5. NEURAL NETWORK

5.1 Introduction

This chapter describes a neural network with 17,544 neurons and 698,625 connections that was created to illustrate and test the theoretical ideas in this project. The first section explains the factors that influenced the design of this network, Section 5.3 gives more details about the modelling and architecture, and Section 5.4 outlines the experimental procedure. Section 5.5 documents the behaviour of the network and the tests that were run on it, and the chapter concludes with some related research in this area and suggestions for future work. The SpikeStream software that was developed to simulate this network is covered in Chapter 6.

5.2 Design

This section looks at some of the decisions that were made about the design of the network, such as the task that it was to carry out, the neuron and synapse models, the size of the network and the software that was used to simulate it.

5.2.1 Task

Although randomly firing neurons can be analyzed for consciousness, it is difficult to describe the phenomenology of a system that lacks systematic relationships with its environment, and so a system was needed that could be analysed for mental states that are functionally or effectively connected to states of a real environment (or a pretty good approximation to it). Since the network was being developed as part of the CRONOS project, the most obvious way to do this was to use the network to control the CRONOS and/ or SIMNOS robots (see sections 1.2.2 and 1.2.3). Although I wanted to test the network on CRONOS as well as SIMNOS, considerable delays in the development of a software interface for CRONOS prevented me from using CRONOS in this PhD.

One of the main aims of this network was the development of something that could be plausibly analyzed for consciousness using Tononi's (2004), Aleksander's (2005) and Metzinger's (2003) theories (see Section 2.6). Whilst the amount of consciousness predicted by Tononi's (2004) theory is largely independent of the network's functionality, both Aleksander (2005) and Metzinger (2003) make explicit links between particular cognitive mechanisms and consciousness, and to increase the likelihood of consciousness in the network it was decided to incorporate some of these mechanisms into it. Since there was considerable overlap between Aleksander's axioms and Metzinger's constraints, and it was difficult to see how some of Metzinger's constraints could be implemented,[1] it was decided to base the network on the cognitive mechanisms specified by Aleksander's axioms. Some of the requirements for a network that implements these axiomatic mechanisms are as follows:

1. *Depiction*. The network would have to include neurons that were sensitive to combinations of sensory and motor information.
2. *Imagination*. The network would have to be able to operate in an offline as well as an online mode. Some form of inhibition of sensory input and motor output could be used to enable the network to operate in isolation from its environment. The network would also have to be capable of changing between online and offline modes in response to its perceptual and imaginative states.
3. *Attention*. The network would have to be able to 'focus' on different parts or aspects of the world.

[1] Transparency is particularly difficult since Metzinger has few suggestions about how it is implemented in the brain.

4. *Volition*. The activity of the network would have to be used to select actions. The use of an 'imagination' mode would enable the perceptual circuitry to be used for planning and a model of the emotions would be needed to evaluate the different actions.

5. *Emotion*. A representation of the system's emotional states would have to be included. Ideally this would be a representation of the states of the system's body, but since SIMNOS only has joint and muscle sensors, this could be a representation of the emotions that the system would experience if it were to carry out that action – something like the 'as if' loop discussed by Damasio (1995).

Once the general functional requirements of the network had been established, the next problem was to select a task that the network could carry out which would utilize all of these mechanisms. The task chosen for this system was the control of SIMNOS's eye movements, with the network's offline states being used to plan which part of the visual field is looked at next. This choice was influenced by O'Regan and Noë's (2001) theories about eye movements and by the research on active vision in experimental psychology (Findlay and Gilchrist 2003). Since this task involves sensory and motor data, it was a good way of implementing Aleksander's depiction axiom and the system's limited field of view meant that it was also a rudimentary form of attention. Accurate or detailed visual perception was not a priority in this project, and so a very basic visual system was used and SIMNOS's environment was populated with a red and blue cube. How the neural network was designed to carry out this task is explained in detail in Section 5.3.

A final desirable property of the network was that it should implement at least one of the models of conscious action put forward in Section 2.7. Since discrete conscious control could be implemented more easily than conscious will, it was decided to focus on conscious control for this system.[2] Whether the system is actually capable of discrete conscious control depends on the predictions that are made about the consciousness of the network, which are discussed in Chapter 7.

5.2.2 Modelling

To increase the system's rating on the OMC scale, the network was designed to be as biologically inspired as possible, but it was not intended that it should be an accurate model of particular brain areas. It was decided to construct the network from spiking neurons because they are more biologically realistic than rate based models and there is a growing body of evidence to suggest that the timing of individual spikes is an important part of the neural code (Maas and Bishop 1999). The high temporal resolution of spiking neural networks also makes them a promising method for motor control and some methods of simulating spiking neural networks are more efficient than rate-based models. For example, with Delorme and Thorpe's (2003) event-based approach, each neuron is only updated when it receives a spike, whereas a traditional rate-based simulation has to update each neuron's state at each time step. Although this advantage is lost when the network has a high average firing rate or connectivity,[3] event-based modelling has a strong performance advantage on spiking networks with low activity levels or low to medium connectivity.

The Spike Response Model (Gerstner and Kistler 2002, Marian 2003) was chosen for the neurons because it is a well established phenomenological model that can be efficiently implemented in an event-based manner. Although the Spike Response Model does not include spontaneous neural activity, many of the models that do include this feature, such as Izhikevich

---

[2] A model of conscious will would have required a reactive layer that could initiate the conscious decisions in response to an environmental trigger.

[3] For example, a synchronous simulation with a time step of 1 ms updates each neuron 1000 times per simulated second. The same update rate occurs in event-based modeling when each neuron is connected to 1000 neurons firing at 1 Hz in simulated time.

(2003), are difficult to implement using event-based simulation.[4] With spiking neural networks the association between two stimuli (Hebb 1949) is commonly learnt using a spike time dependent plasticity (STDP) learning algorithm, which reinforces the weight when the spike arrives before the firing of the neuron and decreases the weight when the spike arrives after the neuron has fired. In earlier work I experimented with the ReSuMe STDP algorithm (Ponulak and Kasiński 2006) and used it to learn the association between the activity of a teacher neuron and basic shapes, such as crosses and squares – see Gamez et al. (2006a). However, the artificial need for a teacher neuron led me to select Brader et. al's (2006) version of STDP learning for the final network, which combines the standard STDP rule with a model of the calcium concentration to improve the long term stability of the learnt information. Full details about the neuron model and learning are given in section 5.3.2 and 5.3.3.

