the precise firing time of an incoming spike.
- τ_s is the synaptic time constant
- *delay* is the time delay a spike takes to travel from a presynaptic neuron to a postsynaptic neuron.

2.5.2.2 The spike response model

This model is a generalization of the integrate-and-fire model where the dynamics of a neuron depends on the last time it received a spike. Depending on parameters, some versions resemble the leaky integrate-and-fire model and others the Hodgkin-Huxley model[26, 30]. However, compared to a leaky integrate-and-fire model, the membrane potential at time step A does not depend explicitly on the voltage at *A-1*. The model presented here is the Spike Response Model developed by Gerstner [31, 30].

The membrane potential of a neuron i at time t is given by:

$$V_i(t) = \sum_j w_j^t \sum_f \epsilon_j^f(s_j^f) + \sum_f \eta_i^f(s_i^f) \tag{2.8}$$

- $s_{ij} = t - t_j^f$ is the difference between the actual time t and the firing time t^f of a spike that has been sent by a neuron j to a neuron i.
- $\sum_j w_j^t$ is the sum of the weights of the inputs.
- $\sum_f \epsilon_j^f(s_j^f)$ represent the effects of the inputs (in Volts) on the membrane voltage by summing the effect of each spike emitted by a neuron j.
- $\sum_f n_i^f(s_i^f)$ is a sum of the potentials used to implement the relative refractory period. Refractoriness and adaptation are modeled by the combined effects of the hyperpolarizations of several previous spikes, rather than only the most recent spike.

One of the main concepts usually implemented is that a spike sent by a neuron takes some time to arrive at the post-synaptic neuron. The spike that arrives has a direct effect ϵ on the membrane potential that will gradually fade as time goes. A possible function simulating this behaviour is:

$$\epsilon(s) = \begin{cases} exp\left(\frac{-(s-\Delta)}{\tau_m}\right)\left(1 - exp\left(\frac{-(s-\Delta)}{\tau_s}\right)\right) & , if\ s \geq \Delta \\ 0 & otherwise \end{cases} \tag{2.9}$$

- τ_m is the neuron time constant.
- τ_s is the synapse time constant.
- Δ is the delay between the generation of a spike at the pre-synaptic neuron and the time of arrival at the synapse.

To simulate the refractory period where the membrane potential of a neuron that just sent a spike is set to a low value and gradually recovers to its resting voltage, the following function can be used:

$$\eta(s) = -exp\left(\frac{-s}{\tau_m}\right) \tag{2.10}$$

2.5.2.3 The Izhikevich simple model (ISM)

This recent model has been created to simulate very realistic neuronal behaviours with low computational complexity (see Figure 2.6) [45, 46]. Such neurons can have different realistic dynamics (bursting, chattering, adaptation, resonance...) (see Figure 2.7).

The ISM model is based on a two-dimensional system of ordinary differential equations of the form:

$$\frac{dV}{dt} = mV^2 + nV + p - u + I(t) \tag{2.11}$$

$$\frac{du}{dt} = a(bV - u) \tag{2.12}$$

- V is the membrane potential of the neuron.
- u is the membrane recovery variable (voltage decay in Volts per second).
- m, n, p are constants defining the model.
- a, b are parameters determining the intrinsic firing pattern of the neuron.
- $I(t)$ is the external current density driving the neuron.

When V reaches the threshold, a spike is sent and the variables are reset to:

$$V = c \tag{2.13}$$

$$u = u + d \tag{2.14}$$

where c, d are constant parameters.

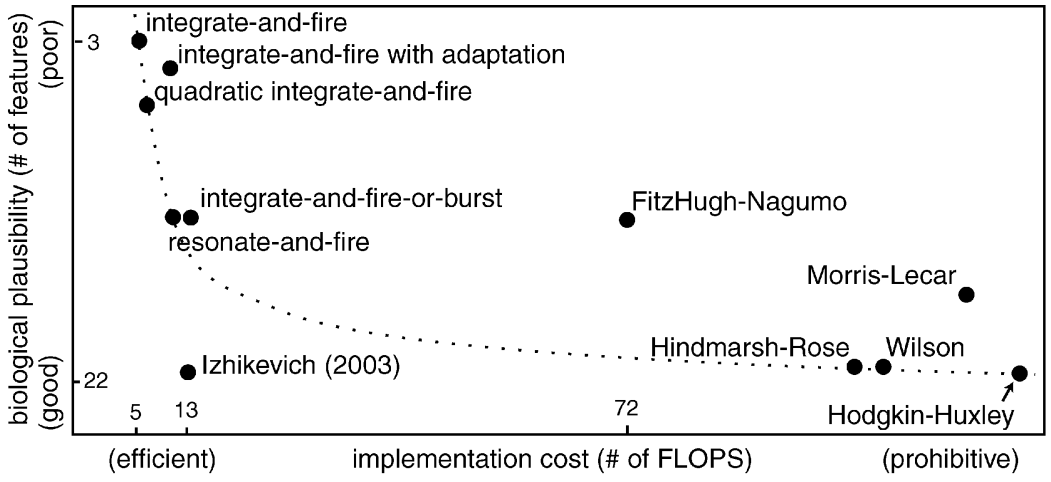


Figure 2.6: Comparison of the neuro-computational properties of spiking and bursting models; "# of FLOPS" is an approximate number of floating point operations (addition, multiplication, etc.) needed to simulate the model during a 1 ms time span. Figure and caption from Izhikevich [46].

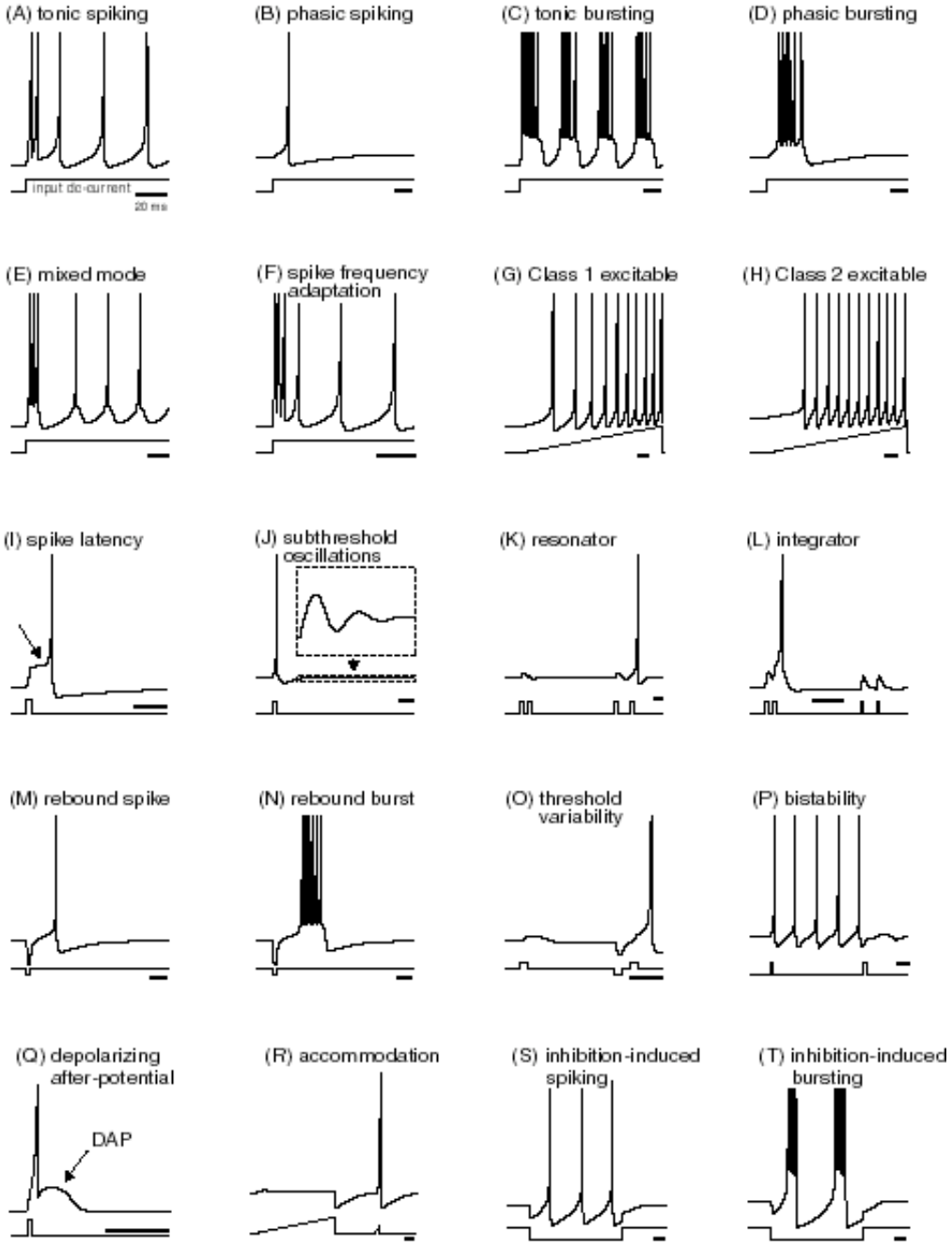


Figure 2.7: Types of neurons simulated with the Izhikevich model. Figure and caption from Izhikevich [46]

2.5.3 Firing rate models

In these models, the communication between neurons is based on the firing rate alone [26, 30, 53]. They neglect precise times of spikes by averaging their numbers to calculate the frequency. The closest models to real neurons use continuous time and a leaky temporal integration of the inputs.

The function of the membrane potential can be the same as for leaky integrate-and-fire neurons:

$$\frac{dv}{dt} = -\frac{v\,(t)}{\tau_m} + \sum I\,(t) \tag{2.15}$$

The main difference to the models presented before, is that the generation of spikes is not represented explicitly. The output firing rate is a continuous but nonlinear function of $v(t)$.

$$f = g(v) \tag{2.16}$$

Usually, g is a sigmoidal function which is monotonically increasing, positive and saturating at a certain value.

$$f = \frac{1}{1 + exp^{-\beta v}} \tag{2.17}$$

where β is a constant parameter that determines the steepness of the sigmoid.

A linear saturated function is also commonly used:

$$f = \begin{cases} 0, & if\, v < 0 \\ v, & if\, 0 \leq v \leq 1 \\ 1, & if\, 1 < v \end{cases} \tag{2.18}$$

2.5.4 McCullogh-Pitts Model

Their model was the first model of a neuron [68]. The neurons act as simple threshold gates. The inputs and outputs are binary. The output value Y is set by an activation function (step function or Heaviside function). If the sum of the weighted inputs ($\sum W_{in} X_n$) is above a certain threshold θ, the output is set to '1' (See Figure 2.5).

$$Y = \begin{cases} 1, & if\ \sum W_{in} X_n \geq \theta \\ 0 & otherwise \end{cases} \tag{2.19}$$

2.6 Conclusion

This chapter has described different models of neurons and two main approaches of representing information about stimuli. One coding strategy is based on the firing rate of a neuron and the other, based on the actual time of firing (spiking neurons). I discussed that, compared to the complex and computationally slow Hodgkin and Huxley model, simple spiking models like integrate-and-fire neurons can run fast enough and have a more realistic behaviour than firing rate ones. I also

mentioned that the usage of spiking neurons allows information to be encoded in different manners using different strategies. For these reasons, more and more researchers are implementing spiking neurons in robots and simulated agents.

Chapter 3

Olfactory Systems

3.1 Introduction

All animals detect and react to chemicals in their environment. These chemicals can be odours or pheromones indicating food resources, shelter or predators [105]. The main sense used to detect these chemicals is olfaction (smell) rather than gustation (taste). The complex nature of the stimulus space makes olfactory systems unique compared to other perceptive systems, such as auditive or visual systems where stimuli can be represented as signals with different frequencies and amplitudes. Chemical signals are not continuous and are highly dependent on the medium propagating them (water, air). Chemicals that diffuse from an odour source without disturbances follow a gradient with a Gaussian shape. However, many odours and pheromones are diffused into air or water, and the chemicals are transported as plumes which are turbulent and unpredictable. Even if chemical signals are spatially and temporally heterogeneous and chaotic [105, 72], animals have evolved incredibly well performing olfactory systems to be able to detect and avoid predators or find mates from very long distances.

Chemical senses are the oldest and they are shared by all living organisms even by bacteria [105, 101, 102]. Animals, which evolved from unicellular organism, are therefore pre-adapted to detect chemicals in their environment [105, 101, 102]. Recent scientific discoveries have shed light on olfactory systems and shown that they are are very similar across the animal kingdom [105, 64, 90].

In this chapter, I first introduce the common features of olfactory systems and how they process chemical information. I will then introduce the theories of how such systems compute. Then, I will present the work on olfaction that has been done in the field of artificial intelligence. Finally, I will talk about the evolution of olfactory systems and how this relates to the research questions that are the subject of my PhD.

3.2 Odour Perception and Discrimination in Nature

Olfactory Receptor Neurons

In order to detect odours, animals must have receptor neurons that react to certain chemicals. An olfactory receptor neuron (ORN; also named olfactory sensory neuron: OSN) is a type of nerve cell that has an end exposed to the outside world, often through an intermediate medium such as skin or a cuticle (see Figure3.1). An ORN is stimulated when a chemical from the outside world binds to an olfactory receptor protein (ORP) present in its cell membrane. This binding triggers chemical reactions and the opening of ion channels leading to the production of an electrical pulse (spike) that will be sent down the axon of the ORN to the brain. The process of mapping chemical stimuli to neural activity of an ORN is called ***transduction***.

The 2004 Nobel Prize winners in Physiology or Medicine, Richard Axel and Linda Buck [9], discovered that around 1,000 ORPs are genetically encoded in a large (probably the biggest) gene family, comprised of some 1,000 different genes (three per cent of our genes). They also found that in mice, ***one ORN expresses only one ORP***. This can be found in almost every species [12, 35]. However, some animals have ORNs that express more than one receptor (e.g C.Elegans... [66, 96]). Although this suggests that they cannot perform odour discrimination as well as other organisms, C.Elegans seem to be able to do so [66, 96]. Another important fact is that odours are usually composed of more than one chemical, sometimes hundreds, therefore each one of them will activate several receptor proteins. This will create a specific spatial pattern of activated ORNs for each odour.

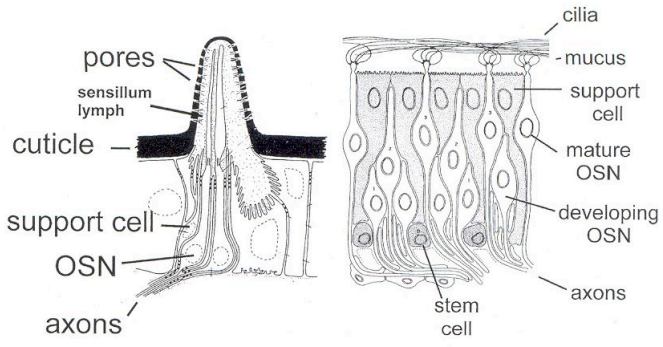


Figure 3.1: Schematic diagrams of: (Left) an insect (moth) olfcatory sensillum (hair) with two OSNs; (Right) vertebrate OSNs in olfactory epithelium. Unlike other neurons, OSNs are short lived and continually replaced from stem cells in the adult, every 30 days or so. Figure and caption from Wyatt [105][pp 168].

Important phenomena happening in ORNs are ***adaptation*** and ***desensitization*** [35, 72]. A reduction in responsiveness always follows the stimulation of an ORN by a particular stimulus. In other words, there is a decline of response to a maintained stimulus and a reduced reaction to repeated stimuli. Humans know this phenomenon well when they use perfume for example, they can smell it when they spray it on themselves but do not anymore after a certain amount of time. Adaptation and disadaptation (recovery from adaptation) are key factors in the perception of chemicals. It seems that the response of ORNs to chemical stimuli depends on the rate of adaptation and disadaptation, and the frequency of the stimulation [72]. To summarize, the frequency of spikes in a ORN depends on the concentration of the stimulus [81] however, the stimulus history matters and there is no simple linear mapping between chemical concentration and firing rate of an ORN.

When these ORNs react to chemicals, they activate different parts of the brain depending on the nature of the chemicals and their roles. However, most ORNs are connected to glomeruli which are nerve junctions used as units for the processing of odour information.

Glomeruli

In olfactory systems, a glomurulus is a cluster of nerve fibers made of a globular tangle of axons from the ORNs and dendrites from other cells. All ORNs expressing the same receptors usually converge on to the same glomerulus (see Figure3.2) [35, 3, 4, 85, 64, 77]. Therefore, ***each glomerulus seems to be specific to a particular receptor***. However, things are not that simple as one receptor protein can react to more than one chemical. So the reaction to a single chemical can create a particular spatial pattern of activated glomeruli [35, 77]. Glomeruli seem to be very important for odour discrimination; they can sum all information coming from ORNs of the same type, therefore increasing the sensitivity to particular chemicals. They are also responsible for encoding stimulus intensity: the higher the concentration, the more glomeruli are recruited [35, 3, 4, 85, 77, 63, 62]. Glomeruli also seem to have direct implications for animal behaviour (for example, in fruit flies, individual glomeruli can mediate innate behaviours like attraction and aversion) [85]. They may also play a major role in the initial perception of odour quality in rats, where the stimulus seem to be encoded by the earliest-activated glomeruli [97]. This suggests that odours are processed using encoding stategies based on the specific time of spikes rather than on simple changes in spike rate. Neuroscientists also found that each glomerulus has an output connection to only one postsynaptic neuron [64, 65, 77, 35, 9] (for example, a projection neuron in the antennal lobe of insects, a mitral cell in the olfactory bulb of vertebrates).

The main features of the early processing taking place in the main antennal lobe or main olfactory bulb can be summarised as follows:

- each ORN expresses one receptor type
- all ORN expressing the same receptor converge on to the same glomerulus
- projection neurons (or mitral cells) receive inputs from one glomerulus
- therefore, it seems that odours are represented by overlapping patterns of projection neuron (or mitral cells) activity (see Figure3.2).

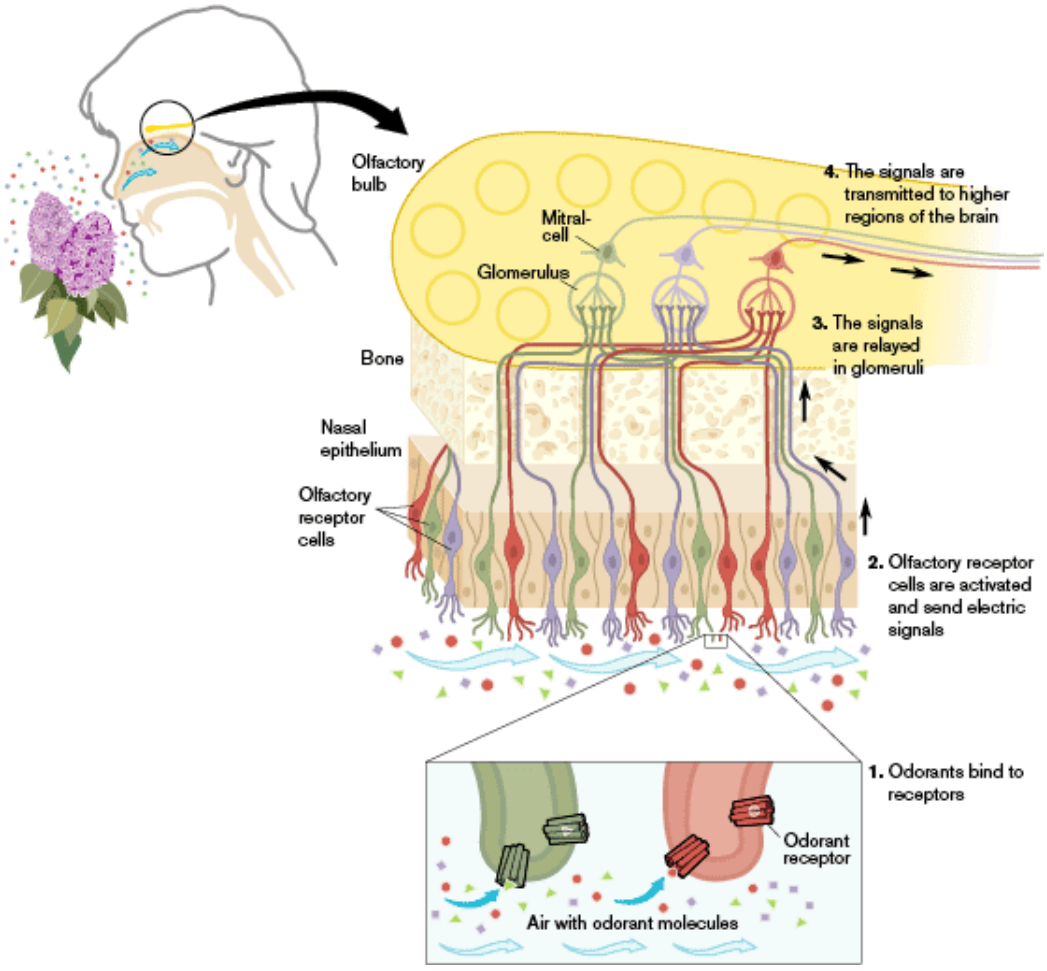


Figure 3.2: Schematic showing different types of olfactory receptor cells (neurons) in the human nose and their connections. All ORNs of the same type, represented by colour here, converge on to the same glomerulus. One mitral cell receives input from one glomeruli. Figure from : "Press Release: The 2004 Nobel Prize in Physiology or Medicine" http://nobelprize.org/nobel_prizes/medicine/laureates/2004/press.html

Similarities across Phyla

In recent years, biologists have found evidence that vertebrates and invertebrates have very similar olfactory pathways (see Figures 3.4 and 3.5)[35].

Most vertebrates have a main and an accessory olfactory system. The main system processes chemical stimuli activating ORNs from the olfactory epithelium (in the nose) that send information to mitral cells in the main olfactory bulb, which then transmit signals to higher levels in the brain. The accessory system processes information coming from the ORNs present in the vomeronasal organ (VNO) to the accessory olfactory bulb, which transmits signals to other areas such as the vomeronasal amygdala. The VNO seems to be important in the perception of social chemosensory stimuli such as pheromones (this is unknown in humans). Some say that the VNO perception is "unconscious" as no information is sent to higher level of the brain known to be used for cognition. However, this is highly debatable and is not the topic of this thesis so I will not go into more details. A summary of olfactory pathways is shown Figure 3.3.

In invertebrates, very similar pathways can be found. In moths for example, odours stimulate ORNs from the antennae which activate projection neurons in the main antennal lobe (equivalent to the main olfactory bulb) that transmit information to Kenyon cells in the mushroom body. These forward the information to higher levels of the brain. In parallel, pheromones stimulate particular ORNs that activate a subsystem in the antennal lobe called the macroglomerular complex (MGC) which transmits information to the mushroom body or to the inferior lateral protocerebrum .

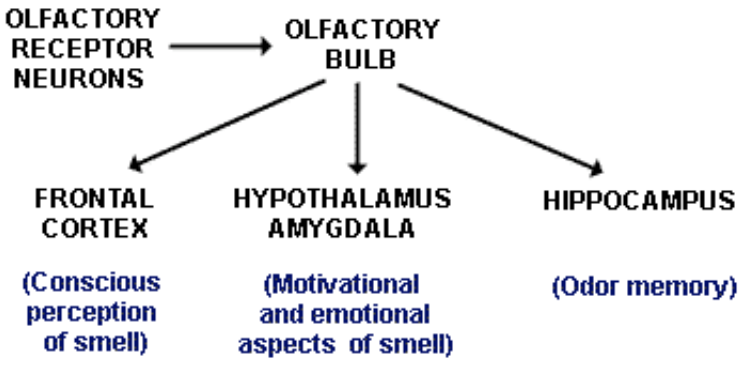


Figure 3.3: Summary of olfactory pathways in Humans.
Figure from http://faculty.washington.edu/chudler/chems.html.

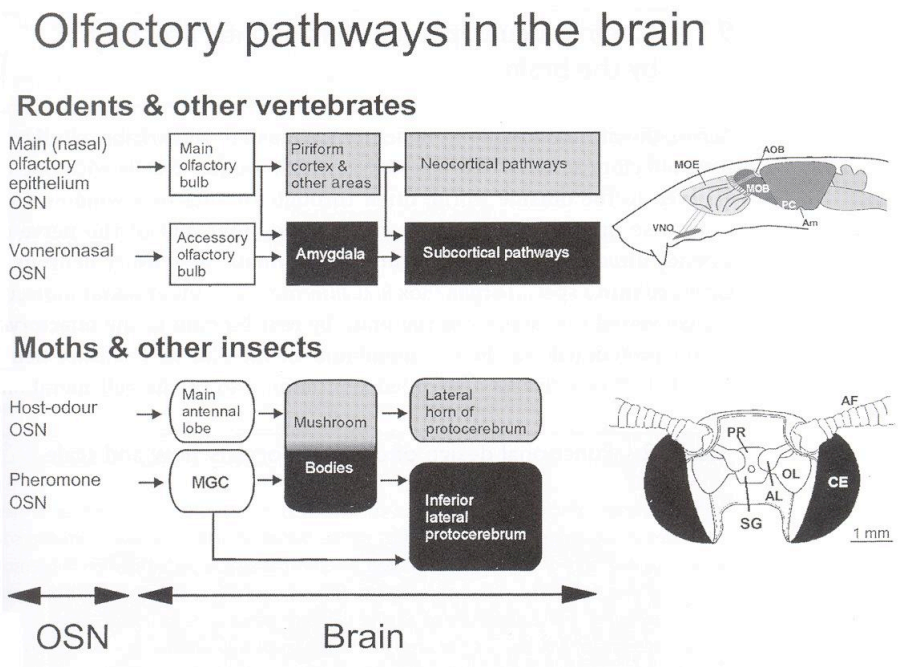


Figure 3.4: Similarities between the olfactory pathways in vertebrates and insects. In rodents and vertebrates (top), the main olfactory bulb (MOB) receives input from olfactory sensory neurons (OSN) in the main epithelium (MOE) and send projections to higher areas in the brain such as the piriform cortex (PC). Rodents and many other vertebrates also have an accessory olfactory bulb (AOB) which receives inputs from sensory neurons in the vomeronasal organ and send projections to areas such as the vomeronasal amygdala (Am). In moths (bottom), the antennal lobe (AL) receives input from OSNs along the length of the antenna (AF). In many insects, the AL in males is divided into two subsystems, one for processing information about general odorant (Main AL), the other devoted to species-specific information about female sexual pheromones (the macroglomerular complex, MGC). Many AL projections converge on the mushroom body in the protocerebrum (PR), whereas some projections from the MGC bypass the mushroom body completely. Other abreviations: OL, opic lobe; CE, compound eye; SG, suboesophageal ganglion. Figure and caption from Wyatt [105] [pp 165]

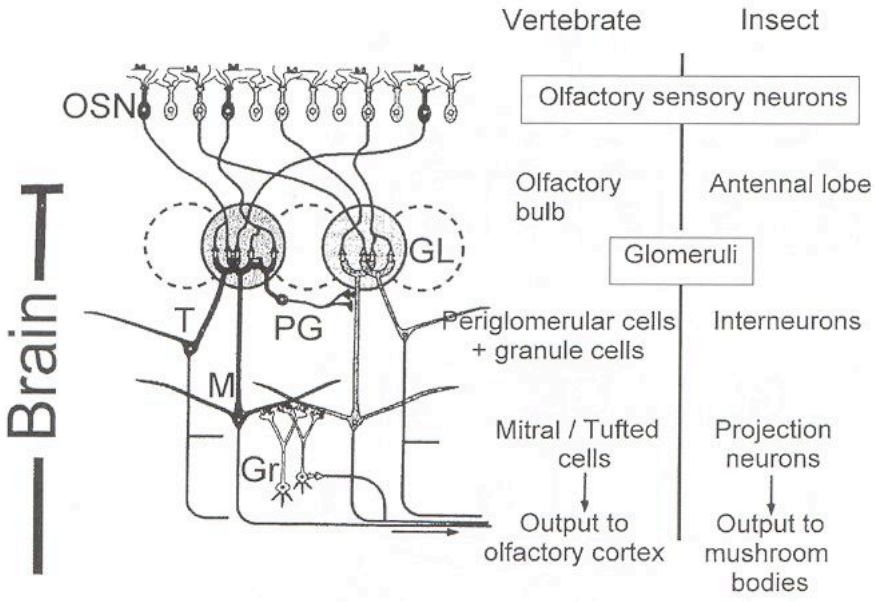


Figure 3.5: Synaptic organisation of the mammalian olfactory bulb and the insect antennal lobe. On the left, the diagram shows two glomeruli (GL) receiving inputs from two types of olfactory sensory neuron (OSN) expressing two different types of odorant receptors. Mitral cells (M) and tufted cells (T) are outputs neurons, and granule cells (Gr) and periglomerular cells (PG) are local interneurons. On the right, the equivalent terms are given for the analogous cells and structures in insects. Figure and caption from Wyatt [105] [pp172].

Qualitative and Quantitative Response to Chemical Concentrations

It is very common for animals to be less attracted or even repelled by high concentrations of an odour [63, 62, 85]. Humans experience this phenomenon quite often. Perfumers and flavourists for example have reported that certain odours are unpleasant at high concentration but were pleasant and perceived as different odours at low concentration. "Indole for example is putrid at high concentration and has a floral odour at low concentration" [63, 62]. As I mentioned earlier, it seems that odour concentration is encoded by the number of stimulated glomeruli: the higher the concentration, the more ORNs are activated and the more glomeruli are recruited [35, 3, 4, 85, 63, 62].

Even though there are huge numbers of possible chemical mixtures which can be defined qualitatively or quantitatively, our olfactory system has evolved to recognize and form memories of approximately 10,000 different odours. There are still many open questions on how olfactory systems identify these odours. However, there are certain theories that are supported by experimental results.

3.3 Theories of Olfactory Computation

Olfactory systems seem to compute a distributed and redundant stimulus representation space, where different odours share usually more than one chemical. The main hypothesis is that olfactory systems have evolved and appear to perform the following processes in parallel (see Figure 3.6) [64, 65, 77]:

1. the **creation of a large coding space** using spatiotemporal activity patterns of projection neurons (or mitral cells) (see Figure 3.6. a). The large size of the coding space is due to the number of possible spatiotemporal combinations. This results in the decorrelation of the representations so that clusters are spread and the overlap of representations can be reduced. This extended coding space can be used in order to represent many different odours.

2. the **creation of sparse representations** of the coding space by using oscillation cycles to select patterns of the activity of projection neurons at each cycle (see Figure 3.6. b). This creates "snapshots" at each cycle of the spatio-temporal patterns created after the decorrelation (1) and activates just a few specific neurons (Kenyon cells in antennal lobe) to represent a particular odour. In insects, these oscillation cycles are generated due to the dynamics of the antennal lobe, specifically by the inhibition of projection neurons by local neurons.

Therefore, the synchronization of firing between different sensory neurons seems to be very important for odour perception and interpretation. The firing rate and the number of sensory neurons are also important in odour recognition when stronger stimuli increase the frequency of firing of individual sensory neurons but also stimulate a larger number of them.

J. Hopfield and C. Brody [36, 7] created simple neural networks using spiking neurons to simulate olfactory processing. In their system, the recognition of an odour was signalled by spike synchronization of specific subsets of “mitral cells”. In their model, oscillations were not due to internal dynamics of the system but were generated by a subthreshold sinusoidal current injected into the neurons. Therefore, synchrony between neurons was driven by an oscillating subthreshold potential.

Different studies have been performed on the perception of simulated chemicals using artificial neural networks and/or robots.

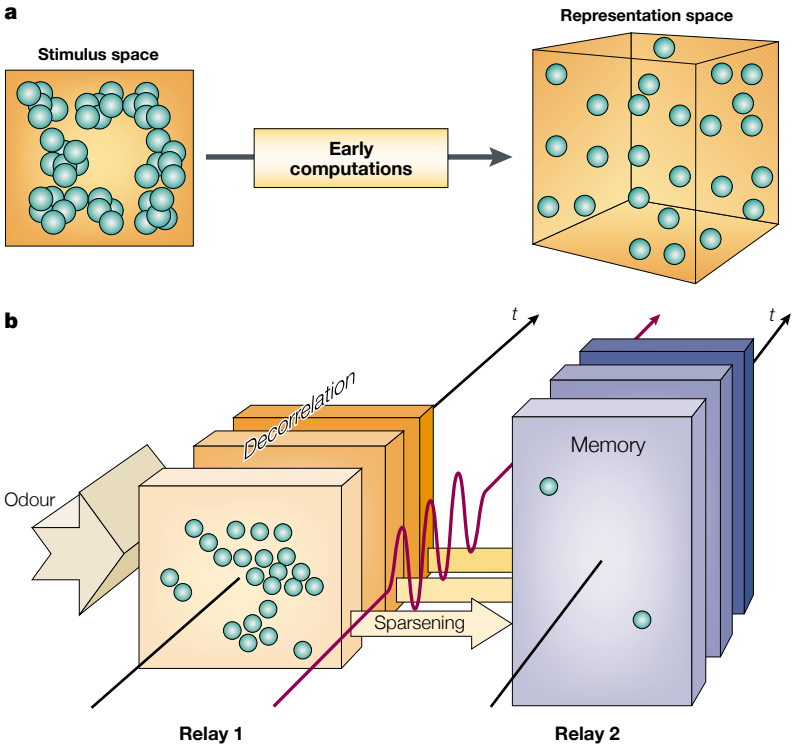


Figure 3.6: Schematic representation of the possible functions of olfactory circuit dynamics and their organization in time and space. **(a)** The early computation that are carried out in the olfactory bulb (OB)/antennal lobe(AL) and their immediate targets could result in both an expansion of the size of the coding space for odours (using spatiotemporal patterning) and a better use of that coding space for the distribution of odour representations. Each sphere in the stimulus space represents a combination of chemical; each sphere in the representation space embodies one (or a family of) spatiotemporal pattern(s). **(b)** As an odour is processed by the first relay (OB or AL), its representation by afferent neurons (pattern of glomerular activation) is given a spatiotemporal format because of dynamics that result from internal connectivity within that circuit. This patterning results in a decorrelation of representations (overlap reduction) over time. At the same time (at least in locust), the spatial patterns of projection neuron activation at each oscillation cycle are compressed into patterns of few active neurons in a large population (relay 2; here the mushroom body). This transformation results in an increase in the specificity of individual neurons' responses, and in a sparsening of representations. The diagrammatic slabs along the time axis represent short time epochs, approximately equivalent to one half of a local field potential oscillation cycle. Each such epoch represents approximately the integration time of neurons in relay 2. So, neurons in relay 2 take short 'snapshots' of the state of relay 1, at times determined by the periodic output of relay1. Figure and caption from [64].

3.4 Olfaction in Artificial Intelligence

A very common task that agents or robots have to perform is taxis, which is a directed movement towards a stimulus source. This can be directed towards a source of light, sound or chemicals. The last one seems to be the most challenging, especially in a real environment where chemicals move in a highly chaotic way. Many AI researchers are studying the behaviour and the brain of insects in order to understand the basis of adaptive behaviours. One of the most impressive animals that can performs chemotaxis are moths. They can detect very low concentrations of pheromones and orient themselves successfully towards the source even if it is situated miles away [105, 51, 50]. They therefore make good candidates as models for agents and real robots [51, 50, 81, 95, 61, 79]. Another popular animal in the AI community that can be attracted or repelled by chemicals is C. *elegans* [107, 108]. The reason for its fame is its very simple neuromuscular system composed of only 302 neurons and 95 muscle cells [21]. Models of the neural circuitry of C. *elegans* have been developed for agents and real robots [8, 59, 17].

3.5 Thoughts on the Evolution of Olfactory Systems

Through evolution, olfactory systems have found a way to perform pattern-recognition efficiently even when the input space is immense and composed of overlapping signals. We can ask ourselves how and why such systems evolved to have the structure I described earlier? There must have been a certain evolutionary pressure for the brain to develop mechanisms used for patterns formation and memorization, but also for recall. Therefore, the creation of easily usable patterns for each odour, must have been the result of selection pressure on animals to be able to learn and recognize a large number of odours. The evolution of communication and the use of pheromones must have been important factors too. Being able to recognise the scent of many different animals that also use a vast amount of different pheromones that can share certain chemical compounds must be of huge importance as an animal should not be attracted by a predator's scent if it wants to survive.

As I mentioned earlier, it seems that one of the key features of olfactory systems to perform such odour discrimination is to have one ORN expressing only one receptor. However, a few animals do not have this feature and one ORN can express more than one receptor (For example, C. *elegans*). Why is it the case? Is it because no selection pressure was present to force them to discriminate as many odours as other animals? Are they still using a primitive form of olfactory system which is on a lower evolutionary scale compared to ours? In order to answer these questions, one might have to study the evolution of C. *elegans* placed in a new environment during thousands of years and see how they adapt...an impossible task. Studying evolution in nature is extremely difficult,

although a few scientists managed to studied the evolution of gene expression in the *Drosophila* olfactory system and found interesting results that can help us to answer these questions.

Animals have evolved and managed to adapt to a changing environment. A good example of such adaptation can be found in drosophilia. "Host plant shifts are particularly important in insect speciation and ecological diversification" [58]. Insect such as *Drosophila* are able to adapt to changes of the chemical cues of host plants. "Changes in chemotactic behavior are an integral part of these shifts as the insect must acquire an attraction and/or overcome the repulsion to the new food source"[58]. The evolution of chemotactic behaviour in *Drosophila* may be due to changes of the gene expression in olfactory sensors [58]. Therefore, it seems that when the environment changes radically and permanently, *Drosophila* found a way to adapt, not by changing the overall structure of its olfactory system or by learning based on synaptic plasticity, but by changing gene expression patterns so that certain ORN could react differently to particular chemical cues. This could give a hint on the evolution of the general structure of olfactory systems and particularly on the reason why in many animals, one ORN expresses only one receptor. There studies on *Drosophila* show that such a system has a great potential to adapt as it does not have to change its wiring completely but only needs to change its sensitivity to certain chemicals reusing existing structures.

Another very interesting fact about the behavioural response of animals to odours is the attraction/aversion by different levels of chemical concentrations. Why should a human be repelled by the scent of an innocent flower when it is present in high concentrations? The reason why certain animals have evolved like this is unclear. However, we can guess that it must have been an evolutionary advantage for our ancestors. Primitive animals might have evolved and were attracted to certain chemicals located in places where food is abundant but avoided high concentration of them which could have been toxic. Another hypothesis is that animals evolved and avoiding high concentrations of chemicals in order to perform odour discrimination more accurately could have been an evolutionary advantage. An animal can supposedly perform much better odour recognition in a place where there are many odours at medium levels rather than several ones at high levels of concentration.

Although these questions are interesting, I had to focus my research on a particular question composed of several defined sub-questions.

3.6 List of Research Questions and Methodology

I decided to focus on the evolution of a simple brain that can control an agent and perceive different chemicals. I believe the bottom-up approach of Artificial Life can be useful to help neuro-scientists to understand complex structures such as olfactory systems. Therefore, my initial goal was to artificially evolve neural controller for agents that need to distinguish different odours and study the similarities (if any) between these controllers and real olfactory systems.

The main research question is:

- how a network of spiking neurons encoding delays can discriminate odours and control an agent?

This question relates to another one:

- how such a neural architecture can encode information (as spatiotemporal patterns using different neural coding strategies) ?

All the reading and preliminary studies I have done led me to related questions of methodology that could link the evolution of olfactory systems architecture and neural encoding, and the evolution of particular behaviour.

- how to evolve a neural controller for agents that can encode temporal delays using spiking neurons and a developmental approach?
- how can a spiking neural network encode information in order to control an agent that is attracted by a low level of concentration but repelled by a high level of the same chemical concentration? Is it necessary to have different types of ORN or is only one type enough?

Chapter 4

The Agent, its Environment and its Response to Pheromones

4.1 Introduction

In this study, I wanted to investigate how information about chemicals (pheromones) could be encoded by sensors in spiking neural networks controlling artificial agents. In order to create chemical sensing agents, I needed to decide which kind of sensory neurons to use. To model the sensory neurons in a biological plausible way and to be able to explore different encoding strategies, I used spiking neurons. One challenge of using a spiking neural network is to decide the coding to use in order to map information received by a sensor that will transform these stimuli into spikes. As I mentioned earlier, different coding strategies can be used:

1. Mapping stimulus intensity to the firing rate of the neuron.
2. Mapping stimulus intensity onto the number of neurons firing at the same time.
3. Mapping stimulus intensity onto the firing delay of the neuron.

In order to use one of these encoding schemes, one needs first to decide how the input current of a sensor should represent its stimulus intensity. The current received by a sensor will be different from the one received by a non-sensory neuron because it will be based on the external stimulus intensity and not the activity of other neurons or sensors. I wanted an agent to be able not only to detect small variations of pheromone concentration but also the whole range of concentrations. Therefore, the agents had to be equipped with sensory neurons that could produce spike trains at different frequencies depending on the pheromone concentration. The ideal case would be to have a linear relationship between the pheromone concentration and the firing rate of the sensory

neuron. Such relationships exist in biological systems. For example in humans, the relationship between the frequency of firing of sensory neurons and pressure on the skin is linear [49]. I tried to find out how to implement such a relationship by carrying out different experiments using different functions for mapping phemonone concentration onto the current injected into the sensory neurons.

In my thesis, all the currents I are current densities in Amperes/Farad and a pheromone concentration refers only to a chemical concentration with no other meanings.

4.2 The Model

4.2.1 Model of Spiking Neuron

As I mentioned earlier, real neural systems have the ability to process information in many ways using different coding strategies mainly based on firing rates of neurons or specific times of spikes. In order to mimic information processing taking place in the brain, many models of neurons with different levels of realism have been created. Models of neurons that simulate the activity of real neurons firing an action potential (spike) are referred to as spiking neurons. It is well known that, compared to the complex and computationally slow Hodgkin and Huxley model, simple spiking models like integrate-and-fire neurons, can run quickly enough and have a more realistic behaviour than firing rate models [25, 26, 30, 45, 46, 53]. This is why more and more researchers are implementing spiking neurons in robots and simulated agents. Therefore, I decided to use a simple model of a spiking neuron that is less computationally expensive but can still generate temporal patterns and where the geometry of the neural network determines its operations in time.

My model of spiking neuron is based on a leaky-integrate-and-fire model [53] which includes synaptic integration and transmission delays. The idea is that a spike sent by a neuron will take some time to arrive at another neuron (see Figure 4.1). This time delay depends on the distance between the sender and the receiver. All the spikes arriving at a neuron are summed to calculate the neuron's input current density (in Amperes per Farad) and membrane potential (in Volts) after every time step ($\Delta t = 0.1\,ms$). Once the membrane potential reaches a certain threshold θ, the neuron will fire and then will be set to 0 for a certain time (refractory period). During this time, the neuron cannot fire another spike even if it is highly stimulated. Many real neurons' membrane potential is around -70mV during resting state. When a neuron fires, its membrane potential will increase rapidly to about 30mV, so the height of a typical spike is approximately 100mV [49]. Without loss of generality, we set the resting potential to 0 and the potential of a spike to 100mV. It is reasonable to set the neuron's threshold at 20mV, the refractory period to 3ms and the membrane time constant τ_m to 50ms [49]. I also decided to set a synaptic time constant τ_s to

2ms. A spike that arrives at a synapse triggers a current given by:

$$I_j(t) = \left(\frac{t - (t_{spike} + delay)}{\tau_s}\right) exp\left(\frac{1 - (t - (t_{spike} + delay))}{\tau_s}\right) \tag{4.1}$$

where $I_j(t)$ is the synaptic input current densisty, t_{spike} corresponds to the time a spike has been sent to the neuron, $delay$ is the time delay in seconds before the spike arrives to the neuron ($delay = coeff_{delay} \times distance$) with $coeff_{delay} = 5 \times 10^{-5}$ (see Figure 4.1).

The change of membrane potential is given by:

$$\frac{dV}{dt} = -\frac{V}{\tau_m} + \sum (I_j W_j) \tag{4.2}$$

where V is the membrane potential, τ_m is the membrane time constant and W_j the synaptic weight.

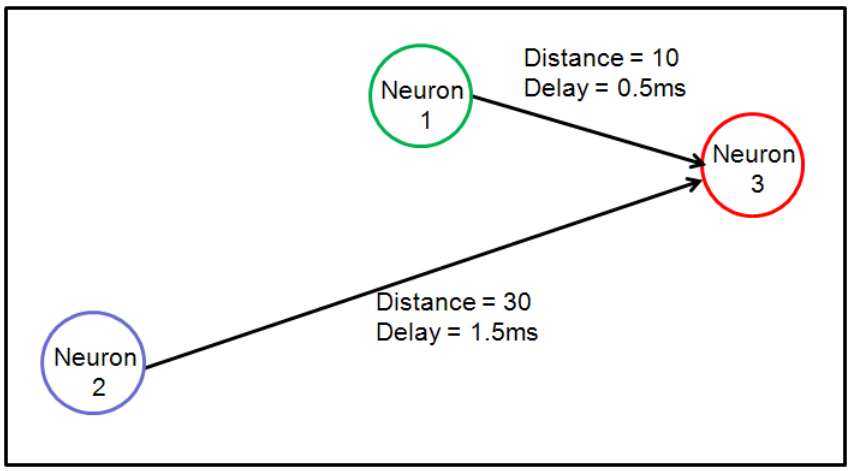


Figure 4.1: Schematic showing output connections from Neuron 1 and 2 to Neuron 3 placed on a 2D space. The distance of the connection between Neuron 1 and Neuron 3 is smaller (10 units) than the connection between Neuron 2 and Neuron 3 (30 units). Therefore, the time delays are different. If Neuron 1 and 2 fire at the same time, the spike sent by Neuron 1 will affect Neuron 3 after $0.5ms$ and the spike sent by Neuron 2 will affect Neuron 3 only after $1.5ms$. Therefore, in this model, transmission delays are encoded by the spatial configuration of the neurons and their connections.

4.2.2 The Agent

For all the experiments I conducted during my PhD, I used a simple model of agent. It had two wheels, one on each side of the agent, providing a differential steering system. Each wheel was controlled by two motor neurons providing forward and backward propulsion. This agent is quite similar to a Braitenberg vehicle or a Khepera robot [6, 25]. The agent also had two antennae placed

on the front of the agent, one orientated on the left and the other one on the right. Each antenna was linked to a sensory neuron. The two antennae were separated widely enough to detect the presence of the chemical gradient. To control the agent, we used a spiking neural network. The sensory and motor neurons placed on the neural substrate form the initial neural network (Figure 4.2). I decided that, in order to move, the agent should be driven by two wheels each controlled by two motor neurons: one to go forward, one to go backward. I created sensors able to detect a chemical gradient. But an agent equipped with such sensors will not move without any stimulus. So I decided for simplicity that an agent should always move forward in the absence of any external input. I performed this by adding a small baseline input current (0.5 A/F) in the motor neurons responsible to go forward. The final velocity of the wheels was calculated by subtracting the firing rate of the motor neurons, responsible for moving the agent forward and backward, running over a certain period of time. The agent was moved by calculating the velocity every 10ms.

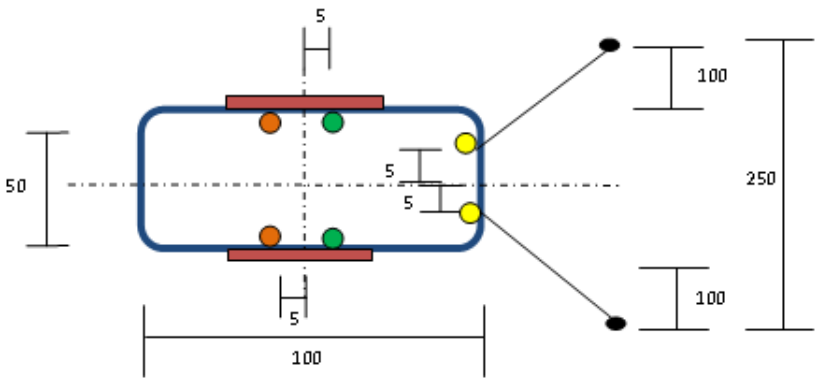


Figure 4.2: An agent equipped with two wheels (red) and two antennae. The sensors (yellow) are linked to the antennae to detect the chemicals (pheromones). The motor neurons in green are responsible to move the agent forward, the others move the agent backward.

In order to move the agent, the velocity V_w (arbitrary unit) of each wheel was calculated using the following equation:

$$V_w = K_v \left(\frac{nb_{forward} - nb_{backward}}{\Delta t_{move}} \right) \tag{4.3}$$

Where K_v is a coefficient, Δt is the time window used to calculate and update the velocity, $nb_{forward}$ is the number of spikes emitted during Δt_{move} by a motor neuron responsible to move the agent forward, and $nb_{backward}$ is the number of spikes emitted during Δt_{move} by a motor neuron responsible to move the agent backward. In these experiments, I used $K_v = 0.3$, and $\Delta t_{move} = 10\,ms$.

4.2.3 The Environment

The world I created has a square shape and could contain agents and chemical concentrations. In artificial life, many simulations have been made using different kind of environments. Usually, chemicals or odours are simulated in a grid world like in NetLogo [100]. However, I modelled a 2D continuous world where chemicals are placed in circular areas as shown in Figure 4.3 similar to [1, 76] where they also modeled lights as circles. I decided to use a simple model of chemicals that are not diffused or evaporated. Apart from an example shown in the first part of Chapter 5 which is explicitly mentioned, the concentration is a linear gradient where the maximum value is situated in the middle of the circular chemical concentration. To calculate the value of the concentration at a given point, I used the following equation:

$$c = Max - (K \times dist) \tag{4.4}$$

Where c is the value of the concentration, Max is the maximum value at the center of the concentration, K is a coefficient and $dist$ is the distance between the point of measurement and the center of the concentration. In these experiments, I used $Max = 300$ and $K = 0.3$. Also, c cannot have a negative value ($c \geq 0$).

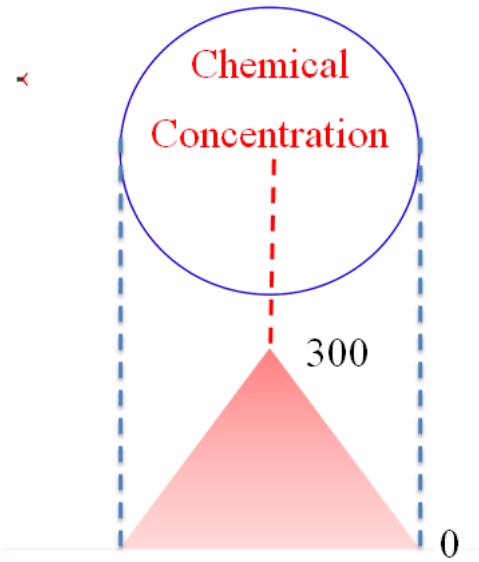


Figure 4.3: An agent (on the left) moving toward a chemical source. The concentration is a linear gradient where the maximum value is situated in the middle

4.3 Experiments

I modelled a sensory neuron as a leaky integrate-and-fire neuron (Figure 4.4) for which the relationship between the current I and firing rate f is well defined and already known (see Figure 4.5 in green). In my experiments, I tried many different functions relating the pheromone concentration P to the current I in order to obtain a desired quasi-linear relationship between the pheromone concentration P and the firing rate f of the sensory neuron (like in Figure 4.5). The sensory current I was always a function of the pheromone concentration P. If the membrane potential, which depends on the current I, reaches a certain threshold ϑ the sensory neuron emits a spike. Therefore, the firing rate of the sensory neuron depends on the relation between pheromone concentration and current (Figure 4.4).

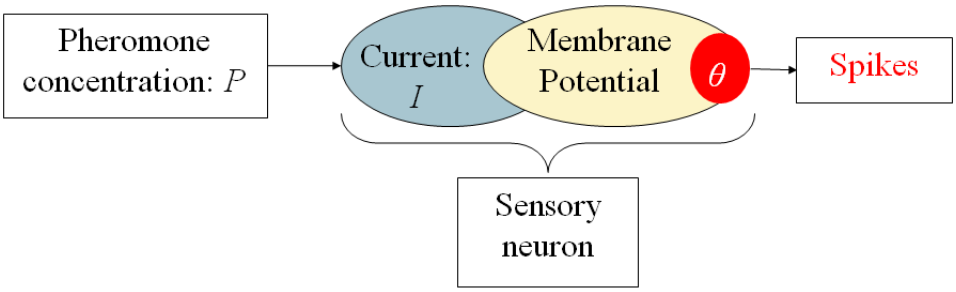


Figure 4.4: Model of a spiking sensory neuron. The pheromone concentration P is first mapped onto current I. As the current I increases, the membrane potential also increases and when it reaches the threshold ϑ, the sensory neuron fires. Therefore, the current I is mapped onto firing rate f.

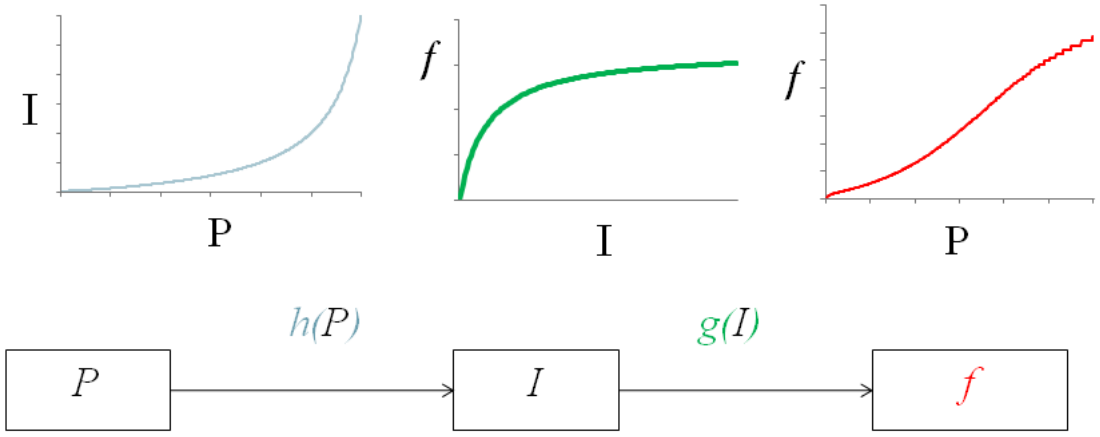


Figure 4.5: Mapping pheromone concentration into spikes. The concentration P is first mapped into current I. Then, the current I is mapped into firing rate f. The resulting relationship between the concentration P and the firing rate f is quasi-linear.

I first set the sensor's input to a concentration of 1 and recorded at what time a spike was emitted in order to determine the frequency (firing rate). I applied the same method to study the firing rate of the sensors over the whole range of pheromone concentration up to some maximum (that I chose to be 300). I did not want the sensory neuron to fire if the concentration was equal to 0 so only the presence of pheromones could stimulate a sensor.

4.3.1 Linear Relationship between Current and Pheromone Concentration

To get a feel for the sorts of firing rates that were possible, I first carried out experiments implementing a simple linear relationship, expressed by Equation (4.5), between the pheromone concentration P and the current I and studied the sensor's firing rate (Fig.4.6).

$$I(p) = Kp \tag{4.5}$$

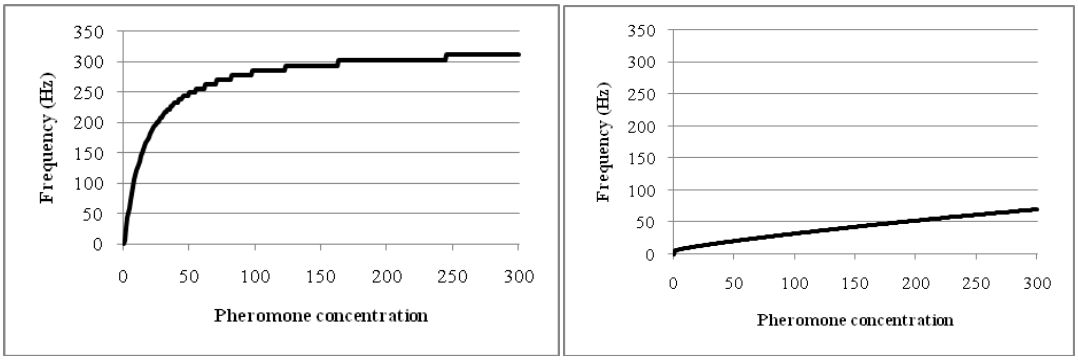


Figure 4.6: Resulting firing rate of sensory neuron using Equation 4.5(left) with $K = 0.41$, and Equation 4.6(right) with $K_1 = 0.41$ and $K_2 = 0.0053$. The maximum firing rate of a neuron is around 300 Hertz.

After a few experiments using different values for K, I realized that the sensor was saturating (Figure 4.6, left) due to the nature of the sensory neuron (leaky integrate-and-fire [53]). In fact, above a small value of pheromone concentration, the current produced was too high and the sensor fired at its maximum rate. After implementation in the agent, I saw that it was not able to detect the difference between a concentration of 200 and 250 for example.

