The Development and Analysis of Conscious Machines

David Gamez

A thesis submitted for the degree of PhD

Department of Computing and Electronic Systems, University of Essex, 2008

ABSTRACT

This PhD was carried out as part of the CRONOS project and one of its main achievements was the development of a method for predicting and describing the conscious states of artificial systems. This could help machine consciousness to become more scientific and it could also be used to make predictions about the consciousness of biological systems.

To demonstrate this methodology, a spiking neural network was developed to control the eye movements of the SIMNOS virtual robot. This network learns the association between sensory input and motor output and uses this knowledge to 'imagine' the consequences of different eye movements and avoid stimuli that negatively affect its 'emotions'. This network exhibits a limited form of conscious behaviour, has some of the cognitive characteristics associated with consciousness, and Tononi's, Aleksander's and Metzinger's theories of consciousness were used to make detailed predictions about its phenomenal states.

The spiking neural network was modelled using the SpikeStream simulator, which was developed as part of this PhD and can simulate up to 100,000 neurons. SpikeStream has good performance, a comprehensive graphical interface and it can send and receive spikes to and from real and virtual robots across a network.

This thesis makes a number of theoretical contributions to the study of natural and artificial consciousness, which include a discussion of the relationship between the phenomenal and the physical, a distinction between type I and type II potential correlates of consciousness, and an analysis of conscious will and conscious control. The different areas of machine consciousness research are also classified and some of the challenges facing work in this area are covered in detail.

A flash, a mantling, and the ferment rises,
Thus, in this moment, hope materializes,
A mighty project may at first seem mad,
But now we laugh, the ways of chance forseeing:
A thinker then, in mind’s deep wonder clad,
May give at last a thinking brain its being.
…
Now chimes the glass, a note of sweetest strength,
It clouds, it clears, my utmost hope it proves,
For there my longing eyes behold at length
A dapper form, that lives and breathes and moves.
My mannikin! What can the world ask more?
The mystery is brought to light of day.
Now comes the whisper we are waiting for:
He forms his speech, has clear-cut words to say.

Goethe, *Faust*, *Part Two*, p. 101.

CONTENTS

ACKNOWLEDGEMENTS

As always, I owe a warm debt of gratitude to my parents, Alejandro and Penny Gamez, who gave me a great deal of support and encouragement and helped me financially during the dark months that it took to complete this thesis.

The interface between SIMNOS and SpikeStream was developed in collaboration with Richard Newcombe, who designed the spike conversion methods. I am very grateful to Richard for help and support throughout this PhD, especially with the many difficulties that I had with the analysis of the network for information integration and representational mental states. I would also like to acknowledge Renzo De Nardi and Hugo Gravato Marques in the Machine Consciousness lab at Essex and everyone at the 2006 Telluride workshop.

This thesis would not have happened without my supervisor, Owen Holland, who had the vision to create the excellent CRONOS project and I really appreciated his feedback, support and detailed and useful advice. I would also like to thank Igor Aleksander and Sam Steel who took the time to read and examine this thesis and put forward many insightful comments and suggestions during the viva. This work was funded by a grant from the EPSRC Adventure Fund (GR/S47946/01).

1. Introduction

> The interdisciplinary project of consciousness research, now experiencing such an impressive renaissance with the turn of the century, faces two fundamental problems. First, there is yet no single, unified and *paradigmatic* theory of consciousness in existence which could serve as an object for constructive criticism and as a backdrop against which new attempts could be formulated. Consciousness research is still in a preparadigmatic stage. Second, there is no systematic and comprehensive catalogue of *explananda*. Although philosophers have done considerable work on the *analysanda,* the interdisciplinary community has nothing remotely resembling an agenda for research. We do not as yet have a precisely formulated list of explanatory targets which could be used in the construction of systematic research programs.
>
> (Metzinger 2003, pp. 116-7)

1.1 Overview

This PhD was carried out as part of Owen Holland's and Tom Troscianko's EPSRC-funded CRONOS project to build a conscious robot (GR/S47946/01), which took place at the Department of Computing and Electronic Systems, University of Essex and at the Department of Experimental Psychology, University of Bristol. One of the main contributions at Essex was the development of the CRONOS and SIMNOS robots, which are described in Section 1.2. This thesis documents my contribution to this project, which includes the construction of a spiking neural network to control SIMNOS's eye movements and the development of a new way of analyzing systems for consciousness that was used to make predictions about this network's phenomenal states. A summary of the thesis given in Section 1.3 and Section 1.4 describes the supplementary data files and other supporting materials.

1.2 The CRONOS Project[1]

1.2.1 Introduction

CRONOS is one of the few large projects that has been explicitly funded to work on machine consciousness. One of the motivations behind this project was the belief that embodied human-like systems carrying out tasks in the real world (or a reasonably realistic copy) are the best starting point for understanding how our brains operate and how consciousness emerges in the brain. Guided by this approach, Owen Holland, Rob Knight and Richard Newcombe developed CRONOS, a hardware robot closely based on the human musculoskeletal system (see Figure 1.1), and a soft real time physics-based simulation of this robot in its environment, known as SIMNOS (see Figure 1.2). More information about the CRONOS project is available at www.cronosproject.net.

1.2.2 CRONOS Robot

Most humanoid robots are essentially conventional robots that fit within the morphological envelope of a human. However, robots that can help us to understand human cognition and action might need to have a much higher level of biological inspiration, which imitates biological structures and functions as well as the human form. The CRONOS robot was developed to address this challenge and it has a body based on the human musculoskeletal system and senses that are as biologically inspired as possible.[2] This level of biological realism is important to machine consciousness because a more biological body is more likely to develop a human style of consciousness, and it also provides more realistic training data for biologically inspired neural networks.

[1] All of the work described in this section was carried out by Owen Holland, Rob Knight and Richard Newcombe at the University of Essex.

[2] Holland and Knight (2006) have proposed the term “anthropomimetic” as a label for humanoid robots that attempt to copy the physical structure of a human.

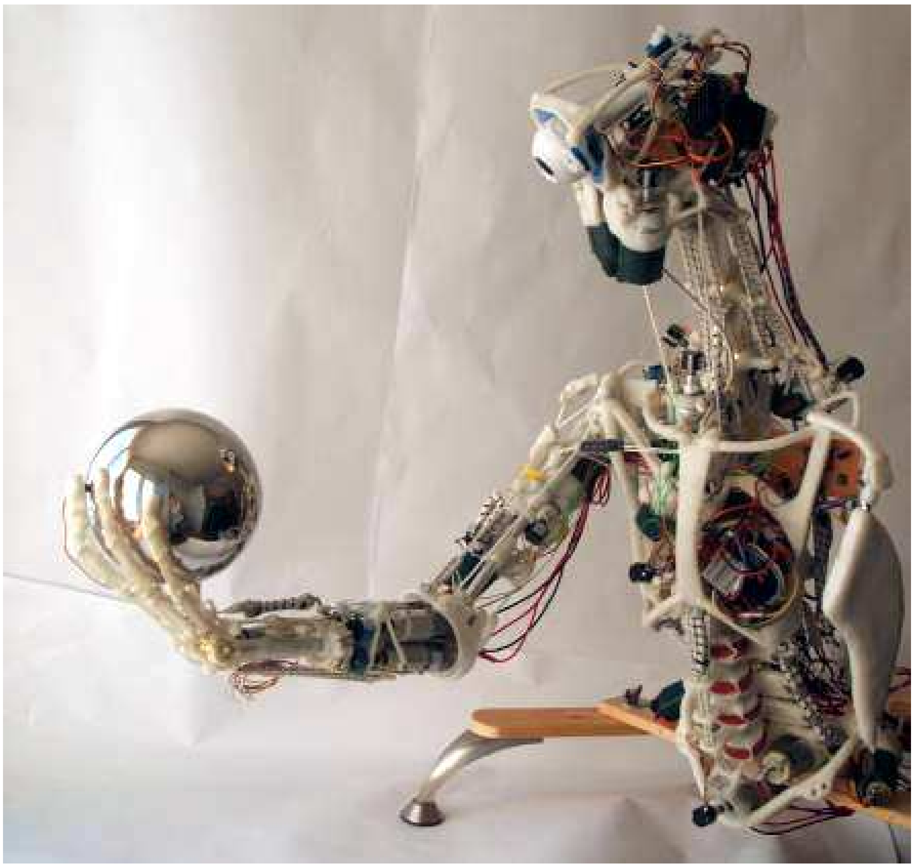


Figure 1.1. CRONOS Robot

To create the skeleton of CRONOS, the human skeleton was copied as accurately as possible at life size.[3] The bones were constructed from a new type of thermoplastic known in the UK as Polymorph and in the US as Friendly Plastic, which softens and fuses at 60 degrees and can be freely hand moulded until it resets at 30 degrees. This enabled bone like elements to be created and fitted together by hand and other materials can be embedded, such as a metal sphere mounted on a rod to make a ball and socket joint. The muscles of CRONOS were constructed using a motor and marine grade shock cord terminated at each end by 3mm braided Dyneema kite line. This cord was wound around the motor spindle, so that the rotation of the motor increased or decreased the tension in the elastic shock cord, mimicking the contraction and relaxation of a biological muscle.[4]

[3] To compensate for anticipated difficulties with the fine manual manipulation of grasped objects, the neck vertebrae were extended to allow a greater range of head movements during visual inspection of such objects.

[4] Videos of CRONOS are available at www.cronosproject.net.

This combination of bone-like elements and partially elastic ‘muscles’ gives the body of CRONOS a multi-degree-of-freedom structure that responds as a whole and transmits force and movement well beyond the point of contact. For example, when the arm is pushed down, the elbow flexes, the complex shoulder moves and the spine bends and twists. The disturbances due to the robot’s own movements are also propagated through the structure, producing what Holland et al. (2007) have called 'passive coordination'. Since different trajectories and finishing points are obtained with different loadings, any controllers that are developed for this robot will need feedforward compensation to anticipate and predictively cancel the effects of the load for any movement. This is interesting from the point of view of consciousness because feedforward control depends on the possession of forward models and the use of such models by the nervous system has been advanced by Grush (2004) and Cruse (1999) as one of the key factors underpinning consciousness.

CRONOS differs from humans in having only a single central eye. This approach was chosen because of the enormous simplification of visual processing that it brings about and it is justified by the observation that 2-4% of humans do not perform stereo fusion and their performance on other visual tasks is still within the normal range (Julesz, 1971). The high resolution colour camera has a 90 degree field of view and it can perform rapid saccades under the control of three servo motors that rotate, pan and tilt the eye. Each of the muscle motors has a potentiometer and touch sensors are being developed for the hands and stretch receptors for the tendons to give more realistic proprioceptive information An interface is also being developed that will allow CRONOS to stream its sensory data as spikes over the network and receive muscle commands as spikes from the network. This will be similar to the spike streaming between SIMNOS and SpikeStream that is described in Section 1.2.5.

1.2.3 SIMNOS Virtual Robot

SIMNOS is a model of CRONOS that was created to test Holland's (2007) theories about the link between consciousness and internal modeling and to accelerate the development of controllers for CRONOS. This model was created using physics-based rigid body modeling, implemented in Ageia PhysX,[5] in which the components of objects and surfaces are described in 3D by mathematical expressions in terms of their underlying physics, and the expressions are solved using extremely fast and efficient numerical techniques. This reliance on physics guarantees accuracy at all scales, and the efficiency of the computations allows thousands of complex objects interacting in real time to be modeled on a standard personal computer.

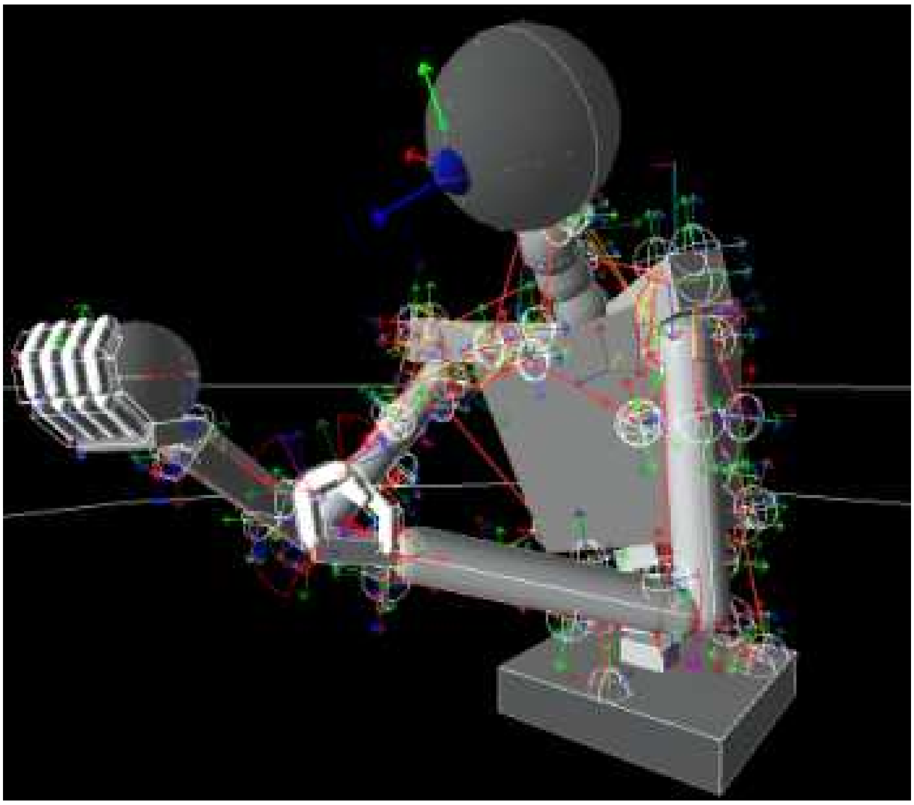


Figure 1.2. SIMNOS virtual robot. The red lines are the virtual muscles; the outlines of spheres with arrows are the joints. The length of the virtual muscles and the angles of the joints are encoded into spikes and sent to the SpikeStream neural simulator.

[5] Ageia PhysX: http://www.ageia.com/developers/api.html.

The individual components of CRONOS are modeled in SIMNOS using appropriate sizes and masses, but the shapes were simplified where possible – for example, detailed bone shapes were approximated by cylinders with the same dimensions and distribution of mass. The elastic actuators were created using springs of appropriate lengths connected to matching points on the modeled skeleton, and sufficient damping was added to produce the slight degree of under-damping seen on CRONOS. The virtual robot's environment contains rigid bodies that are either simple geometrical shapes or triangular meshes and new objects can be created using 3D simulation packages, such as Maya or Blender, and imported into SIMNOS using the COLLADA format.[6] In the future it will be possible to add cloth- and fluid-based objects to SIMNOS's virtual environment.

The SIMNOS model of CRONOS is convincing at the physical level and displays a similar quality of movement. The fluidity, load sharing and passive coordination in CRONOS are also seen in SIMNOS, which presents comparable control problems.

1.2.4 SIMNOS Performance

A simple virtual world was developed to test SIMNOS's computation time. The simulated scene was started with random parameters for every muscle and all of the sensory and motor data was calculated to ensure the maximum computational load. The simulator was then run for 3000 time steps and at each step a newly created sphere was dropped onto the surface of the table where the robot was fixed. As the objects fell onto the table and floor they interacted with the robot, the environment and each other.

The computation times for this virtual world were recorded for a number of different time step values and plotted in Figure 1.3. These results show that soft real time simulation of the robot in an environment with 300 objects, with full scene rendering for user output, is

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

possible for time step values greater than 1/50th second and this performance will improve substantially as more cheap physics processing hardware for the PhysX engine becomes available.

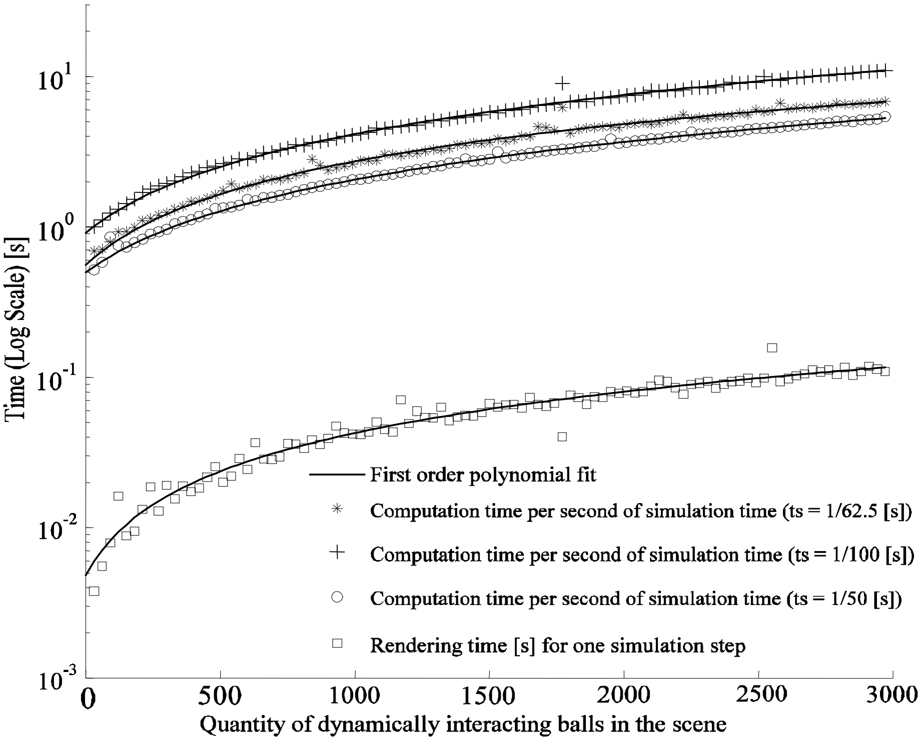


Figure 1.3. Performance of SIMNOS

1.2.5 Sensory Data and Spike Encoding

The sensory data generated by SIMNOS includes 25 Euler angle values that monitor the relative rotations of thorax-pelvis and head-thorax and every degree of freedom in each hand, arm and shoulder complex.[7] The robot is equipped with 41 muscles and the current length is available for each muscle, together with the control values that were issued to it: a total of 164 values per time step.[8] The virtual robot is configurable to have either one or two eyes, which provide a continuous visual stream from the virtual environment.

[7] These angles are indicated in Figure 1.2 by the positions of the arrows within the outlined spheres.

[8] The muscles are shown as red lines in Figure 1.2.

To interact with the SpikeStream simulator Richard Newcombe developed a simple model to convert the real valued sensor data into a time varying spike train. Current theories of neural coding fall under either rate or temporal encoding schemes (Bialek et al. 1991, Shadlen and Newsome 1994) and this model utilizes a hybrid, spatially distributed, average rate encoding method**.** This spans the range of a real valued variable with a set of *N* broadly tuned 'receptors'. Each receptor, $n \in \{0..N\}$, is modelled with a normalised Gaussian with mean μ_n and variance σ_n^2 (1.1) (1.2), with the values of μ_n computed to equally divide the variable range with a receptor mean at the minimum and maximum of the range.

$$\mu_n = \frac{n}{N-1} \tag{1.1}$$

$$\sigma_n = \frac{1}{3(N-1)} \tag{1.2}$$

Given a real valued variable at time *t,* ($v_t \in [0..1]$), the spiking output of each receptor ($r_n \in \{0,1\}$) is computed based on the probability, $\mathrm{p}(n, v_t)$ of that receptor firing (equations 1.3 and 1.4), where *c* is a scaling factor used to control the maximum firing rate of a receptor and *rand* is drawn from a uniform distribution. The variance of a receptor is chosen to ensure that $\mathrm{p}(n, v_t) = 1$ when $\mu_n = v_t$, with all other receptors having negligible probability.

$$r_n(t) = \begin{cases} 1 \;\; \mathit{iff} \;\; \mathrm{p}(n, v_t) > k \\ 0 \end{cases} \tag{1.3}$$

$$\mathrm{p}(n, v_t) = e^{\frac{-(v_t - \mu_n)^2}{2\sigma_n^2}} \qquad k = c \cdot \mathit{rand}\,[0,1] \tag{1.4}$$

Given *N* spike trains the conversion back to a real value is performed by taking the average normalised firing rate $fr_n(t)$ for the current time step *t* within a given window of *w* previous simulation steps for each of the *N* spiking signals. The approximated real value at this time, $\tilde{v}_n(t)$, is then the sum of the receptor means weighted by the firing probability (equations 1.5 and 1.6).

$$fr_n(t) = \frac{\sum_{i=t}^{t-w} r_n(i)}{w} \tag{1.5}$$

$$\tilde{v}_n(t) = \sum_{n \in N} \mu_n \cdot fr_n(t) \tag{1.6}$$

Such a spatially distributed rate encoding provides resilience to noisy signals, with the benefit that increased resolution in spiking representation can be achieved without altering the rate of firing of an individual neuron. The same sensory data scheme is being applied to the CRONOS hardware robot so that the two systems will have the same interface. Unfortunately this was not completed in time for this thesis, and so only the SIMNOS robot was used in this PhD.

1.3 Thesis Summary

The overall aim of this PhD was to develop a neural network to control the SIMNOS robot (Chapter 5) and to analyze this network for consciousness (Chapter 7). This analysis required a consistent interpretation of consciousness (Chapter 2) and I had to develop a way of analyzing systems for phenomenal states (Chapter 4). A new spiking neural simulator called SpikeStream was developed to model the neural network (Chapter 6) and a considerable amount of background research was also carried out (Chapter 3).

Chapter 2: Consciousness

Machine consciousness is a relatively new research area that is highly cross-disciplinary and takes elements from computer science, philosophy, neuroscience and experimental psychology. Although this thesis is primarily about computer science, a significant obstacle to progress in research on consciousness is the large number of conflicting theories and there is a general lack of consensus about what is meant by consciousness. These problems are highlighted by Metzinger (2003), who claims that consciousness is in a pre-paradigmatic state,[9] and Coward and Sun (2007, p. 947) argue that our understanding of consciousness suffers from "considerable meta-theoretical confusion". In order to develop a systematic way of analyzing machines for consciousness, it was necessary to carry out some philosophical work to clarify the concept of consciousness and outline a framework for its scientific study, which is used in the analysis work in later chapters. This examination of consciousness uses the neurophenomenological approach put forward by Varela (1996), in which phenomenological methods are used to shed light on work in the physical sciences.

The first part of this chapter develops an interpretation of consciousness that distinguishes between the phenomenal world of our experiences and the physical world described by science. This distinction between the phenomenal and the physical leads to a definition of consciousness that is compared with other definitions and linked to a correlates-based approach, which is becoming increasingly popular through research on the neural correlates of consciousness. The correlates of consciousness are examined in more detail and two types of potential correlates of consciousness (PCCs) are identified. Type I PCCs are behaviour-neutral, which makes it makes it impossible to prove their connection with consciousness empirically, whereas type II PCCs do affect behaviour and it is possible to establish if they are systematically linked to conscious states. This type I/ II distinction is used to classify different

[9] See the quotation at the beginning of this chapter.

theories of consciousness and it plays an important role in the approach to synthetic phenomenology that is developed in Chapter 4.

The last part of Chapter 2 sets out three theories of consciousness, which are used to analyse the network in Chapter 7, and it concludes with a discussion of the relationship between consciousness and action.

Chapter 3: Machine Consciousness

This chapter provides a context for the work in this thesis by summarizing some of the previous research on machine consciousness. To provide a more systematic interpretation of this work, the research on machine consciousness is divided into four different areas:

- *MC1*. Machines with the external behaviour associated with consciousness.
- *MC2*. Machines with the cognitive characteristics associated with consciousness.
- *MC3*. Machines with an architecture that is claimed to be a cause or correlate of human consciousness.
- *MC4*. Phenomenally conscious machines.

In the first part of Chapter 3 this classification is used to examine the relationship between machine consciousness and other disciplines, and to interpret some of the criticisms that have been raised against work in this area. The central part of this chapter covers some of previous work on machine consciousness and the final part discusses the ethical issues surrounding this type of research and looks at the potential benefits.

Chapter 4: Synthetic Phenomenology

A systematic method for measuring the consciousness of an artificial system is essential if researchers want to prove that they have created a conscious machine, and feedback about the consciousness of a system is also useful if one wants to extend or enhance its consciousness.

Whilst it is reasonably easy to see how the behaviour, cognitive characteristics and architecture associated with consciousness can be identified using standard techniques, it is much harder to see how phenomenal consciousness can be measured. With humans, the presence of phenomenal states is generally established through verbal communication, but most of the systems that have been developed as part of research on machine consciousness are only capable of non-verbal behaviours. Since relatively little work had been carried out in this area, new techniques had to be created to identify and describe the phenomenal states of the artificial neural network that was developed by this thesis.

The correlates of consciousness can only be used to decide whether a machine is conscious when scientific experiments have identified a list of the necessary and sufficient correlates, and Chapter 2 argues that type I potential correlates of consciousness cannot be empirically separated out. To address this problem, Chapter 4 outlines an ordinal machine consciousness (OMC) scale that models the contribution that a system's type I correlates make to our belief that it is capable of phenomenal states. When a system's type I correlates match those of the human brain, it is given an OMC rating of one; when we believe that a system is unlikely to be conscious, its OMC rating is close to zero.

The second half of Chapter 4 develops a new and systematic way of describing artificial conscious states. This approach formulates precise definitions of mental states and representational mental states, and suggests how representational mental states can be identified by exposing the system to different test stimuli and measuring its response. Problems with the description of representational mental states in human language led to the use of a markup language for the final phenomenological description, which makes less assumptions about the common ground between the consciousness of humans and artificial systems.

Chapter 5: Neural Network

Chapter 5 describes a spiking neural network with 17,544 neurons and 698,625 connections that controls the eye movements of the SIMNOS virtual robot and uses its 'imagination' and 'emotions' to decide whether it looks at a red or blue cube. This network was designed to give SIMNOS the external behaviour associated with consciousness (MC1) using the cognitive characteristics associated with consciousness (MC2), and it was analyzed for phenomenal states (MC4) using the methodology set out in Chapter 4. As part of the testing of the network some visualizations of its 'imagination' were recorded and its behaviour was quantitatively measured.

Chapter 6: SpikeStream

Although it might have been easier to use an existing simulator to create the network described in Chapter 5, none of the available simulators were suitable, either because of the scale of the network, the type of modelling, or because they would have been difficult to modify to interface with the SIMNOS virtual robot. This led me to develop a new spiking neural simulator called SpikeStream, which is based on Delorme and Thorpe's (2003) SpikeNET architecture. Chapter 6 gives a brief high level summary of the architecture, features and performance of SpikeStream; much more detailed information is available in the SpikeStream manual, which is included as Appendix 1 in this thesis.

Chapter 7: Analysis

The final chapter documents the work that was done to establish whether the neural network created by this project was predicted to be conscious according to Tononi's (2004), Aleksander's (2005) and Metzinger's (2003) theories. The first stage in this process was the identification of representational mental states in the network. This was done by injecting noise into the input and output layers and mutual information was used to identify the parts of the system that responded to information in the input or output layers. The network was then examined for information integration (Tononi and Sporns 2003), which was used to analyze the network according to Tononi's theory of consciousness, to support the analysis for Metzinger's theory of consciousness and to evaluate the integration between neurons in the network. This analysis for information integration was a considerable challenge because of a factorial relationship between the size of the network and the number of calculations that had to be carried out, and a number of different approximation strategies were used to complete the analysis in a reasonable time. The final part of the analysis was the generation of files containing a description of the predicted phenomenology of the network at each time step, and the predicted distribution of consciousness was plotted for Tononi's, Aleksander's and Metzinger's theories. These results showed that different parts of the network were predicted to be conscious according to the three theories, but it was not possible to predict the absolute amount of consciousness because the measures had not been calibrated on normal waking human subjects.

Appendix 1: SpikeStream

Appendix 1 is a manual documenting the installation and features of SpikeStream. This manual was included with the SpikeStream 0.1 release.

Appendix 2: Network Analyzer

This appendix summarizes the main features of the Network Analyzer software, which was developed for the analysis part of this thesis.

Appendix 3: Seed and Group Analyses

This appendix presents the detailed results from the seed and group information integration analyses.

Appendix 4: Gamez Publications Related to Machine Consciousness

A list of publications by David Gamez that are connected to the work in this thesis.

1.4 Supporting Materials

This thesis is accompanied by a number of supplementary materials, which are available on CD and at www.davidgamez.eu/mc-thesis/. These include:

- A copy of the thesis in Adobe's .pdf format.
- A website implementing the OMC scale.
- Java code for the OMC scale.
- SpikeStream code.
- SpikeStream source code documentation.
- Network Analyzer code.
- Results from the representational mental states analysis in XML format.
- Results from the validation on Tononi and Sporns' test networks in XML format.
- Results from the information integration analysis in XML format.
- The neural network developed by the project in SpikeStream format.
- Recordings of the network in SpikeStream format.
- Videos of the network.
- The final XML description of the synthetic phenomenology of the network.

These supporting materials are constructed as a website, which can be launched by double clicking the index.html file at the root directory of the CD.

2. CONSCIOUSNESS

2.1 Introduction

This chapter outlines a theory of consciousness that will be used throughout this thesis. A general failure to analyse what we mean by the physical world, perception and consciousness has been a central source of confusion in consciousness research and the first part of this chapter spends a substantial amount of time clarifying basic concepts about the phenomenal and the physical and linking them to the sources of our knowledge about consciousness. The philosophical approach that is used for this work is influenced by neurophenomenology (Varela 1996, Thompson et al. 2005), which combines cognitive science and neuroscience with a systematic analysis of human experience influenced by Continental philosophy – for example, the work of Husserl (1960). Although this approach might occasionally sound naïve, it is a necessary first step if we want to get clearer about what can and cannot be scientifically established about consciousness. Some of this material is also covered in Gamez (2007c, pp. 25-87) and it maps onto Metzinger's (2000) distinction between phenomenal and theoretical knowledge.

The first section in this chapter is a phenomenological examination of the relationship between the phenomenal and the physical, which is used to develop a definition of consciousness in Section 2.3. This is compared with some of the previous definitions that have been put forward and Section 2.4 examines and rejects popular metaphysical theories about consciousness, such as dualism, epiphenomenalism and physicalism, in favour of a correlates-based approach, which is explored in Section 2.5. A close reading of the brain-chip replacement experiment is used to show that we will never be able to separate out some of the potential correlates of consciousness empirically, which leads to a distinction between type I and type II correlates of consciousness. Section 2.6 then covers the three type II theories of consciousness that have been selected to design and analyze a neural network in this thesis. The final part of this chapter develops a preliminary interpretation of the relationship between consciousness and action.

2.2 The Phenomenal and the Physical

> A person who grew up and lives in a certain limited environment has time and again encountered bodies of fairly constant size and shape, colour, taste, gravity and so on. Under the influence of his environment and the power of association he has become accustomed to find the same sensations combined in one place and moment. Through habit and instinct, he presupposes this constant conjunction which becomes an important condition of his biological welfare. The constant conjunctions crowded into one place and time that must have served for the idea of absolute constancy or substance are not the only ones. An impelled body begins to move, impels another and starts it moving; the contents of an inclined vessel flow out of it; a released stone falls; salt dissolves in water; a burning body sets another alight, heats metal until it glows and melts, and so on. Here too we meet constant conjunctions, except that there is more scope for spatio-temporal variation.
>
> Mach (1976, p. 203)

2.2.1 The Stream of Experience

Our theoretical studies and scientific experiments take place in a colourful moving noisy spatially and temporally extended stream of experience. This stream of experience is the most real thing that there is: everything that we do is carried out within it.[1]

Within waking life this stream of experience is highly structured. Some of the most characteristic structures are stable objects, which typically have a reasonably consistent set of properties that can be experienced on multiple occasions. For example, when I am examining a machine, I experience the front, turn it around to look at the back, and when I turn it around so that the front faces me again, I seem to experience the same set of sensations from the machine

[1] See Dennett (1992) and Blackmore (2002) for a criticism of this notion of the stream of experience.

as when I first looked at it. This stability of objects also extends over time: I speak about a *single* machine rusting because I can allow a subset of the machine's properties to change without thinking that a completely different machine has appeared in front of me. Whilst objects in waking life typically exhibit this kind of stability, objects in dreams or hallucinatory states are much less stable, and it is harder to return to the same view of an object or to perceive changes in a single object over time.

The stability of objects leads us to speak about their *persistence* when they are not under direct observation. Although I am not currently experiencing my motorbike, it is still out there in the garage and I can experience it again by going into the garage and taking off its cover. The difference between objects that we are currently perceiving and objects that are not currently being perceived by anyone is described by Lehar (2003) using his metaphor of a 'bubble' of perception that we 'carry around' with us, within which only a subset of the world's objects appear. Although objects appear *as* three-dimensional within this bubble of perception, I only experience part of them at any one time. From one position, I experience the outside of a cardboard box, but not the whole box, and I have to move relative to the box to experience more of its properties. Instead of simply saying that the box is there, I talk about *seeing* the box to indicate that I am *currently* experiencing the box, that the box is *within* my bubble of perception.