5.2.3 Network Size

The main constraint on the network's size was the potential performance of the simulator. Both Krichmar et al. (2005) and Shanahan (2006) have demonstrated that networks of the order of 100,000 neurons could be simulated on current equipment, and so this was set as the upper limit on the size of the system. A second constraint on the network's size was the visual input and motor output resolution. In an earlier version of the network, 128 x 128 neuron layers were used to encode the red and blue visual information and 50 neurons were used to encode the length of each muscle. This led to high simulation times that were caused by the large number of connections to and from the input and output layers - particularly from the inhibitory layer. Since high sensory and motor resolution was not a requirement of this project, the red and blue visual input resolutions were reduced to 64 x 64 and 5 neurons were used to encode the length of each muscle.

[4] SpikeStream can run in a synchronous mode, and so it would be possible to experiment with Izhikevich's model in future work.

Another constraint on the network's size was the average number of connections per neuron. In real biological networks cortical neurons have up to 10,000 connections (Binzegger et. al. 2004), but since this system was only aiming at biologically inspired functionality, rather than precise brain modelling, a much more manageable average of 40 connections per neuron was used instead.[5]

A final potential constraint on the network's size was the amount of processing that was required to analyze it for information integration, which can take a great deal of computing power on networks greater than 50 neurons (see Chapter 7). In this thesis, the functionality of the network was given higher priority than the analysis, but in the future this constraint would be worth considering when designing networks that need to be analyzed using computationally intense algorithms.

Given all of these constraints, the final network was constructed with 17,544 neurons and 698,625 connections, which were found to deliver the required functionality with reasonable performance using the SpikeStream simulator that was developed for the project.

5.2.4 Simulator

The size of the network and the choice of neuron model substantially constrained the choice of simulator. To begin with, it was decided not to use simulators, such as NEURON, GENESIS and NCS,[6] which work with complex dendritic trees and would not have been efficient on the point neurons that were selected for this network. Rate-based simulators, such as Topographica,[7] were not suitable for spiking neural networks and I decided against using NRM[8] because I wanted to

---

5 Although the average connectivity is low, it varies widely between different neuron groups: neurons in Eye Pan and Eye Tilt connect to an average of 6 neurons; neurons in Inhibition connect to almost 9000 neurons.

6 NEURON simulator: http://www.neuron.yale.edu/neuron/; GENESIS simulator: http://www.genesis-sim.org /GENESIS/; NCS simulator: http://brain.cse.unr.edu/ncsDocs/.

7 Topographica Neural Simulator: http://topographica.org/Home/index.html.

8 This used to be called Magnus. More information about NRM is available at Barry Dunmall's website: http://www.iis.ee.ic.ac.uk/eagle/barry_dunmall.htm.

use a more biologically inspired approach in this project. Whilst NEST did work with spiking point neurons and had an impressive performance (Diesmann and Gewaltig 2002), the lack of a graphical interface and the fact that it was designed to simulate a fixed period of time led me to reject it for this project. Other unsuitable spiking simulators included the Amygdala library and Mvaspike, which lack graphical interfaces and were not designed for robotic use, and the Spiking Neural Simulator developed by Smith, which can simulate a spiking network for a fixed period of time, but lacks many important features.[9]

The two most promising simulators were SpikeNET, created by Delorme and Thorpe (2003), and SpikeSNNS (Marian 2003). Although I was initially impressed by Delorme and Thorpe's claims about the ability of SpikeNET to efficiently model large networks, there were a number of major limitations in the free version – for example, no delay, a single spike per neuron during each simulation run and the lack of a graphical interface – that would have necessitated major revisions of the software. SpikeSNNS overcame some of these limitations, but since it was based around a single event queue, it would have been difficult to distribute the processing over multiple machines and the SNNS interface is somewhat outdated and difficult to use. All of the simulators that I looked at suffered from the limitation that they were not designed to work with external devices, such as SIMNOS, and they were generally designed to simulate fixed periods of time.

Since a major revision of an existing simulator would have taken a substantial amount of effort and potentially left little of the original code intact, it was decided to create a new simulator that met my requirements and could be more easily extended as these requirements changed. The SpikeStream simulator that was developed for this project is described in Chapter 6.

---

[9] Amygdala simulator: http://amygdala.sourceforge.net/; Mvaspike simulator: http://www-sop.inria.fr/odyssee/softwares/mvaspike/; Spiking Neural Simulator: http://www.cs.stir.ac.uk/~lss/spikes/snn/index.html.

5.3 Network Details

5.3.1 Introduction

This section explains how the network was modelled and gives details about the construction and function of the different layers. This network is a biologically inspired model of aspects of the brain's processing, not a biologically accurate copy, and so the names given to individual layers, such as "Motor Cortex", are only intended to indicate that the layers' functions were inspired by particular brain areas.

5.3.2 Neuron and Synapse Model

The neuron model for these experiments is based on the Spike Response Model (Gerstner and Kistler 2002, Marian 2003), which has three components: a leaky integrate and fire of the weights of the incoming spikes, an absolute refractory period in which the neuron ignores incoming spikes, and a relative refractory period in which it is harder for incoming spikes to push the neuron beyond its threshold potential. The resting potential of the neuron is zero and when it exceeds the threshold the neuron is fired and the contributions from previous spikes are reset to zero. There is no external driving current and the voltage V_i at time t for a neuron i that last fired at $\hat{t}$ is given by:

$$V_i(t) = \sum_j \sum_f w_{ij} e^{-\frac{(t-t_j^{(f)})}{\tau_m}} - e^{n-(t-\hat{t}_i)^m} \mathrm{H}'(t-\hat{t}_i)\,, \tag{5.1}$$

where ω_{ij} is the synaptic weight between i and j, τ_m is the membrane time constant, f is the last firing time of neuron j, m and n are parameters controlling the relative refractory period, and H' is given by:

$$H'(t-\hat{t}_i) = \begin{cases} \infty, & \text{if } 0 \le (t-\hat{t}_i) < \rho \\ 1, & \text{otherwise} \end{cases}. \tag{5.2}$$

for an absolute refractory period ρ. To facilitate the learning algorithm set out in Section 5.3.3, the neuron model also contains a variable c that represents the calcium concentration at time t. Each time the neuron fires, this calcium concentration is increased by C_S and it decays over time according to Equation 5.3, where C_D is the calcium decay constant.

$$c(t) = \sum_i C_S e^{-\frac{t-\hat{t}_i}{C_D}} \tag{5.3}$$

The thresholds given in Table 5.3 were adjusted in each neuron group until the network produced the desired behaviour. The values for the other neuron parameters were based on (Marian 2003) and Brader et al. (2006) and are given in Table 5.1. The synapse model is very basic, with each synapse class passing its weight to the neuron when it receives a spike.