Then, I tried to use the same equation but with an added baseline current and a much smaller slope (K_2) (Equation 4.6). I made these changes knowing that the sensor responds to a small range of currents with a large bandwidth.

$$I(p) = K_1 + K_2 p \tag{4.6}$$

With this equation, we had a more linear relationship between pheromone concentration and the firing rate of the sensor (Figure 4.6, right) so the agent should have been able to detect smaller variations. Unfortunately, the sensor did not use its whole bandwidth and its resolution was relatively poor. Therefore, another kind of functional relationship had to be tried.

4.3.2 Non-linear Relationship between Current and Pheromone Concentration

Concerning the neurons I was using, I knew the limits of currents and the corresponding firing rate. For every cell (motor neurons, sensors, and interneurons):

$I_{min} \approx 0.4$ ($f \approx 0.6\,Hz$)

$I_{max} \approx 20$ ($f \approx 300\,Hz$)

I also knew that the mean firing rate of a leaky integrate-and-fire neuron is given by [Koch, 1999]:

$$< f >= \frac{1}{t_{th} + t_{ref}} = \frac{1}{t_{ref} - \tau ln\left(1 - \frac{V_{th}}{IR}\right)} \tag{4.7}$$

Where:

- t_{th} is the mean time to reach the threshold value
- V_{th} is the threshold voltage (a spike is emitted if the membrane potential is above this value).
- t_{ref} is the refractory period.
- I is the current
- R is the resistance (constant)
- C is the capacitance (constant)
- $\tau = RC$ (time constant)

Given that the sensory neuron is modelled as a leaky integrate-and-fire neuron, I knew the relationship between current and firing rate $F = g(I)$ (see Figure 4.5in green). And I knew that the current is a function of the pheromone concentration $I = h(p)$. Therefore, I had $F = g(h(p))$ so I rearranged Equation (4.7) to find an equation (4.8) for the current (Figure 4.7, left) that would always produce a linear relationship between the pheromone concentration and the firing rate of the sensory neuron (Figure 4.7, right).

$$I = \frac{V_{th}}{R}\left[\frac{1}{1 - exp\left(\frac{t_{ref}}{\tau} - \frac{1}{<f>\tau}\right)}\right] \tag{4.8}$$

To get a linear relationship between the firing rate and the pheromone concentration, I replaced $< f >$ by P and to make sure the frequency was between 0 and 300, I needed:

- $\frac{V_{th}}{R} = 0.4\, mV/\Omega$
- $t_{ref} = 3/1000 = 0.003\, s$
- $\tau = 1/20 = 0.05\, s$

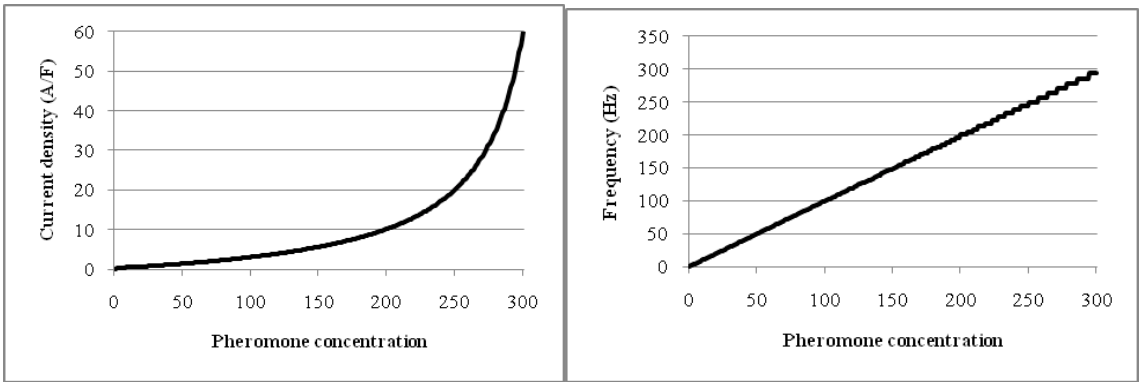


Figure 4.7: Left, current density input to sensory neuron using Equation (4.8). Right, resulting firing rate of sensory neuron. The maximum firing rate of a neuron is around 300 Hertz.

With this equation, an agent is able to detect a small variation in the pheromones concentration using its whole bandwidth. However, I had created Equation (4.8) analytically but I could now use it as a guide to look for a more realistic function that is, one commonly found in biological systems that would describe a similar relationship and have a graph similar to Figure (4.7, left). So I tried to find a biologically plausible function with the right form. There were two candidates, Hill functions and sigmoid functions.

4.3.3 Hill functions

As I mentioned earlier, pheromones and other odours bind to receptor proteins situated in an animal's olfactory sensory neurons. The current generated by the sensory neurons depends on their binding capacity. I first investigated an equation used by biochemists describing the binding of ligand molecules to proteins: a Hill function [91].

$$h(x,k,m) = \frac{x^m}{k^m + x^m} \tag{4.9}$$

Where:

- k is the concentration of molecules when h is equal to 0.5
- m is the Hill coefficient and is considered as an estimate of the number of binding sites of a protein.
- x is the concentration of ligands.

Archibald Hill used this equation in 1910 to describe the binding of oxygen to Hemoglobin. It seems appropriate to use Hill functions to describe the shape of the current produced by the sensor as they are very similar to Equation (4.8).

The first Hill function (4.10) I used was too simple to fit the function (4.8). An example with $m = 1$, $K_1 = 50$ and $K_2 = 100$ is given in Figure 4.8. Once again, I realized empirically that the sensor was saturating quite rapidly.

$$I(p) = K_1 \left(\frac{p^m}{k_2^m + p^m} \right) \tag{4.10}$$

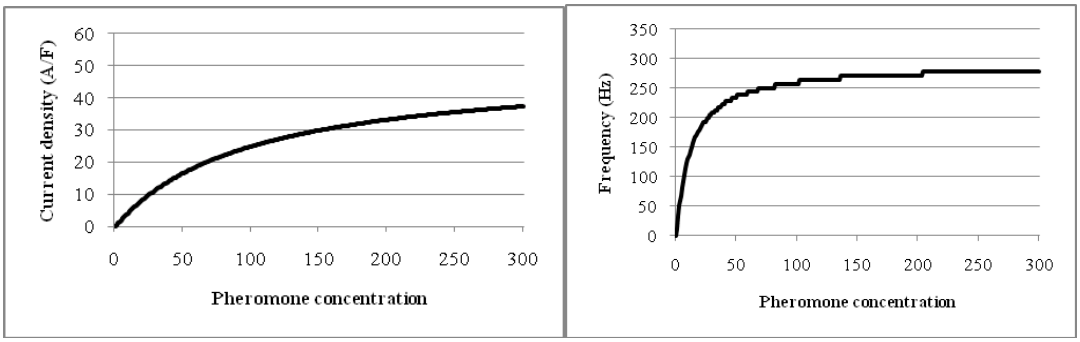


Figure 4.8: Left, current density input to sensory neuron using Equation (4.10). Right, resulting firing rate of sensory neuron.

I used a MATLAB fitting routine to find appropriate constant values for a second Hill function using Equation (4.10), to minimize the difference between the two functions (4.10) and (4.8) (as shown in Figure 4.7, left) in order to have a function that would create a near linear relationship between the pheromone concentration and the firing rate of a sensory neuron like the function (4.8) (Figure 4.9).

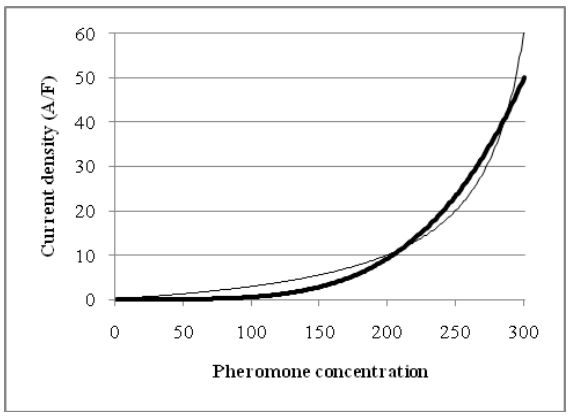


Figure 4.9: Current density input to sensory neuron using Equation (4.10) with $K_1 = 2.38 \times 10^7$, $K_2 = 7104$ and $m = 4.13$. The thin curve is Equation (4.8) and the thick one is Equation (4.10).

Unfortunately, this function was not as good as (4.8). In fact, the sensor could not detect a pheromone concentration of 1. So I decided to add an offset to the function.

$$I(p) = K_1 \left(\frac{p^m}{k_2^m + p^m} \right) + b \tag{4.11}$$

This time, the MATLAB routine found a value for b too high so the sensor could fire even if it did not perceive any pheromones (Figure 4.10, left). So I tried to constrain the value of b to be less than 0.4 (Figure 4.10, right). Unfortunately, the current produced was the same ($= 0.4$) for a large range of small pheromone concentration so the agent could not detect differences of concentration in this range.

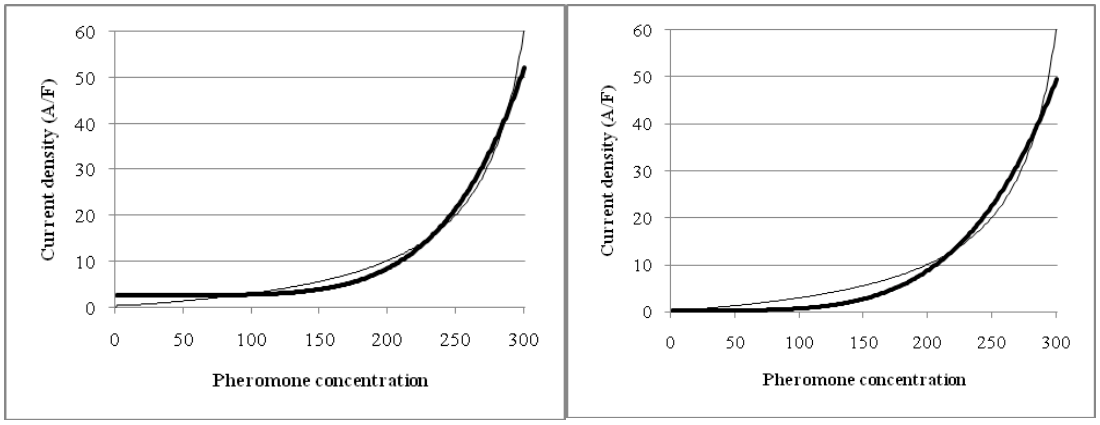


Figure 4.10: Left, current density input to sensory neuron using Equation (4.11) with $K_1 = 2.33 \times 10^6$, $K_2 = 2348$, $m = 5.23$ and $b = 2.65$. The thin curve is Equation (4.8) and the thick one is Equation (4.10). Right, same with $K_1 = 3.45 \times 10^4$, $K_2 = 1378$, $m = 4.297$ and $b = 0.4$.

I concluded that it was difficult to use a Hill function for the sensors' current so that the agents would be able to detect a very small and very high pheromone concentration. Hill functions with coefficients > 1 are sigmoidal so I decided to use a more general sigmoidal function.

4.3.4 Sigmoid function

Due to the fact that the function (4.8) (Figure 4.8, left) resembles the first part of a sigmoid function, I decided to investigate general sigmoid functions.

$$I = K_1 \left(\frac{1}{1 + exp\left(\frac{h-P}{k_2}\right)} \right) \tag{4.12}$$

I also fit this function to (4.8) (Figure 4.11, left). Unfortunately, the sensor could not detect 1 unit of pheromone so I added an offset to the function.

$$I = K_1 \left(\frac{1}{1 + exp\left(\frac{h-P}{k_2}\right)} \right) + b \tag{4.13}$$

I found a function very similar to (4.8) but with an offset too high (Figure 4.11, right). So the sensor was firing even when it did not receive any information.

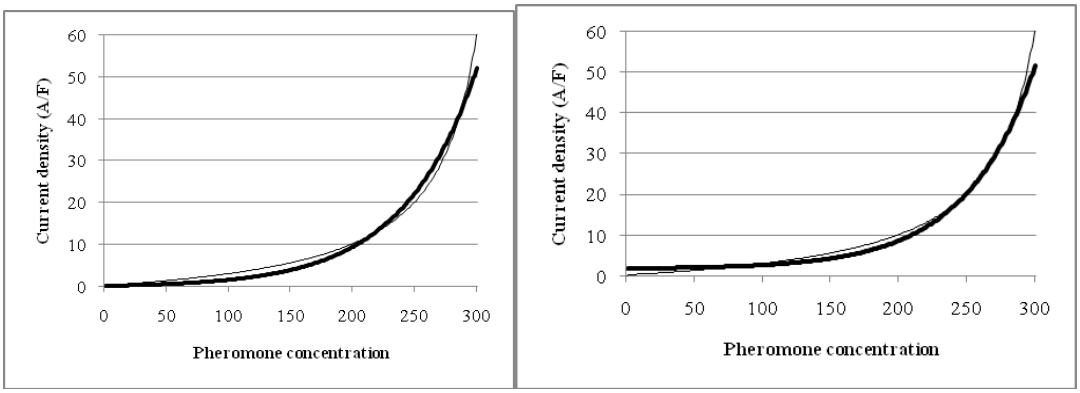


Figure 4.11: Left, current density input to sensory neuron using Equation (4.12) with $K_1 = 2.38 \times 10^8$, $K_2 = 59.35$, and $h = 1210$. The thin curve is Equation (4.8) and the thick one is Equation (4.12). Right, same using Equation (4.13) with $K_1 = 2.7 \times 10^7$, $K_2 = 51$, $h = 973$ and $b = 1.7$.

I therefore constrained b to be less than 0.08 and found a very similar function with a small offset (Figure 4.12, left). After modelling a sensor using this function, I had finally produced a relationship between the pheromone concentration and the sensor's firing rate (Figure 4.12, right) that was, of course, less linear than by using (4.8) but which was biologically plausible and perfectly adequate to allow the agent to detect small and large variation of pheromone concentration.

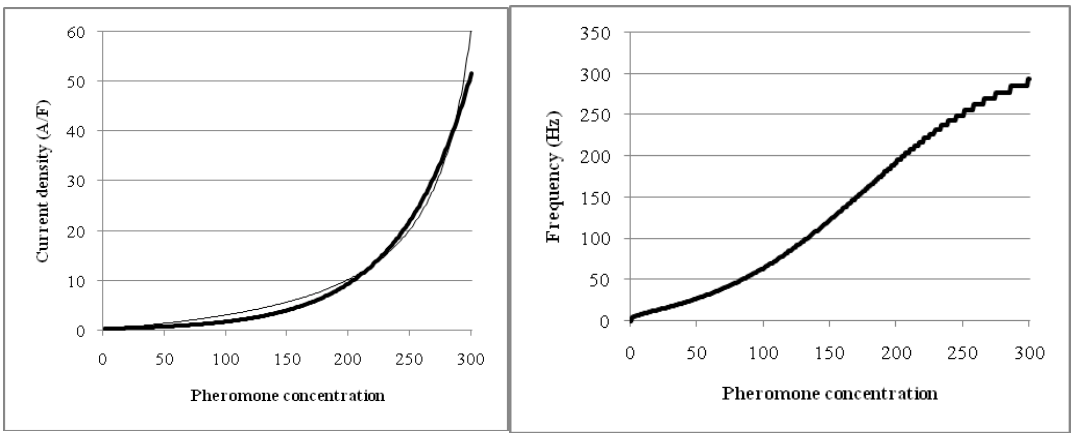


Figure 4.12: Left, Current density input to sensory neuron using Equation (4.13) with $K_1 = 3.9 \times 10^4$, $K_2 = 59$, $h = 691$ and $b = 0.08$. The thin curve is Equation (4.8) and the thick one is Equation (4.13). Right, resulting firing rate of sensory neuron.

This was the final example investigated and was seen to be perfectly adequate for the task

involved.

4.4 Conclusion

The main goal of this study was to create agents able to find and interact to pheromones diffused symmetrically from a point source. In order to achieve this goal, I had to find a model of spiking sensory neuron that could cope with small variations of pheromone concentration and also the whole range of concentrations. I tried many different functions to map the pheromone concentration onto the current of the sensory neuron in order to produce a reasonably linear relationship between the concentration and the firing rate of the sensor. After unsuccessful trials using linear currents, I created an equation that would by definition achieve this task and used it as a model to help me find a similar function that is also used in biology. I concluded that by using a biologically plausible sigmoid function in my model to map pheromone concentration to current, I could produce agents able to detect the whole range of pheromone concentration as well as small variations. The sensory neurons used in my model are able to encode the stimulus intensity into appropriate firing rates (see Figure 4.5) and like real sensory neurons, they saturate when the stimulus intensity reaches a certain value (a concentration of 300 in this case). Moreover, using this model of sensory neurons, I managed to create an agent capable of chemotaxis.

Chapter 5

Study on Neural Coding and Noise in Spiking Neural Controllers.

5.1 Introduction

Animals are able to detect and react to chemicals (odours, pheromones...) present in the environment. The key sense to detect these chemical cues is smell rather than taste[105]. Almost all animals have a similar olfactory system including olfactory sensory neurons (OSN) that are exposed to the outside world and linked directly to the brain. Pheromones and other odour molecules present in the environment are converted into signals in the brain by first binding to the olfactory receptor protein situated in the cell membrane of the OSN. Spikes are then sent down the axon of the OSN [49]. A chemical blend is composed of many molecules that can be detected with tuned odour receptors and therefore, activates a large range of olfactory sensory neurons. Odours are coded by which neurons emit spikes and also by the firing patterns of those neurons sending spikes to others during and after the stimulus. In many vertebrates and insects, oscillations of the neural activity have been recorded in the olfactory systems [105]. Therefore, the synchronization of firing between different sensory neurons seems to be very important for odour perception and interpretation. The firing rate and the number of sensory neurons are also important in odour recognition when stronger stimuli increase the frequency of firing of individual sensory neurons but also stimulate a larger number of them. Different studies have been done on the perception of simulated chemicals using artificial neural networks where neural synchronization occurs [7, 36, 38] and also using robots [50, 61, 79, 81, 95]. I was interested in studying the perception and the behaviour of an agent in response to changes of its environment.

The primary research question of this work was how two encoding strategies can be used to integrate sensory information in order to control a simulated agent. To the best of my knowledge,

no neural architecture, controlling a simulated agent, had been created that encodes the sensory information onto both the firing rate and the synchronization of firing (temporal coincidence of incoming spikes) depending on the environment. As the interaction between the two encoding strategies is complex, I decided to create a simple architecture using a spiking neural network. This model could encode the sensory information onto both the firing rate and the synchronization of firing depending on the environment. As will be seen, the neural network controlled the agent by encoding the sensory information onto temporal coincidences when in a low concentration environment, and could use firing rates when in a high concentration.

5.2 Experiments on Temporal Coincidence and Firing Rate Encoding Strategies

The environment I used for these experiments contained either one or two chemicals denoted by A and B. Each chemical source had a circular shape and the same fixed value all over its surface rather than the linear gradient specified in Chapter 4.

The neural controller was based on a Braitenberg vehicle (anger behaviour) [6] where an agent moves faster toward a stimulus when it detects it (Figure 5.2). My hypothesis was that by using this architecture, the sensory neurons would need to encode the sensory information onto the firing rates, and also onto temporal coincidences between spikes sent by sensors. To verify this hypothesis, I performed three series of tests to study the effect of the starting positions, the sensory delays and the value of the concentrations on the agent's behaviour.

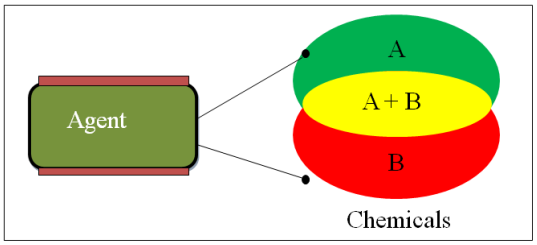


Figure 5.1: An agent and its environment composed of two chemicals A and B.

5.2.1 The Neural Network

As I mentioned earlier, there are three main ways to encode sensory information into spiking neurons based on biological evidence [25, 26, 30, 45, 46, 53]. The most commonly used method consists of mapping the stimulus intensity to the firing rate of the neuron (firing rate encoding). Another method encodes the intensity of the stimulation into the number of spikes sent by different

neurons arriving at a pre-synaptic neuron at the same time (firing synchronization or temporal coincidence encoding). The last main encoding scheme maps the strength of the stimulation in the firing delay of the neuron (delay encoding). As we saw earlier, spatial configuration is an important feature in odour recognition of neurons as is the synchronization of firing between neurons [49, 65, 64, 105]. J. Hopfield and C. Brody [7] created simple neural networks using spiking neurons to simulate an olfactory process. In their system, the recognition of an odour was signalled by spike synchronization in artificial glomeruli.

In my system, the neural network was supposed to detect the blend of two different chemicals and modify the agent's behaviour. I used a model of neural network that allowed me to study synchronization of firing in a simple manner. The neural network could control the agent by encoding the sensory information onto temporal coincidences in a low concentration environment, and firing rates at high concentration. I used the model of a sensory neuron described earlier (Chapter 4) in which the chemical concentration is processed so that a quasi-linear relationship between the concentration and the firing rate of the sensor is produced. In these experiments, each run lasted 600 seconds and the neural network was updated every 0.1ms. Every 10ms, the agent was moved and the sensory inputs updated.

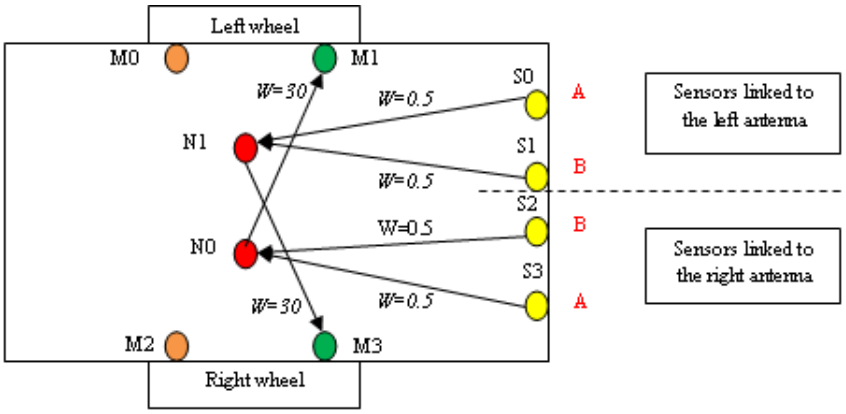


Figure 5.2: Agent's neural controller. The sensors S_0 and S_3 detect the chemical A and the sensors S_1 and S_2 detect the chemical B. The sensory axons' lengths are all similar (delays = 2.5ms). The motor neurons M_1 and M_3 are responsible to move the agent forward. The threshold of the neurons (N_0 and N_1) was set to 4.6 mV. W is the synaptic weight.

5.2.2 Experiment I

The first test was to study the effect of different concentration values of the chemicals on the agent's behaviour. Both concentration values for the chemicals A and B were set to be low. In all the experiments described in this chapter, the concentration range was from 1 to 300. In this instance,

each concentration (A and B) was set to a value of 1 or 2. I tried ten different starting positions and five different settings for the environment: with one chemical A having a concentration of 1, then 2; one chemical B having a concentration of 1, then 2; and finally one chemical A completely overlapping one chemical B.

In this experiments, with a low (1 or 2) concentration of just A or just B, the agent detected the chemical but did not react to it. That is the sensory neurons fired but did not produce a signal sufficient to cause N_0 or N_1 to fire (see more about this in Experiment III). However with a value of 1 for both chemicals A and B, and both present producing a completely overlapping concentration of combined A+B, then the agent reacted and indeed was able to stay inside the combined chemical concentration. This appears to be due to the coincidence of signals emitted by the sensors sensitive to A and B. Figure 5.3 shows an example where an agent starts from the position P2. In this case, the agent was able to stay in the overlapping area.

We can analyse this by looking at Figures 5.3 and 5.4, where we can see that the agent begins by moving horizontally left to right until its right antenna detects the chemicals A and B (T1). At this point, the sensors S_2 and S_3 fire and the temporal coincident arrival of their spikes causes N_0 to fire. M_1 is then stimulated and increases its firing rate turning the left wheel faster than the right one. Soon after this, both antennae detect the chemicals causing also the neuron N_1 to fire so the agent moves straight forward again. At T2, the left antenna of the agent goes outside the overlapping area so the sensors S_0 and S_1 stop firing and therefore, do not stimulate the neuron N_1. The motor neuron M_3 then fires at a lower rate than M_1 resulting in a left turn of the agent to stay inside the area. Finally, from T3, the interaction between the left antenna and the concentration causes the edge-following behaviour. I also recorded the current density and membrane potential of the neuron N_0 during a small interval of time when the agent was inside the blend of chemicals A and B. The input current of the neuron N_0 was increasing when spikes coming from both S_2 and S_3 arrived at the same time. Then, the membrane potential also increased and reached the threshold ϑ (0.0046 Volts) making the neuron N_0 fire. The potential was then set to 0 during the refractory period. As the sensors were synchronized and the delay between them and the neurons were the same, the spikes arrived at the same time to the neuron allowing it to detect them and fire (Figure 5.5, top).

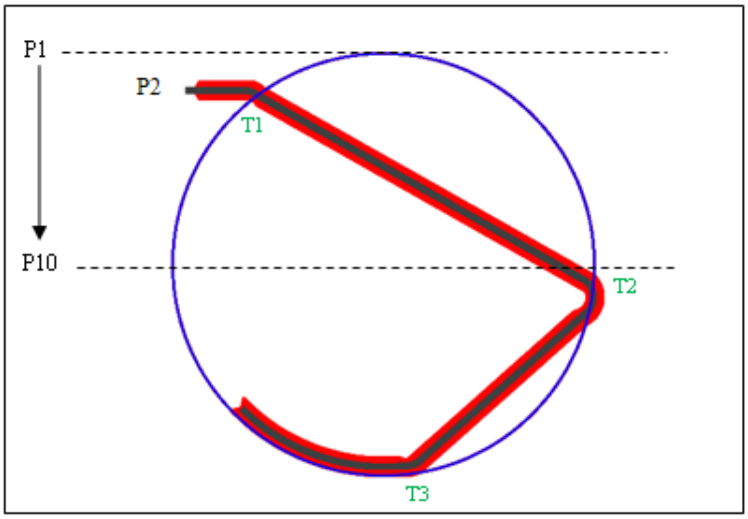


Figure 5.3: Path of the agent starting from the left at P2 just below P1. The circle in the centre represents the two overlapping concentrations of chemicals A and B. P1 to P10 represent 10 different starting positions.

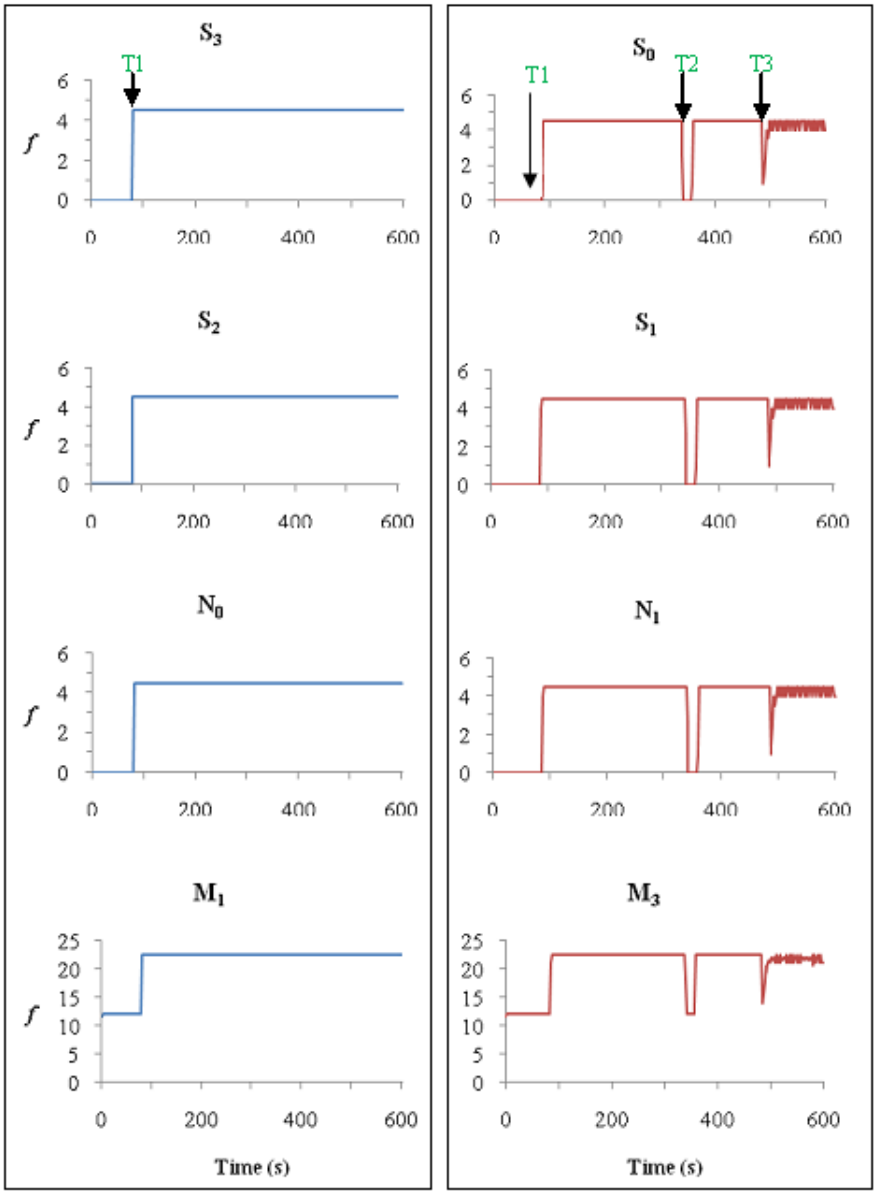


Figure 5.4: Firing rates f (in spikes/s) of the neural network cells recorded every 2s during one run (Experiment shown in Figure 5.3). The motor neurons M_0 and M_1 are not shown here as they do not fire. On the left panel, the sensors detecting the chemicals A (S_3) and B (S_2) from the right antenna activates the neuron N_0 that stimulates the motor neuron M_1 controlling the left wheel to move forward. On the right panel, the sensors detecting the chemicals A (S_0) and B (S_1) from the left antenna activates the neuron N_1 that stimulates the motor neuron M_3 controlling the right wheel to move forward.

5.2.3 Experiment II

The second experiment was to test my hypothesis by modifying the sensory response delays to verify that our architecture necessarily needed to encode the sensory information onto temporal coincidence. I changed the delays by modifying the position of the sensors therefore modifying the length of their axons linked to the neurons. I only changed the delays of the sensors detecting the chemical B (S_1 and S_2). I used the same set up as for the previous experiment. I tried different values of delays (from 1ms to 50ms) and I noticed that a small change (up to 7.5ms) did not modify the agent's behaviour. But a further change in the delays (from 7.5ms) made the agent unable to react to the blend of chemicals A and B so it could not stay inside the concentrations. As in the Experiment I, I recorded the current density and membrane potential of the neuron N_0 during the 0.5s when the agent was inside the chemical blend.

In Figure 5.5, we can see that the current of the neuron N_0 increases when a spike coming from both S_2 and S_3 arrive but as the delay has been changed, the spikes do not arrive at the same time so the current is lower than in Experiment I. Therefore, the neuron's potential increases but never reaches the threshold so the neuron does not fire. This shows that it is the coincidence of spikes coming from the four sensors that was needed to make the agent react. However all of this has been done using low concentrations. Experiment III now considers what happens when the concentrations are increased.

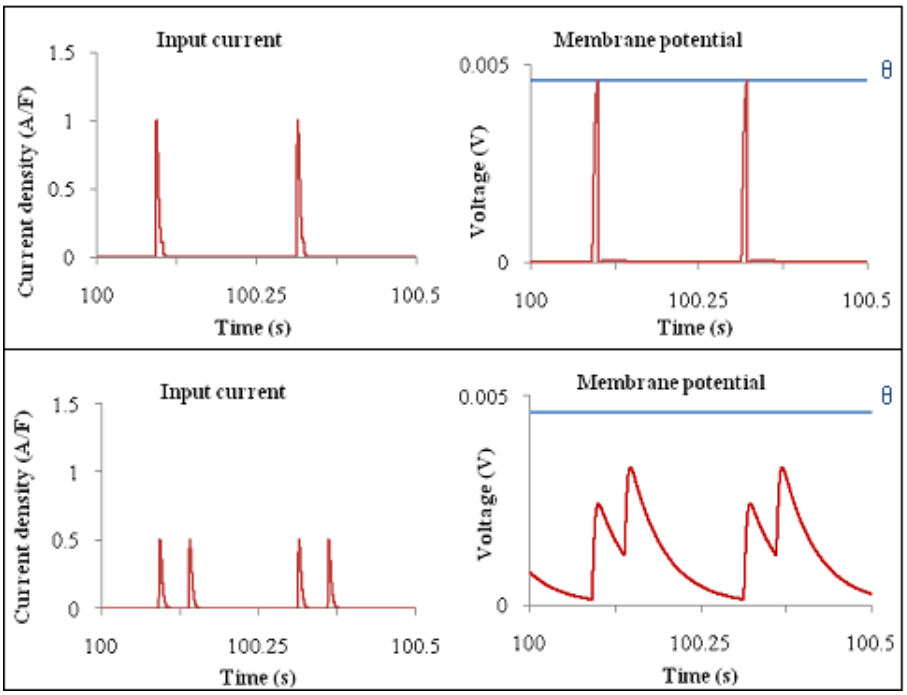


Figure 5.5: Current density (in Amperes per Farad) and membrane potential (in Volts) of the neuron N_0 recorded between 100s and 100.5s. On the top panel (Experiment I), the spikes sent by the sensors arrived at the same time increasing the current density to 1 A/F. The membrane potential was then increased and reached the threshold making the neuron N_0 fire. On the bottom panel (Experiment II), the spikes sent by the sensors were not coincident as the delays were changed to 50ms in this case, so the current was never above 0.5 A/F and therefore, the membrane potential could not reach the threshold to make the neuron N_0 fire.

5.2.4 Experiment III

In order to investigate the use of firing rate encoding, I increased the the concentration values of either A or B, as I already noted that at low concentration A or B on their own did not cause any behavioural change. When the concentration was increased from 1 to above 50, the agent was then able to react to it. So at a much higher concentration, the firing rate method was sufficient to produce a reaction. Therefore, the neural network was seen to react to much lower values of A+B only if both present. I also noted that when using two overlapping chemicals, A and B, as the concentration value increased, modifying the delays had a very little effect and the agent was still able to react to the chemicals. The firing rates were increasing too so the agent was moving faster. In these experiments, the temporal coincidence encoding was not necessary. The sensory information was encoded onto the firing rates of the sensors.

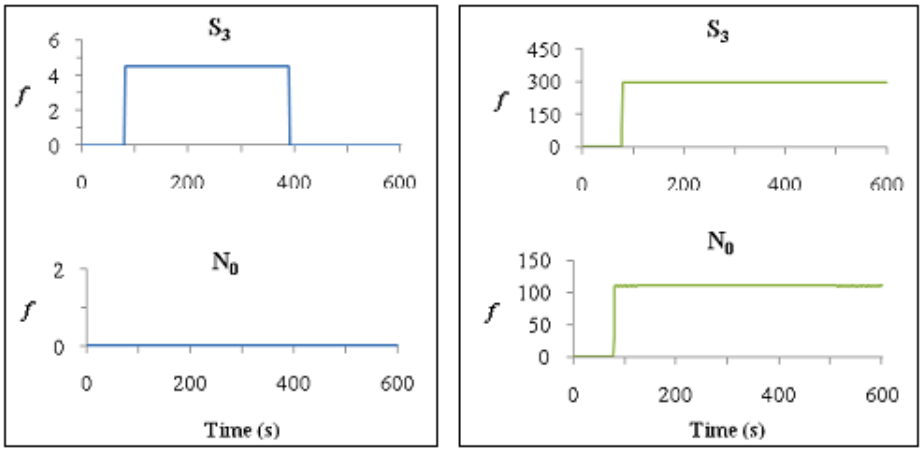


Figure 5.6: Firing rates of the sensor S_3 and neuron N_0 recorded every 2s. On the left panel, as the environment contained a low concentration (=1) of chemical A only, the neuron could not detect it and therefore, the agent did not stay within the chemical source area. On the right panel, the concentration was high (=300) so the neuron could detect it and the agent stayed inside the area.

5.3 Preliminary Conclusion

In this work, I used a simple neural architecture where temporal coincidence and firing rate encoding strategies were both important mechanisms used in different environmental settings. In a low concentration setting, synchronization of spikes sent by the sensors was essential to allow the agent to detect the blend of two chemicals (Figure 5.7). I changed the sensory delays and noticed that the agent was then not able to react to the chemical blend anymore. In these experiments, the agent could not react to only one chemical as only one type of sensors was activated, so temporal coincidence of spikes sent by sensors could not happen. In a high concentration setting, the temporal coincidence between sensors firing was not a necessary condition and the agent was able to stay inside the chemical concentration using just the firing rate encoding strategy even in the presence of just one chemical (Figure 5.8). This model also showed much more sensitivity to the presence of two chemicals than a single chemical. In principle, more than two chemicals can be detected and processed. The architecture presented here also works when the chemical concentration has a linear gradient as described in Chapter 4.

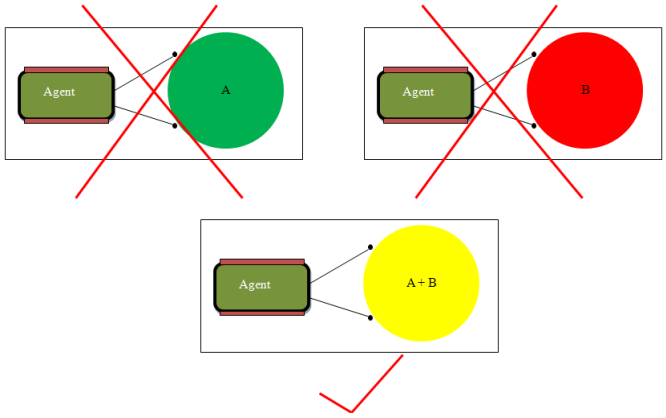


Figure 5.7: At low concentration, the agent was able to detect only the blend of chemicals A+B using temporal coincidence encoding.

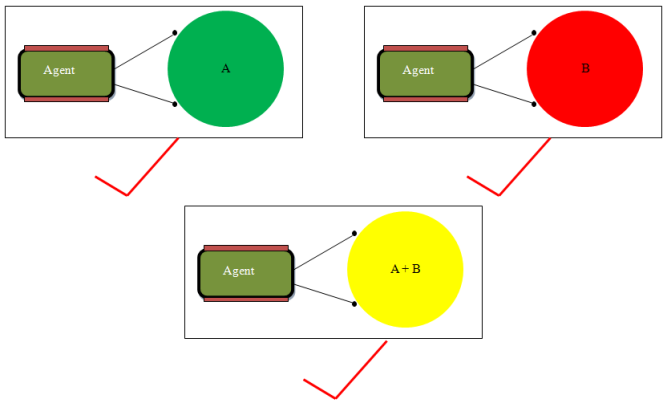


Figure 5.8: At high concentration, the agent was able to detect single chemicals and also the blend of chemicals A+B using firing rate encoding.

5.4 Study on the Role of Neural Noise to Improve the Performance of an Agent.

It is well known that real neuronal systems contain noise [49] which may improve the brain's ability to process information, a phenomenon also called stochastic resonance [34, 73, 74, 99]. Researchers in robotics and artificial life have already implemented simple models of neural noise [15, 27, 47]. In this work, I studied the effect of a more realistic noise model based on a diffusive OU (Ornstein-Uhlenbeck) process [93]. I added this noise in the neural network and studied its effect on the behaviour of the agent. Results suggest a potential function for noise in real biological systems, and highlight that use of features from biological systems can be used to construct better agents.

In the previous experiments, I presented a simple neural architecture where temporal coinci-

dence and firing rate encoding strategies were both important mechanisms used in different environmental settings. In a low concentration setting, synchronization of spikes sent by the sensors was essential to allow the agent to detect the blend of two chemicals. I changed the sensory delays and noticed that the agent was then not able to react to the chemicals anymore. In a high concentration setting, the temporal coincidence between the firing of the sensors was not a necessary condition and the agent was able to stay inside the chemical concentration using just a firing rate encoding strategy. The model showed much more sensitivity to the presence of two chemicals than a single chemical.

To this point, I have used uniform concentrations to simplify the study of the different encoding strategies. However, this model of chemical concentration was not realistic, so I decided to use an environment comprising two non uniform chemical concentration gradients. I tested my architecture in the new environment and noticed that the agent moved outside the concentration when its trajectory was directly along the direction of the gradient (that is, along a radius to the circle) since both of its antennae where instantaneously outside the chemical concentrations. For this reason, I decided to add noise to the neural network. I used a realistic model of noise in the form of a diffusive OU current [93]. This form of colored noise characterizes the subthreshold voltage fluctuations in real neuronal membranes [82]. I added this noise to the total current calculated in Equation (4.2) in each neuron. The noise is described by:

$$\frac{dI_{noise}}{dt} = -\frac{1}{\tau_I}(I - I_0) + \sqrt{D}\xi \tag{5.1}$$

where τ_I denotes the current noise time constant (2ms in our case), I_0 is the mean noise current (0 in our case), $D = \frac{2\sigma^2}{\tau_I}$ is the noise diffusion coefficient, σ is the standard deviation and ξ is a Gaussian white noise (with mean = 0 and standard deviation = 1). The equation 4.2 becomes:

$$\frac{dV}{dt} = -\frac{V}{\tau_m} + \sum (I_j W_j) + I_{noise} \tag{5.2}$$

A discretized version of a diffusive OU current, which is an auto-correlated (integrative) noise, can be written:

$$y_t = a y_{t-1} + e_t \tag{5.3}$$

This reduce to stationary white noise for $a = 0$, stationary first order auto-regressive processes for $0 < a < 1$, brown noise for $a = 1$ (which is identical to the non-stationary random walk) and higher order non-stationary noises for $a > 1$.

5.5 Experiments

I performed a series of tests to find an appropriate level of noise, by modifying σ, in order to have an agent that stays in the gradient chemical blend. I placed the agent at three different positions (Figure 5.9) and tried eight different levels of noise, from no noise $\sigma = 0$ to high noise (Figures 5.10 and 5.11). For each level, I performed 100 runs per position. Each run lasted 300s and I recorded the fitness of an agent during the last 100s. The fitness function was very simple and consisted of the sum of the distance between the agent and the centre of the concentrations measured every time the agent moved. The maximum value of both concentrations was set to 25. By looking at Figures 5.10 and 5.11, we can see that when the agent was starting from P2 or P3, an appropriate level of noise allowed it to stay within the concentration having a higher fitness than an agent without neural noise $\sigma = 0$. We can also note that the level of noise needed to be within a certain range as a low value did not improve the agent's behaviour and a high value disturbed it. The mean fitness value of agents with no noise starting from P3 is not equal to 0 as the agent moved across the chemical. Figure 5.12 shows the improvement afforded by noise in the performance of the agent that arrives at the chemical concentration along a radius (starting from P3).

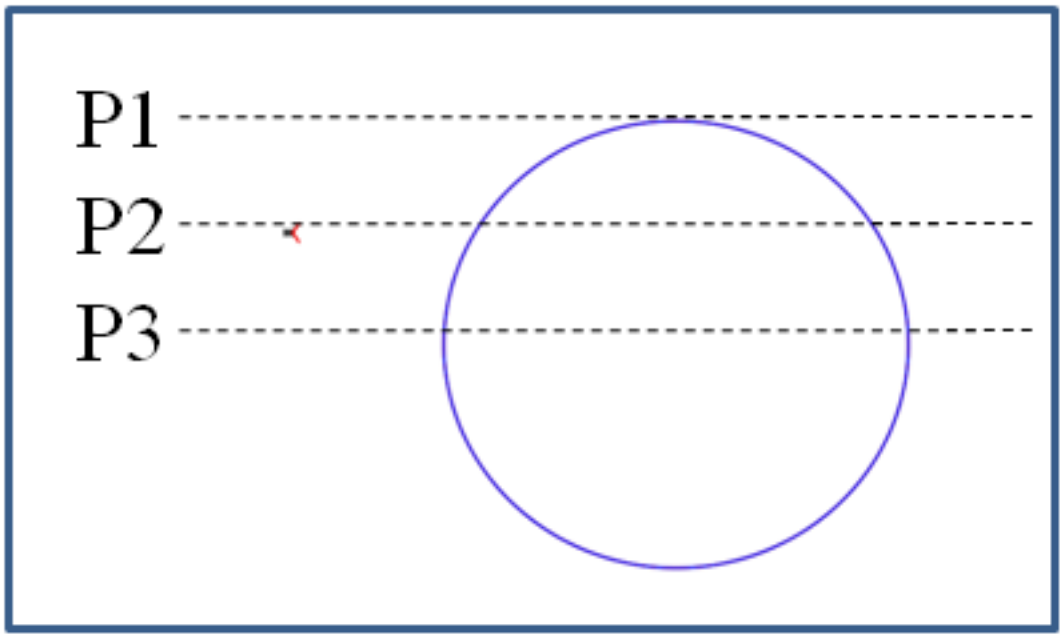


Figure 5.9: Three different starting positions of an agent used for these experiments.

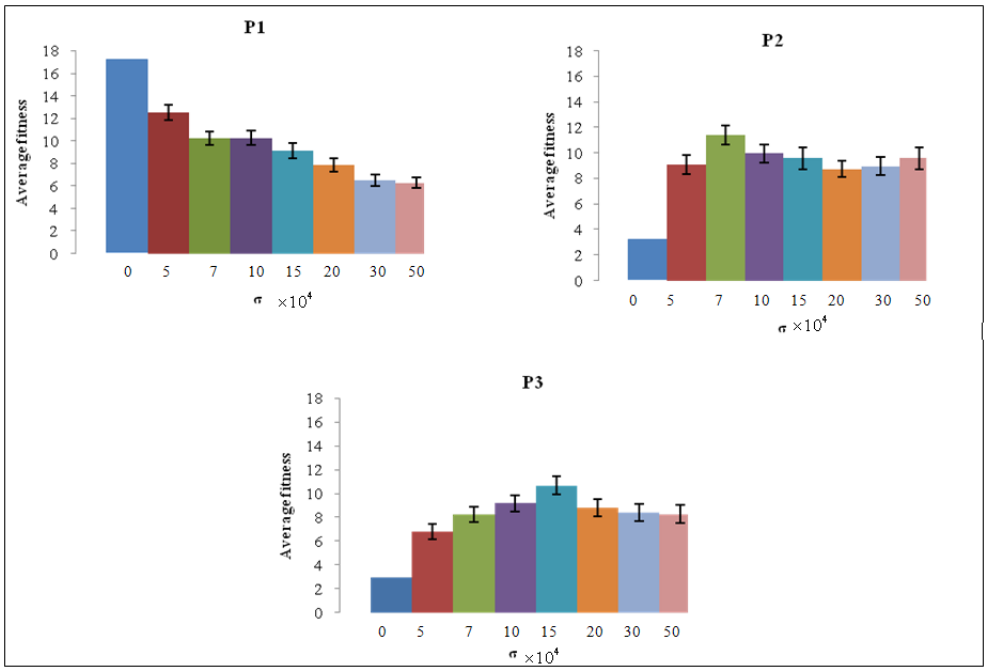


Figure 5.10: Mean fitness values recorded during 100s for an agent starting at the positions P1, P2 and P3 using different levels of noise. The error bars represent standard errors.

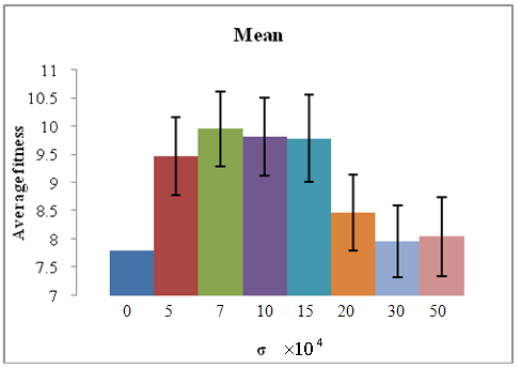


Figure 5.11: Mean of the fitness values displayed in Figure 5.10.

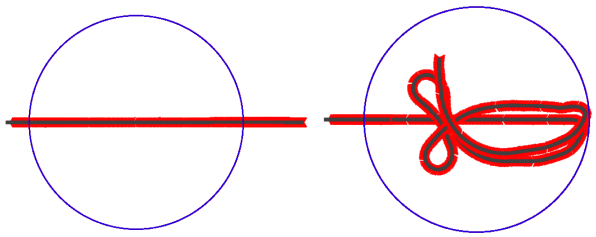


Figure 5.12: Left: path of an agent moving across the blend of chemicals A and B. The agent's neural controller does not have any noise so the agent goes straight as both of its antennae arrived at the same time outside the concentration. Right: path of an agent running over 300s. The agent's neural controller has noise so the agent does not go exactly in a straight line and therefore, can react to the absence of the chemical concentration to stay inside. In both cases, the agent started from the position P3.

5.6 Conclusion

This study was on the effect of noise on the agent's behaviour using the neural architecture used in previous experiments. I used a more complex environment using chemical gradients and a realistic model of neural noise. I found that the overall fitness of the agent was better when a certain amount of noise was added in the neural network. These results suggest that a realistic model of noise can improve an agent's behaviour. This is further evidence that adding biologically realistic features can be beneficial for certain engineering tasks, and suggests a potential function of noise in real biological systems. The effect of biologically realistic noise should be an interesting topic of research in other artificial life scenarios. I need to emphasize the fact that I might have the same results by simulating environmental noise or sensory noise instead. I think it would be interesting to add neural noise in real robotic experiment to study its effects.

Chapter 6

Evolution of Bilateral Symmetrical Spiking Neural Controllers

6.1 Introduction

I the previous experiments, I investigated the relationships between neural coding, noise and the behaviour of an agent controlled by a hand-designed neural network. However, my initial goal was to artificially evolve neural controller for agents that need to distinguish different odours and study the similarities (if any) between these controllers and real olfactory systems. I wanted to study how to evolve a neural controller for agents that can encode temporal delays using spiking neurons and a developmental approach, and also how such a neural network can discriminate odours and control an agent. In order to perform this task, I created different models allowing neural networks to be evolved with different levels of symmetry.

In this chapter, I investigate the importance of bilateral symmetry in artificial neural development, and I introduce three different novel models of a developmental program that grows spiking neural networks on a two-dimensional substrate. Each of these models has different degrees of allowed or enforced symmetry. These developmental programs are evolved, using a genetic algorithm, to allow simulated agents to perform chemotaxis. This chapter begins with a basic introduction on symmetry in nature and how it has been modeled in artificial evolutionary models. Then, I introduce my developmental model, the agent used and the task it had to perform. Further, I describe in more detail our three different models. Then, the simulation and genetic algorithm parameters are presented. This section is followed by the results, the discussion and finally the conclusion.

6.2 Background

6.2.1 Symmetry

For centuries, people have observed and been fascinated by symmetrical patterns found in nature [69, 92, 98]. In our minds, symmetry is often related to something beautiful, well balanced or well proportionate [98]. It has been shown that in many species (even in humans), females prefer males that have symmetrical displays [18]. One possible reason to explain this phenomenon is that symmetry might reflect the high quality of a signaler. Another reason could be that individuals have evolved recognition systems that have common properties and are capable of generalization, and from this could emerge a high sensitivity to symmetries [18]. In living organisms, symmetries arise as a side effect of the creation of axes that will guide cells during development [2, 32, 69, 78, 92, 98, 103]. Cells divide and migrate following gradients that form these axes. They might also create or modify gradients and rearrange themselves to form the most thermodynamically stable pattern [32]. Therefore, it is very likely that cells will be placed symmetrically along different axes to have a system in a state of equilibrium [98]. But due to developmental noise, even the most bilaterally symmetrical animals do not show perfect symmetry. Also, many vertebrates are mainly bilaterally symmetrical about the midline of the body but they have many internal organs that are not bilaterally symmetrical (for example in humans: heart, stomach, spleen...) [2, 32, 78, 103]. Even if the emergence of a bilateral body plan was a key step in evolution, new axes were defined that differentiated head and foot, back and front and left from right, and allowed asymmetrical parts to be created and eventually lead to more complex organisms.

6.2.2 Evolving Neural Controllers

It is usually difficult to create robust and adaptable neural controllers for agents that can perform several different actions. A promising trend is to evolve neural networks using evolutionary computation, (see [16, 23, 24, 57, 75, 67, 80, 83, 89, 106] for reviews). Evolutionary computation approaches allow researchers to design models or systems like neural networks or robots, with little human intervention [25, 23, 24]. With this optimization technique, one can evolve different characteristics of a neural network such as its topology, the synaptic strengths for each connection and/or the learning rule or even evolve the body and the brain of an artificial organism. In the field of Artificial Life, certain researchers try to combine an evolutionary approach with artificial development and learning to generate efficient neural networks having a limited initial knowledge of the architecture needed [23, 24, 56, 5, 67, 75]. An artificial embryogeny (or ontogeny) approach is similar to a natural approach, as an agent or robot can learn and adapt to its surroundings over

its lifetime while the whole population of agents evolves over generations [5]. This approach allows evolved agents or robots to share certain features such as robustness, flexibility, and modularity with certain biological systems [23, 24].

In nature, animals evolved for millions of years and scientists have now evidence that most of them can discriminate many different odours. Moths are a good example of such animals having one of the finest tuned olfactory systems. Even with a relatively small brain, they can find a chemical source situated miles away and diffused at very small quantities (picograms per hour) [51, 105]. Insects like moths have a finely tuned sensory system linked to an olfactory system that can perform odour recognition where timing of spikes, synaptic plasticity and neuromodulation play an important role. Even if moth olfactory systems have been studied extensively, there are still many open questions on how olfactory computation works and how it generates and shapes behaviours. In order to understand olfactory systems, computational model are created to complement experimental results. The bottom-up approach of Artificial Life can be useful to help neuro-scientists to understand complex systems like olfactory systems. Finding an abstract model of development that can generate efficient neural networks is one the most promising goals of evolutionary computation, and its applications in robotics and in neural computation are multiple. Such a model could shed light on the evolution of natural systems like olfactory systems.

There are different approaches in this research area and different ways to encode evolving features into genes. One of the main issues when using evolutionary computation is to choose how to represent the evolvable features as artificial genes. There are three main genetic representation types: direct encoding, developmental encoding and implicit encoding.

- Direct encoding: The easiest manner to evolve a neural network is to use a genome composed of a list, of variable length, of genes that can represent the neurons and/or the links between them. The main disadvantage of this representation is the need to represent each parameter of the network; for example, the threshold of neurons or the weights of the connections. Therefore, the increasing complexity of a network induces an augmentation of the length of the genome. NeuroEvolution of Augmenting Topologies (NEAT) is one of the most famous models using direct encoding [88, 89, 87].

- Developmental encoding (indirect): Another well known genetic representation of artificial neural networks is based on the use of a genome that, when expressed, guides a developmental process leading to the construction of the network [13, 14, 44, 54, 55, 56, 75, 87]. This approach allows a much more compact description of a network than the direct encoding and should produce more complex neural networks.

- Implicit encoding (indirect): The third encoding is based on common knowledge of biological genetic regulatory network (GRN). As in biology, an artificial genome is composed of coding regions and regulatory regions (ignoring non coding regions). Genes can be activated or not by the regulation. In fact, genes regulate the expression, and are regulated by, other genes. Basically, genes act as devices with inputs and outputs terminals (like a neuron for example). Also, a GRN can be used as a developmental encoding scheme [5, 14, 47, 67].

A certain amount of work has been done in evolutionary computation on encoding spatial neural networks by Nolfi [76], Federici [20, 19] and Khan [52], with symmetrical structure using L-systems [10, 11, 37, 70, 71, 86] and grammatical encoding [33, 57]. Stanley also created abstract models generating representations of symmetrical patterns [29, 87]. Some work has also been done where axonal growth has been modeled [48, 94, 52, 84].

Because spatio-temporal patterns are used in real neural systems such as olfactory systems, we need to use spiking neural networks with realistic delays generated by the geometry of the model. So we have created a developmental model generating spiking neural controllers placed in 2D spaces allowing information to be encoded as spatio-temporal patterns, and where bilateral symmetry can be evolved and can be shown to improve both the efficiency of the evolutionary process and the performance of an agent. To the best of our knowledge, this has not been done before.