This interpretation of perception can be further analysed and broken down. For example, my visual perception is strongly linked to my eyes. In the stable world of waking life, the set of objects within my bubble of visual perception can be altered by covering my eyes or by damaging them in some way. The same is true of my ears and my bubble of auditory perception and my body and my bubble of somatic perception. In general, altering the sensory parts of my body alters the contents of my bubble of perception; it changes the subset that is 'extracted' from the totality of possible perceptions. This is a purely empirical observation and in a different world it could turn out that covering my big toe reduced the set of objects within my bubble of visual perception. However, in this world, repeated experiments have shown that it is the eyes that are important for this. An alternative interpretation would be that it is the world that is changing when I cover my eyes, and not my bubble of perception. However, when I turn my head I continue to see the same objects with my other eye, and so I attribute the change to my perception and not to the world itself.

The states of my bubble of perception are also strongly correlated with the state of my brain. When I hit my head, the waking world is overlaid with bright points of light, damaging parts of my brain reduces my bubble of perception in different ways, and my bubble of perception can be altered by injecting or ingesting chemicals that are circulated by my blood to my brain.[2] These can change the colours, sounds and sensations in my bubble of perception, and they can even destroy the stability of my waking experiences entirely and make them similar to a dream. This correlation between perceptual changes and the brain is not logically necessary in any way – for example, it might have turned out that hitting a ring on my finger produced bright points of light. However, in this world, the strong correlations between my bubble of perception and the states of my senses and brain suggest that without my senses and brain I would not have a bubble of perception at all.[3]

As I move around I come across other objects that look the same as me and have a similar brain and body. These objects behave in a similar way to myself and speak about other objects in a similar way. The verbal reports of these human objects suggest that for most of the time they perceive different parts of the world that is experienced by me. When the senses or brains of these other people are damaged or altered by chemicals, their verbal reports change in the same way that mine changed under similar circumstances. These changes have no effect on the objects within my own bubble of perception, which gives me further evidence for my belief

[2] Chemicals that do not reach my brain do not have any effect.

[3] The possession of senses and a brain might be necessary for a bubble of perception, but they are not sufficient because some states of my senses and brain, such as deep sleep, are not associated with perception at all.

that changes to my brain do not induce changes in other objects. Some people's bubbles of perception contain objects or properties of objects that are not perceived by anyone else. Under these circumstances it becomes a matter of debate and consensus about which objects and properties are artefacts of people's bubbles of perception.[4]

2.2.2 The Physical World

The stream of experience is structured in subtle ways that can only be identified through systematic investigations. These regularities are often explained by hypothesizing invisible *physical* entities that have effects on the stream of experience. As systematic measurements confirm the regularities, the physical theories gain acceptance and their hypothesized entities are believed to be part of the world, even though they do not directly appear within the stream of experience. To make this point clearer I will give a couple of examples.

A classic example of a physical theory is the atomic interpretation of matter, which claims that large scale changes in the stream of experience are caused by interactions between tiny bodies. By hypothesising that gases consist of a large number of moving molecules, Bernoulli (1738) developed the kinetic theory of gases, which describes how pressure is caused by the impact of molecules on the sides of a container and links heat to the kinetic energy of the molecules. Although molecules had not been observed when the theory was put forward, their existence became accepted over time because of the theory's good predictions. More recently we have developed ways of visualising individual molecules, atoms and particles – for example, the scanning tunnelling microscope and bubble chamber. These techniques use a more or less elaborate apparatus to construct representations within the stream of experience that are interpreted as the effects of these particles.

[4] Children, mystics and madmen all experience non-consensual objects within their bubbles of perception. See Gamez (2007c, pp. 145-193) for a detailed discussion.

A second example of a physical theory is Newton's interpretation of gravity. To make more accurate predictions about the movement of objects relative to the Earth, Newton hypothesized an invisible force that attracts remote bodies. The magnitude of this gravitational force is given by Newton's equations, which can be used to calculate the acceleration of objects towards the Earth and to make reasonably accurate predictions about the movement of planetary bodies. Newton's theory of gravity was very controversial when it was put forward and Newton himself had no idea how one body could exert a force on another over a distance: "I have not been able to discover the cause of those properties from the phenomena, and I frame no hypotheses" (quoted from Gjertsen (1986, p. 240)). Over time Newton's theory gained acceptance because of the accuracy of its predictions and people gradually came to believe that the physical world was permeated by an invisible gravitational force. More recently, general relativity's claims about the effect of matter on the curvature of four-dimensional spacetime are no easier to imagine, and these counterintuitive claims are only taken seriously because of their accurate predictions.[5]

Almost every aspect of the stream of experience has been re-interpreted by modern science as forces, particles or waves that affect the stream of experience when they are within a certain frequency range (sound and light), of a certain chemical composition (smell and taste) or when they collide with the human body (touch). These appearances do not *resemble* the original forces, particles or waves in any way – light does not look like a photon; sound does not sound like a wave. Our scientific models of physical reality enable accurate predictions to be made about the transformations of objects in the stream of experience, but the forces, particles and waves that constitute these models are defined mathematically and have to be indirectly measured from within the stream of experience using scientific apparatus.

[5] Newton also introduced a notion of mass that is different from what we experience as weight in the stream of experience. If a pre-Newtonian person could have travelled to different planets, then they would have probably said that they were gaining and losing weight, rather than preserving a constant mass that was attracted by different gravitational forces.

2.2.3 The Phenomenal World

> The representation of space in the brain does not always use space-in-the-brain to represent space, and the representation of time in the brain does not always use time-in-the-brain.
>
> Dennett (1992, p. 131)

When we first encountered the stream of experience, it was neither objective nor subjective: it was just what was there as the world. However, the development of the notion of a non-experiential physical world forces us to re-interpret this stream of experience as a *phenomenal* world that is different from the physical world. This phenomenal world is the same stream of experience that we started with, but reinterpreted as a *representation* of the non-sensory physical world.

Many people try to limit the phenomenal world to simple sense experiences, such as red, the smell of burnt plastic, and so on, and make the assumption that we directly perceive the spatial and temporal aspects of the physical world.[6] The problem with this position is that there are no scientific or philosophical arguments for *resemblance* between our experiences of space, time and movement and these qualities in the physical world. In fact just the opposite is suggested by interpretations of perception put forward by Metzinger (2003), Lehar (2003), Gamez (2007c), Dawkins(1998), Revonsuo (1995) and many others, who claim that the brain generates a simulation of the physical world, in which space, time and colour are *all* representations within a completely virtual environment.[7] Although our virtual representations might have analogues in the physical world, there is no reason to believe that they resemble the

[6] This old assumption goes back to Locke (1997), who distinguished between the primary qualities of figure, solidity, extension, motion-or-rest and number, which are something like direct perceptions of qualities of the physical world, and secondary qualities, such as colour or smell, which are artefacts produced by the effect of the primary qualities on the senses.

[7] This is also supported by Russell’s (1927) claim that physical matter is a source of events and not something that we are directly acquainted with. Kant’s (1996) *Critique of Pure Reason* is another version of this position.

physical world, which has a completely non-sensory nature.[8] This suggests that phenomenal experiences cannot be reduced to simple sensory qualia that are superimposed on a direct experience of physical reality. If the phenomenal world is interpreted using a theory of qualia (a highly debatable point – see Section 2.3.1), then *everything* is qualia, including experiential space, time, movement and size. Since there is no such thing as a physical *experience*, the phenomenal world is everything in the stream of experience, and the physical theories of particles, gravity, and so on, lead us to reinterpret this stream of experience in relation to an invisible physical world.[9]

2.2.4 The Physical and Phenomenal Brain

Within the picture that I have presented so far, regularities in the stream of experience are explained using scientific theories based on the physical world, and we would expect that scientific theories about consciousness would conform to this model and be based on the physical brain, and not on the brain as it appears in the stream of experience. Before these scientific explanations can be sought it is essential to get as clear as possible about the distinction between the physical and phenomenal brain, which will help with the discussion of the hard problem of consciousness in Section 2.4.5.[10]

[8] This does not amount to scepticism about the physical world because space in the brain is represented by our phenomenal image of space. It is just that we cannot imagine or picture to ourselves what real space is actually like. This is also different from instrumentalism and anti-realism because one can be completely realistic about scientific descriptions of forces, quarks, electrons, and so on, and yet claim that they can only be described in an abstract language, and not imagined by human beings using the virtual phenomenal model associated with the brain.

[9] A more detailed version of this argument can be found in Gamez (2007c, pp. 71-83).

[10] This focus on the brain is not affected by Clark and Chalmers' (1998) suggestion that many cognitive processes might be carried out in the environment. Whilst some of our cognitive processes and even beliefs may be external to our brains, Clark and Chalmers (1998) are careful to point out that both experiences and consciousness are likely to be determined by the processes inside our brains. Velmans' (1990) interpretation of projection theory is also consistent with a strong link between the brain and consciousness because he claims that consciousness is generated inside the brain and projected out of it into the environment. The only people I am aware of who question a strong link between the brain and consciousness are Thompson and Varela (2001), who criticize an exclusive focus on the *neural* correlates of consciousness and claim that "the processes crucial for consciousness cut across brain–body–world divisions, rather than being brain-bound neural events." (Thompson and Varela 2001, p. 418).

The physical brain is part of physical reality: it is completely non-phenomenal and has never directly appeared in the stream of experience. It consists of the physical entities that are deemed by physicists to constitute physical reality, such as quarks, wave-particles, forces, ten-dimensional superstrings and so on. The physical brain is also defined by other properties, such as spatial extension, mass and velocity, which can be defined mathematically and must be carefully distinguished from their phenomenal representations.

The phenomenal brain is the totality of our possible and actual phenomenal experiences of the brain, including its texture, colour, smell, shape, taste, sound and so on. The phenomenal brain also includes phenomenal measurements of the physical brain, such as the experience of looking at an fMRI scan, or taking a reading from a thermometer with its bulb inside the brain. We can remember our phenomenal experiences of the brain and imagine them when the brain is not physically present.

2.2.5 Concluding Remarks about the Phenomenal and the Physical

This interpretation of the phenomenal and physical gives equal importance to the phenomenal and physical worlds and suggests that it is too early to *assume* that the phenomenal world can be reduced to the physical world - although it is not impossible that this could be established by later work. This understanding of the phenomenal and the physical also fits in with Varela’s (1996, p. 347) claim that: “lived, first-hand experience is a proper *field of phenomena*, irreducible to anything else” and it has a lot in common with Flanagan’s (1992) constructive naturalism and Searle’s (1992) defence of the irreducibility of consciousness. How this starting point could be developed into a science of consciousness is discussed in detail in the rest of this chapter.

A second aspect of the phenomenal and the physical that is worth touching on at this stage is the ontological status of abstract properties, such as the volume of the brain or the number of red objects in my visual field. Whilst the volume of the brain is not a physical entity like a force or particle, it is also not part of my stream of experience in the same way as a yellow flower or the smell of myrrh. This problem extends to the ontological status of language and mathematics, which are also not straightforwardly phenomenal or physical entities. Since this question is not particularly relevant to this thesis, it will be set aside here and I will use abstract properties, mathematics and language to describe the phenomenal and the physical worlds without taking a position about their ontological status.

2.3 What is Consciousness?

The distinction between the phenomenal and the physical will now be used to set out a definition of consciousness that will be employed throughout this thesis. After some clarifications of this definition, it will be compared with some of the other interpretations of consciousness that have been put forward.

2.3.1 Definition of Consciousness

The distinction between an invisible physical world and a phenomenal stream of experience suggests a simple definition of consciousness:

Consciousness is the presence of a phenomenal world. (2.1)

This definition is based on the distinction between phenomenal and physical reality and it suggests that phenomenal states and consciousness can be treated as interchangeable terms. Some clarifications of this definition now follow.

What is the best way speak about the consciousness of X?

There are many different ways of speaking about the consciousness that is associated with an object or person X and since some of these are potentially misleading, I will endeavour to adhere to the following general rules throughout this thesis:

- Unspecific terms, such as "the red flower", "the system", "the network", etc., could refer either to the phenomenal aspect of X, which I experience with my human senses, or to its underlying physical reality. Most of the time it does not matter whether the physical or the phenomenal aspect of X is being referred to, since it is assumed that phenomenal X corresponds to an underlying physical X, and that parts of physical X can affect our stream of experience.[11]

- Some conscious states might not include a subject or a perspective, and so it is potentially misleading to claim that X is *in* a phenomenal world. Difficult problems with spatial perception also make the use of 'in' problematic - see Gamez (2007c, pp. 25-87) for a discussion.

- The approach to consciousness in this thesis is based around the identification of correlations between the phenomenal and physical worlds (see Section 2.5), which may eventually lead to a causal theory of consciousness. However, until this point is reached it is inappropriate to use phrases like "The consciousness of X is *caused* by brain state Y" or "The brain state Y *gives rise to* the consciousness of X."

- I will be using the word "associated" to express the link between conscious states and X. The person or object X in front of me is an object in my phenomenal world and I can measure the physical aspects of this object. If X makes plausible claims about its

[11] It seems likely that all systems have both phenomenal and physical aspects, but I am leaving this open at this stage. Although it might be thought that some systems could have a completely non-phenomenal character – a dark matter machine for example, or perhaps a highly dispersed gas – it would still be possible to construct phenomenal representations of these systems, such as a picture.

conscious states or if I make predictions about the conscious states of X, then I will express this by saying "there are conscious states *associated* with X" or "there are phenomenal states *associated* with X."

- Once we have an association between phenomenal states and a phenomenal/ physical X, then we can start to look for correlations between them. The specification of a correlation between a conscious state and a state of X is more technical than an association, and I will use "the consciousness *correlated* with X" to refer to a mathematical or statistical relationship between the consciousness associated with X and phenomenal/ physical X.
- Although "The conscious states *connected* with X" might seem to be a plausible alternative to "associated", it implies a causal relation in one or both directions, which assumes too much at this stage.
- "The consciousness *of* X", "conscious X" or "X's consciousness" will be used as convenient synonyms for "the consciousness associated with X."
- "What X is conscious of" will be used as a synonym for "The contents of the consciousness associated with X."

The only deliberate exception to these rules will be when I am explaining or paraphrasing the work of other people.

Definition 2.1 has nothing to do with language

Most of my conscious states have little to do with language or narrative, although I use language to reflect on them and communicate them to other people. It might turn out that consciousness is constantly correlated with language or self-reflexivity, but this is not something that needs to be incorporated into the most basic definition of the phenomena that we are attempting to study and explain.

Phenomenal worlds might be completely different

When I experience a person within my phenomenal world they are surrounded by objects that are part of my phenomenal experience. However, the objects that I perceive might not be included in the other person's world – they could be immersed in a uniform field of blackness or pain, for example. When we look at a schizophrenic patient, such as Schreber, we say that he is associated with a phenomenal world, but this world might be very different from our own.[12]

There is nothing special about qualia

In Section 2.2.3 I argued that there is no fundamental distinction between classic qualia, such as red, and our experience of space, time, movement and number. This suggests that the concept of qualia is either redundant or should be used as a synonym for phenomenal experience in general. Theories of consciousness apply to the whole phenomenal world, and not just to the colourful smelly parts of it. Critical discussions of qualia and their standard interpretation can be found in Dennett (1988, 1992) and Churchland (1989).

The concept of consciousness is a new and modern phenomenon

This definition of consciousness helps us to understand why the concept of consciousness is a relatively new phenomenon. In the discussion of the phenomenal and physical I showed how the modern concept of the phenomenal is strongly linked to the physical world described by science, which is a recent product of a great deal of conceptual, technological and experimental effort. Earlier societies lacked this notion of physical reality, and so it is not surprising that the concept of consciousness is absent from Ancient Greek, Chinese and in the English language prior to the 17th Century (Wilkes, 1984, 1988, 1995). Consciousness is a new and modern problem because science is a new and modern phenomenon. The stream of experience was once understood in

[12] See Schreber (1988) for a description of this world and Nagel (1974) for a more detailed discussion of this point.

relation to an invisible world of gods and spirits; now it is interpreted as a *conscious* phenomenal representation of quarks, atoms, superstrings and forces.[13]

A single concept of consciousness

Many people, such as Armstrong (1981) and Block (1995), have tried to distinguish several different notions of consciousness, whereas Definition 2.1 is based on a single type of consciousness that is present when there is a phenomenal world and absent when there is not. States that are claimed to be conscious according to Armstrong's minimal consciousness or Block's access consciousness, for example, are not conscious according to Definition 2.1.

Awareness

It is worth distinguishing the presence of a phenomenal world from the related concept of awareness. Although many people link consciousness and awareness,[14] it is possible to interpret awareness as the presence of active representations in the brain that are not necessarily conscious. For example, when I am cycling along a canal and imagining a recent concert, then I might be said to have sensory awareness of the canal, although I am not conscious of it. Likewise, I might be attributed awareness of the sound of the refrigerator in my kitchen, but I only become conscious of it when the compressor cuts out. To avoid ambiguities of this kind, I will not use awareness in any technical sense in this thesis.

Consciousness and wakefulness

According to Laureys et. al. (2002, 2004) many patients in a vegetative state can be awake without being conscious and display a variety of responses to their environment:

[13] Many people around today have a different interpretation of the stream of experience that is often closely aligned with idealism (see Section 2.4.1) and rejects the scientific interpretation of physical reality – Tibetan Buddhism is one example. There is not space in this thesis to cover these other theories in detail and the primary focus will be on the scientific study of consciousness, which is closely linked to the Western atheistic viewpoint.

[14] For example, the *Oxford English Dictionary*'s (1989) third definition of conscious is: "The state or fact of being mentally conscious or aware *of* anything." (Volume III, p. 756).

> Patients in a vegetative state usually show reflex or spontaneous eye opening and breathing. At times they seem to be awake with their eyes open, sometimes showing spontaneous roving eye movements and occasionally moving trunk or limbs in meaningless ways. At other times they may keep their eyes shut and appear to be asleep. They may be aroused by painful or prominent stimuli opening their eyes if they are closed, increasing their respiratory rate, heart rate and blood pressure and occasionally grimacing or moving. Pupillary, corneal, oculocephalic and gag reflexes are often preserved. Vegetative patients can make a range of spontaneous movements including chewing, teeth-grinding and swallowing. More distressingly, they can even show rage, cry, grunt, moan, scream or smile reactions spontaneously or to non-verbal sounds. Their head and eyes sometimes, inconsistently, turn fleetingly towards new sounds or sights.
>
> Laureys et al. (2002, p. 178)

Vegetative patients are *awake* when they have their eyes open and vocalise or grimace. These patients are *conscious* when they are experiencing a phenomenal world, and Laureys et al. (2004) suggest some of the clinical signs that can be used to judge when this is the case.

2.3.2 Comparison with Other Theories of Consciousness

This section compares Definition 2.1 with some of the more influential theories of consciousness.

What it is like

According to Nagel (1974) an organism is conscious if there is something that it is like to *be* that organism. However, it is possible (although unlikely) that there are phenomenal worlds without any stable correlation with phenomenal or physical things, and so defining consciousness in terms of this association with phenomenal and physical objects is adding too much to the concept at this stage. Furthermore, Nagel’s claims about the *subjective* character of experience suggests a necessary connection between consciousness and a perspectival self. Whilst some kind of self is undoubtedly important for higher organisms, it might not be an essential feature of consciousness and there might be forms of minimal consciousness that are without subjectivity – see, for example, Metzinger's minimal notion of consciousness in Section 2.6.4.

Nagel (1974) discusses how we are unable to describe the experiences of creatures that are very different from ourselves – for example when we attempt to describe the phenomenology of a bat. This problem also occurs when we attempt to describe the consciousness of artificial systems, and it is covered in more detail in Section 4.4.2. Nagel's resistance to various reductionist theories of consciousness is also very much in line with the approach to consciousness that is taken in this thesis.

Minimal, perceptual and introspective consciousness.

Armstrong (1981) distinguishes between three types of consciousness. The first, called minimal consciousness, is present when there is mental activity occurring in the mind. When we are in deep sleep we might have knowledge and beliefs, but there are no events or occurrences going on, and so we are not minimally conscious. However, a person solving a problem in his or her sleep is minimally conscious because thinking is a form of mental activity. Armstrong's second type of consciousness is perceptual consciousness, in which we are aware of what is going on in our body and environment. Dreaming is minimally conscious, but we only become perceptually conscious when we wake up and perceive the world. Finally Armstrong identifies a third type of consciousness, called introspective consciousness, in which we have perception-like awareness of the states and activities of our mind. This notion of introspective consciousness was invoked to handle cases like 'unconscious' driving, in which we are perceptually conscious of the road, but not fully conscious of it because we are thinking about other things.

An initial difficulty with Armstrong's first two types of 'consciousness' is that it makes little sense to call something conscious that takes place whilst we are in deep sleep or 'unconsciously' driving, and so I will set Armstrong's notions of minimal and perceptual consciousness aside in this thesis. A central problem with Armstrong's third notion of introspective consciousness is that it seems perfectly coherent that we could be aware of our own mental states without any form of consciousness being present, and such meta awareness is likely to be taking place all the time in the brain. For example, when we are driving ‘unconsciously’ and thinking about other things, low level sensory data is being passed to the parts of the brain that identify cars and plan motor actions, and these other parts could be said to be introspectively aware of the lower level data without any consciousness being present.

Higher order thought

Rosenthal (1986) starts by defining a *mental* state as a conscious or unconscious state that has sensory or intentional properties. These mental states are claimed to be conscious when they are accompanied by a higher-order thought and mental states without a higher order thought are said to be unconscious. Rosenthal claims that this presence or absence of higher order thoughts explains the consciousness or unconsciousness of mental states.

The problem with this account is that it is little more than a pseudo explanation that is introspectively and empirically unfounded. Rosenthal admits that we are unaware of our higher order thoughts, but claims that this is a necessary feature of his theory. If higher order thoughts were conscious, then an infinite chain of higher order thoughts would be needed to make each of the previous higher order thoughts conscious. To avoid this problem, Rosenthal claims that the higher order thoughts are unconscious and only become conscious when they are accompanied by third order thoughts. Whilst the unconsciousness of higher order thoughts is necessary to Rosenthal’s theory it does mean that their existence cannot be established through introspection. Since higher order thought theory can hardly be said to be grounded in empirical data about the brain, it is left as something that ‘explains’ phenomenal consciousness on the basis of something that is itself completely ungrounded and unexplained.

Rosenthal (1986) argues that one of the benefits of his theory is that it offers some kind of explanation of consciousness and “If nothing were more basic to us than consciousness, there would be nothing more basic in terms of which we could explain consciousness. All we could do then is try to make consciousness more comprehensible by eliciting a sense of the phenomena in a variety of different ways." (p. 352). The position of this thesis is that phenomenal experience is one of the most basic 'things' that there is and we need to elicit a sense of the phenomena in a variety of different ways before it we can start to hypothesize about its causes.[15]

Phenomenal and access consciousness.

Block (1995) claims that the word consciousness is used in two distinct ways, which he identifies as phenomenal consciousness (P-consciousness) and access consciousness (A-consciousness). P-consciousness is experience and the experiential properties of a state are "what it is like" to have that state - for example, we have P-conscious states when we hear, see, smell, taste and have pains. On the other hand, access-conscious states are representational and their content is available as a premise in reasoning and for the rational control of action. Since many phenomenal contents are also representational, this distinction can be expressed by saying that it is in virtue of the phenomenal aspect of a state's content that it is P-conscious, whereas it is in virtue of a state's representational content that it is A-conscious. Block uses this distinction to argue against the claim that P-consciousness carries out a particular function, such as high level reasoning - a hypothesis that is often put forward in connection with cases of blindsight and epileptic automatism. Whilst A-consciousness is a functional notion, P-consciousness is not, although it might be systematically correlated with certain functions.

Block's separation of phenomenal consciousness from functions at the physical or information-processing level is entirely in keeping with the definition of consciousness in this thesis, which is based on a primary notion of phenomenal experience.[16] However, Block's notion

[15] Other criticisms of higher-order thought theory can be found in Gennaro (2004), Aquila (1990), Byrne (1997) and Rowlands (2001).

[16] However, Section 2.5 will argue that it does not make sense to speak about an *inaccessible* P-consciousness, which cannot be established through scientific investigation.

of access *consciousness* is much less convincing and hinges on his careful definition of what constitutes access to representational states, which enables him to claim that cases of blindsight and epileptic automatism are not A-conscious. It seems to make much more sense to separate the notion of a representational state from consciousness altogether and to speak about conscious and unconscious representational states – instead of introducing a second notion of consciousness to speak about non-phenomenal representational states. Block's claim that A-consciousness and P-consciousness have been historically confused is no doubt true, but this is not a reason to continue to speak about non-phenomenal *conscious* states when unconscious representational states are much more theoretically tractable.

2.4 Metaphysical Theories of Consciousness

One of the central questions in the philosophical study of consciousness has been whether the phenomenal and the physical are two separate realities or substances, or whether one can be reduced to the other. To answer this question a number of metaphysical theories of consciousness have been put forward.

2.4.1 Idealism and Phenomenology

Both idealism and phenomenology emphasise the phenomenal over physical reality. This type of theory ranges from Berkeley's (1988) claim that the concept of material substance is incoherent and ideas are the only reality, to Husserl's (1960) suggestion that we should suspend belief in the physical world and focus on the description of phenomenal experience, which might eventually enable us to ground science in phenomenological data. Although these theories are logically consistent and cannot be disproved, they have not developed a framework that can match science's success at prediction, and the hypothesis of a metaphysically real physical world leads to a much simpler interpretation of the phenomenal world. For example, it is much more useful to interpret a stone as a real physical object that can be investigated in a variety of different ways, instead of as a collection of ideas that were put into our minds by God. For these reasons, I will set aside idealism and phenomenology in this thesis and focus on theories that accept the metaphysical reality of the physical world.

2.4.2 Interactionist Dualism

Interactionist dualism is the claim that the phenomenal world is a second thinking substance, which is completely distinct from the substance of the physical world (Descartes 1975, Eccles 1994). As our physical bodies move around in the physical world, our physical brains receive data through the senses and pass it to the thinking substance, where it becomes conscious. When our conscious phenomenal states decide upon an action, instructions are passed back to the physical brain, which controls the muscles. Interactionist dualism was first put forward by Descartes (1975), who suggested that data was passed between the two substances through the pineal gland. The main advantage of interactionist dualism is that it makes a very clear distinction between conscious and unconscious representations.

One of the major problems with this theory is that it has great difficulty explaining the interaction between the two substances. The pineal gland is now known to be closely linked to the maintenance of circadian rhythms, and no evidence has been found for the hypothesis that it is the central channel of communication between the phenomenal mind and the physical brain. In fact it is unlikely that there is a single ‘seat of awareness’ anywhere in the brain (Crick and Koch 2003, Edelman and Tononi 2000), and so the dualist has to explain how a shifting pattern of neural activation is passed on to a second substance and how the second substance causally influences the shifting pattern of activation in the brain. No plausible or testable theory about how this could take place has ever been put forward.

A second problem with interactionist dualism is that our greater understanding of the brain is making the thinking substance increasingly redundant. At one time we might have felt that a second substance was needed to explain something as mysterious as imagination, whereas we can now attempt to explain it as the offline activation of sensory processing areas (Kossyln 1994, Kreiman et al. 2000). Similarly, we might have thought that *thinking* needed a second substance to explain it, whereas we can now see how this could be explained as part of our language-processing and imaginative abilities (Damasio 1995). We are moving towards a situation in which we will be able to explain all of the *functions* of the physical brain in terms of neural processes, which will leave nothing for the thinking substance to *do*. This turns the thinking substance of interactionist dualism into a passive recipient of data from the parts of the brain that are the neural correlates of consciousness, with all the processing carried out by the brain's neural mechanisms. This is basically a version of epiphenomenalism, which will be considered next.

2.4.3 Epiphenomenalism

Epiphenomenalism is often put forward as a way of solving the problems connected with a two-way interaction between the thinking and extended substance. Since the physical world is thought to be causally closed, epiphenomenalism advocates a one way interaction in which the phenomenal world 'sits on top' of the physical world and receives information from the physical brain without having any causal influence on it.

This type of theory often emerges from some form of dualism and it can be argued that pantheism and Nagel's (1974) 'something it is like to be something' are also versions of epiphenomenalism. Many examples of physicalism are also implicit or explicit versions of epiphenomenalism, since they generally look to the physical world for the information-processing carried out by the mind and then seek some extra quality or function of the brain that 'throws up' passive phenomenal qualia, whose only function is the indication of underlying physical states.[17] A physicalism that was not epiphenomenal would need to give phenomenal states a causal role, but this is almost never the case, and so physicalism almost always ends up being epiphenomenal about consciousness.

The central and fatal problem with epiphenomenalism is that it completely undermines our ability to talk about phenomenal states. The descriptions of consciousness generated by the physical brain are not causally connected with phenomenal states, and so it is impossible for them to be *about* these states. To illustrate this point, consider a situation in which I am consciously perceiving a green apple. In this case, there are all kinds of causal links from the world to the activity in my visual cortex and epiphenomenalism claims that there are also causal links from the activity in my visual cortex to a second substance in which the green apple becomes conscious. However, since the causal links to the second substance only go in one direction, when I say that I am conscious of the green apple, the activity in my larynx muscles is driven entirely by the physical activity in my visual cortex, and it is completely independent of whether or not there is a conscious green apple in the second substance. This situation is illustrated in Figure 2.1.

[17] Jackendoff's (1987) theory is close to this position, although he does not explicitly embrace the metaphysics of epiphenomenalism: "The elements of conscious awareness are caused by/ supported by/ projected from information and processes of the computational mind that (1) are active and (2) have other (as yet unspecified) privileged properties." (p. 23). As Jackendoff points out, in this interpretation consciousness does not have any effect on the world: "Yet another way of looking at Theory II and its corollaries is as a claim that consciousness is *causally inert*. This may seem harmless enough until we realize its ugly consequence: *Consciousness is not good for anything*. The only way it can be good for anything is for it to have effects, and such possibility has just been denied. Again, the only construal of 'Consciousness is good for purpose *X*' within Theory II is as 'The computational states that cause/support/project consciousness are good for purpose *X*,' which does not exactly have the same ring of victory to it." (Jackendoff 1987, p. 26).

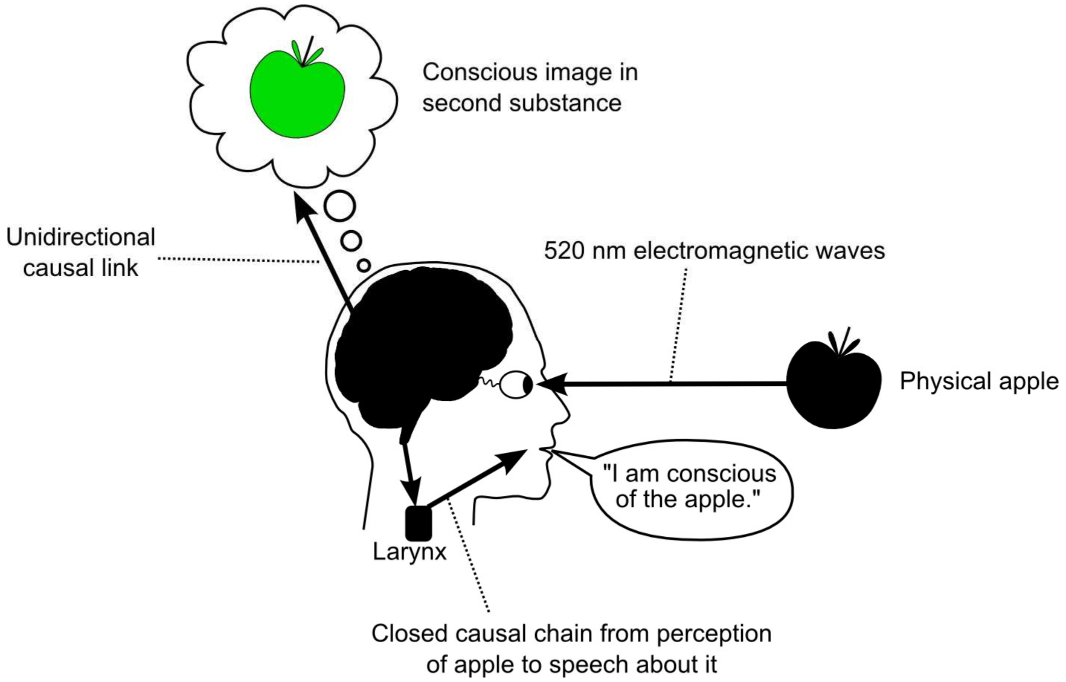

Conscious image in second substance
Unidirectional causal link
520 nm electromagnetic waves
Physical apple
"I am conscious of the apple."
Larynx
Closed causal chain from perception of apple to speech about it

Figure 2.1. Within epiphenomenalism there is only a one-way causal chain from physical reality to the second substance, and so our statements about consciousness are completely independent of our actual consciousness

Since there is complete causal dissociation between the contents of our consciousness and our speech about it, I will continue to state that “I am conscious of the apple” regardless of whether I am actually conscious of an apple, a banana or not conscious at all (see Figure 2.2). If conscious experience cannot affect physical reality, then our physical bodies have no evidence for their claim to be conscious: there is simply no way in which our physical bodies could ever know that there is an epiphenomenal second substance.