Parameter	**Value**
C_S	1
C_D	60
P	1 ms
τ_m	1
M	0.8
N	3
Minimum postsynaptic potential	-5

Table 5.1. Parameters common to all neurons

5.3.3 Learning

Learning in this network was carried out using Brader et. al's (2006) spike time dependent learning algorithm. In Brader et. al.'s model the internal state of the synapse is represented by

X(t) and the efficacy of the synapse is determined by whether *X(t)* is above a threshold. In my model, the state of the synapse is represented by a weight variable, *w*, which is the amount by which the post-synaptic membrane potential is increased when the neuron fires. When a spike is received at time t_{pre}, this variable *w* is changed according to equations 5.4 and 5.5:

$$w \rightarrow w + a \quad if \quad V(t_{pre}) > \theta_V \quad and \quad \theta_{up}^l < c(t_{pre}) < \theta_{up}^h \tag{5.4}$$

$$w \rightarrow w - b \quad if \quad V(t_{pre}) \leq \theta_V \quad and \quad \theta_{down}^l < c(t_{pre}) < \theta_{down}^h \tag{5.5}$$

where *a* and *b* are jump sizes, θ_V is a voltage threshold, *c(t)* is the calcium concentration at time *t*, and θ_{up}^l, θ_{up}^h, θ_{down}^l and θ_{down}^h are thresholds on the calcium variable. In the absence of a pre-synaptic spike or if the two conditions in equations 5.4 and 5.5 are not satisfied, the weight changes at the rate given by equations 5.6 and 5.7:

$$\frac{dw}{dt} = \alpha \quad if \quad w > \theta_w \tag{5.6}$$

$$\frac{dw}{dt} = -\beta \quad if \quad w \leq \theta_w \tag{5.7}$$

where α and β are positive constants and θ_w is a threshold. If *w* drops below 0 or exceeds 1, then it is held at these boundary values. The parameters that were used for training the network are given in Table 5.2. These parameters were initially set using Brader et. al ‘s (2006) values and then fine tuned until the network successfully learnt the association between motor output and visual input, as outlined in Section 5.4.

Parameter	**Value**
θ_{up}^{l}	4
θ_{up}^{h}	120
θ_{down}^{l}	0
θ_{down}^{h}	4
θ_{V}	0.4
a	0.01
b	0.01
θ_{w}	0.7
α	0.00001
β	0.00001

Table 5.2. Synapse parameters used during training

5.3.4 Experimental Setup

The network was created in SpikeStream (see Chapter 6 and Appendix 1) and connected to the eye of the SIMNOS virtual robot using the synchronized TCP interface described in sections 6.4 and A1.9.2. Spikes were sent from the network to set the pan and tilt of SIMNOS's eye, and when a spike containing red or blue visual information was received from SIMNOS, the value of 0.8 was added to the voltage of the neuron that corresponded to the location of the red or blue data in the visual field.

To set up the environment of SIMNOS, a red and blue cube were created in Blender[10] and loaded into the SIMNOS environment using the Collada format.[11] The head and body of SIMNOS were then put into kinematic mode, which enabled them to be placed in an absolute position and made them unresponsive to spikes from the network, and the eye was moved in

[10] Blender 3D animation software: www.blender.org.

[11] COLLADA format: www.collada.org.

front of the red and blue cubes so that it could only view one cube at a time - see figures 5.1 and 5.2.

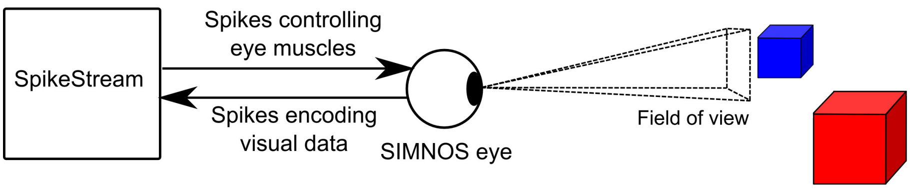

Figure 5.1. Experimental setup with the eye of SIMNOS in front of red and blue cubes

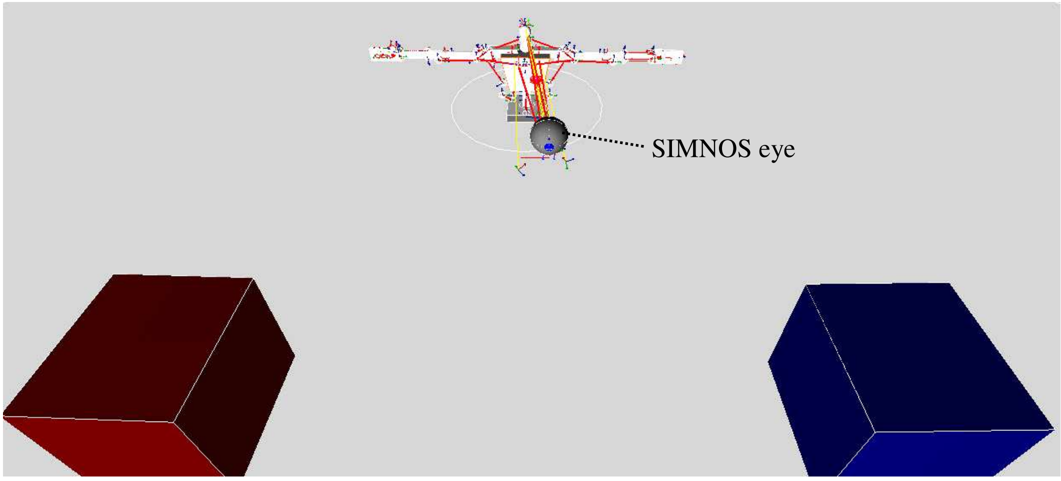

Figure 5.2. Screenshot of SIMNOS in front of the red and blue cubes

5.3.5 Architecture

The network is organized into ten layers whose overall purpose is to direct SIMNOS's eye towards 'positive' red features of its environment and away from 'negative' blue objects. To carry out this task it includes an 'emotion' layer that responds differently to red and blue stimuli and neurons that learn the association between motor actions and visual input. These neurons are used to 'imagine' different eye movements and select the ones that are predicted to result in a positive visual stimulus – in other words a planning process is carried out that changes the part of the world that is looked at by the system.

An illustration of the connections between the layers is given in Figure 5.3, and Figure 5.4 shows a view of the network in SpikeStream. The parameters for the layers are given in

Table 5.3, the details about the connections between the layers can be found in Table 5.4 and a SpikeStream file for this network is included in the Supporting Materials. The next two sections highlight some of the key functions of the network and describe the design and functionality of the individual layers in more detail.