6.3 My Approach

In this study, I used developmental programs that allowed information to be encoded as spatio-temporal neural activity patterns. I created three new developmental models initially inspired by Kodjabachian and Meyer's SGOCE paradigm [54, 55, 56, 22, 44] and NEAT [29, 88, 89]. By using them, I wanted to see how bilateral symmetry in neural networks could be generated and affect the behavior of an agent. In my models, a developmental program was expressed in a genome and when interpreted, it would create one or more intermediate neurons with one or more connections to make the whole neural network grow. Like in [89, 88], one of the key ideas in our approach is based on complexification. An initial genome is first composed of only one gene creating only one neuron when expressed. Then, during evolution, new genes can be added via mutations creating more neurons and more connections, therefore adding more complexity to the system. Another important concept of this model is targeting [89]. I used a 2D neural substrate where spiking neurons (with synaptic integration and transmission delays) are placed and can grow connections to target locations. Evolution can therefore generate neural networks able to encode external

information as spatio-temporal patterns. Adding in a new neuron can change connectivity for other neurons.

I first created a model where parameters of each neuron were encoded in the genome. I then created two variations of it allowing bilateral symmetrical clones of neurons to be created. The first one allowed the evolution of symmetrical neurons and the second one enforced the symmetry for every neuron. There were no restrictions about the number of neurons that each model could create. All the networks evolved with the three models could have the same number of neurons and connections. I also decided to have neural development performed in two stages: first creating every neuron on the substrate, then creating all the connections. This was inspired by biological systems where neurons first divide, then migrate to a certain location and finally create connections [2, 32, 49]. Some neurons might eventually die but I decided not to model apoptosis in our model to deal with complexity incrementally.

Even though, my models can be considered to be hybrids using an indirect encoding with a guided development and are not strictly developmental by the definition, they are weakly developmental in the sense that they generates the neural network in stages. I defined my models as developmental for the following reasons:

- One genotype can generate different phenotypes depending on the initial neural controller.
- A phenotype can only be created using developmental phases (creation of neurons, then connections) that express the genes.

Therefore, my models use indirect encodings and include a certain level of abstraction of biological development. The approach used in this work is summarized in the following figure (Figure 6.1).

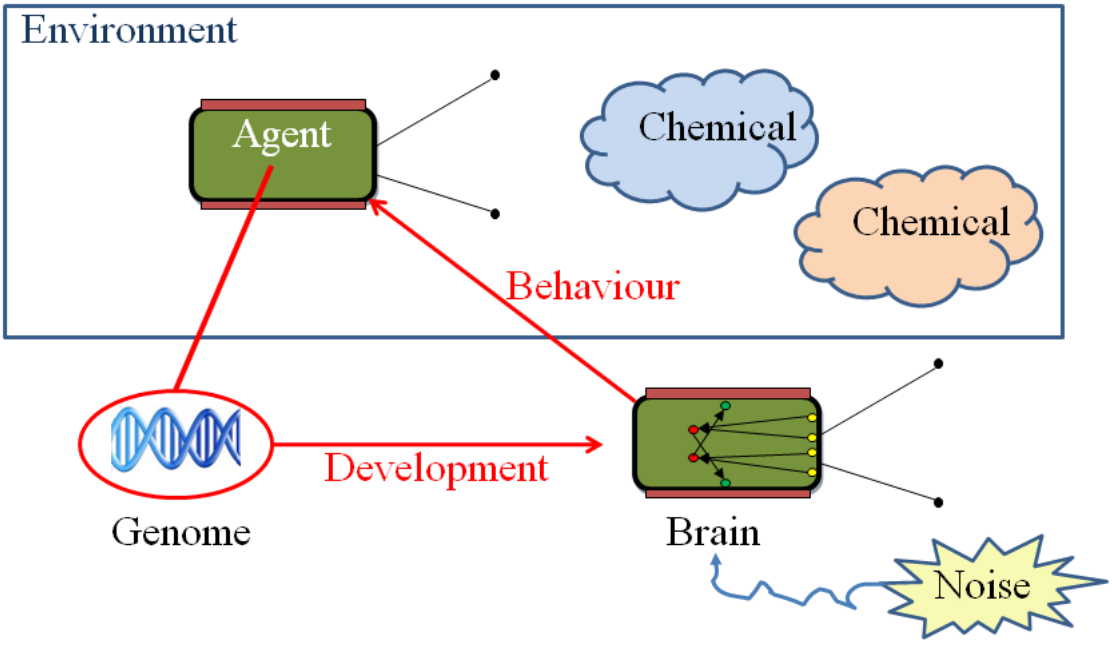


Figure 6.1: Approach used in this work: an agent is situated in an environment where it has to react to chemicals. This agent is controlled by a spiking neural controller. Information on the neural controller is encoded by its genome. Genes are then interpreted during development to generate the controller (brain). Noise is applied to the neurons of the controller. The resulting neural network controls the agent allow it to behave differently depending on the chemicals it encountered. A genetic algorithm is used to evolve the neural controller of the agent.

6.3.1 The Agent

I decided to evolve an agent to perform a simple task which was to stay inside a chemical concentration in a simulated continuous environment. I used the same agent as in the previous experiments controlled by a spiking neural network. The sensory and motor neurons placed on the neural substrate form the initial neural network (Figure 6.2).The complete neural network was created by using a developmental program. In order to increase the speed of the simulations, I decided to both update the neural network and move the agent at the same time, using a time step of 1ms. Having a smaller time step for the neural network update is computationaly expensive (I was using 0.1ms in the previous experiments). Also, by moving at every time step instead of every ten steps, the agent could travel more distance in the same time period.

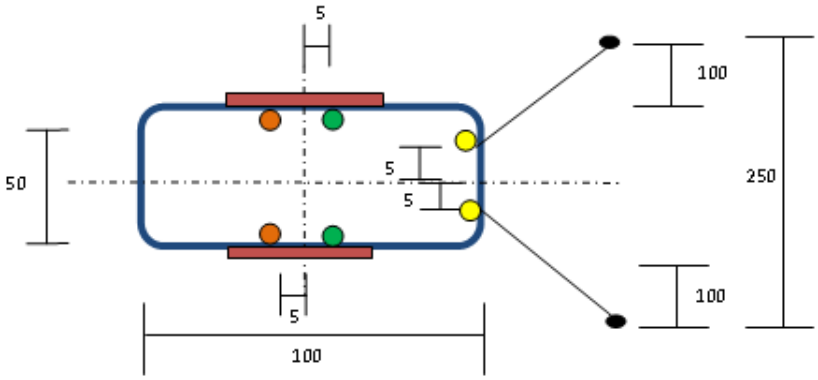


Figure 6.2: An agent equipped with two wheels and two antennae used to detect chemicals.

In order to move the agent, the velocity V (arbitrary unit) of each wheel was calculated using the following equation:

$$\tau_{motor}\frac{dV}{dt} = (V_0 - V) + K_v\left(\delta\left(t - t_f\right) - \delta\left(t - t_b\right)\right) \tag{6.1}$$

Where δ is the Dirac function (pulse) defined by $\delta(x) = 0$ when $x \neq 0$ and $\delta(x) = 1$ when $x = 0$. I decided, for simplicity, that an agent should always move forward in the absence of any external input so I set up the parameters accordingly: $V_0 = 0.5$ is the default velocity (the agent is always moving straight by default), $K_v = 5$ is the speed coefficient, $\tau_{motor} = 0.05$ is the time constant in seconds, t_f is the time when the most recent spike was emitted by the motor neuron responsible for turning the wheel forward, t_b is the time when the most recent spike was emitted by the motor neuron responsible for turning the wheel backward. The agent was moved by calculating the velocity every time step.

6.4 Developmental Model

6.4.1 Mapping Genotype to Phenotype

The initial developmental program I created that constructs the neural network consists of a genome which is an array of modules. A module M must have a gene, which I denote P, encoding the position $(x,\ y)$ of an intermediate neuron, and can have genes encoding possible connections, denoted C (Figure 6.3). The neuron is placed on a 2D Cartesian coordinates system with its origin situated in the centre of the agent. If a new module is created, it will be added to the end of the genome. A module is valid if it is composed of only one P gene but not if it is only composed of C genes. A C gene encodes the different parameters for a connection of a neuron. That includes an angle ϑ and a distance d to determine where it connects, a synaptic strength (w) and a *type*

(afferent or efferent). A neuron can also have connections even if they are not encoded in the module defining its properties. The reason is that other neurons can create efferent and afferent connections to this neuron.

When an agent is created, it only has an initial neural network (Figure 6.2). There are no intermediate neurons, only motor neurons and sensors. If the genome of an agent is composed of at least one module, the complete neural network can be created by executing the developmental program expressed in the genes, reading the genome from the beginning to the end. With only one module, only one intermediate neuron will be created but it can have more than one connection. The neural network is constructed by the developmental program in two steps by reading the genome twice. First, all the neurons are created in the 2D substrate by reading all the P genes. Secondly, all the connections are created by reading all the C genes (Figure 6.3).

When reading a C gene, a target position for a given neuron is defined to determine to which neuron it will be connected to. The target position is given by the angle θ (in radians) and the distance parameter d relative to the neuron. A neuron creates a connection to the closest cell to this target position (Figure 6.4). Self connections are therefore possible. Motor neurons cannot have output connections and sensory neurons cannot have input connections. A target position can be situated outside the substrate. In this case, a connection will still be created linking the closest neuron.

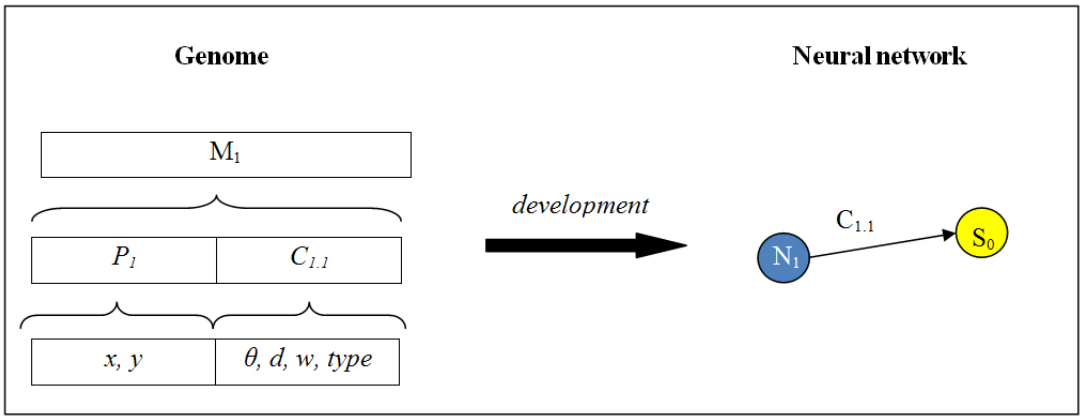


Figure 6.3: Illustration of the mapping used between the genotype (left) and the phenotype (right). In this example, a neuron N_1 and a connection to a sensor S_0 are created on the neural substrate, by executing the genome during the developmental phase. Here, the genome is composed of only one module M_1, therefore creating only one neuron. This module encapsulates a gene P encoding the position of the neuron, and a gene C encoding the parameters used to set up a connection (see Figure 6.4 for the creation of a connection).

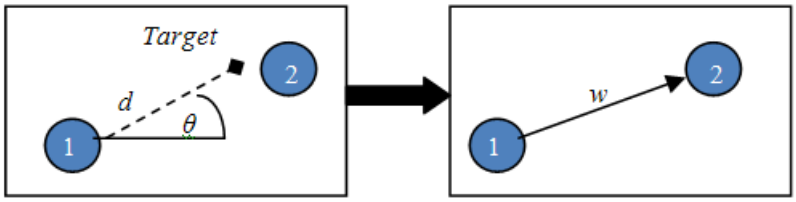


Figure 6.4: Creation of a connection by neuron 1 in two steps. First, neuron 1 places a target point on the substrate depending on the distance d and angle ϑ parameters. Secondly, the closest neuron to the target point gets connected to neuron 1. The type of connection (input or output) depends on the parameter *type* and the synaptic strength (weight) is encoded by the parameter w.

Parameters	Ranges of values
x	$[-50; 50]$
y	$[-25; 25]$
w	$[-15; 15]$
θ	$[0; 2\pi]$
d	$[1; 100]$
type	afferent / efferent

Table 6.1: Ranges of values used for the parameters of the genes. The range of values for the weigths (w) was chosen to have neurons showing realistic dynamics.

6.4.2 Genetic Algorithm

I used a classical genetic algorithm (Figure 6.5) to evolve an agent that could perform chemotaxis. The initial population was composed of 100 agents. Each one of them was equipped with four motor neurons and two sensors composing the initial neural network. Initially, they all had a genome composed of one module encoding one neuron, placed in the middle of the substrate, and one initial connection, having parameters randomly initialized. Therefore, the genome of these agents had one module composed of one P gene and one C gene. Then, each agent was subject to mutations. After mutations, these agents were placed in the initial population and the GA could begin. Once all the agents were evaluated, the agents were ranked by fitness and the ten fittest ones were copied to the next generation. Ninety new individuals were created and added to the next generation‘s population by selecting two parents for each, using a tournament selection of size

five. A new individual was created by cross-over of the two parents. Out of these 90 new agents, twenty were mutated. The genetic algorithm lasted for 1000 generations.

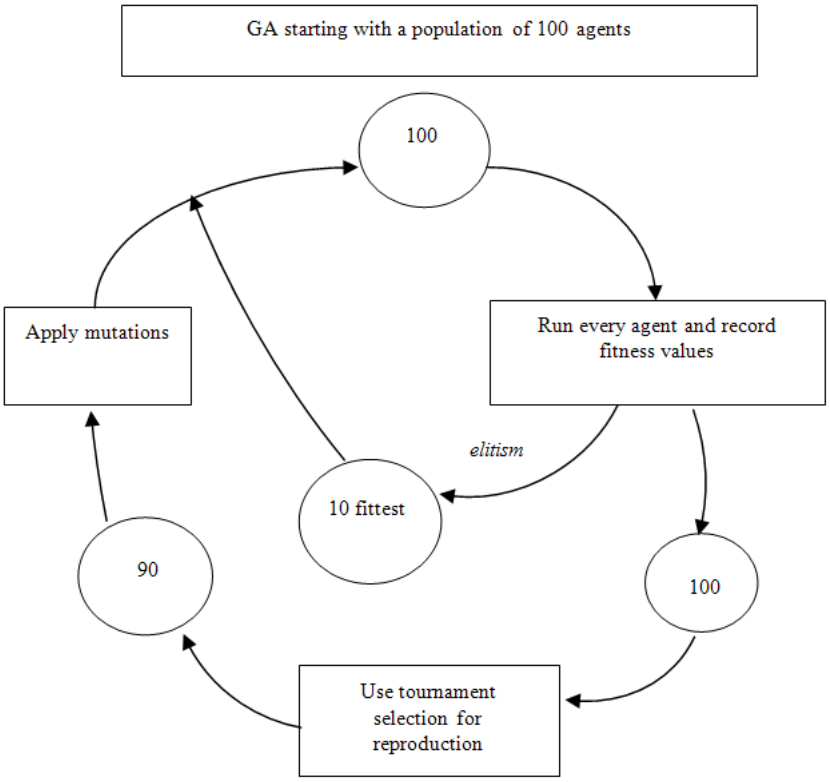


Figure 6.5: Genetic algorithm with parameters. The population is composed of 100 agents. For the first experiment, each agent had two runs of 200 seconds starting from different places (left and right of the chemical concentration). In the second experiment, each agent had only one run of 300s. For both experiments, the fitness rewarded an agent that stayed inside the chemical gradient. Once all the agents were evaluated, the agents were ranked by fitness and the ten fittest ones were copied to the next generation. Ninety new individuals were created and added to the next generation‘s population by selecting two parents for each, using a tournament selection of size five. A new individual was created by cross-over of the two parents. Out of these 90 new agents, twenty were mutated. We ran the GA for 1000 generations.

6.4.2.1 Genetic Operators

The use of the following genetic operators allowed complexification of the genome by adding, modifying or removing new genes.

Mutation

In my model, mutations occur with the same probability independently of the size of the genome. Twenty agents were randomly chosen from the 90 new agents created by the tournament selection and mutated. Three kinds of mutations were used in this GA. A mutation could add or delete neurons, add or delete connections and modify the values of the parameters of the genes. Each mutation was performed within a certain range of values added to the original ones (Table6.2), and these parameters were maintained within certain values defined earlier (Table6.1). For example, if the value of the parameter x of a P gene was 49, and a mutation tried to add 5 to x ($x = 54$), x would be set to its maximum value 50 due to the range of values used. Here is the simple algorithm of the mutation process:

For all twenty agents:

1. Mutate each module:

   a) 5% chances to add a new connection.

   b) 5% chances to remove a randomly selected connection.

   c) Choose randomly one of the following mutations:

   - Pick randomly one connection, choose randomly one parameter and mutate it.
   - Add a random value to parameter x of a P gene.
   - Add a random value to parameter y of a P gene.

2. 5% chance to add a new module (new neuron).

3. 5% chance to remove a randomly selected module (neuron and connections).

When a new module is added to the genome, the new neuron always has one randomly initialized connection. The new neuron is placed randomly in the vicinity of the last neuron created on the substrate (last encoded in the genome). A new connection added is also always randomly initialized.

Parameters	Ranges of values
x	$[-5; 5]$
y	$[-5; 5]$
sym (only for EVO_SYM)	true / false
w	$[-5; 5]$
θ	$[-\pi/4; \pi/4]$
d	$[-2; 2]$
$type$	afferent / efferent

Table 6.2: Ranges of mutations used for the parameters of the genes

Cross-over

Neural selection is applied here by crossing-over modules at the same position. By doing so, each neuron should be able to specialize more quickly during evolution.

Here is an example: Two agents A1 and A2 are selected to create a new agent A3 (Figure 6.6). The maximum number of modules a new agent can have depends on its parents. In this case, agent A1 has five modules and A2 has three of them. Therefore, the new agent A3 will have at maximum five modules (i.e. five neurons). The crossover process will make five selections of modules and at each selection, there is an equal chance of selecting the agent A1 or A2. Therefore, each module of the same position has 50% chances to be selected and copied. If at the fourth selection, for example, the chosen agent is A2, which does not have any more modules at this position, nothing will be added to the genome of agent A3 at this stage. But another module can be copied from A1 if this one is chosen during the fifth selection, and this module, originally from position 5, will become a module of position 4 of agent A3.

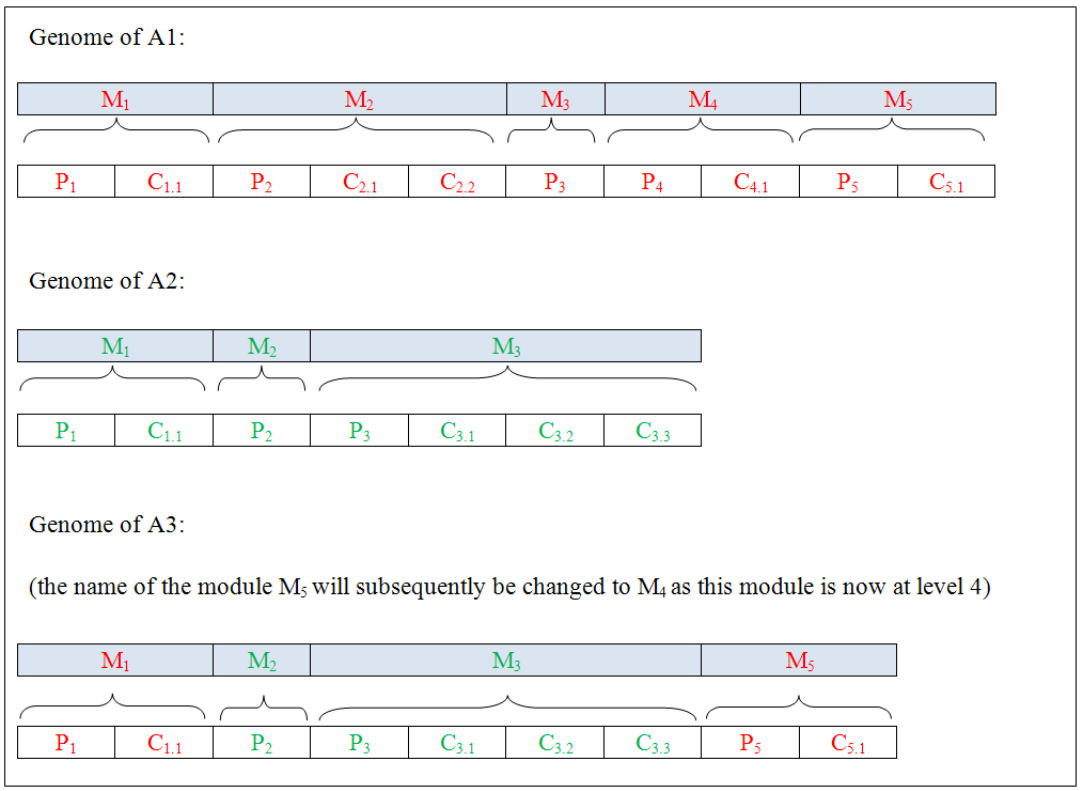


Figure 6.6: Example of a random process creating a new agent A. At each selection, there is an equal chance of picking A1 or A2:
A1 is selected first (first module M_1 is copied), A2 is then selected (second module M_2 is copied), A2 is selected again (third module M_3 is copied), A2 is selected again (but does not have more than 3 modules so nothing happens), A1 is selected (fifth module M_5 is copied), and finally, there are no more modules to copy.

6.4.3 Symmetry

In addition to the original model in which symmetry is not explicitly encoded (denoted NO_SYM), I created two variations of it allowing bilateral symmetrical clones of neurons to be created. The first one allowed the evolution of symmetrical neurons (denoted EVO_SYM) and the second one enforced the symmetry for every neuron (denoted ENF_SYM).

6.4.3.1 Evolvable Symmetry: EVO_SYM

This model is a modification of NO_SYM. The main concept of this model is to introduce genetically encoded bilateral symmetry with respect to the longitudinal axis (rostrocaudal axis) of the agent. The idea is that instead of encoding two neurons that are similar but are positioned on opposite sides of the midline (x-axis), the genome could encode only one neuron but with an

extra evolvable parameter allowing the creation of its symmetrical clone; this allows compressing genetic information. In fact, the initial neural network is symmetrical, and therefore the evolutionary process should be able to use this important embedded feature. This model is based on an abstraction of a gradient that could form the horizontal axis. Compared to NO_SYM, C genes are still the same but P genes have an additional Boolean parameter *sym*. This parameter sym plays an important role. If it is activated (set to true), a clone of the actual neuron will be created and placed symmetrically to the x-axis (Figure 6.7). If the parent neuron is situated on the x-axis, its clone will be created in a close random place around it. The development of the neural network is very similar to NO_SYM. The only difference is that during the first step of development (creation of neurons), each created neuron will have a symmetrical clone if its parameter sym is set to true. A clone of a neuron has its y parameter set to - y and all the connection parameters ϑ set to - ϑ. Therefore, the clone of a neuron is horizontally symmetric and its connections are also symmetric (Figure 6.7). The neural growth is still performed in two steps by reading the genome twice. First, all the neurons (and their possible symmetrical clones) are created in the 2D substrate by reading all the P genes. Secondly, all the connections are created by reading all the C genes.

6.4.3.2 Enforced Symmetry: ENF_SYM

This developmental model is almost the same as NO_SYM. The only difference is the systematic creation of a symmetrical clone for every neuron. Every time a neuron is added to the substrate by executing the genome, a symmetrical clone is also created, as in EVO_SYM (Figure 6.7). But compared to EVO_SYM, ENF_SYM does not encode the possible symmetry in the genome. The creation of symmetrical neurons is an automatic process always occurring during the first step of the development of the neural network.

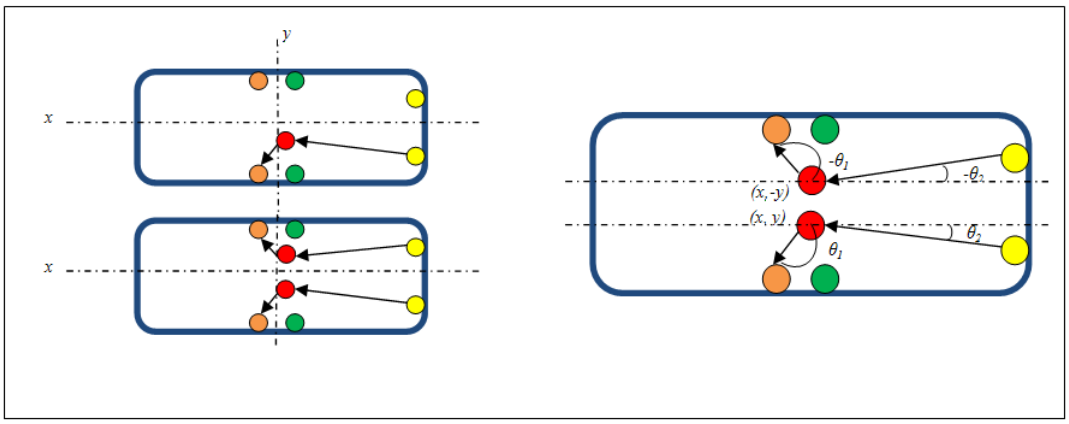


Figure 6.7: Left. At the top, the 2D substrate of an agent with the initial neural network and one intermediate neuron (in red) having two connections is shown. At the bottom, the symmetrical clone has been added. Right. Drawing showing the coordinates and the angle of the connections of the intermediate neuron (bottom) and its symmetrical clone (top).

6.5 Experiments and Results

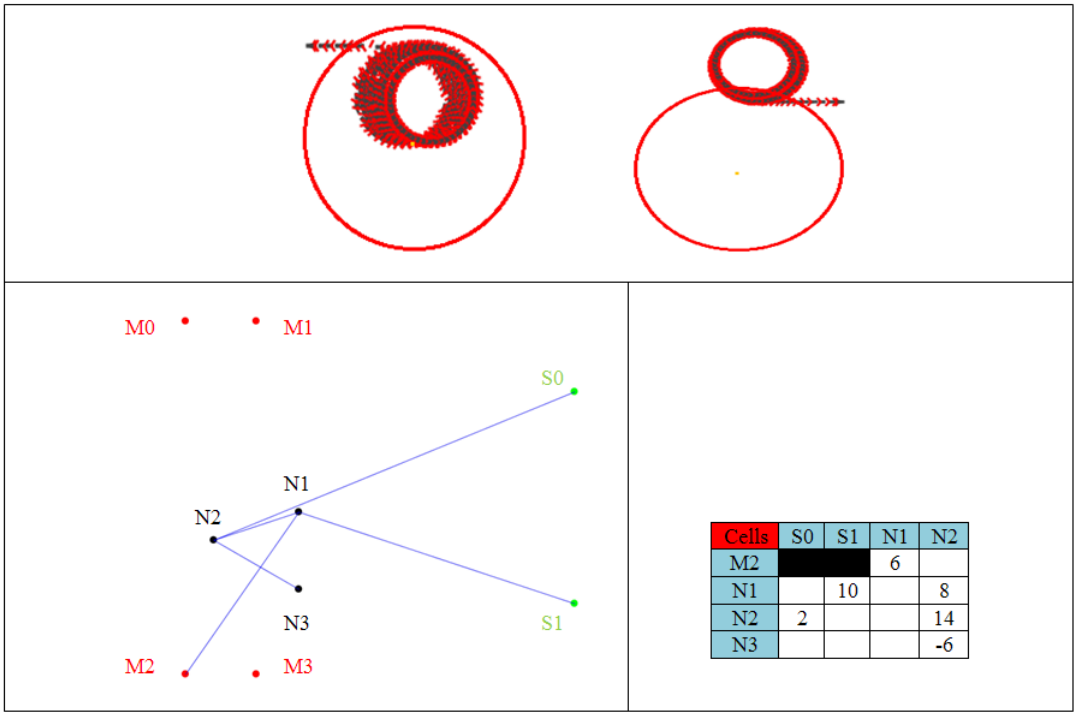
I performed two series of tests. First, I evolved simulated agents that stay in a fixed chemical source. I then evolved agents to stay inside a moving chemical source. In each series, I performed seven GA runs for each developmental model in order to study the importance of symmetry in neural development. For the first series of tests, each agent had two runs of 200 seconds and started from different locations (left and right of the fixed chemical concentration). The fitness function was very simple and consisted of the sum of the inverse distances between the agent and the centre of the concentration during the last 50s of a run. The fitness of an agent was the sum of the fitness values recorded for the two runs. For the second series of tests, I evolved agents able to stay within a moving concentration. One agent and one chemical source were placed in a toroidal world. Compared to the first series, the time of a run was longer (300s). During a run, an agent was always placed at the same place with a random angle of initial movement and the chemical concentration was placed randomly in the world. The concentration was then set moving randomly in the environment. The fitness function was also different and started to be calculated only when the agent was touching the concentration (recording time was initialized at this point). The fitness is the sum of the inverse distances, divided by the recording time. I used a resolution of 1ms (1 time step) for every simulations.

6.5.1 Fixed Chemical Source

I compared the three different developmental models to study the importance of symmetry in neural development. For each model: NO_SYM, EVO_SYM and ENF_SYM, I performed seven runs of the GA described earlier. In each run, I recorded the mean and highest fitness values at each generation.

6.5.1.1 Using NO_SYM

I am showing here the path and the neural network of the fittest agent that evolved using NO_SYM (Figure 6.8).This agent had a very basic behavior which was to perform a clockwise rotation. In fact, it was using only one motor neuron resulting in a very a bad performance (fitness plotted in Figure 6.11).

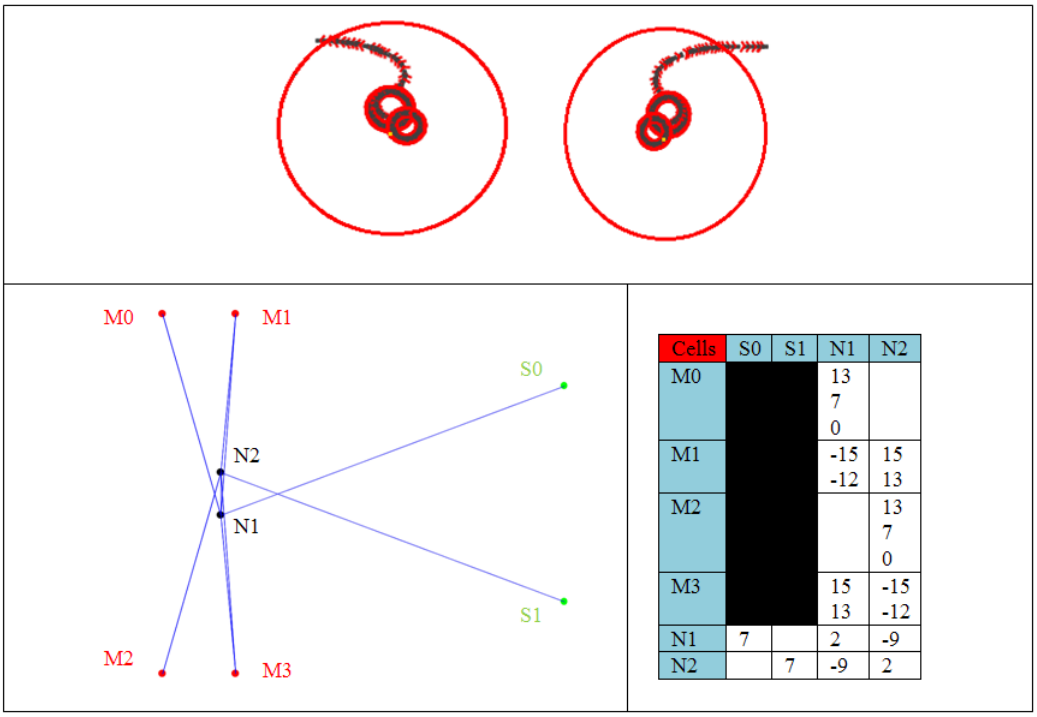


Cells	S0	S1	N1	N2
M2			6	
N1		10		8
N2	2			14
N3				-6

Figure 6.8: Top. Path of the fittest agent starting from two different positions. Left. Neural network of the fittest agent. The motor neurons are in red, the sensory neurons are in green and the intermediate neurons are in black. Right. Connection matrix showing the synaptic strength of a connection linking a cell (top row) and another cell (left column). Notice that there cannot be any connections coming from a motor neuron or going to a sensor as well as connections linking a sensor to another sensor or a motor neuron directly.

6.5.1.2 Using EVO_SYM

I am showing here the path and the neural network of the fittest agent that evolved using EVO_SYM (Figure 6.9). This agent performed well (fitness plotted in Figure 6.11) and used every sensor and motor neuron. It used only two symmetrical neurons where only one neuron was encoded in the genome. The neuron N_1 takes an input from the sensor S_0 and stimulates M_0 and M_3 and inhibits M_1 allowing the agent to turn quickly. N_1 also has a self connection. The neuron N_2 has the same symmetrical connections. I can also notice that both neurons are inhibiting each other. This neural network can be seen as an advanced Braitenberg vehicle [6]. An important remark I can make is that the neurons created more than one connection to the same motor neuron. I assume that it is due to the limit value the weights can have [-15; 15]. Therefore I can see that the system can easily adapt to certain constraints.

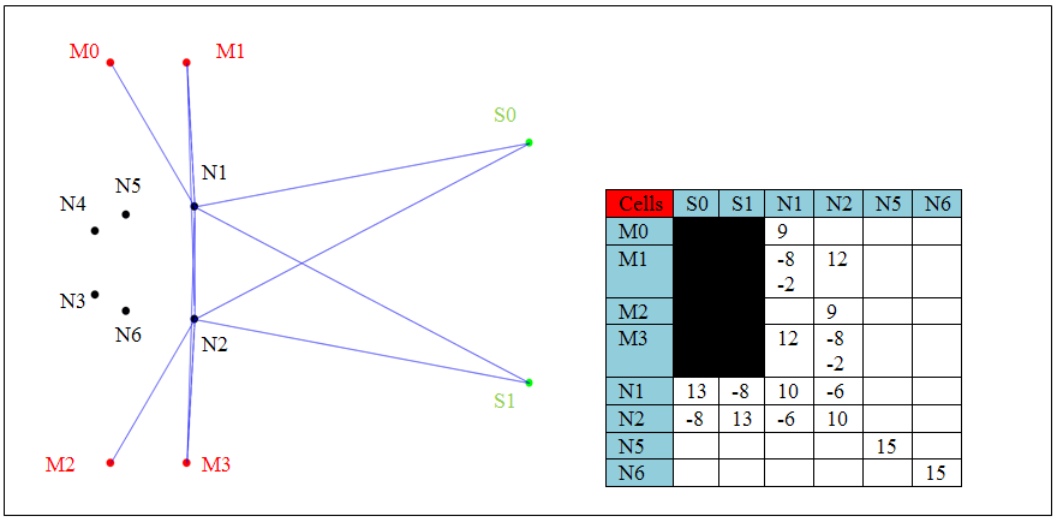


Cells	S0	S1	N1	N2
M0			13 7 0	
M1			-15 -12	15 13
M2				13 7 0
M3			15 13	-15 -12
N1	7		2	-9
N2		7	-9	2

Figure 6.9: Top. Path of the fittest agent starting from two different positions. Left. Neural network of the fittest agent. The motor neurons are in red, the sensory neurons are in green and the intermediate neurons are in black. Right. Connection matrix showing the synaptic strength of a connection linking a cell (top row) and another cell (left column). Notice that there cannot be any connections coming from a motor neuron or going to a sensor as well as connections linking a sensor to another sensor or a motor neuron directly.

6.5.1.3 Using ENF_SYM

I am showing the neural network of the fittest agent that evolved using ENF_SYM (Figure 6.10). This agent also performed well (fitness plotted in Figure 6.11) and used every sensor and motor neuron. It used only two symmetrical neurons where only one neuron was encoded in the genome. Two other neurons were encoded in the genome but they are useless since they have no connections. The neuron N_1 takes an excitatory input from the sensor S_0 and an inhibitory input from S_1. It stimulates M_0 and M_3 and inhibits M_1 allowing the agent to turn quickly. N_1 has a strong self connection. The neuron N_2 has the same symmetrical connections. I can also notice that both neurons are inhibiting each other. The main difference with the evolved neural network using EVO_SYM is the inhibitory connection coming from one of the sensors. The neurons N_3, N_4, N_5, and N_6 can be seen as evolutionary artifacts that could become useful later in time.



Cells	S0	S1	N1	N2	N5	N6
M0			9			
M1			-8 -2	12		
M2				9		
M3			12	-8 -2		
N1	13	-8	10	-6		
N2	-8	13	-6	10		
N5					15	
N6						15

Figure 6.10: Left. Neural network of the fittest agent. The motor neurons are in red, the sensory neurons are in green and the intermediate neurons are in black. Right. Connection matrix showing the synaptic strength of a connection linking a cell (top row) and another cell (left column). Notice that there cannot be any connections coming from a motor neuron or going to a sensor as well as connections linking a sensor to another sensor or a motor neuron directly.

6.5.1.4 Summary of Results

I evolved agents to stay inside a chemical concentration as close as possible from its center. I found that the GAs implementing the developmental models using symmetry (EVO_SYM, ENF_SYM) evolved good neural networks so the agent performed well (Figure 6.11) since it went into the chemical and stayed close to the centre of the concentration. I saw that in all the seven GAs,

EVO_SYM evolved a neural network with symmetrical neurons. In fact, the neural controllers evolved with EVO_SYM or ENF_SYM were very similar. NO_SYM did not manage to evolve an optimal solution as the others did, and had an overall pretty bad performance. Therefore, the first series of tests showed us that without evolvable or enforced symmetry, the system could not evolve and find an optimal solution so the importance of symmetry is clearly shown here (Figure 6.11). I need to emphasize that the initial neural network (placing the sensors and motor neurons on the substrate) is symmetrical as is the agent's body (wheels, antennae). Therefore, the use of symmetry (one module encoding two neurons) is clearly an advantage in evolution. This might not be always the case.

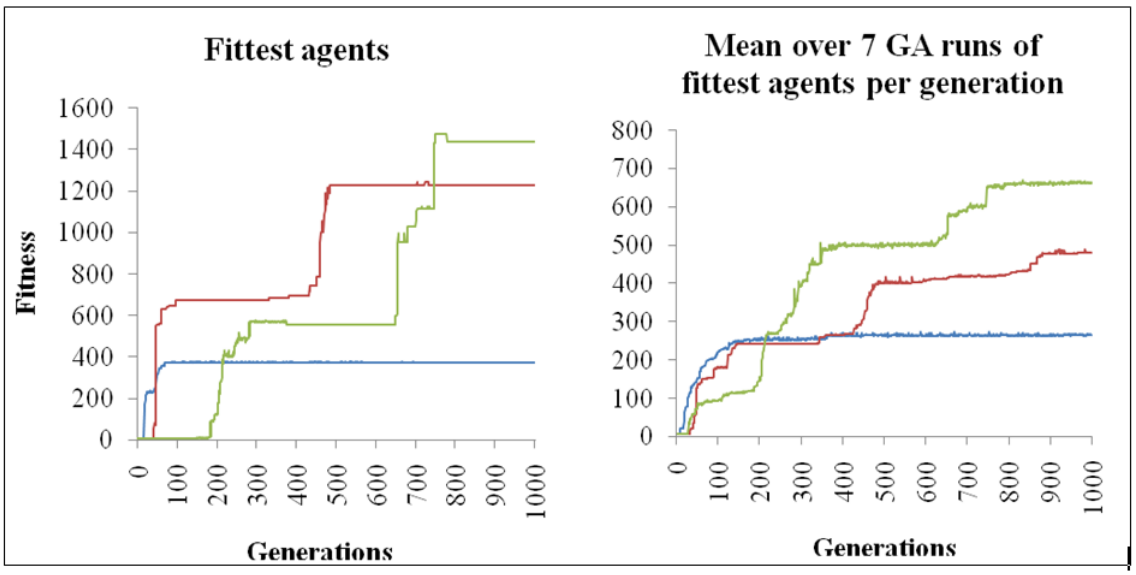


Figure 6.11: Left. Graph showing, for each developmental program used, the highest fitness values per generation of the run which had the fittest agent of all the seven runs. Right. Graph showing, for each developmental program used, the average highest fitness values of the 7 runs at each generation. The colors in the graphs represent: NO_SYM in blue, EVO_SYM in red, and ENF_SYM in green.

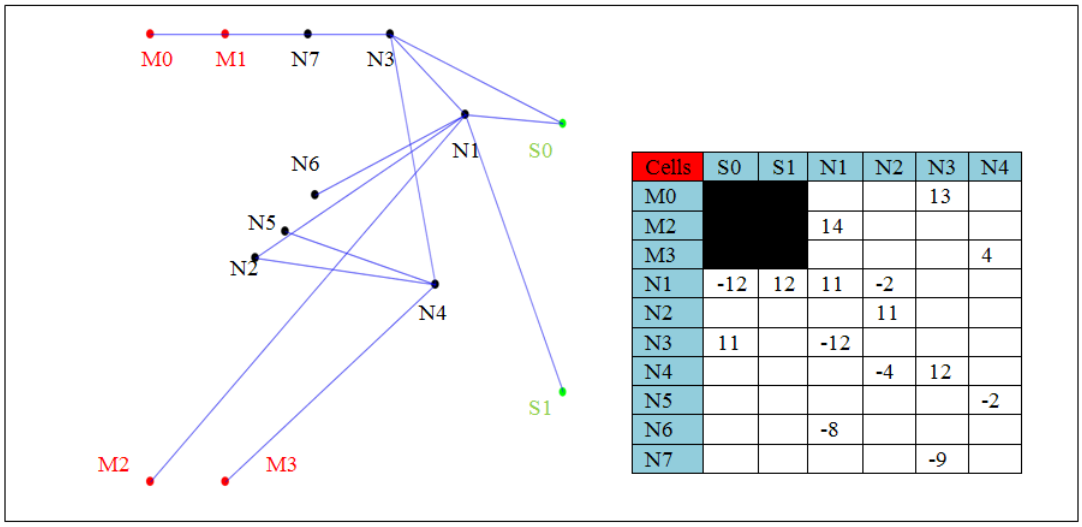
6.5.2 Moving Chemical Source

The next step was to evolve agents able to stay within a chemical source that was moving. In this experiments, the whole circle representing the chemical was moving, and not the gradient within the circle. One agent and one chemical source were placed in a toroidal world. To increase the chance for the agents to encounter the chemical during a run, I decided that the chemical should almost cover the whole surface (90%). As for the previous task, I compared the three different

developmental models to study the importance of symmetry in neural development. For each model: NO_SYM, EVO_SYM and ENF_SYM, I performed seven GA runs. Compared to the GA used earlier, the time of a run was longer (300s). At the start of a run, an agent was always placed in the same position with a random angle of direction. Also the chemical concentration was placed randomly in the world. The concentration was then moved randomly in the environment. The fitness function was also different and started to be calculated only when the agent was first touching the concentration (recording time). The fitness is the sum of the inverse distances, divided by the recording time. This allows us to take into account the time an agent spent inside the concentration once it detected it.

6.5.2.1 Using NO_SYM

Using this model, only one agent managed to be evolved to perform the task and even this one barely succeeded. The fittest agent shown in Figure 6.12 managed to follow the moving concentration only partially (Fitness plotted in Figure 6.15).

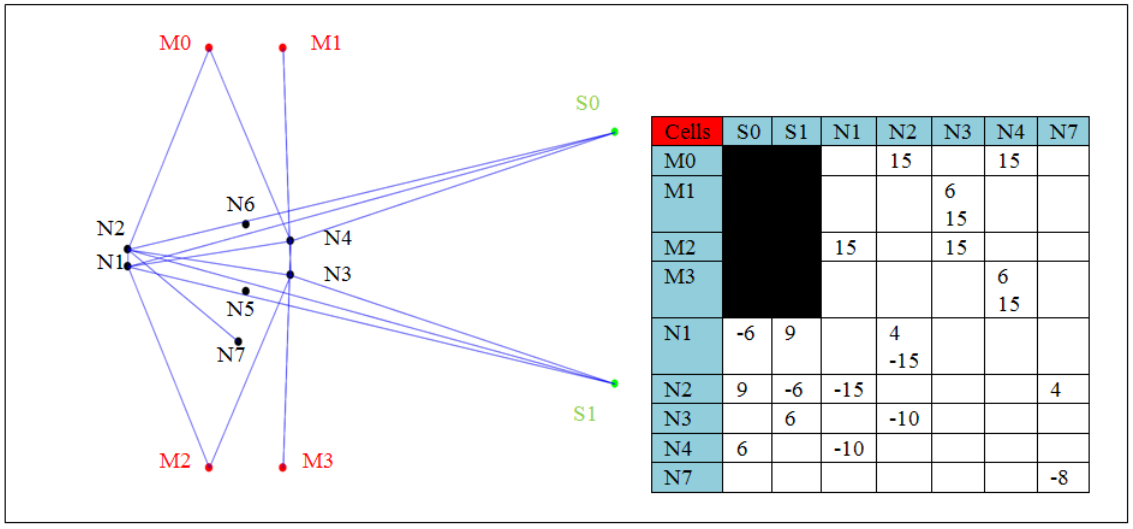


Cells	S0	S1	N1	N2	N3	N4
M0					13	
M2			14			
M3						4
N1	-12	12	11	-2		
N2				11		
N3	11		-12			
N4				-4	12	
N5						-2
N6			-8			
N7					-9	

Figure 6.12: Left. Neural network of the fittest agent. The motor neurons are in red, the sensory neurons are in green and the intermediate neurons are in black. Right. Connection matrix showing the synaptic strength of a connection linking a cell (top row) and another cell (left column). Notice that there cannot be any connections coming from a motor neuron or going to a sensor as well as connections linking a sensor to another sensor or a motor neuron directly.

6.5.2.2 Using EVO_SYM

Once more, EVO_SYM always evolved a neural network with symmetrical neurons in all seven GAs. Figure 6.13 shows the neural controller of the fittest agent evolved using EVO_SYM (fitness plotted in Figure 6.15). Neuron N_1 takes an inhibitory input from sensor S_0 and an excitatory input from S_1. It stimulates M_2 and inhibits N_4 allowing the agent to turn more quickly as N_4 stimulates M_0. Neuron N_2 is symmetrical to N_1 so it has the same symmetrical connections. Both neurons inhibit each other. Neuron N_3 takes input from sensor S_1 and stimulates the motor neurons M_1 and M_2 so the agent can turn quickly. Neuron N_4 is symmetrical to N_3 so it has the same symmetrical connections. I notice that two other symmetrical neurons (N_5 and N_6) and a non symmetrical neuron (N_7) exist but they do not modify the overall neural activity of the controller. In fact N_5 and N_6 do not have any connections, and neuron N_7 does not have any input except an inhibitory connection from itself. This shows that symmetrical neurons (N_1 and N_2) can also have asymmetrical connections. N_5, N_6 and N_7 can be seen as evolutionary artifacts that could become useful in time or disappear. This neural network has more complexity that the one shown in Figure 6.9 . The main differences between the two are the two layers of neurons and inhibitory connections coming from the sensors. Again, I noted that neurons N_3 and N_4 created more than one connection to the motor neurons. I assume that it is due to the limit values the weights can have [-15; 15] (see Table 6.1). Therefore, I again can see that the system can easily adapt to circumvent certain constraints. I ran the previous agent showed in 6.9 to see if it could follow the moving concentration and I saw that it could not perform this task. It seems that a more complex neural network was necessary to perform this second task.

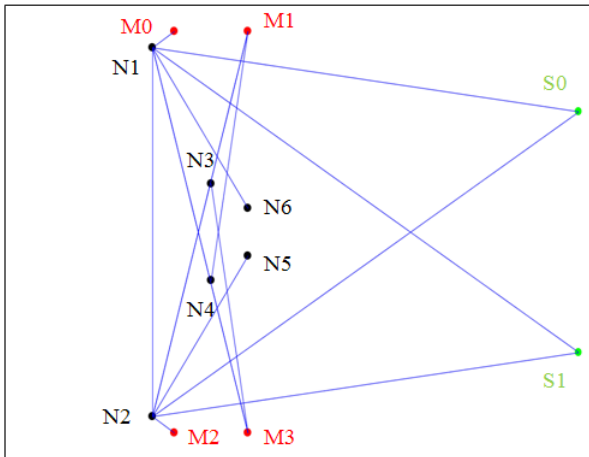


Cells	S0	S1	N1	N2	N3	N4	N7
M0				15		15	
M1					6 15		
M2			15		15		
M3						6 15	
N1	-6	9		4 -15			
N2	9	-6	-15				4
N3		6		-10			
N4	6		-10				
N7							-8

Figure 6.13: Left. Neural network of the fittest agent. The motor neurons are in red, the sensory neurons are in green and the intermediate neurons are in black. Right. Connection matrix showing the synaptic strength of a connection linking a cell (top row) and another cell (left column). Notice that there cannot be any connections coming from a motor neuron or going to a sensor as well as connections linking a sensor to another sensor or a motor neuron directly.

6.5.2.3 Using ENF_SYM

As for the previous experiments, the model using enforced symmetry evolved neural networks that could perform the required task in all the seven GAs. The fittest agent shown here had a good performance and managed to follow the moving concentration with great success (fitness plotted in Figure 6.15).



Cells	S0	S1	N1	N2	N3	N4	N5	N6
M0			8					
M1				15 15		7		
M2				8				
M3			15 15		7			
N1	6	-2	4	-8				-5
N2	-5	6	-8	4			-5	

Figure 6.14: Left. Neural network of the fittest agent. The motor neurons are in red, the sensory neurons are in green and the intermediate neurons are in black. Right. Connection matrix showing the synaptic strength of a connection linking a cell (top row) and another cell (left column). Notice that there cannot be any connections coming from a motor neuron or going to a sensor as well as connections linking a sensor to another sensor or a motor neuron directly.

6.5.2.4 Summary of Results

In each run, I recorded the mean and highest fitness values at each generation. The first graph (Figure 6.15, left) shows the highest fitness values at each generation, of the run which had the fittest agent of all the seven runs. Then, I calculated the average highest fitness values of the seven runs at each generation (Figure 6.15, right). In this series of test, I saw again that the developmental models using bilateral symmetry generated better neural controllers than NO_SYM. Once more, EVO_SYM always evolved a neural network with symmetrical neurons in all the seven GAs.

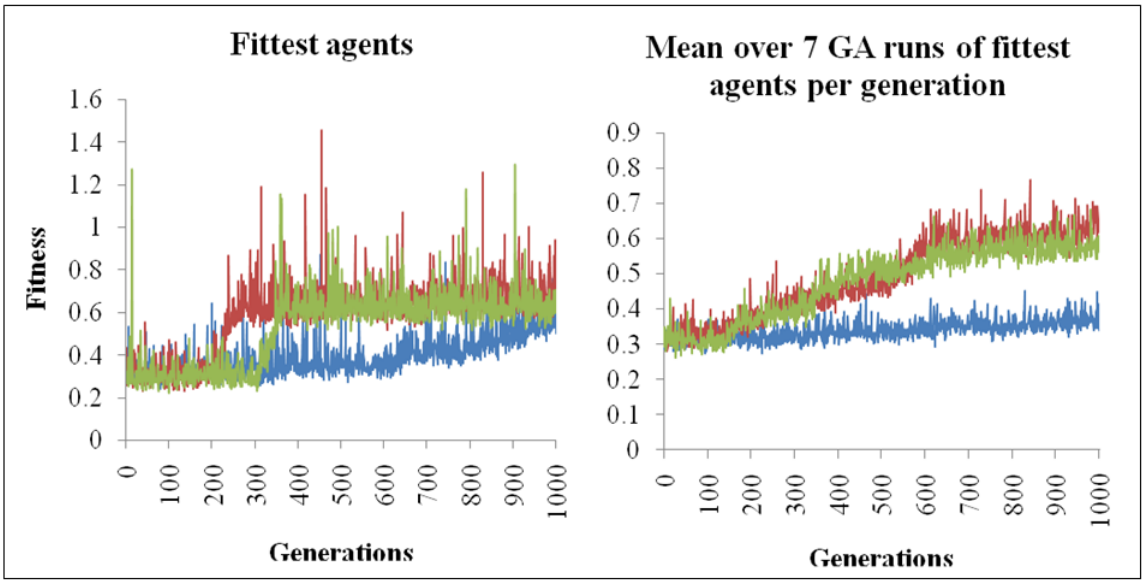


Figure 6.15: Left. Graph showing, for each developmental program used, the highest fitness values per generation of the run which had the fittest agent of all the seven runs. Right. Graph showing, for each developmental program used, the average highest fitness values of the 7 runs at each generation. The colors in the graphs represent: NO_SYM in blue, EVO_SYM in red, and ENF_SYM in green.

6.6 Conclusion

In this chapter, I have presented three novel developmental models allowing information to be encoded in space and time using spiking neurons placed on a 2D substrate. In two of these models, I introduced neural development that can use bilateral symmetry. I showed that these models created neural controllers that permit agents to perform chemotaxis, and do so much better than controllers with no symmetry. I also have shown that with the model using evolvable symmetry (EVO_SYM), neural bilateral symmetry was often evolved and was found to be beneficial for the agents. Perhaps this is not surprising. Firstly, the agent has bilaterally placed sensors and actuators. Secondly, the task of chemotaxis also has implicit symmetry: a chemical to the left triggers a turn to the left and symmetrically, a chemical to the right triggers a turn to the right. I think the size of the search space is almost the same for NO_SYM and EVO_SYM but much smaller for ENF_SYM. It is interesting to note that EVO_SYM and ENF_SYM converge on a solution in reasonable time and that EVO_SYM and ENF_SYM performed almost equivalently. The use of symmetry was clearly advantageous allowing faster evolution. Using NO_SYM, no correct solutions were found during the allocated time, however this model should in theory find a

correct solution if the genetic algorithm would run longer. Also, there were no restrictions about the number of neurons that each model (NO_SYM, EVO_SYM, ENF_SYM) could create. All the networks evolved with the three models could have the same number of neurons and connections.

It is important to note that complexification, targeting and neural selection are important concepts in the model. I used a 2D neural substrate where spiking neurons are placed and can grow connections to target locations. Therefore, the geometric configurations of the neural network significantly matter. Since I used spiking neurons with conduction delays, distances separating connected neurons result in time delays between the points in time when spikes are sent by a neuron, and the times they are received by another neuron. A neural network generated by my developmental models can encode information not only using firing rate encoding but also using temporal coincidence or delay encoding. Evolution can therefore generate neural networks able to encode external information as spatio-temporal patterns. In the following chapters, a section will be presenting a study on the neural coding strategies used by the fittest evolved neural controllers.

I have noticed from the results that sometimes more than one connection linking two cells was created. This is due to the limits of the weights we imposed, showing that the system can easily adapt to certain limiting constraints. I have seen that connections between symmetrical parts of the neural controller could be evolved and inhibitory connections of symmetrical neurons were also often evolved. Neural controllers grown with NO_SYM could have symmetrical neurons, but did so with an extremely low probability.

I have to emphasize the fact that the initial neural network, placed on the substrate, is bilaterally symmetrical. Most physical robots are also bilaterally symmetric, and therefore, I assumed that mapping sensors and motors to sensory and motor neurons on the neural substrate could be done in a direct manner when implementing my model on an agent or a real robot. In this case, it biased evolution to find an appropriate solution that uses this embedded symmetry. It would be very interesting to see if bilateral symmetry would still arise and be beneficial when evolving the morphology of the agent as well as the neural substrate. Cells could migrate on the substrate and differentiate to become sensors, neurons and motor neurons.

Many modifications of this model can be introduced. For example, it is possible to encode the threshold of a neuron or add different axes of symmetry in the genome. Other developmental models could have been created where only one gene could have created symmetrical neurons for the entire neural network. However, I decided to use EVO_SYM to permit the creation of both symmetrical and asymmetrical parts, and therefore to increase complexity.

In a system as complex as a spiking neural network placed on a 2D substrate in which both neural position and connectivity are evolved, the exploitation of the physical symmetry of an agent

has significant advantages allowing a more compact genetic representation leading to a relatively fast evolution of efficient neural networks. This work was the first, as far as I know, to present developmental models where spiking neurons are generated in space and where bilateral symmetry can be evolved and proved to be beneficial in this context. I think that studying how evolutionary processes can be affected by symmetrical structures in neural networks is of major importance and will have beneficial repercussions on Artificial Life research. I also emphasize that the creation of neural controllers having the possibility to use different neural coding strategies, using spiking neurons, is a very interesting and promising approach.

Chapter 7

Evolution of Olfactory Attraction and Aversion in Agents Controlled by Spiking Neural Networks

7.1 Introduction

The motivation of this work is based on a paper by Semmelhack and Wang entitled "Select Drosophila glomeruli mediate innate olfactory attraction and aversion"[85]. In this paper, as also mentioned in Chapter 3, they show that the glomerular anatomy and physiology has direct implications for animal behaviour; for example, in fruit flies, individual glomeruli can mediate innate behaviours like attraction and aversion. Fruit flies are highly attracted by the smell of vinegar, which they associate with their favourite food source, rotting fruit. However, they are repelled by vinegar when the concentration is too high. The authors showed that a higher concentration of vinegar excites an additional glomerulus and its activation is necessary and sufficient to mediate a behavioural switch (attraction/aversion). As I mentioned earlier, it seems that odour concentration is encoded by the number of stimulated glomeruli: the higher the concentration, the more olfactory receptor neurons (ORNs) are activated and the more glomeruli are recruited [35, 3, 4, 85, 63, 62]. Therefore, a high concentration of chemicals would activate specific sensory neurons that stimulate particular glomeruli.