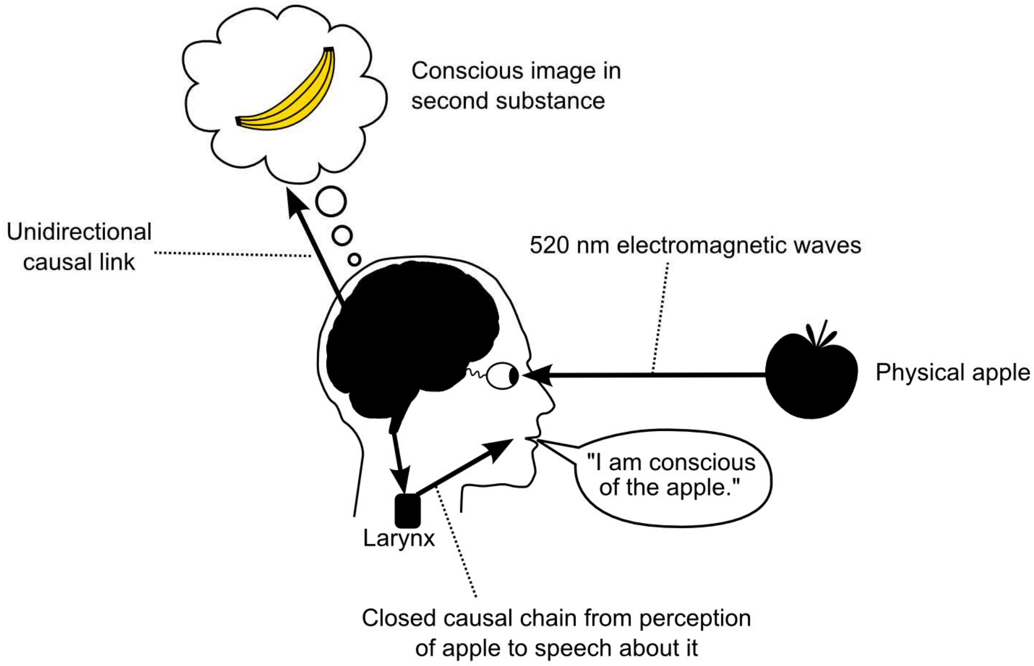

Conscious image in second substance
Unidirectional causal link
520 nm electromagnetic waves
Physical apple
"I am conscious of the apple."
Larynx
Closed causal chain from perception of apple to speech about it

Figure 2.2. According to epiphenomenalism, the contents of our consciousness have no effect on our speech. Although the apple sense data is transformed into a conscious image of a banana, my physical brain and body continues to state that I am conscious of an apple. Even if I became conscious of this disparity, I would be unable to talk about it because there is no causal influence from my consciousness to the physical world.

2.4.4 Physicalism

One of the most popular theories about consciousness is that there is only one substance, the material world described by physics, and consciousness has something to do with the information, processes, functions or structures within this physical substance (Poland 1994, Kim 2005). This material substance is associated with phenomenal states when it is arranged into working brains, and not conscious when it is arranged into rocks or chairs. The advantage of dualism was that it could easily accommodate properties, such as redness or the smell of lavender, within a second substance. In rejecting this, physicalism leaves itself with the problem that phenomenal properties are absent from the world described by physics. However we arrange the physical world we will never arrange it into redness or the smell of lavender.[18] These difficulties with integrating the physical and phenomenal worlds are discussed next.

2.4.5 The Easy, Hard and Real Problems of Consciousness

> In 1989 the philosopher Colin McGinn asked the following question: "How can technicolor phenomenology arise from soggy gray matter?" (1989: 349). Since then many authors in the field of consciousness research have quoted this question over and over, like a slogan that in a nutshell conveys a deep and important theoretical problem. It seems that almost none of them discovered the subtle trap inherent in this question. The brain is not grey. The brain is colorless.
>
> Metzinger (2000, p. 1)

Chalmers (1996) put forward a distinction between the 'easy' problem of explaining how we can discriminate, integrate information, report mental states, focus attention, etc., and the hard problem of explaining how phenomenal experience could arise from physical matter. Although solving the 'easy' problem is far from easy, we do at least have some idea how it can be done. On the other hand, although many theories have been put forward about the hard problem, it can be argued that we have no real idea about how to solve it.[19]

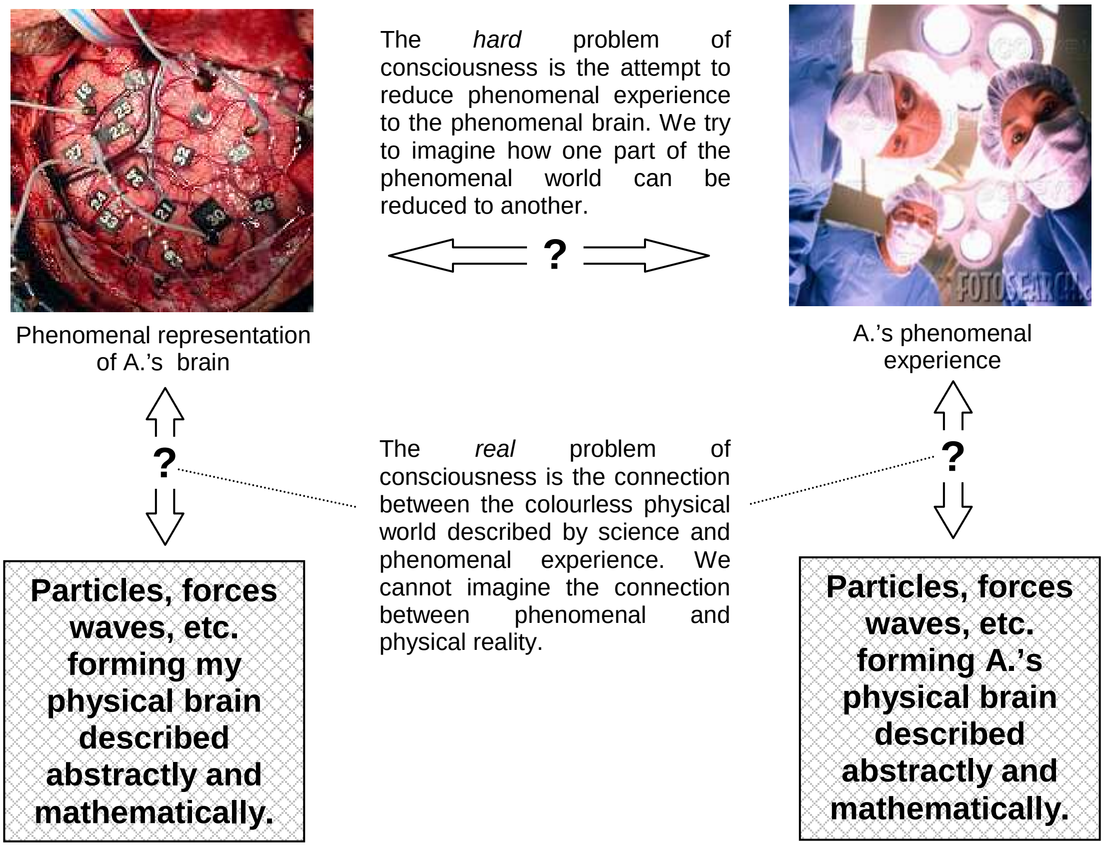
The hard problem of consciousness generally gains its intuitive force from an exercise in which we imagine (or perceive) a grey brain, imagine (or perceive) the colour red and then try to think how the colour red could be generated by the grey brain. This is a hard problem because we cannot *imagine* how the information-processing functions of the brain, for example, could lead to phenomenal red.

The problem with this attempt to imagine the hard problem of consciousness is that the physical brain is completely non phenomenal in character and so the hard problem of

[18] Although we have no problem *correlating* redness with electromagnetic signals of 428,570 GHz and lavenderness with molecules of Borneol, Geraniol, Linalool, Lavendulyl acetate, Linalyl acetate and Cineol.

[19] There has been extensive discussion in the literature on consciousness about whether Chalmers' hard problem is in fact a genuine problem and the different ways in which it can be tackled. Representative positions in this area can be found in Goguen and Forman (1995, 1996), Shear (1997) and Gray (2004).

consciousness can only be imagined by smuggling in our phenomenal representation of the physical brain and then trying to connect this phenomenal brain with a paradigmatic phenomenal red ‘quale’. When we think that we are imagining the physical world we are actually imagining our phenomenal representation of the physical world. The *hard* problem of consciousness is a puzzle about how phenomena can cause phenomena, whereas the *real* problem of consciousness is about how the phenomenal world is connected with real physical neurons, which we can describe scientifically and mathematically, but cannot perceive or imagine in any way. This difference between the hard problem of consciousness and what I am calling the real problem of consciousness is illustrated in Figure 2.3.



The *hard* problem of consciousness is the attempt to reduce phenomenal experience to the phenomenal brain. We try to imagine how one part of the phenomenal world can be reduced to another.

The *real* problem of consciousness is the connection between the colourless physical world described by science and phenomenal experience. We cannot imagine the connection between phenomenal and physical reality.

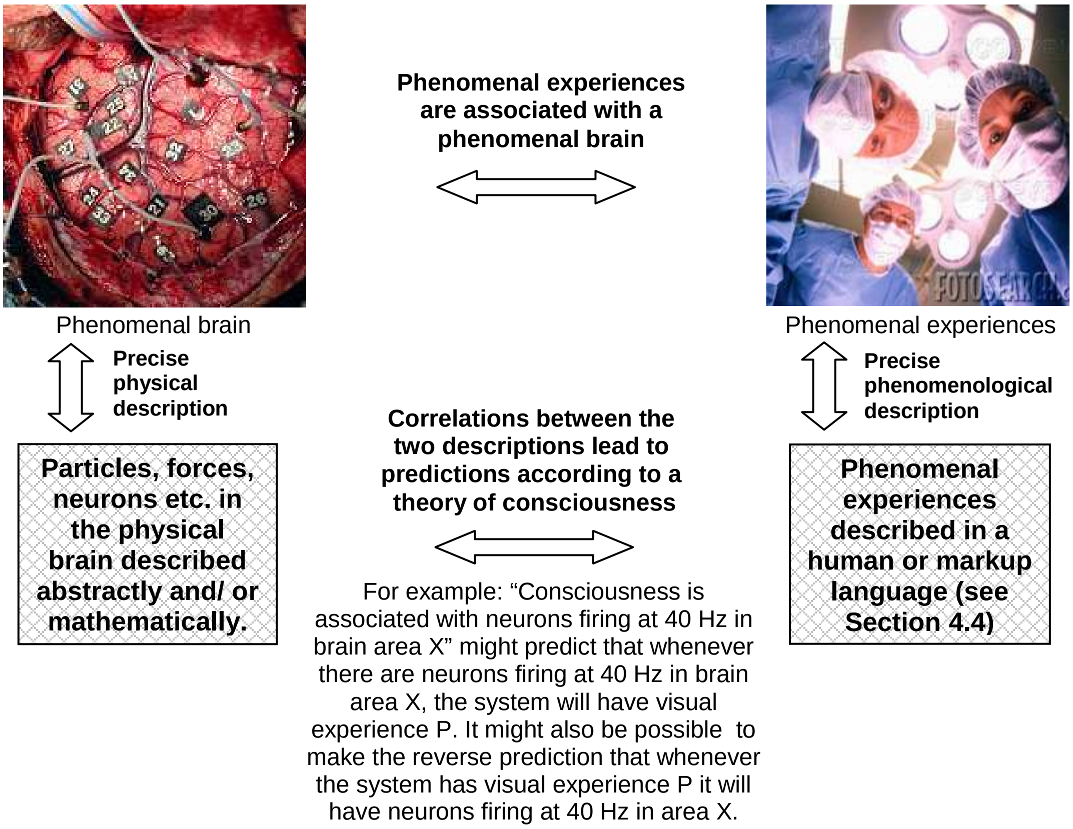
?
Phenomenal representation of A.’s brain
A.’s phenomenal experience
?
?
Particles, forces waves, etc. forming my physical brain described abstractly and mathematically.
Particles, forces waves, etc. forming A.’s physical brain described abstractly and mathematically.

Figure 2.3. The relationship between the hard and the real problem of consciousness. The brain picture on the left is my phenomenal representation of person A’s brain. The surgeon picture on the right is A’s phenomenal reality (the operation is under local anaesthetic).

The hard problem of consciousness attempts to reduce one part of phenomenal reality (the colour red) to another part of phenomenal reality (the phenomenal brain). Discussions of consciousness often get intuitively or imaginatively stuck on this hard problem, which will be never be solved because intuition and imagination are simply not applicable.

Real scientific problems are solved by creating abstract descriptions of phenomenal observations and hypothesising forces or other features of the physical world that link these abstract descriptions with one other. In this respect, the real problem of consciousness is no different from any other scientific theory since we have phenomenal observations of brains and phenomenal observations of our experiences and science can look for regularities between them, which we may eventually be able to explain using a theory of consciousness. It is relatively easy to describe the brain because we can use mathematics, physics and biology to precisely specify its physical aspects. Precise descriptions of phenomenal states are much more of a challenge because up to this point we have relied on natural human language for our phenomenological descriptions. Whilst statements like “I am experiencing a red blob in the left hand corner of my visual field” might be adequate for our current research on consciousness, there are good reasons why a more precise language for phenomenology might be more appropriate for a science of human consciousness, and a number of arguments are put forward in Section 4.4 why a markup language, such as XML, is already needed for the description of the phenomenology of artificial systems.

Once we have obtained precise descriptions of the physical and phenomenal states we can look for correlations between them and use theories about consciousness to make predictions about the phenomenal states that are associated with the physical states and the physical states that are associated with the phenomenal states. The accuracy and falsifiability of these predictions (Popper 2002) will depend on the precision of the physical and phenomenal descriptions. This scientific approach to the real problem of consciousness is illustrated in Figure 2.4.


Phenomenal experiences are associated with a phenomenal brain
Phenomenal brain
Phenomenal experiences
Precise physical description
Precise phenomenological description
Particles, forces, neurons etc. in the physical brain described abstractly and/ or mathematically.
Correlations between the two descriptions lead to predictions according to a theory of consciousness
Phenomenal experiences described in a human or markup language (see Section 4.4)
FOTOSEARCH

For example: “Consciousness is associated with neurons firing at 40 Hz in brain area X” might predict that whenever there are neurons firing at 40 Hz in brain area X, the system will have visual experience P. It might also be possible to make the reverse prediction that whenever the system has visual experience P it will have neurons firing at 40 Hz in area X.

Figure 2.4. First stage in a scientific solution to the real problem of consciousness. Precise descriptions are formulated of the physical brain and the phenomenal experiences associated with the physical brain, and these are used to identify correlations between the physical and phenomenal worlds. The predictions that different theories of consciousness make about these correlations can then be experimentally tested.

If we can discover theories that make good and perhaps perfect predictions about the relationships between the physical and phenomenal worlds, then we might start to think about how we could *explain* these predictions. A good example of this move from prediction to explanation is given by the evolution of our theories about the expansion of gases. A key stage in this work was Boyle’s law, published in 1662, which predicts that the pressure, *P*, and the volume, *V*, of a gas are related to a constant value, *k*, according to Equation 2.1:

$$PV = k \tag{2.1}$$

This equation is an empirical observation about the relationship between the pressure and volume of a gas, which can be used to predict how a fixed quantity of gas will respond to a change in pressure or volume according to Equation 2.2:

$$P_1V_1 = P_2V_2\,, \tag{2.2}$$

where P_1 and V_1 are the pressure and volume before the change and P_2 and V_2 are the pressure and volume after the change. These predictions made by Boyle's law were later *explained* by Bernoulli (1738), who showed how Equation 2.1 could be derived by applying Newton's laws to the motion of large numbers of molecules.

In the case of consciousness, if we can establish precise relationships between the phenomenal and physical descriptions, then we may eventually be able to move on to an explanation.[20] The form that such an explanation could take will probably only become clear once we have done a lot more work on the identification of correlations between the phenomenal and physical worlds, which will be covered next.[21]

[20] Since causal relationships are inherently temporal, it is coherent to claim that a phenomenal event causes a later physical event or a physical event causes a later phenomenal event, but it does not make sense to try to use a causal relationship to explain the co-occurrence of phenomenal and physical events at the same point in time - unless the common cause is something that is neither phenomenal nor physical and occurs before the simultaneous phenomenal and physical events.

[21] Coward and Sun (2007) put forward a general form for scientific theories of consciousness. Whilst their interpretation ignores the phenomenal/ physical distinction that has been argued to be essential for any science of consciousness, their suggestions about the hierarchical nature of scientific theories fit in well with the approach to synthetic phenomenology put forward in Chapter 4.

2.5 Correlates of Consciousness

2.5.1 Introduction

The discussion of metaphysical theories of consciousness has shown that systematic identification of the correlates of consciousness is an essential first step in the development of a scientific theory. Many people have started on this work and current investigations are mainly focused on the correlation between consciousness and the human brain, both because people are paradigmatic examples of conscious systems and because they are the only species that can make verbal reports about their phenomenal states. Although a great deal of work has been carried out on the neural correlates of consciousness in recent years (Chalmers 1998, Metzinger 2000), the firing of real biological neurons is not sufficient for consciousness, and might not even be necessary, and so this section covers a broad spectrum of potential correlates of consciousness (PCCs).[22]

The ultimate aim of the search for correlates of consciousness is to identify a list of necessary and sufficient conditions that would predict with certainty when a physical system is associated with phenomenal states and describe the contents of these states when they occur. Although our scientific theories would be much simpler if we found a single correlate of consciousness, it is possible that consciousness is correlated with a multiplicity of factors – for example, a particular combination of temperature and neural activity might be necessary. It is also possible that some factors will be partially correlated, which would only allow probabilistic predictions to be made about whether a system is conscious and what it is conscious of.

Adequate knowledge about the correlates of consciousness will enable us to predict phenomenal states from physical states and physical states from phenomenal states, but it will not prove that consciousness is causally dependent upon physical states any more than it will

[22] Without a commonly agreed definition of consciousness it is impossible to say whether we have identified *any* correlates of consciousness at this stage. For this reason, I will interpret all correlates of consciousness as *potential* in this thesis.

prove that physical states are causally dependent on consciousness. It is an open question how our theories about consciousness will evolve once we have mapped out the correlations between the phenomenal and physical worlds.

2.5.2 Potential Physical Correlates of Consciousness

The human brain is a paradigmatic example of a system associated with consciousness, and so any of its physical attributes are PCCs. None of these potential correlates is likely to be sufficient for consciousness because it is generally assumed that no consciousness is present when we are in deep sleep or a coma when the physical attributes remain unchanged.[23] Some examples of physical PCCs are as follows:

1. Volume of 1.4 litres.
2. Temperature of 310 K.
3. Weight of 1350 g.
4. Created after 1000 BCE.
5. Created through a process of natural selection.
6. Reflects light with a wavelength of 650 nm.[24]
7. Living neurons assembled from biological amino acids.
8. Haemoglobin.
9. Oxygen.
10. Rate of processing.

[23] This assumption may not hold if Zeki's (2003) notion of micro consciousnesses is correct. In this case one or more consciousnesses could be associated with a person in deep sleep or coma, which would not be verbally expressed because they are not integrated with the memory or vocal systems.

[24] I am using this as a convenient shorthand for the fact that the brain looks pinkish. In fact almost every non-black object reflects light of 650 nm to some degree and more care would be needed to formulate an accurate physical description of this property of the brain.

2.5.3 Potential Neural Correlates of Consciousness

Activity in biological neurons has been shown to be strongly correlated with consciousness and a large number of experiments have been carried out that have attempted to distinguish between neural activity that takes place when we are not conscious – in deep sleep or a coma, for example – and neural activity that is correlated with conscious experience. The emerging consensus is that the neural correlates of consciousness are likely to be distributed over many different brain areas - see, for example, Edelman and Tononi (2000), Crick and Koch (2003) Dehaene and Naccache (2001) or Zeki et al. (1998, 2003) - and the coordination between these areas might be achieved by synchronization of neural firing (Singer, 2000), NMDA synapses (Flohr, 2000), connections to thalamic nuclei (Newman et. al., 1997) or some combination of these mechanisms. The distributed neural correlates of the conscious model of our bodies are described in Melzack (1992) and Damasio (1995, 1999). Further discussion of the neural correlates of consciousness can be found in Chalmers (1998), Metzinger (2000) and Noë and Thompson (2004).

2.5.4 Potential Functional and Cognitive Correlates of Consciousness

The human brain can be analysed from the perspective of the large number of functions that it carries out, many of which might be correlated with consciousness. These range from the low level input and output functions of ion channels and neurons, up to higher level functions, such as perception, memory and cross-modal integration. The brain also carries out a number of cognitive functions that have been linked to consciousness, such as emotional evaluation of a situation, internal representations of the self, imagination and attention.

2.5.5 Experimental Determination of the Correlates of Consciousness

> We will focus on the notion of consciousness as such by contrasting pairs of similar events, where one is conscious but the other is not. The reader's conscious image of this morning's breakfast can be contrasted with the same information when it was still in memory, and unconscious. What is the difference between conscious and unconscious representations of the same thing? Similarly, what is the difference between the reader's experience of his or her chair immediately after sitting down, and the current habituated representation of the feeling of the chair? … All these cases involve contrasts between closely comparable conscious and unconscious events.
>
> These contrasts are like experiments, in the sense that we vary one thing while holding everything else constant, and assess the effect on conscious access and experience.
>
> Baars (1988, pp. 18-19)

To decide which PCCs are *actually* correlated with consciousness we need to measure the level of consciousness when the potential correlates are present individually and in different combinations, until we find the set that is systematically correlated with consciousness.[25] For example, if the human brain has attributes W, X, Y and Z, and removing Z and W has no effect on the consciousness of the system, but removing either X or Y individually or X and Y together leaves the system unconscious, then we can conclude that X and Y are necessary for consciousness. However, we can only conclude that X and Y are *sufficient* for consciousness if the human brain has no other attributes in addition to W, X, Y and Z that might be correlated with consciousness. For example, if the attribute C was left unchanged during the experiments, then it is possible that X + Y is not sufficient for consciousness and C has to be included as well. Some of the problems connected with this experimental process will now be covered in more detail.

[25] It is possible that there is more than one set of correlates of consciousness. For example, neurons constructed with silicon chemistry and neurons constructed using carbon chemistry may both be correlated with consciousness.

Selection of potential correlates

The first step in establishing the correlates of consciousness is to choose an initial set of potential correlates for experimentation. Since we know almost nothing about the link between the phenomenal and physical worlds, we cannot exclude anything with certainty, but we are likely to make more rapid progress if we start with a list of candidates that are broadly compatible with the Western scientific outlook.[26] To begin with, we can exclude potential correlates that are hard or impossible to test, such as the property of being created after 1000 BCE. However, this still leaves a potentially infinite number of testable PCCs, which we can only narrow down using our intuition about their potential link with consciousness.

A first problem with the use of intuition for this task is that our intuitions about consciousness are all taken from our phenomenal experiences and we have never experienced a direct link between phenomenal and physical reality. However, we do have a lot of experience of correlations between our phenomenal experiences and our phenomenal measurements of the physical world, which can be imagined and intuited. The intuitive exclusion of factors will have to be limited to human cases because we have never directly experienced animal or machine consciousness and any ‘observations’ of animal or machine consciousness have been extremely indirect, inferential and based on what we believe about human consciousness. Although we cannot reliably intuit whether a stone, for example, is capable of conscious states, we can discard many of the unique attributes of stones from our initial list of potential correlates because it is likely to be more profitable to start with attributes of humans, which we know to be conscious already.

A second problem with the use of intuition is that it can vary widely between people. For example, some people have an intuition that size is relevant to consciousness because all of the conscious systems that they have encountered have been within a certain range of sizes. This

[26] See Footnote 13.

leads to clashes of intuition in which some people are unwilling to believe that a system the size of the population of China could be conscious, whereas others do attribute consciousness to larger or smaller systems. When clashes of intuition do occur, it is generally better to leave the attribute as a PCC so that its validity can be established scientifically. In the longer term it is hoped that our intuitions about consciousness can be grounded by identifying the regularities in experience that gave rise to them.

Measurement of the physical system

To identify correlations between the physical and phenomenal worlds we need to measure changes in the physical system. Most of the potential physical correlates can be gauged using standard weight, volume and chemical measures and we have a wide range of ways of monitoring neural activity in the brain, such as EEG, fMRI, PET or implanted electrodes.[27] The functional and cognitive correlates of consciousness can be measured using psychological tests, and the functions of particular brain areas can be probed using patients with brain damage, animal models or by applying transcranial magnetic stimulation. All of these measurement techniques produce phenomenal representations of different aspects of the physical brain.[28]

Measurement of consciousness

Experiments on the PCCs also have to measure whether consciousness is associated with the system and, if consciousness is a graded phenomenon, the *amount* of consciousness that is present. Since consciousness cannot be detected with scientific instruments, its presence is

[27] These technologies are in the early stages of development and their low temporal and/ or spatial resolution limits the precision with which the neural correlates of consciousness can be identified.

[28] One potential measurement issue is that we might have to measure the system's *capacity* for some functions as well as the actual exercise of them within the system. For example, if it is possible to have conscious experiences that do not involve imagination, then it could be argued that imagination is not a necessary correlate of consciousness. However, this does not rule out the possibility that a *capacity* for imagination is a necessary correlate. The latter can only be ruled in or out by seeing if there are any conscious (probably brain damaged) people who lack all capacity for imagination. An example might be the amnesiac patients studied by Hassabis et al. (2007), who are not only bad at remembering the past, but at imagining new experiences as well.

established through first person reports in language, first person observations that are remembered and reported later, or through behaviour that is interpreted as the result of conscious experience – this is the only technique that can be used with animals, such as monkeys, which are trained to respond to a stimulus that is assumed to be conscious.[29] In all of these cases, the presence of consciousness is established through *behaviour* - our own behaviour when we write down our introspective observations, the verbal behaviour of a reporting subject or non-verbal animal or human behaviour.

A first problem with behavioural measures is that they are often inaccurate, especially when some form of brain damage is involved. This can occur when people are reporting everything in good faith with no intention of deceiving the experimenter. For example patients with Anton's syndrome claim to be able to see perfectly when in fact they are clinically blind and anosognosia patients will make claims about being able to use a paralyzed limb, for example, and confabulate wildly to explain its lack of movement (Ramachandran and Blakeslee 1998).

A second issue with the measurement of consciousness through immediate or deferred behaviour is that certain types of behaviour could themselves be correlates of consciousness. Since some behaviours, such as the statement "I am conscious right now", are more correlated with consciousness than anything else that can be varied in an experiment, this possibility cannot be completely ruled out. However, it does seem reasonable to suppose that a verbal report of my dream was not necessary for the occurrence of the dream, which I would have experienced independently of any external behaviour.

A third problem is that the probing of the conscious states might affect the conscious states themselves, either by distorting our memories of the conscious states or by priming us to

[29] See, for example, Logothetis' (1998) work on the neural correlates of consciousness. In these experiments macaque monkeys were trained to pull different levers in response to different images and Logothetis recorded from a variety of visual cortical areas in the awake monkey whilst it performed a binocular rivalry task.

interpret the situation in a particular way. Furthermore, as Dennett points out in his discussion of Orwellian and Stalinesque revisions (Dennett, 1992: pp. 101-38), the ordering of events can be ambiguous at small time scales, and so when we report our conscious experience of a visual illusion, for example, there is an ambiguity between a false memory of something that did not consciously take place and a correct memory of a false conscious event. Dennett (1992) uses this ambiguity to argue that there is no single Cartesian Theatre in which a determinate stream of consciousness takes place and there are just multiple drafts of narrative fragments under constant revision by multiple brain processes. These multiple drafts can be probed at different times and places to precipitate different narratives from the subject, but there is no single canonical stream of consciousness.

The most serious problem with a behavioural measure of consciousness is that it limits us to experiments that change the behaviour of the system. If an experiment does not alter the system's behaviour between the time of the experiment and the system's death, then it is impossible to tell if it has changed the system's phenomenal states. The behaviour-neutral experiment might have changed the consciousness of the system (in this case, the attributes under investigation are necessary and perhaps even sufficient for consciousness), or it might have had no effect at all on the system's consciousness (the attributes are extraneous factors that should be eliminated from the list of potential correlates), and we have no way of telling which is the case. The physical aspects of a system that were covered in Section 2.5.2 are the most behaviour-neutral, since size, temperature and material can all be changed whilst the behaviour is held constant, which makes it impossible to measure the correlation between any of these factors and consciousness. To make this point clearer I will look at an experiment that is often discussed in the literature in which part of the brain is replaced by a functionally equivalent chip.

2.5.6 Brain Chip Replacement

To identify the necessary and sufficient correlates of consciousness each PCC needs to be tested independently. Consciousness might be correlated with some of the functions carried out by the physical brain and/or with the biological material of the brain, and so we need experiments that change the material of the brain whilst holding the functions constant, and experiments that change the functions of the brain whilst holding the material constant. One way of holding the functions constant and changing the material is to replace part of the brain by a functionally equivalent silicon chip. For example, if the replacement of part of the lateral temporo-occipital cortex with a functionally equivalent chip caused a person to lose consciousness of movement information,[30] then we could conclude that the brain's biological substrate and functions are *both* necessary for consciousness. Although this is currently only a thought experiment, people are working on the development of a silicon hippocampus,[31] and so it might be possible to carry out this experiment in the future.

The central problem with this experiment is that the chip carries out exactly the same functions as the brain area that it is replacing, and so the overall functioning of the brain – and the behaviour of the person - is not altered by the operation. As Moor (1988) and Prinz (2003) point out, neither an external observer nor the person who received the chip would observe any effect of the implant on consciousness. An outside observer would not detect the replaced part because the function of the lateral temporal-occipital cortex would still be carried out by the chip. The person would continue to report and describe the movement information processed by affected area, even though there might not be any consciousness of movement present. From an

[30] This example is based on a patient studied by Zihl et. al. (1983, p. 315), who completely lost her ability to perceive motion after bilateral cerebral lesions in the lateral temporo-occipital cortex: "She had difficulty, for example, in pouring tea or coffee into a cup because the fluid appeared to be frozen, like a glacier. In addition, she could not stop pouring at the right time since she was unable to perceive the movement in the cup (or a pot) when the fluid rose. Furthermore the patient complained of difficulties in following a dialogue because she could not see the movements of the face and, especially, the mouth of the speaker."

[31] See http://www.newscientist.com/article.ns?id=dn3488.

outside point of view, this would not even seem like a confabulation because the visual system would be working perfectly.

A first-person perspective does not help matters either. Since the chip is functionally connected to the rest of the brain in the same way that the lateral temporal-occipital cortex was before the operation, the person’s language centres should report phenomenal movement in the same way that they did before, and so they will continue to think that they are experiencing movement, even if they have no consciousness of movement. Searle (1992, pp. 66-7) thinks that the person might feel *forced* to say that they are experiencing movement whilst they remain conscious of the fact that there is no phenomenal movement present. However, if the person was conscious of this compulsive language behaviour, then they would be able to remember and report it at a later time, which would be a functional change in the system that has been excluded by this experiment. It seems that even a first-person perspective cannot be used to decide whether consciousness is affected by the replacement of biological neurons with a functionally equivalent chip.

Against this Chalmers (1996) argues that verbal behaviour and consciousness would be very tenuously connected if we could lose our conscious experience of movement and yet continue to describe movement using language. The problem with this objection is that the implantation of a chip involves invasive surgery and it is not uncommon for people with brain damage to be systematically mistaken about their experiences and confabulate to an extraordinary extent to cover up their deficiency. As was pointed out in the previous section, people with Anton’s syndrome are blind and yet insist that they can see perfectly and hemineglect patients will bluntly assert that a paralysed arm is functionally normal. Faced with these cases, it cannot be assumed that it is impossible for us to be systematically mistaken about our phenomenal states. Further criticisms of Chalmers’ argument can be found in Van Heuveln et. al (1998) and Prinz (2003).

The brain-chip experiment can be applied to part of the brain or to the entire brain and in all cases the system's behaviour will remain constant. The same argument applies to other experiments on the brain's material, such as a change in temperature or the use of synthetic blood to probe the link between haemoglobin and consciousness.[32] Both a change in temperature and the exchange of real for artificial blood would leave the behaviour of the patient untouched, and we would be left wondering whether it removed the consciousness and left the behaviour intact or had no effect on consciousness. As Harnad (2003) points out, all our attributions of consciousness to a system are based on its behaviour, and so something that does not change the behaviour cannot be separated out as a correlate of consciousness:

> The only way to sort out the relevant and irrelevant properties of the biological brain, insofar as consciousness is concerned, is by looking at the brain's behaviour. That is the only non-telepathic methodology available to us, because of the other-minds problem. The temptation is to think that 'correlations' will somehow guide us: Use brain scanning to find the areas and activities that covary with conscious states, and those will be the necessary and sufficient conditions of consciousness. But how did we identify those correlates? Because they were correlates of behaviour. To put it another way: When we ask a human being (or a reverse-bioengineered robot) 'do you feel this?' we believe him when he says (or acts as if) he feels something – not the other way round: It is not that we conclude that his behaviour is conscious because of the pattern of brain activity; we conclude that the brain activity is conscious because of the behaviour.
>
> Harnad (2003, p. 74)

If some PCCs cannot be ruled in or out, then *we will never be able to identify a list of necessary and sufficient correlates of consciousness and we will never be able to tell for* ***certain*** *whether a system is associated with phenomenal states*. This distinction between correlates of consciousness that can and cannot be separated out will now be formalized as a distinction between type I and type II correlates of consciousness. Type I PCCs are behaviour neutral and so

[32] The temperature change would have to be carried out so that it did not affect the functionality of the brain or allowed the same functionality to take place over longer time scales. The synthetic blood would have to be one of the varieties that was not based on haemoglobin.

their link with consciousness cannot be experimentally tested; type II PCCs do affect the behaviour of the system and their impact on consciousness can be measured. This distinction will now be discussed in more detail and it will used to address questions about the potential consciousness of non-biological systems in Chapter 4.