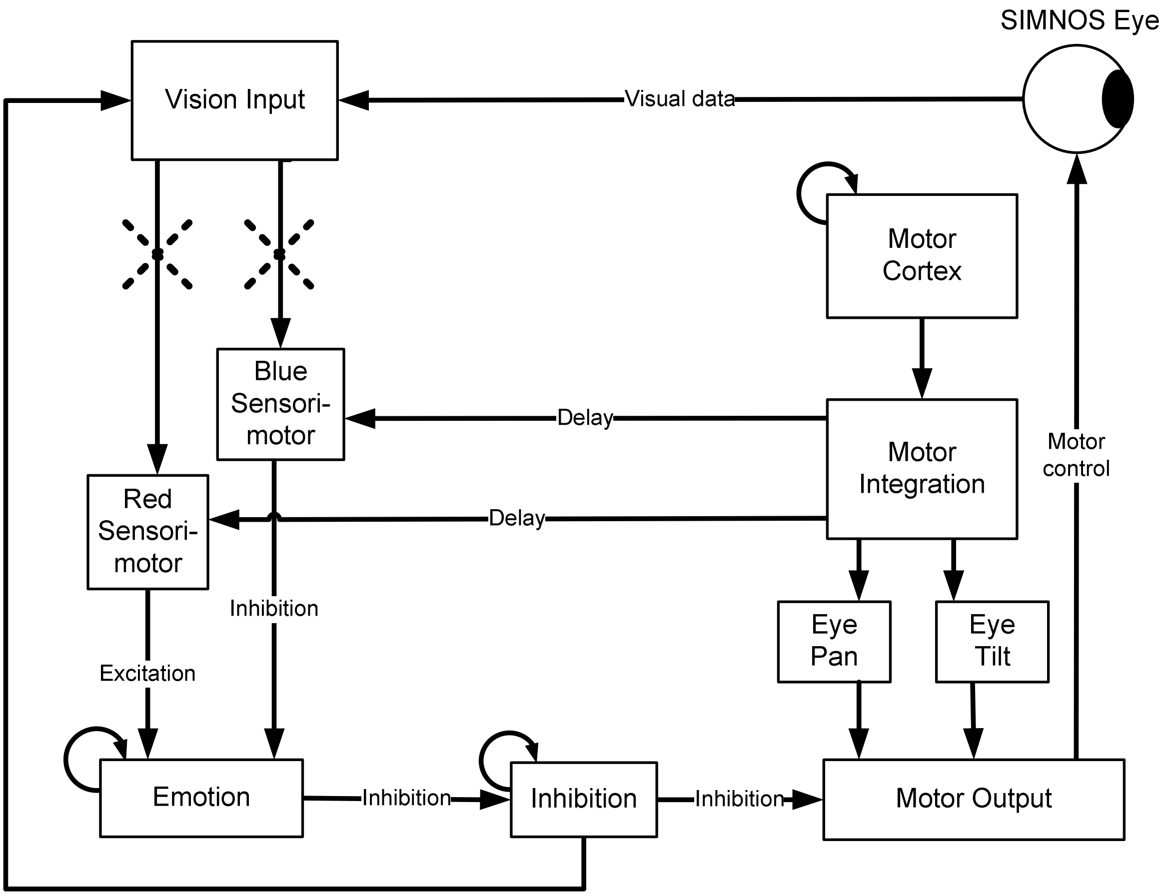

Figure 5.3. Neural network with SIMNOS eye. Arrows indicate connections within layers, between layers or between the neural network and SIMNOS. The connections marked with dotted crosses were disabled for the imagination test in Section 5.5.2.

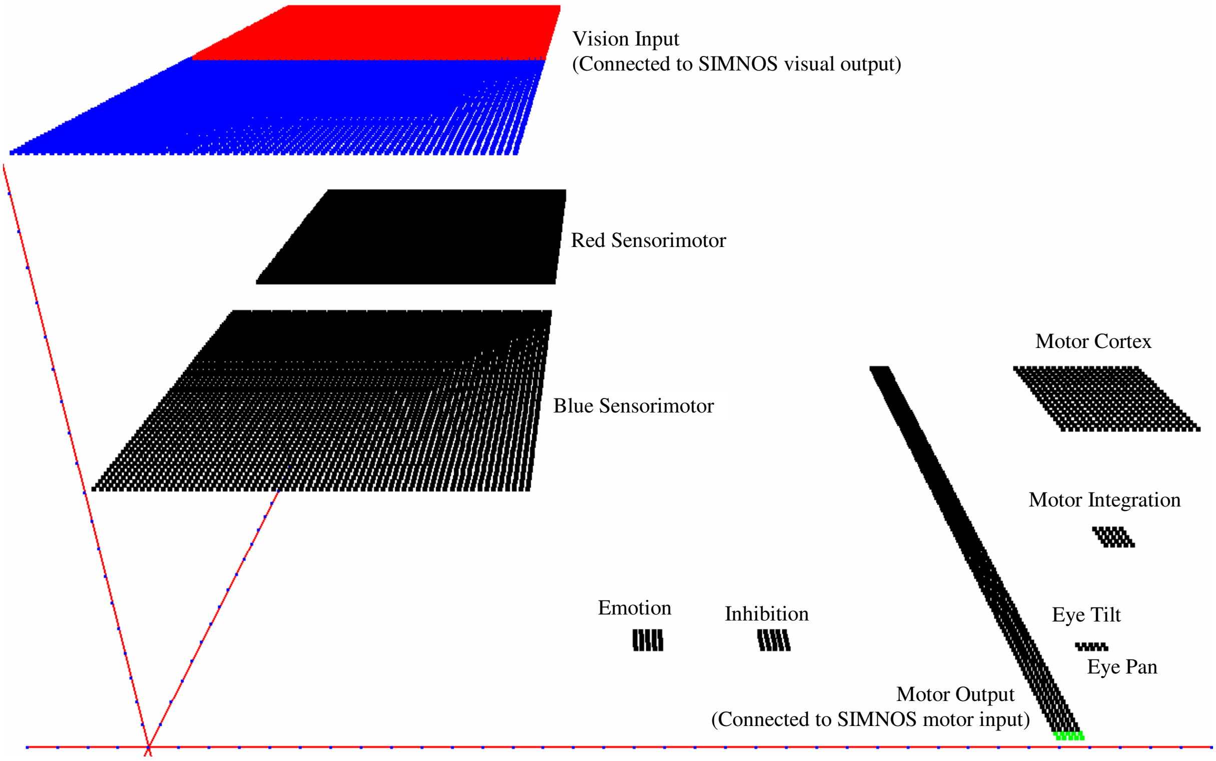

Figure 5.4. The network in SpikeStream. The red and blue sensitive parts of Vision Input are highlighted in red and blue. The neurons in Motor Output that set the pan and tilt of SIMNOS's eye are highlighted in green.