To study this phenomenon in a simple manner, I evolved an agent to be attracted by a low level of concentration but repelled by a high level of the same chemical concentration. I first did experiments where the agent could use only one kind of sensors that responded to the whole range of concentrations. In the second experiments, the agents could use an additional pair of sensors

only activated by high concentrations. I wanted to investigate how can a spiking neural network encode information in order to control an agent that is attracted by a low level of concentration of a chemical but repelled by a high level of the same chemical. I also wanted to see if the neural controllers needed to be equipped with different types of olfactory sensory neurons or if only one type would be sufficient.

7.2 Experiments

In all of the following experiments, agents were evolved to be attracted by a low level of concentration [0, 150[(in arbitrary units) but repelled by a high level of concentration [150, 300]. I conducted experiments where the agents were evolved to perform both tasks during a run. I used the same agent and genetic algorithm as in my previous experiments. Each agent had five runs where they were placed randomly in the environment. The fitness function was simple and consisted of adding or subtracting, at each time step, the concentration value present at the centre of the agent (divided by 1000) to its energy value. An agent had an initial energy value set to 0. In the range of low concentration [0, 150[, the value of the concentration measured at the centre of the agent was added to the agent's energy value. In the range of high concentration [150, 300], the value of the concentration was subtracted from the agent's energy value. Therefore, the agents evolved to maximize their energy value by moving as close as possible to a concentration of 150 and by avoiding higher values (see Figure 7.2).

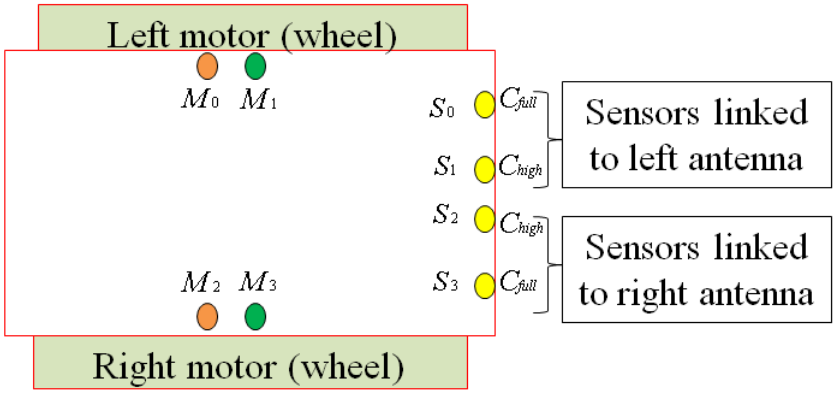


Figure 7.1: Initial neural network of an agent. Each wheel is controlled by two motor neurons, one to move forward (green) and one to move backward (orange). There are two pairs of sensors, one (S_0 and S_3: type C_{full}) to react to the full range of the chemical concentration [0, 300], and one (S_1 and S_2: type C_{high}) to react only to a high level of concentration [150, 300].

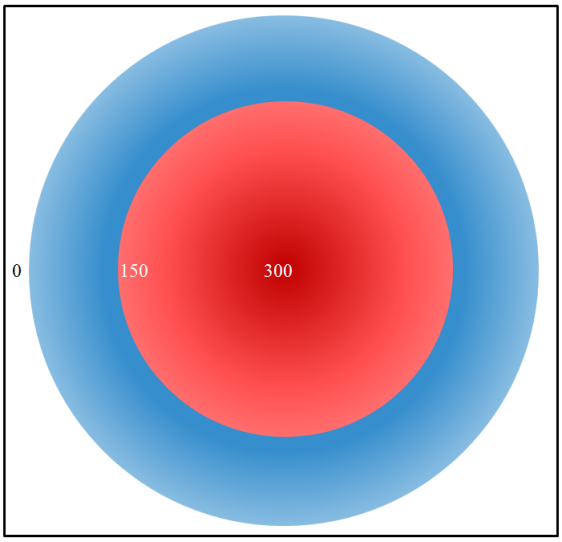


Figure 7.2: Circular area where the chemical is present. The blue gradient represent the area where the agent has to stay, and the red gradient represent the area that the agent has to avoid.

7.2.0.5 Using a Fixed Chemical Source

In these experiments, a chemical source was placed in the middle of the environment and could not move.

In the first experiments, the agent was equipped with an initial neural network composed of sensors of type C_{full} only (reacting to the full range [0, 300]) (Figure 7.1). Seven GA runs were performed using EVO_SYM and seven using ENF_SYM (see Chapter 6 for details on GA). The agents that evolved in these experiments had a simple neural network and behaved quite simply. Most of the fittest agents were just turning in circles in the low concentration and ignoring the high concentration. Using ENF_SYM, five out of seven GAs evolved a successfull agent. In these five, three turned in small circles and two turned around the center at a certain distance. Using EVO_SYM, all the GAs evolved a correct agent which turned in small circles. In fact, the evolved neural networks and behaviours were not really interesting so I will not go into more detail here.

In the second experiments, the agents were equipped with an additional pair of sensors (C_{high}), which reacted only to high concentrations [150;300] (Figure 7.1). As previously, seven GA runs were performed using EVO_SYM and seven using ENF_SYM. Most of the evolved agents were just turning in circles in the low concentration and were not using the second type of sensors (C_{high}). Using ENF_SYM, five out of seven GAs evolved a correct agent. In these five, one was moving in big circles around the center and the rest of them were turning in small circles in the low concentration. Using EVO_SYM, all the GAs evolved a correct agent that turned in small circles. Like in the previous experiments, the evolved neural networks and behaviours were not

really interesting as the task that they had to perform was too simple and could simply be solved by circling behaviour.

Therefore, I decided to use a moving chemical source so the agents would have to follow the concentration and stay in a certain range. This task was more complex and biologically realistic than just using a fixed chemical.

7.2.0.6 Using a Moving Chemical Source

In these experiments, I decided to equip the agents with the two pairs of sensors from the beginning to see if they would use all four of them or not (C_{full} and C_{high}, see Figure 7.1). As in the previous experiments, seven GA runs were performed using EVO_SYM and seven using ENF_SYM. The mean fitness values over seven GA runs of the fittest agents per generation can be seen in Figure 7.3.

Using EVO_SYM, the GAs did not manage to evolve an agent that could perform the desired task within the given time. The fitness of the two best agents is shown Figure 7.4, left.

Using ENF_SYM, only two out of seven GAs evolved a correct agent (Figure 7.4, right). These two GAs evolved agents that managed to be attracted by the low level of concentration and repelled by a high level (Figures 7.8 and 7.5), and interestingly, they were using only the sensors C_{full}, which responded to the whole concentration range. The fittest agent of all these simulations is described in the following section.

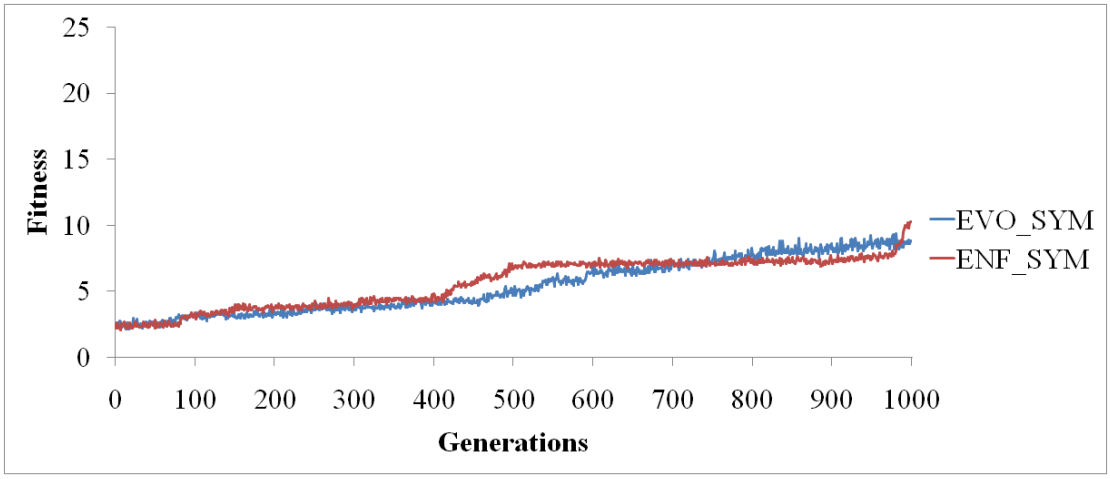


Figure 7.3: Mean fitness values over seven GA runs for the fittest agents per generation. Two sets of experiments were conducted, one using EVO_SYM and the other one using ENF_SYM.

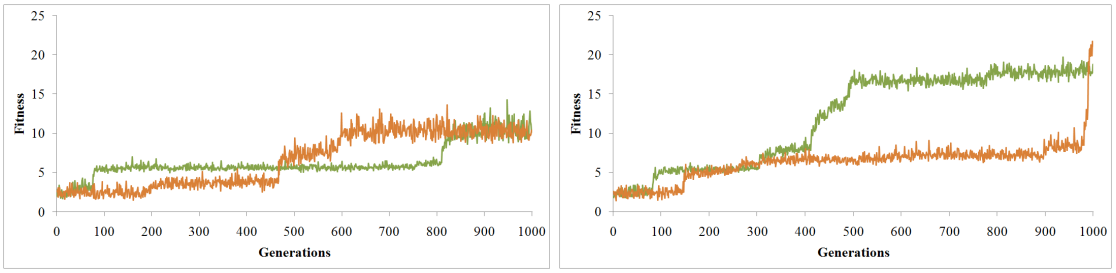


Figure 7.4: Fitness value of the two fittest agents evolved using EVO_SYM (left) and ENF_SYM (right).

Having evolved the agents using a moving chemical source, they were tested on further 300 seconds runs. These could be using a moving or chemical source, though all the ones illustrated here use a fixed chemical sorce for clarity purposes. The following pictures (Figure 7.5) shows a typical run that lasted for 300 seconds of an agent that did not work (left, fitness=7.95) and of the fittest agent (right, fitness=19.85). On the left, the agent manages to avoid the high concentration values, however fails most of the time to stay within the chemical area. On the right, the agent starts from outside the chemical concentration, then moves in and avoids the centre of the chemical gradient where the concentration is too high ($\geqslant 150$), while staying in the lower range of concentration (< 150).

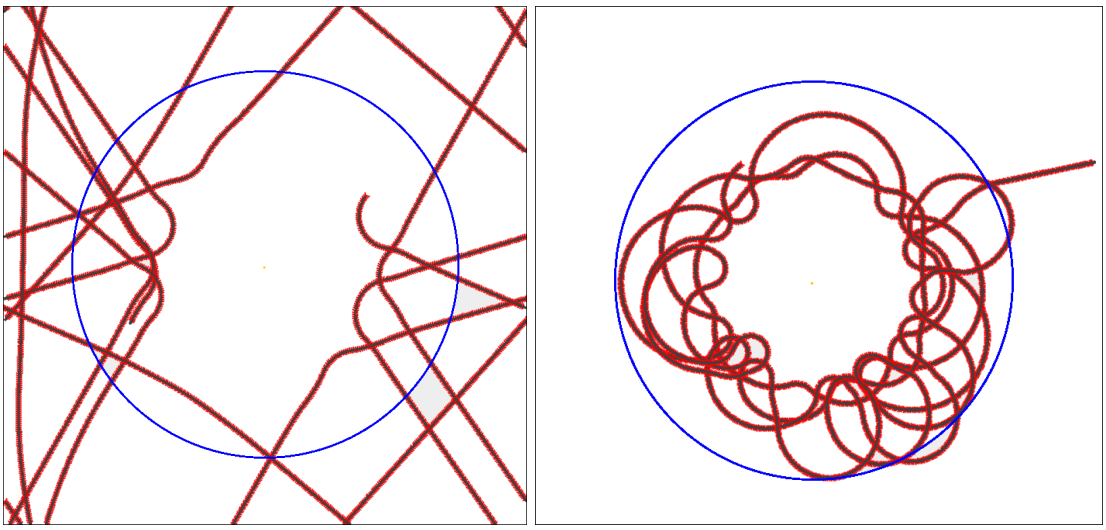


Figure 7.5: Examples of a run lasting 300 seconds where the agent is attracted by the chemical (right) but repelled when its concentration value is too high (left and right).

7.2.0.7 Analysis of the Fittest Evolved Spiking Neural Controller

In this section, I will describe in detail the neural network and the resulting behaviour of the fittest agent that evolved using ENF_SYM. The evolved neural network (Figure 7.6) is composed of the initial neural controller (Figure 7.1), three neurons (N_0, N_2, N_4) that were encoded in the genome, and their symmetrical clones (N_1, N_3, N_5). The weight matrix is shown in Table 7.7 (left), and the matrix showing the transmission delays for each connection is in Table 7.7 (right).

Interestingly, this agent did not evolve connections to the sensors S1 and S2 that get stimulated only at high concentration [150, 300]. This is quite surprising as using sensors C_{high} should make the task easier, however artificial evolution found a solution without using these additional sensors. Therefore, this agent does not need specific sensors like a fruit fly does. However, I have to emphasize that this agent had to react to only one chemical while flies have to distinguish hundreds of them. Also, my model of chemical diffusion is very simple and not realistic. These facts may explain why this agent only needs to use one type of sensors C_{full}.

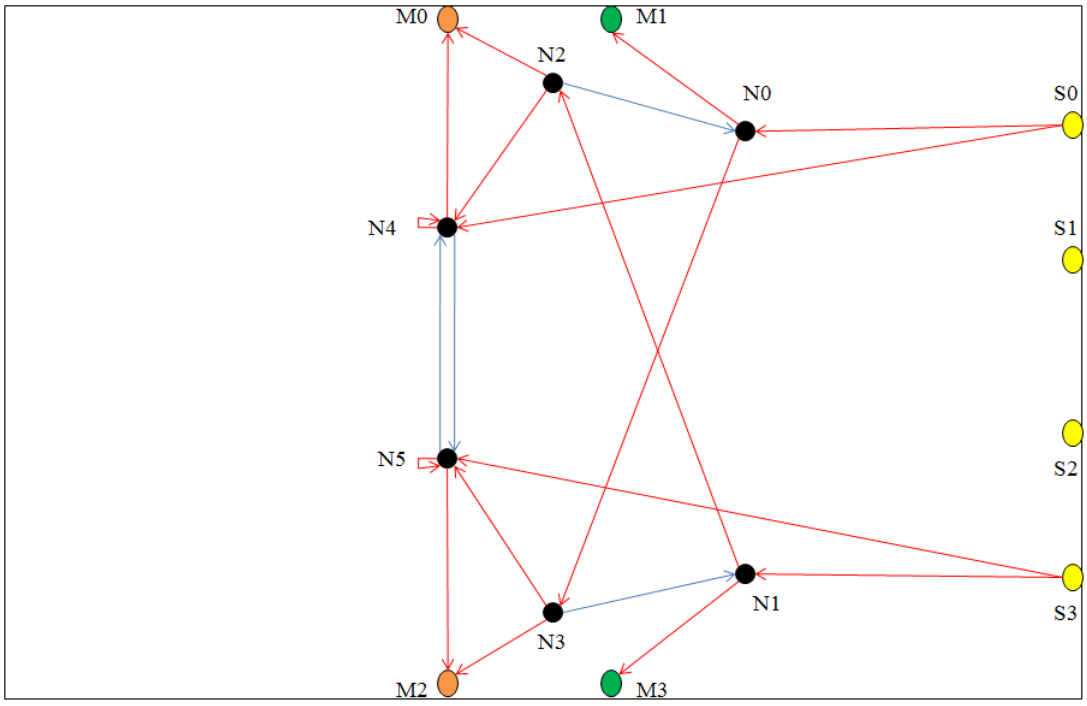


Figure 7.6: Neural network of the fittest agent. The four motor neurons and two sensors of the initial neural network were connected. Six interneurons were created during evolution, giving a total of 28 connections. The connections depicted in red are excitatory and the ones in blue are inhibitory. Self-connections of N_0 or N_1 are not shown here as they have zero weights and do not affect the neural activity of the controller (see Figure 7.7).

PRE

POST	S_0	S_3	N_0	N_1	N_2	N_3	N_4	N_5
M_0					12, 11		9	
M_1			15, 3					
M_2						12, 11		9
M_3				15, 3				
N_0	8		0		-8			
N_1		8		0		-8		
N_2				8				
N_3			8					
N_4	7				13		9	-12, -2
N_5		7				13	-12, -2	9

PRE

	S_0	S_3	N_0	N_1	N_2	N_3	N_4	N_5
M_0					0.4		0.85	
M_1			0.71					
M_2						0.4		0.85
M_3				0.71				
N_0	1.8		0		0.69			
N_1		1.8		0		0.69		
N_2				1.85				
N_3			1.85					
N_4	2.77				0.73		0	0.8
N_5		2.77				0.73	0.8	0

Figure 7.7: Left: weight matrix of the neural network from Figure 7.6 showing the connections linking presynaptic cells (top row) to postsynaptic cells (left column). Two values in the same box means that there are two connections, for example, N_0 has two output connections stimulating M_1. Right: transmission delay matrix of the neural network from Figure 7.6 showing the transmission delays (in ms) of connections linking presynaptic cells (top row) to postsynaptic cells (left column). Self-connections have a delay of 0 ms.

In order to understand completely how the behaviour of the agent emerged from the neural activity of the controller, I recorded the firing rates, total input current and membrane potential of each neuron during a run lasting 13 seconds. In this run, the agent was first placed in a low concentration area of the chemical. It then moved towards the centre, being attracted to a higher concentration. The agent was then repelled and turned away from the chemical until the concentration was low enough so the agent moved again towards the centre of the chemical source. Figure 7.8 shows the path of the agent and Figure 7.9 shows the firing rates recorded during the run. T1 to T6 represent key points in the run which will be referred to in the text. A table of the points, giving their significance is given in Table 7.1.

T1	the agent is attracted to the chemical and turns right
T2	the agent approaches higher concentrations
T3	the agent starts turning left, being repelled by high concentrations
T4-T6	the agent prepares to turn left to avoid high concentrations

Table 7.1: Particular points in the run referred to in the text and relevant graphs.

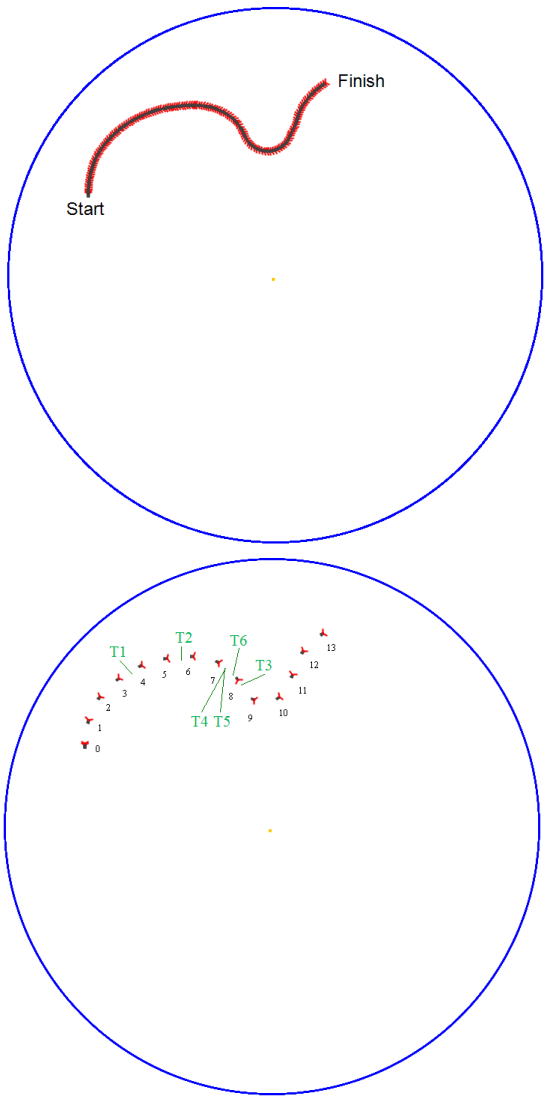


Figure 7.8: Path of an agent recorded during 13 seconds where the agent is attracted by the chemical first, and repelled when its concentration value is too high. The numbers in the right panel correspond to seconds. For each second, the mean firing rates of the neurons were recorded (see Figure 7.9). T1-T6 are particular points referred to in the text and marked on relevant graphs.

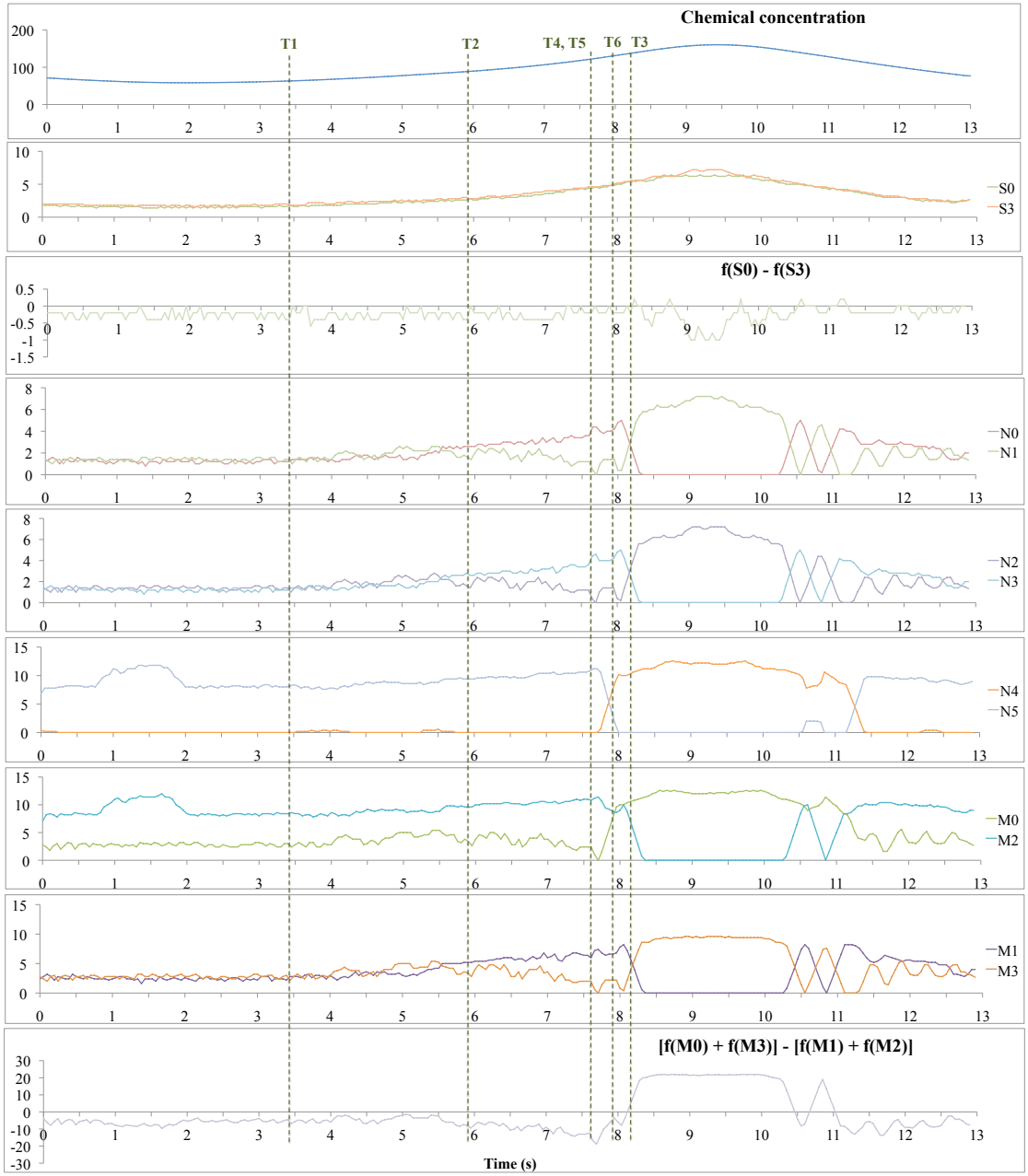


Figure 7.9: Concentration values and firing rates of the neurons recorded during the run shown in Figure 7.8. The number of spikes fired by each neuron was recorded every 50ms. The values show in the graphs are average rates during a window of 250ms (100ms before and 100ms after) sliding by 50ms for each value. The horizontal axis indicates time, and the vertical axes correspond to the chemical concentration value (top graph), or the firing rates of the cells (other graphs). The last graph indicates the movement direction of the agent. If the value is positive, the agent is turning left, otherwise, it is turning right. If the value is equal to zero, the agent is moving straight forward.

Attraction

I will first explain first how the agent is attracted to the chemical. The agent performs attraction behaviour from the beginning of the run and continues it for approximately 8 seconds. During this period of time, the main neurons responsible for this behaviour are the sensor S_3, neuron N_4 and motor neuron M_2. From the beginning, the sensor S_3 fires slightly more than S_0 as the concentration is higher on the right of the agent than on the left. This causes neuron N_5 to fire more than N_4 and due the excitatory self-connection of N_5 and the inhibitory connection between them, it dominates and inhibits N_4 completely. As N_5 stimulates the motor neuron M_2 which slows down the ipsilateral wheel, the agent turns right towards higher concentrations of the chemical. The subsystem made of N_4 and N_5 is a bistable system where one neuron always dominates and inhibit the other one. This subsystem is one of the key component of the neural network allowing the agent to perform its desired behaviour.

During this time, the neuron pairs N_0 and N_1, and N_2 and N_3 fire spikes at very similar firing rates, respectively even though N_0 and N_1 inhibit each other via N_2 and $N3$, none of them completely inhibits the other one as N_5 does with N_4. Therefore, the motor neurons M_0, M_1 and M_3 fire at approximately the same rate.

To understand why N_0 or N_1 are not inhibiting each other completely, I recorded the total input current and membrane potential of these neurons during one second, from $t = 3s$ to $t = 4s$, around point T1 (Figure 7.10). In these graphs, one can clearly see that both are inhibited at certain times but still fire quite often. To see in more detail why one neuron would fire, I recorded the input current produced by individual synapses (connections) between $t = 3.4s$ and $t = 3.5s$ (Figure 7.11). From this figure, I could see that N_0, for example, needed to have the following two events happening at the same time in order to start firing (example at $t \approx 60ms$, point T1 in graphs):

- a spike sent by sensor S_0 to N_0
- no or fading inhibition caused by a spike sent by N_2 to N_0

Therefore, spikes sent by sensors S_0 and S_3 need to be out of phase and provide excitation when contralateral inhibition is minimal.

The same mechanism is responsible for the contralateral equivalent cell N_1 firing as well. During this time of the run, the firing rates of the sensors S_0 and S_3 are relatively low, due to the level of the concentration. Therefore, these event have a high probability to happen resulting in neurons N_0 and N_1 both firing at similar firing rates.

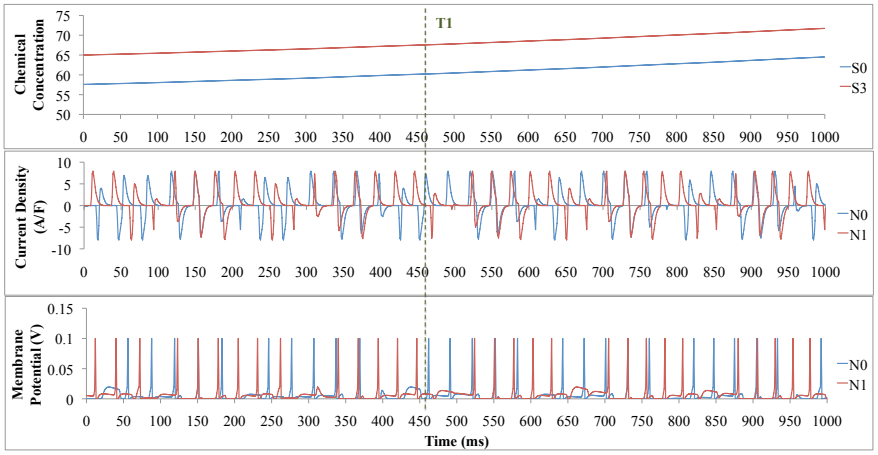


Figure 7.10: Concentration values, total input current and membrane potential of the neurons N_0 and N_1 recorded during $1ms$ bins for $1s$ between $t = 3s$ and $t = 4s$.

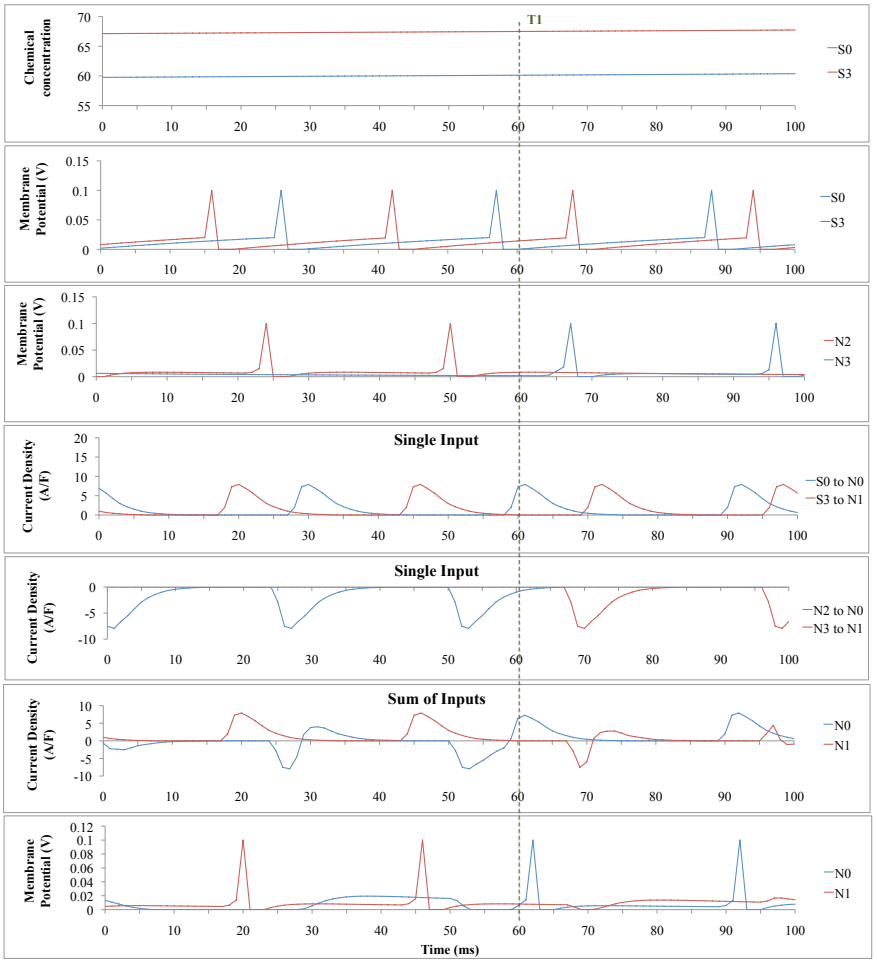


Figure 7.11: Concentration values, individual and total input current, and membrane potential of the neurons N_0 and N_1 recorded during $1ms$ bins for $100ms$ between $t = 3.4s$ and $t = 3.5s$.

During the attraction phase, an interesting phenomenon happens when the agent gets closer to a concentration of approximately 80, and when N_0 starts to fire more than N_1 (around point T2). To understand this, I recorded the total input current and membrane potential of these neurons when this phenomenom happens during one second, from $t = 5.5s$ to $t = 6.5s$ (Figure 7.12). In these graphs, I could clearly see that the input current of N_0 was higher than the current of N_1 for longer periods of time. To see in more detail why N_0 started dominating over N_1, I recorded the input current produced by individual synapses (connections) between $t = 5.9s$ and $t = 6.0s$ (Figure 7.13). From this figure, I could understand that the same mechanism previously explained was responsible for these dynamics. The neuron N_1 stops firing when the inhibition from N_3 coincides more and more with excitation from S_3. As spikes coming from the sensors are more and more out of phase, N_1 is not stimulated enough (at $t \approx 45ms$, point T2 in graphs) to the benefit of N_0. N_1 will not fire for a certain time cancelling its inhibition on N_0 which fires every time it receives a spikes from S_0. As the agent gets closer to a concentration of 150, the firing rates of the sensors are increasing, reducing the probability for this to happen compared to the case presented in Figures 7.10 and 7.11.

Interestingly, as N_0 fires more than N_1, the motor neurons M_1 and M_2 are more stimulated than M_0 and M_3 making the agent turn faster towards the centre of the concentration. I have to emphasize that, in certain runs where the concentration was also higher on the right than on the left, but where the agent was placed differently with different orientations, the neuron N_1 fired more than N_0 during the attraction phase. Therefore, the agent was not going faster towards the centre of the concentration and started to turn slowly away from the high concentration values. However, in these runs, the neural dynamics during the aversion phase (that will be described next) was similar.

Therefore, it seems that the most important phenomenom when the concentration is around 80, is that one of the neurons N_0 or N_1 starts to dominates over the other one. This dynamic is in fact essential for the next phase of the agent's behaviour which is aversion.

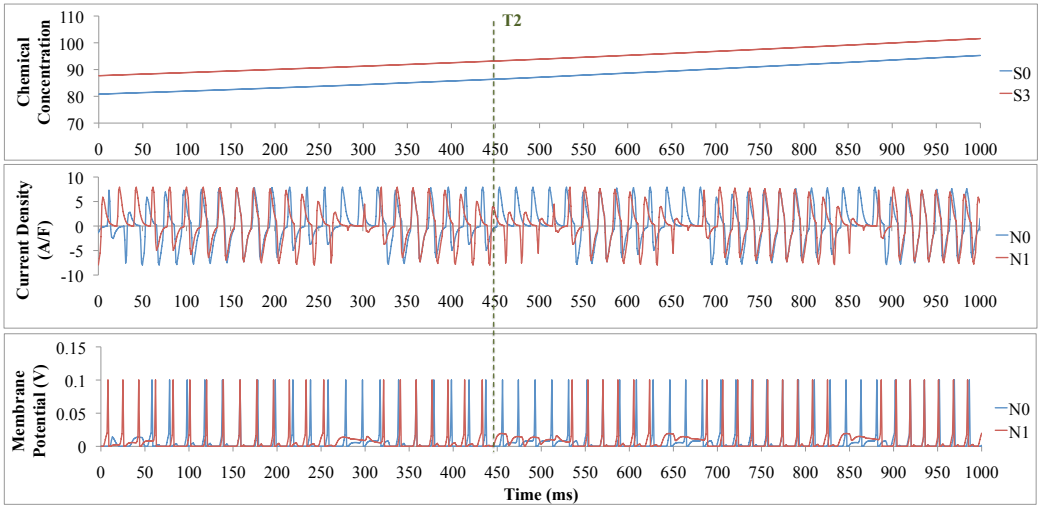


Figure 7.12: Concentration values, input current and membrane potential of the neurons N_0 and N_1 recorded during $1ms$ bins for 1s between $t = 5.5s$ and $t = 6.5s$.

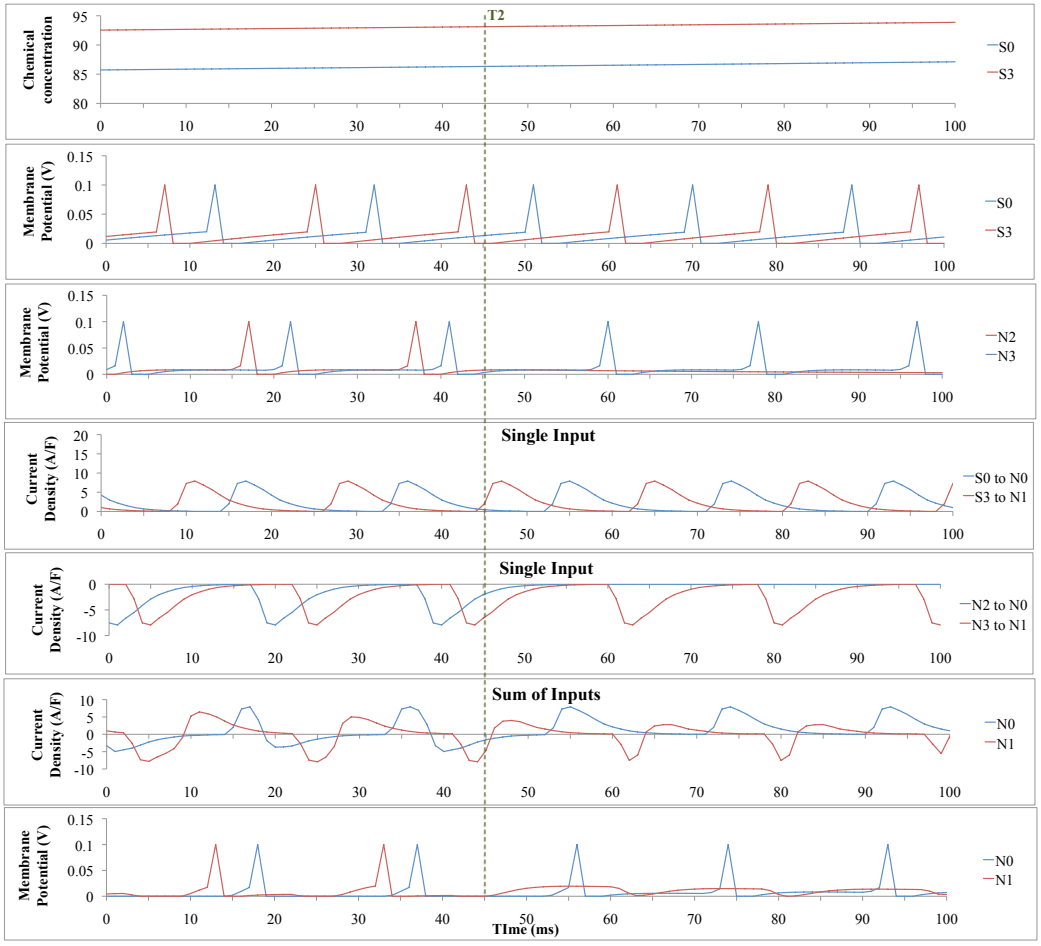


Figure 7.13: Concentration values, single and total input current, and membrane potential of the neurons N_0 and N_1 recorded for $100ms$ between $t = 5.9s$ and $t = 6.0s$.

Aversion

In the run shown in Figures 7.8 and 7.9, the agent changes its behaviour completely starting from $t = 8s$. To explain in detail this behavioural switch of the agent, I recorded the membrane potential of all neurons and the total input current of N_0 and N_1 during one second, from $t = 7.5s$ to $t = 8.5s$ (Figure 7.14). Up to this point, the agent is moving towards high values of the concentration, increasing the firing rates of the sensors S_0 and S_3, and also reducing slightly the difference between them as the agent is increasingly facing the centre of the chemical concentration. At the same time, the difference between firing rates of N_0 and N_1 is increasing. From these graphs, I first could see that N_0 was firing more than N_1, as described earlier, for approximately $800ms$. During this time, N_0 fired for $\approx 300ms$ intervals alternating with N_1 firing for $\approx 60ms$. Surprisingly, N_1 started to fire more than N_0 after $\approx 800ms$. To understand this in detail, I recorded the input current produced by individual synapses (connections) between $t = 8.25s$ and $t = 8.35s$ around point T3 (Figure 7.15). From this figure, I realized that, as for the attraction phase, when the sensors S_0 and S_3 are out of phase ($t \approx 30ms$, point T3 in graphs), the neuron N_1 starts to fire. During approximately $35ms$, both N_0 and N_1 fire and then N_1 fires more that N_0. The domination of N_1 over N_0 increases the firing rates of the motor neurons M_0 and M_3 which make the agent turn left, away from high levels of concentration. The fact that the neuron N_1 now fires more than N_0 via N_2 is sufficient to explain why the agent avoids high concentration values. I also noticed from Figure 7.9, that neuron N_1 inhibits N_0 completely during the aversion phase. This is due to the fact that as the agent turns left, the sensor S_3 fires much more that S_0 stimulating the neuron N_1 more than N_0, and also due to the fact that N_1 started to fire more than N_0 earlier.

In fact, another interesting phenomenon happens at $t \approx 500ms$ (point T6) (Figure 7.14) when neuron N_4 starts to fire and N_5 stops. This happens exactly during a periodic phase when N_1 fires more than N_0 for short period of time. I recorded the input current produced by individual synapses (connections) for both neurons N_4 and N_5 at two different times. In both cases, the neuron N_1 starts to fire and N_0 stops. Therefore, N_2 starts to fire and N_3 stops. Firstly, from $t = 7.60s$ and $t = 7.63s$ (points T4-T5) neuron N_1 starts to fire but the activity of the neuron N_4 does not change (Figure 7.16, left). Secondly, from $t = 7.97s$ and $t = 8.0s$ (point T6) neuron N_1 starts to fire and the activity of the neuron N_4 changes as it starts to fire (Figure 7.16, right). In the first case, at $t \approx 15ms$ (point T4 in graphs), a spike coming from the sensor S_0 and from the neuron N_2 arrive approximately at the same time at N_4. However, a spike that inhibits N_4 also arrives from N_5. Therefore, the temporal coincidence of spikes sent by S_0 and N_2 is not sufficient

as the neuron N_4 is not stimulated enough. At $t \approx 25ms$ (point T5 in graphs), a spike sent by N_2 stimulates N_4. Even if this spike arrives when the inhibition from N_5 is minimal, the input current of neuron N_4 does not increase enough for the membrane potential to reach the firing threshold.

In the second case, from $t = 7.97s$ until $t = 8.0s$, neuron N_4 starts to fire at $t = 16ms$ (point T6). N_4 fires a spike when the following events happen at the same time:

- a spike sent by sensor S_0 stimulates N_4
- a spike sent by N_2 stimulates N_4
- no or fading inhibition caused by a spike sent by N_5 to N_4

Therefore, the temporal coincidence of spikes sent by the sensor S_0, the neuron N_2 and the minimal effect of the inhibition from the neuron N_5 causes the neuron N_4 to fire.

By looking at Figure 7.14, I noticed that when neuron N_4 starts to fire around $t \approx 500ms$, it stimulates motor neuron M_2. However, as the neuron N_0 still fires more than N_1, the motor neurons M_1 and M_2 are still stimulated. This results in slowing down the agent for a short period of time ($375ms$) until the neuron N_1 starts to fire more than N_0 at $t \approx 800ms$. From this moment, only the motor neurons M_0 and M_3 are stimulated, controlling the agent to turn right quickly, away from the high level of concentration.

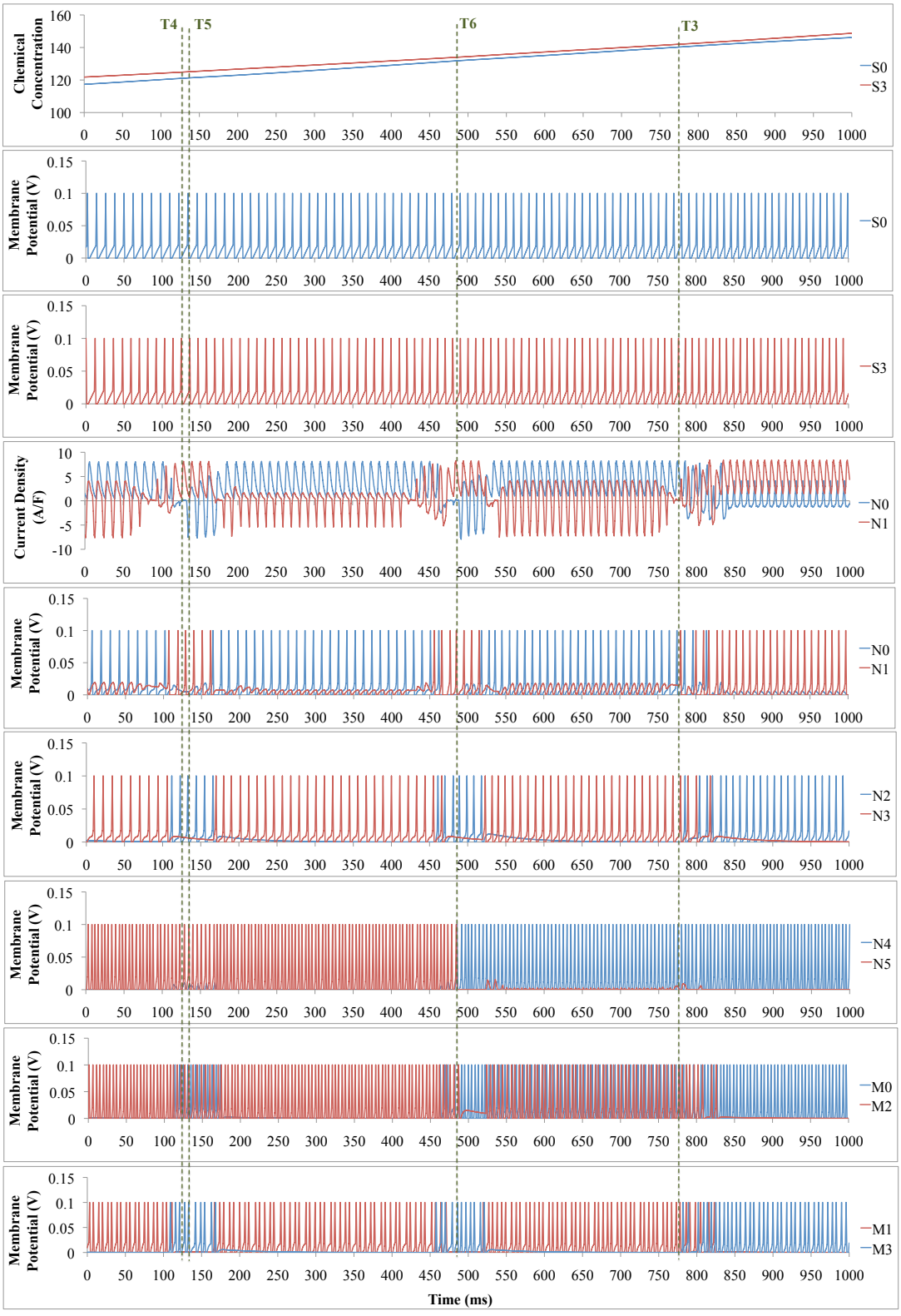


Figure 7.14: Concentration values, membrane potential of every neuron, and input current of the neurons N_0 and N_1 recorded during $1ms$ bins for 1s between $t = 7.5s$ and $t = 8.5s$.

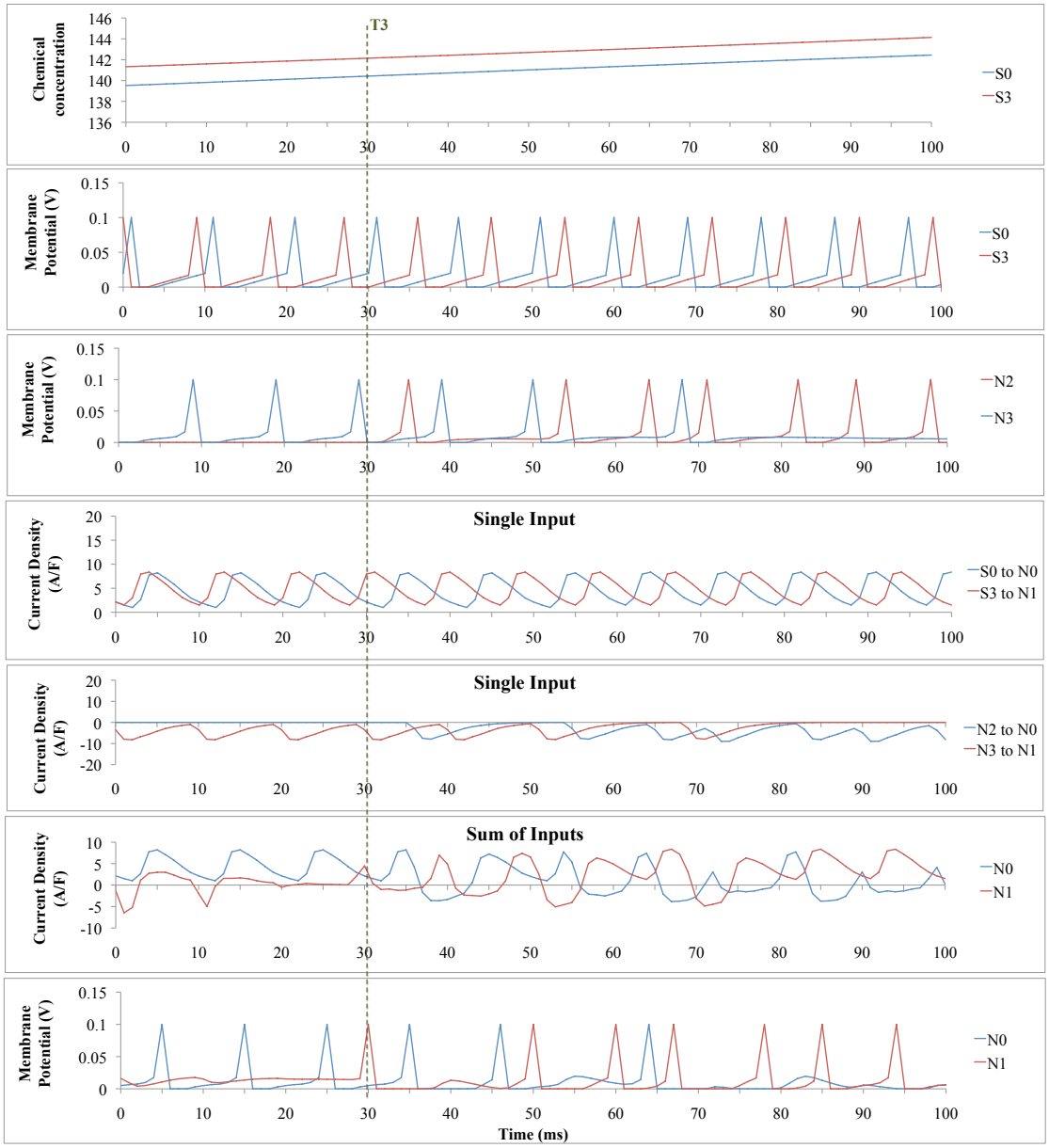


Figure 7.15: Concentration values, individual and total input current, and membrane potential of the neurons N_0 and N_1 recorded during $1ms$ bins for $100ms$ between $t = 8.25s$ and $t = 8.35s$.

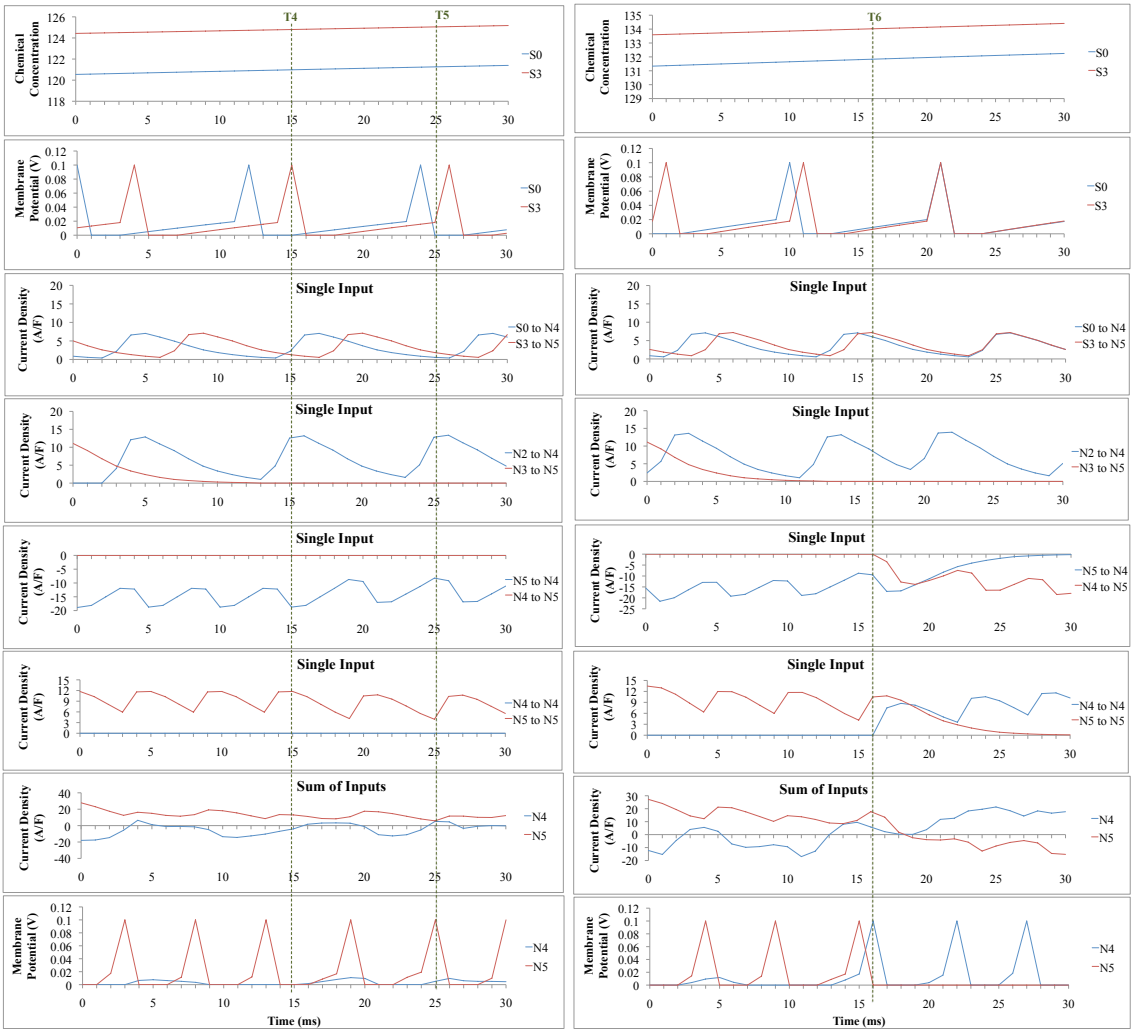


Figure 7.16: Concentration values, individual and total input current, and membrane potential of the neurons N_4 and N_5 recorded during $1ms$ bins for $30ms$ between $t = 7.60s$ and $t = 7.63s$ (left) and between $t = 7.97s$ and $t = 8.0$ (right).

To summarize:

- the bistable subsystem composed of neurons N_4 and N_5 is mainly responsible for motor control and has a direct role for attraction at low concentrations.
- the neurons N_0 and N_1 are responsible for switching the activity of the neurons N_4 and N_5 when the concentration is high enough. They perform this task using two mechanisms:
  - N_0 and N_1 can fire when they receive spikes sent out of phase by sensors S_0 and S_3 which arrive when inhibition is minimal. The sensors change between firing in phase and out phase, as one of them fires more than the other one, being more stimulated as it detects a higher concentration. This causes both neurons N_0 and N_1 to fire, even though they are inhibited by each other. When the concentration is close to 80, one neuron will fire more than the other one in an alternating fashion. These firing patterns are transmitted to the neurons N_2 and N_3 and are used by the second mechanism:
  - the temporal coincidence of spikes sent by sensors S_0 (or S_3), and by neurons N_2 (or N_3) and the minimal effect of the inhibition from the neuron N_5 (or N_4) causes N_4 (or N_5) to fire.

7.2.0.8 Robustness to Synaptic Weights Changes

In the previous section, I have identified the neurons that are mainly responsible for the behaviour of the agent. This implies that particular connections with certain strengths and delays are important. It seemed that the only connections that did not affect much of the behaviour of the agent are from neurons N_0 to M_1, N_1 to M_3, N_2 to M_0 and N_3 to M_2. In order to see the importance of each symmetrical connection, I conducted experiments where I modified the values of the synaptic weights of the neural network and looked at the resulting behaviour of the agent. The synaptic weights were reduced by 25, 50 or 100%. For each connection (and its symmetrical equivalent), I performed 100 runs of $300s$ duration using a moving chemical source. Without modification of the synaptic weights, the agent has an average fitness of ≈ 18.94 with a standard deviation of 3.13. The original values for the weights can be seen in Figure 7.7. The results of this study are shown in Figure 7.17. The results are for the reduction of the weights for both a connection and its symmetrical equivalent. In Figure 7.17 and the next paragraph, I refer just to the left hand connection, but I mean both it and its symmetrical equivalent.