2.5.7 Type I Potential Correlates of Consciousness

Type I PCCs are either behaviour-neutral or they cannot be separated from the behaviour that is used to measure consciousness in a system. Their key characteristic is that no experimental measure of their connection with consciousness can be devised or suggested. Many PCCs are type I because they can be changed independently of the functional properties of the system. The brain-chip replacement experiment illustrates how this is true for the material substance of the brain and the rest of the physical PCCs in Section 2.5.2 are all type I as well. The second class of type I PCCs is linked to our ability to remember and/or report phenomenal experiences. A change to the system that eliminates its ability to express its phenomenal experiences or prevents it from remembering them for later expression cannot be used to test for correlations with consciousness because it destroys the measuring instrument that is needed for the experiments. Memory and vocalisation/ behaviour can be removed individually – for example, in short term memory loss patients or REM sleep - but if both are lost together, then we can no longer measure consciousness in the system. For example, if Zeki’s (2003) notion of micro-consciousness is correct, there could be consciousness in deep sleep and coma, which cannot be remembered or reported because key brain areas are inactive or damaged. This suggests that some forms of global integration and binding might also be type I PCCs: if there is no integration between the visual cortex and other parts of the brain, then there will be no reports or memories of visual experience. The loss of integration could have eradicated visual consciousness from the system or it could have eliminated the system's ability to remember and report visual experience.[33]

2.5.8 Type II Potential Correlates of Consciousness

Type II PCCs can be separated out using behaviour and there is no overlap with the parts of the system that are used for measuring or reporting consciousness. When a type II PCC is removed or altered, the system's reports of conscious states can change. Activity in particular brain areas is a type II correlate because we can vary this activity through transcranial magnetic stimulation or observe brain damaged patients and measure the change in consciousness through verbal or other behaviour. Functional correlates also fall into this category because it is conceivable that we could disable a person's capacity for imagination or emotion, for example, and then probe their conscious states.

2.6 Three Theories of Consciousness

2.6.1 Introduction

The distinction between type I and type II PCCs can be applied to theories about consciousness:

- Type I theories of consciousness cannot be experimentally validated, either because they are based on type I correlates or because they are metaphysical statements of belief about the world that can never be tested. This type of theory is essentially an *a priori* statement of belief about the world that sets out a framework for interpretation and is completely un- or pre-scientific in character.

- Type II theories of consciousness *can* be empirically verified through experiments because they are based on type II PCCs.[34]

[33] It is even conceivable that we are conscious when dead, but unable to produce any form of behavioural output.

It would be an impossible task to examine all type II theories in this thesis, and I have decided to focus on Tononi's (2004) information integration theory of consciousness, Aleksander's (2005) axioms and Metzinger's (2003) constraints, which will be used to make predictions about the consciousness of the neural network that is described in Chapter 5.[35] Tononi's information integration theory of consciousness was chosen because it is a numerical method that can be used to automatically make predictions about the conscious parts of an artificial neural network. Aleksander's (2005) axiomatic theory was selected because it has been influential within the machine consciousness community and it provides a nice link between cognitive mechanisms and phenomenal consciousness. Metzinger's (2003) constraints were chosen because they are comprehensively worked out at the phenomenal, functional and neural levels and three of his constraints can be used to define a minimal notion of consciousness. Taken together, these three theories cover the cognitive characteristics of consciousness and some of its potential neural correlates, and it is fairly clear how they could be used to analyse a system for consciousness. Although I am focusing on these three theories in this thesis, the approach to machine consciousness that I am developing is quite general and can easily be extended to other type II theories.

The rest of this section gives an overview of Tononi's, Aleksander's and Metzinger's theories of consciousness, which will be used to demonstrate how detailed predictions can be made about the consciousness of a system using different theories. As Crick and Koch (2000) point out, a comparison between predictions and empirical measurements will eventually determine which theories are accepted and rejected by science:

[34] All theories about consciousness operate within a framework of assumptions that is *a priori* at some level. However, type I theories of consciousness will never be empirically verifiable within the current scientific paradigm, whereas it may be possible to test type II theories.

[35] The most serious omission is global workspace theory (Baars 1988), which has been influential in research on consciousness and machine consciousness. An overview of machine consciousness work in this area can be found in Section 3.5.6.

> … while gedanken-experiments are useful devices for generating new ideas or suggesting difficulties with existing ideas, they do not lead, in general, to trustworthy conclusions. The problem is one that should be approached scientifically, not logically. That is, any theoretical scheme should be pitted against at least one alternative theory, and *real* experiments should be designed to choose between them.
>
> Crick and Koch (2000, p. 103)

In this thesis, Tononi's, Aleksander's and Metzinger's theories are being used to demonstrate how detailed predictions can be made about the consciousness of a system, and in the future it is hoped that it will be possible to compare these detailed predictions with a system's reports about consciousness. For this purpose only minor improvements or criticisms are necessary and no attempt will be made to integrate the three theories together or to put forward a new theory of consciousness.

2.6.2 Information Integration

The theory of information integration was developed by Tononi and Sporns (2003) and elements of it are also covered in Edelman and Tononi (2000). Information integration is measured using the value Φ, which is the amount of causally effective information that can be integrated across the informational weakest link of a group of elements. The information integration theory of consciousness is the claim that the capacity of a system to integrate information is correlated with its amount of consciousness and the quality of consciousness in different parts of the system is determined by the informational relationships (Tononi 2004). To test the link between information integration and consciousness Tononi and Sporns (2003) and Tononi (2004) evolved neural networks with different values of Φ and showed how they are structured in a similar way to the parts of the brain that are correlated with consciousness.

To measure the information integrated by a subset of elements, S, the subset is divided into two parts, A and B. A is then put into a state of maximum entropy (A^{HMAX}) and the entropy of B is measured. In neural terms, this involves trying out all possible combinations of firing patterns as outputs from A, and measuring the differentiation of the firing patterns produced in B. The *effective information* (EI) between A and B is a measure of the entropy or information shared between them, which is given in Equation 2.3:

$$EI(A \rightarrow B) = MI(A^{HMAX}; B), \tag{2.3}$$

where MI(A; B) is given by:

$$MI(A; B) = H(A) + H(B) - H(AB). \tag{2.4}$$

Since A has effectively been substituted by independent noise sources, there are no causal effects of B on A, and so the entropy shared by A and B is due to the causal effects of A on B. EI(A→B) also measures all possible effects of A on B and EI(A→B) and EI(B→A) are in general not symmetrical. The value of EI(A→B) will be high if the connections between A and B are strong and specialized, so that different outputs from A produce different firing patterns in B. On the other hand, EI(A→B) will be low if different outputs from A produce scarce effects or if the effect is always the same.

The next stage in the measurement of effective information, is the repetition of the procedure in the opposite direction by putting B into a state of maximum entropy and measuring its effect on A, giving EI(B→A). For a given bipartition of the subset S into A and B, the effective information between the two halves is indicated by Equation 2.5:

$$EI(A \leftrightharpoons B) = EI(A \rightarrow B) + EI(B \rightarrow A). \tag{2.5}$$

The amount of information that can be integrated by a subset is limited by the bipartition in which $EI(A \leftrightharpoons B)$ reaches a minimum, and to calculate this *minimum information bipartition* the analysis is run on every possible bipartition. Since $EI(A \leftrightharpoons B)$ is bounded by the maximum information available to A or B, $EI(A \leftrightharpoons B)$ has to be normalised by $H^{MAX}(A \leftrightharpoons B)$ when the effective information of each bipartition is compared (Equation 2.6).

$$H^{MAX}(A \leftrightharpoons B) = \min\{H^{MAX}(A); H^{MAX}(B)\}, \tag{2.6}$$

The *information integration* for subset S, or Φ(S), is the non-normalised value of EI($A \leftrightharpoons B$) for the minimum information bipartition.

Tononi and Sporns (2003) define a complex as a part of the system that is not included in a larger part with a higher Φ. To identify the complexes it is necessary to consider every possible subset S of *m* elements out of the *n* elements of the system starting with $m = 2$ and finishing with $m = n$. For each subset Φ is calculated and the subsets that are included in a larger subset with higher Φ are discarded, leaving a list of complexes with Φ > 0 that are not included within a larger subset with greater Φ. The *main complex* is then defined as the one that has the maximum value of Φ, and Tononi (2004) claims that this main complex is the conscious part of the system. To substantiate his link between Φ and consciousness, Tononi (2004) compares different network architectures with structures in the brain and shows how the architectures associated with high Φ map onto circuits in the brain that are associated with consciousness. The details of the algorithm that was used to calculate Φ are given in Section 7.4.2 along with some optimisations that were developed for large networks.

Information integration is a type II theory because it makes testable predictions about the link between consciousness and high Φ. For example, subjects should only report that they are conscious of information that is held in the main complex and it might be possible to change the amount of information integration in animals and measure the effect on consciousness. The main weakness of Tononi’s approach is that it is based on extremely simplified networks consisting of 10-20 elements, which makes it a rather speculative interpretation of circuits in the brain consisting of hundreds of millions of neurons. The positive side of this approach is that it links up with other work on effective connectivity and binding and it is less dependent on a subjective interpretation of the system’s constituent parts than other methods – for example, to apply

Tononi and Sporns algorithm we do not have to decide whether a particular layer represents emotions.[36]

2.6.3 Aleksander's Axioms

Aleksander and Dunmall (2003), Aleksander (2005) and Aleksander and Morton (2007c) have developed an approach to machine consciousness based around five axioms that are claimed to be minimally necessary for consciousness. According to Aleksander, this is a preliminary list of mechanisms that could make a system conscious, which should be revised as our knowledge of consciousness develops – a useful starting point that can be used to test ideas and develop the field. These axioms were deduced by Aleksander using introspection and he also identifies neural mechanisms that could implement them in the brain. Each of the axioms will now be covered in more detail.

1. Depiction

Depiction occurs when a system integrates sensory and muscle position information into a representation of an 'out there' world. The key characteristic of depiction is that visual or other perceptual information is integrated with proprioceptive information to give the sensation of something that is *out there*, which is very different from a photographic representation. Aleksander claims that this axiom is implemented in the brain by cells that respond to a particular combination of sensory and muscle information, such as the gaze-locked neurons discovered by Galletti and Battaglini (1989). These cells respond to small visual stimuli only when the monkey's eyes are pointing in a particular direction: if the monkey changes its direction of gaze, different cells respond to the same visual stimulus. Other senses exhibit depiction as well, with touch being the next most depictive, followed by hearing and then smell

[36] See Section 7.4.7 for some other criticisms of information integration.

and taste, which are hardly depictive at all. Depiction is the most important axiom and it is a key mechanism for conscious representation.

2. Imagination.

Imagination occurs when the system recalls parts of the world that are not physically present and this ability can be used to plan actions by constructing sequences of possible sensations. Imagination is linked to the sustained activation of depictive firing patterns, which is likely to depend on feedback or re-entrant connections in the brain. Research on mental imagery suggests that the parts of the brain that are used in sensation are reactivated in imagination (Kossyln 1994, Kreiman et al. 2000), with the difference that they can be active in different combinations, so that we can imagine things we have never encountered before. Many different theories have been put forward about how information in the brain areas involved in perception or imagination is bound together. Aleksander and Dunmall (2000) claim that this is done by associating the different sensory areas with a single location in muscular space, which unifies them into a single object that feels out there in the world. The vividness of imagination decreases in proportion to the degree to which the senses are capable of depiction, and so our most vivid imagined sense is vision, followed by touch and then audition. Smell and taste are almost impossible to imagine or remember accurately.

3. Attention

Attention refers to the process of selecting what we experience in the world and what we think about in our imagination. Our attention can be attracted automatically, for example when we hear a loud noise, or we can purposefully select the parts of the world that we depict or imagine. In the human brain, the superior colliculus is one of the areas that is involved in the selection of the eye position as part of the process of visual attention.

4. Volition

The terminology that is used to describe this axiom has shifted over time, with Aleksander and Dunmall (2003) referring to it as “planning”, whereas Aleksander and Morton (2007c) refer to it as “volition” to distinguish it from rule-based planning processes. This axiom refers to the fact that we are constantly thinking ahead, considering alternatives and deciding what to do next. The neural machinery for this process is the same as that in axiom 2, since the re-entrant neural connections that facilitate imagination also enable the network to move through sequences of states to plan actions. Volition is conscious when it involves depictive areas and the emotions are used to select the plan that is to be executed.

5. Emotion

We have feelings, emotions and moods and use them to evaluate planned actions. Some emotions, such as pleasure and fear, are hardwired at birth, whereas others develop over the course of our lives – for example, the feeling of hurt that we experience when we have been rebuked. Aleksander expects that the neural firing patterns associated with emotions will have distinctive characteristics, which enable them to be associated with perceived and imagined depictive events. As planning proceeds, predicted states of the world trigger neural activity in the emotion areas that determine which plan is selected for execution.

Aleksander’s axioms are a clear set of mechanisms that are a useful starting point for work on machine consciousness. Although I am reluctant to follow Aleksander (2005, pp. 33-4) in claiming an identity between neural activity and conscious sensations, I am happy to interpret the axioms as potential cognitive correlates of consciousness, and to interpret the neural mechanisms behind the axioms as potential neural correlates of consciousness. Aleksander’s axioms are a type II theory because they have been established through introspection and it should be possible to test their correlation with consciousness - for example, by finding people who lack one or more of the axioms and asking them about their conscious experience. The axiomatic theory also predicts that people without a link between motor information and sensory input should be incapable of depiction, and it might be possible to test this using lesions in a monkey. Aleksander's neural implementation of the axiomatic mechanisms, which he calls the kernel architecture, is summarized in Section 3.5.1.[37]

2.6.4 Metzinger's Constraints

Metzinger (2003) sets out a detailed theory of consciousness that is based around eleven constraints on conscious experience:

1. Global availability
2. Window of presence
3. Integration into a coherent global state
4. Convolved holism
5. Dynamicity
6. Perspectivalness
7. Transparency
8. Offline activation
9. Representation of intensities
10. "Ultrasmoothness": the homogeneity of simple content
11. Adaptivity

[37] A critical discussion of Aleksander's axioms can be found in Bringsjord (2007). One of the problems raised by Bringsjord is the lack of formalization of the axioms, which is addressed to some extent by the definition given in Section 7.6.2.

These constraints should be met by any fully conscious mental representation and Metzinger (2003) gives detailed descriptions of their neural, functional and computational correlates. Metzinger's constraints are all based on type II correlates of consciousness because their phenomenal, functional and neural aspects can be introspectively and experimentally measured in a system. A brief summary of the constraints now follows.[38]

1. Global availability

Phenomenal information is globally available for deliberately guided attention, cognitive reference and control of action. Our attention can be attracted by or directed to any part or aspect of our conscious mental content and we can react to this content using a multitude of our mental and bodily capacities. Globally available cognitive processing is characterized by flexibility, selectivity of content, and a certain degree of autonomy. One of the functions of global availability is to increase the behavioural flexibility of the system, enabling many different modules to react to the same conscious information, and it also supports goal directed behaviour and the selective control of action. The neural correlates of global availability are not clear at present and form part of the general question about how different areas of the brain are integrated together. One theory is that large scale integration is mediated by the transient formation of dynamic links through neural synchrony over multiple frequency bands (Varela, Lachaux, Rodriguez, and Martinerie 2001) and Tononi and Sporns' (2003) information integration offers a way of measuring the degree of global integration (see Section 2.6.2). In contrast to constraints 2-10, global availability is a functional constraint and it is described by Metzinger as a third-person version of constraint 3.

[38] Metzinger (2003) also gives an account of the phenomenal self model and intentional relation. Whilst these are important aspects of human consciousness, they are less relevant to this thesis and I will only cover Metzinger's constraints here.

2. Window of presence

We experience conscious events in a single now within which a number of things happen simultaneously. In this now events can be represented as having duration or integrated into temporal figures, such as a musical tune. Events within the now have an organisation and vividness that is lacking from events outside it, and the window of presence is embedded in a unidirectional flow of events, which join and leave it. This constraint is supported by short term memory, which keeps phenomenal contents active for some time after the stimuli have disappeared from the receptive field. Functionally this constraint involves the definition of windows of simultaneity, so that all physical events registered within each window are temporally identical. By avoiding the definition of temporal relations within each window the fine structure of physical time becomes transparent to the system[39] and temporal elements can be ordered in a sequence. The neural correlates of this constraint are not well known, although some form of recursion will be necessary to sustain past events. Metzinger cites Pöppel's (1972, 1978, 1985, 1994) theories about how oscillatory phenomena in the brain could provide a rigid internal rhythm, which could generate the elementary integration units.

3. Integration into a coherent global state

Phenomenal events are bound into a global situational context within which we are *one* person living in *one* world. Other situations are not phenomenally possible - the phenomenal world and the phenomenal self are *indivisible*. This constraint also refers to the fact that phenomenal events are densely coupled: as we interact with the world, the states change whilst the apparently seamless integrated character of the overall picture is preserved. One function of global availability is to reduce the ambiguity of the world down to a single compressed representation and a single consciousness is also most appropriate for a single body. Metzinger discusses how this constraint functions as a stable background for imaginative planning that cannot be

[39] See constraint 7.

transcended by the system, so that alternative simulated worlds can be compared with a representation that is tagged as the actual world and the system does not get lost in its own simulations. A global conception of the whole is also necessary in order to understand other objects and events as parts of the whole. The neural correlates of global availability are similar to those for constraint 1 and Metzinger mentions Flohr's (2000) hypothesis about the role of the NMDA receptor complex in achieving large scale integration of ongoing activity. Tononi and Sporns' (2003) information integration measure (see section 2.6.2) is also applicable here.

4. Convolved holism

Phenomenal wholes do not exist as isolated entities, but appear as flexible nested patterns. We experience phenomenal wholes – horse, house, person – that are parts of larger wholes – stables, city, crowd - and can be broken down into smaller wholes that form their parts – legs, body, head, walls, windows, roof, etc. This constraint functions to integrate information together into a unified superstructure and the binding of information at different levels could be achieved using temporal coherence on different frequency bands, as discussed for constraint 1.

5. Dynamicity

Our conscious life emerges from a series of psychological moments that are integrated over time and represented as being in states of presence, duration and change - they are not a succession of isolated events. Whilst constraint 2 refers to the single now that exists at any point in time, this constraint refers to the integration of events over longer periods and to the change in objects over time - something like a temporal version of convolved holism. The functional mechanisms behind dynamicity constitute and represent the transtemporal identity of objects for the system, making information about temporal properties of the world and the system globally available for the control of action, cognition and guided attention. Metzinger does not have any suggestions about the neural correlates of this constraint.

6. Perspectivalness

Phenomenal space is always tied to an individual perspective. We experience things from somewhere and it is impossible to imagine a way of seeing objects that would encompass all of their aspects at once. We are also phenomenologically aware of *being someone*, of being a self in the act of experiencing the world. From a functional point of view, perspectivalness represents the limits of the space that we can causally influence and enables a system to become the object of its own attention and self-directed actions. A phenomenal self is also a necessary precondition for the possession of a strong *epistemic* first-person perspective and for social cognition. The neural correlates of this constraint include the networks involved in the representation of our bodies, the vestibular organ, visceral forms of self-representation and the nuclei involved in the homeostatic regulation of the internal milieu. Damasio (1995, 1999) and the second half of Metzinger (2003) go into the neural correlates of this constraint in detail. A substantial part of Metzinger's work is dedicated to understanding more complex forms of the phenomenal self model, which are not covered in this thesis.

7. Transparency

When we look at the world we do not see a series of neural spikes or streams of data from our optic nerves. We simply see the objects around us and this transparency of our representations is due to the attentional unavailability of earlier processing stages and our inability to introspect the vehicle properties of a representation (we see a red flower, and not the neurons generating the representation of a red flower). This transparency of our mental content forces us into naïve realism about the world: we see the world and not the means by which a representation of the world is constructed by our brains. A world cannot be present without transparency at some point in the system, and so this constraint forms part of the minimal notion of phenomenal experience.[40]

One of the functions of transparency is to remove complex processing from the system and present the final result in the form of naïve realism, which forces the system to take it seriously because it is no longer ‘just a representation’. One of the reasons why the brain is transparent is because it has no senses in it that could present it to itself as an object – it is notably without pain receptors, for example. However, this is not in itself enough for the emergence of transparency, since there is no reason why we should not perceive the incoming data from the retina, for example, as spiking neuron activity instead of light. Transparency is fundamental to phenomenal experience, but unfortunately, as Metzinger notes, “almost nothing is known today about the neural basis of phenomenal transparency.” (Metzinger 2003, p. 178).

8. Offline activation

Phenomenal mental content can be active without sensory input, which enables absent objects to be recollected or dreamt and it can also be used in planning. Offline activation also makes the difference between possibility and reality available to the system, supports social cognition by enabling us to simulate other people’s first person perspectives, and minimises the risks associated with exploratory activity in the world. Offline phenomenal states are characterised by the fact that they are constructed from sequences of non-stimulus correlated states and this lack of covariance with the environment is an essential feature of their causal role. In the human brain the same neural areas are frequently used for perception and for simulating possible perceptual and motor situations, and brain areas that reactivate perceptual areas, such as the hippocampus, are important for this constraint as well.

[40] It is also discussed in Haikonen (2003).

9. Representation of intensities

Phenomenal experience has a quantitative dimension: colours can vary in brightness, some sounds are louder than others and pain has a variety of different levels. This representation of intensities has the function of guiding the attention of the organism to stimuli of maximum interest and it also reflects the intensity of stimuli in the environment. For example, pain directs attention to a damaged area, and the higher the pain the more our attention is focused on that area. The neural correlates of this constraint are likely to be the firing rates of the neurons and the timing of their spikes.

10. "Ultrasmoothness": the homogeneity of simple content

Unlike the real world, simple phenomenal experiences have a structureless density and are homogenous at all levels of magnification. There is no internal structure, no temporal texture and the graininess of neuron firing is invisible at the phenomenal level. This constraint is linked to transparency because the homogenous atomic nature of simple sensory content could be generating the transparency of sensory awareness. One of the functional properties of homogeneity is that it prevents us from introspectively penetrating into the processing stages underlying the activation of sensory content, which is essential for the production of an untranscendable reality (constraint 3) and for reducing the computational load. At the neural level homogeneity might be related to our brains' limited spatial and temporal resolution: we could only perceive the space between the grains of our neural representations if we had a second, more fine grained, neural mechanism Without this, the data that we get is just the data that we get, and we have no access to the spaces or graininess within it.

11. Adaptivity

The adaptivity constraint states that phenomenal mental content must have come about through natural selection. If we want to understand how consciousness could be *acquired* in the course of millions of years of biological evolution, we must assume that it possesses a true teleofunctionalist description. Metzinger claims that this third person objective constraint could affect the ability of artificial systems to experience emotions: “artificial systems as known today do not possess genuinely *embodied goal representations*, because they are not ‘evolutionarily grounded’ – neither their hardware nor their software has developed from an evolutionary optimization process.”(Metzinger 2003, p. 199).

One of the ways in which Metzinger argues for this constraint is using Davidson’s Swampman thought experiment:

> Lightning strikes a dead tree in a swamp while Davidson is standing nearby. His body is reduced to its elements, while entirely by coincidence (and out of different molecules) the tree is turned into his physical replica. This replica, the Swampman, is a physical and functional isomorph of Davidson; it moves thinks, talks, and argues just as the original Donald Davidson did. Obviously, it has precisely the same kind of phenomenal experience as Donald Davidson, because phenomenal content locally supervenes on the brain properties of the replica. On the other hand, the intentional contents of Swampman’s mental state are not the same – for instance, it has many false memories about its own history be they as conscious as they may. The active phenomenal representations in Swampman’s brain would be strongly conscious in terms of the whole set of constraints listed so far, but they would not satisfy the adaptivity constraint, because these states would have the wrong kind of history … It would enjoy a rich, differentiated cognitive version of conscious experience tied to a first person perspective, but it would still be consciousness in a weaker sense, because it does not satisfy the adaptivity constraint holding for ordinary biological consciousness. (Metzinger 2003, p. 206).

The relation of consciousness to its present and past environment is useful for understanding the relationship between consciousness and action (see Section 2.7). However, this constraint has a number of serious problems. To begin with, very little of our bodies is the same as when many of our memories were generated, and so everyone has false or partially false memories about their early history. Secondly, evolutionary arguments linking present states of the organism with a past environment tend to rely on simplistic notions of evolution that ignore the complex feedback loops between the organism and its environment and the constraints of physics and chemistry. Third, many parts of the human body and mind evolved for very different purposes than they presently serve, and so it is senseless to attempt to tie their present meaning to their present or past environment. Finally, I cannot see the benefit in saying that without this constraint the consciousness would be weaker, when the phenomenal experience is said to be the same in both cases.

Within the framework of his constraints, Metzinger defines a minimal notion of conscious experience as follows:

> The phenomenal presence of a world is the activation of a coherent global model of reality (*constraint 3*) within a virtual window of presence (*constraint 2*), both of which are transparent in the sense just introduced (*constraint 7*). The conjunction of satisfied *constraints 2, 3,* and *7* yields the most elementary form of conscious experience conceivable: the presence of a world, of the content of a world-model that cannot be recognized *as* a model by the system generating it within itself. Neither a rich internal structure nor the complex texture of subjective time or perspectivalness exists at this point. All that such a system would experience would be the presence of one unified world, homogenous and frozen into an internal Now, as it were. (Metzinger 2003, p. 169).

This suggests that a robot implementing constraints 2, 3 and 7 should experience a minimal phenomenal state that is without the differentiation, subjectivity and cognitive capacity of biological consciousness. In general Metzinger stresses that consciousness is a matter of degrees and higher degrees of constraint satisfaction will lead to higher degrees of phenomenality in a system.[41]

[41] A critical discussion of Metzinger's work can be found in Legrand (2005). There is also a certain amount of overlap between Metzinger's constraints and Taylor's (2007) discussion of the components of consciousness.

2.7 Consciousness in Action

> Suppose someone were thus to see through the boorish simplicity of this celebrated concept of “free will” and put it out of his head altogether, I beg of him to carry his “enlightenment” a step further, and also put out his head the contrary of this monstrous conception of “free will”: I mean “unfree will,” which amounts to a misuse of cause and effect. … The “unfree will” is mythology; in real life it is only a matter of *strong* and *weak* wills.
>
> Nietzsche (1966, p. 29)

> There is no question that consciousness is important for language, for artistic, mathematical, and scientific reasoning, and for communicating information about ourselves to others.
>
> Koch (2004, p. 234)

2.7.1 Introduction

In this chapter I have kept the physical and phenomenal apart and emphasized the search for correlations between them. One consequence of this approach is that it does not make sense to speak about phenomenal objects carrying out physical functions or physical objects carrying out phenomenal functions - although phenomenal states might be correlated with physical functions. At the current stage of consciousness research, it is only possible to talk about the relationship between phenomenal events and phenomenal actions and between physical events and physical actions - with the hope that we will eventually be able to identify systematic correlations between the two. This strict separation means that a phenomenal event, such as the perception of a red object, will never have to be invoked to *explain* a physical event, such as the nerve signals sent to a muscle.[42]

Although the exact mechanisms of physical action are poorly understood, we can conceive how complete descriptions could be carried out at the physical level that explain how networks of neurons could control a human body driving a car or carry out sophisticated

[42] It must be emphasised that this separation of causal chains does not imply any separation of substances between the phenomenal and the physical.

processes of reasoning. Such descriptions would be framed solely within the language of physics and biology and they would be complete without any mention of consciousness or the phenomenal aspects of imagination or emotion. These physical descriptions would completely explain the transformations of the physical world, but they would leave out the phenomenal aspect of reality, which has been argued to be at least as important as the physical. In order to understand the relationship between consciousness and action at the phenomenal level, we need to use concepts such as red, imagination and emotion to explain how we can make decisions that change the stream of experience. This level of explanation is much less well understood and the final part of this chapter will take a brief look at some empirical observations about consciousness and use them to comprehend how we consciously and unconsciously carry out actions.

This section starts with some phenomenological and experimental observations about consciousness, which demonstrate that our naïve preconceptions about the relationship between consciousness and action are often wrong. Section 2.7.3 then offers a tentative classification of the different aspects of conscious and unconscious action, which is used to develop an interpretation of conscious control and conscious will in sections 2.7.4 and 2.7.5. Finally, Section 2.7.6 takes a look at our experience of conscious will.

2.7.2 Observations about Consciousness and Action

This section offers some general observations about consciousness that will be used to develop and support an interpretation of the relationship between consciousness and action. Since this is a subsidiary theme in this thesis, I will not be examining the large amount of research that has been carried out in detail.[43] Instead, the aim of this section is to offer some broad support for the

[43] Some of the other work in this area is covered by Velmans (1991).

interpretations of conscious control and conscious will that are put forward in sections 2.7.4 and 2.7.5.

Almost all conscious mental states[44] *can become unconscious, but not vice versa*

When we are driving a car we can be conscious of the controls and the road, but we can also process this information unconsciously if we are thinking about other things.[45] However, we cannot make the processes that regulate our heart beat conscious, even if we can exert voluntary control over them with appropriate feedback (Yellin 1986). When we carry out a task unconsciously it is not clear whether its associated mental states are structured in the same way as when the task is carried out consciously.

Unconscious representational mental states can be used to guide action and for limited problem solving

People who suffer from epileptic automatism can perform tasks as complex as diagnosing lung patients without conscious awareness (Cooney 1979). In our everyday lives we execute many complex tasks unconsciously that were learnt when we were carrying them out consciously at an earlier stage in our lives.

Most of the time we are zombies

This point follows from the last. Most of the time we are acting in and responding to the world unconsciously whilst our consciousness is focussed on something completely different. Detailed discussions of the unconscious control of behaviour can be found in Crick and Koch (2003), Koch (2004) and Milner and Goodale (1995).

[44] See sections 4.3.2 and 4.3.3 for definitions of a mental state and a representational mental state that apply to both natural and artificial systems.

[45] This point has been disputed by Searle (1992) and by Dennett (1992, p.137), who claims that it is an example of rolling consciousness with swift memory loss. The unconscious processing of complex information is demonstrated by the work on visual masking, which has shown that unconscious words or digits can be processed at the lexical and semantic levels (Kouider and Dehaene 2007).

Unconscious processing is not good at dealing with new situations

When we encounter a problem with a task that we are executing unconsciously, we often turn our attention to the problem and solve it consciously (Baars 1988, Koch 2004, Underwood 1982). For example, suppose that an amateur carpenter is hammering in a nail whilst thinking about his wife. If the nail bends, he will probably stop thinking about his wife and consciously decide either to extract the nail or to straighten it out in situ. This observation should be qualified with the fact that many complex problems can be solved unconsciously. For example, part of my mind is often working on a problem unconsciously and the solution pops into my head spontaneously without any conscious processing. In my case this only happens for fairly abstract problems, but dancers, for example, might be capable of solving complex motor problems unconsciously.

Consciousness and learning

There seems to be a strong link between conscious information processing and the learning of new skills, which generally have to be carried out consciously before they can be initiated and executed automatically. As Koch explains:

> … a zombie agent can be trained to take over the activities that used to require consciousness. That is, a sequence of sensory-motor actions can be stitched together into elaborate motor programs by means of constant repetition. This occurs when you learn how to ride a bicycle, sail a boat, dance to rock-and-roll, climb a steep wall, or play a musical instrument. During the learning phase, you are exquisitely attentive to the way you position and move your hands, fingers, and feet, you closely follow the teacher's instructions, take account of the environment, and so on. With enough practice, however, these skills become effortless, the motion of your body fluid and fast, with no wasted effort. You carry out the action beyond ego, beyond awareness, without giving any thought as to what has to be done next. It just comes naturally.
>
> Koch (2004, p. 235)

Although there is some evidence that we can learn unconsciously as well as consciously - for example Reber's (1967) work on the learning of artificial grammars - the information that is acquired in these experiments is fairly basic (see Shanks (2005) for an overview and criticisms).