	Area	**Size**	**Threshold**	**Noise**	**Device**
1	Vision Input	64 × 128	0.5	-	SIMNOS vision[12] weight 0.8
2	Red Sensorimotor	64 × 64	0.8	-	-
3	Blue Sensorimotor	64 × 64	0.8	-	-
4	Emotion	5 × 5	2	-	-
5	Inhibition	5 × 5	0.1	20% weight 1.0	-
6	Motor Cortex	20 × 20	1.5	20% weight 0.6	-
7	Motor Integration	5 × 5	0.65	-	-
8	Eye Pan	5 × 1	0.7	-	-
9	Eye Tilt	5 × 1	0.7	-	-
10	Motor Output	5 × 135	0.1	-	SIMNOS muscles

Table 5.3. Layer parameters

[12] Spikes from SIMNOS change the voltage of the corresponding neurons in Vision Input with a weight of 0.8.

Projection	**Arbor**	**Connection Probability**	**Weight**	**Delay**
Vision Input→Red Sensorimotor	D	1.0	1.0	0
Vision Input→Blue Sensorimotor	D	1.0	1.0	0
Red Sensorimotor →Emotion	U	0.5	0.5	0
Blue Sensorimotor →Emotion	U	0.5	-0.5	0-5
Emotion→Emotion	ECIS 5/ 10	0.5 / 0.5	0.8 ± 0.2 / -0.8 ± 0.2	0-5
Emotion→Inhibition	U	1.0	-1.0	0-5
Inhibition→Inhibition	ECIS 5/10	0.5/ 0.5	0.8 ± 0.2 / -0.8 ± 0.2	0-5
Inhibition→Vision Input	U	1.0	-1.0	0
Inhibition→Motor Output	U	1.0`	-1.0	0
Motor Cortex→Motor Cortex	ECIS 1.7/ 30	0.99/ 0.99	0.8/ -0.8	2
Motor Cortex→Motor Integration	T	1.0	0.5	0
Motor Integration→Red Sensorimotor	U	1.0	0.5	11
Motor Integration→Blue Sensorimotor	U	1.0	0.5	11
Motor Integration→Eye Pan	T	1.0	1.0	0
Motor Integration→Eye Tilt	T	1.0	1.0	0
Eye Pan→Motor Output	D	1.0	1.0	0
Eye Tilt→Motor Output	D	1.0	1.0	0

Table 5.4. Connection parameters. Unstructured connections (U) connect at random to the neurons in the other layer with the specified connection probability. Topographic connections (T) preserve the topology and use many to one or one to many connections when the layers are larger or smaller than one other. Excitatory centre inhibitory surround (ECIS) connections have excitatory connections to the neurons within the excitatory radius and inhibitory connections between the excitatory and the inhibitory radius - for example, ECIS 5/50 has excitatory connections to neurons within 5 units of each neuron and inhibitory connections to neurons from 5 to 50 units away. A device connection (D) connects a layer to part of an input or output layer that is connected to an external device, such as a robot or camera. So, for example, Red Sensorimotor connects to the part of Vision Input that receives red visual input from SIMNOS.

5.3.6 Network Functions

Input and output

The spikes containing visual data from SIMNOS's eye are routed so that red and blue visual data is passed to different halves of Vision Input as shown in Figure 5.4. The Motor Output layer is a complete map of all the 'muscles' of SIMNOS and the activity in each of the five neuron rows is sent as spikes across the network to SIMNOS, where it sets the length of the virtual muscles. The only rows in Motor Output that were active in these experiments were the ones controlling eye pan and tilt, which are highlighted in green in Figure 5.4.

Self-sustaining activity

Three of the layers – Motor Cortex, Emotion and Inhibition – have recurrent positive connections, which enable them to sustain their activity in the absence of spikes from other layers. A random selection of 20% of the neurons in Inhibition and Motor Cortex are injected with noise at each time step by adding 1.0 or 0.6 to their voltage (see Table 5.3), and this enables them to develop their self sustaining activity in the absence of spikes from other layers. The neurons in Emotion can only develop their self-sustaining activity when they receive spikes from Red Sensorimotor.

Selection of motor output

The position of SIMNOS's eye is selected by the activity in Motor Cortex, which has long range inhibitory connections that limit its self-sustaining activity to a single small cluster of 2-4 neurons. The activity in Motor Cortex is passed by topographical connections to one or two neurons in Motor Integration, which is a complete map of all the possible combinations of eye pan and eye tilt. The activity in Motor Integration is then topographically transmitted through Eye Pan and Eye Tilt to Motor Output and passed by SpikeStream over the Ethernet to SIMNOS, where it is used to set the lengths of the eye pan and eye tilt muscles.

Learning

A delay along the connection between Motor Integration and Red Sensorimotor ensures that spikes from a motor pattern that points the eye at a red stimulus arrive at Red Sensorimotor at the same time as spikes containing red visual data. When these spikes arrive together, the STDP learning algorithm increases the weights of the connections between Motor Integration and the active neurons in Red Sensorimotor, and decreases the weights of the connections between Motor Integration and inactive neurons in Red Sensorimotor. The same applies to the connections between Motor Integration and Blue Sensorimotor, except that the association between motor patterns and blue visual data is learnt. Prior to the learning, repeated activation of Motor Integration neurons within a short period of time fires all of the neurons in Red/ Blue Sensorimotor. Once the learning is complete, spikes from Motor Integration only fire the neurons in Red/ Blue Sensorimotor that correspond to the pattern that is predicted to occur when the eye is moved to that position.

Online and offline modes

Inhibition has a large number of negative connections to Vision Input and Motor Output, which prevent the neurons in Vision Input and Motor Output from firing when Inhibition is active. I have called this the ‘imagination’ or *offline* mode because in this situation the network is isolated from its environment and no spikes from SIMNOS are processed by the network or sent by the network to SIMNOS. When the neurons in Inhibition are not firing, the neurons in Vision Input are stimulated by spikes from SIMNOS and the neurons in Motor Output send spikes to SIMNOS to set the position of the eye, and this will be referred to as the *online* mode of the network. The switch between online and offline modes is controlled by Emotion, which is connected to Inhibition with negative weights, so when Emotion is active, Inhibition is inactive and vice versa. Emotion enters a state of self-sustaining activity when it receives spikes with positive weights from Red Sensorimotor, and its state of self-sustaining activity ceases when it receives spikes with negative weights from Blue Sensorimotor.