From these results, I could identify clearly the connections that play a major role and also how difficult it was for the genetic algorithm to find the values of the weights. As I suspected, changes of the synaptic strength of the connection between N_2 and M_0 does not modify the behaviour of the agent. The connection between N_0 to M_1 is slightly more important as removing it completely reduces the fitness to 14. All the other connections play a major role however, with one connection being very sensitive to a weight decrease of only 25%. This is the case for the inhibitory connection between N_2 and N_0. As I showed earlier, this connection is important as the relationship between N_0 and N_1 allows avoidance of high concentration of the chemical. The strongest effect seen in this graph is found when removing the connection between S_0 and N_0. The resulting mean fitness value is negative, meaning that the agent does not avoid concentration values between 150 and 300. This is consistent with my earlier results that these connections played a major role in aversion.

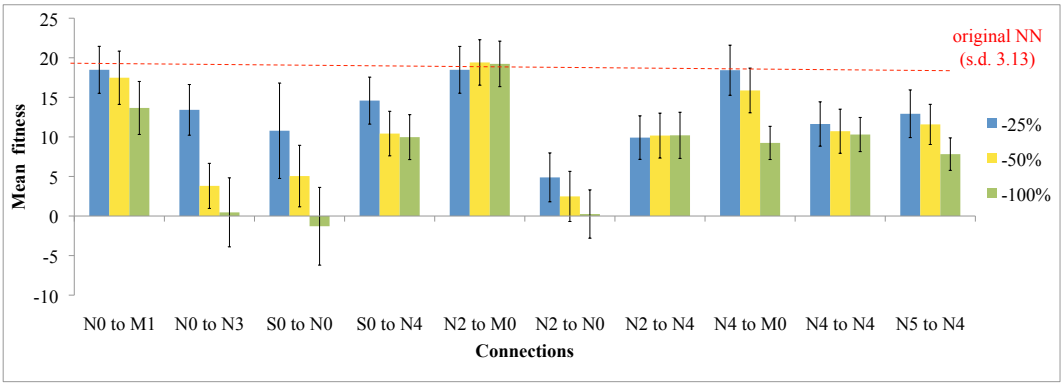


Figure 7.17: Mean fitness values recorded over 100 runs for each experiment where the strength of each connection and its symmetrical equivalent (not shown here) was reduced by 25%, 50% and 100%. The error bars show the standard deviation.

7.2.0.9 Robustness to Transmission Delays Changes

In order to see the importance of the transmission delay of each symmetrical connections, I conducted experiments where I modified the values of the transmission delays of the neural network and looked at the resulting behaviour of the agent. The original values for the delays can be seen in Figure 7.7. The synaptic delays were reduced by 25, 50 or 100%, and also increased by 25, 50 or 100%. For each connection (and its symmetrical equivalent), I performed 100 runs of 300s duration using a moving chemical source. Again, without modification of the conduction delays, the agent has a average fitness of ≈ 18.94 with a standard deviation of 3.13. The results of this study are shown in Figures 7.18 and 7.19. Again, I use the left hand connection as an abbreviation for both it and its symmetrical equivalent.

The first thing I noticed is that the effect of changing the delays is not as strong as changing the weights. Many delay changes do not affect the behaviour of the agent. However, certain connections are sensitive to changes in delays. In the initial analysis of the neural network, I noticed that the transmission delays between the sensors S_0 and S_3, and the neurons N_0, N_1, N_4 and N_5 seem to be quite important. The results shown in the following figures confirms this. Changing by $\pm 100\%$ the transmission delay between N_0 and N_3 also decreases the performance of the agent. The transmission delay of the inhibitory connection between N_5 and N_4 is also sensitive to changes. I also noticed previously that the timing of spikes sent via this connection was key to allowing aversion.

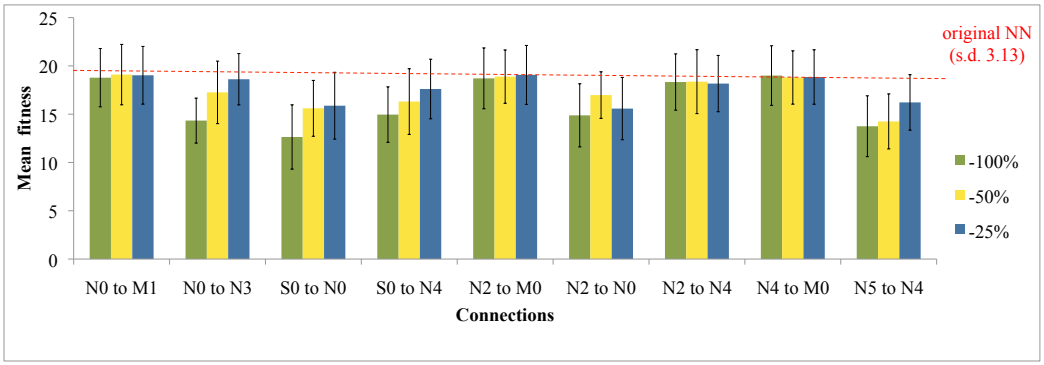


Figure 7.18: Mean fitness values recorded over 100 runs for each experiment where the transmission delay of each connection and its symmetrical equivalent (not shown here) was reduced by 25%, 50% and 100%. The error bars show the standard deviation.

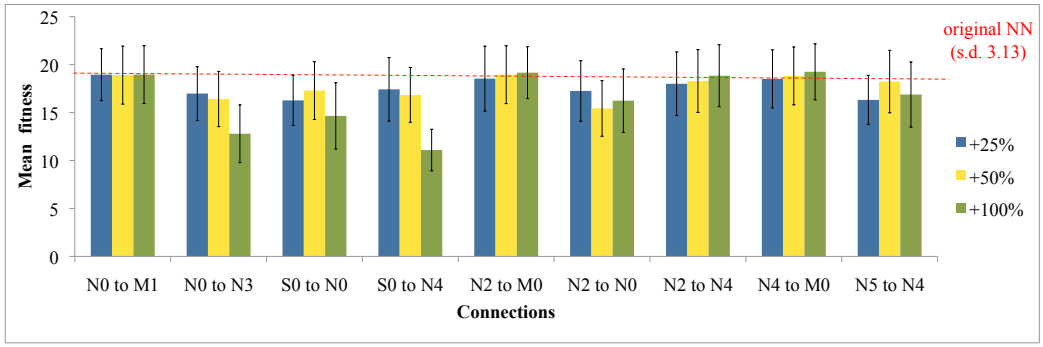


Figure 7.19: Mean fitness values recorded over 100 runs for each experiment where the transmission delay of each connection and its symmetrical equivalent (not shown here) was increased by 25%, 50% and 100%. The error bars show the standard deviation.

Knowing that the transmission delay of the connection between S_0 and N_4 (and its symmetrical equivalent from S_3 to N_5) is important to allow the agent to change direction, I perform two runs where I recorded the activity of the neural controller. In the first run, I used the original neural network, and in the second run, I reduced the transmission delays by 50% between S_0 and N_4 (and its symmetrical equivalent from S_3 to N_5). In both runs, the agent started from the same place having the same orientation. The runs are shown in Figure 7.20. The agent controlled by the original NN managed to stay inside the chemical (Figure 7.20, left). However, the agent controlled by the modified NN behaved similarly during half the simulation, and then changed direction and moved outside the chemical (Figure 7.20, right). To understand why the agent behaved in this way, I recorded the membrane potential of the neurons and the input current of N_4 and N_5 for the agent with reduced transmission delays. By looking at Figure 7.21, I noticed that the neuron

N_5 starts to fire at $t \approx 250ms$ and N_4 stops, resulting in a higher stimulation of motor neuron M_2 compared to M_0. As I have seen from the previous analysis of the neural controller, this causes the agent to change direction. I therefore looked at the input current and membrane potential of the neurons N_4 and N_5 when N_5 starts to fire and N_4 stops (Figure 7.22). I noticed that the temporal coincidence of a spikes received by N_5 and sent by the sensor S_3, and by the neuron N_3, and the minimal effect of the inhibition on N_5 from the neuron N_4 causes N_5 to fire. The exact same mechanism is used by the original NN for aversion, however modifying the delay of the connection between S_0 and N_4 (and S_3 and N_5) allowed spikes to become coincident that would not have been previously, and this allowed the movement out of the chemical. Therefore, the agent behaved as if it was avoiding high concentration values and changed direction. I have to emphasize that, in other runs where the agent started at different places, the same phenomenon happened, however the agent could rectify its trajectory and move back towards the chemical source. This results shows that the timing of spikes sent by different neurons is a key element in the neural coding strategies used in this neural network.

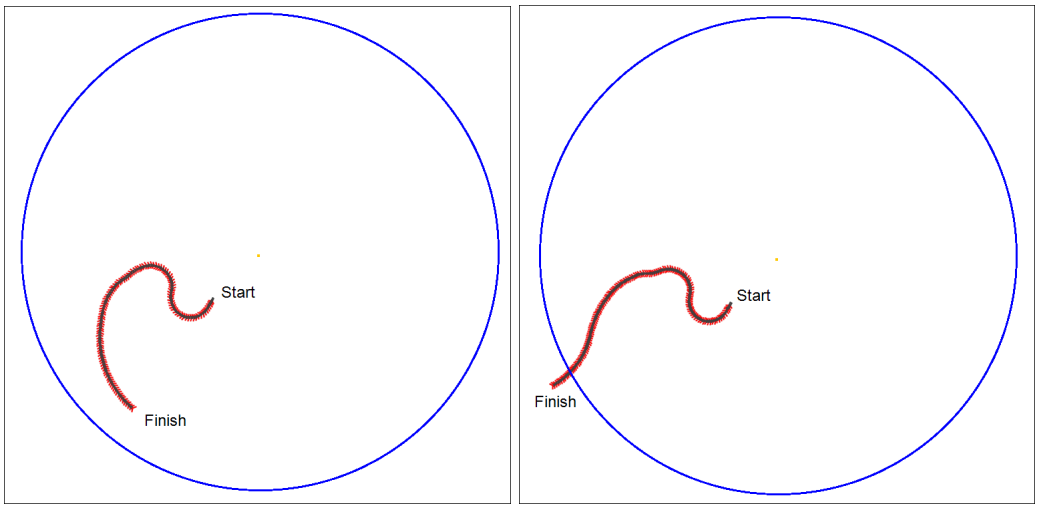


Figure 7.20: Path of an agent recorded during 13 seconds. Left: agent controlled by the original neural network. Right: agent controlled by the modified neural network (delay between S_0 and N_4, and S_3 and N_5 reduced by 50%).

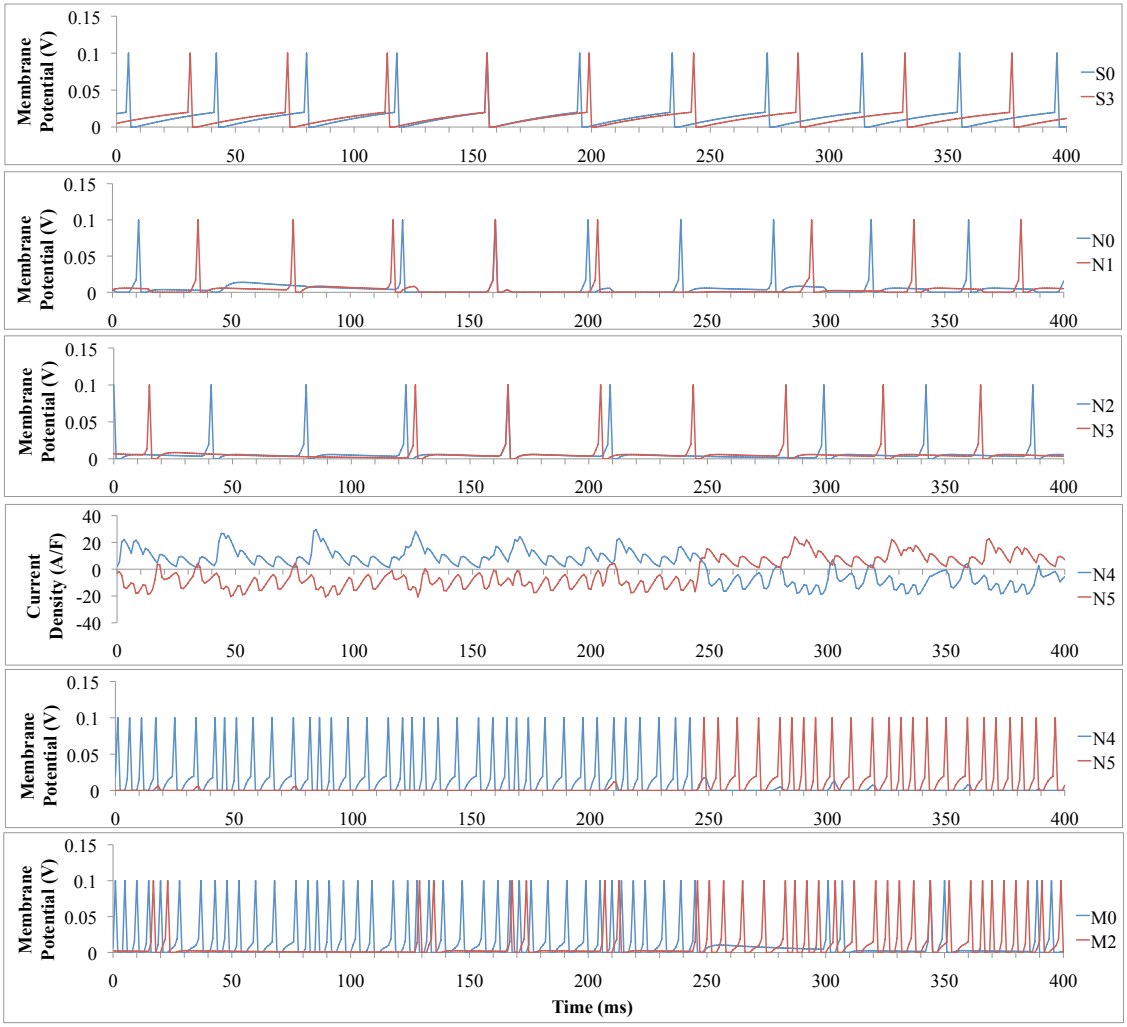


Figure 7.21: Membrane potential of the neurons, and input current of the neurons N_0 and N_1 recorded during $1ms$ bins for 1s when the agent changes direction to move outside the chemical.

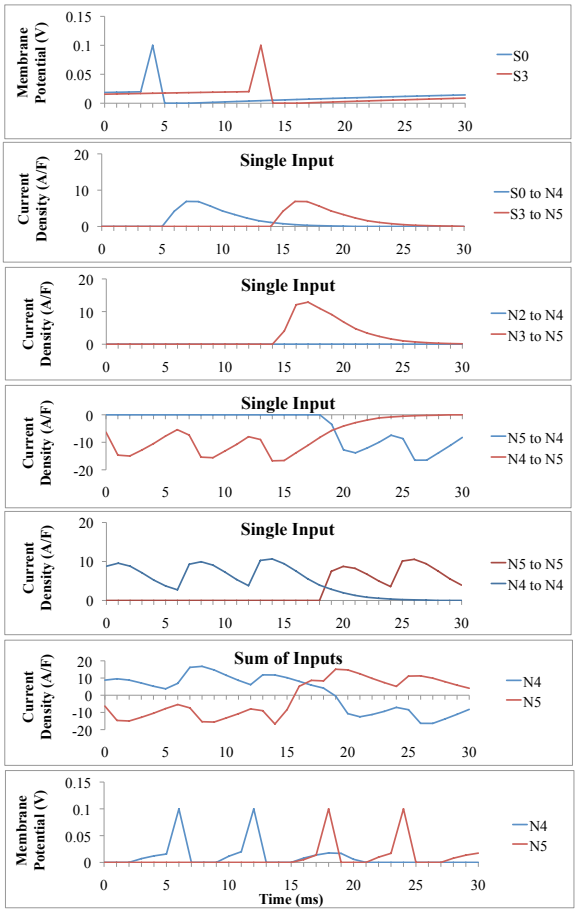


Figure 7.22: Individual and total input current, and membrane potential of the neurons N_4 and N_5 recorded using $1ms$ bins for $30ms$ when the neuron N5 starts to fire and N4 stops.

7.2.0.10 Robustness to Environmental Changes

In this section, I show results of experiments where the agent was placed in different environments to see if it could adapt easily to environmental changes. I performed five different experiments with 100 runs lasting $300s$ each time to see how the agent behaved in different environments. In each experiment, the radius of the circular area, in which the chemical is present, could be modified by changing the value of the coefficient K (see Equation 4.4). The radius of the chemical concentration could be calculated simply using: $radius = max/K$ with $max = 300$. The initial value of K used for the genetic algorithm was 0.3 (giving $radius = 1000$). Therefore, I conducted experiments with smaller and larger areas of the chemical where the values used for K were: $0.1, 0.2, 0.3, 0.4, 0.5$ giving radius values of 3000, 1500, 1000, 750 and 600. The size of the toroidal world also depended on the radius of the concentration so the chemical would always cover most of the environment: the bigger the concentration was, the bigger the world was. The agent was placed randomly in the environment for each of the 100 runs. I noticed that the agent managed to adapt quite well to chemicals of different concentration sizes (Figure 7.23).

Using a large environment and wide chemical concentration ($radius = 3000$), the agent was performing relatively well (for example in Figure 7.24, right) except when it was started from inside the area of high concentration. In this case, the agent had difficulties in getting outside the area of high concentration during the allocated time (example Figure 7.24, left), explaining the high value of the standard deviation.

Surprisingly, the agent performed better when the radius of the chemical was slightly bigger ($radius = 1500$) than the original chemical ($radius = 1000$) (example Figure 7.25). Even if the agent started from the middle of the concentration, it could always move away and stay in the low range of concentration (as in Figure 7.25, left).

When using smaller chemical concentration areas ($radius = 750$ and $radius = 600$), the agent was still performing relatively well even though the task was more difficult (examples Figure 7.26). When the radius was 600, the agent had difficulties in staying inside the chemical concentration but managed to avoid high concentrations.

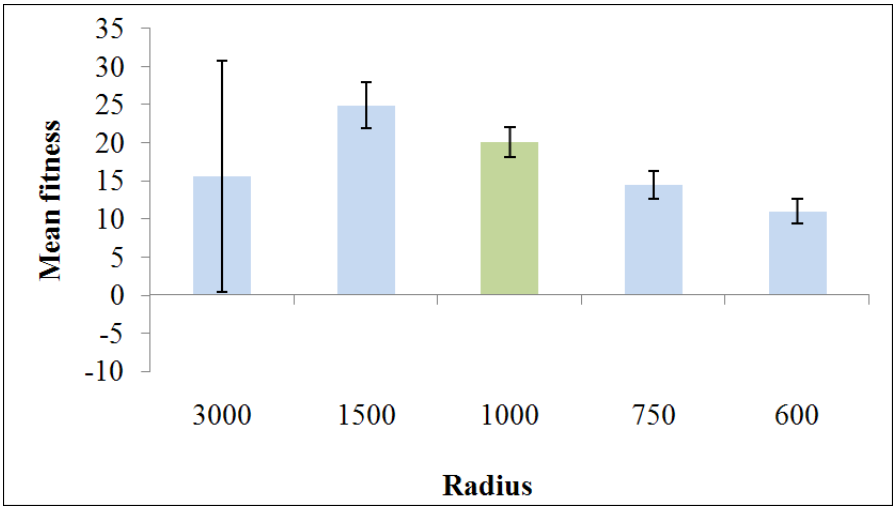


Figure 7.23: Mean fitness values, with standard deviations, of 100 runs performed in five different settings using chemical concentrations of different sizes.

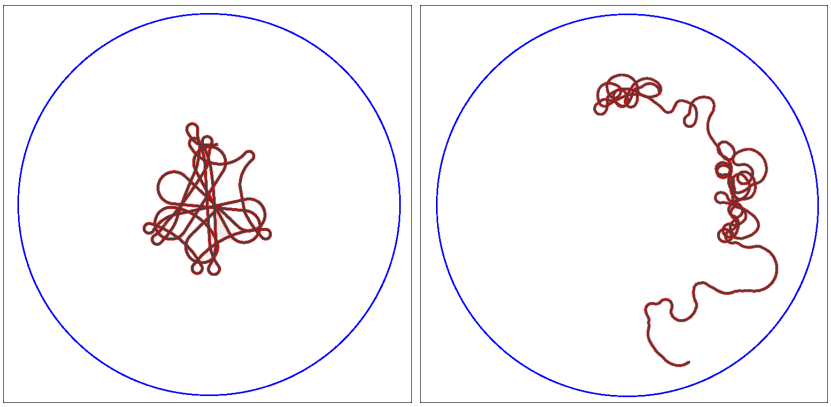


Figure 7.24: Path of an agent recorded during one run using a chemical concentration with a radius of 3000. On the left, the agent starts from the middle and cannot move to the low concentration area (fitness: -64.97). On the right, the agent started from a position in the low concentration area and stayed inside, avoiding the high concentration (fitness: 32.79).

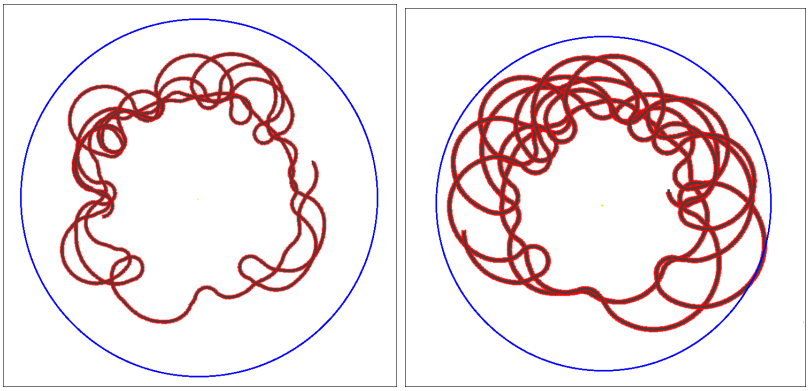


Figure 7.25: Path of an agent recorded during one run using a chemical concentration with a radius of 1500 (left), and 1000 (right). In both runs, the agent performed well (fitness left: 26.38; fitness right: 20.45).

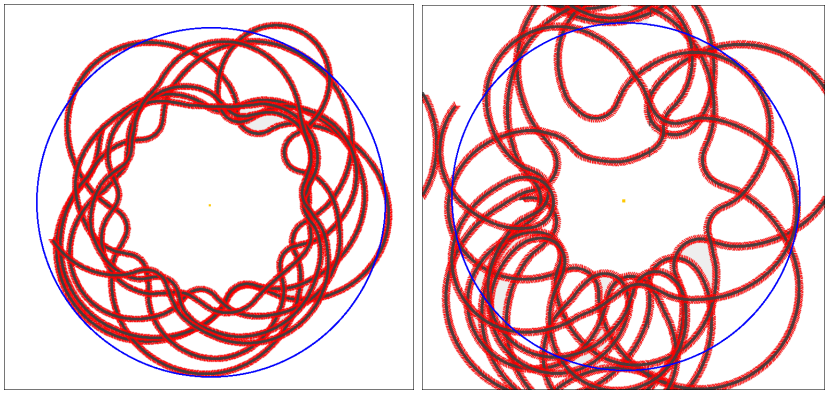


Figure 7.26: Path of an agent recorded during one run using a chemical concentration with a radius of 750 (left), and 600 (right). In both runs, the agent performed relatively well (fitness left: 12.28; fitness right: 7.04).

I also performed experiments to see how the agent would perform attraction and aversion individually. As in the previous section, I performed two different experiments with 100 runs lasting 300s each time to see how this agent behaved. In the first experiment, I used a range of concentration of $[0, 150[$ so the agent was attracted the whole area of chemical. In the second experiment, the agent had to avoid a chemical with a range of concentrations of $[150, 300]$. In this case, the highest fitness value an agent could have is 0 (if the agent did not get inside the chemical at all).

I noticed that the agent performed both tasks independently very well (Figure 7.27). In the low range of concentration, the agent was moving towards its source (Figure 7.28, left) allowing it to have a high fitness value. In the high range of concentration, if the agent started inside, it could

move outside and stay there (Figure 7.28, middle).

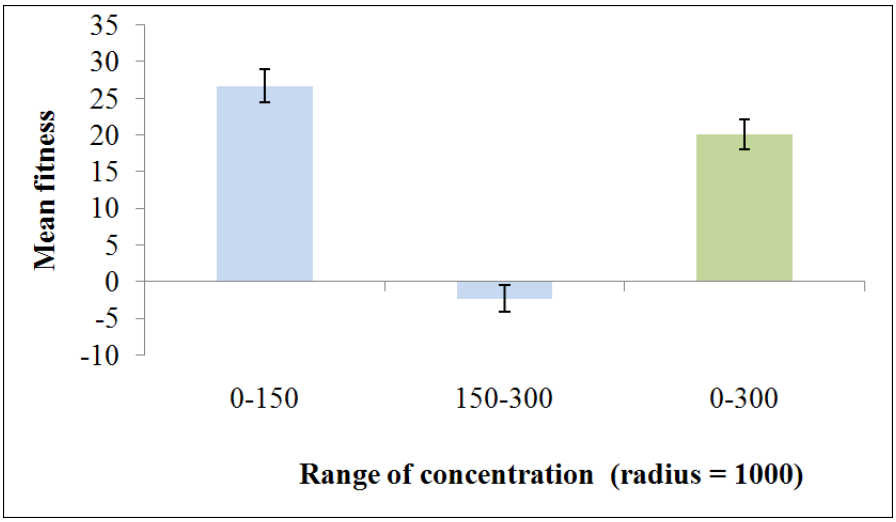


Figure 7.27: Mean fitness values, with standard deviations, of 100 runs performed in two different settings using two ranges of chemical concentrations. The green bar on the right is the value recorded using the original setting (radius: 1000, range [0, 300]).

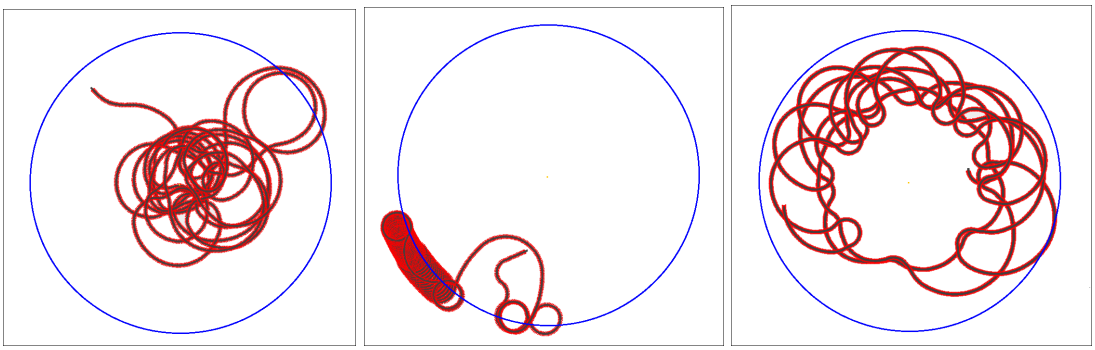


Figure 7.28: Path of an agent recorded during one run using a chemical with a range of concentration of $[0, 150[$ (left), and of $[150, 300]$ (middle). In both runs, the agent was performing well (fitness left: 26.05; fitness middle: -9.03). The picture on the right shows a run recorded using the original setting (radius: 1000, range [0, 300]) for comparison.

In the final experiments, I looked at the behaviour of the agent when it had to react to two different uniform concentrations, one that should attract it ($C = 100$), and one that should repel it ($C = 200$). As in the previous experiment, I noticed that the agent stayed inside the chemical with a concentration value of 100, and avoided the chemical with a concentration value of 200 (Figure 7.30).

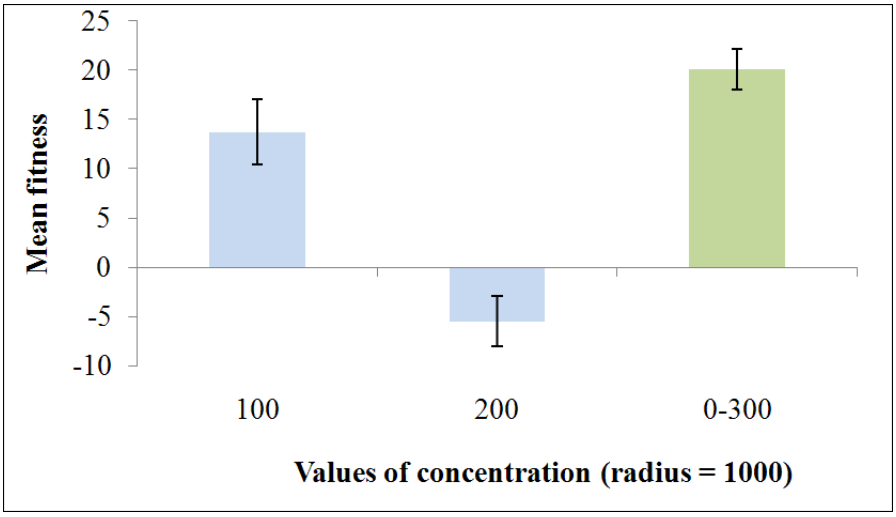


Figure 7.29: Mean fitness values, with standard deviations, of 100 runs performed in two different settings using two uniform chemical concentrations having different values.The green bar on the right is the value recorded using the original setting (radius: 1000, range [0, 300])

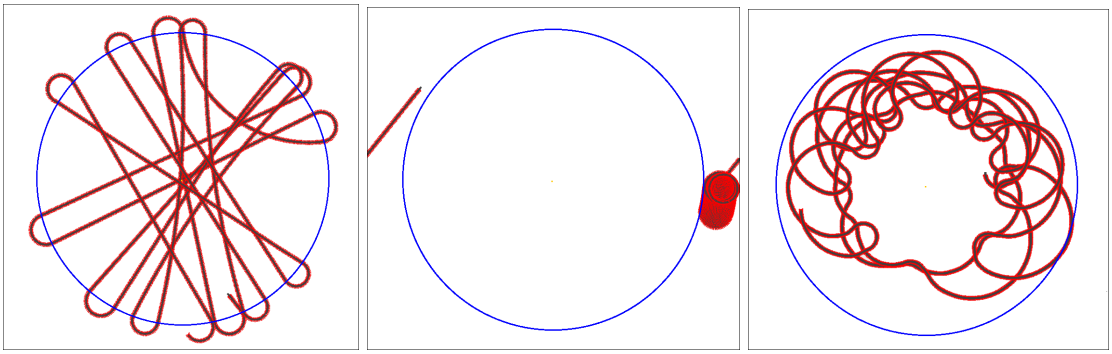


Figure 7.30: Path of an agent recorded during one run using a chemical with concentration value of 100 (left), and of 200 (middle). In both runs, the agent was performing well (fitness left: 18.01; fitness middle: -1.03). The picture on the right shows a run recorded using the original setting (radius: 1000, range [0, 300]) for comparison.

7.3 Conclusion

In this chapter, I have shown that using my developmental model, a neural controller with complex dynamics, and where timing of spikes is a key element, can be evolved. In the experiments presented here, agents were evolved to be attracted by a low level of concentration$[0, 150[$ but repelled by a high level of concentration $[150, 300]$. The agents were evolved to maximize their energy value by moving as close as possible to a concentration of 150 and avoiding higher values.

I wanted to see how a spiking neural network could encode information in order to control an agent that is able to perform this task. I was also interested to see if it was necessary to have different types of olfactory sensory neurons that fire at different levels of concentration. The fittest agent that evolved did not use the pair of sensors that fired only when the concentration was higher than 150. This shows that this agent does not need specific sensors like fruit flies do. I have to emphasize that this agent had to react to only one chemical while flies have to distinguish hundreds of them. Also, my model of chemical diffusion is very simple and not entirely realistic. These facts may explain why this agent only uses one type of sensor. However, artificial evolution managed to create an efficient neural controller that uses complex neural coding based on firing rates and temporal coincidence, where bilateral inhibition played a major role. I also showed that the agent could adapt quite well to different environments where the area of a chemical had a different radius.

In this chapter, I have presented a detailed study on the neural dynamics of the fittest evolved agent and showed that it used specific encoding based on temporal coincidence and firing rates. This analysis is the first, as far as I know, to have been done on an evolved neural network.

These results shows that an artificial approach to the study of natural processes like olfaction is feasible, and can give a better understanding of different neural coding strategies used by natural systems that have evolved during thousands of years.

Chapter 8

Conclusion

This thesis addressed the problem of creating simulated agents controlled by neural networks that share features with biological olfactory systems. My motivation was to investigate the relationship between the morphology and the physiology of a neural network and the behaviour of an agent controlled by such a network. I was also interested to study how information is encoded in such a system when evolved. I used an approach often found in Artificial Life using evolutionary computation and developmental models to generate neural controllers for agents that had to react to simulated chemicals. When investigating the neural controller of agents, I used an approach inspired by neuroscience. I was then able to look for similarities with biological olfactory systems. In the three years of my PhD, I conducted experiments in which I managed to evolved neural controllers for simulated agents that had relatively complex behaviours. The main results presented in each chapter will be brought together and summarised here.

8.1 Main Novel Contributions

The research conducted during my PhD have made the following contributions to the fields of Artificial Life, Artificial Intelligence and Neuroscience:

- I have created models that evolved developmental neural controllers for agents that could perform a relatively realistic and difficult task, and could encode information in space and time. I also have shown that in my models, the use of symmetrical structures had major benefits for the evolution of neural controllers. This work is important in that it addresses the problems of creating abstract models of development, of how information is encoded by a neural system and of generating agents that have realistic behaviours.

- I have conducted a detailed analysis of the neural dynamics on an evolved neural controller.

This has shown that my model generates controllers that use rather sophisticated neural coding strategies involving detailed temporal information. This work is important in that it shows the direct relationship between the activity of a neural controller and the resulting behaviour of an agent. It also shows how single spikes sent at specific moments can modify the whole activity of a network and the behaviour of an agent. To the best of my knowledge, this is the most detailed study of this type.

- I have also investigated in a simple manner two different neural coding strategies used by a simple neural architecture. I showed that both temporal coincidence (of spikes) and firing rate encoding strategies were important mechanisms that can be used by the same neural network in different environmental conditions. This work is relatively important in that it addresses the problem of how a neural system can use different neural coding strategies depending on external conditions.

- I have shown that by using a biologically plausible sigmoid function in my model to map pheromone concentration to the total input current of a leaky integrate-and-fire neuron, I could produce agents able to detect the whole range of pheromone concentration as well as small variations. The sensory neurons used in my model are able to encode the stimulus intensity into appropriate firing rates. This work is relatively important in that it shows how to map a stimuli into the firing rate of a spiking neuron using a biologically realistic approach.

- I have used a realistic model of neural noise and showed that it improves the behaviour of an agent. This work is relatively important in that it addresses the problem of the effect of noise on the performance of neural systems.

8.2 Summary of the Thesis

A summary of each chapter is presented here:

Chapter 2 gave a review on neural systems, different models of neurons used in the literature and two main coding strategies to represent information about stimuli. One coding strategy is based on the firing rate of a neuron and the other, based on the actual time of firing (spiking neurons). I discussed that simple spiking models, like integrate-and-fire neurons, can run fast enough compared to the complex and computationally slow Hodgkin and Huxley model, and still have a more realistic behaviour than firing rate ones. The usage of spiking neurons allows

information to be encoded in different manners using different strategies. For these reasons, more and more researchers are implementing spiking neurons in robots and simulated agents.

Chapter 3 first introduced the common features of olfactory systems and how they process chemical information. It then introduced the theories of how such systems compute. Then, it presented the work on olfaction that has been done in the field of Artificial Intelligence. Finally, it included a discussion on the evolution of olfactory systems and presented how this relates to the research questions that were the subject of my PhD.

Chapter 4 presented the model of neurons used in this thesis (leaky integrate-and-fire). It also presented the agent and its environment, and preliminary experiments conducted on the creation of the sensory neurons able to encode the stimulus intensity into appropriate firing rates. The main goal of these experiments was to create agents capable of finding and reacting to chemicals diffused uniformely from a point source. In order to achieve this goal, I had to find a model of spiking sensory neuron that could cope with small variations of pheromone concentration but could also react to the whole range of concentrations. It is already known that the mapping between the current and the firing rate of a leaky integrate-and-fire neuron is non-linear. Therefore, I tried many different functions to map the pheromone concentration onto the current of the sensory neuron in order to produce a reasonably linear relationship between the concentration and the firing rate of the sensor. After unsuccessful trials using linear currents, I derived a function that would necessarily give an exact linear relationship and used it as a model to help me find a similar function that is also used in biology. I concluded that by using a biologically plausible sigmoid function in my model to map pheromone concentration to current, I could produce agents able to detect the whole range of pheromone concentration as well as small variations. The sensory neurons used in my model are able to encode the stimulus intensity into appropriate firing rates. Moreover, using this model of sensory neurons, I managed to create an agent capable of chemotropotaxis.

Chapter 5 first presented experiments conducted using two different neural coding strategies in a neural controller of an agent. In this work, I used a simple neural architecture where temporal coincidence (of spikes) and firing rate encoding strategies were both important mechanisms used in different environmental conditions. In a low chemical concentration setting, synchronization of spikes sent by the sensors was essential to allow the agent to detect the blend of two chemicals. I changed the sensory delays and noticed that the agent was then not able to react to the chemicals anymore. However, in a high chemical concentration setting, the temporal coincidence between sensors firing was not a necessary condition and the

agent was able to stay inside the chemical concentration using just the firing rate encoding strategy even in the presence of just one chemical. This model also showed much more sensitivity to the presence of two chemicals than a single chemical. In principle, more than two chemicals can be detected and processed.

The second part of this chapter discussed the effect of noise on the agent's behaviour using the neural architecture from the previous experiments. I constructed a more complex environment using chemical gradients and a realistic model of neural noise. I found that the overall fitness of the agent was better when a certain amount of noise was added in the neural network. These results suggest that a realistic model of noise can improve an agent's behaviour. This is further evidence that adding biologically realistic features can be beneficial for certain engineering tasks, and suggests a potential function of noise in real biological systems. The effect of biologically realistic noise should be an interesting topic of research in other artificial life scenarios. I need to emphasize the fact that I might have the same results by simulating environmental noise or sensory noise instead. I think it would be interresting to add neural noise in real robotic experiment to study its effects.

Chapter 6 summarised work undertaken using an evolutionary approach and three novel developmental models allowing information to be encoded in space and time using spiking neurons placed on a 2D substrate. In two of these models, I introduced a neural developmental model that can use bilateral symmetry. I have shown that these models created neural controllers that permit agents to perform chemotaxis, and do so much better than controllers that were evolved from models that made no intrinsic use of symmetry. I also have shown that with the model using evolvable symmetry (EVO_SYM), neural bilateral symmetry was often evolved and was found to be beneficial for the agents. I have shown that the use of symmetry was clearly advantageous allowing faster evolution. Using NO_SYM, no correct solutions were found during the allocated time; however, this model should in theory find a correct solution if the genetic algorithm would run longer. Also, there were no restrictions about the number of neurons that each model (NO_SYM, EVO_SYM, ENF_SYM) could create. All the networks evolved with the three models could have the same number of neurons and connections. It is important to note that complexification, targeting and neural selection are important concepts in the model. I used a 2D neural substrate where spiking neurons are placed and can grow connections to target locations. Therefore, the geometric configurations of the neural network significantly matter. Since I used spiking neurons with tranmission delays, distances separating connected neurons result in time delays between the points in

time when spikes are sent by a neuron, and the times they are received by another neuron. A neural network generated by my developmental models can encode information not only using firing rate encoding but also using the time of spikes. Evolution can therefore generate neural networks able to encode external information as spatio-temporal patterns. In a system as complex as a spiking neural network placed on a 2D substrate in which both neural position and connectivity are evolved, the exploitation of the physical symmetry of an agent has significant advantages allowing a more compact genetic representation leading to a relatively fast evolution of efficient neural networks. This work was the first, as far as I know, to present developmental models where spiking neurons are generated in space and where bilateral symmetry can be evolved and proved to be beneficial in this context. I think that studying how evolutionary processes can be affected by symmetrical structures in neural networks is of major importance and will have beneficial repercussions on Artificial Life research. One aspect of Artificial Life investigates major transitions in artificial and real evolution and symmetry surely plays an important role in this process. I also emphasize that the creation of neural controllers having the possibility to use different neural coding strategies, using spiking neurons, is a very interesting and promising approach.

Chapter 7 presented experiments that have shown that using my developmental models, a neural controller exhibiting complex dynamics where timing of spikes was a key element, could be evolved. In the experiments presented in this chapter, agents were evolved to be attracted by a low level of concentration but repelled by a high level of concentration. The agents were evolved to maximize their energy value by moving as close as possible to a concentration of 150 and avoiding higher values. The artificial evolution managed to create an efficient neural controller that uses complex neural coding based on firing rates and spike timing, and where bilateral inhibition played a major role. Even though the evolved neural controller has a small number of neurons and connections, it exhibits an emergent coding strategy that is relatively complex. In this experiment, I also have shown that the agent could adapt quite well to different environments such as when the chemical covered different extents of the environment. In this chapter, I presented a detailed study on the neural dynamics of the fittest evolved agent and show that it used specific encoding based on temporal coincidence and firing rate. Such things as bistable subsystems were evolved. It was interesting to note that none of the fittest neural networks that were evolved made any use of the different types of sensors. In all cases, these networks only made use of the sensors that reacted to all levels of chemical concentration and they did this by making use of spike timing changes when the

chemical concentration increased. This analysis is the first, as far as I know, to have been done on an evolved neural network. These results shows that an Artificial Life approach to the study of natural processes like olfaction is feasible, and can give a better understanding of different encoding strategies used by neural systems.

8.3 Directions of Future Research

During the three years of my PhD, I have created a framework that permits the creation of agents and chemicals placed in a simulated 2D environment. The agents were controlled by spiking neural networks that could encode information using temporal patterns. My work could be used by others in the future to study other subjects such as:

- the evolution of chemical communication in a multi-agent system. Agents could be equipped with "glands" that could emit chemicals when stimulated by other neurons. Different types of chemicals having different effects (attraction, repulsion...) could be emitted. Collective behaviours could also be studied in these experiments where agents would have to perform certain tasks that would need agents to collaborate, and where using communication would be advantageous.

- the evolution of sex in a population of agents that would have to survive in an environment by gathering food resources and avoid dangerous areas. In such an environment, no specific fitness function would be used. The only objective the agents would have to perform would be to live and self-replicate when consumming enough resources. Mutations could occur during self-replication. Alternatively, the agents could use sexual reproduction. One agent could evolve to emit a chemical that would attract other agents. These agents would meet, and the genome from the two different agents could be used to create a new agent. It could be interesting to see if a population of agents evolve to use sexual reproduction or if it can survive just by using self-replication.

- the comparison of the results presented in my thesis with results from experiments that would be done using different types of genetic algorithms and genetic encodings. Some encodings could possibly allow different types of symmetry to be used.

The results of this program of research suggest several extensions to my work which may help in the understanding of the processes invoded in olfactory systems.

- More realistic models of odours plumes could be used, where odour molecules distributed in filaments are transported by the wind. Therefore, the concentration will not be in the form

of a continuous gradient. An easier alternative would be to simulate a real plume using a probabilistic function which requires less computational power.

- Habituation could be added. This is an important phenomena happening in olfactory receptor neurons. It results in a decline of response to a maintained stimulus and a reduced reaction to repeated stimuli. In silk moths, habituation is induced by a 30-second exposure to pheromone and results have shown that the sensitivity of the moth decreases for 1.5h and then gradually recovers. The addition of habituation, in the form of sensory desensitization, on the neural activity and behaviour of an evolved agent could then be analysed.

- Another possible future study could be the addition of synaptic plasticity. In nervous systems, the strength of connections can usually be modified via synaptic plasticity. Synaptic plasticity in the model could be introduced and a study undertaken on its effects on the neural controller activity and the behaviour of an evolved agent.

- Also, the role of neuromodulation in an evolved neural controller could be investigated. In silk moths, the neuromodulator serotonin enhances the sensitivity to pheromones. The variation of serotonin levels correlates with circadian variation of the male sensitivity to a pheromones. The effects of neuromodulation on the neural controller activity and the behaviour of an evolved agent could then also be analysed.

My model should also be able to allow an agent to learn in a multiple domain area. Different types of sensors and actuators (motor neuron) could be added to the initial neural network for this purpose. As evolution is a slow process, generating an agent that can perform different tasks should be faster if evolutionary computation would be used in combination with a model that allows online learning (using synaptic plasticity) and developmental prunning (modification of connections during a run).

8.4 Publications

During the three years of my PhD, I have attended several international conferences and published the results of my research.

- My first paper entitled “Optimal receptor response functions for the detection of pheromones by agents driven by spiking neural networks”was presented at the 19th European Meeting on Cybernetics and Systems Research, in Vienna (2008). This paper is the basis of Chapter 4.

- My second paper entitled "Adaptive Olfactory Encoding in Agents Controlled by Spiking Neural Networks" was presented at the Tenth International Conference on Simulation of Adaptive Behaviour, From Animals to Animats (SAB2008) in Osaka (2008) .

- My third paper entitled "Optimal noise in spiking neural networks for the detection of chemicals by simulated agents" was presented at the Eleventh International Conference on the Simulation and Synthesis of Living Systems (ALIFE XI), in Winchester (2008). These two papers are the basis of Chapter 5.

- My fourth paper entitled "Evolution of Bilateral Symmetry in Agents Controlled by Spiking Neural Networks" was presented at the second IEEE Symposium on Artificial Life, in Nashville (2009). This paper is the basis of Chapter 6.

- I also submitted a paper to the Artificial Life Journal. It has been reviewed and I will resubmit it with corrections in January.

Bibliography

[1] Ampatzis, C., Tuci, E., Trianni, V., & Dorigo, M. (2008) Evolution of Signaling in a Multi-Robot System: Categorization and Communication. *Adaptive Behavior*, **16**, 5–26.

[2] Arthur, W. (1997) *The Origin of Animal Body Plans, A Study in Evolutionary Developmental Biology*. Cambridge University Press.

[3] Asahina, K., Pavlenkovich, V., & Vosshall, L. (2008) The survival advantage of olfaction in a competitive environment. *Curr Biol*, **18**, 1153–1155.

[4] Asahina, K., Louis, M., Piccinotti, S., & Vosshall, L. (2009) A circuit supporting concentration-invariant odor perception in drosophila. *Journal of Biology*, **8**, 9.

[5] Bongard, J. & Pfeifer, R. (2003) Evolving complete agents using artificial ontogeny. *Morpho-functional Machines: The New Species (Designing Embodied Intelligence)*, pp. 237–258.

[6] Braitenberg, V. (1984) *Vehicles: Experiments in Synthetic Psychology*. MIT Press, Cambridge, MA.

[7] Brody, C. & Hopfield, J. (2003) Simple networks for spike-timing-based computation, with application to olfactory processing. *Neuron*, **37**, 843–852.

[8] Bryden, J. & Cohen, N. (2008) Neural control of caenorhabditis elegans forward locomotion: the role of sensory feedback. *Biological Cybernetics*, **98**, 339–351.

[9] Buck, L. & Axel, R. (1991), A novel multigene family may encode odorant receptors: A molecular basis for odor recognition.

[10] Channon, A. & Damper, R. (1998) The evolutionary emergence of socially intelligent agents. Edmonds, B. & Dautenhahn, K. (eds.), *Socially Situated Intelligence: a workshop held a SAB'98*.

[11] Channon, A. & Damper, R. (1998) Evolving novel behaviours via natural selection. Adami, C., Belew, R., Kitano, H., & Taylor, C. (eds.), *Proceedings of the 6th Conference on the*

simulation and synthesis of living systems (ALIFE VI), Los Angeles, CA, USA, pp. 384–388, MIT Press.

[12] Couto, A., Alenius, M., & Dickson, B. J. (2005) Molecular, anatomical, and functional organization of the drosophila olfactory system. *Current Biology*, **15**, 1535 – 1547.

[13] De Matos, A. M., Suzuki, R., & Arita, T. (2007) Heterochrony and evolvability in neural network development. *Artificial Life and Robotics*, **11**, 175–182.

[14] Dellaert, F. & Beer, R. D. (1996), A developmental model for the evolution of complete autonomous agents.

[15] Di Paolo, E. A. (2003) Spike-timing dependent plasticity for evolved robots. *Adaptive Behavior*, **10**, 243–263.

[16] Dongyong, Y. & Yuta, S. (2002), Evolving mobile robot controllers using evolutionary algorithm.

[17] Dunn, N., Pierce-Shimomura, J., Conery, J., & Lockery, S. (2006) Clustered neural dynamics identify motifs for chemotaxis in caenorhabditis elegans. *Neural Networks, 2006. IJCNN '06. International Joint Conference on*, 0-0, pp. 547–554.

[18] Enquist, M. & Johnstone, R. A. (1997) Generalization and the evolution of symmetry preferences. *Proceedings of the Royal Society, Biological Science*, **264**, 1345–1348.

[19] Federici, D. (2005) Evolving developing spiking neural networks. *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, Sept., vol. 1, pp. 543–550 Vol.1.

[20] Federici, D. (2005) A regenerating spiking neural network. *Neural Networks*, **18**, 746 – 754, iJCNN 2005.

[21] Ferree, T. C. & Lockery, S. R. (1999) Computational rules for chemotaxis in the nematode c. elegans. *Journal of Computational Neuroscience*, **6**, 263–277.

[22] Filliat, D., Kodjabachian, J., & Meyer, J. (1999) Incremental evolution of neural controllers for navigation in a 6 legged robot. *Proceedings of the Fourth International Symposium on Artificial Life and Robotics*, **In Sugisaka and Tanaka, editors, Oita Univ. Press.**

[23] Floreano, D., Dürr, P., & Mattiussi, C. (2008) Neuroevolution: from architectures to learning. *Evolutionary Intelligence*, **1**, 47–62.

[24] Floreano, D., Husbands, P., & Nolfi, S. (2008) Evolutionary robotics. Siciliano, B. & Khatib, O. (eds.), *Springer Handbook of Robotics, Chapter 61*, Springer.

[25] Floreano, D. & Mattiussi, C. (2001) Evolution of spiking neural controllers for autonomous vision-based robots. *Proceedings of the international Symposium on Evolutionary Robotics From intelligent Robotics To Artificial Life*, **2217, T. Gomi, Ed. Lecture Notes In Computer Science Springer-Verlag, London, 38-61.**

[26] Florian, R. V. (2003) Biologically inspired neural networks for the control of embodied agents. *Technical report Coneural-03-03 Version 1.0.*.

[27] Florian, R. V. (2006), Spiking neural controllers for pushing objects around.

[28] Fontaine, B. & Peremans, H. (2009) Bat echolocation processing using first-spike latency coding. *Neural Networks*, **In Press, Corrected Proof**, –.

[29] Gauci, J. & Stanley, K. O. (2007), Generating large-scale neural networks through discovering geometric regularities.

[30] Gerstner, W. & Kistler, W. M. (2002) *Spiking Neuron Models. Single Neurons, Populations, Plasticity*. Cambridge University Press.

[31] Gerstner, W. (1999), The spike response model.

[32] Gilbert, S. F. (2006) *Developmental Biology, Eighth Edition*. Sinauer Associates, 8 edn.

[33] Gruau, F. (1993) Genetic synthesis on modular neural networks. Forrest, S. (ed.), *Proceedings of the 5th International Conference on Genetic Algorithms*, San Francisco, CA, USA, pp. 318–325, Morgan Kaufmann.

[34] Hänggi, P. (2002) Stochastic resonance in biology: how noise can enhance detection of weak signals and help improve biological information processing. *Chemphyschem*, **3**, 285–290.

[35] Hildebrand, J. G. & Shepherd, G. M. (1997) Mechanisms of olfactory discrimination: converging evidence for common principles across phyla. *Annu Rev Neurosci*, **20**, 595–631.

[36] Hopfield, J. (1999) Odor space and olfactory processing: Collective algorithms and neural implementation. *PNAS.*, **96**, 12506–12511.

[37] Hornby, G. S. & Pollack, J. B. (2001) Body-brain co-evolution using l-systems as a generative encoding. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, New York, NY: ACM.

[38] Hoshino, O., Kashimori, Y., & Kambara, T. (1998) An olfactory recognition model based on spatio-temporal encoding of odor quality in the olfactory bulb. *Biological Cybernetics*, **79**, 109–120.

[39] Huxley, A. & Hodgkin, A. (1952) The components of membrane conductance in the giant axon of loligo. *Journal of Physiology*, **1**, 473–496.

[40] Huxley, A. & Hodgkin, A. (1952) Currents carried by sodium and potassium ions through the membrane of the giant axon of loligo. *Journal of Physiology*, **1**, 449–472.

[41] Huxley, A. & Hodgkin, A. (1952) The dual effect of membrane potential on sodium conductance in the giant axon of loligo. *Journal of Physiology*, **1**, 497–506.

[42] Huxley, A. & Hodgkin, A. (1952) Measurement of current-voltage relations in the membrane of the giant axon of loligo. *Journal of Physiology*, **1**, 424–448.

[43] Huxley, A. & Hodgkin, A. (1952) A quantitative description of membrane currentand its application to conduction and excitiation in nerve. *Journal of Physiology*, **1**, 500–544.

[44] Ijspeert, A. & Kodjabachian, J. (1998) Evolution and development of a central pattern generator for the swimming of a lamprey. *Research Paper No 926*, **Dept. of Artificial Intelligence, University of Edinburgh**.

[45] Izhikevich, E. M. (2003) Simple model of spiking neurons. *IEEE Transactions on Neural Networks*.

[46] Izhikevich, E. M. (2004) Which model to use for cortical spiking neurons. *IEEE Transactions on Neural Networks*.

[47] Jacobi, N., Husbands, P., & Harvey, I. (1995), Noise and the reality gap: The use of simulation in evolutionary robotics. 648380 704-720.

[48] Kaiser, M., Hilgetag, C. C., & van Ooyen, A. (2009) A Simple Rule for Axon Outgrowth and Synaptic Competition Generates Realistic Connection Lengths and Filling Fractions. *Cereb. Cortex*, **19**, 3001–3010.

[49] Kandel, E. R., Schwartz, J. H., & Jessell, T. M. (2000) *Principles of Neural Science*. McGraw-Hill, 4th edn.

[50] Kanzaki, R., Nagasawa, S., & Shimoyama, I. (2005) Neural basis of odor-source searching behavior in insect brain systems evaluated with a mobile robot. *Chemical Senses*, **30**, 285–286.

[51] Kanzaki, R. (2007) How does a microbrain generate adaptive behavior? *International Congress Series*, **1301**, 7 – 14, brain-Inspired IT III. Invited and selected papers of the 3rd International Conference on Brain-Inspired Information Technology 2006 held in Hibikino, Kitakyushu, Japan between 27 and 29 September 2006.

[52] Khan, G., Miller, J., & Halliday, D. (2009) In search of intelligent genes: The cartesian genetic programming computational neuron (cgpcn). *Evolutionary Computation, 2009. CEC '09. IEEE Congress on*, May, pp. 574–581.

[53] Koch, C. (1999) *Biophysics of Computation: Information Processing in Single Neurons*. Oxford University Press: New York, New York.

[54] Kodjabachian, J. & Meyer, J. (1997) Evolution and development of modular control architectures for 1-d locomotion in six-legged animats.

[55] Kodjabachian, J. & Meyer, J. (1997) Evolution of a robust obstacle-avoidance behavior in khepera: A comparison of incremental and direct strategies. *IEEE transactions on neural network*, **9**.

[56] Kodjabachian, J. & Meyer, J. (1998) Evolution and development of neural controllers for locomotion, gradient-following, and obstacle-avoidance in artificial insects. *IEEE transactions on neural network*, **9**.

[57] Komosinski, M. & Rotaru-Varga, A. (2001) Comparison of different genotype encoding for simulated 3d agents. *Artificial Life*, **7**, 395–418.

[58] Kopp, A., Barmina, O., Hamilton, A. M., Higgins, L., Mcintyre, L. M., & Jones, C. D. (2008) Evolution of gene expression in the drosophila olfactory system. *Mol Biol Evol*, **25**, 1081–1092.

[59] Kosinski, R. & Zaremba, M. (2007) Dynamics of the model of the caenorhabditis elegans neural network. *Acta Physica Polonica B*, **38**, 2201–.

[60] Kreiman, G. (2004) Neural coding: computational and biophysical perspectives. *Physics of Life Reviews*, **1**, 71 – 102.

[61] Kuwana, Y. & Shimoyama, I. (1998) A pheromone-guided mobile robot that behaves like a silkworm moth with living antennae as pheromone sensors. *The International Journal of Robotics Research*, **17**, 924–933.

[62] Laing, D., Legha, P., Jinks, A., & Hutchinson, I. (2003) Relationship between Molecular Structure, Concentration and Odor Qualities of Oxygenated Aliphatic Molecules. *Chem. Senses*, **28**, 57–69.

[63] Laing, D., Panhuber, H., & Baxter, R. (1978) Olfactory properties of Amines and n-Butanol. *Chem. Senses*, **3**, 149–166.

[64] Laurent, G. (2002) Olfactory network dynamics and the coding of multidimensional signals. *Nature Reviews in Neuroscience*, **3**, 884–895.