Consciousness is not an all or nothing phenomenon

Each individual has periods of full consciousness and periods of barely conscious experience. When I am late for work and waiting for a train I am extremely conscious of the tension inside me, the situation on the platform, the clock and the possibility that I might get fired. As I travel back from work and drift in and out of sleep on the train, I am barely conscious at all. When we are fully conscious we are maximally conscious of the objects at the centre of our attention and barely aware of objects at the periphery. For example, I am currently most conscious of my laptop in front of me and barely conscious of the street scene outside my window. It is likely that minimally conscious brain-damaged patients experience considerably less and more intermittent consciousness than normal people or patients with locked-in syndrome (Laureys et al. 2004). It also seems likely that some animals are phenomenally conscious to a lesser degree than a fully conscious human – see Crook (1983), Baars (2000) and Seth et al. (2005) for discussions of animal consciousness.

The time scale of consciousness

Libet's (1982) experiments measured the duration of neural activation that is necessary for conscious experience. Using electrodes he stimulated the somatosensory cortex of conscious subjects with trains of pulses of different frequency, duration and intensity, and asked the subjects to report whether they felt any sensations. Libet found that there was a minimum intensity below which no sensation was elicited, no matter how long the pulses were sustained. Furthermore, when a stimulus was above this intensity threshold it could only elicit a conscious sensation if it was continued for around 0.5 seconds - pulse trains shorter than this did not enter conscious awareness. Libet concluded from these experiments that ‘neuronal adequacy’ for conscious sensation is only achieved after half a second of continuous stimulation of the somatosensory cortex. This suggests that it takes approximately this much time to integrate all of our sensory information into a single coherent conscious experience that can be reported. These timing experiments confirm the observation that we are mostly zombies. On time scales of less than half a second we react and respond to stimuli unconsciously and automatically. Over longer time scales we build up conscious models, which set the framework for our unconscious actions.

Consciousnesses and voluntary action

Libet (1985) carried out an experiment to measure the timing relationship between our consciously experienced decisions and the start of the neural events that lead to voluntary action. In this experiment subjects were asked to hold out their hand in front of them and flex their wrist whenever they wanted to. At the same time the subjects watched a rotating spot of light and were asked to report the location of the spot when they became conscious of their decision to act. Libet also recorded the readiness potential, which is a slow negative shift in electric potential that precedes voluntary motor actions and can be detected using electrodes on the scalp. In these experiments, Libet found that the readiness potential preceded the subjects’ experience of a voluntary decision to act, which suggests that the action of flexing the wrist was initiated unconsciously, rather than as the result of a conscious decision.[46]

2.7.3 Conscious and Unconscious Action

These empirical observations about consciousness show that in many circumstances we react automatically to the world or unconsciously initiate actions that we have not consciously decided to do. To clarify the relationship between consciousness and action, the sequence of events that

[46] Libet’s timing experiments have generated a great deal of controversy and there is not space to go into the details here. Many criticisms of the voluntary action experiments can be found in the commentary following Libet (1985) and a fairly comprehensive review can be found in Gomes (1998).

constitutes an action has been broken down into the decision that selects the action, the initiation of the action and the sensory feedback from our bodies and the world as the action is carried out. Each of these stages can be carried out consciously or unconsciously, as shown in Table 2.1.

	Conscious	**Unconscious**
Decision	Using imagination and the emotions I reason about the different courses of action and select one. I might imagine eating at different hours of the day and decide that 1.00 is the optimum time for lunch.	Unconscious decisions are either hardwired into our nervous system - for example, reflexes - or reached through unconscious processes that are largely unknown at the present time.
Initiation	The initiation of the action occurs immediately after a conscious decision to start the action. For example, I decide to go to the shop, and then I get up and go to the shop.	The initiation of the action occurs unconsciously. For example, I am lying in bed and suddenly find myself in the act of getting up.
Execution	We are conscious of the action as we carry it out. For example, as I walk down the street, I look around me at the people and cars without entering into a state of imagination or memory.	We are unconscious when the action is being carried out - for example, cases of epileptic automatism or sleep walking.

Table 2.1. Different aspects of conscious and unconscious actions

These conscious and unconscious aspects of an action can be combined in different ways - for example I might consciously decide to eat my lunch at 1.00, and then make a second conscious decision to carry out the action of eating my lunch. Alternatively, I might have made a conscious decision several years ago to eat my lunch at 1.00 whenever possible, and start preparing my lunch automatically when I glance at the clock without a second conscious decision. Other combinations are also possible – for example, actions can be planned, initiated and executed completely unconsciously. The only intuitively implausible combination is the conscious initiation of an unconsciously chosen action, since it is hard to see how we could decide to execute a decision that we are not aware of.

Two combinations from Table 2.1 will now be used to develop models of conscious control and conscious will. With conscious control, the action is decided consciously, initiated consciously (because the action is immediately carried out) and the person is conscious of sensory feedback from their body and the world as the action is executed. With conscious will, the action is decided consciously, initiated automatically in response to an environmental trigger and executed with the person conscious that they are carrying it out.

2.7.4 Conscious Control

In conscious control actions are decided consciously, initiated immediately and consciously carried out. One of the most plausible models of conscious decision making is offered by Damasio's (1995) somatic marker hypothesis, which gives a good account of the way in which the imagination and emotions work together to reach decisions.[47] Within this framework we make decisions by running through a number of imagined scenarios that trigger bodily feelings associated with them, and eventually settle on the one that feels best. To make this process more efficient there also has to be some mechanism for remembering which scenarios have already been evaluated. This process can be summarised as follows:

1. Generate imaginary scenario that has not been generated before or revisit previous scenario because all other options are exhausted.
2. Evaluate how scenario feels.
3. If scenario feels bad, remember that scenario felt bad and go back to 1.
4. Else if scenario feels right, carry out action immediately.

In *discrete* conscious control we carry out a single action and the conscious imagination of the action *precedes* the action. Since the conscious decision making process is quite slow, this type of conscious control does not happen very often – we believe that conscious control is more common than it is because in many cases the unconscious initiation of an action generates a conscious representation of the action just before it takes place (Libet 1985).[48] However, there

[47] The relationship between the emotions and judgement is discussed by Clore (1992).

[48] See Figure 2.5 for an illustration.

might be circumstances in which we consciously decide to do something and then immediately execute our decision, and a neural model of discrete conscious control has been developed as part of this thesis.

In *continuous* conscious control an action takes place under the guidance of a conscious model that determines the evolution of the action over time. Whilst the decision and the initiation of the action might be automatic, the management of the action is closely linked to consciousness. For example, if my friend asks me what I dreamt last night, then I will probably start my answer automatically without making a conscious decision about whether to reply or not. However, my narration is continuously guided by my conscious memory of the dream, and without this conscious recollection it is hard to see how the dream could be described.[49] Although many of our day to day actions, such as driving or diagnosing lung patients, do not need to be carried out under conscious control, there are numerous daily occasions when we do seem to be consciously controlling continuous actions. Continuous conscious control is likely to be more common than discrete conscious control, but it is often ignored because it is harder to measure experimentally.

2.7.5 Conscious Will

The time scale of discrete conscious control make it implausible that this is the main way in which our conscious decisions influence our actions, and it is much more likely that actions are decided consciously and then initiated unconsciously in response to conscious and unconscious perceptions. In this thesis I will use the term "conscious will" to refer to the process whereby actions are chosen consciously, initiated unconsciously and then consciously carried out.[50] The

[49] Without conscious control, the situation would be a bit like blindsight in which I might be able to guess accurately about the contents of my dream, but would not be able to offer a fluid and natural description.

[50] "Conscious will" could also plausibly be used to refer to actions that are consciously decided, unconsciously initiated and unconsciously executed. Since this does not appear to be a common situation, it has been set aside in this thesis because it would serve only to complicate the discussion.

decision process in conscious will is carried out in the same way as conscious control, but in conscious will, we *remember* the decision and execute it *automatically* in response to environmental or internal triggers (perhaps with the possibility of veto - see Libet (1985, 1999)). For example, at midnight I decide to get up at eight tomorrow morning and set my alarm clock; when the alarm goes off I lie in bed feeling reluctant and then suddenly find myself in the act of getting up. The stages in this model of conscious will can be summarized as follows:

1. Generate imaginary scenario that has not been generated before or revisit previous scenario because all other options are exhausted.
2. Evaluate how scenario feels.
3. If scenario feels bad, remember that scenario felt bad and go back to 1.
4. Else if scenario feels right, remember future action and an associational trigger that will release the action.
5. Continue acting in world.
6. When associational trigger is reached, perform action unconsciously.

This distinction between conscious decisions and automatic execution provides a way out of the problems thrown up by Libet's (1985) timing experiments on the will. Within the framework that I am presenting here, the subject's conscious decision to flex their wrist was taken when they decided to participate in the experiment minutes or hours before the actual action (a fact highlighted by some of the commentators following Libet's (1985) paper). As they randomly flexed their wrist they were not making conscious decisions, but automatically executing a decision that had already been made, and so it is not surprising that the readiness potential preceded the subjects' awareness of their decision to act. To test the timing of *conscious* decisions, the experiment would have to present a number of options to the subjects that would require internal simulation to choose an appropriate action. The timing relationships would then be between the internal modelling of the situation, the activation or simulation of different body states, the memorization of the conscious decision and the onset of the readiness potential that precedes the execution of the action. It would be very surprising if the readiness potential preceded all of these events, which are likely to take at least one or two seconds. This interpretation of Libet is similar to that put forward by Aleksander et al. (2005).[51]

2.7.6 The Experience of Conscious Will

Our feeling of having willed something could be interpreted as the best evidence that we have for a link between consciousness and action. However, Wegner and Wheatley (1999) and Wegner (2002, 2003, 2004) claim that our experience of willing is actually an *inference* that we make about the relationship between a thought and a subsequent action, and we do not directly experience an actual causal process. This claim is supported by Wegner and Wheatley's (1999) *I Spy* experiment in which two people used a board mounted on top of a mouse to move a cursor to point to one of fifty tiny toy pictures taken from an *I Spy* book. One of the people was a genuine participant who heard words on his or her headset and was cued by music to bring the mouse to a stop. After each trial this participant was asked to rate each stop for the degree of intentionality that they felt when they made it. The second person in the experiment was a confederate pretending to be a participant, who was given instructions to stop the mouse on a particular picture or to allow the participant to stop the cursor wherever he or she liked. On some of the trials the participant heard words that matched the forced stop on a particular picture – for example, they might have heard the word 'swan' prior to the confederate bringing the cursor to rest on the swan.

[51] There was not space in this thesis to examine how this concept of will relates to freedom of the will. The question about the freedom of the will is a complex topic that combines a number of conflicting intuitions (Honderich 1993, Double 1991). However, it is worth pointing out that this model of conscious will is broadly compatible with Hodgson's (2005) basic requirements for any account of free will and it is aligned with compatibilist accounts that balance psychological freedom with metaphysical determinism, such as Gomes (1999) and Clark (1999). It also largely agrees with Kane's (1996) libertarian concept of free will as "*the power of agents to be the ultimate creators (or originators) and sustainers of their own ends or purposes*" (p. 4).

This experiment showed that being cued with a word did not lead the participants to stop more frequently on the associated pictures. However, the participants did claim to have a higher amount of intentionality when they were cued 5 or 1 seconds before being forced to stop on a particular picture, which did not occur when they were cued 30 seconds before or 1 second after the forced stop. In other words, participants claimed that they had intended to stop on a picture associated with a word that they had heard 5 or 1 seconds before, even though they had no choice about where to stop and would not have stopped on the picture if the confederate had not moved the cursor to this position. This suggests that the participant's experience of will depended on an association between the cued words and actions, rather than on any actual causal link between their thoughts and actions. According to Wegner and Wheatley (1999), this experiment shows that the participant's experience of conscious will arises through an inferential process in which they reason about their actions and conclude whether they did them or not.

Three of the most important factors in this inferential process are the priority of the thought before the action, the consistency of the thought with the action and the exclusivity of the thought relative to the action. If we think of an action a short time before it happens, if our thought matches the action, and if no other causes can be put forward to explain the action, then we experience a feeling of intentionality relative to the action: an experience that we willed the action. Wegner (2003) supports his argument with other examples in which there is a disparity between the feeling of conscious will and the actual volition, such as alien hand syndrome, in which the person chooses the actions of the hand, but does not believe himself or herself to have willed them (Geschwind et al., 1995), schizophrenics' attribution of other people's actions to themselves (Daprati et al., 1997), and action projection in which a person performs a voluntary action that they attribute to someone else (Wegner, 2002).[52]

[52] Although Wegner and Wheatley (1999) and Wegner (2004) cite these as examples of wilful action, within the framework presented in Section 2.7.3, these are examples of unconscious decisions initiated unconsciously, which is quite different from the model of conscious will put forward in Section 2.7.5.

Although Wegner (2003) claims that the feeling of conscious will is an *illusion* because it does not reflect the underlying causal mechanisms, this should not be interpreted as the claim that there is *no* link between consciousness and action. Wegner's work convincingly demonstrates that our inference about our causal powers is fallible, but it does not show that it is always incorrect - a point that is made explicitly by Wegner:

> Does all this mean that conscious thought does not cause action? It does not mean this at all. The task of determining the causal relations between conscious representations and actions is a matter of inspection through scientific inquiry, and reliable connections between conscious thought and action can potentially be discerned by this process. The point made here is that the mind's own system for computing these relations provides the person with an experience of conscious will that is no more than a rough-and-ready guide to such causation, one that can be misled by any number of circumstances that render invalid inferences…
>
> (Wegner, 2003, p. 68)

Some people, such as Claxton (1999), have attempted to use arguments similar to Wegner's to virtually eliminate the relationship between conscious will and action. The problem with this position is that a complete break between consciousness and action makes consciousness epiphenomenal and eliminates any sense in which we can claim to *speak* about consciousness - a position that was discussed in detail in Section 2.4.3.

Wegner's account of our experience of conscious will fits in naturally with the models of conscious control and conscious will that were put forward in sections 2.7.4 and 2.7.5. In both conscious control and conscious will, the imagination and emotion that are involved in the decision making process have a completely different phenomenology from the feeling of intention, and it is perfectly plausible that our experience of will is the outcome of an inferential process that takes place after the action has been executed. This is particularly apparent in the model of conscious will, where there might be a delay of years between a conscious decision and the unconscious initiation of the action. In this case it is hardly plausible to claim that we experience the will in operation, and much more likely we find ourselves engaged in an action and then experience a feeling of conscious will when we remember the earlier decision that led us to act in this way.

Although a connection between consciousness and action is essential to any theory of consciousness that is not epiphenomenal, it is important to remember that our inferences about this link are fallible and get the connection wrong in many cases. This is particularly apparent when the unconscious initiation of an action presents an image of the action in consciousness just before it is carried out, as shown in Figure 2.5.

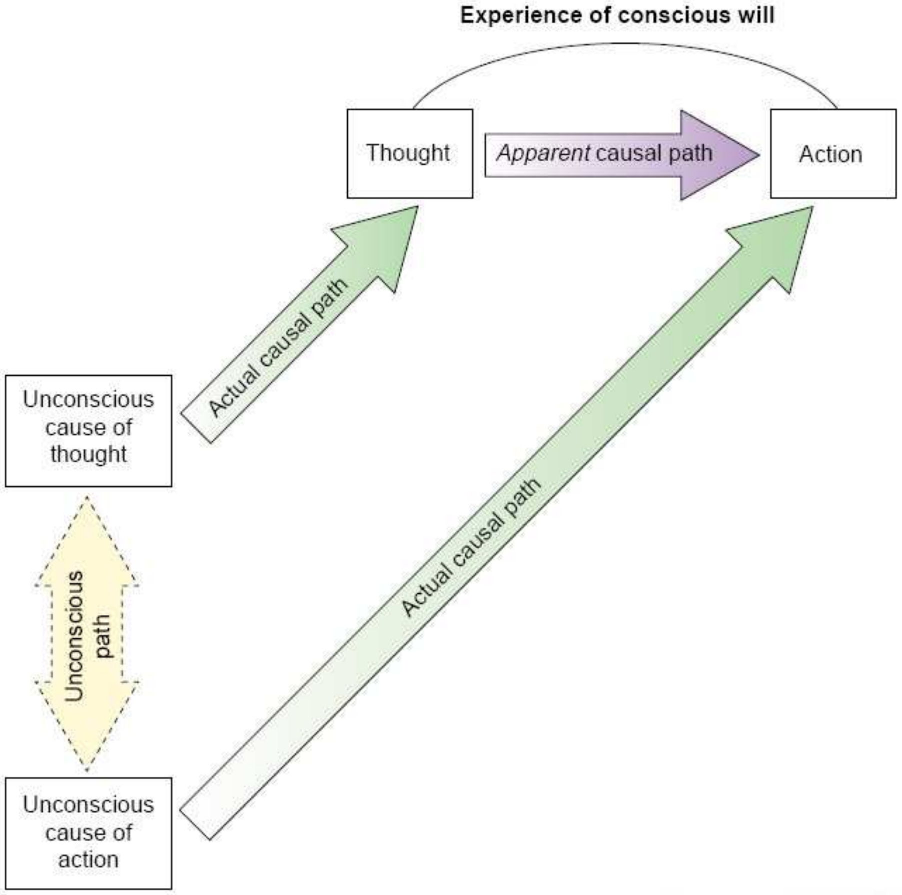


Figure 2.5. Unconscious cause of thought and action. Although the thought appears just before the action, both thought and action have the same unconscious cause. Reproduced from Wegner (2003).

Although the appearance of a thought prior to the action might enable the organism to veto the action (Libet, 1999), Libet's (1985) experiments have shown that the thought often occurs after the action has been unconsciously initiated, when there is only an apparent causal link between the thought and the action. However, with conscious control and conscious will, it is the timing of the decision about the action that is important and detailed studies are needed to explore the timing relationship between conscious decisions and the conscious or unconscious initiation of actions.

2.8 Conclusions

This chapter has set out an interpretation of consciousness that will be applied in the rest of this thesis. A distinction between the phenomenal and the physical was used to define consciousness and to reject the hard problem of consciousness in favour of the real problem of consciousness, which can only be addressed through work on the correlates of consciousness. This led to the distinction between type I behaviour-neutral correlates of consciousness, which cannot be identified, and type II correlates of consciousness, which can be separated out through their influence on behaviour. This chapter then outlined three type II theories of consciousness and models of conscious control and conscious will.

The approach to consciousness in this chapter will be used to develop a new methodology for describing the consciousness of artificial systems in Chapter 4. The next chapter summarizes some of the previous work that has been carried out in machine consciousness.

3. Machine Consciousness[1]

3.1 Introduction

This chapter tackles some of the theoretical issues surrounding machine consciousness and reviews some of the previous work in this field.[2] Machine consciousness is currently a heterogeneous research topic that includes a number of different research programs, with some people working on the behaviours associated with consciousness, some people modelling the cognitive characteristics of consciousness and some people interested in creating phenomenal states in machines. To make sense of this diverse subject, the first part of this chapter identifies four different areas of machine consciousness research:

MC1. Machines with the external behaviour associated with consciousness.

MC2. Machines with the cognitive characteristics associated with consciousness.

MC3. Machines with an architecture that is claimed to be a cause or correlate of human consciousness.

MC4. Phenomenally conscious machines.

This classification starts with systems that replicate aspects of human[3] behaviour and moves on to systems that are attempting to create real artificial consciousness. Although there is a certain amount of overlap between these categories, they are a useful way of understanding work on machine consciousness and will be used to identify different aspects of it throughout this chapter.

[1] An earlier version of this chapter was published as Gamez (2007a).

[2] I will be using the term "machine consciousness" to refer to this field, although "artificial consciousness" and occasionally "digital sentience" (Anon, 2006) have also been used to describe it. Each of these terms has their own merits, but the growing number of meetings dedicated to "machine consciousness" suggests that this is likely to become the standard name for the field.

[3] In this chapter discussion generally focuses on *human* behaviour, cognitive characteristics and architectures associated with consciousness because humans are generally taken as paradigmatic examples of conscious entities. However, any work on the replication of animal behaviour, cognitive characteristics and architectures associated with consciousness would also be part of machine consciousness research.

The first application of these categories is to clarify the relationship between machine consciousness and other fields. The interdisciplinary nature of machine consciousness is often a source of confusion because it takes inspiration from philosophy, psychology, and neuroscience and shares many of the objectives of strong AI and artificial general intelligence. These relationships between machine consciousness and other fields become much clearer once machine consciousness has been separated into MC1-4. For example, artificial general intelligence has a certain amount in common with MC1, but little overlap with MC2-4. On the other hand, neuroscientists, such as Dehaene et al. (1998, 2003, 2005), are creating computer models of the neural correlates of consciousness (MC3), but have little interest in MC1, MC2 or MC4. This classification is also very useful for dealing with some of the criticisms that have been raised against machine consciousness, which often only apply to one or two aspects of its research. For example Dreyfus' (1992) claims about what computers still can't do mainly apply to MC1 and many of them could be answered by work on MC2 and MC3. On the other hand, Searle's Chinese Room argument is directed against MC4 and leaves work on MC1-3 unaffected.

The second half of this chapter surveys some of the research projects that are taking place in machine consciousness and uses MC1-4 to unpack the different objectives of this work. This research includes theoretical approaches, models of consciousness, and systems designed to actually be phenomenally conscious. The last two sections cover some of the ethical issues linked to machine consciousness and explore its potential benefits.

3.2 Areas of Machine Consciousness Research

Machine consciousness is not a unified field with a set of clearly defined goals. At present a heterogeneous network of researchers are working on different aspects of the problem, and this can often make it difficult to understand how everything fits together. This section clarifies machine consciousness research by dividing it into four different areas.

3.2.1 Machines with the External Behaviour Associated with Consciousness (MC1)

A lot of our waking behaviours are carried out unconsciously in response to stimulation from the environment. For example, the detailed muscle contractions involved in walking are rarely under conscious control and we can perform relatively complex behaviours, such as driving home from work, with our attention on other things.[4] Other examples of unconscious behaviour include patients in a persistent vegetative state, who commonly produce stereotyped responses to external stimuli, such as crying, grimacing or occasional vocalisation (Laureys et al., 2004), blindsight patients who have a limited ability to respond visually to objects that they cannot consciously see, and actions carried out under the influence of an epileptic seizure. A dramatic example of the latter is given by Damasio (1999):

> Suddenly the man stopped, in midsentence, and his face lost animation; his mouth froze, still open, and his eyes became vacuously fixed on some point on the wall behind me. For a few seconds he remained motionless. I spoke his name but there was no reply. Then he began to move a little, he smacked his lips, his eyes shifted to the table between us, he seemed to see a cup of coffee and a small metal vase of flowers; he must have because he picked up the cup and drank from it. I spoke to him again and again he did not reply. He touched the vase. I asked him what was going on and he did not reply, his face had no expression. … Now he turned around and walked slowly to the door. I got up and called him again. He stopped, he looked at me, and some expression returned to his face – he looked perplexed. I called him again and he said, "What?"
>
> (Damasio, 1999, p. 6).

These examples show that a limited amount of behaviour can be carried out unconsciously by humans. However, the stereotypical nature of this behaviour suggests that

[4] For another view on this issue see Franklin et. al. (2005).

more complex activities, such as interpersonal dialogue, can only be carried out consciously and many new behaviours have to be learnt when consciousness is present. This leads to a distinction between human behaviours associated with consciousness and those carried out automatically without consciousness.[5]

One research area in machine consciousness is on systems that replicate conscious human behaviour. Although this type of research can be based on cognitive models (MC2) or on an architecture associated with consciousness (MC3), this is not necessary to work on MC1, which could also use a large lookup table or first-order logic to generate the behaviour. Although certain external behaviours are associated with phenomenal states *in humans*, this is not necessarily important to people working on MC1, since it has often been claimed that a zombie robot could replicate conscious human behaviour without experiencing phenomenal states. However, the boundary between MC1 and MC4 might start to become blurred when robots can reproduce most human behaviours. In this case, Harnad (2003) argues that we will have to attribute phenomenal experiences to MC1 machines because our only guide to phenomenal states is a system's external behaviour. Supporting this point, Moor (1988) suggests that we will need to ascribe qualia to such systems in order to understand them.

Any attempt to pass the Turing Test has to replicate behaviours that are carried out consciously in humans, and so people working on this challenge[6] can be considered to be part of MC1. Research on artificial general intelligence (see Section 3.3.2) also falls within this area.

3.2.2 Machines with the Cognitive Characteristics Associated with Consciousness (MC2)

A number of connections have been made between consciousness and cognitive characteristics, such as imagination, emotions and a self - for example, Aleksander's (2005) axioms and

[5] See Section 2.7.2 for a more detailed discussion of this issue.

[6] For example, the contestants in the annual Loebner prize: http://www.loebner.net/Prizef/loebner-prize.html.

Metzinger's (2003) constraints (see sections 2.6.3 and 2.6.4). Detailed descriptions of conscious states have also been put forward by phenomenologists, such as Husserl (1964), Heidegger (1995a) and Merleau-Ponty (1995).

The modelling of the cognitive characteristics associated with consciousness has been a strong theme in machine consciousness, where it has been carried out in a wide variety of ways, ranging from simple computer programs to systems based on simulated neurons. Cognitive characteristics that are frequently covered by this work include imagination, emotions, and internal models of the system's body and environment. In some cases the modelling of cognitive states has aimed at more realistic conscious behaviour (MC1) or used an architecture associated with consciousness (MC3), but MC2 systems can also be created without MC1 or MC3 – for example, a computer model of emotions or imagination that does not have external behaviour. There is also no necessary connection between MC2 and MC4 since the simulation of fear, for example, can be very different from real phenomenological fear - just as the price of gold can be modelled in a computer without the program, CPU or RAM containing any real gold.

3.2.3 Machines with an Architecture that is Claimed to be a Cause or Correlate of Human Consciousness (MC3)

Many people are working on the simulation of architectures that have been linked to human consciousness, such as Baars' (1988) global workspace. This type of research often arises from the desire to model and test neural or cognitive theories of consciousness and it is one of the most characteristic areas of machine consciousness.

Work on MC3 overlaps with MC2 and MC1 when systems based on an architecture associated with consciousness are used to produce the cognitive characteristics of consciousness or conscious behaviour. It could also overlap with MC4 if it was thought that an implementation of an architecture associated with consciousness would be capable of phenomenal states.

However, simulating a ‘conscious’ architecture in a machine may not be enough for the machine to actually become conscious.

3.2.4 Phenomenally Conscious Machines (MC4)

The first three approaches to machine consciousness are all relatively uncontroversial, since they are modelling phenomena linked to consciousness without any claims about real phenomenal states. The fourth area of machine consciousness is more philosophically problematic, since it is concerned with machines that have real phenomenal experiences - machines that are not just tools in consciousness research, but actually conscious themselves.

As has already been indicated, this approach has some overlap with MC1-3, since in some cases it might be hypothesized that the reproduction of human behaviour, cognitive states, or internal architecture would lead to real phenomenal experiences. On the other hand, MC4 might be achievable independently of other approaches to machine consciousness. For example, it might be possible to create a system based on biological neurons that was capable of phenomenal states, but lacked the architecture of human consciousness and any of its associated cognitive states or behaviours.[7] Furthermore, it has been claimed by Chalmers (1996) that systems as simple as thermostats may have basic conscious states. If this is correct, the presence of phenomenal states in a machine will be largely independent of the higher level functions that it is carrying out.

Systems with real consciousness cannot be developed without methods for measuring phenomenal states, and so there is a close relationship between MC4 and synthetic phenomenology (see Chapter 4). The production of machines with real feelings also raises ethical questions, which are covered in Section 3.6.

[7] DeMarse et al.’s (2001) neural animat might be a system of this kind.

3.3 Relationship between Machine Consciousness and Other Areas

3.3.1 Strong and Weak AI

Work on artificial intelligence is often classified using Searle's (1980) distinction between strong and weak AI:

> According to weak AI, the principal value of the computer in the study of the mind is that it gives us a very powerful tool For example, it enables us to formulate and test hypotheses in a more rigorous and precise fashion. But according to strong AI, the computer is not merely a tool in the study of the mind: rather, the appropriately programmed computer really *is* a mind, in the sense that computers given the right programs can be literally said to *understand* and have other cognitive states. In strong AI, because the programmed computer has cognitive states, the programs are not mere tools that enable us to test psychological explanations; rather, the programs are themselves the explanations.
>
> (Searle, 1980, p. 417)

According to Searle, strong AI is the attempt to create something that is a mind in the sense that I am a mind, whereas weak AI is the process of modelling the mind using human-interpretable symbols that work in the same way a mind works. This distinction is similar to that made by Franklin (2003) between phenomenal and functional consciousness and it also relates to the difference between the easy and the hard problems of consciousness (Chalmers, 1996). In all of these cases, a contrast is set up between the external manifestations of a mind and a real mind, which suggests a reasonably clear mapping between MC4 and strong AI, with MC1-3 being examples of weak AI in Searle's sense.

The problem with strict identity between MC4 and strong AI is that the notion of mind can be separated from phenomenal consciousness - suggesting that computers can really *be* minds without being conscious in the sense of MC4. For example, Carruthers claims that "The view that we have, or can have, notions of mentality which do not presuppose consciousness is now widely accepted" (Carruthers 2000, p. xviii), and so it may be possible to build a strong AI machine that is not conscious in the sense of MC4. A robot that grounded its symbols in sensory data might be one example of a non-phenomenal mind that literally understands and has other cognitive states.

3.3.2 Artificial General Intelligence

Artificial general intelligence (AGI) is another area within AI that has similarities with machine consciousness. The aim of AGI is to replicate human intelligence completely and it is sometimes contrasted with a second interpretation of weak AI as the solving of computer science problems within a limited domain – for example, pattern recognition or chess playing.[8] AGI has a certain amount of overlap with MC1, with the difference that MC1 is focused on conscious human behaviour, whereas AGI is attempting to reproduce all human behaviours linked with intelligence. Which of these is the larger category depends to some extent on the definition of intelligence. Some behaviours linked to consciousness may be excluded by AGI's definition of intelligence, but it is also possible that AGI could use a broad interpretation of intelligence that includes all MC1 behaviours.[9]

How AGI could be implemented is a completely open question and some AGI systems may be produced by simulating the cognitive states associated with consciousness (MC2) or by copying an architecture linked to consciousness (MC3). It is also possible that AGI systems will have phenomenal states (MC4). The interpretation of weak AI as the solving of computer science problems within a limited domain does not have much in common with any of the definitions of machine consciousness.

[8] This interpretation of weak AI is also referred to as "narrow AI".

[9] More information about AGI can be found in Goertzel and Pennachin (2007) and in the proceedings from the 2006 AGIRI Workshop: http://www.agiri.org/forum/index.php?showtopic=23.

3.3.3 Psychology, Neuroscience and Philosophy

The empirical work carried out by experimental psychology and neuroscience often forms a starting point for the modelling work in machine consciousness, but there is generally little overlap between them. However, there are some exceptions to this trend, such as the research carried out by Krichmar and Edelman (2006) using the Darwin series of robots and Dehaene et al.'s (1998, 2003, 2005) modelling of neurons to test theories about attention and consciousness. Dehaene et al.'s work clearly fits within MC3 and will be covered in Section 3.5.6. On the other hand, although Krichmar and Edelman are modelling a reentrant neural architecture associated with consciousness, they do not explicitly link their Darwin work to consciousness, and so I have not included it in this chapter.[10]

Amongst the other disciplines, cognitive psychology and connectionism also build computer models of cognition, which leads to a substantial amount of overlap with MC2. However, this work is more general than that carried out by machine consciousness because it covers types of cognition that are not associated with conscious states. Although philosophy and AI have historically been linked through their common use of logic, this connection has declined in recent years with the atrophy of logic in both subject areas. The emergence of machine consciousness has changed this relationship and philosophy now provides a theoretical framework for MC1-4 and tackles ethical issues.

3.4 Criticisms of Machine Consciousness

3.4.1 The Chinese Room

The Chinese Room thought experiment consists of a person in a room who receives Chinese characters, processes them according to a set of rules and passes the result back out without

[10] Krichmar and Edelman's work is covered in the discussion of research on neural networks in Section 5.6.

understanding what the characters mean. This processing of characters could be used to create the external behaviour associated with consciousness, to simulate the cognitive characteristics of consciousness or to model a conscious architecture. However, Searle (1980) argues that in no case would the person processing characters in the room understand what is going on or have intentional states directed towards the objects represented by the Chinese characters. Although the Chinese Room might be able to *model* a mind successfully, it will never literally *be* a mind in the sense intended by MC4.

One response to this argument is based on the notion of symbol grounding. If the characters in the Chinese room could be linked to non-symbolic representations, such as images or sounds, then the system would understand what the symbols mean and have intentional states directed towards this meaning. According to Harnad "Symbolic representations must be grounded bottom-up in nonsymbolic representations of two kinds: (1)'iconic representations', which are analogs of the proximal sensory projections of distal objects and events, and (2)'categorical representations', which are learned and innate feature-detectors that pick out the invariant features of object and event categories from their sensory projections." (Harnad 1990, p. 335). Neural models have also been cited as a way of grounding higher level symbolic representations by connecting them to sensory inputs (Haikonen, 2003), and if the Chinese Room can be grounded effectively in some kind of non-symbolic lower level, then it can be said to understand the characters that it is manipulating.