5.3.7 Overview of Individual Layers

Motor Cortex

This layer was designed to select a motor pattern at random and sustain it for a period of time. These motor patterns are used to set the lengths of the eye pan and eye tilt muscles in SIMNOS, and in 'imagination' mode these patterns need to be sustained to overcome the delays between the selection of an appropriate motor pattern, the 'imagination' of that pattern and the removal of inhibition that allows the pattern to be executed. Short range excitatory and long range inhibitory connections in Motor Cortex encourage a small patch of neurons to fire at each point in time and this active cluster of firing neurons occasionally changes because a random selection of 20% of the neurons in Motor Cortex are injected with noise at each time step by adding 0.6 to their voltage. The topographic connections between Motor Cortex and Motor Integration enable the active cluster of neurons in Motor Cortex to send spikes to just one or two neurons in Motor Integration.

Motor Integration

Each neuron in this layer represents a different combination of eye pan and eye tilt. Activity in Motor Cortex stimulates one or two neurons in Motor Integration and this activity is transformed through Eye Pan and Eye Tilt into a pattern of motor activity that is sent to SIMNOS's eye. The activity in Motor Integration is also sent along delayed connections to Red Sensorimotor and Blue Sensorimotor, where it is used to learn the relationship between motor output and red and blue visual input.

Eye Pan

This layer connects topographically to Motor Output, where it stimulates the row corresponding to eye pan in SIMNOS. Eye Pan receives topographic connections from Motor Integration.

Eye Tilt

This layer connects topographically to Motor Output, where it stimulates the row corresponding to eye tilt in SIMNOS. Eye Tilt receives topographic connections from Motor Integration.

Motor Output

This layer is a complete map of all the 'muscles' of SIMNOS and the activity in each of the five neuron rows in this layer sets the length of one of SIMNOS's virtual muscles. In these experiments, only eye pan and eye tilt were used and the rest of the muscles were locked up by setting them into kinematic mode. The neurons highlighted in green in Figure 5.4 are topographical connected to Eye Pan and Eye Tilt, and strong inhibitory connections between Inhibition and Motor Output ensure that there is only activity in Motor Output (and motor output from the network) when Inhibition is inactive.

Vision Input

This layer is connected to SIMNOS's visual output so that each spike from SIMNOS stimulates the appropriate neuron in this layer with a weight of 0.8, with one half responding to red visual input from SIMNOS and the other half responding to blue visual input. When Inhibition is inactive the spikes from SIMNOS fire the neurons in Vision Input; when Inhibition is active, a large negative potential is injected into the neurons in Vision Input, which prevents this layer from responding to visual information.

Red Sensorimotor and Blue Sensorimotor

Red Sensorimotor and Blue Sensorimotor are topographically connected to the red and blue sensitive parts of Vision Input. Positive connections between Red Sensorimotor and Emotion cause Emotion to develop self-sustaining activity when Red Sensorimotor is active. Negative connections between Blue Sensorimotor and Emotion inhibit the self-sustaining activity in Emotion. Red Sensorimotor and Blue Sensorimotor receive delayed copies of the motor output from Motor Integration and the synapses on these connections use Brader et al.'s (2006) STDP rule to learn the association between motor output and visual input.

Emotion

This layer plays an analogous role to emotions in the human brain, although in a greatly simplified form.[13] Recurrent positive connections within Emotion enable it to sustain its activity once it has been stimulated: spikes from Red Sensorimotor set Emotion into a self-sustaining state; spikes from Blue Sensorimotor inhibit it. Emotion inhibits Inhibition, so when Emotion is active, Inhibition is inactive, and vice versa.

Inhibition

A random selection of 20% of the neurons in Inhibition are injected with noise at each time step by adding 1.0 to their voltage, which enables Inhibition to develop its self sustaining activity in the absence of spikes from other layers. When Inhibition is active it inhibits Motor Output and Vision Input and puts the system into its offline 'imagination' mode. Negative connections from Emotion cause the neurons in Inhibition to be inactive when Emotion is active.

[13] To be a true emotion this layer would have to receive connections from the robot's body. Since this is not the case, the activity in this layer is more like the 'as if' loop described by Damasio (1995). The limitations of this emotion model are discussed in more detail in Section 7.6.1.

5.4 Experimental Procedure

The first part of the experiments was a training phase in which the network learnt the association between motor output and visual input. Since the 'imagination' mode interfered with this training, it was disabled by blocking the connections from Inhibition. During the training phase spontaneous activity in Motor Cortex changed the position of SIMNOS's eye, copies of the motor signals were sent from Motor Integration to Red/ Blue Sensorimotor, and the synapse classes on these connections used Brader et. al.'s (2006) rule to learn the association between motor output and red and blue visual input. By monitoring the changes in the weights over time it was empirically determined that a training period of 50,000 time steps (or 50 seconds of simulated time at 1 ms time step resolution) enabled the network to learn the association between motor output and visual input for most combinations of eye pan and eye tilt.

Once the network had been trained, Inhibition was reconnected and the network was observed and tested. For both the training and testing a time step resolution of 1 ms was found to offer a good balance between the accuracy and speed of the simulation.

5.5 Operation of the Network

5.5.1 Overview

During the training phase, the network spontaneously generated eye movements to different parts of its visual field and learnt the association between an eye movement and a visual stimulus. After training, the network was fully connected up and Motor Cortex moved SIMNOS's eye around at random until a blue object appeared in its visual field. This switched the network into its offline 'imagination' mode, in which it generated motor patterns and 'imagined' the red or blue visual input that was associated with these potential eye movements. This process continued until it 'imagined' a red visual stimulus that positively stimulated Emotion. This removed the inhibition, and SIMNOS's eye was moved to look at the red stimulus. Videos of the network in operation are available in the Supporting Materials.

5.5.2 Imagination Test

This was a rough qualitative evaluation of the associations that the network had learnt between motor output and visual input. In this test Red Sensorimotor and Blue Sensorimotor were disconnected from Vision Input (the dotted crosses in Figure 5.3), so that they only received input from Motor Integration, and Vision Input continued to receive visual input from SIMNOS’s eye, which remained under the control of Motor Cortex. If the system had learnt the association between motor output and visual input, then the activity in Red/ Blue Sensorimotor, caused by Motor Integration, should match the activity in Vision Input, which was driven by real visual input.