[65] Laurent, G., Wehr, M., & Davidowitz, H. (1996) Temporal representations of odors in an olfactory network. *Journal of Neuroscience*, **16**, 3837–3847.

[66] L'Etoile, N. D. & Bargmann, C. I. (2000), Olfaction and odor discrimination are mediated by the c. elegans guanylyl cyclase odr-1.

[67] Mattiussi, C. & Floreano, D. (2007) Analog genetic encoding for the evolution of circuits and networks. *IEEE transactions on evolutionary computation*, **11**, 596–607.

[68] MCCullogh, W. S. & Pitts, W. (1943) A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, **5**, 115–133.

[69] Meinhardt, H. (1982) *Models of Biological Pattern Formation*. Academic Press.

[70] Miconi, T. & Channon, A. (2005) A virtual creatures model for studies in artificial evolution. al., D. C. e. (ed.), *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2005)*, Edinburgh, UK, IEEE Press.

[71] Miconi, T. & Channon, A. (2006) An improved sytem for artificial creatures evolution. Rocha, L. e. a. (ed.), *Proceedings of the 10th Conference on the simulation and synthesis of living systems (ALIFE X)*, Bloomington, Indiana, USA, MIT Press.

[72] Moore, P. A. (1994) A model of the role of adaptation and disadaptation in olfactory receptor neurons: implications for the coding of temporal and intensity patterns in odor signals. *Chem. Senses*, **19**, 71–86.

[73] Mori, T. & Kai, S. (2002) Noise-induced entrainment and stochastic resonance in human brain waves. *Physical Review Letters*, **88**, 218101, copyright (C) 2007 The American Physical Society Please report any problems to prolaaps.org PRL.

[74] Moss, F., Ward, L., & Sannita, W. (2004) Stochastic resonance and sensory information processing: a tutorial and review of application. *Clin Neurophys*, **27**, 677– 682.

[75] Nolfi, S. & Floreano, D. (2000) Encoding, mapping, and development. *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines, Chapter 9.*, pp. 223–240, MIT Press/Bradford Books.

[76] Nolfi, S., Miglino, O., & Parisi, D. (1994) Phenotypic plasticity in evolving neural networks. Tech. rep., Technical Report PCIA-94-05, Institute of Psychology, C.N.R.

[77] Nowotny, T., Huerta, R., Abarbanel, H., & Rabinovich, M. (2005) Self-organization in the olfactory system: one shot odor recognition in insects. *Biological Cybernetics*, **93**, 436–446.

[78] Palmer, A. R. (2004) Symmetry breaking and the evolution of development. *Science*, **306**, 828–833.

[79] Payton, D., Daily, M., Estowski, R., Howard, M., & Lee, C. (2001) Pheromone robotics. *Auton. Robots*, **11**, 319–324, 591897.

[80] Pfeifer, R., Iida, F., & Bongard, J. (2005) New robotics: Design principles for intelligent systems. *Artificial Life, Special Issue on New Robotics, Evolution and Embodied Cognition*, **11**, 99–120.

[81] Pyk, P., Bermúdez i Badia, S., Bernardet, U., Knüsel, P., Carlsson, M., Gu, J., Chanie, E., Hansson, B., Pearce, T., & J. Verschure, P. (2006) An artificial moth: Chemical source localization using a robot based neuronal model of moth optomotor anemotactic search. *Autonomous Robots*, **20**, 197–213, 10.1007/s10514-006-7101-4.

[82] Rudolph, M. & Destexhe, A. (2003) Characterization of subthreshold voltage fluctuations in neuronal membranes. *Neural Computation*, **15**, 2577–2618, 948895.

[83] Ruppin, E. (2002) Evolutionary autonomous agents: A neuroscience perspective. *Nature Reviews Neuroscience*, **3**, 132–141.

[84] Rust, A. G., Adams, R., George, S., & Bolouri, H. (1997), Designing development rules for artificial evolution.

[85] Semmelhack, J. L. & Wang, J. W. (2009) Select drosophila glomeruli mediate innate olfactory attraction and aversion. *Nature*, **459**, 218–223.

[86] Sims, K. (1994) Evolving 3d morphology and behavior by competition. Brooks, R. & Maes, P. (eds.), *Proceedings of the 4th International Workshop on the Synthesis and Simulation of Living Systems (ALIFE IV)*, pp. 28–39, MIT Press.

[87] Stanley, K. O. (2007) Compositional pattern producing networks: A novel abstraction of development. *Genetic Programming and Evolvable Machines Special Issue on Developmental Systems*, **8**, 131–162.

[88] Stanley, K. O. & Miikkulainen, R. (2002) Evolving neural networks through augmenting topologies. *Evolutionary Computation*, **10**, 99–127.

[89] Stanley, K. O. & Miikkulainen, R. (2003) A taxonomy for artificial embryogeny. *Artificial Life journal*, **9**, 93–130.

[90] Stensmyr, M., Giordano, E., Balloi, A., Angioy, A., & Hansson, B. (2003) Novel natural ligands for drosophila olfactory receptor neurones. *J Exp Biol*, **206**, 715–724.

[91] Stryer, L. (1988) *Biochemistry*. W. H. Freeman and Company, New York., third edition edn.

[92] Thompson, D. (1917) *On Growth and Form*. Cambridge University Press (1992).

[93] Uhlenbeck, G. E. & Ornstein, L. S. (1930) On the theory of brownian motion. *Physical Review*, **36**, 823–41.

[94] Vaario, J., Onitsuka, A., & Shimohara, K. (1997) Formation of neural structures. *in Proceedings of the Fourth European Conference on Artificial Life, ECAL97*, pp. 214–223, The MIT Press.

[95] Webb, B. (1998) Robots crickets and ants: models of neural control of chemotaxis and phonotaxis. *Neural Networks*, **11**, 1479–1496.

[96] Wes, P. D. & Bargmann, C. I. (2001) C. elegans odour discrimination requires asymmetric diversity in olfactory neurons. *Nature*, **410**, 698–701.

[97] Wesson, D. W. W., Carey, R. M. M., Verhagen, J. V. V., & Wachowiak, M. (2008) Rapid encoding and perception of novel odors in the rat. *PLoS biology*, **6**.

[98] Weyl, H. (1952) *Symmetry*. Princeton Press.

[99] Wiesenfeld, K. & Moss, F. (1995) Stochastic resonance and the benefits of noise: from ice ages to crayfish and squids. *Nature*, **373**, 33–36.

[100] Wilensky, U. (1999), Netlogo, http://ccl.northwestern.edu/netlogo/resources.shtml.

[101] Wilson, D. & Stevenson, R. (2006) *Learning to Smell: Olfactory Perception from Neurobiology to Behavior*, pp. 64–75.

[102] Wilson, R. & Mainen, Z. (2006) Early events in olfactory processing. *Annu Rev Neurosci*, **29**, 163–201.

[103] Wolpert, L. (2005) Development of the asymetric human. *European Review*, **13**, 97–103.

[104] Wolpert, L., Smith, J., Jessell, T., & Lawrence, P. (2007) *Principles of Development*. Oxford University Press, third edn.

[105] Wyatt, T. (2003) *Pheromones and Animal Behaviour, Communication by Smell and Taste*. Cambridge University Press.

[106] Yao, X. (1999) Evolving artificial neural networks. *Proceedings of the IEEE*, **87**, 1423–1447.

[107] Zhang, Y., Lu, H., & Bargmann, C. I. (2005) Pathogenic bacteria induce aversive olfactory learning in caenorhabditis elegans. *Nature*, **438**, 179–184.

[108] Zuckerman, B. M. & Jansson, H. (1984) Nematode chemotaxis and possible mechanisms of host/prey recognition. *Annual Review of Phytopathology*, **22**, 95–113.

Appendix A

Development in Biological Systems

Introduction

Every living organisms have the same basic properties and undergo the same major biological processes. They have a metabolism which allows them to grow during development and maintain homeostasis (stable state). The also reproduce (sexually or asexually) and evolve through the means of natural selection.

Development is a natural process responsible for creating and modifying an organism during its lifetime. It results from the coordinated behaviours of cells [wolpert]. Every living creature is an ensemble of cells interacting with each other. The behaviour of a cell is controlled by genes containing information on where and when to produce (synthesize) proteins. These proteins affect cells from various ways depending on their nature: they affect their genetic expression (which gene is to be "executed"), modify their shape, change signals they produce or respond to, regulate how fast they proliferate, and change how they migrate (move). Gene expression is the the main underlying process of development as it guides how a genotype (genes) gives rise to a particular phenotype (trait of an organism). As all the cells composing a living creature generally carry the same genetic code, the changes that occur during development are due to differences of gene expression by different cells. This allow cells to specialize and become skin cells, heart cells or nerve cells as neurons for example.

In this chapter, I will present the major processes that occur during development nature. As my research is partly about artificial development in sexually reproducible agents, I will omit asexual reproduction and focus on development occurring after sexual reproduction in animals. I will also present breiefly how symmetrical and asymmetrical forms arise during development in living organisms. Then, I will introduce some work that has been done in artificial life and robotics and conclude this chapter.

Embryogenesis

Development begins once reproduction has occurred. In many sexually reproducible organisms, it starts when a **gamete** cell (a sperm cell) fuses with another gamete cell (an egg) during **fertilization** (also called syngamy). From this moment, the **zygote** (fertilized egg) will divide into smaller cells forming an organism defined as an **embryo**. This process is called **cleavage**. The development of the embryo, taking place from fertilization to birth, is defined as **embryogenesis**. In humans, an embryo is called a fetus nine weeks after fertilization.

The zygote divides to creates **germs cells**. Germs cells are fundamental for heredity. They can produce other germs cells that will become gametes and they can also create **somatic cells** that are the building blocks of the body. The main difference between germ cells and somatic cells is that only germ cells will transmit genetic information to offsprings. Any characteristics that the body may have acquired during lifetime is not transmitted to the germline, therefore not transmitted to offsprings. Only gametes transmit genetic information from parents to offsprings.

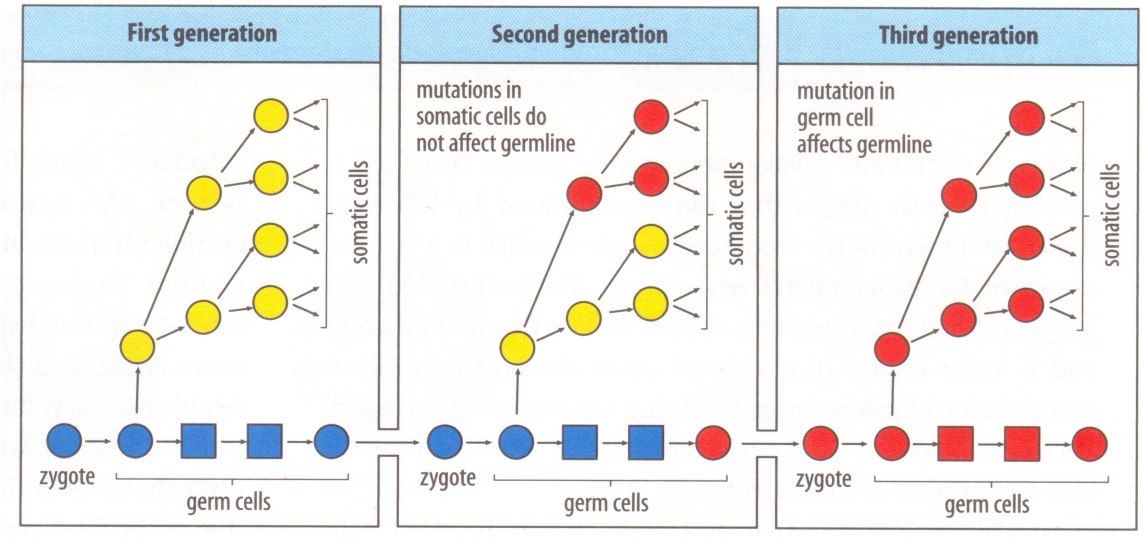


Figure 8.1: The distinction between germ cells and somatic cells. In each generation germ cells give rise to both somatic cells and germ cells, but inheritance is through the germs cells only. Changes that occur due to mutation in somatic cells can be passed on to their daughter cells but do not affect the germline. Figure and caption from Lewis Wolpert [104].

In many animals, germ cells and somatic cells are **diploid** which means that they contain two copies of each **chromosomes** (pieces of DNA containing genetic material). However, gametes are have only one copy of each chromosome (**haploid**). The process involved in the creation of gametes is a reduction division called **meiosis**. During this process, a germ cell divides and create gametes

that have only half of the whole set of chromosomes.

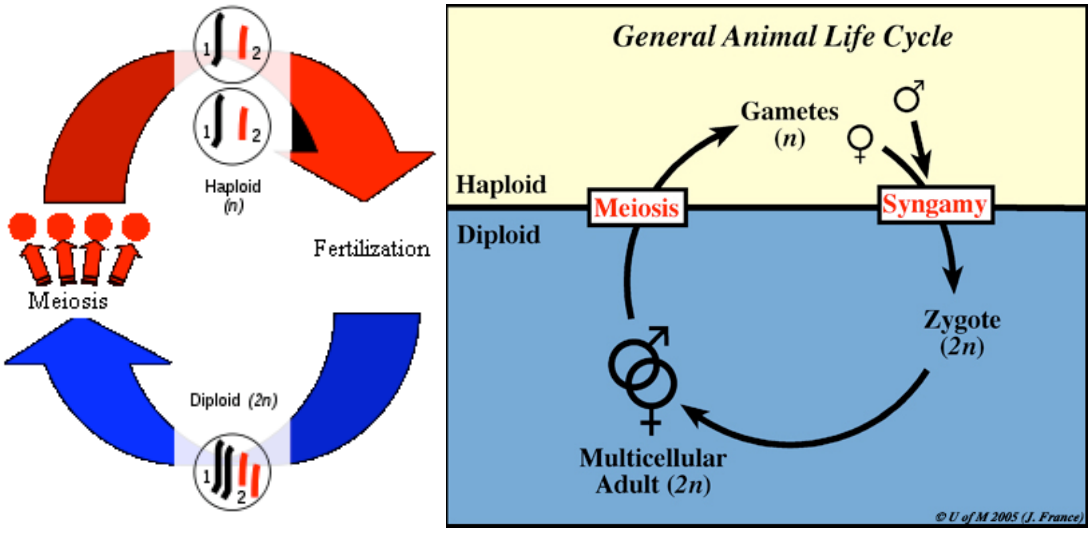


Figure 8.2: Sexual reproduction. In the first stage of sexual reproduction, "meiosis," the number of chromosomes is reduced from a diploid number (2n) to a haploid number (n). During "fertilization," haploid gametes come together to form a diploid zygote and the original number of chromosomes (2n) is restored. Figure and caption from: http://en.wikipedia.org/wiki/Sexual_reproduction

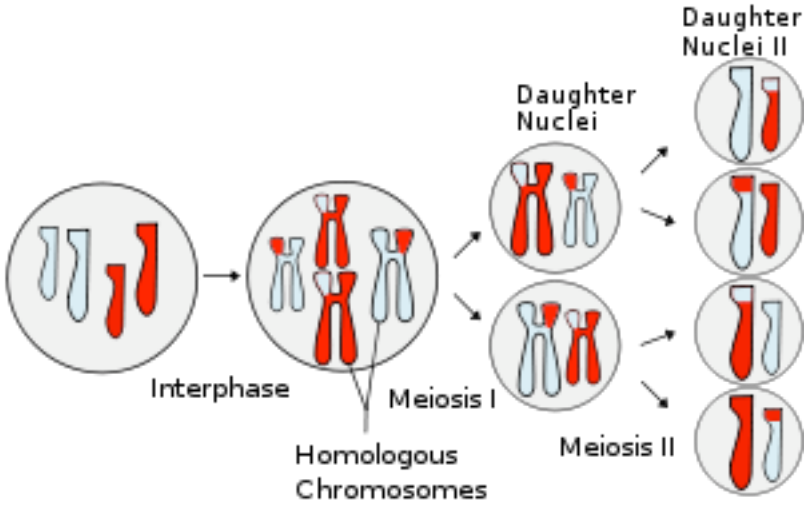


Figure 8.3: Events involving meiosis, showing chromosomal crossover. Meiosis is a reductional division in which the number of chromosomes per cell is halved. Figure and caption from: http://en.wikipedia.org/wiki/Meiosis

During sexual reproduction, gametes cells come from two different individuals and each one carries a different sets of chromosomes. After fertilization, once both gametes have fused, each one transmit its set of chromosomes. Therefore, the zygote being created by this process is diploid as

it contains a set of chromosomes coming from each parent. Consequently, the zygote inherits from both parents but also differs from both.

During lifetime, every living organism undergo some major developmental processes.

Major Processes of Development

There are many processes involved in development. I present in this section the main ones that are not independent or strictly sequential.

Cell division

This process can take place during early development but also later during cells proliferation and tissue growth [104]. **Cleavage** is the division of the zygote into smaller cells that do not grow. The only processes that take place during cleavage are DNA replication (creates a copy of genetic material), mitosis (separation of the chromosomes), and cell division (the two daughter cells have similar sets of chromosome). Cells also divide during cells proliferation and tissue growth, however they can change ,ass as they can grow compared to cells that divide during cleavage.

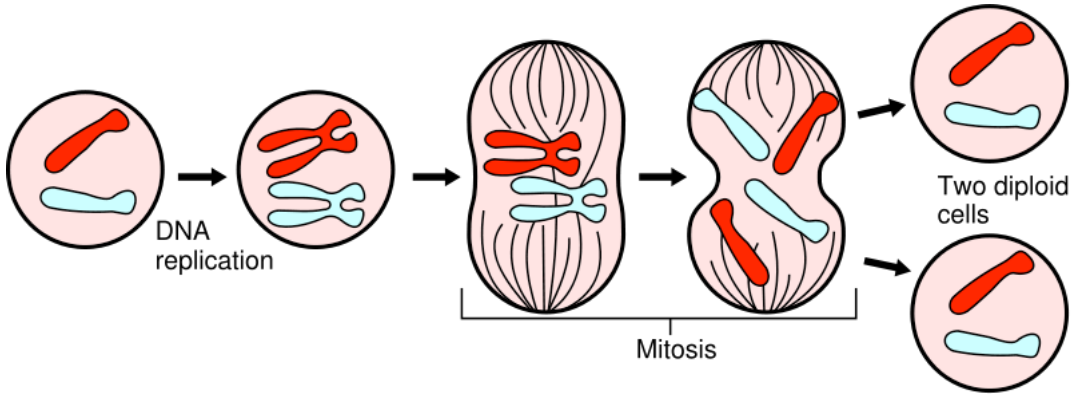


Figure 8.4: Cleavage of a cell. Figure from http://en.wikipedia.org/wiki/Mitosis.

Cell differentiation

In this process, cells are specializing as they become structurally and functionally different from each others. Cell differentiation is a gradual process and cells often divide many times before they are fully differentiated. In humans, a zygote give rise to at least 250 types of cell [104]. Cell differentiation and pattern formation are closely related. Cells will differentiate differently depending on the body plan. A good example of this relation is the distinction between arms and legs in humans. Both contain exactly the same types of cells, however they are arranged in a different way.

Pattern formation

During development, cells divide and migrate to different locations at different times. By doing so, they create a well organised structure where each cell "knows" whether to become an arm cell or an eye cell for example. Pattern formation is the consequence of cellular and molecular mechanisms in different organisms at different stages of development [104]. The initial phase of patterning is the creation of a **body plan** which set up the main axes of the body in order to define where the head (anterior) and tail (posterior) should be, and defined the underside (ventral) and back (dorsal). Many animals have a head at one end and a tail at the other and have a left and right side of the body being **bilaterally symmetrical,** which means that they are a mirror image of each other [104]. These bilateral symmetrical animals form the sub-group named Bilateria of the major group: the animal kingdom (Animalia). They all have a main body axis called **antero-posterior axis** (from head to tail), and a **dorso-ventral axis** (back to belly). Quite often, eggs shows a distinct **polarity** meaning that one end differs from the other. This polarity in an egg could possibly be the origin of the main body axis that will be generated after fertilization.

An important process taking place during pattern formation is the allocation of cell to different **germ layers**. As the cells divide and migrate to different locations, they are acquiring different characteristics so some will specialized and become part of a muscle, skin or neurons for example. The process of specialization is called **cell differentiation** and will be described later in this chapter. In many animals, three germ layers are created: endoderm, mesoderm, ectoderm. These three different germ layers give rise to every organ in a fully developed body.

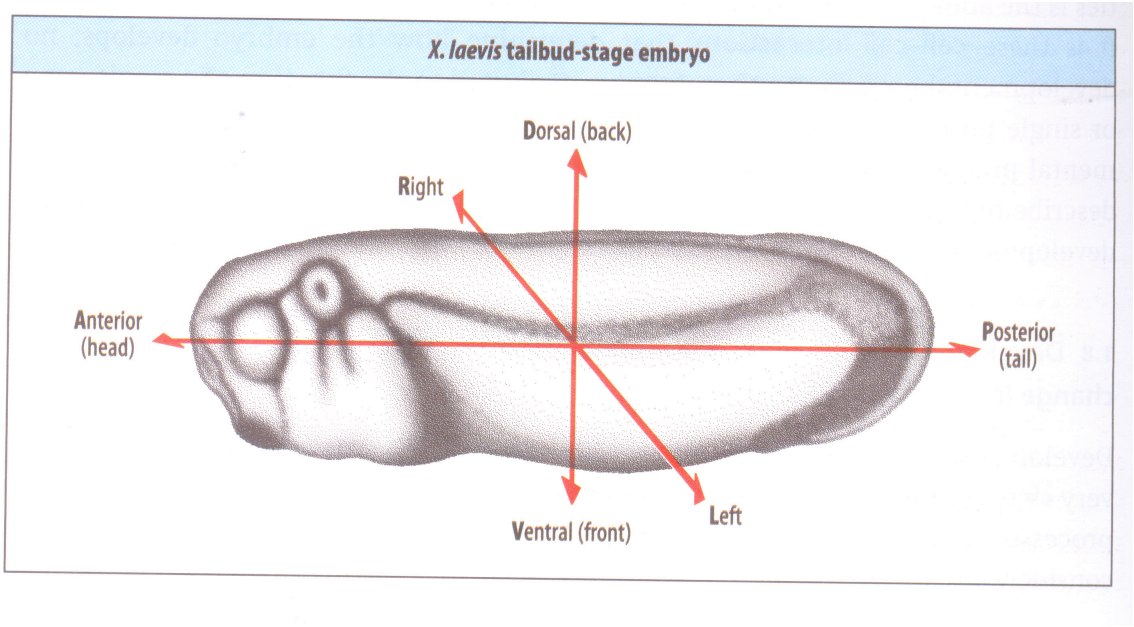
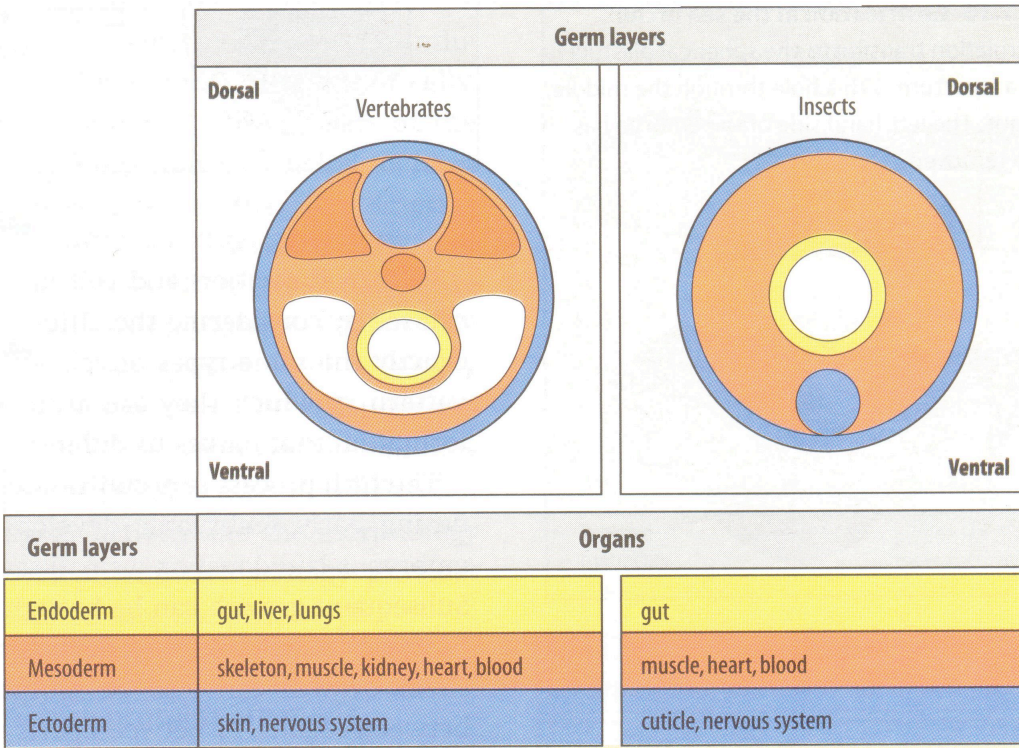


Figure 8.5: The main axis of a developing embryo. The anterio-posterior axis and the dorso-ventral axis are at right angles to one another, as in a coordinate system. Figure and caption from [104].



Germ layers	Organs	
Endoderm	gut, liver, lungs	gut
Mesoderm	skeleton, muscle, kidney, heart, blood	muscle, heart, blood
Ectoderm	skin, nervous system	cuticle, nervous system

Figure 8.6: Germ layers and organs they give rise to, in most vertebrates and insects. Figure from [104].

Pattern formation can also involve the interpretation of **positional information**. This means that cells can "know" where they are and differentiate according to their position in the body. One of the mechanism used to specifiy positions to cell is by using gradient of chemicals. These chemicals involved in pattern formation called **morphogens**, can decrease concentration from one end to another end in order to have positions encoded as concentration levels. The cells can respond differently to certain levels of concentration where **threshold concentration** can induce different behaviours. Depending on these thresholds, a cell can differentiate differently. "Threshod concentrations can represent the amount of morphogen that must bind to receptors to activate an intra-celllar signalling system, or concentrations of transcription factors requied to activate particular genes."[104] The French flag model is a good example of how positional information can be used. Positional information can also be specified by timing mechanisms or direct intercellular interactions.

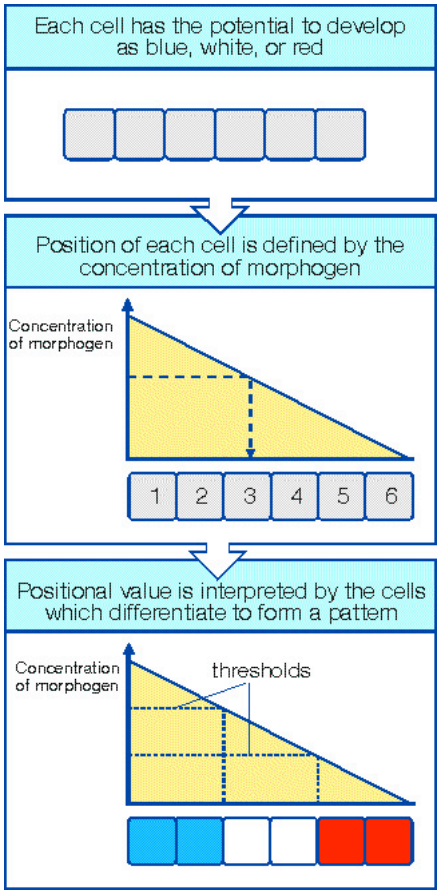


Figure 8.7: The French flag model of pattern formation. Each cell in a line of cells has the potential to develop as blue, white, or reed. The line of cells is exposed to a concentration gradient of some substance and each cell acquires a positional value defined by the concentration at that point. Each cell then interprets the positional value it has acquired and differentiates into blue, white, or red, according to a predetermined genetic program, thus forming the French flag pattern. Substances that can direct the development of cells in this way are known as morphogens. The basic requirements of such a system are that the concentration of substance at either end of the gradient must remain different from each other but constant, thus fixing boundaries to the system. Each cell must also contain the necessary information to interpret the positional values. Interpretation of the positional value is based upon different threshold responses to different concentrations of morphogen. Figure and caption from [104].

Another mechanism that can used for positional specification is based on **cytoplasmic localization** and **asymmetric cell division**. This type of division results in having daughter cells different from each other. Their differences does not depend on environmental cues but on their lineage. Asymmetric cell division hapens when cytoplasmic **determinants** like proteins are not distributed

evently in an egg or a cell. Therefore, when this cell or egg divides, the daugther cells will inherit different determinants that will play an important role in the cell behaviour.

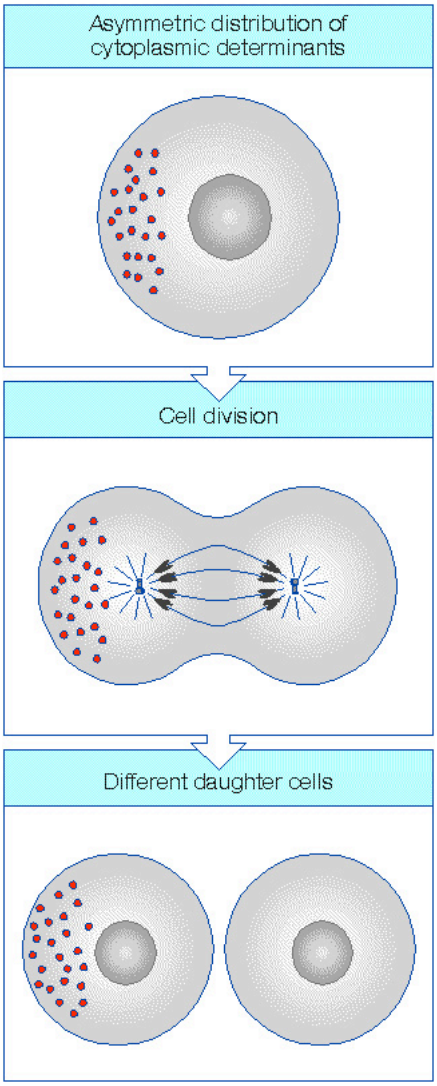


Figure 8.8: Cell division with asymmetric distribution of cytoplasmic determinants. If a particular molecule is distributed unevently in the parent cell, cell division will rsults in its being shared unequally between the cytoplasm of the two daughter cells. The more localized the cytoplasmic determinant is in the parent cell, the more likely it will be that one daughter cell will receive all of it and the other none, thus producing a distinct difference between them. Figure and caption from [104].

Most of the time however, cells become different one from another due to changes of their environment (presence of different morphogens for example) and cell to cell interaction called **cell signaling**. As a matter of fact, cells communicate and can send signals between each other in order to influence the development of adjacent cells. This process is called **induction**. Different types of

induction exist where cells can respond concentration levels of a signal or not.

One of the important mechanisms involve in pattern formation is **lateral inhibition**. Certain cells differentiate and secrete an inhibitory molecule around them preventing cells in their neighborhood to do so. This mechanism allow cells to be more or less regularly spaced with respect to one another.

Morphogenesis

This major developmental process involve changes of form. As it develops, an embryo undergo major changes in form. One of the main changes is called **gastrulation** during which the main body plan emerge and gut is formed. Cell migration is often involve in morphogenesis. For example, most the cells of the human face are derived from cells that come from the back of the embryo [104].

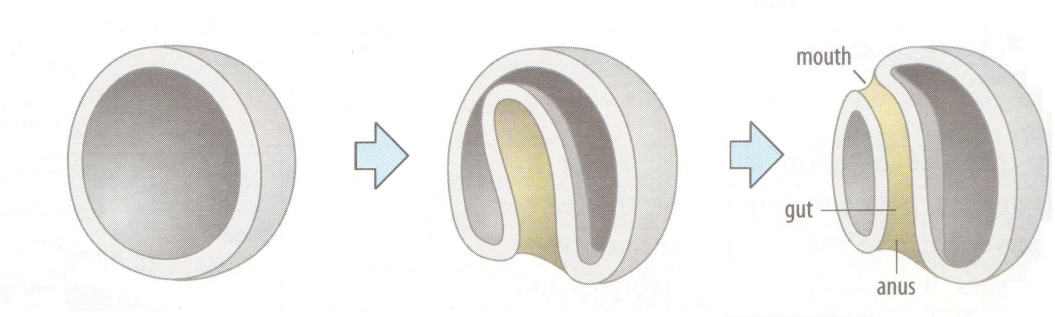


Figure 8.9: Gastrulation in the sea urchin. Gastrulation transforms the spherical blastula into a structure whith a hole through the middle, the gut. The left-hand side of the embryo has been remove. Figure and caption from [104].

Growth

Growth means an increase of size. During development, the overall body grows as the cells composing it can multiply, increase their size, and depose extracellular material such as bone. Time is really important in growth as different rates of growth between organs can generate major changes in the shape of the embryo. Therefore, growth can also be morphogenetic.

Cell migration

This process involve the movement or migration of cells from one place to another. They often performing **chemotaxis** as they migrate to a location depending on the pattern of concentration of certain chemicals.

Cell death

Death plays an important role during development as certain cells are "programmed" in the genome to die at certain stages. Cell death, called **apoptosis**, is partly responsible for the creation of fingers and toes from a continuous piece of tissues.

Gene expression and protein synthesis

As I mentioned earlier, the behaviour of cells is controlled by genes that "decide" when and where proteins are created. The behaviour of cells resulting in development depends largely on this proteins. Therefore gene expression plays a major role in development. Pattern formation during early development is mainly due to changes of patterns of gene activity. A gene create a particular protein when it is "switched on" so it can be **transcribed** (copied) into messenger RNA (mRNA), and this mRNA is then **translated** into protein. Certain genes do not code for proteins. They can encode RNA molecules like rRNA (ribosomal RNA), tRNA (transfer RNA) or miRNA (micro RNA). Certain miRNAs are involved in the regulation of genes in development. Certain proteins encoded by certain genes also regulate the expression of other genes. **Gene regulation** is the process by which certain genes are turned on and other are turned off. Not all of the genome is made of developmental genes. There are DNA sequences that regulate gene expression are called **control region** or **regulatory region** and are usually situated close to the gene they control. A gene is switched on when a gene-regulatory protein called **transcription factor** binds to the **promoter region** or the **enhancer region** of its control region. Usually, developmental genes are controlled by a regulatory region that has more than one **regulatory module** called **cis-regulatory module**, and each module has multiple binding sites where different types of transcription factors can bind to. These complex control regions allow a developmental gene to be switched on and off at the different times and places during development [wolpert]. Gene regulation is very complex as some genes can share the same **regulatory module** and other can have only some of the binding sites of a **regulatory module** in common. "Thus, an organism's genes are linked in complex interdependent networks of expression through their regulatory modules and the proteins that bind to them." [104] These networks are called **genetic regulatory networks** (GRNs).

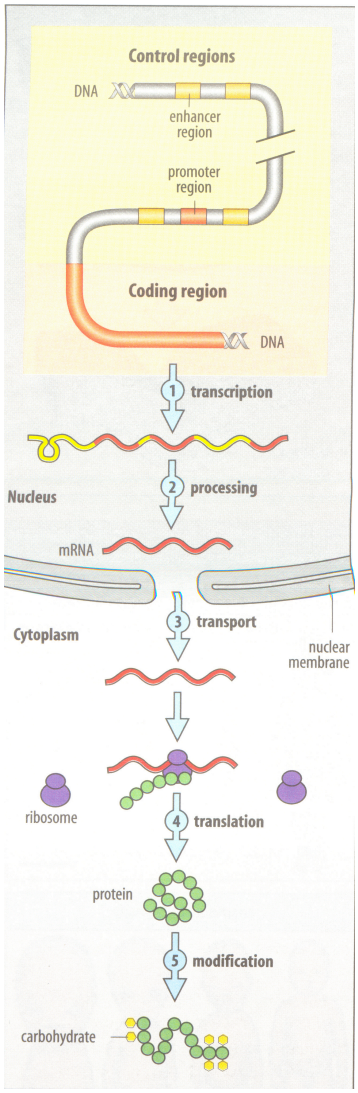


Figure 8.10: Gene expression and protein synthesis. A protein-coding gene comprises a stretch of DNA that contains a coding region, which contains the instruction for making the protein, and adjacent control regions-promoter and enhancer regions- at which the gene is switched on or off. The promoter region is the site at which RNA polymerase binds and starts transcribing. The enhancer may be thousands of base pair distant from the promoter. Transcription of the gene into RNA (1) may be either stimulated or inhibited by transcription factors that bind to promoter and enhancer regions. The RNA formed by transcription is sliced to remove introns (yellow) and processed within the nucleus (2) to produce mRNA that is exported to the cytoplasm (3) for translation into protein at the ribosomes (4). Control of gene expression and protein synthesis occurs mainly at the level of transcription but also occurs at later stages. For example, mRNA may be degraded before it can be translated. If it is not translated immediately it may be stored in inactive form in the cytoplasm for translation at some later stage. Some proteins require post-translational modification (5) to become biologically active. Figure and caption from [104].

Appendix B

Published Papers

Optimal receptor response functions for the detection of pheromones by agents driven by spiking neural networks

Nicolas Oros, Volker Steuber, Neil Davey, Lola Cañamero, Rod Adams
Science and Technology Research Institute
University of Hertfordshire
AL10 9AB
United Kingdom
{N.Oros, V.Steuber, N.Davey, L.Canamero, R.G.Adams}@herts.ac.uk

Abstract

The goal of the work presented here is to find a model of a spiking sensory neuron that could cope with small variations in the concentration of simulated chemicals and also the whole range of concentrations. By using a biologically plausible sigmoid function in our model to map chemical concentration to current, we could produce agents able to detect the whole range of concentration of chemicals (pheromones) present in the environment as well as small variations of them. The sensory neurons used in our model are able to encode the stimulus intensity into appropriate firing rates.

1 Introduction

In this study, we want to investigate the encoding of information about pheromones in spiking neural networks controlling artificial agents. Initially, the pheromones are diffused symmetrically from a point source (Figure 1). In order to create pheromone sensing agents, we need to decide which kind of sensory neurons we want to use. To model the sensory neurons in a biological plausible way and to be able to explore different encoding strategies, we used spiking neurons. One challenge of using a spiking neural network is to decide the coding to use in order to map information received by a sensor that will transform these stimuli into spikes.

Different coding strategies can be used [Floreano and Mattiussi, 2001]:

a) Mapping stimulus intensity to the firing rate of the neuron.
b) Mapping stimulus intensity onto the number of neurons firing at the same time.
c) Mapping stimulus intensity onto the firing delay of the neuron.

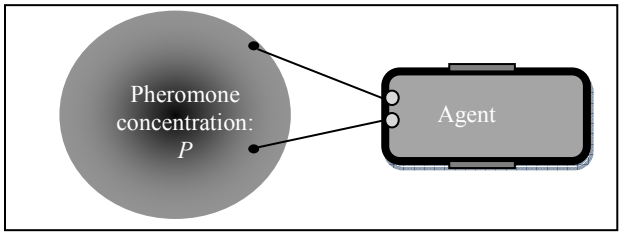


Figure 1. An agent equipped with two wheels and two antennae linked to two sensory neurons used to detect pheromones.

In order to use one of these encoding schemes, one needs first to decide how the input current of a sensor should represent its stimulus intensity. The current received by a sensor will be different from the one received by a non-sensory neuron because it will be based on the external stimulus intensity and not the activity of other neurons or sensors. We want an agent to be able not only to detect small variations of pheromone concentration but also the whole range of concentrations. Therefore, the agents must be equipped with sensory neurons that can produce spike trains at different frequencies depending on the pheromone concentration. The ideal case would be to have a linear relationship between the pheromone concentration and the firing rate of the sensory neuron. Such relationships exist in biological systems. For example in humans, the relationship between the frequency of firing of sensory neurons and pressure on the skin is linear [Kandel *et al.*, 2000]. We tried to find out how to implement such a relationship by carrying out different experiments using different expressions for the sensory neuron's current.

2 Experiments

We modelled a sensory neuron as a leaky integrate-and-fire neuron and tried different equations to calculate its input current. The sensory current I was always calculated depending on the pheromone concentration P. If the membrane potential, which depends on the current I, reaches a certain threshold θ the sensory neuron emits a spike. Therefore, the firing rate of the sensory neuron

depends on the relation between pheromone concentration and current (Figure 2). In our experiments, we tried many different functions relating the pheromone concentration P to the current I in order to obtain a desired quasi-linear relationship between the pheromone concentration and the firing rate of the sensory neuron.

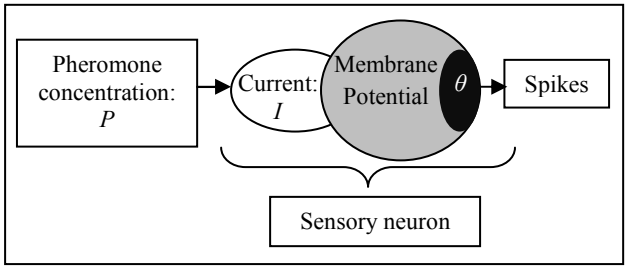


Figure 2. Mapping pheromone concentration into spikes.

We first set the sensor's input to a concentration of 1 and we recorded at what time a spike was emitted in order to determine the frequency (firing rate). We applied the same method to study the firing rate of the sensors over the whole range of pheromone concentration up to some maximum (that we chose to be 300). We did not want the sensory neuron to fire if the concentration was equal to 0 so only the presence of pheromones could stimulate a sensor. Afterwards, we modelled each different kind of sensory neuron as part of an agent and looked at the agent's behaviour.

2.1. Linear relationship between current and pheromone concentration

We first carried out experiments implementing a simple linear relationship, expressed by Equation (1), between the pheromone concentration P and the current I (Figure 3.a) and studied the sensor's firing rate (Figure 3.b.).

$$I = KP \qquad (1)$$

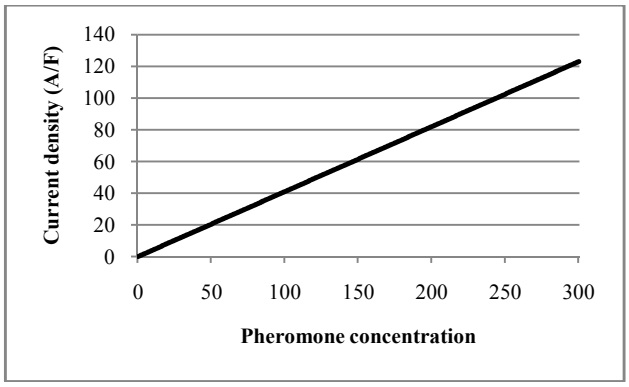


Figure 3.a. Current density (in Ampere per Farad) input to sensory neuron using Equation (1) with K=0.41.

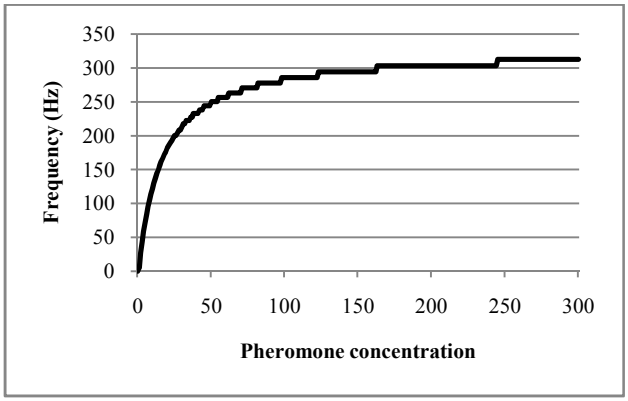


Figure 3.b. Resulting firing rate of sensory neuron. The maximum firing rate of a neuron is around 300 Hertz.

After a few experiments using different values for K, we realized that the sensor was saturating (Figure 3.b) due to the nature of the sensory neuron (leaky integrate-and-fire [Koch, 1999]). In fact, above a small value of pheromone concentration, the current produced was too high and the sensor fired at its maximum rate. After implementation in the agent, we saw that it was not able to detect the difference between a concentration of 200 and 250 for example.

2.2. Linear relationship with offset between current and pheromone concentration

Then, we tried to use the same equation but with an added baseline current and a much smaller slope (K_2) (Equation (2) and Figure 4.a). We made these changes knowing that our sensor responds to a small range of currents with a large bandwidth.

$$I = K_1 + K_2 P \qquad (2)$$

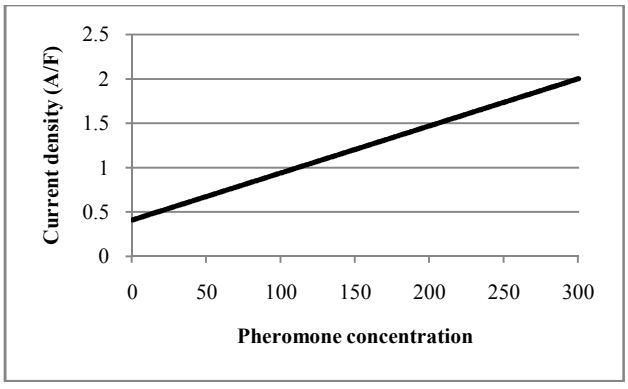


Figure 4.a. Current density input to sensory neuron using Equation (2) with K_1 = 0.41 and K_2 = 0.0053. Note that in this graph, the ordinate scale is different than in Figure 3.a and the current density is very low.

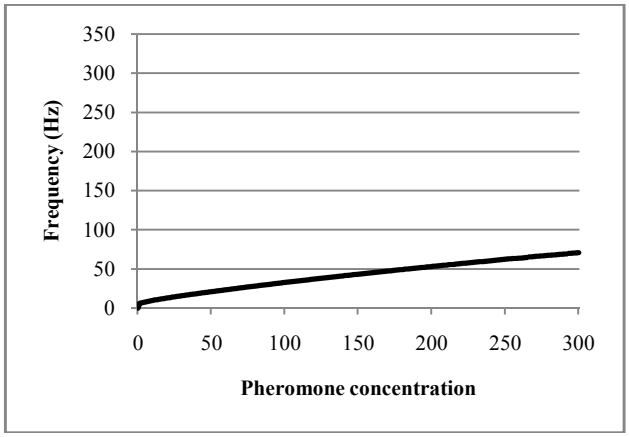


Figure 4.b. Resulting firing rate of sensory neuron.

With this equation, we had a more linear relationship between pheromone concentration and the firing rate of the sensor (Figure 4.b) so the agent should have been able to detect smaller variations. Unfortunately, the sensor did not use its whole bandwidth and its resolution was relatively poor. Therefore, another kind of functional relationship had to be tried.

2.3. Non-linear relationship between current and pheromone concentration

Concerning the neurons we are using, we know the limits of currents and the corresponding firing rate. For every cell (motorneurons, sensors, and interneurons):
$I_{min} \approx 0.4$ ($f \approx 0.6\ Hz$)
$I_{saturation} \approx 20$ ($f \approx 300\ Hz$)

We also know that the firing rate of a leaky integrate-and-fire neuron is given by [Koch, 1999]:

$$< f > = \frac{1}{t_{th}+t_{ref}} = \frac{1}{t_{ref}-\tau ln\left(1-\frac{V_{th}}{IR}\right)} \qquad (3)$$

Where:

- t_{th} is the mean time to reach the threshold value
- V_{th} is the threshold voltage (a spike is emitted if the membrane potential is above this value).
- t_{ref} is the refractory period.
- I is the current
- R is the resistance (constant)
- C is the capacitance (constant)
- $\tau = RC$ (time constant)

Given that our sensory neuron is modelled as a leaky integrate-and-fire neuron, we rearranged Equation (3) to find an equation (4) for the current (Figure 5.a) that would always produce a linear relationship between the pheromone concentration and the firing rate of the sensory neuron (Figure 5.b).

$$I = \frac{V_{th}}{R}\left[\frac{1}{1-exp\left(\frac{t_{ref}}{\tau}-\frac{1}{<f>\tau}\right)}\right] \qquad (4)$$

To get a linear relationship between the current and the pheromone concentration, we replaced $< f >$ by P and to make sure the frequency was between 0 and 300, we needed:

- $\frac{V_{th}}{R} = 0.4\ mV/\Omega$
- $t_{ref} = 3/1000 = 0.003\ s$
- $\tau = 1/20 = 0.05\ s$

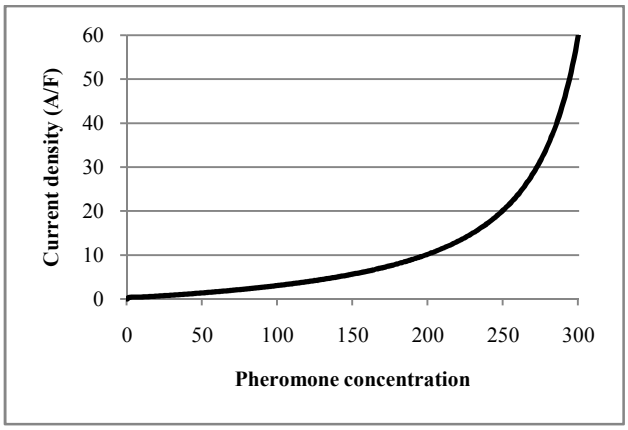


Figure 5.a. Current density input to sensory neuron using Equation (4).

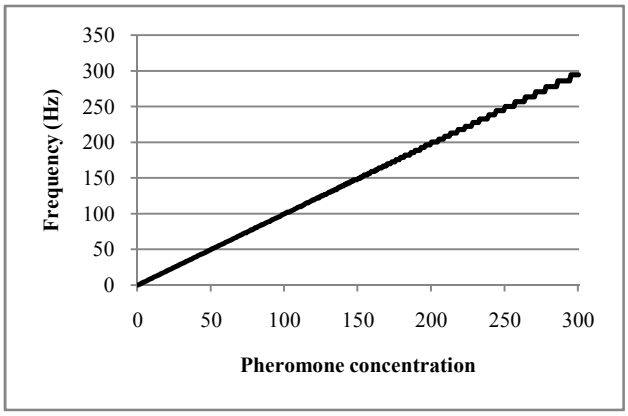


Figure 5.b. Resulting firing rate of sensory neuron.

With this equation, an agent is able to detect a small variation in the pheromones concentration using its whole bandwidth. We created Equation (4) artificially but we can use it as a guide to look for an equation commonly found in biological systems that describes a similar relationship and a graph similar to Figure 5.a.

2.4. Hill functions

We know that pheromones and other odours bind to receptor proteins situated in an animal's olfactory sensory neurons [Wyatt, 2003]. The current generated by the sensory neurons depends on their binding capacity. We

first investigated an equation used by biochemists describing the binding of ligand molecules to proteins: a Hill function [Stryer, 1988].

$$h(x,k,m) = \frac{x^m}{k^m + x^m} \quad (5)$$

Where:

- k is the concentration of molecules when h is equal to 0.5
- m is the Hill coefficient and is considered as an estimate of the number of binding sites of a protein.
- x is the concentration of ligands

Archibald Hill used this equation in 1910 to describe the binding of oxygen to Hemoglobin. It seems appropriate to use Hill functions to describe the shape of the current produced by the sensor as they are very similar to Equation (4).

The first Hill function (6) we used was too simple to fit the function (4). An example with m = 1, K_1 = 50 and K_2 = 100 is given in Figures 6.a. and 6.b. Once again, we realized that the sensor was saturating quite rapidly.

$$I = K_1 \left(\frac{P^m}{K_2^m + P^m} \right) \quad (6)$$

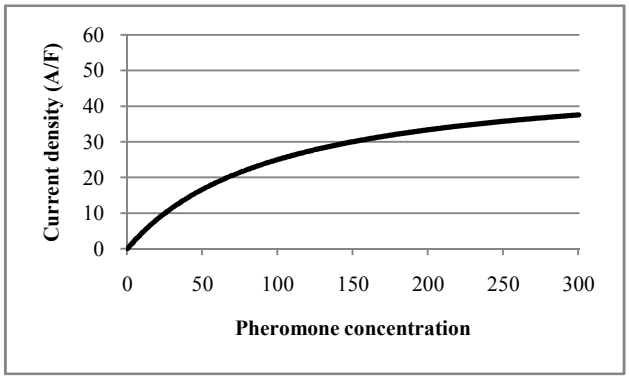


Figure 6. Current density input to sensory neuron using Equation (6).

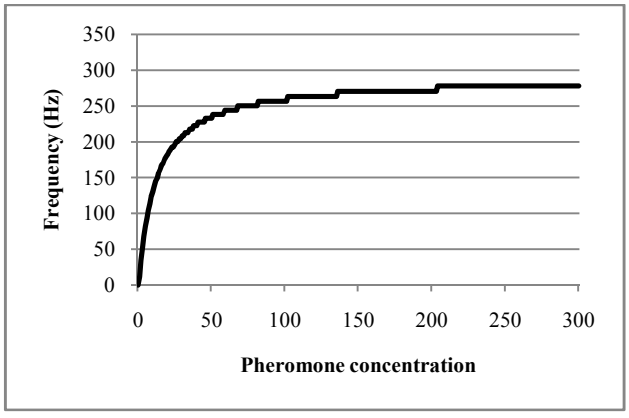


Figure 6.b. Resulting firing rate of sensory neuron.

We used a MATLAB fitting routine to find appropriate constant values for a second Hill function using Equation (6), to minimize the difference between the two functions (6) and (4) (as shown in Figure 5.a.) in order to have a function that would create a near linear relationship between the pheromone concentration and the firing rate of a sensory neuron like the function (4) (Figure 7).

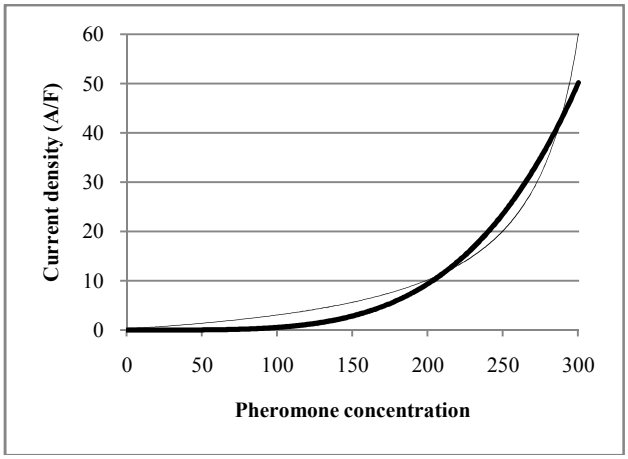


Figure 7. Current density input to sensory neuron using Equation (6) with $K_1 = 2.38*10^7$, K_2 = 7104 and m = 4.13. The thin curve is Equation (4) and the thick one is Equation (6).

Unfortunately, this function was not as good as (4). In fact, the sensor could not detect a pheromone concentration of 1. So we decided to add an offset to the function.

2.5. Hill functions with offset

$$I = K_1 \left(\frac{P^m}{K_2^m + P^m} \right) + b \quad (7)$$

This time, the MATLAB routine found a value for b too high so the sensor could fire even if it did not perceive any pheromones (Figure 8). So we tried to constrain the value of b to be less than 0.4 (Figure 9). Unfortunately, the current produced was the same (= 0.4) for a large range of small pheromone concentration so the agent could not detect differences of concentration in this range. We concluded that it was difficult to use a Hill function for the sensors' current so that the agents would be able to detect a very small and very high pheromone concentration. Hill functions with coefficients > 1 are sigmoidal so we decided to use a more general sigmoidal function.

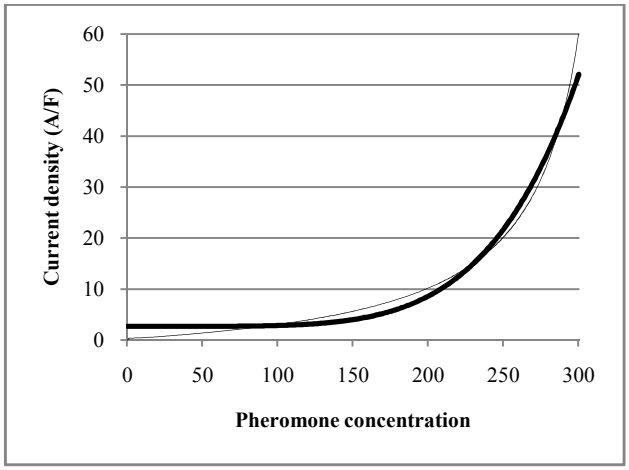


Figure 8. Current density input to sensory neuron using Equation (7) with $K_1 = 2.33*10^6$, $K_2 = 2348$, $m = 5.23$ and $b = 2.65$. The thin curve is Equation (4) and the thick one is Equation (7).