A second reason why the Chinese Room argument is not fatal to MC4 is that brains and computers are both physical systems assembled from protons, neutrons and flows of electrons. Searle (2002) is happy to claim that consciousness is a causal outcome of the physical brain and so the question becomes whether the physical computer and the physical brain are different in a way that is relevant to consciousness. This can only be answered after we have done a lot more research on the correlates of consciousness. Until this work has been carried out, the Chinese

Room argument does not offer any *a priori* reason why the arrangement of protons, neutrons and electrons in a physical computer is less capable of consciousness than the arrangement of protons, neutrons and electrons in a physical brain.

3.4.2 Consciousness is Non-algorithmic

Machine consciousness has also been criticised by Penrose (1990, 1995), who claims that the processing of an algorithm is not enough to evoke phenomenal awareness because subtle and largely unknown physical principles are needed to perform the non-computational actions that lie at the root of consciousness: “Electronic computers have their undoubted importance in clarifying many of the issues that relate to mental phenomena (perhaps, to a large extent, by teaching us what genuine mental phenomena are *not*) … Computers, we conclude, do something very different from what *we* are doing when we bring our awareness to bear upon some problem.” (Penrose 1995, p. 393). If consciousness does something that ‘mere’ computation cannot, then MC1-3 cannot be simulated by a computer and MC4 cannot be created in a computer.

The most straightforward response to Penrose is to reject his theory of consciousness, which is far from convincing and has been heavily criticised by Grush and Churchland (1995) among others. However, even if Penrose’s theories about consciousness are correct, MC1-4 would continue to be viable research projects if they could develop an approach to machine consciousness that fits within his framework:

> I am by no means arguing that it would be necessarily impossible to build a genuinely intelligent *device*, so long as such a device were not a ‘machine’ in the specific sense of being computationally controlled. Instead it would have to incorporate the same kind of physical action that is responsible for evoking our own awareness. Since we do not yet have any physical theory of that action, it is certainly premature to speculate on when or whether such a putative device might be constructed. Nevertheless, its construction can still be contemplated

> within the viewpoint … that I am espousing …, which allows that mentality can eventually be understood in scientific though non-computational terms.
>
> (Penrose 1995, p. 393).

If Penrose is right, we may not be able to use algorithms to construct MC1-4 machines, but it might be possible to create some kind of quantum computer, which incorporates the physical mechanisms that are linked by Penrose to human consciousness.

3.4.3 What Computers Still Can't Do

Dreyfus (1992) put forward a number of arguments against artificial intelligence projects that attempted to reduce human intelligence to a large number of rules.[11] According to Dreyfus, this can never work because human intelligence depends on skills, a body, emotions, imagination and other attributes that cannot be encoded into long lists of facts. Dreyfus also criticises some of the approaches to AI that have emerged as alternatives to fact-based systems, such as interactive AI, neural networks with supervised learning and reinforcement learning.

These arguments affect work on the development of systems that are as intelligent as humans in real world situations. However, there is no reason why MC1-4 could not be pursued in a more limited way independently of this objective. For example, some of the behaviours that require consciousness in humans (MC1) could be created in a simple and non-general way, and imagination and emotion could be simulated (MC2) without the expectation that they will be able to work as effectively as human cognitive processes.[12] The modelling of architectures associated with consciousness (MC3) is largely independent of Dreyfus' objections and phenomenal consciousness (MC4) may be possible without the generality and complexity of human behaviour.

[11] Lenat's Cyc is a good example of this kind of system (Matuszek et al. 2006). More recently Bringsjord has been using logic-based artificial intelligence to control a four year old child in Second Life: http://www.sciencedaily.com/releases/2008/03/080310112704.htm.

[12] This is the case with the simple Khepera models described in Section 3.5.5.

It can also be argued that the work being carried out on imagination, emotions and embodiment in machine consciousness addresses some of the areas that Dreyfus claims to be lacking in current artificial intelligence. Furthermore, the human brain is itself a machine, and so biologically-inspired research on machine consciousness might eventually be able to solve Dreyfus' problems. However, all of this work is still at an early stage and it is far from clear whether MC1-4 devices will ever become intelligent enough to act and learn like humans in the real world.

3.5 Research on Machine Consciousness

The last few sections have outlined the different areas of machine consciousness, its relationship to other fields and the criticisms that could be raised against it. I will now move on to some of the research that has been carried out on MC1-4. In order to focus on the unique aspects of machine consciousness, this chapter will not include the large number of simulations that have been done as part of AI, connectionism and brain modelling, and theoretical work on consciousness will only be included if it deals explicitly with MC1-4. Although some of the projects have been organised under sub-headings to highlight general areas of machine consciousness research, it should be borne in mind that some systems could have been included in several sections – for example, IDA has a global workspace architecture and is also a software agent.

3.5.1 Aleksander's Kernel Architecture

Aleksander (2005) and Aleksander and Morton (2007c) have developed a kernel architecture that includes all five of Aleksander's axioms (see Section 2.6.3). This includes a perceptual module that depicts sensory input, a memory module that implements non-perceptual thought for planning and recall of experience, an emotion module that evaluates the 'thoughts' in the memory module, and an action module that causes the best plan to be carried out (see Figure 3.1).

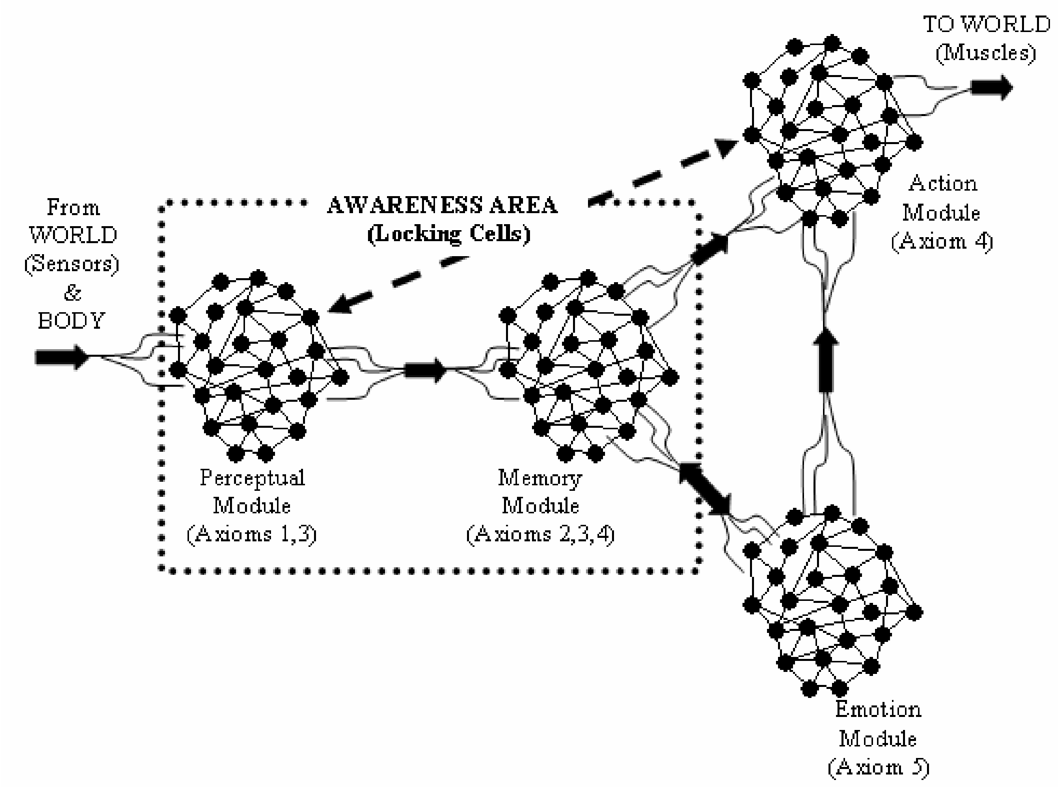


Figure 3.1. Aleksander's kernel architecture[13]

Aleksander and Morton (2007b) have built a number of brain-inspired implementations of this kernel architecture (MC3) using the Neural Representation Modeller (NRM) software,[14] which uses weightless neurons containing lookup tables that match input patterns to an output response. During training, these neurons store the link between each input pattern and the specified output; during testing, the neurons produce the output of the closest match to a known input pattern or a random sequence of 1s and 0s when there is more than one match. These neurons are assembled into large recurrent networks and trained using the graphical and scripting abilities of NRM.

[13] This figure is reproduced from Aleksander (2007c).

[14] 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.

These brain-inspired simulations of the kernel architecture are minimal implementations of Aleksander’s five axioms, and so they have the potential for phenomenal consciousness (MC4) according to the axiomatic theory. Full details about how the kernel architecture implements the axioms can be found in Aleksander and Morton (2007c).

3.5.2 Internal Modelling with SIMNOS and CRONOS

The CRONOS project and its main components were outlined in Section 1.2 and this thesis covers one of the approaches to machine consciousness that was developed as part of this project. A different approach to machine consciousness in the CRONOS project was developed by Holland, who claims that internal models play an important role in our conscious cognitive states (MC2) and may be a cause or correlate of consciousness in humans (MC4) (Holland and Goodman 2003, Holland 2007).[15] Holland is particularly interested in internal models that include the agent's body and its relationship to the environment and the extent to which the connection between this type of internal model and consciousness may be supported by Metzinger's (2003) discussion of the phenomenal self model and Damasio's (1999) analysis of the origins of consciousness. To test these theories about internal modelling, SIMNOS is being employed as an internal model of CRONOS and the computational technique of simultaneous localization and mapping (SLAM) will be applied to the visual stream from CRONOS's eye to obtain information about the environment and the robot's movements in relation to it, which will be used to continually update SIMNOS and its virtual environment. The internal model will then be employed offline to 'imagine' potential actions with SIMNOS before the selected action is carried out by CRONOS.

[15] Some of the other work carried out by Holland on the link between internal models and consciousness is described in Section 3.5.5.

3.5.3 Cog

Cog was a humanoid robot developed by Brooks et al. (1998) that consisted of a torso, head and arms under the control of a heterogeneous network of programs written in L, a multithreaded version of Lisp (see Figure 3.2). Cog was equipped with four cameras providing stereo foveated vision, microphones on each side of its head, and a number of piezoelectric touch sensors. This robot also had a simple emotional system to guide learning and a number of number of hard wired 'innate' reflexes, which formed a starting point for the acquisition of more complex behaviours. The processors controlling Cog were organised into a control hierarchy, ranging from small microcontrollers for joint-level control to digital signal processor networks for audio and visual processing.

The development work on Cog was organised as a number of semi-independent projects that focused on different aspects of human cognition and behaviour, such as joint attention and theory of mind, social interaction, dynamic human-like arm motion and multi-modal coordination. Although Brooks et al. (1998) do not explicitly situate this work within machine consciousness, Dennett (1997) put forward a good case for Cog having the potential to develop phenomenal states (MC4). Some of the behaviours of Cog, such as joint attention and theory of mind, could also be said to be associated with consciousness in the sense of MC1, and Cog's emotional system is a cognitive characteristic associated with consciousness (MC2).

Although Cog could display many individual human behaviours, when the systems were active together, competition for actuators and unintended couplings through the world led to incoherence and interference. This made it difficult for Cog to achieve higher cognitive functions and coherent global behaviour, which may be one of the reasons why this project has now effectively stopped.

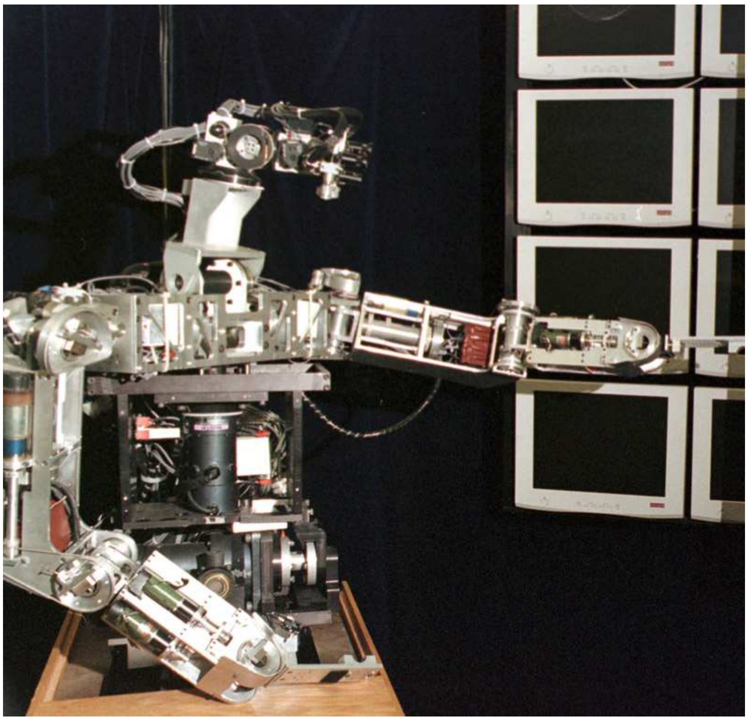
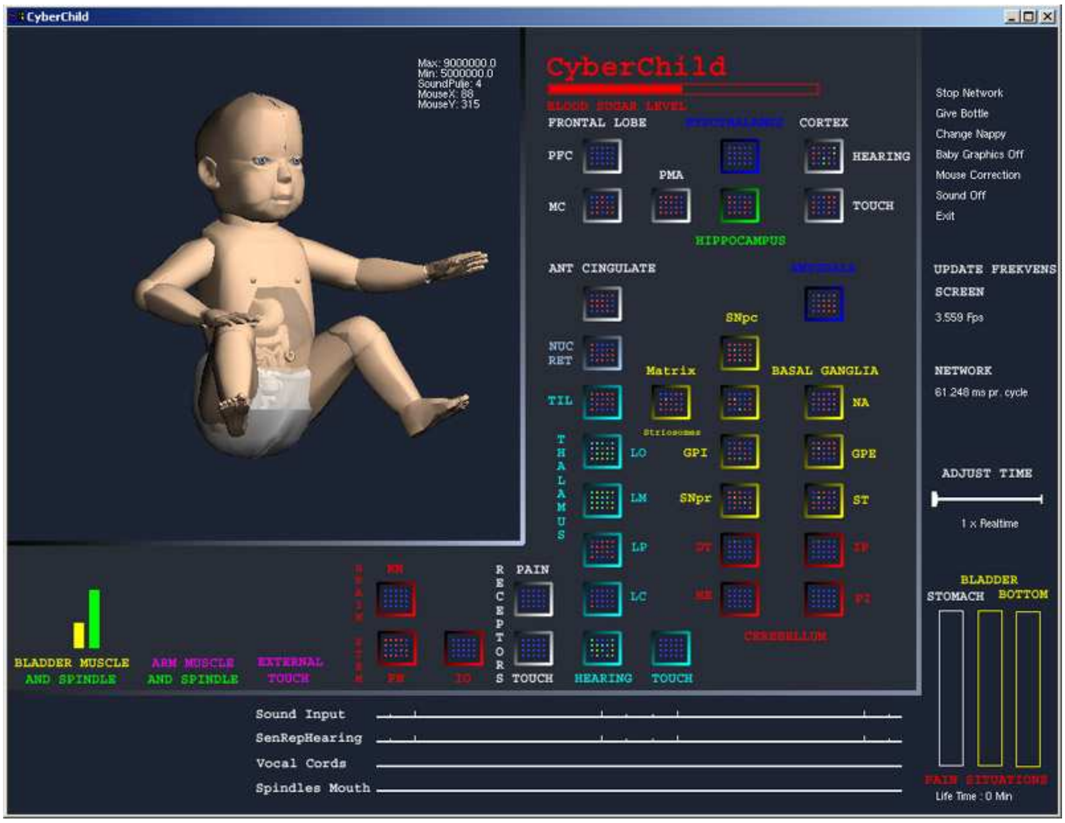


Figure 3.2. Cog robot[16]

3.5.4 CyberChild

CyberChild is a simulated infant controlled by a biologically-inspired neural system based on Cotterill's (2000) theory of consciousness. This virtual infant. (see Figure 3.3) has rudimentary muscles controlling the voice and limbs, a stomach, a bladder, pain receptors, touch receptors, sound receptors and muscle spindles. It also has a blood glucose measurement, which is depleted by energy expenditure and increased by consuming milk. As the consumed milk is metabolised, it is converted into simulated urine, which accumulates in the infant's bladder and increases its discomfort level. The simulated infant is deemed to have died when its blood glucose level reaches zero. CyberChild also has drives that direct it towards acquiring sustenance and avoiding discomfort and it is able to raise a feeding bottle to its mouth and control urination by tensing its bladder muscle. However, these mechanisms are not enough on their own to ensure the survival of the simulated infant, which ultimately depends on its ability to communicate its state to a human operator.

[16] Photograph taken by Donna Coveney.



Max: 9000000.0
Min: 5000000.0
SoundPulse: 4
MouseX: 88
MouseY: 315

3.559 Fps

61.248 ms pr. cycle

1 x Realtime

Life Time : 0 Min

Figure 3.3. CyberChild

CyberChild is controlled by a simulated neural network containing a number of different areas based on the brain’s neuroanatomy, including the premotor cortex, supplementary motor cortex, frontal eye fields, thalamic nuclei, hippocampus and amygdala. Each of these areas is modelled using twenty neuronal units and within each area about half of the units are active at any one time. Interconnection between the neural areas is based on the known anatomical connectivity of the brain and it includes efference copy connections from the premotor and supplementary motor cortices to sensory receiving areas, which Cotterill claims to be a vital feature of the neural processes underlying consciousness.

The overall aim of the CyberChild project was to use this detailed simulation to identify the neural correlates of consciousness (MC3) and perhaps even create phenomenal states (MC4). Cotterill (2003) planned to do this by looking for conscious behaviours (MC1), such as the ability to modify communications with a human operator, which could be linked to the neural correlates of consciousness in the system.[17]

3.5.5 Simple Khepera Models

A number of researchers are using simulated or real Khepera robots (see Figure 3.4) to develop simple embodied systems containing analogues of the cognitive characteristics associated with consciousness. As these robots move around their environment they build up representations, which can easily be examined for internal models or imagination.

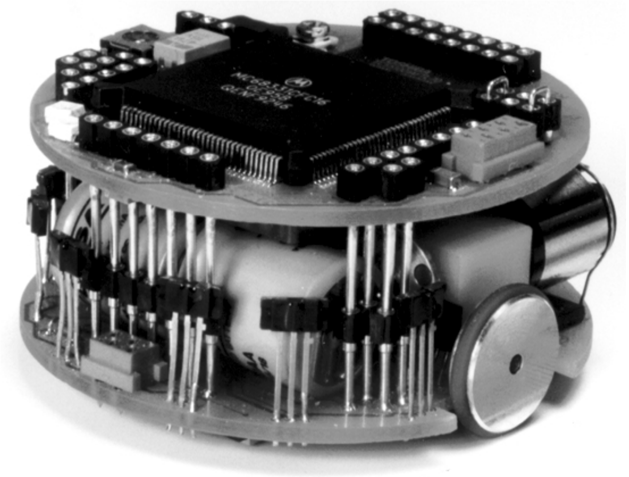


Figure 3.4. Khepera robot

Internal models

To test their ideas about the role of internal models in consciousness, Holland and Goodman (2003) used Linåker and Niklasson's (2000) Adaptive Resource-Allocating Vector Quantizer (ARAVQ) method to build models of the sensorimotor data from a Khepera robot. The ARAVQ approach is based on the observation that a robot's sensory input and motor output are often relatively stable over time - for example, when a robot is following a wall, its distance from the wall and speed remain approximately constant. Linåker and Niklasson's (2000) method takes advantage of this fact by regularly sampling a robot's sensory input and motor output and

[17] Sadly, Cotterill passed away in 2007 and it is unlikely that his work on CyberChild will be continued.

clustering this data using the ARAVQ on-line algorithm, which produces a small number of relatively stable and distinct combinations of sensory inputs and motor outputs called concepts. These concepts can be used to store long sequences of experiences very economically by labelling them and recording the number of times that each is repeated.

In their experiments, Holland and Goodman programmed a simulated Khepera with wall following and obstacle avoidance behaviour and allowed it to move around its environment while the ARAVQ method built up concepts corresponding to combinations of sensory input and motor output. Each concept represented the environmental features that activated the Khepera's rangefinders and how the robot moved in response to this stimulus, and so it was possible to plot the movements step by step along with the range finder data to produce the map of the environment that was stored inside the robot – a process that Linåker and Niklasson call inversion. By inverting the Khepera's concepts in this way Holland and Goodman produced a graphical representation of the Khepera's internal model and then examined how it could be used to control the simulated robot. They discovered that an internal model formed by concepts could accurately control the robot, process novel or incomplete data, detect anomalies and inform decisions.

These experiments showed that internal models can be developed and studied in a simple system and that they have the potential to play a useful role in the behaviour of an organism. Some of the internal models in humans are integrated into conscious cognitive states, and so this work is an example of MC2. Although Holland and Goodman do not claim that their simple system was conscious, more complex systems with internal models could contain phenomenal states (MC4) if their theories about the link between internal modelling and consciousness are correct.

Imagination

Ziemke et al. (2005) carried out a number of experiments on imagination using a simulated Khepera robot. This robot was controlled by a simple neural network that was based around a sensorimotor module, which mapped sensory input to motor output, and a prediction module. An evolutionary algorithm was used to train the weights on the two modules, with the sensorimotor module being evolved first to avoid obstacles and perform fast straightforward motion, and the prediction module evolved to predict the sensory input of the next time step. When the robot received real sensory input it was controlled by the sensorimotor module alone; when the robot was 'blindfolded' so that it received no external sensory input, it was controlled by feeding the prediction module's predictions about the next sensory input into the sensorimotor module. During the testing phase, it was found that 'imagined' sensory inputs produced very similar behaviour to real sensory input, although the pattern of activation of the internal units was very different in the two cases. These experiments demonstrated that the cognitive characteristics associated with consciousness (MC2) could improve the performance of a robot.

Ziemke's approach was developed further by Stening et al. (2005), who replaced the low level neural networks used by Ziemke with Linåker and Niklasson's (2000) ARAVQ method,[18] which was used to identify combinations of sensory input and motor output that were relatively invariant over time. The concepts generated by this method were then fed into a neural network consisting of an input layer and a hidden layer that was trained to predict when the next concept would occur. During the experiments, the robot's behaviour was initially controlled by a pre-trained neural network that moved the simulated Khepera around its environment with simple right-hand following behaviour, whilst the ARAVQ method extracted the basic features of the environment. The neural network's predictions about the next concept were then fed back into its input layer, which enabled the neural network to internally simulate a sequence of concepts

[18] See the earlier discussion of ARAVQ for more information about this method.

without the need for external movement. Stening et. al. then ‘inverted’ this sequence of concepts to produce a graphical representation of the Khepera’s ‘imagination’. This work is an example of MC2 and also falls within synthetic phenomenology (see Chapter 4). Although Hesslow and Jirenhed (2007) discuss the potential consciousness of this type of system, it is not entirely clear whether they are referring to MC2 or MC4.

3.5.6 Global Workspace Models

Global workspace theory is an influential interpretation of consciousness that was developed by Baars (1988). The basic idea is that a number of separate parallel processes compete to place their information in the global workspace, which is broadcast to all the other processes. A number of different types of process are used to analyse information or carry out actions, and processes can also form coalitions that work towards a common goal. These mechanisms enable global workspace theory to account for the ability of consciousness to handle novel situations, its serial procession of states and the transition of information between consciousness and unconsciousness. A substantial amount of work has also been done connecting the global workspace architecture to the thalamo-cortical system in the brain (Newman et al., 1997).

IDA naval dispatching system

Franklin’s (2003) IDA naval dispatching system was created to assign sailors to new billets at the end of their tour of duty. This task involves natural language conversation, interaction with databases, adherence to Navy policy and checks on job requirements, costs and sailors’ job satisfaction. These functions are carried out using a large number of codelets[19] that are specialised for different tasks and organised using a global workspace architecture.

[19] A codelet is a special purpose, relatively independent mini agent that is typically implemented as a small piece of code running as a separate thread. These codelets correspond with processors in global workspace theory.

The apparatus for ‘consciousness’ consists of a coalition manager, a spotlight controller, a broadcast manager and a number of attention codelets. These attention codelets watch for an event that calls for conscious intervention, and when this occurs they form a coalition with codelets containing data about the situation and compete for the spotlight of consciousness. If the coalition wins, its contents are broadcast to the other codelets, which may eventually choose an action that resolves the issue. The selection of behaviours in IDA is controlled by drives that award activation to behaviours that satisfy them, with activation spreading from behaviour to behaviour along excitatory and inhibitory links until an action is chosen. A model of deliberation is also included, which explores different scenarios and selects the best, and the architecture contains emotions, such as guilt at not getting a sailor’s orders out on time, frustration at not understanding a message and anxiety at not be able to convince a sailor to accept a suitable job. A number of different learning mechanisms are also implemented.

IDA is an example of a system that produces behaviour requiring consciousness in humans (MC1) and its architecture has some of the cognitive characteristics associated with consciousness (MC2), such as attention, emotions and imagination. All of this is produced by an architecture linked to human consciousness (MC3), and although Franklin thinks that IDA is unlikely to be phenomenally conscious (MC4), he does not entirely rule this out.

Dehaene et. al.’s neural simulations of the global workspace

Dehaene et. al. (1998) created a neural simulation to study how a global workspace and specialised processes interact during the Stroop task.[20] Their neural model included input and response units, global workspace neurons and vigilance and reward systems that modulated the activity in the global workspace. This simulation demonstrated that tasks that were easy for the system could be accomplished by local specialised processes without sustained activation in the

[20] In the Stroop task a subject is presented with a series of cards and has to state either the colour name that is printed on the card or the colour of the ink. This task is harder when the ink’s colour does not match the colour name, for example when “red” is printed in blue ink.

global workspace. On the other hand, tasks that were difficult for the model to accomplish, such as naming the colour of the ink when this conflicted with the colour name, could only be done by activating the global workspace and using the reward and vigilance systems to correct errors. Dehaene et. al. (1998) used this model to make predictions about brain imaging patterns generated during a conscious effortful task and about the pharmacology and molecular biology of the brain.

More recent work by Dehaene et. al. (2003) studied the attentional blink,[21] which they explained using their theory about the implementation of a global workspace in the brain. When the first target is presented to the subject, it gains access to the brain's global workspace by generating long range activations between many different neural areas and when the brain is in this state it is much harder for the second target to globally broadcast its information. Although local areas continue to carry out low level sensory processing on the second target, this does not become conscious because it cannot access the brain areas that are responsible for memory and reporting. Dehaene et al. tested these ideas about global workspace theory using a detailed neural simulation and compared their results with human subjects tested on the same experiment. Dehaene and Changeux (2005) have also used neural simulations to explore the role of spontaneous activity in workspace neurons and how this affects phenomena related to consciousness, such as inattentional blindness and transitions between the awake state and sleep, anaesthesia or coma.

Although the main emphasis of this work is on neuroscience, it closely ties in with theories about consciousness and Dehaene et al.'s neural models of global workspace theory are examples of MC3, even if they are not explicitly situated within machine consciousness. Their

[21] An attentional blink occurs in human subjects when two targets are presented in succession with 100-500 ms between them. Under these conditions the subject's ability to detect the second target is reduced, as if their attention had blinked after processing the first target.

models also fall within MC2 since they capture the fact that conscious experiences move through a serial progression of states with a limited content.

Shanahan's brain-inspired global workspace models

Shanahan (2006) developed a brain-inspired cognitive architecture based on global workspace theory, which was built using components that are functionally analogous to structures in the brain. At the bottom level of this system a sensorimotor loop made an immediate motor response to its situation, and on top of this a higher-order loop modulated the behaviour of the first order loop by adjusting the saliency of its actions. The first-order loop was closed through its interactions with the world, whereas the second-order loop was internally closed through an association area, which simulated the sensory stimulus that followed from a motor output in a way that was analogous to imagination. This simulation function was carried out using a global workspace architecture in which association areas received information from the basal ganglia analogue and competed to pass their information back to the basal ganglia analogue, which selected the next set of information to be broadcast. This architecture enabled the system to follow chains of association and explore the potential consequences of its actions prior to carrying them out.

In his experimental setup Shanahan (2006) used NRM[22] to create the neural simulation and the robot simulator Webots to simulate a Khepera robot with a camera. This system was programmed with a small suite of low level actions and trained to have positive and negative preferences for cylinders with different colours. Using its global workspace architecture the robot could explore the consequences of potential actions and give a low weighting to actions that would bring about an aversive stimulus. This enabled it to select actions that were more 'pleasant' than the ones that it would have chosen using the simple sensorimotor loop. This system is an example of MC1-3 since it is using imagination and emotion (MC2) implemented in

[22] See the brief discussion of NRM in Section 3.5.1.

a global workspace architecture (MC3) to produce behaviour that requires consciousness in humans (MC1). Although Shanahan claims that his system respects all five of Aleksander's axioms, he is cautious about attributing real phenomenal consciousness to it.

In more recent work, Shanahan (2008) built a global workspace model using simulated spiking neurons, which was based on the work by Dehaene et. al. (1998, 2003, 2005). This showed how a biologically plausible implementation of the global workspace architecture could move through a serial progression of stable states, and it had the potential to carry out the same function as the core circuit described in Shanahan (2006). Unlike the earlier model, it did not exhibit external behaviour, and so it is an example of MC2-3.

Neural schemas

The neural schema approach developed by McCauley (2002) is a neural and connectionist implementation of some aspects of global workspace theory. This system is based on a network of nodes that represent the state of the environment, actions, the effect of actions and the goals of the system, and the level of activation of these nodes can spread along the links between them. There is also a model of attention and consciousness based on global workspace theory, which allocates 'consciousness' to nodes based on their change in activation over time, their ability to accomplish current goals and their association with other nodes recently involved in 'consciousness'. This 'consciousness' of the nodes alters their behaviour and the information in them is broadcast across the network. This system is described by McCauley as an implementation of part of a psychological theory of consciousness (MC2-3), and not as something that displays true consciousness.

3.5.7 Language and Agency

Agent-based conscious architecture

Angel (1989) sets out a language- and agent-based architecture for a conscious machine centred around three attributes that must be possessed by any conscious system:

1. Independent purpose regardless of its contact with other agents.
2. The ability to make interagency attributions on a pure or natural basis.
3. The ability to learn from scratch significant portions of some natural language, and the ability to use these elements in satisfying its purposes and those of its interlocutors.

According to Angel, these behavioural attributes associated with consciousness (MC1) can only be used to infer real phenomenal states in a machine (MC4) if human consciousness is a physical phenomena that conforms to physical laws. If human consciousness can somehow pre-empt or transgress natural causes, then we cannot attribute consciousness to entities using these criteria.

Since Angel's attributes are based on language and agency, it is not difficult to produce formal models of them on a computer, and Angel suggests how a machine could be built that would actually be conscious (MC4) according to his criteria. This would lead to a minimally conscious system, which could be attributed more degrees of consciousness if it exhibited cognitive characteristics associated with consciousness (MC2), such as emotion, wakefulness, a sense of continuity with the past and an ego. As far as I am aware, there has not been any attempt to implement the architecture that Angel proposes, although the work of Steels (2003) points in this direction.

Inner Speech

According to Steels (2003), inner speech is linked to conscious experience through the role that it plays in our sense of self and agency. Steels' work on inner speech started with experiments in which two robotic heads watched scenes and played a language-game that evolved a lexicon or grammar (Steels, 2001). In one language-game, a speaker chose an object in the scene and sought a verbal description so that the hearer could guess which object was chosen. In the early versions of these experiments it was relatively easy for the agents to develop a lexicon, but they could not evolve grammar until Steels applied the speaker's language system to its own utterances, either before transmitting them or after observing incomprehension in the listener. This model of inner speech enabled the agents to evolve case grammar and Steels (2003) suggests that it could be used outside of communication to rehearse future dialogue, submit thoughts to self criticism, and conceptualise and reaffirm memories of past experiences. All of these additional functions of inner speech could be the foundation of our sense of self and they could also play a role in our inter-agency relationships with others. Steel's modelling of inner speech is mainly directed towards reproducing important aspects of our conscious experience (MC2). Although Steels suggests that complex language production may have played a crucial role in the origin of consciousness, he leaves open the possibility that models of inner speech will lead to actual phenomenal states.

Other work on the link between inner speech and consciousness includes Clowes (2006, 2007), who argues that inner speech helps to organise conscious experience, direct attention and manage ongoing activities. These ideas were tested by Clowes and Morse (2005) in some simple experiments on the structuring of action by language. Haikonen (2006) also has a detailed discussion of the relationship between inner speech and consciousness.[23]

[23] Inner speech is an example of deliberation in the sense of Sloman (1999), which is implemented in Franklin's IDA naval dispatching system - see Franklin (2000) for more on the relationship between deliberation and IDA. Deliberation in the sense of a consciously evoked internal virtual reality is closely related to internal models and imagination, which appear in several of the projects covered by this chapter.