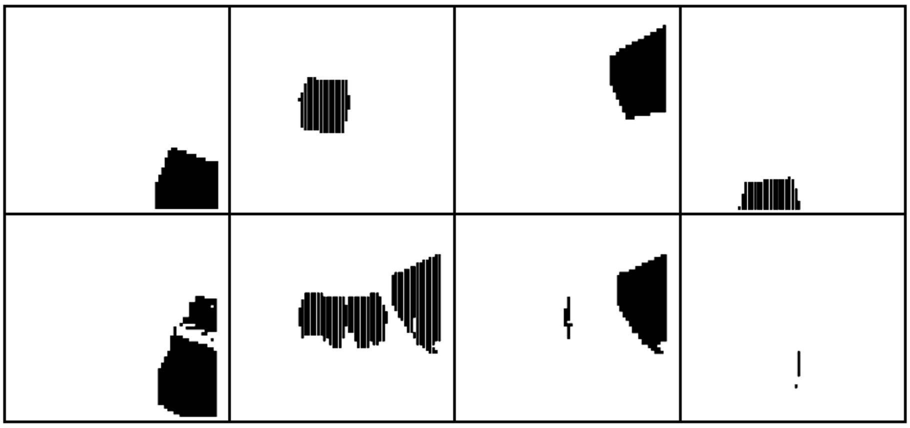

Figure 5.5. Examples of the contrast between real visual input (top row) and imagined visual input (bottom row)

During the imagination test visual inspection of Vision Input, Red Sensorimotor and Blue Sensorimotor showed that the ‘imagined’ visual inputs were reasonably close to the real visual inputs, but often a larger area of Red Sensorimotor or Blue Sensorimotor was activated than would have been caused by visual input alone. It also happened that several different patterns were activated simultaneously in Red Sensorimotor and Blue Sensorimotor, which was probably caused by oscillation in Motor Integration between two different positions during training.

Furthermore, Red/.Blue Sensorimotor sometimes contained areas of active neurons when the real stimulus was just off screen, which was again probably due to multiple neurons in Motor Integration being simultaneously active during training. Some examples of the contrast between imagined and real visual input are given in Figure 5.5.

5.5.3 Behaviour Test

This network was designed to use its 'imagination' to reduce its exposure to 'negative' blue visual input and a test was run to establish whether it achieved this objective. In this test, the untrained network was run for 100,000 time steps (100 seconds of simulated time) with Emotion and Inhibition disabled, and the activity in the red and blue sensitive parts of Vision Input was recorded. The 'imagination' circuit was then trained and connected, and the measurements were repeated. This procedure was carried out five times with the SIMNOS environment set up from scratch on each run to reduce potential biases towards the red or blue cubes that might have been introduced by the manual positioning of the robot's eye.

The results of the behaviour test are presented in Figure 5.6 and Figure 5.7, which show that the activity in the blue visual area was substantially reduced when the 'conscious' circuits were in operation. This suggests that if the 'negative' blue stimulus was capable of damaging the system, then the cognitive mechanisms associated with consciousness could play a useful role in the life of the organism.[14]

[14] These cognitive mechanisms might have to be combined with a reflex that moves the eye away from the damaging stimulus whilst the imagination is taking place – see Section 5.7 for a discussion of this point. It is also worth noting that the imagination did not have to be particularly accurate to carry out this function.

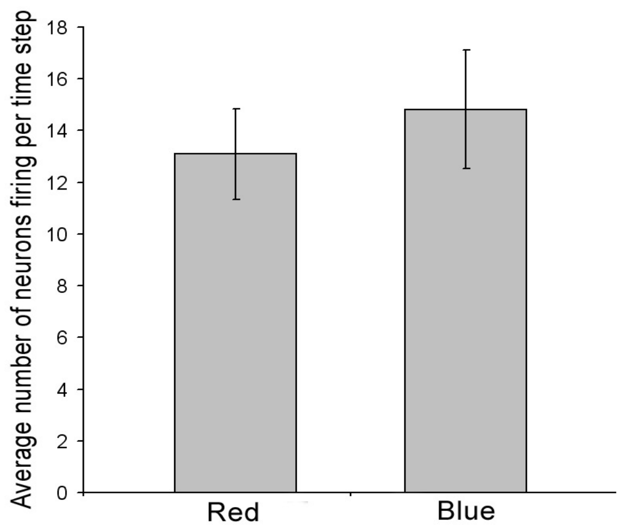

Figure 5.6. Average number of neurons firing per time step in the red and blue sensitive parts of Vision Input when the cognitive mechanisms associated with consciousness were disabled [15]

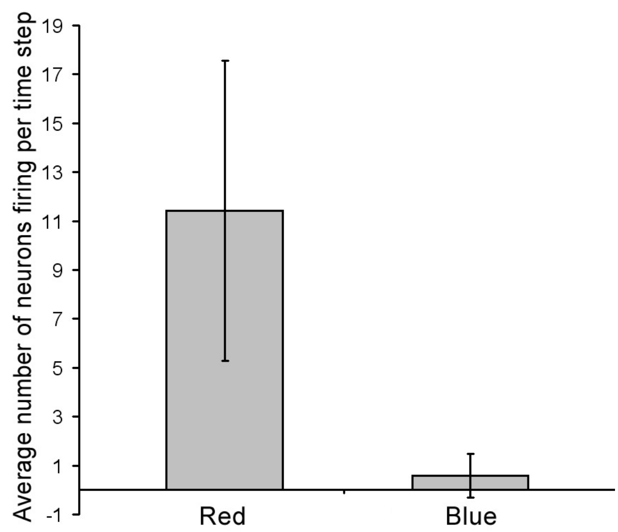

Figure 5.7. Average number of neurons firing per time step in the red and blue sensitive parts of Vision Input when the cognitive mechanisms associated with consciousness were enabled [16]

5.6 Previous Work

Previous work on neural networks in machine consciousness – for example, Aleksander (2005), Shanahan (2006, 2008) and Cotterill (2003) – has already been covered in Chapter 3, and so this section focuses on research on simulated neural networks that is not explicitly related to machine consciousness. The simulation of neural networks is an extremely large topic and only a few of the most significant or relevant projects are covered here.

A number of experiments have been carried out by Krichmar and Edelman (2006) using robots controlled by simulated neural networks that are closely based on the brain. For example, Krichmar et. al (2005) developed a system that learnt to navigate to a hidden platform from an arbitrary starting position using only visual landmarks and self-movement cues. The robotic part

[15] The error bars are +/- 2 standard deviations.