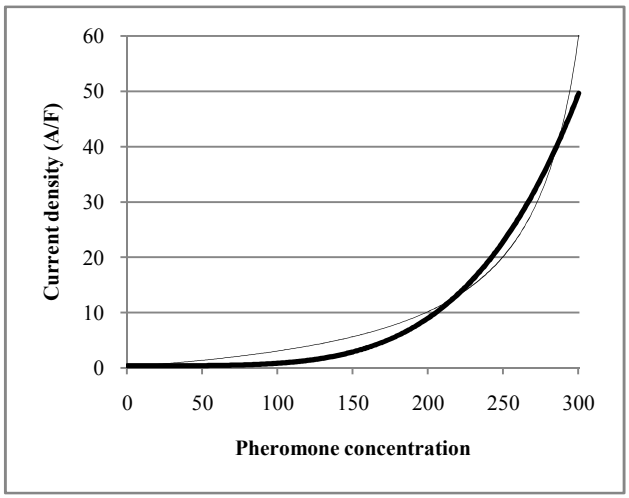


Figure 9. Current density input to sensory neuron using Equation (7) with $K_1 = 3.45*10^4$, $K_2 = 1378$, $m = 4.297$ and $b = 0.4$. Dashed curve is Equation (4) and the other one is Equation (7).

2.6. Sigmoid function

Due to the fact that the function (4) (Figure 5.a) resembles the first part of a sigmoid function, we decided to investigate general sigmoid functions.

$$I = K_1\left(\frac{1}{1+exp\left(\frac{h-P}{K_2}\right)}\right) \qquad (8)$$

We also fit this function to (4) (Figure 10). Unfortunately, the sensor could not detect 1 unit of pheromone so we added an offset to the function.

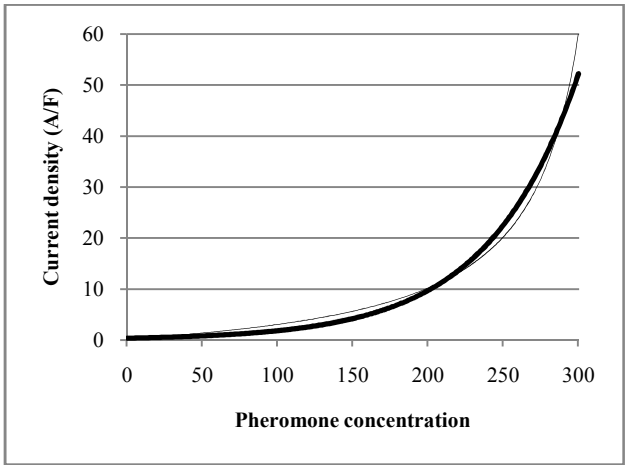


Figure 10. Current density input to sensory neuron using Equation (8) with $K_1 = 2.38*10^8$, $K_2 = 59.35$ and $h = 1210$. The thin curve is Equation (4) and the thick one is Equation (8).

2.7. Sigmoid function with offset

$$I = K_1\left(\frac{1}{1+exp\left(\frac{h-P}{K_2}\right)}\right) + b \qquad (9)$$

We found a function very similar to (4) but with an offset too high (Figure 11). So the sensor was firing even when it did not receive any information. We therefore constrained b to be less than 0.08 and found a very similar function with a small offset (Figure 12.a). After modelling a sensor using this function, we finally produced a relationship between the pheromone concentration and the sensor's firing rate (Figure 12.b) that was less linear than by using (4) but perfectly adequate to allow the agent to detect small and large variation of pheromone concentration.

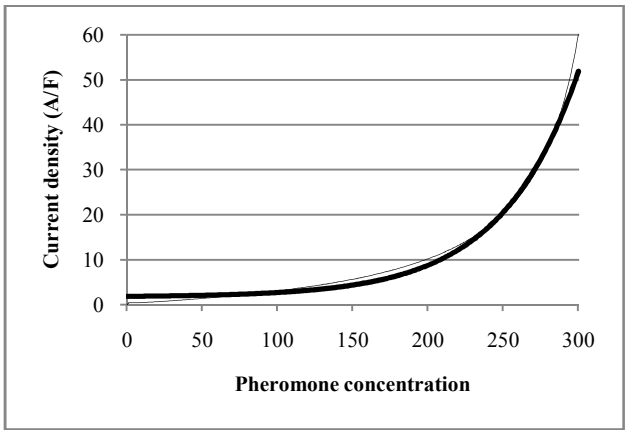


Figure 11. Current density input to sensory neuron using Equation (9) with $K_1 = 2.7*10^7$, $K_2 = 51$, $h = 973$ and $b = 1.7$. The thin curve is Equation (4) and the thick one is Equation (9).

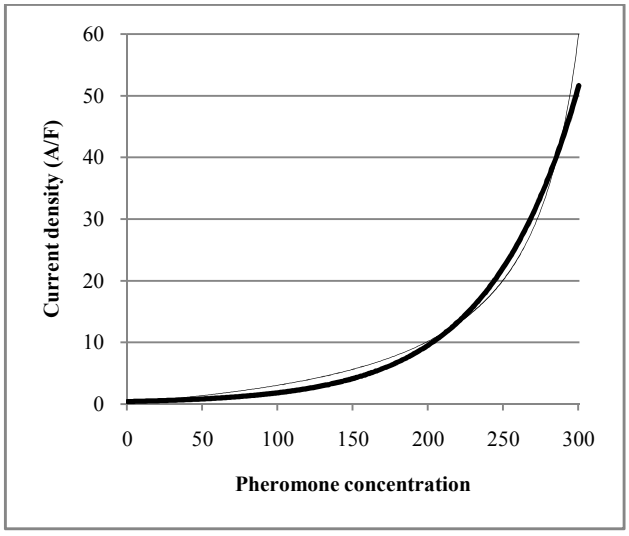


Figure 12.a. Current density input to sensory neuron using Equation (9) with $K_1 = 3.9*10^4$, $K_2 = 59$, $h = 691$ and $b = 0.08$. The thin curve is Equation (4) and the thick one is Equation (9).

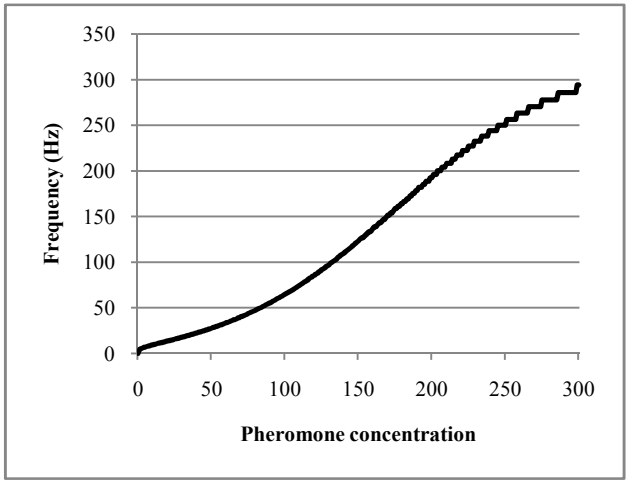


Figure 12.b. Resulting firing rate of sensory neuron.

3 Conclusion

The long-term goal of our research that sets the context for this study is to create agents able to find and interact to pheromones diffused symmetrically from a point source. In order to achieve this goal, we had to find a model of spiking sensory neuron that could cope with small variations of pheromone concentration and also the whole range of concentrations. We tried many different functions to map the pheromone concentration onto the current of the sensory neuron in order to produce a linear relationship between the concentration and the firing rate of the sensor. After unsuccessful trials using linear currents, we created an equation that would by definition achieve this task and used it as a model to help us find a similar function that is also used in biology. We concluded that by using a biologically plausible sigmoid function in our model to map pheromone concentration to current, we could produce agents able to detect the whole range of pheromone concentration as well as small variations. The sensory neurons used in our model are able to encode the stimulus intensity into appropriate firing rates. Moreover, using this model of sensory neurons, we managed to create a simulated robot capable of chemotropotaxis. We are currently studying how an agent can use two different types of information encoding strategies depending on the level of chemical concentration.

References

[Floreano and Mattiussi, 2001] Dario Floreano and Claudio Mattiusi. "Evolution of Spiking Neural Controllers for Autonomous Vision-Based Robots". *In Proceedings of the international Symposium on Evolutionary Robotics From intelligent Robotics To Artificial Life.* T. Gomi, Ed. Lecture Notes In Computer Science, vol. 2217. Springer-Verlag, London, 38-61. 2001.

[Kandel et al., 2000] Eric R. Kandel and James H. Schwartz and Thomas M. Jessell. *Principles of Neural Science*, 4th edition, McGraw-Hill. 2000.

[Koch, 1999] Christof Koch. *Biophysics of computation, Information processing in single neurons*, Oxford University Press: New York, New York. 1999.

[Stryer, 1988] Lubert Stryer. *Biochemistry*, Third Edition, W. H. Freeman and Company, New York. 1988.

[Wyatt, 2003] Tristram D. Wyatt. *Pheromones and Animal Behaviour*, *Communication by Smell and Taste*, Cambridge University Press. 2003.

Adaptive Olfactory Encoding in Agents Controlled by Spiking Neural Networks

Nicolas Oros, Volker Steuber, Neil Davey, Lola Cañamero, and Rod Adams

Science and Technology Research Institute
University of Hertfordshire
AL10 9AB
United Kingdom
{N.Oros,V.Steuber,N.Davey,L.Canamero,R.G.Adams}@herts.ac.uk

Abstract. We created a neural architecture that can use two different types of information encoding strategies depending on the environment. The goal of this research was to create a simulated agent that could react to two different overlapping chemicals having varying concentrations. The neural network controls the agent by encoding its sensory information as temporal coincidences in a low concentration environment, and as firing rates at high concentration. With such an architecture, we could study synchronization of firing in a simple manner and see its effect on the agent's behaviour.

Keywords: spiking neural network, neural encoding, firing rate, temporal coincidence.

1 Introduction

Animals are able to react to chemicals (odours, pheromones…) present in the environment. The key sense to detect these chemical cues is smell [18]. Almost all animals have a similar olfactory system including olfactory sensory neurons (OSN) that are exposed to the outside world and linked directly to the brain. Pheromones and other odour molecules present in the environment are converted into signals in the brain by first binding to the olfactory receptor protein situated in the cell membrane of the OSN. Spikes are then sent down the axon of the OSN [10]. A chemical blend is composed of many molecules that can be detected with tuned odour receptors and therefore, activates a large range of olfactory sensory neurons. Odours are coded by which neurons emit spikes and also by the firing patterns of those neurons sending spikes to others during and after the stimulus. In many vertebrates and insects, oscillations of the neural activity have been recorded in the olfactory systems [18]. Therefore, the synchronization of firing between different sensory neurons seems to be very important for odour perception and interpretation. The firing rate and the number of sensory neurons are also important in odour recognition when stronger stimuli increase the frequency of firing of individual sensory neurons but also stimulate a larger number of them.

M. Asada et al. (Eds.): SAB 2008, LNAI 5040, pp. 148–158, 2008.

Different studies have been done on the perception of simulated chemicals using artificial neural networks where neural synchronization occurs [2, 6, 7] and also using robots [11, 13, 15-17]. We were interested in studying the behaviour of an agent in response to changes of its environment. The primary research question is how two encoding strategies can be used to integrate sensory information in order to control a simulated agent. To the best of our knowledge, no neural architecture, controlling a simulated agent, has been created that encodes the sensory information onto both the firing rate and the synchronization of firing (temporal coincidence) depending on the environment. As the interaction between the two encoding strategies is complex, we decided to create a simple architecture using a spiking neural network. This model could encode the sensory information onto both the firing rate and the synchronization of firing depending on the environment. The neural network controlled the agent by encoding the sensory information onto temporal coincidences in a low concentration environment, and firing rates at high concentration.

2 Environment

We created a simulation of a continuous world including an agent and a maximum of two chemicals. We decided to use a simple model of chemicals that are not diffused and evaporated but with concentrations that can be calculated directly at any given point. In this experiment, each chemical source had a circular shape and the same fixed value all over its surface. Our agent was equipped with two antennae and a differential steering system using two wheels. The two antennae were separated widely enough to detect the presence of the chemical concentration (Fig. 1 & 2). The left and right wheels were situated in the appropriate position (middle of each side) so the gravity centre was in the middle of the agent's body (excluding the antennae).

To control the agent, we had to decide which neurons' model to use in order to study firing synchronization of the sensors.

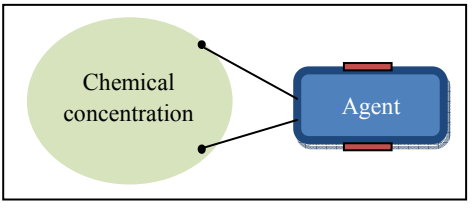


Fig. 1. An agent equipped with two wheels and two antennae used to detect chemicals

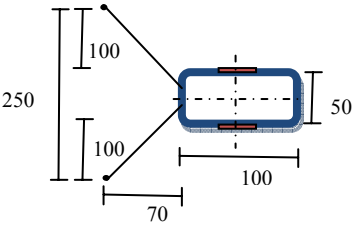


Fig. 2. Properties of an agent. Units are arbitrary.

3 Neural Network

There are three main ways to encode the intensity of sensory information into spiking neurons based on biological evidences [3-5, 8, 9, 12] . The most commonly used method consists of mapping the stimulus intensity to the firing rate of the neuron

(firing rate encoding). Another method encodes the intensity of the stimulation into the number of spikes sent by different neurons arriving at a pre-synaptic neuron at the same time (firing synchronization or temporal coincidence encoding). The last main encoding scheme maps the strength of the stimulation in the firing delay of the neuron (delay encoding). As we saw earlier, spatial configuration is an important feature in odour recognition of neurons as is the synchronization of firing between neurons [10, 14, 18]. J. Hopfield and C. Brody [2, 6] created simple neural networks using spiking neurons to simulate an olfactory process. In their system, the recognition of an odour was signalled by spike synchronization in artificial glomeruli. In our system, the neural network was supposed to detect the blend of two different chemicals and modify the agent's behaviour. We used a model of neural network that allowed us to study synchronization of firing in a simple manner. The neural network could control the agent by encoding the sensory information onto temporal coincidences in a low concentration environment, and firing rates at high concentration.

a) Models of Spiking Neurons

It is well known that compared to the complex and computationally slow Hodgkin and Huxley model, simple spiking models like integrate-and-fire neurons can run quickly enough and have a more realistic behaviour than firing rate ones [4, 5, 8, 9, 12]. This is why more and more researchers are implementing spiking neurons in robots and simulated agents. Therefore, we decided to use a simple model of a spiking neuron. Our model is based on a leaky-integrator model which includes synaptic integration and conduction delays. The idea is that a spike sent by a neuron will take some time to arrive at another neuron. This time delay depends on the distance between the sender and the receiver. All the spikes arriving at a neuron are summed to calculate the neuron's input current (in Amperes per Farad) and membrane potential (in Volts) after every time step ($\Delta t = 0.1ms$). Once the membrane potential reaches a certain threshold θ, the neuron will fire and then will be set to 0 for a certain time (refractory period). During this time, the neuron cannot fire another spike even if it is highly stimulated. Many real neurons' membrane potential is around -70mV during resting state [10]. When a neuron fires, its membrane potential will increase rapidly to about 30mV, so the height of a typical spike is approximately 100mV [10]. We set the resting potential to 0 and the potential of a spike to 100mV. It is reasonable to set the neuron's threshold at 20mV, the refractory period to 3ms and the membrane time constant τ_m to 50ms. We also decided to set a synaptic time constant τ_s to 2ms: a spike that arrives at a synapse triggers a current given by:

$$I_j(t) = \left(\frac{t-(t_{spike}+delay)}{\tau_s}\right) \exp\left(1 - \frac{t-(t_{spike}+delay)}{\tau_s}\right) \quad (1)$$

where $I_j(t)$ is the synaptic input current, t_{spike} corresponds to the time a spike has been sent to the neuron, $delay$ is the time delay in seconds before the spike arrives to the neuron (*delay* = *coeff_delay* * *distance*) with *coeff_delay* = 5.10^{-5}.

The change of membrane potential is given by:

$$\frac{dV}{dT} = -\left(\frac{V}{\tau_m}\right) + \sum(I_j W_j) \quad (2)$$

where V is the membrane potential, τ_m is the membrane time constant and W_j the synaptic weight.

b) Sensory Neurons

We created a model of a spiking sensory neuron in which the chemical concentration is processed so that a quasi-linear relationship between the concentration and the firing rate of the sensor is produced. Such relationships exist in biological systems. For example in humans, the relationship between the frequency of firing and pressure on the skin is linear [10]. We used a two step process where two biologically realistic non-linear mappings between sensory information and input current and between input current and firing rate results in a linear relationship. The sensory neurons used in our model are able to encode the stimulus intensity, measured at the tip of the antenna, into sensory input current using a biologically plausible sigmoid function. In this paper, we are using a very simple model chemical concentration that has only one value. Therefore, the sensory neurons encode this value onto the appropriate firing rate. The sensors were configured in order to distinguish a large range of concentrations between 1 and 300. Over this 300, they were saturating.

c) Motor Neurons

We decided that, in order to move, the agent should be driven by two wheels each controlled by two motor neurons: one to go forward, one to go backward. We created sensors able to detect a chemical gradient. But an agent equipped with such sensors will not move without any stimulus. So we decided for simplicity that an agent should always move forward in the absence of any external input. We performed this by adding a small baseline input current (0.5 A/F) in the motor neurons responsible to go forward. The final velocity of the wheels was calculated by subtracting the firing rate of the motor neurons, responsible for moving the agent forward and backward, running over a certain period of time.

4 Experiments

We used the agent and world described in Section 2. The world contained either one or two chemicals denoted by A or B. One agent, placed in the world, was controlled by a simple spiking neural network implementing the neurons described in Section 3. The neural controller was based on a Braitenberg vehicle (anger behaviour) [1] where an agent moves faster toward a stimulus when it detects it (Fig. 3).

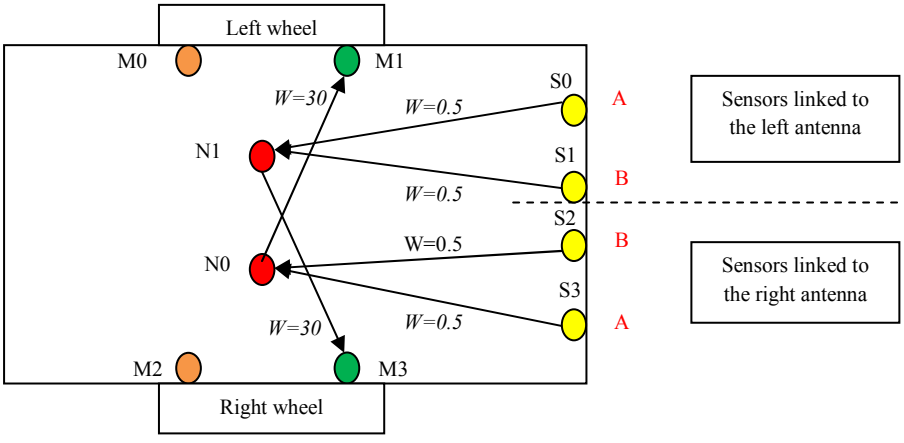


Fig. 3. Agent's neural controller. The sensors S0 and S3 detect the chemical A and the sensors S1 and S2 detect the chemical B. The sensory axons' lengths are all similar (delays = 2.5ms). The motor neurons M1 and M3 are responsible to move the agent forward. The threshold of the neurons (N0 and N1) was set to 4.6 mV. *W* is the synaptic weight.

Our hypothesis was that by using this architecture, the sensory neurons needed to encode the sensory information onto the firing rates, and also onto temporal coincidences between spikes sent by sensors. To verify this hypothesis, we performed three series of tests to study the effect of the starting positions, the sensory delays and the value of the concentrations on the agent's behaviour.

a) Experiment I

The first test was to study the effect of the agent's starting position on its behaviour. Both concentration values for the chemicals A and B were set to be low. In all the experiments described in this paper, the concentration range was from 1 to 300. In this instance, A and B concentrations were set to 1 or 2. We tried ten different starting positions and five different settings for the environment: with one chemical A, one chemical B, and finally one concentration of the chemical A overlapping with one concentration of the chemical B. Each run lasted 600 seconds and the neural network was updated every 0.1ms. Every 10ms, the agent was moved and the sensory inputs updated. In these experiments, the agent could detect double concentrations of one chemical (A or B) but did not react to it. However, the agent was able to react only to the blend of both chemicals A and B, where it stayed inside the overlapping concentrations. We recorded the agent's neural activity during each run. Figure 4 shows an example where an agent starts from the position P2. In this case, the agent was able to stay in the overlapping area.

By looking at Figure 5, we can see that the agent begins by moving horizontally left to right until its right antenna detects the chemicals A and B (T1, Fig. 4 and 5). At this point, the sensors S2 and S3 fire and the temporal coincident arrival of their

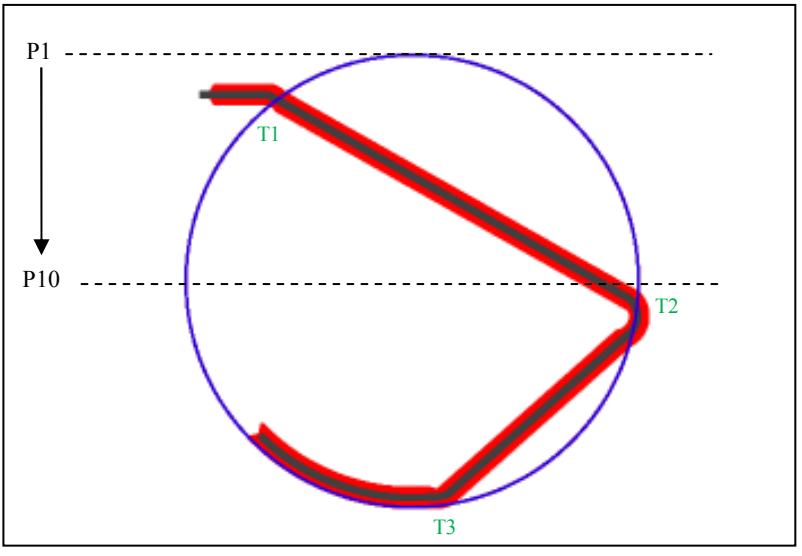


Fig. 4. Path of the agent starting from the left at P2. The circle in the centre represents the two overlapping concentrations of chemicals A and B.

spikes causes N0 to fire. M1 is then stimulated and increases its firing rate turning the left wheel faster than the right one. Soon after this, both antennae detect the chemicals causing also the neuron N1 to fire so the agent moves straight forward again. At T2, the left antenna of the agent goes outside the overlapping area so the sensors S0 and S1 stop to fire and therefore, do not stimulate the neuron N1. The motor neuron M3 then fires at a lower rate than M1 resulting in a left turn of the agent to stay inside the area. Finally, from T3, the interaction between the left antenna and the concentration causes the edge-following behaviour.

We also recorded the current density and membrane potential of the neuron N0 during a small interval of time when the agent was inside the blend of chemicals A and B. The input current of the neuron N0 was increasing when spikes coming from both S2 and S3 arrived at the same time. Then, the membrane potential also increased and reached the threshold θ (0.0046 Volts) making the neuron N0 fire. The potential was then set to 0 during the refractory period. As the sensors were synchronized and the delay between them and the neurons were the same, the spikes arrived at the same time to the neuron allowing it to detect them and fire (Fig. 6).

b) Experiment II

The second experiment was to test our hypothesis by modifying the sensory response delays to verify that our architecture necessarily needed to encode the sensory

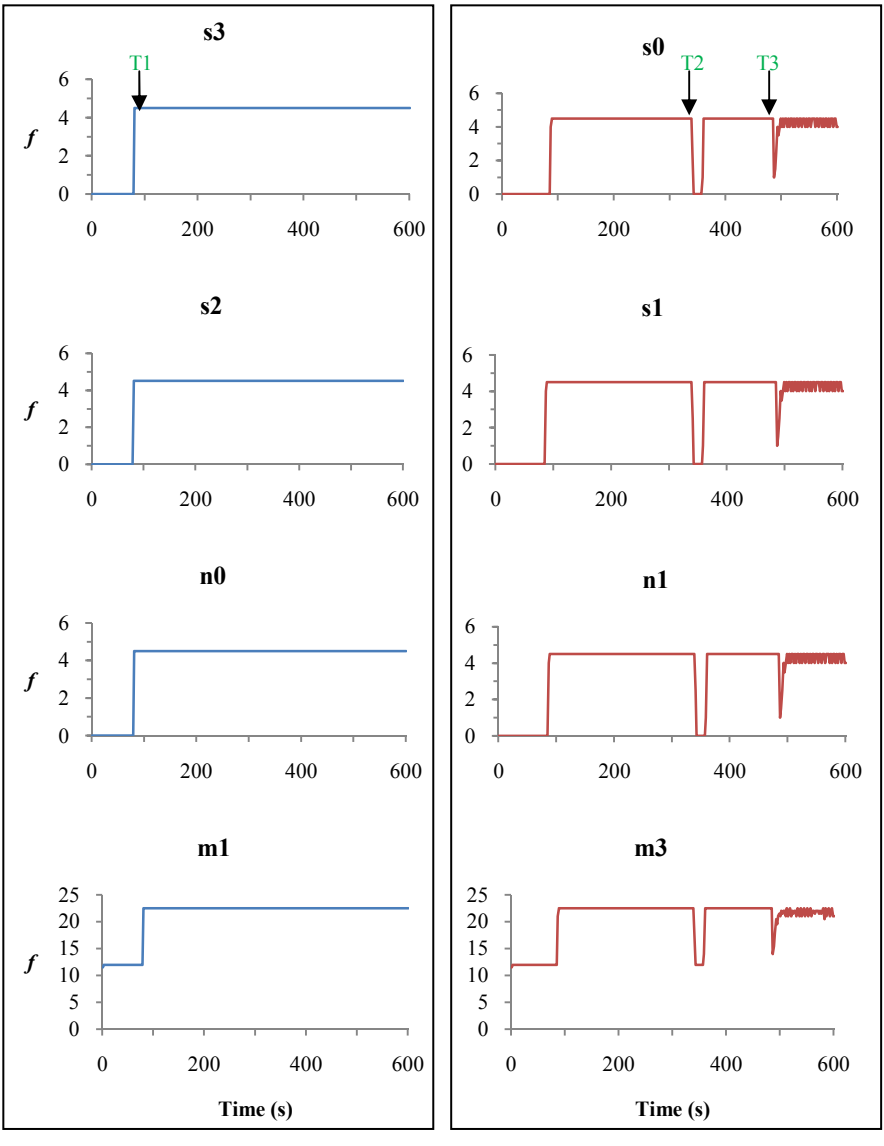


Fig. 5. Firing rates *f* (in spikes/s) of the neural network cells recorded every 2s during one run (Experiment shown in Fig. 4). The motor neurons M0 and M1 are not shown here as they do not fire. On the left panel, the sensors detecting the chemicals A (S3) and B (S2) from the right antenna activates the neuron N0 that stimulates the motor neuron M1 controlling the left wheel to move forward. On the right panel, the sensors detecting the chemicals A (S0) and B (S1) from the left antenna activates the neuron N1 that stimulates the motor neuron M3 controlling the right wheel to move forward.

information onto temporal coincidence. We changed the delays by modifying the position of the sensors therefore modifying the length of their axons linked to the neurons. We only changed the delays of the sensors detecting the chemical B (S1 and S2). We used the same set up as for the experiment shown in Figure 4.

We tried different values of delays (from 1ms to 50ms) and we noticed that a small change (up to 7.5ms) did not modify the agent's behaviour. But a further change in the delays (from 7.5ms) made the agent unable to react to the blend of chemicals A and B so it could not stay inside the concentrations.

As in the Experiment I, we recorded the current density and membrane potential of the neuron N0 during 0.5s when the agent was inside the chemical blend. In Figure 6, we can see that the current of the neuron N0 increases when a spike coming from both S2 and S3 arrive but as the delay has been changed, the spikes do not arrive at the same time so the current is lower than in Experiment I. Therefore, the neuron's potential increases but never reaches the threshold so the neuron does not fire (Fig. 6).

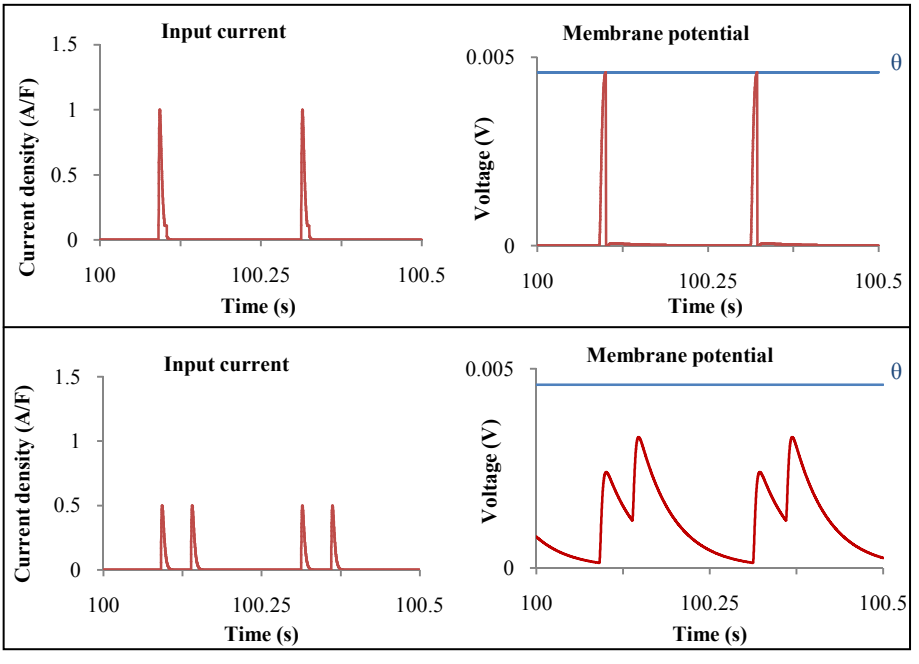


Fig. 6. Current density (in Amperes per Farad) and membrane potential (in Volts) of the neuron N0 recorded between 100s and 100.5s. On the top panel (Experiment I), the spikes sent by the sensors arrived at the same time increasing the current density to 1 A/F. The membrane potential was then increased and reached the threshold making the neuron N0 fire. On the bottom panel (Experiment II), the spikes sent by the sensors were not coincident as the delays were changed to 50ms in this case, so the current was never above 0.5 A/F and therefore, the membrane potential could not reach the threshold to make the neuron N0 fire.

c) Experiment III

In order to investigate the use of firing rate encoding, we used only one concentration of either A or B and increased it. When the concentration was augmented from 1 to above 50, the agent was then able to react to it. Therefore, the neural network showed much more sensitivity to two chemicals than to one. We also realized when using two overlapping chemicals A and B, as the concentration value increased, modifying the delays had a minor effect and the agent was still able to react to the chemicals. The firing rates were increasing too so the agent was moving faster. In these experiments, the temporal coincidence encoding was not necessary. The sensory information was encoded onto the firing rates of the sensors.

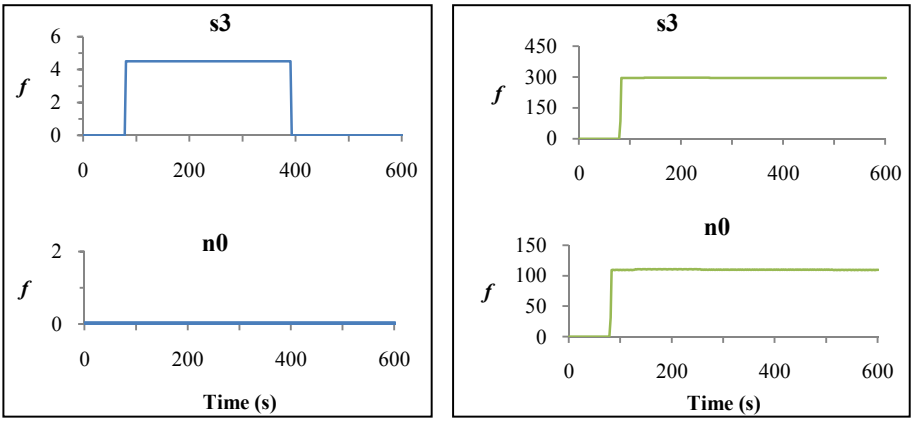


Fig. 7. Firing rates of the sensor S3 and neuron N0 recorded every 2s. On the left panel, as the environment contained a low concentration (=1) of chemical A only, the neuron could not detect it and therefore, the agent did not stay within the chemical source area. On the right panel, the concentration was high (=300) so the neuron could detect it and the agent stayed inside the area.

5 Conclusion

We presented in this paper a simple neural architecture where temporal coincidence and firing rate encoding strategies were both important mechanisms used in different environmental settings. In a low concentration setting, synchronization of spikes sent by the sensors was essential to allow the agent to detect the blend of two chemicals. We changed the sensory delays and noticed that the agent was then not able to react to the chemicals anymore. In a high concentration setting, the temporal coincidence between sensors firing was not a necessary condition and the agent was able to stay inside the chemical concentration using just the firing rate encoding strategy.

Interestingly, the model showed much more sensitivity to the presence of two chemicals than a single chemical. In principle, more than two chemicals can be detected and processed. The architecture presented here also works when the chemical concentration has a linear gradient.

Future work will investigate evolving such architectures using a developmental model (evolving the number of neurons and their connections, the synaptic weights, and delays of the neural network). Moreover, we will add noise to the neural network and use a more complex environment.

References

1. Braitenberg, V.: Vehicles: Experiments in Synthetic Psychology. MIT Press, Cambridge (1984)
2. Brody, C.D., Hopfield, J.J.: Simple Networks for Spike-Timing-Based Computation, with Application to Olfactory Processing. Neuron 37, 843–852 (2003)
3. Floreano, D., Mattiussi, C.: Evolution of Spiking Neural Controllers for Autonomous Vision-Based Robots. In: Gomi, T. (ed.) ER-EvoRob 2001. LNCS, vol. 2217, pp. 38–61. Springer, Heidelberg (2001)
4. Florian, R.V.: Biologically inspired neural networks for the control of embodied agents. Technical report Coneural-03-03 Version 1.0 (2003)
5. Gerstner, W., Kistler, W.M.: Spiking Neuron Models. Single Neurons, Populations, Plasticity. Cambridge University Press, Cambridge (2002)
6. Hopfield, J.J.: Odor space and olfactory processing: Collective algorithms and neural implementation. PNAS 96, 12506–12511 (1999)
7. Hoshino, O., Kashimori, Y., Kambara, T.: An olfactory recognition model based on spatio-temporal encoding of odor quality in the olfactory bulb. Biological Cybernetics 79, 109–120 (1998)
8. Izhikevich, E.M.: Simple model of spiking neurons. IEEE Transactions on Neural Networks (2003)
9. Izhikevich, E.M.: Which model to use for cortical spiking neurons. IEEE Transactions on Neural Networks (2004)
10. Kandel, E.R., Schwartz, J.H., Jessell, T.M.: Principles of Neural Science. McGraw-Hill, New York (2000)
11. Kanzaki, R., Nagasawa, S., Shimoyama, I.: Neural Basis of Odor-source Searching Behavior in Insect Brain Systems Evaluated with a Mobile Robot. Chemical Senses 30, 285–286 (2005)
12. Koch, C.: Biophysics of Computation: Information Processing in Single Neurons. Oxford University Press, New York (1999)
13. Kuwana, Y., Shimoyama, I.: A Pheromone-Guided Mobile Robot that Behaves like a Silkworm Moth with Living Antennae as Pheromone Sensors. The International Journal of Robotics Research 17, 924–933 (1998)
14. Laurent, G., Wehr, M., Davidowitz, H.: Temporal Representations of Odors in an Olfactory Network. Journal of Neuroscience. 16, 3837–3847 (1996)
15. Payton, D., Daily, M., Estowski, R., Howard, M., Lee, C.: Pheromone Robotics. Auton. Robots. 11, 319–324 (2001)

16. Pyk, P., Bermúdez i Badia, S., Bernardet, U., Knüsel, P., Carlsson, M., Gu, J., Chanie, E., Hansson, B., Pearce, T.J., Verschure, P.: An artificial moth: Chemical source localization using a robot based neuronal model of moth optomotor anemotactic search. Autonomous Robots 20, 197–213 (2006)
17. Webb, B.: Robots crickets and ants: models of neural control of chemotaxis and phonotaxis. Neural Networks 11, 1479–1496 (1998)
18. Wyatt, T.D.: Pheromones and Animal Behaviour, Communication by Smell and Taste. Cambridge University Press, Cambridge (2003)

Optimal noise in spiking neural networks for the detection of chemicals by simulated agents

Nicolas Oros, Volker Steuber, Neil Davey, Lola Cañamero, Rod Adams

Science and Technology Research Institute
University of Hertfordshire
AL10 9AB
United Kingdom
{N.Oros, V.Steuber, N.Davey, L.Canamero, R.G.Adams}@herts.ac.uk

Abstract

We created a spiking neural controller for an agent that could use two different types of information encoding strategies depending on the level of chemical concentration present in the environment. The first goal of this research was to create a simulated agent that could react and stay within a region where there were two different overlapping chemicals having uniform concentrations. The agent was controlled by a spiking neural network that encoded sensory information using temporal coincidence of incoming spikes when the level of chemical concentration was low, and as firing rates at high level of concentration. With this architecture, we could study synchronization of firing in a simple manner and see its effect on the agent's behaviour. The next experiment we did was to use a more realistic model by having an environment composed of concentration gradients and by adding input current noise to all neurons. We used a realistic model of diffusive noise and showed that it could improve the agent's behaviour if used within a certain range. Therefore, an agent with neuronal noise was better able to stay within the chemical concentration than an agent without.

Introduction

Animals are able to detect and react to chemicals (odours, pheromones…) present in the environment. The key sense to detect these chemical cues is smell rather than taste (Wyatt, 2003). Almost all animals have a similar olfactory system including olfactory sensory neurons (OSN) that are exposed to the outside world and linked directly to the brain. Pheromones and other odour molecules present in the environment are converted into signals in the brain by first binding to the olfactory receptor protein situated in the cell membrane of the OSN. Spikes are then sent down the axon of the OSN (Kandel et al., 2000). A chemical blend is composed of many molecules that can be detected with tuned odour receptors and therefore, activates a large range of olfactory sensory neurons. Odours are coded by which neurons emit spikes and also by the firing patterns of those neurons sending spikes to others during and after the stimulus. In many vertebrates and insects, oscillations of the neural activity have been recorded in the olfactory systems (Wyatt, 2003). Therefore, the synchronization of firing between different sensory neurons seems to be very important for odour perception and interpretation. The firing rate and the number of sensory neurons are also important in odour recognition when stronger stimuli increase the frequency of firing of individual sensory neurons but also stimulate a larger number of them.

Different studies have been done on the perception of simulated chemicals using artificial neural networks where neural synchronization occurs (Brody & Hopfield, 2003; Hopfield, 1999; Hoshino et al., 1998) and also using robots (Kanzaki et al., 2005; Kuwana & Shimoyama, 1998; Payton et al., 2001; Pyk et al., 2006; Webb, 1998). We were interested in studying the perception and the behaviour of an agent in response to changes of its environment. The primary research question is how two encoding strategies can be used to integrate sensory information in order to control a simulated agent. To the best of our knowledge, no neural architecture, controlling a simulated agent, has been created that encodes the sensory information onto both the firing rate and the synchronization of firing (temporal coincidence of incoming spikes) depending on the environment. As the interaction between the two encoding strategies is complex, we decided to create a simple architecture using a spiking neural network. This model could encode the sensory information onto both the firing rate and the synchronization of firing depending on the environment. The neural network controlled the agent by encoding the sensory information onto temporal coincidences in a low concentration environment, and firing rates at high concentration.

It is well known that real neuronal systems contain noise (Kandel et al., 2000) which may improve the brain's ability to process information, a phenomenon also called stochastic resonance (Hänggi, 2002; Mori & Kai, 2002; Moss et al., 2004; Wiesenfeld & Moss, 1995). Researchers in robotics and artificial life have already implemented simple models of neural noise (Di Paolo, 2003; Florian, 2006; Jacobi et al., 1995). Here we study the effect of a more realistic noise model based on a diffusive OU (Ornstein-Uhlenbeck) process (Uhlenbeck & Ornstein, 1930). We added this noise in the neural network and studied its effect on the behaviour of the agent. Our results suggest a potential function for noise in real

biological systems, and highlight that features of biological systems can be used to construct better agents.

Environment

We created a simulation of a continuous world including an agent and a maximum of two chemicals. We decided to use a simple model of chemicals that are not diffused and evaporated but with concentrations that can be calculated directly at any given point. Our agent was equipped with two antennae and a differential steering system using two wheels. The two antennae were separated widely enough to detect the presence of the chemical concentration (Fig. 1). The left and right wheels were situated on the sides of the agents. To control the agent, we had to decide which neurons' model to use in order to study firing synchronization of the sensors.

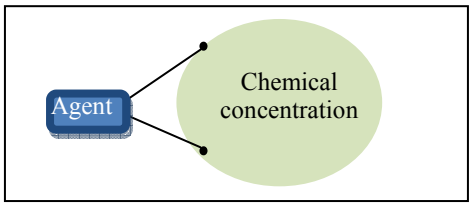


Figure 1. An agent equipped with two wheels and two antennae used to detect chemicals.

Neural Network

There are three main ways to encode the intensity of sensory information into spiking neurons based on biological evidences (Floreano & Mattiussi, 2001; Florian, 2003; Gerstner & Kistler, 2002; Izhikevich, 2003, 2004; Koch, 1999) . The most commonly used method consists of mapping the stimulus intensity to the firing rate of the neuron (firing rate encoding). Another method encodes the intensity of the stimulation into the number of spikes sent by different neurons arriving at a pre-synaptic neuron at the same time (firing synchronization or temporal coincidence encoding). The last main encoding scheme maps the strength of the stimulation in the firing delay of the neuron (delay encoding). As we saw earlier, spatial configuration is an important feature in odour recognition of neurons as is the synchronization of firing between neurons (Kandel et al., 2000; Laurent et al., 1996; Wyatt, 2003). J. Hopfield and C. Brody (Brody & Hopfield, 2003; Hopfield, 1999) created simple neural networks using spiking neurons to simulate an olfactory process. In their system, the recognition of an odour was signalled by spike synchronization in artificial glomeruli. In our system, the neural network was supposed to detect the blend of two different chemicals and modify the agent's behaviour. We used a model of neural network that allowed us to study synchronization of firing in a simple manner. The neural network could control the agent by encoding the sensory information onto temporal coincidences in a low concentration environment, and firing rates at high concentration.

Models of Spiking Neurons

It is well known that compared to the complex and computationally slow Hodgkin and Huxley model, simple spiking models like integrate-and-fire neurons can run quickly enough and have a more realistic behaviour than firing rate ones (Floreano & Mattiussi, 2001; Florian, 2003; Gerstner & Kistler, 2002; Izhikevich, 2003, 2004; Koch, 1999). This is why more and more researchers are implementing spiking neurons in robots and simulated agents. Therefore, we decided to use a simple model of a spiking neuron. Our model is based on a leaky-integrator model which includes synaptic integration and conduction delays. The idea is that a spike sent by a neuron will take some time to arrive at another neuron. This time delay depends on the distance between the sender and the receiver. All the spikes arriving at a neuron are summed to calculate the neuron's input current density (in Amperes per Farad) and membrane potential (in Volts) after every time step ($\Delta t = 0.1ms$). Once the membrane potential reaches a certain threshold θ, the neuron will fire and then will be set to 0 for a certain time (refractory period). During this time, the neuron cannot fire another spike even if it is highly stimulated.

Many real neurons' membrane potential is around -70mV during resting state. When a neuron fires, its membrane potential will increase rapidly to about 30mV, so the height of a typical spike is approximately 100mV (Kandel et al., 2000). We set the resting potential to 0 and the potential of a spike to 100mV. It is reasonable to set the neuron's threshold at 20mV, the refractory period to 3ms and the membrane time constant τ_m to 50ms (Kandel et al., 2000). We also decided to set a synaptic time constant τ_s to 2ms: a spike that arrives at a synapse triggers a current given by:

$$I_j(t) = \left(\frac{t-(t_{spike}+delay)}{\tau_s}\right) exp\left(1 - \frac{t-(t_{spike}+delay)}{\tau_s}\right) \quad (1)$$

where $I_j(t)$ is the synaptic input current, t_{spike} corresponds to the time a spike has been sent to the neuron, $delay$ is the time delay in seconds before the spike arrives to the neuron (*delay* = *coeff_delay* * *distance*) with *coeff_delay* $= 5.10^{-5}$.

The change of membrane potential is given by:

$$\frac{dV}{dT} = -\left(\frac{V}{\tau_m}\right) + \sum(I_j W_j) \quad (2)$$

where V is the membrane potential, τ_m is the membrane time constant and $W_j$the synaptic weight.

Sensory Neurons

We created a model of a spiking sensory neuron in which the chemical concentration is processed so that a quasi-linear relationship between the concentration and the firing rate of the sensor is produced (Oros et al., 2008). Such relationships exist in biological systems. For example in humans, the relationship between the frequency of firing and pressure on the skin is linear (Kandel et al., 2000). We used a two step process where two biologically realistic non-linear mappings between sensory information and input current and between

input current and firing rate results in a linear relationship. Researchers in robotics and artificial life use a linear direct mapping between the sensory information and the firing rate (Di Paolo, 2002, 2003; Florian, 2006). The sensory neurons used in our model are able to encode the stimulus intensity, measured at the tip of the antenna, into sensory input current using a biologically plausible sigmoid function (Oros et al., 2008). This current is injected to the sensor's membrane potential that increases, making the sensor fire into appropriate firing rates. Therefore, the sensory neurons encode the concentration value onto the appropriate firing rate. The sensors were configured in order to distinguish a large range of concentrations between 1 and 300. Over 300, they were saturating.

Motor Neurons

We decided that, in order to move, the agent should be driven by two wheels each controlled by two motor neurons: one to go forward, one to go backward. We created sensors able to detect a chemical gradient. But an agent equipped with such sensors will not move without any stimulus. So we decided for simplicity that an agent should always move forward in the absence of any external input. We performed this by adding a small baseline input current (0.5 A/F) in the motor neurons responsible to go forward. The final velocity of the wheels was calculated by subtracting the firing rate of the motor neurons, responsible for moving the agent forward and backward, running over a certain period of time. The agent was moved by calculating the velocity every 10ms.

Temporal Coincidence

We used the agent and world described above. The environment contained either one or two chemicals denoted by A or B. In this experiment, each chemical source had a circular shape and the same fixed value all over its surface. One agent, placed in the world, was controlled by a simple spiking neural network implementing the neurons described in the previous section.

The neural controller was based on a Braitenberg vehicle (anger behaviour) (Braitenberg, 1984) where an agent moves faster toward a stimulus when it detects it (Fig. 2).

Our hypothesis was that by using this architecture, the sensory neurons needed to encode the sensory information onto the firing rates, and also onto temporal coincidences between spikes sent by sensors. To verify this hypothesis, we performed three series of tests to study the effect of the starting positions, the sensory delays and the value of the concentrations on the agent's behaviour.

Experiment I

The first test was to study the effect of the agent's starting position on its behaviour. Both concentration values for the chemicals A and B were set to be low. In all the experiments described in this paper, the concentration range was from 1 to 300. In this instance, A and B concentrations were set to 1 or 2. We tried ten different starting positions and five different settings for the environment: with one chemical A, one chemical B, and finally one concentration of the chemical A overlapping with one concentration of the chemical B. Each run lasted 600 seconds and the neural network was updated every 0.1ms (so the run lasted 6,000,000 time steps). Every 10ms, the agent was moved and the sensory inputs updated.

In these experiments, the agent could detect double concentrations of one chemical (A or B) but did not react to it. However, the agent was able to react only to the blend of both chemicals A and B, where it stayed inside the overlapping concentrations. We recorded the current density and membrane potential of the neuron N0 during a small interval of time when the agent was inside the blend of chemicals A and B (Fig. 3, top). The input current of the neuron N0 was increasing when spikes coming from both S2 and S3 arrived at the same time. Then, the membrane potential also increased and reached the threshold θ (0.0046 Volts) making the neuron N0 fire. The potential was then set to 0 during the refractory period. As the sensors were synchronized and the delay between them and the neurons were the same, the spikes arrived at the same time to the neuron allowing it to detect them and fire.

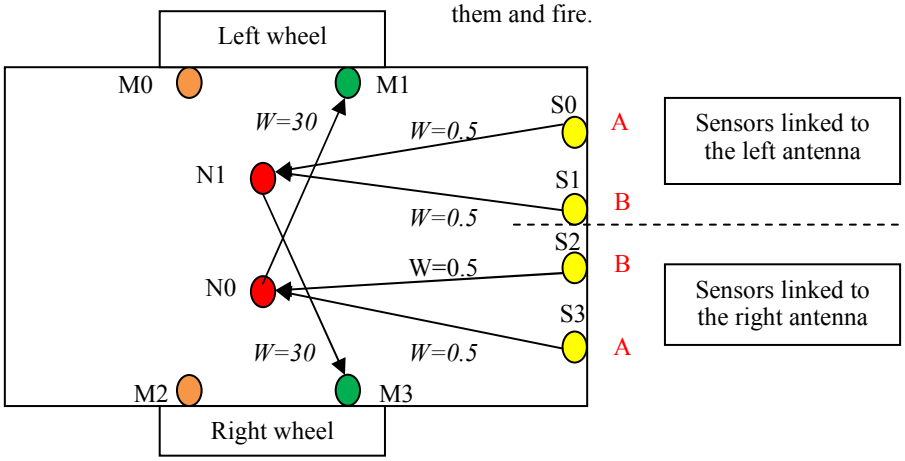


Figure 2. Agent's neural controller. The sensors S0 and S3 detect the chemical A and the sensors S1 and S2 detect the chemical B. The sensory axons' lengths are all similar (delays = 2.5ms). The motor neurons M1 and M3 are responsible to move the agent forward. The threshold of the neurons (N0 and N1) was set to 4.6 mV. *W* is the synaptic weight.

Experiment II

The second experiment was to test our hypothesis by modifying the sensory response delays to verify that our architecture necessarily needed to encode the sensory information onto temporal coincidence. We changed the delays by modifying the position of the sensors therefore modifying the length of their axons linked to the neurons. We only changed the delays of the sensors detecting the chemical B (S1 and S2).

We used one of the Experiment I 's setups where the agent was staying in the chemical blend of the chemicals A and B having a concentration of 1 each. We tried different values of delays (from 1ms to 50ms) and we noticed that a small change (up to 7.5ms) did not modify the agent's behaviour. But a further change in the delays (from 7.5ms) made the agent unable to react to the blend of chemicals A and B so it could not stay inside the concentrations.

As in the Experiment I, we recorded the current density and membrane potential of the neuron N0 during 0.5s when the agent was inside the chemical blend.

In Figure 3 (bottom), we can see that the current of the neuron N0 increases when a spike coming from both S2 and S3 arrive but as the delay has been changed, the spikes do not arrive at the same time so the current is lower than in Experiment I. Therefore, the neuron's potential increases but never reaches the threshold so the neuron does not fire (Fig. 3, bottom).

Experiment III

In order to investigate the use of firing rate encoding, we used only one concentration of either A or B and increased it. When the concentration was augmented from 1 to above 50, the agent was then able to react to it. Therefore, the neural network showed much more sensitivity to two chemicals than to one. We also realized when using two overlapping chemicals A and B, as the concentration value increased, modifying the delays had a minor effect and the agent was still able to react to the chemicals. The firing rates were increasing too so the agent was moving faster. In these experiments, the temporal coincidence encoding was not necessary. The sensory information was encoded onto the firing rates of the sensors.

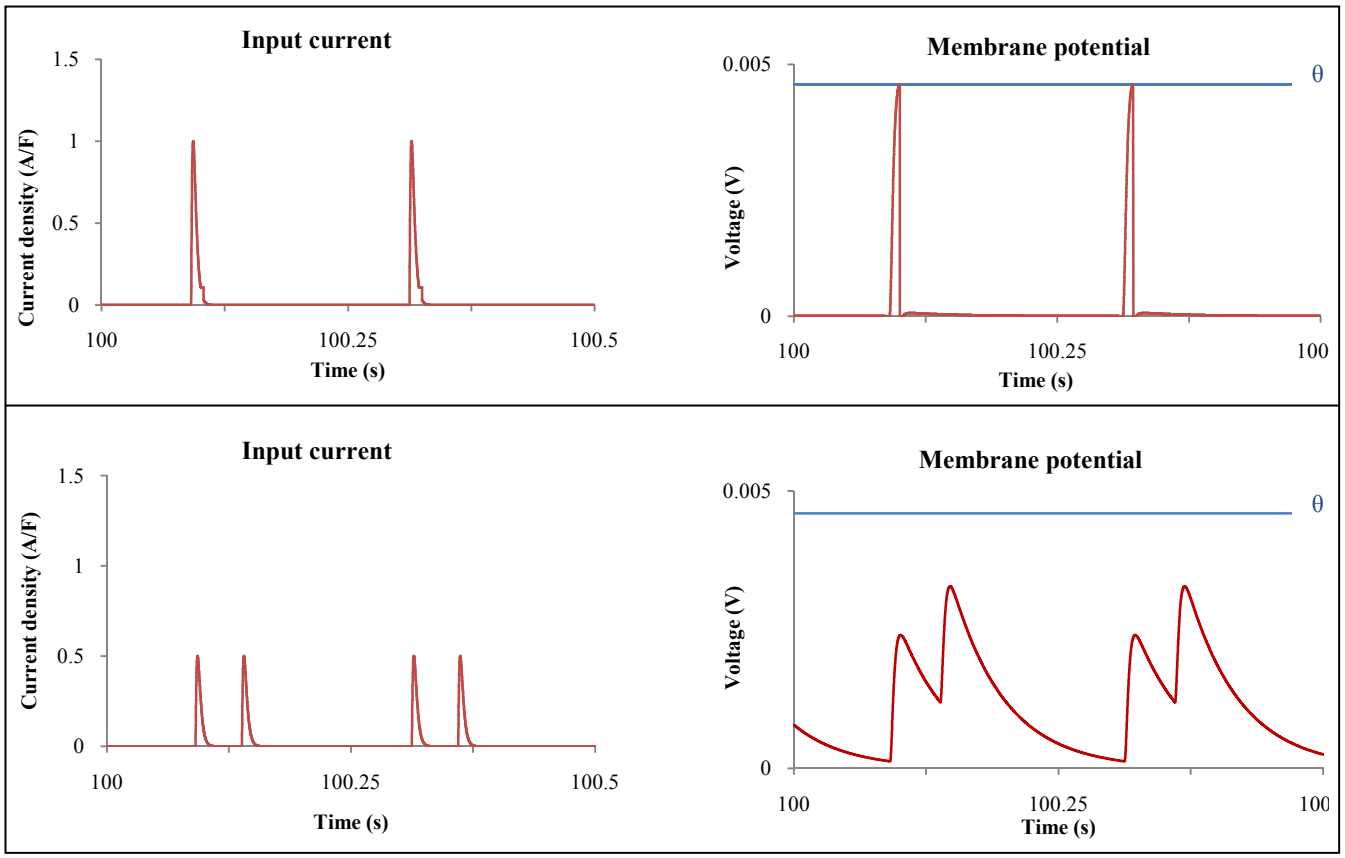


Figure 3. Current density (in Amperes per Farad) and membrane potential (in Volts) of the neuron N0 recorded between 100s and 100.5s. On the top panel (Experiment I), the spikes sent by the sensors arrived at the same time increasing the current density to 1 A/F. The membrane potential was then increased and reached the threshold making the neuron N0 fire. On the bottom panel (Experiment II), the spikes sent by the sensors were not coincident as the delays between the sensors (S1 and S2) and the neurons (N0 and N1) were changed (to 50ms in this case). Therefore the current was never above 0.5 A/F so the membrane potential could not reach the threshold to make the neuron N0 fire.

Diffusive noise

In the previous experiments, we presented a simple neural architecture where temporal coincidence and firing rate encoding strategies were both important mechanisms used in different environmental settings. In a low concentration setting, synchronization of spikes sent by the sensors was essential to allow the agent to detect the blend of two chemicals. We changed the sensory delays and noticed that the agent was then not able to react to the chemicals anymore. In a high concentration setting, the temporal coincidence between the firing of the sensors was not a necessary condition and the agent was able to stay inside the chemical concentration using just a firing rate encoding strategy. Interestingly, the model showed much more sensitivity to the presence of two chemicals than a single chemical. To this point, we have used uniform concentrations to simplify the study of the different encoding strategies. However, this model of chemical concentration was not realistic, so we decided to use an environment comprising two non uniform chemical concentration gradients. We tested our architecture in the new environment and noticed that the agent moved outside the concentration when its trajectory was along the direction of the gradient since both of its antennae where instantaneously outside the chemical concentrations. For this reason, we decided to add noise to the neural network.