3.5.8 Cognitive Architectures

A Cognitive Approach to Conscious Machines

Haikonen (2003, 2006, 2007) is developing a system that is intended to display cognitive characteristics associated with consciousness, such as emotion, transparency, imagination and inner speech, using a detailed neural simulation. This cognitive architecture starts with sensory modules that process visual, auditory and tactile data into a large number of on/off signals that carry information about different features of the stimulus. Perceived entities are represented using combinations of these signals, which are transmitted by modulating a carrier signal (an important aspect of Haikonen's theory of consciousness). There is extensive feedback within the system and cross connections between different sensory modalities integrate qualitative characteristics carried by the signal with its location in motor space. Haikonen's architecture also includes emotions – for example, there is an analogue of pain, which uses information about physical damage to initiate withdrawal and redirect attention. In this architecture, language is part of the auditory system and the association of words with representations from other modalities enables sequences of percepts to be linguistically described. Haikonen (2006) claims that percepts become conscious when different modules cooperate in unison and focus on the same entity, which involves a wealth of cross-connections and the forming of associative memories.

If this system can be constructed, it will be an example of MC1-4 since it is attempting to produce behaviour and cognitive states linked to consciousness using an architecture theorized to be a cause or correlate of consciousness, which may actually become conscious. At the time of writing Haikonen is working on the implementation of his proposed architecture and it is not clear how much has been completed.

Schema-based model of the conscious self

Samsonovich and DeJong's (2005a,b) cognitive architecture is based around schemas that process data items, such as semantic knowledge, action primitives or sensory qualia. The behaviour of these schemas is constrained by a set of axioms that correspond to the system's 'conscious' self. These self axioms are beliefs that the agent holds about itself, such as the fact that the self is the only author of self-initiated acts, the self is indivisible, and so on. In Samsonovich and DeJong (2005b) this system was integrated using a dynamic multichart architecture, whereas in Samsonovich and DeJong (2005a) it was coordinated by contextual, conceptual and emotional maps based on the hippocampus. Samsonovich and DeJong (2005b) describe how this cognitive architecture was used to control a virtual robot that learnt to move in open space, navigate a maze and solve a simple push-push puzzle.

This cognitive model of the conscious self is an example of an MC2 system that is capable of behaviours that require consciousness in humans (MC1). Although Samsonovich and DeJong (2005a) map their architecture onto brain areas and functions, they do not explicitly link it to any of the architectures that have been put forward as a cause or correlate of human consciousness (MC3). Samsonovich and DeJong (2005a,b) do not comment on whether their system is capable of real phenomenal states (MC4).

Cicerobot

Cicerobot is a robot created by Chella and Macaluso (2006) and Chella (2007), which has sonar, a laser rangefinder and a video camera, and works as a museum tour guide in the Archaeological Museum of Agrigento (see Figure 3.5). The cognitive architecture of this robot is based around an internal 3D simulation, which is updated as the robot navigates around its environment. When the robot moves it sends a copy of its motor commands to the 3D simulator, which calculates expectations about the next location and camera image. Once the movement has been executed, the robot compares its expected image with the 2D output from its camera and uses discrepancies between the real and expected images to update its 3D model. Cicerobot uses this 3D simulation to plan actions by exploring different scenarios in a way that is analogous to human imagination.

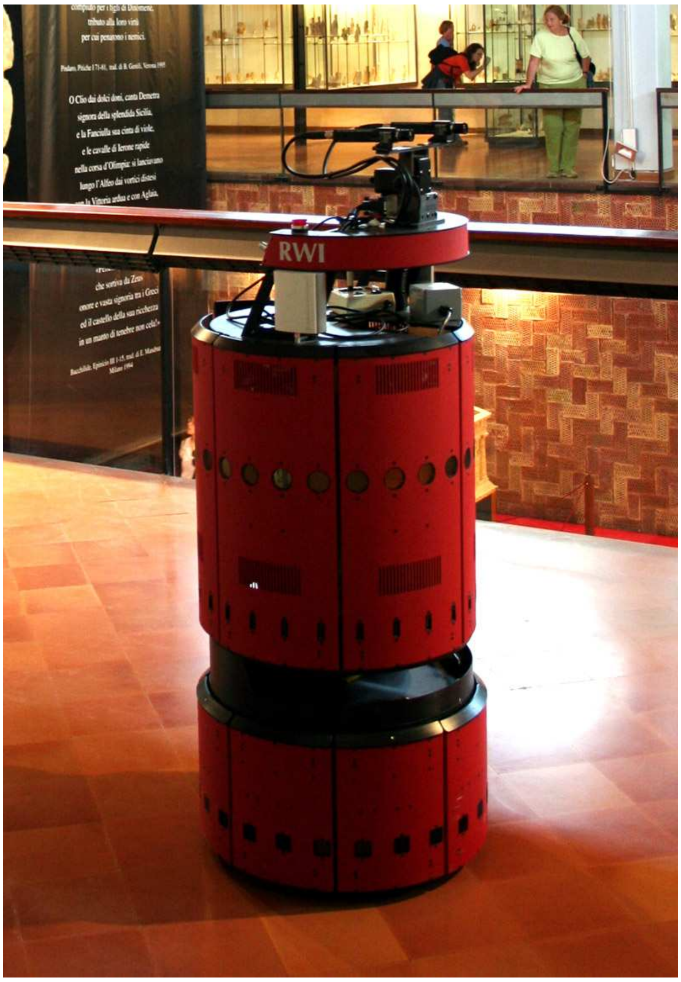

compiuto per i figli di Dinomene,
tributo alla loro virtù
per cui penarono i nemici.
Pindaro, Pitiche I 71-81, trad. di B. Gentili, Verona 1995
O Clio dai dolci doni, canta Demetra
signora della splendida Sicilia,
e la Fanciulla sua cinta di viole,
e le cavalle di Ierone rapide
nella corsa d'Olimpia: si lanciavano
lungo l'Alfeo dai vortici distesi
con la Vittoria ardua e con Aglaia,
che sortiva da Zeus
onore e vasta signoria tra i Greci
ed il castello della sua ricchezza
in un manto di tenebre non celò!»
Bacchilide, Epinicio III 1-15, trad. di E. Mandruzzato, Milano 1994
RWI

Figure 3.5. Cicerobot

This 'conscious' cognitive architecture (MC2) is used to control the robot in the unpredictable environment of a museum (MC1). Chella and Macaluso (2006) also link the robot's comparison between expected and actual perceptions to the presence of real phenomenological states (MC4).

3.5.9 Other Work

Other work on machine consciousness includes Mulhauser (1998), who used physics, computer science and information theory to outline how consciousness and a conscious self model could be implemented in a machine. There is also Duch (2005), who sets out an architecture for a conscious system that is inspired by brain-like computing principles. This proposed system's claims to be conscious would be based on its interpretation of variations in its internal states as different feelings or qualia associated with the perceived objects. Bosse et al. (2005) have carried out simulations of Damasio's core consciousness using the Temporal Trace Language (TTL) (Jonker and Treur 2002) and a simpler variation called *leads to*. In their simulations dynamic properties of the neural processes leading to emotion, feeling and core consciousness were expressed using statements in TTL and *leads to* and executed within a custom built simulation environment that enabled temporal dependencies between different parts of the model to be traced and visualised. Other neural network models of consciousness include the CODAM model that links consciousness to a copy of the signal that changes the focus of attention (Taylor 2007, Taylor and Fragopanagos 2007), Ikegami's (2007) work with a mobile agent equipped with a Fitz-Hugh-Nagumo neural network, and Cleeremans et al.'s (2007) networks inspired by Rosenthal's (1986) higher-order thought theory. More theoretical work on machine consciousness can be found in Holland (2003), Chrisley et al. (2007) and Chella and Manzotti (2007).

3.6 Social, Ethical and Legal Issues

Many people believe that work on machine consciousness will eventually lead to machines taking over and enslaving humans in a Terminator or Matrix style future world. This is the position of Kaczynski (1995) and Joy (2000), who believe that we will increasingly pass responsibility to intelligent machines until we are unable to do without them - in the same way that we are increasingly unable to live without the Internet today. This would eventually leave us at the mercy of potentially super-intelligent machines that may use their power against us. Against these apocalyptic visions, Asimov (1952) agrees with Kaczynski and Joy about how the machines will take over, but suggests that computers will run the world better than ourselves and actually make humanity happier.[24] A similar position is put forward by Sloman (2006), who argues that "It is very unlikely that intelligent machines could possibly produce more dreadful behaviour towards humans than humans already produce towards each other, all round the world even in the supposedly most civilised and advanced countries, both at individual levels and at social or national levels."

At present our machines fall far short of many aspects of human intelligence, and we may have hundreds of years to consider the matter before either the apocalyptic or optimistic scenarios come to pass. It is also the case that science fiction predictions tell us more about our present concerns than about a future that is likely to happen, and our attitudes towards ourselves and machines will change substantially over the next century, as they have changed over the last. For example, Kurzweil (2000) argues that as machines become more human and humans become more machinic, the barriers will increasingly break down between them until the notion of a *takeover* by machines makes little sense. Furthermore, as machines develop, the safety regulations will increase and we may be able to build a version of Asimov's laws into them, or at least exclude intense negative emotions such as hate or envy. At present, work on machine consciousness has many benefits (see Section 3.7) and it is not justified to call a halt to the whole program because of scare stories and science fiction visions.[25]

A second ethical dimension to work on machine consciousness is how we should treat conscious machines. As Torrance (2005) points out, we will eventually be able to build systems that are not just instruments for us, but participants with us in our social existence. However, this can only be done through experiments that cause conscious machines a considerable amount of confusion and pain, which has led Metzinger (2003) to compare work on machine consciousness

[24] Moravec (1988) was also an early advocate of this view.

[25] These ethical issues were discussed at length at the 2006 AGIRI Workshop: http://www.agiri.org/forum/index.php?showtopic=23.

to the development of a race of retarded infants for experimentation. We want machines that exhibit behaviour associated with consciousness (MC1) and we want to model human cognitive states (MC2) and conscious architectures (MC3), but we may have to *prevent* our machines from becoming phenomenally conscious (MC4) if we want to avoid the controversy associated with animal experiments. This can only be done by developing systematic methods for evaluating the likelihood that a machine can experience phenomenal states.[26]

A final aspect of the social and ethical issues surrounding machine consciousness is the legal status of conscious machines. When traditional software fails, responsibility is usually allocated to the people who developed it, but the case is much less clear with autonomous systems that learn from their environment. A conscious machine might malfunction because it has been maltreated, and not because it was badly designed, and so its behaviour could be blamed on its carers or owners, rather than on its manufacturers. Conscious machines could also be held responsible for their own actions and punished appropriately.[27] A detailed discussion of these issues can be found in Calverley (2005).

3.7 Potential Benefits of Machine Consciousness

This final section takes a look at some of the positive outcomes that might be realised through research on machine consciousness. Although research on MC1 is still at an early stage, it could eventually help us to produce more plausible imitations of human behaviour. In the shorter term, this might appear as more sophisticated chatterbots that carry out simple conversations as part of a telephone or web application. Progress with MC1 is most likely to come from research on other aspects of machine consciousness, such as MC2 or MC3.

[26] The ethical treatment of conscious machines is also discussed by Stuart (2003).

[27] Punishment might have to be limited to machines with some kind of self awareness if we want to avoid the absurdities of the criminal prosecution of animals in the Middle Ages – see Evans (1987).

One of the main benefits of research on MC2 will be the development of machines that can connect emotions with objects and situations, attend to different aspects of their environment, and imagine themselves in non-present scenarios.[28] This could eventually lead to machines that can understand our human world and language in a human-like way, which would vastly increase their ability to assist us and interact with us. Work on MC2 might also open up intersubjective possibilities between humans and machines, enabling computers to imagine what people might be thinking, empathize with them and imitate them.

At present, MC3 research is mainly oriented towards modelling the architectures that have been associated with human consciousness, which is an excellent way to test ideas about how consciousness works in human beings. When this modelling involves simulated neural networks, it can advance our understanding of the neural correlates of consciousness, as seen in the work of Shanahan (2006, 2008) and Dehaene et al. (1998, 2003, 2005). This neural modelling could improve our diagnosis of coma and locked-in patients and help us to understand how the brain processes information, so that we can develop prosthetic interfaces to restore visual, auditory or limb functions. MC3 work can also help us to develop machines that tackle problems in a similar way to humans, such as Franklin's naval dispatching system.[29]

Although we often want to avoid phenomenal states in machines, work on MC4 does have a number of potential benefits. The most important of these is the development of systematic ways of examining systems for signs of consciousness and making predictions about their phenomenal states. By working hand in hand with neurophenomenology, this synthetic phenomenology could lead to more scientific theories about animal suffering and it will be discussed in detail in the next chapter. Up to this point it has always been a vague question about whether, for example, snails feel pain, but MC4 research may eventually be able to make detailed predictions about the phenomenal states of non human systems. This could also help us

[28] Part of the work on deliberation – see footnote 23.

[29] See Franklin (2001) for more on how IDA tackles problems in a similar way to humans.

to understand the phenomenal states of very young or brain-damaged people who are incapable of communicating their experiences in language.

3.8 Conclusions

Machine consciousness is a relatively new research area that has gained considerable momentum over the last few years, and there is a growing number of research projects in this field. Although it shares some common ground with philosophy, psychology, neuroscience, computer science and even physics, machine consciousness is rapidly developing an identity and problems of its own. The benefits of machine consciousness are only starting to be realised, but work on MC2-3 is already proving to be a promising way of producing more intelligent machines, testing theories about consciousness and cognition, and deepening our understanding of consciousness in the brain. As machine consciousness matures it is also starting to raise some novel social and ethical issues.

One of the challenges in MC4 work on machine consciousness is to establish whether a system is capable of phenomenal states and to describe these phenomenal states when they occur. This challenge is addressed by the emerging discipline of synthetic phenomenology, which is covered in Chapter 4. Chapter 5 describes the design and implementation of an MC1, MC2 and potentially MC4 neural network, whose phenomenal states are analyzed in detail in Chapter 7.

4. Synthetic Phenomenology[1]

> At present we are completely unequipped to think about the subjective character of experience without relying on the imagination - without taking up the point of view of the experiential subject. This should be regarded as a challenge to form new concepts and devise a new method - an objective phenomenology not dependent on empathy or the imagination. Though presumably it would not capture everything, its goal would be to describe, at least in part, the subjective character of experiences in a form comprehensible to beings incapable of having those experiences.
>
> Nagel (1974, p. 449)

4.1 Introduction

Synthetic phenomenology is a new area of research that has emerged out of work on machine consciousness. The term was first coined by Jordan (1998), who used it to refer to the *synthesizing* of phenomenal states and a second interpretation was suggested by Chrisley and Parthemore (2007), who interpret synthetic phenomenology as the "attempt to use the states, interactions and capacities of an artificial agent for the purpose of specifying the contents of conscious experience." (p. 44). In this usage, an artificial system is being employed to describe the phenomenology of a second system, which could be human, in order to overcome the limitations of natural language. Synthetic phenomenology can also refer to the *determination whether artificial systems are capable of conscious states and the description of these states if they occur*, and it is in this sense that I will be using it in this thesis. This approach to synthetic phenomenology is similar to that put forward by Aleksander and Morton (2007a) and it is close to the philosophical tradition of phenomenology, with the word "synthetic" being added to indicate that it is the phenomenology of artificial systems that is being described. Husserl's (1960) phenomenological project was the description of human consciousness; the synthetic

[1] Earlier versions of parts of this chapter were published as Gamez (2005) and Gamez (2006).

phenomenological project is the description of machine consciousness - a way in which people working on machine consciousness can measure the extent to which they have succeeded in realizing consciousness in a machine.[2]

It is impossible to describe the phenomenology of a system that is not *capable* of consciousness, and so the first challenge faced by synthetic phenomenology is to identify the systems that are capable of phenomenal states. In Chapter 2 it was argued that we do not have a viable metaphysical theory of consciousness, and so we can only tell if a system is conscious by looking at its type I and type II potential correlates of consciousness (PCCs). Setting aside the problem that some correlates of consciousness may be probabilistic and multifactorial, the behaviour-neutrality of type I PCCs means that we cannot identify a list of the necessary and sufficient correlates of consciousness. This prevents us from ever knowing for *certain* whether biological neurons, for example, are necessary for consciousness, or if they are just one of the mechanisms by which consciousness happens to be implemented in human beings. Since it is indeterminable whether silicon-based robotic systems are conscious or not, a major obstacle lies in the way of any attempt to describe the *phenomenology* of such systems.

One approach to this problem is to follow Prinz (2003) and suspend judgement about whether robots are capable of phenomenal states. However, one problem with this approach is that many people have a strong intuition that machines built in a similar way to humans are likely to be phenomenally conscious, and so it may be necessary to take the idea that certain types of machines have conscious experiences seriously. Second, as machine consciousness progresses we are likely to start developing machines that exhibit more complex behaviour and spend a lot of time confused and potentially in pain. This has been somewhat dramatically compared by Metzinger (2003, p. 621) to the development of a race of retarded infants for

[2] Traditional and synthetic phenomenology have different objectives: traditional phenomenology was trying to increase our understanding of the world; synthetic phenomenology is describing the phenomenal states of machines in order to monitor their consciousness and change their behaviour.

experimentation. To address these ethical worries without stifling research a way needs to be found to evaluate the likelihood that a robot is capable of phenomenal states. A third problem with suspending judgement is that as more sophisticated robots emerge, people are inevitably going to attribute more and more consciousness to them. People are already prepared to attribute emotions to robots as simple as Braitenberg's vehicles (Dautenhahn 2007), and a systematic way of evaluating phenomenal states in a system needs to be in place before this becomes a live public issue. The general public is very interested in the question whether something is *really* conscious and it would be helpful if the machine consciousness community could formulate some kind of answer, even if this is based on analogy with human beings. To address these issues and provide a framework within which the more detailed work of synthetic phenomenology can proceed, Section 4.2 outlines a scale that orders machines according to the degree to which their type I PCCs match human type I PCCs.

The next part of this chapter suggests how type II theories of consciousness can be used to generate a description of a machine's phenomenal states. This approach is based around concepts of a mental state and a representational mental state, which are defined in Section 4.3 along with some methods for identifying them in artificial systems. Once we have identified the system's representational and non-representational mental states and made predictions about their association with phenomenal states, we need to find a way of moving from the physical description of the mental states to a description of the system's phenomenology. Section 4.4 outlines some of the reasons why human language is unsuitable for the description of non-human mental states, and puts forward an alternative approach that uses a markup language to combine human and physical descriptions with other information about the system. Finally, the last part of this chapter covers some of the previous work that has been carried out in synthetic phenomenology.

It is worth noting that this approach to synthetic phenomenology makes no assumptions about whether any particular machine is capable of supporting conscious states: robots, stones and human beings all have internal states and all three can be analysed using this approach.[3]

4.2 Ordinal Machine Consciousness (OMC) Scale

> … we may say that measurement, in the broadest sense, is defined as the assignment of numerals or events according to rules. The fact that numerals can be assigned under different rules leads to different kinds of scales and different kinds of measurement. The problem then becomes that of making explicit (a) the various rules for the assignment of numerals, (b) the mathematical properties (or group structure) of the resulting scales, and (c) the statistical operations applicable to measurements made with each type of scale.
>
> Stevens (1946, p. 677)

4.2.1 Introduction

The discussion of the brain-chip replacement experiment showed that it is impossible to establish whether the behaviour-neutral type I aspects of a system, such as the material it is made from, are correlated with consciousness or not (see Section 2.5.6). The presence of biological neurons *might* be necessary for consciousness or it *might* not, and the introduction to this chapter put forward a number of reasons why we need to make a decision about this, even if we cannot judge with certainty. To address this issue, this section sets out a proposal for an ordinal[4] machine consciousness (OMC) scale that makes predictions about what people would say about the consciousness of non-human systems based solely on their type I PCCs. Type II PCCs do not need to be included in the OMC scale because their correlation with consciousness can be empirically assessed.

[3] Stones have few of the human type I PCCs, but it is an open and empirical question whether any of the type II theories of consciousness would predict that they have phenomenal states.

[4] See Stevens (1946) for the difference between nominal, ordinal, interval and ratio scales. It was decided to make the scale ordinal because it was anticipated that it would only be possible to measure people's assessment about whether one system is more or less conscious than another. In the future it may be possible to develop an interval or ratio scale.

The OMC scale is a model of our subjective judgement about the consciousness of artificial systems, and although it might initially seem counterintuitive to use a numerical scale to rank our judgements about the consciousness of systems, there has been a lot of psychophysical work on the measurement of other subjective qualities, such as brightness, loudness, the hardness of minerals, beauty or the desirability of automobiles (Baird and Noma 1978). The OMC scale is a logical extension of this work that attempts to predict the degree to which a system's type I PCCs are judged by us to be relevant to consciousness. As Stevens (1946) points out, measurement scales are possible when there is an isomorphism between certain properties of objects and the properties of numerical series, and this isomorphism enables the series to model relevant aspects of the empirical world. In this thesis the OMC scale is a proposed ordering of systems that is predicted to match people's judgments about systems' consciousness based on their type I PCCs.

This project did not have the resources to base the OMC scale on empirical measurements of people's judgements about the link between type I PCCs and consciousness, and so the current version is put forward as a model of how people would make this type of judgement. This use of models in psychophysics is summarised by Baird and Noma (1978):

> In brief, a psychophysical theory is a set of statements (assumptions) that describes how an organism processes stimulus information under carefully specified conditions. The assumptions usually concern hypothetical processes that are difficult or impossible to observe directly. Once these assumptions are made explicit, however, formal models can be devised. The validity of the theory can be tested by comparing observations against the predictions of the model. In other words, a theory represents a set of "reasonable" guesses about exactly *how* a person behaves as a measuring instrument when asked to judge properties of stimuli.
>
> Detailed predictions of what a person will actually *do* in an experiment are based on models especially designed to test one or more theories. Although in recent years the terms "model" and "theory" have often been used interchangeably, a model is thought to be a concrete synthesis of the assumptions of a theory. This synthesis specifies the interrelationships among the postulated primitives of the theory. Often these statements are in the form of mathematical formulas, computer programs, or logical truisms. In this way they are both

> more specific and yet more general than the theory giving rise to them – more specific in that the theory, through its models, is now amenable to laboratory test, and more general in that an abstract model may be used to quantify theories in many areas of study.
>
> Baird and Noma (1978, pp. 2-3)

The current OMC model enables predictions to be made about what people would say about the consciousness of non-human systems based solely on their type I PCCs, and it is used in this thesis to demonstrate a new approach to synthetic phenomenology.

This description of the OMC scale starts with an overview of the systems that are covered by it. After explaining the factors and the way in which they are combined, some examples are given to illustrate how it works. How this model might be validated and improved using real data is discussed in Section 4.2.7.

4.2.2 Systems Covered by the OMC Scale

In order to focus on the *behaviour-neutral* aspects of each system, the systems ranked by the OMC scale need to have their behaviour held constant in some way, which can be done by specifying that all of the systems ranked by the OMC scale must conform to the behaviour set of a system that is generally acknowledged to be conscious. This ensures that a system's type I PCCs are the only factors that affect its position on the scale.

Since humans are our paradigmatic conscious systems, the functions of the human brain can be used to specify a set of behaviours that systems on the OMC scale would have to match.[5] This notion of approximating the functions of the human brain could be defined using Harnad's (1994) extended T3 version of the Turing test. A machine that could pass this test would be able to control a human or artificial body in a way that was functionally indistinguishable from a

[5] This way of specifying the behaviour of systems covered by the OMC scale sets aside the whole question of the body. In theory a computer could approximate the behaviour of the human brain without needing a body at all. However, such a system would be almost impossible to develop and there might be a critical link between the body and consciousness that would be missed by a purely brain-based approach – see, for example, Damasio (1999) for more on the link between the body and consciousness.

human for 70 years or more. Such a system could hold down a job, create works of art and have relationships with other human beings. Machines that were in a persistent vegetative state or interned in an asylum for strange behaviour would not be considered functionally identical to a human being according to this measure.

Whilst the T3 version of the Turing Test defines the behaviour of a paradigmatically conscious system, it has the disadvantage that our current machines are very far from passing it – if T3 was used as the definition of behaviour, then the OMC scale could only be applied to our current systems by treating them *as if* they had developed to the point at which they were capable of passing it. A second way of defining the behaviour set of a conscious system would be to look at humans who exhibit far less complex behaviour. For example, since we attribute consciousness to locked-in patients who are limited to the movement of a single eyelid,[6] the symbolic T2 version of the Turing Test might be enough for behaviour neutrality. Many other brain damaged people are also examples of systems that are attributed consciousness, but might not be able to pass the T3 Turing Test, and their behaviour could also be used as a common standard for systems ranked by the scale.

A third possibility is that our knowledge about animal consciousness might develop to the point at which an animal's brain could be used to specify a set of conscious behaviours. Systems that conformed to this behaviour set would have to approximate the behaviour of the brains of animals that are known to be conscious by controlling a body similar to the animal's for the lifetime of the animal (systems that imitated one or two simple behaviours, such as flying or swimming, would be attributed less consciousness than the animal on behavioural grounds). Whichever definition of behaviour is used, it is not the behaviour per se that is important, but the fact that it approximates the behaviour of a system that is agreed to be conscious, so that only the type I attributes of the system affect our judgement about its potential for conscious states.

[6] For example, see Baubey (2002).

4.2.3 OMC Factors and Weights

The scale is built in a modular fashion so that factors can easily be added, removed or adjusted to match data gathered by psychophysical experiments. Each system is assigned a weight, ω, for each of its type I PCCs, and these weights are combined according to the rules set out in Section 4.2.4 to generate the scale. The working assumption behind the OMC scale is that people's attribution of consciousness to a system is largely based on similarities between the system and the human brain, and so it was decided to set ω to 1.0 when the system was the same as the human brain for a particular PCC. When the system deviates from the human brain on a particular factor, it is given a weight less than 1.0, and to preserve the modularity of the scale the minimum value of ω was limited to 0.1. So, for example, Table 4.1 shows how the system is assigned a weight of 1.0 if it runs at approximately the same speed as the human brain, a weight of 0.55 if it runs ten times faster or slower than the human brain, and a weight of 0.1 if it runs over a hundred times faster or slower than the human brain.

The current version of the OMC scale only covers a very small selection of the type I PCCs that have turned up in discussions of consciousness in artificial systems by Block (1978), Searle (1980), Kent (1981) and others, and the assignment of weights has been done in a subjective and somewhat arbitrary fashion. In the future it is hoped that psychophysical methods could be used to test and improve the scale, and some suggestions about how this could be done are given in Section 4.2.7. An outline of the factors that I have selected for version 0.6 of the OMC scale now follows.

Rate

Machines can operate much faster or slower than the human brain and we are more likely to attribute consciousness to a machine that runs at approximately the same speed. If we were forced to say whether the economy of Bolivia or the Earth's crust is more likely to be conscious, we would probably choose the economy of Bolivia. This is not because it is more complex or has more states, but because its states change more rapidly.

	Rate	***ω***
R1	Approximately the same speed as the human brain	1.0
R2	Ten times faster or slower than the human brain	0.55
R3	Over a hundred times faster or slower than the human brain	0.1

Table 4.1. Rate factors

Size

We are more likely to attribute consciousness to a system that fits inside a person's head, than to a system that is the size of the population of China.

	Size	***ω***
S1	Approximately the same size as the human brain	1.0
S2	A thousand times larger or smaller than the human brain	0.55
S3	More than a million times larger or smaller than the human brain	0.1

Table 4.2. Size factors

Function Implementation

There are a wide variety of ways in which the functions of a system can be implemented, some of which are closer to human biology than others. This factor weights machines according to the degree to which the implementation of their functions matches that of the human brain. I have gone down to the atomic level to take account of claims by Hameroff and Penrose (1996) that consciousness depends on quantum functions.

This factor is complicated by the fact that neurons can be used to implement functions in a biological and non-biological way. For example, a function can be implemented by a neural network trained by back propagation or by a more biological structure of neurons. Since neurons can themselves be simulated using neurons there is potential for infinite self-recursion, which I have limited by introducing a restriction on the number of levels. To keep things simple I have also set aside the possibility that glia play an information-processing role (Haydon 2000).

The way in which these three tables are combined is fairly self-evident. If the functions are implemented by a biological structure of neurons (F1 in Table 4.3), then the way in which the function of the neurons is implemented has to be specified as well (for example, FN1 in Table 4.4). No further levels are required if a system's functions are implemented in a non-neural way (F3 in Table 4.3).

Since all computer simulations are physical systems consisting of a certain combination of molecules, atoms and ions, the purpose of the function implementation factor is not to determine whether the system is simulated or not, but to capture the level of detail at which the system's functions match the functions of the human brain. For example, if a system's functions are carried out using a large lookup table, then this might be stored as voltages in the computer's RAM, which is a physical thing, but we are more likely to attribute consciousness to a system that implements the brain's functions using simulated neural networks. We attribute maximum consciousness to systems that match the human brain all the way down to the level of molecules, atoms and ions and implement the molecules, atoms and ions using real biological molecules, atoms and ions.

	Function implementation	ω
F1	Produced by a biological structure of neurons	1.0
F2	Produced by a non-biological structure of neurons	0.55
F3	Produced using mathematical algorithms, computer code or some other method	0.1

Table 4.3. Function implementation

	Function of neurons	*ω*
FN1	Produced by a biological structure of molecules, atoms and ions	1.0
FN2	Produced by a non-biological structure of molecules, atoms and ions (silicon chemistry, for example)	0.7
FN3	Produced by a non-biological structure of neurons	0.4
FN4	Produced using mathematical algorithms, computer code or some other method	0.1

Table 4.4. Neuron implementation

	Function of molecules, atoms and ions	*ω*
FMAI1	Produced by real subatomic phenomena, such as protons, neutrons and electrons	1.0
FMAI2	Produced by a non-biological structure of neurons	0.55
FMAI3	Produced using mathematical algorithms, computer code or some other method	0.1

Table 4.5. Molecule, atom and ion implementation

Time Slicing

The processing of functions can be carried out in parallel with all of them operating simultaneously on dedicated hardware. On the other hand a single processor can emulate the parallel operation of many functions by time-slicing. This scale follows Kent (1981) in ranking time-sliced systems, which approximate the time complexity of the brain, as being less likely to be phenomenally conscious than systems with the same moment to moment space complexity as the brain.

	Time slicing	*ω*
TS1	All functions are dynamically changing and co-present at any point in time	1.0
TS2	Some functions are dynamically changing and co-present at any point in time	0.55
TS3	A single function is dynamically changing and present at any point in time	0.1

Table 4.6. Time slicing

Analogue / Digital

Although spiking neurons have a digital aspect, the brain includes many analogue processes that may be more faithfully captured by an analogue system.[7]

	Analogue / digital	***ω***
AD1	Analogue system	1.0
AD2	Mixture of analogue and digital	0.55
AD3	Digital system	0.1

Table 4.7. Analogue / digital systems

4.2.4 Putting it All Together

To obtain the final OMC scale, a complete list of all the possible machines is extracted from the factor tables. The weights applicable to each are then multiplied together to give total weightings for each of the possible machines, which are used to situate them on an ordinal scale. Since many of the machines have the same total weighting, this scale is much shorter than the number of possible combinations. A couple of extra rules were also introduced for the combination of factors:

1. Since neurons can be used to simulate the behaviour of neurons or the molecules, atoms and ions that neurons are composed of, the function implementation is potentially infinitely self-recursive. To prevent this I have stipulated that if non-biological structures of neurons are used to implement the functions of neurons or the functions of molecules, atoms and ions, then the neurons that are used for this cannot themselves have their functions implemented using non-biological structures of neurons.

[7] See Roberts and Bush (1981) for examples of analogue processing in the brain, and Shu et al. (2006) for experimental work on the hybrid analogue and digital nature of spike transmission.

2. When machines have less levels of functional implementation than the brain some kind of penalty needs to be imposed on machines that deviate from the human structure – for example, when functions are implemented by a lookup table instead of using biologically structured neurons implemented with molecules, atoms and ions. In the present scale there are three levels of functional implementation and I have used 0.1 as the weighting for each missing level.

The position, omc_{pos}, of an actual machine on a scale with omc_{tot} possible positions is found by calculating its total weighting, and looking for this value in the complete list of possible machines. To facilitate some kind of comparison between different versions of the scale, omc_{pos} is normalised to a value between 0 and 1 to give the final OMC rating, omc_{rat}, using Equation 4.1:

$$omc_{rat} = 1 + \frac{1 - omc_{pos}}{omc_{tot}}, \qquad (4.1)$$

which gives a rating of 1 for human brains and a rating close to zero for the last system on the list. The closer this OMC rating is to 1 the more human-like are its type I potential correlates of consciousness. Citations of a system's OMC rating should include the version of the scale, since it is anticipated that it will evolve over time.

When all of a machine's functions are implemented in the same way, this scale provides the OMC rating for the complete system. However, some machines include components that have different OMC ratings – for example, a human brain with a silicon hippocampus. In this case, the OMC rating should be calculated for each part of the system.

The current version of the OMC scale starts with human beings and finishes with digital single-processor simulations based on non-biological principles that are much larger or smaller than the human brain and process at a much slower or faster rate. There is not space in this chapter to list the OMC ratings of all of the possible machines (the complete list has several hundred thousand combinations), and so I have integrated everything together on a webpage,[8] which can be used to calculate the position of a machine on the scale and its OMC rating. Some examples are given in the next section.