[16] The error bars are +/- 2 standard deviations.

of this system was a wheeled robot base equipped with a camera and odometry and infra red sensors. The simulated nervous system had 50 neural areas, including a visual system, head direction system, hippocampus, basal forebrain, value or reward system, and an action selection system. The complete network had 90,000 neuronal units, which were modelled using a rate-based model, and 1.4 million connections. The neural network was simulated on a Beowulf cluster of 12 1.4GHz computers that communicated wirelessly with the robot. Using innate behaviours for exploration, obstacle avoidance and platform detection, the robot moved around its environment until it detected the hidden platform and the run was terminated. After a number of runs, the robot learnt to locate the platform and could travel directly to it from multiple starting points. Krichmar et al.'s (2005) analysis of the neural system showed that it had developed place specific units, similar to those identified in rodents, that were sensitive to a combination of visual and self-movement cues, and Krichmar et al. were able to trace functional pathways within the nervous system using their backtracing method.[17]

Larger scale simulations of biological neural networks have been created by the Blue Brain project (Markram 2006), which is attempting to produce a biologically accurate model of a single cortical column, consisting of around 10,000 neurons interconnected with 30 million synapses. This project is simulating the neurons in this column at a high level of detail using Neocortical Simulator 7 and NEURON 8, which are running on an IBM Blue Gene supercomputer containing 8192 processors and 2 TB of RAM – a total of 22 x 1012 teraflops processing power. The first simulation of the rat cortical column was carried out in 2006 and it is currently running at about two orders of magnitude slower than real time. The main objective of this project is to reproduce the behaviour of in vitro rat tissue, and so the stimulation is not connected to sensory input and it has not been used to control the behaviour of a real or virtual robot.

[17] This backtracing method is described in more detail in Section 4.3.4.

A larger and less detailed neural model has been developed by Ananthanarayanan and Modha (2007), who simulated a network with 55 million single-compartment spiking neurons and 442 billion synapses. This model was run on a 32,768 processor Blue Gene/L with 8TB memory, and one second of simulation time could be processed in 9 seconds per Hertz of average neuronal firing rate. This system was created to demonstrate the possibility of large scale cortical simulations and the neurons were connected probabilistically together without any attempt at biological plausibility.

There has also been some substantial work on the development of large scale neural models in silicon. For example, Boahen is developing the Neurogrid system, which will consist of 1 million silicon neurons and 6 billion synaptic connections (Silver et. al., 2007). This uses an analogue circuit to emulate a real neuron's ion-channels and the spikes between neurons are routed digitally. Another significant hardware project is SpiNNaker, which is attempting to simulate a billion spiking neurons in real time using a large array of power-efficient processors (Furber et. al., 2006).[18]

Other related work on the simulation of neural networks is that by Grand (2003), who used a network of more than 100,000 neurons to control a pongid robot, and Izhikevich et. al. (2004) have carried out simulations of 100,000 neurons and 8.5 million synapses to study the self organization of spiking neurons into neuron groups. More recently Izhikevich claims to have created a much larger scale simulation of 100 billion neurons and 10^{15} synapses. According to his website, it took 50 days on a Beowulf cluster of 27 processors to calculate a second of simulation time for this network.[19]

[18] See http://intranet.cs.man.ac.uk/apt/projects/SpiNNaker/.

[19] This research is discussed on his website: http://vesicle.nsi.edu/users/izhikevich/human_brain_simulation, but I have not been able to find any publications on it.

5.7 Discussion and Future Work

A first problem with this network is that its visual processing is very basic and its actions are limited to the panning and tilting of a single eye. In the future more sophisticated visual processing could be added to the network along the lines of that developed by Krichmar et al. (2005), and it could be designed to plan and execute more complex actions.

A second limitation is that the motor patterns are selected randomly in the offline mode and then a decision is made about whether to execute them or not. Even with just 25 eye pan/ tilt combinations it often took more than 5,000 time steps (5 simulated seconds) to find a motor combination that was associated with a red object and switched the network out of its 'imagination' mode. Future versions of this network might be able to address this problem by using a learnt association between emotions and colours and between colours and motor actions to prime the motor choices - when the network 'imagined' the colour that positively stimulated its emotion system, an appropriate motor pattern could be selected automatically.

A third problem with the network is that it is not clear whether it would perform any better than a simple reflex that moved the eye away from the 'negative' stimulus to a random part of the visual field. Such a reflex would reduce the activity in the blue input layer in the same way as the imagination circuit, but with a great deal less complexity. However, the imagination circuit would have an advantage when there were a large number of blue objects in the visual field, which would increase the probability that a random motor action would select another blue object. In this case, imagination should perform better since it would only execute actions directed towards red objects.

When blue visual input is inhibited, the eye continues to point at the blue stimulus, and so the organism's retina would burn out if it was actually directed at a painful visual stimulus, such as the sun. To solve this problem, some kind of reflex would be needed to move the eye away whilst the imagination was taking place. However, if blue is simply an unattractive or depressing visual stimulus – a second dead and decaying SIMNOS, for example - then the inhibition of visual input is a successful strategy.

This network has all of the components needed for the model of discrete conscious control that was set out in Section 2.7.4, since it can imagine different scenarios, evaluate its emotional response to them and immediately execute a selected action. The question whether this network is actually conscious as it selects and executes its actions is addressed in Section 7.9.7. This network cannot model conscious will (see Section 2.7.5) because it does not have a reactive layer that would enable its actions to be executed automatically in response to environmental stimuli. When this network is deliberating, the eye is static, whereas a system implementing conscious will would continue to react to the world whilst it was planning future actions, with these reactions being a mixture of past decisions and hardwired behaviours. In future work a reactive layer could be added to the network that would have its parameters set by the 'imagination' circuit in a similar way to the model developed by Shanahan (2006).

The current system has only been implemented on the virtual SIMNOS robot, but some people, such as Thompson and Varela (2001), believe that real physical embodiment may be necessary for consciousness. The realistic physical nature of the SIMNOS simulation should address many of these worries and in the future the neural network could be used to control the CRONOS robot when the software interface is ready.

5.8 Conclusions

This chapter has presented a spiking neural network that uses some of the cognitive mechanisms that have been associated with consciousness to control the eye movements of the SIMNOS virtual robot. This network enables SIMNOS to avoid 'negative' stimuli and it is also an example of a neural system that can learn the association between sensory input and motor output and use this knowledge to plan actions.

The next chapter outlines the SpikeStream simulator that was developed to model this network and Chapter 7 describes how this network was analysed for phenomenal states using Aleksander's, Metzinger's and Tononi's theories of consciousness.