We used a realistic model of noise in the form of an diffusive OU current (Uhlenbeck & Ornstein, 1930). This form of colored noise characterizes the subthreshold voltage fluctuations in real neuronal membranes (Rudolph & Destexhe, 2003). We added this noise to the total current calculated in Equation (2) in each neuron. The noise is described by:

$$\frac{dI(t)}{dt} = -\frac{1}{\tau_I}\left(I(t) - I_0\right) + \sqrt{\frac{2\sigma^2}{\tau_I}}\xi(t) \qquad (3)$$

where τ_I denotes the current noise time constant (2ms in our case), I_0 is the mean synaptic current (0 in our case), σ is the noise diffusion coefficient and $\xi(t)$ is a white Gaussian noise (with mean = 0 and standard deviation = 1).

We performed different series of tests to find appropriate level of noise, by modifying σ, in order to have an agent that stays in the gradient chemical blend. We placed the agent at three different positions (Fig. 6) and tried eight different levels of noise (Fig. 4 and 5). For each level, we performed 100 runs per position. Each run lasted 300s and we recorded the fitness of an agent during the last 100s. The fitness function was very simple and consisted of the sum of the distance between the agent and the centre of the concentrations measured every time the agent moved. The maximum value of both concentrations was set to 25.

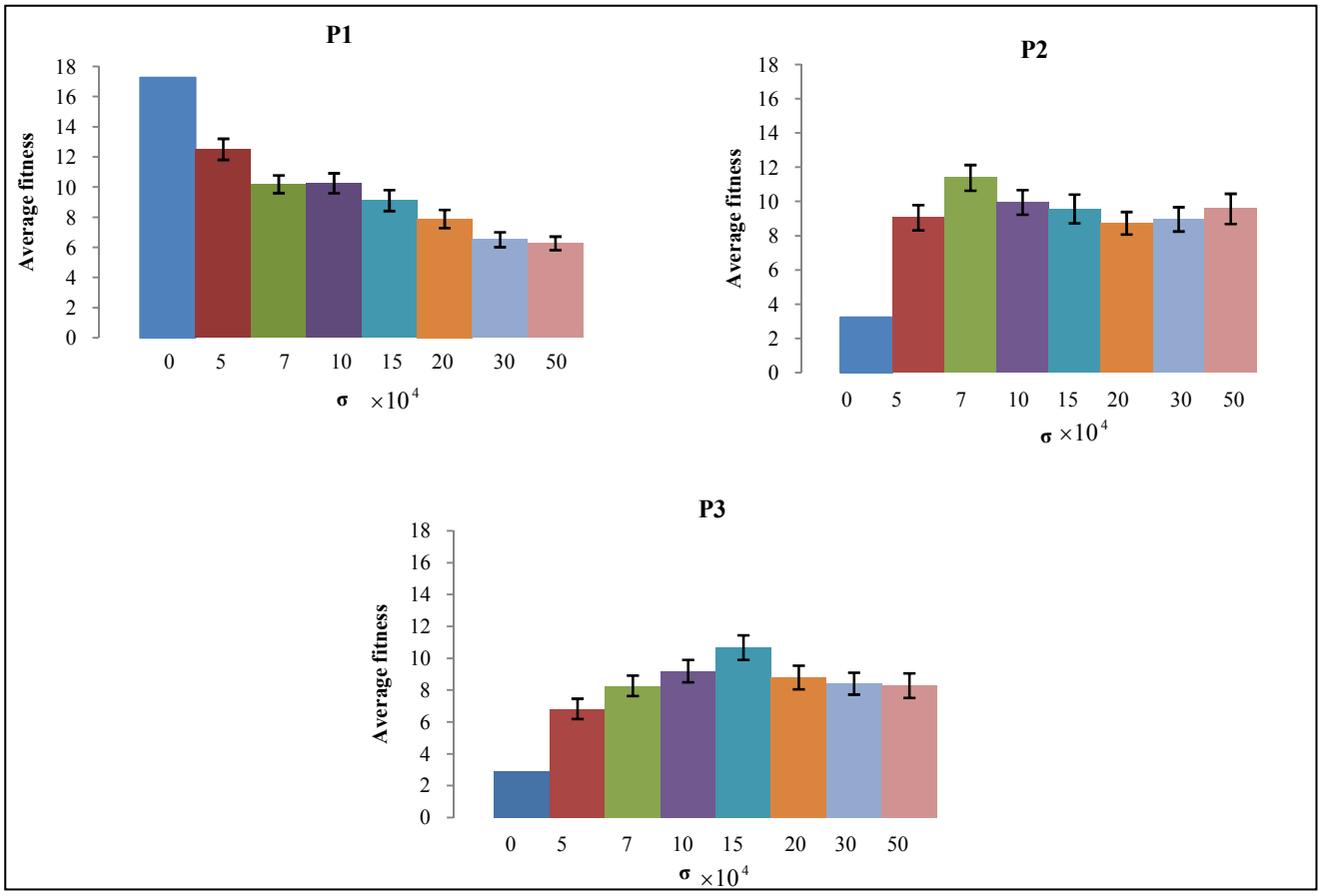


Figure 4. Mean fitness values recorded during 100s for an agent starting at the positions P1, P2 and P3 using different levels of noise ($\sigma \times 10^4$). The error bars represent standard errors.

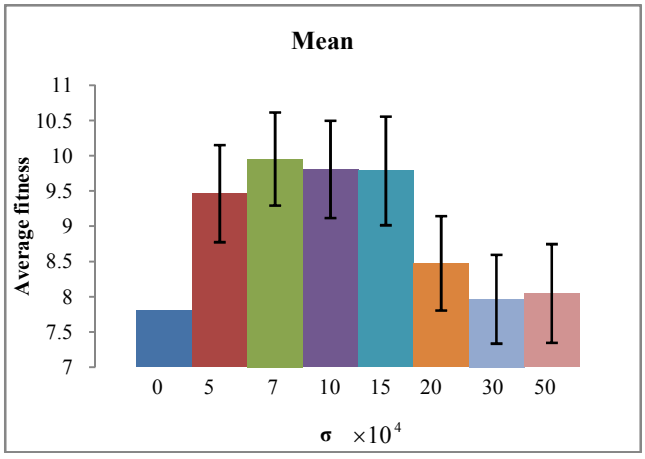


Figure 5. Mean of the fitness values displayed in Figure 4. ($\sigma \times 10^4$).

By looking at Figures 4 and 5, we can see that when the agent was starting from P2 or P3, an appropriate level of noise allowed it to stay within the concentration having a higher fitness than an agent without neural noise. We also note that the level of noise needed to be within a certain range as a low value did not improve the agent's behaviour and a high value disturbed it. We noticed as well that the agent was more sensitive to noise in low concentration areas than in high concentration areas.

Conclusion

We first presented in this paper a simple neural architecture where temporal coincidence and firing rate encoding strategies were both important mechanisms used in different environmental settings. In a low concentration setting, synchronization of spikes sent by the sensors was essential to allow the agent to detect the blend of two chemicals. We changed the sensory delays and noticed that the agent was then not able to react to the chemicals anymore. In a high concentration setting, the temporal coincidence between sensors firing was not a necessary condition and the agent was able to stay inside the chemical concentration using just the firing rate encoding strategy. Interestingly, the model showed much more sensitivity to the presence of two chemicals than a single chemical. Our results showed that a spiking neural network could be used to control an agent and could encode external stimuli in more than one way. The second study was on the effect of noise on the agent's behaviour using the same neural architecture. We used a more complex environment using chemical gradients and a realistic model of neural noise. We found that the overall fitness of the agent was better when a certain amount of noise was added in the neural network. Our results suggest that a realistic model of noise can improve an agent's behaviour. This is further evidence that adding biologically realistic features can be beneficial for certain engineering tasks, and suggests a potential function of noise in real biological systems. The effect of biologically realistic noise should be an interesting topic of research in other artificial life scenarios.

Our future work will be to see if we can evolve such architecture using a developmental model (evolving the number of neurons and their connections, the synaptic weights, and delays of the neural network).

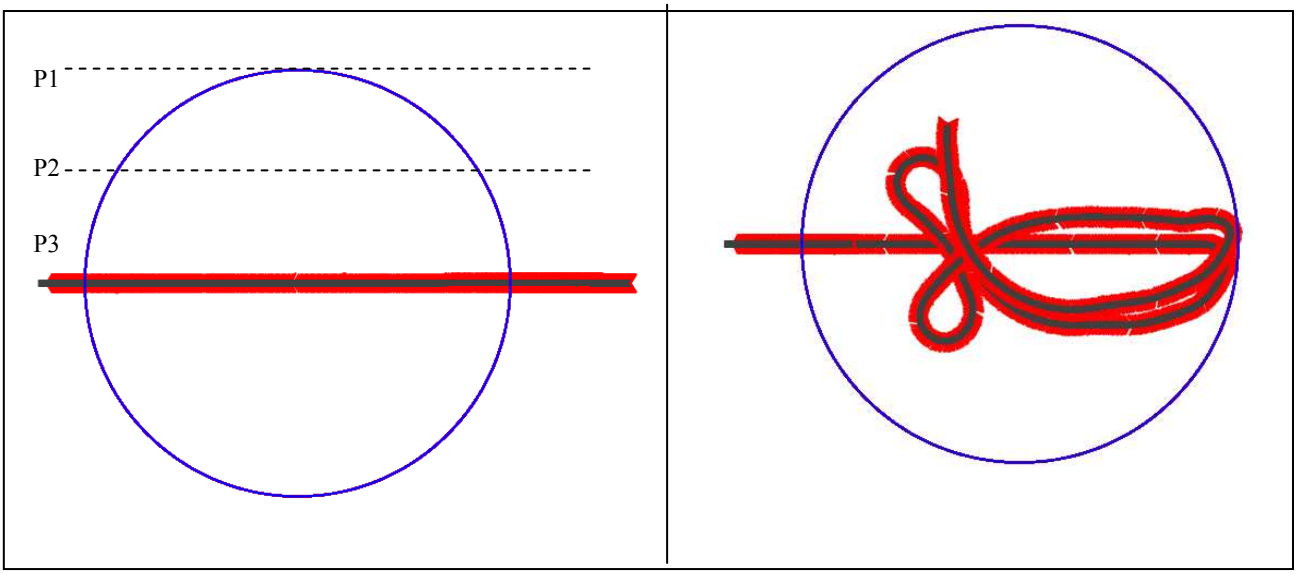


Figure 6. Left panel: path of an agent moving across the blend of chemicals A and B. The agent's neural controller doesn't have any noise so the agent goes straight as both of its antennae arrived at the same time outside the concentration. Right panel: path of an agent running over 300s. The agent's neural controller has noise so the agent does not go exactly in a straight line and therefore, can react to the absence of the chemical concentration to stay inside.

References

Braitenberg, V. (1984). *Vehicles: Experiments in Synthetic Psychology*: MIT Press, Cambridge, MA.

Brody, C. D., & Hopfield, J. J. (2003). Simple Networks for Spike-Timing-Based Computation, with Application to Olfactory Processing. *Neuron, 37*, 843-852.

Di Paolo, E. A. (2002). *Evolving spike-timing dependent plasticity for robot control.* Paper presented at the EPSRC/BBSRC International Workshop: Biologically-inspired Robotics, The Legacy of W. Grey Walter, WGW'2002, Bristol.

Di Paolo, E. A. (2003). Spike-Timing Dependent Plasticity for Evolved Robots. *Adaptive Behavior, 10*, 243-263.

Floreano, D., & Mattiussi, C. (2001). Evolution of Spiking Neural Controllers for Autonomous Vision-Based Robots. *Proceedings of the international Symposium on Evolutionary Robotics From intelligent Robotics To Artificial Life, 2217, T. Gomi, Ed. Lecture Notes In Computer Science Springer-Verlag, London, 38-61.*

Florian, R. V. (2003). Biologically inspired neural networks for the control of embodied agents. *Technical report Coneural-03-03 Version 1.0.*

Florian, R. V. (2006). *Spiking neural controllers for pushing objects around.* Paper presented at the Proceedings of the Ninth International Conference on the Simulation of Adaptive Behavior (SAB'06), Rome.

Gerstner, W., & Kistler, W. M. (2002). *Spiking Neuron Models. Single Neurons, Populations, Plasticity*: Cambridge University Press.

Hänggi, P. (2002). Stochastic resonance in biology: how noise can enhance detection of weak signals and help improve biological information processing. *Chemphyschem 3*(3), 285-290.

Hopfield, J. J. (1999). Odor space and olfactory processing: Collective algorithms and neural implementation. *PNAS., 96*(22), 12506-12511.

Hoshino, O., Kashimori, Y., & Kambara, T. (1998). An olfactory recognition model based on spatio-temporal encoding of odor quality in the olfactory bulb. *Biological Cybernetics, 79*, 109-120.

Izhikevich, E. M. (2003). Simple model of spiking neurons. *IEEE Transactions on Neural Networks*.

Izhikevich, E. M. (2004). Which model to use for cortical spiking neurons. *IEEE Transactions on Neural Networks*.

Jacobi, N., Husbands, P., & Harvey, I. (1995). Noise and the Reality Gap: The Use of Simulation in Evolutionary Robotics, *Proceedings of the Third European Conference on Advances in Artificial Life*: Springer-Verlag.

Kandel, E. R., Schwartz, J. H., & Jessell, T. M. (2000). *Principles of Neural Science* (4th ed.): McGraw-Hill.

Kanzaki, R., Nagasawa, S., & Shimoyama, I. (2005). Neural Basis of Odor-source Searching Behavior in Insect Brain Systems Evaluated with a Mobile Robot. *Chemical Senses, 30*(1), 285-286.

Koch, C. (1999). *Biophysics of Computation: Information Processing in Single Neurons*: Oxford University Press: New York, New York.

Kuwana, Y., & Shimoyama, I. (1998). A Pheromone-Guided Mobile Robot that Behaves like a Silkworm Moth with Living Antennae as Pheromone Sensors. *The International Journal of Robotics Research, 17*, 924-933.

Laurent, G., Wehr, M., & Davidowitz, H. (1996). Temporal Representations of Odors in an Olfactory Network *Journal of Neuroscience, 16*(12), 3837-3847.

Mori, T., & Kai, S. (2002). Noise-Induced Entrainment and Stochastic Resonance in Human Brain Waves. *Physical Review Letters, 88*(21), 218101.

Moss, F., Ward, L., & Sannita, W. (2004). Stochastic resonance and sensory information processing: a tutorial and review of application. *Clin Neurophys, 27*, 677– 682.

Oros, N., Steuber, V., Davey, N., Cañamero, L., & Adams, R. G. (2008). *Optimal receptor response functions for the detection of pheromones by agents driven by spiking neural networks.* Paper presented at the 19th European Meetings on Cybernetics and Systems Research, Vienna.

Payton, D., Daily, M., Estowski, R., Howard, M., & Lee, C. (2001). Pheromone Robotics. *Auton. Robots, 11*(3), 319-324.

Pyk, P., Bermúdez i Badia, S., Bernardet, U., Knüsel, P., Carlsson, M., Gu, J., et al. (2006). An artificial moth: Chemical source localization using a robot based neuronal model of moth optomotor anemotactic search. *Autonomous Robots, 20*(3), 197-213.

Rudolph, M., & Destexhe, A. (2003). Characterization of subthreshold voltage fluctuations in neuronal membranes. *Neural Computation, 15*(11), 2577-2618.

Uhlenbeck, G. E., & Ornstein, L. S. (1930). On the theory of Brownian Motion. *Physical Review, 36*, 823-841.

Webb, B. (1998). Robots crickets and ants: models of neural control of chemotaxis and phonotaxis. *Neural Networks, 11*, 1479-1496.

Wiesenfeld, K., & Moss, F. (1995). Stochastic resonance and the benefits of noise: from ice ages to crayfish and SQUIDs. *Nature, 373*, 33–36.

Wyatt, T. D. (2003). *Pheromones and Animal Behaviour, Communication by Smell and Taste*: Cambridge University Press.

Evolution of Bilateral Symmetry in Agents Controlled by Spiking Neural Networks

Nicolas Oros, Volker Steuber, Neil Davey, Lola Cañamero, and Rod Adams

Science and Technology Research Institute, University of Hertfordshire, AL10 9AB, United Kingdom

{N.Oros, V.Steuber, N.Davey, L.Canamero, R.G.Adams}@herts.ac.uk

***Abstract*—We present in this paper three novel developmental models allowing information to be encoded in space and time, using spiking neurons placed on a 2D substrate. In two of these models, we introduce neural development that can use bilateral symmetry. We show that these models can create neural controllers for agents evolved to perform chemotaxis. Neural bilateral symmetry can be evolved and be beneficial for an agent. This work is the first, as far as we know, to present developmental models where spiking neurons are generated in space and where bilateral symmetry can be evolved and proved to be beneficial in this context.**

I. Introduction

In order to investigate the importance of bilateral symmetry in artificial neural development, here we introduce three different novel models of a developmental program that grow spiking neural networks on a two-dimensional substrate. Each of these models has different degrees of allowed or enforced symmetry. These developmental programs are evolved, using a genetic algorithm, to allow simulated agents to perform chemotaxis. This paper begins with a basic introduction on symmetry in nature and how it has been modeled in artificial evolutionary models. Then, we introduce our developmental model, the agent used and the task it had to perform. Further, we describe in more detail our three different models. Then, the simulation and genetic algorithm parameters are presented. This section is followed by the results, the discussion and finally the conclusion.

A. Symmetry

For centuries, people have observed and been fascinated by symmetrical patterns found in nature [1-3]. In our minds, symmetry is often related to something beautiful, well balanced or well proportionate [3]. It has been shown that in many species (even in humans), female prefer males that have symmetrical displays [4]. One possible reason to explain this phenomenon is that symmetry might reflect the high quality of a signaler. Another reason could be that individuals have evolved recognition systems that have common properties and are capable of generalization, and from this could emerge a high sensitivity to symmetries [4]. In living organisms, symmetries arise as a side effect of the creation of axes that will guide cells during development [1-3, 5-8]. Cells divide and migrate following gradients that form these axes. They might also create or modify gradients and rearrange themselves to form the most thermodynamically stable pattern [6]. Therefore, it is very likely that cells will be placed symmetrically along different axes to have a system in a state of equilibrium [3]. But due to developmental noise, even the most bilaterally symmetrical animals do not show perfect symmetry. Also, many vertebrates are mainly bilaterally symmetrical about the midline of the body but they have many internal organs that are not bilaterally symmetrical (for example in humans: heart, stomach, spleen…) [5-8]. Even if the emergence of a bilateral body plan was a key step in evolution, new axes were defined that differentiated head and foot, back and front and left from right, and allowed asymmetrical parts to be created and eventually lead to more complex organisms.

B. Evolutionary Computation

In order to understand the importance of symmetry in development, certain researchers in artificial intelligence have created abstract developmental models that generate neural controllers for robots or simulated agents. It is always a difficult task to create robust and adaptable neural controllers for agents that can perform many different actions. It is even more difficult if you want to reuse existing controllers and add new modules so an agent can learn and perform new tasks. A promising trend is to evolve neural networks using evolutionary computation. There are different approaches in this research area and many different ways to encode evolving features into genes [9-14]. A certain amount of work has been done in evolutionary computation on encoding spatial neural networks [15], with symmetrical structure using L-systems [16-21] and grammatical encoding [22, 23]. Stanley also created abstract models generating representations of symmetrical patterns [24, 25]. To the best of our knowledge, no one has created a developmental model generating spiking neural controllers placed in 2D spaces, where bilateral symmetry can be evolved, and improved the performance of an agent to perform certain tasks.

C. Our Approach

In this study, we used developmental programs that allowed information to be encoded as spatio-temporal neural activity patterns. We created three new developmental models initially inspired by Kodjabachian and Meyer's

978-1-4244-2763-5/09/$25.00 ©2009 IEEE

SGOCE paradigm [26-30] and NEAT [24, 25, 31]. By using them, we wanted to see how bilateral symmetry in neural networks could be generated and affect the behavior of a simulated robot. In our models, a developmental program was expressed in a genome and when executed, it would create one or more intermediate neurons with one or more connections to make the whole neural network grow. Like in [31, 32], one of the key ideas in our approach is based on complexification. An initial genome is first composed of only one gene creating only one neuron when expressed. Then, during evolution, new genes can be added via mutations creating more neurons and more connections, therefore adding more complexity to the system. Another important concept of this model is targeting [32]. We used a 2D neural substrate where spiking neurons (with synaptic integration and conduction delays) are placed and can grow connections to target locations. Evolution can therefore generate neural networks able to encode external information as spatio-temporal patterns.

We first created a model where parameters of each neuron were encoded in the genome (NO_SYM). We then created two variations of it allowing bilateral symmetrical clones of neurons to be created. The first one allowed the evolution of symmetrical neurons (EVO_SYM) and the second one enforced the symmetry for every neuron (ENF_SYM). We also decided to have neural development performed in two stages: first creating every neuron on the substrate, then creating all the connections. This was inspired by biological systems where neurons first divide, then migrate to a certain location and finally create connections [5, 6, 33]. Some neurons might eventually die but we decided not to model apoptosis in our model to deal with complexity incrementally.

D. The agent and its Task

We decided to evolve an agent to perform a simple task which was to stay inside a chemical concentration in a simulated continuous environment (Fig. 3). The agent has two wheels, one on each side of the agent, providing a differential steering system. Each wheel is controlled by two motor neurons providing forward and backward propulsion. The agent also has two antennae placed on the front of the agent, one orientated on the left and the other one on the right. Each antenna is linked to a sensory neuron. The two antennae are separated widely enough to detect the presence of the chemical gradient (Fig. 1). To control the agent, we used a spiking neural network. The sensory and motor neurons placed on the neural substrate form the initial neural network (Fig. 2).The complete neural network was created by using a developmental program.

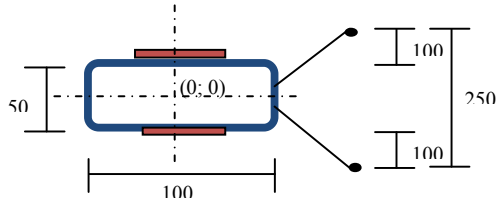


Fig. 1. Properties of an agent equipped with two wheels and two antennae. Units are arbitrary.

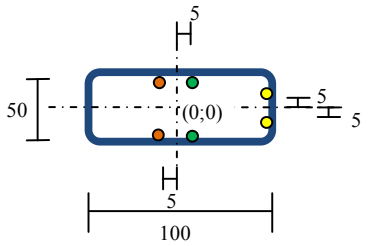


Fig. 2. 2D substrate of an agent with initial neural network. The two sensory neurons are shown on the right in yellow. The motor neurons move the agent forward (green) or backward (orange) by turning the wheel.

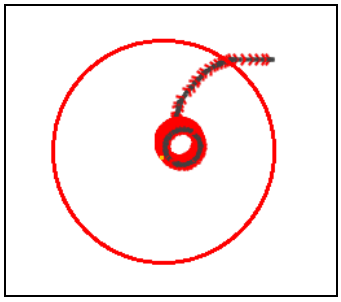


Fig. 3. Path of an agent moving towards the middle of a fixed chemical concentration (red circle). The concentration is a linear gradient where the maximum value is situated in the middle.

II. METHODS

A. Spiking Neurons

We used a leaky-integrate and fire model with synaptic integration and conduction delays already described in [34, 35]. We also used a realistic model of noise in the form of an diffusive OU (Ornstein-Uhlenbeck) current [36]. This form of colored noise reproduces the subthreshold voltage fluctuations in real neuronal membranes. We added this noise to the total input current of each neuron. The noise current $I(t)$ is described by:

$$\frac{dI(t)}{dt} = -\frac{1}{\tau_I}\left(I(t) - I_0\right) + \sqrt{D}\,\xi(t) \qquad (1)$$

where τ_I denotes the current noise time constant (2ms in our case), I_0 is the mean noise current (0 in our case), $D = 2\sigma^2 / \tau_I$ is the noise diffusion coefficient, σ is the standard deviation (0.0007 in our case) and $\xi(t)$ is a Gaussian white noise (with mean = 0 and standard deviation = 1). The motor neurons used to control the wheels are modeled in the same way. However, the sensory neurons are based on this model but have a different expression to calculate the input current. We created a model of a spiking sensory neuron in which the chemical concentration is processed so that a quasi-linear relationship between the concentration and the firing rate of the sensor is produced. The sensory neurons were already described in detail in [34, 35, 37].

B. Chemical Concentration

We decided to use a simple model of chemicals that are not diffused and evaporated. The concentration is a linear

gradient where the maximum value is situated in the middle of the circular chemical concentration.

C. Agent Movements

In order to move the agent, we calculated the velocity $V(t)$ (arbitrary unit) of each wheel using the following equation:

$$\tau_{motor}\frac{dV}{dt} = (V_0 - V) + K_v\left(\delta(t - t_f) - \delta(t - t_b)\right) \quad (2)$$

Where δ is the Dirac function (pulse) defined by $\delta(x) = 0$ when $x \neq 0$ and $\delta(x) = 1$ when $x = 0$.

We decided for simplicity that an agent should always move forward in the absence of any external input so we set up the parameters accordingly: $V_0 = 0.5$ is the default velocity (the agent is always moving straight by default), $K_v = 5$ is the speed coefficient, $\tau_{motor} = 0.05$ is the time constant in seconds, t_f is the time when the most recent spike was emitted by the motor neuron responsible to turn the wheel forward, t_b is the time when the most recent spike was emitted by the motor neuron responsible to turn the wheel backward. The agent was moved by calculating the velocity every time step.

III. Developmental Programs

A. Without Symmetry: NO_SYM

The developmental program constructing the neural network consists of a genome which is an array of modules. A module must have a gene, which we denote *N*, encoding the position (*x, y*) of an intermediate neuron, and can have genes encoding possible connections, denoted *C*. The neuron is placed on a 2D Cartesian coordinates system with its origin situated in the centre of the agent (Fig. 1 and 2). If a new module is created, it will be added to the end of the genome. A module is valid if it is composed of only one *N* gene but not if it is only composed of *C* genes. A *C* gene encodes the different parameters for a connection of a neuron. That includes an angle θ and a distance *d* to determine where it connects (see Section III. B), a synaptic strength (*w*) and a *type* (afferent or efferent). A neuron can also have connections even if they are not encoded in the module defining its properties. The reason is that other neurons can create efferent and afferent connections to this neuron.

When an agent is created, it only has an initial neural network (Fig. 2). There are no intermediate neurons, only motor neurons and sensors. If the genome of an agent is composed of at least one module, the complete neural network can be created by executing the developmental program expressed in the genes, reading the genome from the beginning to the end. With only one module, only one intermediate neuron will be created but it can have more than one connection. The neural network is constructed by the developmental program in two steps by reading the genome twice. First, all the neurons are created in the 2D substrate by reading all the *N* genes. Secondly, all the connections are created by reading all the *C* genes.

When reading a *C* gene, a target position for a given neuron is defined to determine to which neuron it will be connected to. The target position is given by the angle θ (in radians) and the distance parameter *d* relative to the neuron. A neuron creates a connection to the closest cell to this target position (Fig. 4). Self connections are therefore possible. Motor neurons cannot have output connections and sensory neurons cannot have input connections. A target position can be situated outside the substrate. In this case, a connection will still be created linking the closest neuron.

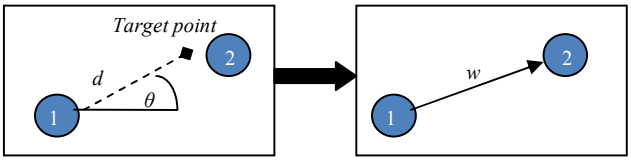


Fig. 4. Creation of a connection by neuron 1 in two steps. First, neuron 1 places a target point on the substrate depending on the distance *d* and angle θ parameters. Secondly, the closest neuron to the target point gets connected to neuron 1. The type of connection (input or output) depends on the parameter *type* and the synaptic strength (weight) is encoded by the parameter *w*.

TABLE 1
RANGES OF VALUES USED FOR THE PARAMETERS OF THE GENES

Parameters	Ranges of values
x	[-50,50]
y	[-25,25]
w	[-15,15]
θ	[0,2π]
d	[1,100]
type	afferent / efferent

B. Evolvable Symmetry: EVO_SYM

This model is a modification of NO_SYM. The main concept of this model is to introduce genetically encoded bilateral symmetry with respect to the longitudinal axis of the agent. The idea is that instead of encoding two neurons that are similar but are positioned on opposite sides of the midline (*x-axis*), the genome could encode only one neuron but with an extra evolvable parameter allowing the creation of its symmetrical clone; this allows compressing genetic information. In fact, the initial neural network is symmetrical, and therefore the evolutionary process should be able to use this important embedded feature. This model is based on an abstraction of a gradient that could form the horizontal axis. Compared to NO_SYM, *C* genes are still the same but *N* genes have an additional Boolean parameter *sym*. This parameter *sym* plays an important role. If it is activated (set to true), a clone of the actual neuron will be created and placed symmetrically to the *x-axis* (Figs. 5 & 6). If the parent neuron is situated on the *x-axis*, its clone will be created in a close random place around it.

The development of the neural network is very similar to NO_SYM. The only difference is that during the first step of development (creation of neurons), each created neuron will have a symmetrical clone if its parameter sym is set to true.

A clone of a neuron has its y parameter set to - y and all the connection parameters θ set to - θ. Therefore, the clone of a neuron is horizontally symmetric and its connections are also symmetric (Figs. 5 & 6). The neural growth is still performed in two steps by reading the genome twice. First, all the neurons (and their possible symmetrical clones) are created in the 2D substrate by reading all the N genes. Secondly, all the connections are created by reading all the C genes.

C. Enforced Symmetry: ENF_SYM

This developmental model is almost the same as NO_SYM. The only difference is the systematic creation of a symmetrical clone for every neuron. Every time a neuron is added to the substrate by executing the genome, a symmetrical clone is also created, as in EVO_SYM (Figs. 5 & 6). But compared to EVO_SYM, ENF_SYM does not encode the possible symmetry in the genome. The creation of symmetrical neurons is an automatic process always occurring during the first step of the development of the neural network.

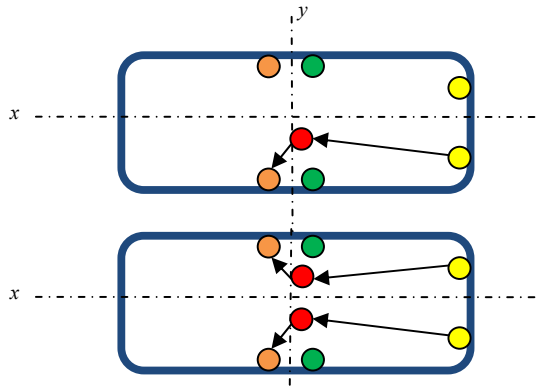


Fig. 5. At the top, the 2D substrate of an agent with the initial neural network and one intermediate neuron (in red) having two connections is shown. At the bottom, the symmetrical clone has been added.

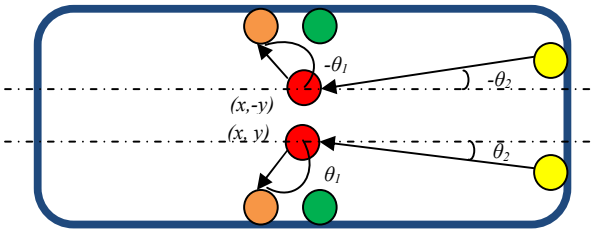


Fig. 6. Drawing showing the coordinates and the angle of the connections of the intermediate neuron (bottom) and its symmetrical clone (top).

IV. Experiments

We performed two series of tests. First, we evolved simulated agents that stay in a fixed chemical concentration. We then evolved agents to stay inside a moving concentration. In each series, we performed seven GA runs for each developmental model in order to study the importance of symmetry in neural development.

For the first series of tests, each agent had two runs of 200 seconds and started from different locations (left and right of the fixed chemical concentration). The fitness function was very simple and consisted of the sum of the inverse distances between the agent and the centre of the concentration during the last 50s of a run. The fitness of an agent was the sum of the fitness values recorded for the two runs.

For the second series of tests, we evolved agents able to stay within a moving concentration. One agent and one chemical source were placed in a toroidal world. Compared to the first series, the time of a run was longer (300s). During a run, an agent was always placed at the same place with a random angle of initial movement and the chemical concentration was placed randomly in the world. The concentration was then set moving randomly in the environment. The fitness function was also different and started to be calculated only when the agent was touching the concentration (recording time was initialized at this point). The fitness is the sum of the inverse distances, divided by the recording time. We used a resolution of 1ms (1 time step) for every simulations.

A. Genetic Algorithm

We used a classical genetic algorithm (Fig. 7) to evolve an agent that could perform chemotaxis. The initial population was composed of 100 agents. Each one of them was equipped with four motor neurons and two sensors composing the initial neural network (Fig. 2). Initially, they all had a genome composed of one module encoding one neuron, placed in the middle of the substrate, and one initial connection, having parameters randomly initialized. Therefore, the genome of these agents had one module composed of one N gene and one C gene. Then, each agent was subject to mutations (see Section IV.B). After mutations, these agents were placed in the initial population and the GA could begin. Once all the agents were evaluated, the agents were ranked by fitness and the ten fittest ones were copied to the next generation. Ninety new individuals were created and added to the next generation‘s population by selecting two parents for each, using a tournament selection of size 5. A new individual was created by cross-over of the two parents (see Section IV.B.). Out of these 90 new agents, twenty were mutated. The genetic algorithm lasted for 1000 generations.

B. Genetic Operators

The use of the following genetic operators allowed complexification of the genome by adding, modifying or removing new genes.

Mutation - In our model, mutations occur with the same probability independently of the size of the genome. Twenty agents were randomly chosen from the 90 new agents created by the tournament selection and mutated. Three kinds of mutations were used in this GA. A mutation could add or delete neurons, add or delete connections and modify the values of the parameters of the genes. Each mutation was performed within a certain range of values added to the

original ones (Tables 2 & 3), and these parameters were maintained within certain values defined earlier (Table 1). For example, if the value of the parameter x of a N gene was 49, and a mutation tried to add 5 to x (x = 54), x would be set to its maximum value 50 due to the range of values used. Here is the simple algorithm of the mutation process:

For all twenty agents:

I. Mutate each module:
   1. 5% chances to add a new connection.
   2. 5% chances to remove a randomly selected connection.
   3. Choose randomly one of the following mutations:
      - Pick randomly one connection, choose randomly one parameter and mutate it.
      - Add a random value to parameter x of a N gene.
      - Add a random value to parameter y of a N gene.

II. 5% chance to add a new module (new neuron).

III. 5% chance to remove a randomly selected module (neuron and connections).

When a new module is added to the genome, the new neuron always has one randomly initialized connection. The new neuron is placed randomly in the vicinity of the last neuron created on the substrate (last encoded in the genome). A new connection added is also always randomly initialized.

TABLE 2
RANGES OF MUTATIONS USED FOR THE PARAMETERS OF THE GENES

Parameters	Ranges of mutation
x	[-5;5]
y	[-5;5]
sym (only for EVO_SYM)	true / false
θ	[-π/4;π/4]
d	[-2;2]
w	[-5;5]
type	afferent / efferent

Cross-over - Neural selection is applied here by crossing-over modules at the same position. By doing so, each neuron should be able to specialize more quickly during evolution. Here is an example:

Two agents A_1 and A_2 are selected to create a new agent A_3. The maximum number of modules a new agent can have depends on its parents. In this case, agent A_1 has five modules and A_2 has three of them. Therefore, the new agent A_3 will have at maximum five modules (i.e. five neurons). The crossover process will make five selections of modules and at each selection, there is an equal chance of selecting the agent A_1 or A_2. Therefore, each module of the same position has 50% chances to be selected and copied. If at the fourth selection, for example, the chosen agent is A_2, which does not have any more modules at this position, nothing will be added to the genome of agent A_3 at this stage. But another module can be copied from A_1 if this one is chosen during the fifth selection, and this module, originally from position 5, will become a module of position 4 of agent A_3.

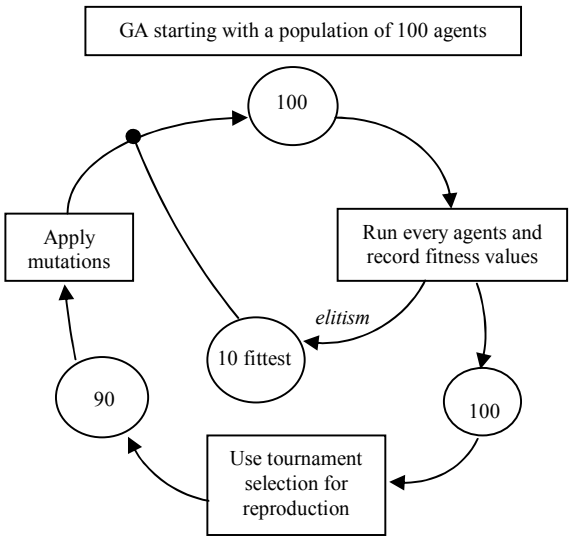


Fig. 7. Genetic algorithm with parameters. The population is composed of 100 agents. For the first experiment, each agent had two runs of 200 seconds starting from different places (left and right of the chemical concentration). In the second experiment, each agent had only one run of 300s. For both experiments, the fitness rewarded an agent that stayed inside the chemical gradient. Once all the agents were evaluated, the agents were ranked by fitness and the ten fittest ones were copied to the next generation. Ninety new individuals were created and added to the next generation's population by selecting two parents for each, using a tournament selection of size five. A new individual was created by cross-over of the two parents (see Section IV.B.). Out of these 90 new agents, twenty were mutated. The genetic algorithm lasted for 1000 generations. We ran the GA for 1000 generations.

V. RESULTS

In the first series of tests, we evolved agents to stay inside a chemical concentration as close as possible from its center (Fig. 3). We found that the GAs implementing the developmental models using symmetry (EVO_SYM, ENF_SYM) evolved good neural networks so the agent went in and stayed close to the centre of the concentration. We saw that in all the seven GAs, EVO_SYM evolved a neural network with symmetrical neurons. In fact, the neural controllers evolved with EVO_SYM or ENF_SYM were very similar. NO_SYM did not manage to evolve an optimal solution as the others and had an overall pretty bad performance. Therefore, the first series of tests showed us that without evolvable or enforced symmetry, the system could not evolve and find an optimal solution. In Fig. 8, we can see the neural controller of the fittest agents evolved using NO_SYM. We can clearly see that it is not bilaterally symmetric and in fact, the agent implementing it performed rather badly. Fig. 9 shows the neural network of the fittest agent evolved using EVO_SYM. This agent was performing well and used both sensors and motor neurons. It used only two symmetrical neurons where only one neuron was encoded in the genome. Neuron N1 is taking an input from

the sensor S0 and is stimulating M0 and M3 and is inhibiting M1 allowing the agent to turn quickly. N1 also has an excitatory self connection. Neuron N2 has the same symmetrical connections. We also noticed that both neurons are inhibiting each other. This neural network can be seen as an advanced Braitenberg vehicle [38]. The trajectory of this agent is shown in Fig. 3.

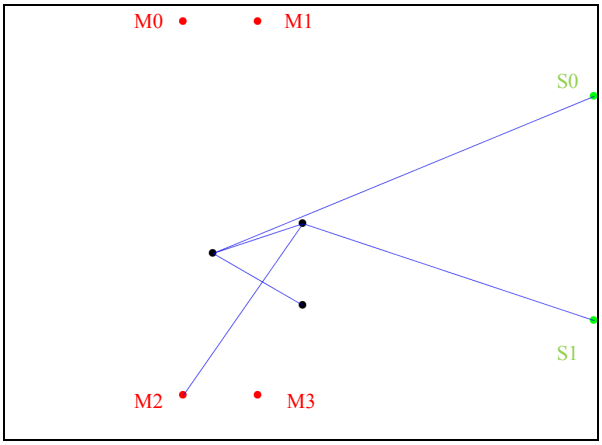


Fig. 8. Neural network of the fittest agent using NO_SYM evolved to stay in a fixed concentration. Motor neurons are depicted in red, sensory neurons in green and intermediate neurons in black.

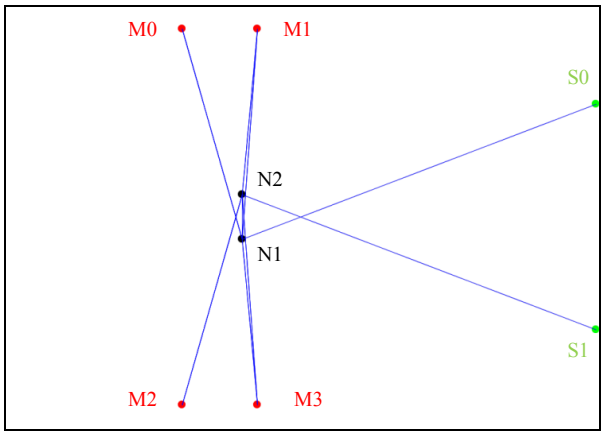


Fig. 9. Neural network of the fittest agent using EVO_SYM evolved to stay in a fixed concentration. Motor neurons are depicted in red, sensory neurons in green and intermediate neurons in black.

In the second series of test, we saw again that the developmental models using bilateral symmetry generated better neural controllers than NO_SYM (Fig. 11). Once more, EVO_SYM always evolved a neural network with symmetrical neurons in all the seven GAs. Fig. 10 and Table 3 show the neural controller of the fittest agent evolved using EVO_SYM. Neuron N1 is taking an inhibitory input from sensor S0 and an excitatory input from S1. It is stimulating M2 and is inhibiting N4 allowing the agent to turn more quickly as N4 stimulates M0. Neuron N2 is symmetrical to N1 so it has the same symmetrical connections. We also noticed that both neurons are inhibiting each other. Neuron N3 takes input from the sensor S1 and stimulates the motor neurons M1 and M2 so the agent can turn quickly. Neuron N4 is symmetrical to N3 so it has the same symmetrical connections. We notice that two other symmetrical neurons (N5 and N6) and a non symmetrical neuron (N7) exist but they do not modify the overall neural activity of the controller. This shows that symmetrical neurons (N1 and N2) can also have asymmetrical connections. N5, 6 and 7 can be seen as evolutionary artefacts that could become useful in time or disappear. This neural network has more complexity that the one shown in Fig. 9. The main differences between the two are the two layers of neurons and inhibitory connections coming from the sensors. We also noted that neurons N3 and N4 created more than one connection to the motor neurons. We suppose that it is due to the limit values the weights can have [-15; 15] (see Table 1). Therefore, we can see that the system can easily adapt to circumvent certain constraints.

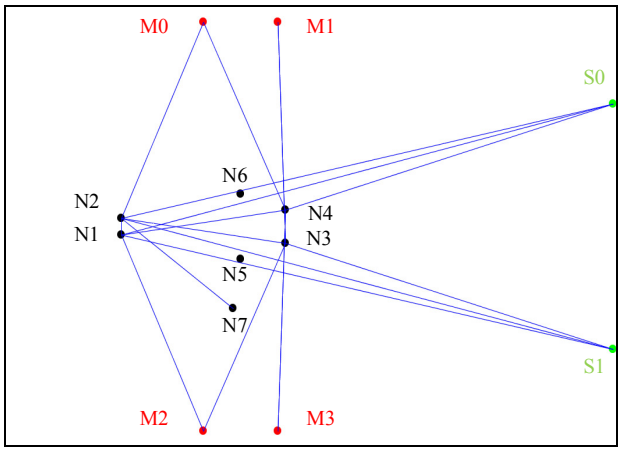


Fig. 10. Neural network of the fittest agent using EVO_SYM evolved to stay in a moving concentration. Motor neurons are depicted in red, sensory neurons in green and intermediate neurons in black.

TABLE 3
WEIGHT MATRIX OF THE NN FROM FIG. 9 SHOWING THE CONNECTIONS LINKING CELLS (TOP ROW) TO OTHER CELLS (LEFT COLUMN)

Cells	S0	S1	N1	N2	N3	N4	N7
M0				15		15	
M1					6, 15		
M2			15		15		
M3						6, 15	
N1	-6	9		4, -15			
N2	9	-6	-15				4
N3		6		-10			
N4	6		-10				
N7							-8

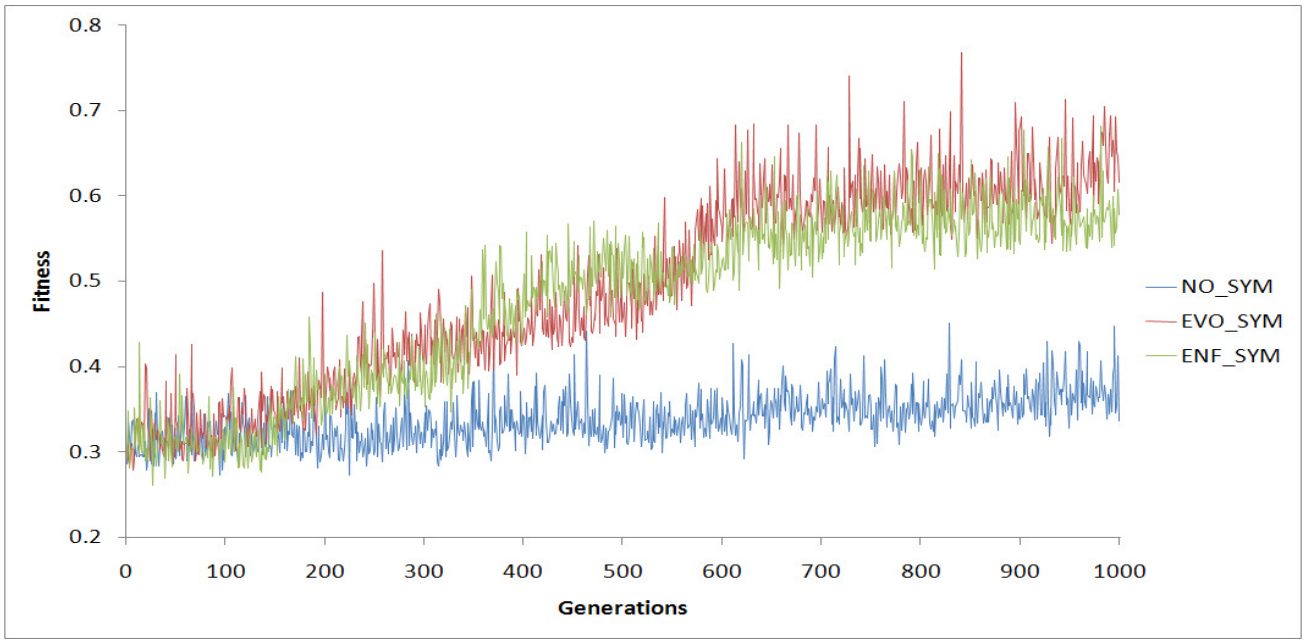


Fig. 11. Fitness mean values over seven GA runs of fittest agents per generation. The agents were evolved to perform chemotaxis with a moving target. This graph shows that the use of bilateral symmetry (EVO_SYM and ENF_SYM) created neural controllers performing considerably better than without symmetry (NO_SYM).

VI. Discussion

In this paper, we have shown that bilateral symmetric neural networks can be evolved using a genetic algorithm and our developmental models, and have better performances than non-symmetrical ones. Perhaps this is not surprising. Firstly, the agent has bilaterally placed sensors and actuators. Secondly, the task of chemotaxis also has implicit symmetry: a chemical to the left triggers a turn to the left and symmetrically, a chemical to the right triggers a turn to the right.

Complexification, targeting and neural selection are important concepts in our model. We use a 2D neural substrate where spiking neurons are placed and can grow connections to target locations. Therefore, the geometric configurations of the neural network significantly matter. Since we use spiking neurons with conduction delays, distances separating connected neurons encode time delays between the points in time spikes are sent by a neuron, and the time they are received by another neuron. A neural network generated by our developmental models can encode information not only using firing rate encoding but also using temporal coincidence or delay encoding [34, 35, 39]. Evolution can therefore generate neural networks able to encode external information as spatio-temporal patterns. More detailed analysis of the activity of the different neural networks that evolved will be done in the future to see which neural encodings were really used.

We have noticed from our results that sometimes more than one connection linking two cells was created. This is due to the limits of the weights used, showing that the system can easily adapt to certain limiting constraints. We have seen that connections between symmetrical parts of the neural controller could be connected and inhibitory connections of symmetrical neurons were often evolved. Also, neural controllers grown with NO_SYM could have symmetrical neurons, but did so with an extremely low probability.

We have to emphasize the fact that the initial neural network, placed on the substrate, is bilaterally symmetrical. Most physical robots are also bilaterally symmetric, and therefore, we assume that mapping sensors and motors to sensory and motor neurons on the neural substrate could be done in a direct manner when implementing our model on a simulated and real robot. In this case, it biased evolution to find an appropriate solution that uses this embedded symmetry. It would be very interesting to see if bilateral symmetry would still arise and be beneficial when evolving the morphology of the agent as well as the neural substrate. Cells could migrate on the substrate and differentiate to become sensors, neurons and motor neurons.

Many modifications of this model can be done. For example, adding the possibility to encode the threshold of a neuron or different axes of symmetry in the genome. Other developmental models could have been created where only one gene could have created symmetrical neurons for the entire neural network. However, we decided to use EVO_SYM to permit the creation of both symmetrical and asymmetrical parts, and therefore to increase complexity.

VII. Conclusion

In this paper, we have presented three novel developmental models allowing information to be encoded in space and time using spiking neurons placed on a 2D substrate. In two of these models, we introduce neural development that could use bilateral symmetry. We showed

that these models created neural controllers that permit agents to perform chemotaxis, and do so better than controllers with no symmetry. We also have showed that with EVO_SYM, neural bilateral symmetry was often evolved and was found to be beneficial for the agents. This work is the first, as far as we know, to present developmental models where spiking neurons were generated in a 2D space and where bilateral symmetry could be evolved and was proved to be beneficial in this context.

In future work, we will use incremental evolution to generate agents that can perform more than one task. Our long term interest is to study the emergence of chemical communication in a population of artificial agents.

REFERENCES

[1] D. A. Thompson, *On Growth and Form*: Cambridge University Press (1992), 1917.

[2] H. Meinhardt, *Models of Biological Pattern Formation*: Academic Press, 1982.

[3] H. Weyl, *Symmetry*: Princeton Press, 1952.

[4] M. Enquist and R. A. Johnstone, "Generalization and the evolution of symmetry preferences," *Proceedings of the Royal Society, Biological Science,* vol. 264, pp. 1345-1348, 1997.

[5] W. Arthur, *The Origin of Animal Body Plans, A Study in Evolutionary Developmental Biology*: Cambridge University Press, 1997.

[6] S. F. Gilbert, *Developmental Biology, Eighth Edition*, 8 ed.: Sinauer Associates, 2006.

[7] A. R. Palmer, "Symmetry Breaking and the Evolution of Development," *Science,* vol. 306, pp. 828-833, 2004.

[8] L. Wolpert, "Development of the asymetric human," *European Review,* vol. 13, pp. 97-103, 2005.

[9] Y. Dongyong and S. i. Yuta, "Evolving Mobile Robot Controllers Using Evolutionary Algorithm," in *SICE 2002, Proceedings of the 41st SICE Annual Conference*. vol. 4, 2002, pp. 2184-2189.

[10] D. Floreano, P. Husbands, and S. Nolfi, "Evolutionary Robotics," in *Springer Handbook of Robotics, Chapter 61*, B. Siciliano and O. Khatib, Eds.: Springer, 2008.

[11] R. Pfeifer, F. Iida, and J. Bongard, "New Robotics: Design Principles for Intelligent Systems," *Artificial Life, Special Issue on New Robotics, Evolution and Embodied Cognition,* vol. 11, pp. 99-120, 2005.

[12] D. Floreano, P. Dürr, and C. Mattiussi, "Neuroevolution: from architectures to learning," *Evolutionary Intelligence,* vol. 1, pp. 47-62, 2008.

[13] E. Ruppin, "Evolutionary Autonomous Agents: A Neuroscience Perspective," *Nature Reviews Neuroscience,* vol. 3, pp. 132-141, 2002.

[14] X. Yao, "Evolving Artificial Neural Networks," *Proceedings of the IEEE,* vol. 87, pp. 1423-1447, 1999.

[15] S. Nolfi, O. Miglino, and D. Parisi, "Phenotypic Plasticity in Evolving Neural Networks," Technical Report PCIA-94-05, Institute of Psychology, C.N.R., Rome 1994.

[16] A. Channon and R. I. Damper, "The Evolutionary Emergence of Socially Intelligent Agents," in *Socially Situated Intelligence: a workshop held a SAB'98*, 1998.

[17] A. Channon and R. I. Damper, "Evolving Novel Behaviours via Natural Selection," in *Proceedings of the 6th Conference on the simulation and synthesis of living systems (ALIFE VI)*, Los Angeles, CA, USA, 1998, pp. 384-388.

[18] T. Miconi and A. Channon, "A virtual creatures model for studies in artificial evolution," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2005)*, Edinburgh, UK, 2005.

[19] T. Miconi and A. Channon, "An Improved Sytem for Artificial Creatures Evolution," in *Proceedings of the 10th Conference on the simulation and synthesis of living systems (ALIFE X)*, Bloomington, Indiana, USA, 2006.

[20] G. S. Hornby and J. B. Pollack, "Body-Brain Co-evolution Using L-systems as a Generative Encoding," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, 2001.

[21] K. Sims, "Evolving 3d morphology and behavior by competition," in *Proceedings of the 4th International Workshop on the Synthesis and Simulation of Living Systems (ALIFE IV)*, 1994, pp. 28-39.

[22] M. Komosinski and A. Rotaru-Varga, "Comparison of Different Genotype Encoding for Simulated 3D Agents," *Artificial Life,* vol. 7, pp. 395-418, 2001.

[23] F. Gruau, "Genetic synthesis on modular neural networks," in *Proceedings of the 5th International Conference on Genetic Algorithms*, San Francisco, CA, USA, 1993, pp. 318-325.

[24] J. Gauci and K. O. Stanley, "Generating Large-Scale Neural Networks Through Discovering Geometric Regularities," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2007)*: New York, NY: ACM, 2007.

[25] K. O. Stanley, "Compositional pattern producing networks: A novel abstraction of development," *Genetic Programming and Evolvable Machines Special Issue on Developmental Systems,* vol. 8, pp. 131-162, 2007.

[26] D. Filliat, J. Kodjabachian, and J. A. Meyer, "Incremental evolution of neural controllers for navigation in a 6 legged robot," *Proceedings of the Fourth International Symposium on Artificial Life and Robotics,* vol. In Sugisaka and Tanaka, editors, Oita Univ. Press., 1999.

[27] A. Ijspeert and J. Kodjabachian, "Evolution and development of a central pattern generator for the swimming of a lamprey," *Research Paper No 926,* vol. Dept. of Artificial Intelligence, University of Edinburgh, 1998.

[28] J. Kodjabachian and J. A. Meyer, "Evolution and Development of Neural Controllers for Locomotion, Gradient-Following, and Obstacle-Avoidance in Artificial Insects," *IEEE transactions on neural network,* vol. 9, 1998.

[29] J. Kodjabachian and J. A. Meyer, "Evolution of a Robust Obstacle-Avoidance Behavior in Khepera: A Comparison of Incremental and Direct Strategies," *IEEE transactions on neural network,* vol. 9, 1997.

[30] J. Kodjabachian and J. A. Meyer, "Evolution and development of modular control architectures for 1-d locomotion in six-legged animats," 1997.

[31] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evolutionary Computation,* vol. 10, pp. 99-127, 2002.

[32] K. O. Stanley and R. Miikkulainen, "A Taxonomy for Artificial Embryogeny," *Artificial Life journal,* vol. 9, pp. 93-130, 2003.

[33] E. R. Kandel, J. H. Schwartz, and T. M. Jessell, *Principles of Neural Science*, 4th ed.: McGraw-Hill, 2000.

[34] N. Oros, V. Steuber, N. Davey, L. Cañamero, and R. G. Adams, "Adaptive Olfactory Encoding in Agents Controlled by Spiking Neural Networks," in *Proceedings of The 10th international conference on Simulation of Adaptive Behaviour, From Animals to Animats (SAB)*, Osaka, Japan, 2008, pp. 148-158.

[35] N. Oros, V. Steuber, N. Davey, L. Cañamero, and R. G. Adams, "Optimal noise in spiking neural networks for the detection of chemicals by simulated agents," in *Proceedings on the Eleventh International Conference on Artificial Life*, Winchester, UK, 2008.

[36] G. E. Uhlenbeck and L. S. Ornstein, "On the theory of Brownian Motion," *Physical Review,* vol. 36, pp. 823-41, 1930.

[37] N. Oros, V. Steuber, N. Davey, L. Cañamero, and R. G. Adams, "Optimal receptor response functions for the detection of pheromones by agents driven by spiking neural networks," in *19th European Meetings on Cybernetics and Systems Research*, Vienna, 2008.

[38] V. Braitenberg, *Vehicles: Experiments in Synthetic Psychology*: MIT Press, Cambridge, MA., 1984.

[39] D. Floreano and C. Mattiussi, "Evolution of Spiking Neural Controllers for Autonomous Vision-Based Robots," *Proceedings of the international Symposium on Evolutionary Robotics From intelligent Robotics To Artificial Life,* vol. 2217, T. Gomi, Ed. Lecture Notes In Computer Science Springer-Verlag, London, 38-61., 2001.