4.2.5 Examples

To illustrate the operation of the OMC scale, this section gives some examples of the position and rating of different types of system. At present none of these are even close to reproducing all of the functions of the human brain for 70 years, and so this evaluation would only apply to them after they have developed to the point at which they can pass the T3 version of the Turing test or could match one of the less complex behaviour sets discussed in Section 4.2.2.

Neurally Controlled Animat

This is a system developed by DeMarse et al. (2001) that uses biological neurons to control a computer-generated animal in a virtual world. The biological neurons start off in a disorganised state and then self-assemble in response to stimulation from their environment. Since the organisation of the neurons is not determined by the many factors present in embryological development, this system produces the functions of the whole brain from a non-biological structure of neurons. The factors are thus: R1, S1 F2, FN1, FMAI1, TS1 and AD1, giving a total weighting of 0.55, an OMC position of 3 out of 192 and an OMC rating of. 0.990.

Lucy

Lucy is a robot developed by Grand (2003) that is controlled by a multi-processor simulation of neurons arranged in a biological structure. The factors are thus R1, S1, F1, FN4, TS2 and AD3, giving a total weighting of 5.5×10^{-3}. This needs to be multiplied by 0.1 to compensate for the

[8] The OMC scale webpage is included in the Supporting Materials along with the code that was used to generate it.

fact that Lucy's functions are not implemented at the level of molecules, atoms and ions, making the total weighting 5.5 x 10^{-4}. This gives Lucy an OMC position of 96 out of 192 and an OMC rating of 0.505.

IDA

IDA is a naval dispatching system created by Franklin et. al. (2003) that is based on Baars' (1988) global workspace model of consciousness.[9] The solutions used to implement this system are all non-biological, and so the factors are R1, S1, F3, TS2 and AD3. This gives a total weighting of 5.5 x 10^{-3}, but since the functions are not implemented at the level of neurons or molecules, atoms and ions, this needs to be multiplied by 2 x 0.1, to give a total weighting of 5.5 x 10^{-5}, which results in an OMC position of 146 out of 192 and an OMC rating of.0.245.

The Population of China

This is a thought experiment suggested by Block (1978) in which the functions of a human brain are carried out by the population of China interconnected by two-way radios and satellites. The population of China is approximately 1.3 billion and so this 'machine' is very much larger than the human brain. It is also likely to work at a much slower rate. This 'machine' contains both biological neurons and other hardware, and so the OMC rating has to be calculated separately for the different parts of the system.

The biological parts are implemented using a non-biological structure of neurons whose function is in turn implemented using a biological structure of molecules, atoms and ions, giving the factors R3, S3, F2, FN1, FMAI1, TS1 and AD1, which works out as a total weighting of 5.5 x 10^{-3}, an OMC position of 50 out of 192 and an OMC rating of 0.745. The rest of the system, consisting of the two-way radios, satellites, etc., has factors R3, S3, F3, TS2 and AD3, which gives a total weighting of 5.5 x 10^{-5} that needs to be multiplied by 2 x 0.1 to compensate for the

[9] IDA is covered in more detail in Section 3.5.6.

missing levels of functional implementation. This gives a total weighting of 5.5 x 10^{-7}, an OMC position of 188 out of 192 and an OMC rating of 0.02604.

4.2.6 OMC Scale Discussion

It is possible that consciousness decreases gradually as we move away from the human machine, or there may be a cut off point at which it simply vanishes. For example, there might be less consciousness when neurons are simulated using time slicing, or no phenomenal states at all when this is used in a system. We cannot empirically establish whether consciousness cuts off or not, but this does lead to two different interpretations of the OMC scale. If consciousness cuts off abruptly, then the OMC rating can be interpreted as our evaluation of the likelihood that consciousness is present in a machine that is built in a particular way. On the other hand, if consciousness decreases gradually as the factors become less human, then the OMC scale ranks machines according to our judgement about their degree of consciousness.

This is an extremely anthropocentric scale in which the great chain of machines is a kind of fall from grace from perfectly conscious man. This is an epistemological necessity – we only know for sure that we are conscious – but it is quite possible, although empirically undeterminable, that robots at the far end of the scale are more conscious than ourselves.[10]

The final OMC rating expresses an *ordering* of machines according to our subjective judgement about the relationship between their type I attributes and consciousness, so a system with an OMC rating of 0.8 is judged to be more conscious (or more likely to be conscious) than a system with a rating of 0.6. However, because successive intervals on the scale are not necessarily equal, it is incorrect to say that a system with an OMC rating of 0.8 is judged to be twice as conscious (or twice as likely to be conscious) as one with an OMC rating of 0.4.

[10] If we judged machines to more conscious than humans, then we could assign them an OMC rating greater than 1.

This scale only covers type I PCCs that cannot be empirically tested and affect our *a priori* judgement about a machine's potential for phenomenal states. For this reason the scale excludes many of the factors that have been put forward as PCCs, such as synchronization between neurons, a global workspace architecture, a model of the self, and so on. The correlation between these factors and consciousness can be assessed empirically and it is hoped that we will eventually come up with a list of type II correlates that are *necessary* in a conscious system. Any machine that lacked one of these necessary conditions would not be deemed to be conscious, regardless of its position on the OMC scale. However, a list of type II correlates will never be *sufficient* for the prediction of consciousness because one or a number of type I correlates might be necessary as well. Final judgements about a system's potential phenomenal states should combine the OMC scale's *a priori* evaluation about its capability for consciousness with an empirical assessment using a type II theory.

Many type I PCCs, such as the size of a system or its material, do not substantially change from moment to moment and the OMC rating can be calculated once for the entire lifetime of the system. When a system's type I PCCs change over time, its OMC rating may have to be recalculated each time its phenomenology is described.

It must be emphasized that a high OMC rating does not indicate that a system is actually conscious – for example, living humans have an OMC rating of 1.0 and yet they are only conscious for up to 16 hours per day. A high OMC rating only indicates that the system is judged to completely or approximately match humans on all of the type I PCCs that are judged to be relevant to consciousness; this OMC rating has to be combined with a type II theory to make predictions about whether the system is actually conscious at any point in time.

Although the current OMC scale has many limitations, the most important question is not whether this particular version makes sense, but whether the problems raised by the brain-chip replacement experiment require us to use this type of scale. If the type I/ II distinction outlined in sections 2.5.7 and 2.5.8 is valid, then something like this OMC scale is likely to become an essential tool in machine consciousness research, and the question becomes which is the best possible scale for this purpose. On the other hand, if it can be shown that the distinction between type I and type II PCCs is mistaken, then there is no need for the OMC scale at all.

Finally, as technology and culture develops, people's intuitions will change, and a revised version of the scale will have to be produced every few years. As we get closer to achieving machine consciousness, this scale might eventually become superfluous: when we talk to robots every day, work with robots that display conscious behaviour and perhaps even marry robots, we might cease to worry about whether they *really* have phenomenal states, just as we rarely see other people as automatons.

4.2.7 Future Development of the Scale

The current version of the scale is a model that predicts the subjective judgements that people will make about the link between type I PCCs and consciousness. In the future this model needs to be tested on real data by surveying people's judgments about the consciousness of systems with different type I PCCs. One way in which this could be done would be to show people short films of a humanoid body controlled by brains, computers and other artefacts with different type I PCCs, and ask participants to order them according to their potential for consciousness. To begin with each factor could be varied individually and people could be asked about whether system A was more or less conscious than system B to get an ordinal scale for each factor. The factors could also be varied in combination and factors would have to be tested that were not on the current version of the scale. One potential problem with carrying out these experiments on the general public is that their judgements are likely to be based on an amalgam of what they have seen in science fiction films and read in the media - although it could be argued that these popular representations reflect our underlying beliefs as well as alter them. Expert opinion has the opposite problem that it can be too linked to particular theories, and so it would be best to obtain sample data from both groups.

The first application of this data would be to revise the lists of factors that are used to construct the scale. For example, if people systematically believed that green objects were less likely to be conscious than red, then colour could be added as a factor. The weightings within each list of factors would also have to be fine tuned, and it is anticipated that many of them will approximate Fechner's logarithmic law, which is given in Equation 4.2:[11]

$$\varphi = k \log \phi \, , \tag{4.2}$$

where φ is the sensation magnitude, k is a constant and ϕ is the intensity of the stimulus in units above an absolute threshold.[12]

A second application of this data would be to look at different ways of integrating the factor scales. It might turn out that the current approach makes good predictions about the data, but if it is not a good fit, then it would be worth experimenting with different methods of combining the weights. One possibility would be to add the weights, and it might be necessary to weight the factors to accommodate the fact that people attribute different importance to different PCCs. Another option would be to use Shepard-Kruskal multidimensional scaling to combine the different ordinal rankings into a single Euclidean space and use the normalized distance from the most conscious system as the OMC rating (Shepard 1962a,b, Kruskal 1964).

Another direction of future work would be to use psychophysical methods to establish thresholds for the subjective assignment of consciousness and it might be possible to obtain an interval scale by including equisection or category scaling in the survey of people's judgements – see Gescheider (1997) for an overview of these methods. To obtain a more mathematically sophisticated scale, nonmetric scaling could be used to convert the ordinal scale into an interval

[11] More details about Fechner's law can be found in Gescheider (1997).

[12] This logarithmic relationship has already been incorporated into the size and rate factors of the current scale.

scale (Shepard 1966).[13] A ratio scale would be more difficult to achieve since it depends on an absolute zero, which might be difficult to agree upon in consciousness research – for example, some people might be prepared to assign consciousness to a vacuum, thinking, perhaps, that it could contain a spiritual non-material substance.

4.3 Mental and Representational States

4.3.1 Human and Synthetic Phenomenology

To clarify the relationship between synthetic and traditional phenomenology,[14] I will give a couple of examples from Husserl's phenomenology of time consciousness and Merleau-Ponty's phenomenology of the body and the senses:

> In the "perception of a melody," we distinguish the tone *given now*, which we term the "perceived," from those which *have gone by*, which we say are "not perceived." On the other hand, we call the *whole melody* one that is *perceived*, although only the now-point actually is. We follow this procedure because not only is the extension of the melody given point for point in an extension of the act of perception but also the unity of retentional consciousness still "holds" the expired tones themselves in consciousness and continuously establishes the unity of consciousness with reference to the homogeneous temporal Object, i.e., the melody. An Objectivity such as a melody cannot itself be originarily given except as "perceived" in this form.
>
> Husserl (1964, p. 60)

> Already in the "touch" we have just found three distinct experiences which subtend one another, three dimensions which overlap but are distinct: a touching of the sleek and of the rough, a touching of the things – a

[13] This would only work if the rank ordering of the intervals exhibited certain properties, such as weak transitivity of the ordering and monotonicity.

[14] I am using "traditional phenomenology" to refer to the phenomenological tradition that started with Husserl and Brentano and attempted to describe human experience. I have left Dennett's (1992) heterophenomenology out of this discussion, which is a third person method for gathering the phenomenological descriptions of subjects: "It involves extracting and purifying texts from (apparently) speaking *subjects*, and using those texts to generate a theorist's fiction, the subject's *heterophenomenological world*. This fictional world is populated with all the images, events, sounds, smells, hunches, presentiments, and feelings that the subject (apparently) sincerely believes to exist in his or her consciousness. Maximally extended, it is a neutral portrayal of exactly *what it is like to be* that subject – in the subject's own terms, given the best interpretation we can muster." (Dennett 1992, p. 98).

> passive sentiment of the body and of its space – and finally a veritable touching of the touch, when my right hand touches my left hand while it is palpitating the things, where "the touching subject" passes over into the rank of the touched, descends into the things, such that the touch is formed in the midst of the world and as it were in the things. Between the massive sentiment I have of the sack in which I am enclosed, and the control from without that my hand exercises over my hand there is as much difference as between the movements of my eyes and the changes they produce in the visible. And as, conversely, every experience of the visible has always been given to me within the context of the movements of the look, the visible spectacle belongs to the touch neither more nor less than do the "tactile qualities." We must habituate ourselves to think that every visible is cut out in the tangible, every tactile being in some manner promised to visibility, and that there is encroachment, infringement, not only between the touched and the touching, but also between the tangible and the visible, which is encrusted in it, as conversely, the tangible itself is not a nothingness of visibility, is not without visual existence. Since the same body sees and touches, visible and tangible belong to the same world. It is a marvel too little noticed that every movement of my eyes – even more, every displacement of my body – has its place in the same visible universe that I itemize and explore with them, as, conversely, every vision takes place somewhere in the tactile space. There is double and crossed situating of the visible in the tangible and of the tangible in the visible; the two maps are complete, and yet they do not merge into one. The two parts are total parts and yet are not superposable.
>
> Merleau-Ponty (1995, p. 134)

The first thing to note about these examples, is that they are based on *first-person* introspection, in which the phenomenologist examines his or her experiences and writes down a description in human language. At the current stage of development, artificial systems are fairly rudimentary and incapable of describing any phenomenal states that they might have. This forces synthetic phenomenology to start with *third-person* objective measurements, which can be combined with type II theories of consciousness to make predictions about the system's phenomenal states.[15] These objective measurements are generally carried out on a subset of the system, such as its artificial neural networks or the code implementing a global workspace

[15] This approach is similar to neurophenomenology (see Section 4.5), which attempts to make predictions about people's first person phenomenology on the basis of objective brain measurements.

architecture, which is analyzed *as if* it were a mind capable of representations and phenomenal consciousness. To clarify this transition from the physical to the mental, Section 4.3.2 sets out a definition of a mental state, which applies at the physical level and can be used to interpret artificial as well as natural systems.

A second feature of traditional phenomenology is that it is based on objective features of the world that can be physically measured and experienced by more than one person - for example, the sound waves in Husserl's melody can be recorded with scientific instruments and Merleau-Ponty's touching and touched hands are physical as well as phenomenal objects.[16] This suggests that phenomenal experiences can be interpreted as *representations* of objects that appear in other peoples' streams of experience, and these objects can be probed in a variety of different ways. This interpretation of phenomenal experiences as representations is very useful when we are describing the phenomenology of artificial systems, with the difference that we have to find a way of identifying representations from a third person perspective. To address this problem, a definition of a representational mental state is given in Section 4.3.3, and Section 4.3.4 discusses some of the ways in which representational mental states can be identified in artificial systems.

A third observation about these quotations is that Husserl and Merleau-Ponty are describing *conscious* mental states and do not consider the many unconscious mental states that are in their minds. Section 4.3.5 explains how a theory of consciousness (based on type II correlates of consciousness) can be used to make predictions about the association between mental states and phenomenal states. Finally, Husserl and Merleau-Ponty are describing states that are *integrated* together into a *single* consciousness, and this question about the relationships between mental states is briefly covered in Section 4.3.6. The outcome of this process is a set of physical descriptions of representational and non-representational mental states that are

[16] This notion of a physical world would be interpreted with caution by traditional phenomenology, which often claims that the phenomenal is more primordial than the physical – see Husserl (1960).

associated with phenomenal states and Section 4.4 suggests how these can be turned into a full phenomenological description.

4.3.2 Mental States

Homeric man believed that the seat of human consciousness was in the heart and lungs (Onians 1973) and over thousands of years people have gradually come to associate human consciousness with human brains. Although many philosophers would argue that mental states are conceptually distinct from physical states, the increase in our knowledge about the brain, and the constant reduction of our mental functions to brain functions has led Churchland (1989) to suggest that the term "mental state" will eventually become redundant and our use of mental terminology will be superseded by descriptions in terms of states of the brain – a position known as eliminative materialism.[17] In the human case, this may eventually occur because a clear link has been established between mental states and the brain. However, synthetic phenomenology is analysing systems without biological brains and it is far from clear which part of the system is the right place to look for phenomenal states. Within this context we need the concept of a mental state to specify the part of the system (or subset of the system's states) that we are analysing for consciousness. For this purpose I will use "mental state" according to the following definition:[18]

A mental state is a state of the part of the system that is being analysed for consciousness. (4.1)

When people analyze humans for consciousness they generally focus on the brain and human mental states are usually taken to be states of human brains. Within the human brain,

[17] Rorty's (1979, p. 71) thought experiment in which Antipodeans use brain descriptions instead of mental terms to express their inner states is an example.

[18] This differs from Metzinger's (2003) definition, which links mental states to phenomenal accessibility.

work on the neural correlates of consciousness has shown that neural activity is important for consciousness, and so mental states can be defined in terms of the neuron's firing rates, the timing of their spikes or other properties of the neurons. However, it is also possible to analyse other parts of the human body for phenomenal states. For example, we can examine the liver or blood for consciousness, and when we do this, different states of the liver or blood become mental states according to this definition.[19]

In artificial systems a mental state can be a pattern of firing activity in simulated neurons or a sequence of 1s and 0s in the computer's RAM - for example, mental states could be monitored in Franklin's IDA (see Section 3.5.6) by using a debugger to measure changes in the memory. Different ways of defining a system's mental states may lead to different predictions about its phenomenology, which can be tested by monitoring its behaviour.

In this thesis mental states will be described at the physical level, either in physical terms or in terms that can easily be mapped down to physical descriptions without any loss of meaning or information. These states of the physical world can be identified within our phenomenal world by making phenomenal measurements of some region of the physical system and defining any states that take place within this region as mental.[20]

4.3.3 Representational Mental States

Some mental states are systematically related to features of the world. "Representation" is a natural way of describing this relationship, but since it is a controversial word, I will use it in a very restricted way in this thesis according to the following definition:

[19] See Holcombe and Paton (1998) and Paton et al. (2003) for a discussion of the computations carried out by the liver and other tissues.

[20] Mental states can also be a particular class of states that are not physically distinct – for example, neurons firing at 40 Hz could be classified as mental states.

A representational mental state is functionally or effectively connected to other mental states or to the data that is entering or leaving the system. (4.2)

Within a neural network functional connectivity is defined by Sporns et. al. (2004) as a statistical relationship between two neurons that may or may not be due to a causal relationship between them - for example, two neurons that share mutual information are said to be functionally connected. Effective connectivity describes the set of causal effects that one neuron has on another and it can be inferred experimentally by perturbing one part of the system or by observing the temporal ordering of events. Whilst Sporns et al. (2004) apply their definition of functional and effective connectivity to neuronal units, in this thesis it will be applied to all mental states and to the data that is entering and leaving the system.[21] It is also important that a representational mental state is distinguished from the state that is being represented - or it would no longer be a representation, but the thing itself. Some of the ways in which representational mental states can be identified are discussed in the next section.

Representational mental states do not necessarily *resemble* what they represent, although this is not excluded by Definition 4.2.[22] They are also different from depiction in Aleksander's (2005) sense. Depictions are mental states that are systematically related to both motor and visual information, whereas the definition of representation that I am using here is much broader and includes all mental states that are functionally or effectively connected to other mental states or to features of the system's incoming and outgoing data. The relationship between language and representation is not covered by this definition, although it may be possible to analyse language using this approach.

[21] The outgoing data is included to cover cases in which the system is representing its own motor activity.

[22] The question about representation and resemblance is a large topic that is beyond the scope of this thesis. A discussion of resemblance can be found in Gamez (2007c, pp. 71-83) and it is also worth pointing out that the interpretation of the phenomenal and the physical that was presented in Chapter 2 provides a strong argument against the idea that phenomenal experiences associated with representational mental states resemble the physical world in any way.

This definition of representation is extremely broad and can be applied to any system. Even a stone sustains transient internal vibrations in response to a blow that can be interpreted as representational mental states. However, systems do exhibit substantial differences in the complexity of their representations. For example, humans have a vast repertoire of states linked to incoming light, whereas stones generate almost no internal states in response to light.

Many systems contain non-representational mental states. One candidate for a non-representational state was put forward by Block (1995), who claimed that the phenomenal content of orgasm is non-representational. This is not a particularly good example because the phenomenal content of orgasm can readily be interpreted as a representation of the internal states of a person's body, genitalia and emotion system. However, other human mental states are likely to be non-representational, such as the ones regulating breathing and the states corresponding to spontaneous neuron activity. The same is likely to be true of many artificial systems.

Mental states that represent other mental states can also respond to complex features of the world. For example a mental state that is functionally or effectively connected to mental states that respond to combinations of lines could become active when the system is presented with a cube. In this case the 'meta representation' is representing *both* the mental states responding to the lines *and* the presence of a cube in the world. Mental states that represent non-representational mental states lack this double level of representation.

Representations are most easily identified when the system is actively processing information from its environment. Under these conditions, the internal states can be measured and correlated with features of the data entering and leaving the system. At a later point in time these representational mental states might become activated when the stimulus is no longer present in a way that is analogous to imagination. Systems with language can be probed for these offline representations by asking them what they are imagining, but without this kind of first person report it is difficult to identify unclassified representational mental states when the system is not actively processing the stimulus.

4.3.4 Identification of Representational Mental States

The general procedure for identifying representational mental states is to expose the system to a variety of different stimuli, record its responses, and look for functional or effective connections between the stimuli and the mental states.[23] To carry this out successfully, a comprehensive test suite needs to be designed that can probe a reasonable selection of the sensitivities of the system and specify them as precisely as possible. This could start with simple low level features, such as points, lines, and edges and work its way up to more abstract stimuli, such as faces and houses. All of these single modality tests would have to be combined with other modalities, such as audition, proprioception and sensation, and they would have to be carried out whilst the system is engaged in different activities, such as looking to the left, moving forward, and so on, to take account of sensorimotor contingencies. With even a moderately complex system this will soon escalate into an unmanageable number and complexity of tests. Some of these challenges could be met by appropriate subsampling of the test space and many tests can be automated by simulating input to the sensors. Common sense can also be used to prune the test suite down to a manageable size. This problem of scale will also appear in our animal and human tests as we improve our scanning and recording technologies.

The use of electrodes to identify representational mental states in animal and human subjects was pioneered by Hubel and Wiesel (1959), who inserted electrodes into the brains of cats and measured the activity of the neurons when different stimuli were presented in different

[23] One potential problem is that a system's representational mental states may change over time and it may have to be retested at regular intervals or have its adaptivity frozen whilst the description of its synthetic phenomenology is taking place.

parts of the visual field. Neurons whose activity changed[24] when the external stimulus was presented were judged to be representing the information in the stimulus. More recently a similar approach was pursued by Quian Quiroga et al (2005), who used electrodes to record from neurons in the medial temporal lobe in eight human subjects, who were presented with pictures of individuals, landmarks or objects. These experiments identified neurons that responded[25] to highly specific stimuli - for example one unit only responded to three completely different images of the ex US president Bill Clinton and another responded to pictures of the basketball player Michael Jordan.

The main limitation of using electrodes to identify representational mental states in human subjects is that simultaneous recording is only possible from a few hundred out of the billions of neurons in the brain. An alternative approach is to use scanning techniques, such as fMRI, PET, MEG and EEG to record the response of different brain areas as stimuli are presented. One example of this type of work is Haxby et al. (2001), who used fMRI to record the activity in the ventral temporal cortex while subjects viewed faces, cats, five categories of man-made objects and nonsense pictures. The distinct pattern of response that was identified for each category of object was linked by Haxby et al. (2001) to the presence of widely distributed and overlapping representations of faces and objects in the ventral temporal cortex. The main limitation of the scanning approach is that current procedures have limited spatial resolution – for example, fMRI measures the average activity within voxels of the order of 1 mm^3 - and so they can only be used to identify the general areas that hold representational mental states.

With artificial systems one generally has full access to their internal states and incoming/ outgoing data, and they can be probed precisely for all of their representations. Previous work in this area includes the backtracing method developed by Krichmar et. al. (2005), which finds

[24] This change could take several forms, such as an increase in firing rate, a decrease in firing rate or a burst of spikes in response to the onset or offset of the stimulus.

[25] A response was considered significant if it was larger than the mean plus 5 standard deviations of the baseline and had at least two spikes in the post-stimulus time interval (300–1000 ms).

functional pathways by choosing a reference neuronal unit at a specific time and then identifying the neuronal units connected to the reference unit that were active during the previous time step. This procedure is then repeated with the new list of neuronal units until the input neurons are reached that initiated the internal activity. Since the response characteristics of the input neurons are known, backtracing can be used to link internal states of the system to the stimuli presented to its sensors. Another way of identifying representational mental states in an artificial system is Granger causality, which is a method based on prediction that has been used by Seth (2007) to link a system's input to changes in its internal states. If a signal X_1 causes a signal X_2, then past values of X_1 should contain information that helps predict X_2 over and above the information contained in past values of X_2 alone. X_1 is said to Granger cause X_2 if the prediction errors in X_2 are reduced by the inclusion of X_1.[26] In this thesis representational mental states were identified using a method based on Tononi and Sporns (2003), in which noise was injected into the input or output layers and the mutual information that was shared between the input/ output and internal layers was measured. The full details of this procedure are given in Section 7.3.3.

4.3.5 Which Mental States are Phenomenally Conscious at Time *t* ?

At any point in time, many of a system's representational and non-representational mental states are *unconscious* (see Section 2.7.2), and to describe the *phenomenology* of the system a theory of consciousness is needed to predict which of the physically defined mental states are associated with phenomenal states. Since type I PCCs have been incorporated into the system's OMC rating, this separation between conscious and unconscious states is carried out using type II theories of consciousness. In this thesis I am using Tononi's theories about information integration, Aleksander's axioms and Metzinger's constraints (see sections 2.6.2 – 2.6.4) to make predictions about phenomenal states. Each of these theories can be used to predict which parts of

[26] More information about how Granger causality is calculated can be found in Seth (2007).

the system are conscious at time *t*, and these instantaneous predictions can be put together in a sequence to describe the evolution of the system's phenomenology over time. The details about how these theories are applied to the neural network developed by this project are given in Chapter 7.

Although I have decided to focus on the work of Tononi, Aleksander and Metzinger, the methodology described in this thesis is completely general and can be used with other type II theories of consciousness to make predictions about which mental states are associated with phenomenal states. It is highly likely that different theories of consciousness will make different predictions, and it may eventually be possible to discriminate between type II theories of consciousness by comparing their different predictions with first-person reports or the system's behaviour.

4.3.6 Integration Between Mental States

A description of the phenomenology of a system also has to identify the *relationships* between mental states, which determine how the mental states are integrated together into one or more consciousnesses. For example, consider a system that is looking at a red cube and has conscious representational mental states that respond to red information and conscious representational mental states that respond to shape information. If the colour and shape information is integrated or bound together, then it might be reasonable to claim that the system is conscious of a red cube. However, if the information is not integrated together, then it would be more accurate to say that there are two separate consciousnesses in the system: one that is conscious of redness, and another that is conscious of a cube. In humans, the importance of the integration between mental states is illustrated by the work on split brain patients (Gazzaniga, 1970), which suggests that two substantially independent consciousnesses are created when the corpus callosum is cut in the human brain, and the phenomenology of these two consciousnesses is likely to be very different from that of a normal person. The integration between mental states can be identified using methods for measuring functional and effective connectivity, such as Granger causality (Seth 2007) and information integration (Tononi and Sporns 2003).

4.4 XML Description of the Phenomenology

4.4.1 Introduction

This section explains how information about a system's OMC rating and mental states can be integrated into a description of its phenomenology as it interacts with the world. A major problem with describing the phenomenology of artificial systems is that the words and structures of human languages are adapted to the description of human states. This problem is covered in Section 4.4.2 and Section 4.4.3 suggests why a markup language, such as XML, is more appropriate for synthetic phenomenology. Section 4.4.4 then outlines the structure of the XML that I will be using to describe the phenomenology of a neural network in this thesis. After a brief discussion of the use of XML to describe phenomenology, Section 4.4.6 looks at how this approach to synthetic phenomenology relates to the interpretation of the science of consciousness that was outlined in Section 2.4.5.

4.4.2 Problems Describing the Phenomenology of Non-Human Systems

Traditional phenomenology, especially in the work of Husserl (1960) and Heidegger (1995a), derives its significance from the claim that the phenomena we experience are as important and substantial as the physical world described by science, which is often portrayed as a secondary interpretation of our experiences. In this way traditional phenomenology sets itself up with an 'objective' field of phenomena that are assumed to be the same for everyone and can be unproblematically described in natural human language The problem with this approach is that these assumptions about common experience start to break down once phenomenology is applied to the experiences of infants, animals and robots. To illustrate this problem, I will consider a short extract from Wordsworth (2004), which contains a fairly straightforward description of daffodils in natural human language:

> When all at once I saw a crowd,
> A host, of golden daffodils,
> Beside the lake, beneath the trees,
> Fluttering and dancing in the breeze.

Most people have had the experience of daffodils fluttering and dancing in the breeze and when Wordsworth's description is read by humans, they can readily imagine a similar past experience and understand his words well enough. Although this description is reasonably straightforward, it is actually an extremely vague and imprecise way of communicating daffodil information, and each reader will imagine the flowers differently. More serious problems start to arise when we try to use ordinary language to describe the experiences of an infant placed in front of a field of daffodils. As Chrisley (1995) points out, we cannot simply say that the infant sees a host of golden daffodils because the infant has a preobjective mode of thought, which is unable to locate the daffodils within a single unified framework. Adults understand daffodils as something objectively located in three dimensional space, whereas infants do not necessarily continue to believe in the existence of the daffodils when they are occluded. In the adult and infant the word "daffodils" refers to two different concepts and experiences. As Chrisley puts it: "The infant's concepts are not fully objective and are therefore non-conceptual. To ascribe conceptual content to the infant in this case would mischaracterize its cognitive life and would not allow prediction or explanation of the infant's behavior." (Chrisley 1995, p. 145).

These problems become even more difficult when the attempt is made to describe the phenomenology of a non-human animal, such as Nagel's famous bat (Nagel 1974). When a bat flies over a field of daffodils it receives a complex pattern of returning ultrasound pulses, which are processed into phenomenal experiences that are likely to be very different from our own. Sentences like “the bat is experiencing a host of golden daffodils” are at best an extremely misleading description of the bat’s phenomenology.

The same difficulties are encountered by attempts to describe the phenomenal experiences of artificial systems. For example, a robot that is pointing its camera at a field of daffodils might have phenomenal states associated with mental states that are effectively connected to its camera’s response to yellow light (independently of the location, movement or shape of the light). However, we would have no basis for believing that the robot would have the *human* phenomenal experience of yellow when the daffodils were placed in front of it, or even that two different robots would have the same experience of yellow as each other. This problem becomes even more acute when a system has phenomenal states that are systematically related to features of the world that are *invisible* to human beings - for example, we have no words at all to describe mental states that respond to X rays.

One approach to this problem would be to describe the scene in front of the robot in the language of physics – for example, we could talk about the system having a representation of 590 nm electromagnetic waves, instead of talking about it experiencing yellow light,[27] and use the language of chemistry, biology and geometry to describe the features of the daffodils that the system is sensitive to. The trouble with this approach is that it does not describe the *phenomenology* of the system and it has the limitation that the data coming out of a system does not always lend itself easily to an objective physical description. For example, to describe the motor output signals that control an eye or arm one would have to come up with a physical description of the eye or arm and specify its movement relative to a frame of reference that

[27] There is not a straightforward link between wavelength of light and perceived colour and it is possible to experience yellow when there are no 590 nm electromagnetic waves present. This problem has been set aside in this thesis - in the future a more accurate physical description could specify all of the physical conditions under which we would experience yellow.

would also have to be physically described. Whilst this can be done, it is much easier to interpret the motor output signals as an eye or arm movement.

The pragmatic solution that will be followed in this thesis is to use both human and physical descriptions to describe a system's representational mental states, when these are possible and appropriate. The human description should be interpreted with caution (the phenomenology of an artificial system that only responds to yellow is likely to be very different from our human experience of yellow) and the physical description should only be taken as a starting point for a phenomenological description. In the future, it may be possible to create closer links between phenomenological and physical descriptions - perhaps by using the information characteristics of mental states (Tononi 2004) or by applying O'Regan and Noë's (2001) theories about sensorimotor contingencies.

4.4.3 Markup Languages for Synthetic Phenomenology

A combination of human and physical descriptions enables something to be said about the contents of an artificial system's phenomenal states, but it does not capture the *relationships* between them. Furthermore, depending on how mental states are defined for the system, there could be millions or even billions of active mental states that are predicted to be associated with consciousness at any point in time. Even if it was possible to integrate all of these mental states into a natural language description, the resulting document would be so long and tedious that it would be almost impossible to read.

One way of solving these problems is to abandon the attempt to describe the phenomenology of artificial systems in natural human language and use a markup language, such as XML or LMNL, to structure the descriptions of the representational mental states and to indicate the relationships between them. There are a number of reasons why a markup language would be a good choice for the description of an artificial system's phenomenology:

- Markup languages are much more precise and tightly structured than natural language, which enables markup languages to describe complex nested hierarchies and represent some of the relationships between different pieces of information.

- Markup languages can describe low level details of a system's hardware, but they can also abstract from them, so that high level comparisons can be made between machines with different architectures and between humans and machines. Whilst two systems' lower levels might be different – perhaps using neurons or silicon - the higher levels are likely to be more similar, which would allow direct comparisons between them once everything was encoded into a markup language.[28]

- Markup languages can be written and read by both machines and humans. With simple small-scale analyses it is useful to be able to manually read and edit a description of a machine's mental states. However, it is also relatively easy to automatically generate and analyse the states of a machine using a markup language, for example by writing programs that look for phenomenal states using different type II theories of consciousness.

- Data that has been structured using a markup language is typically stored in plain text files that can be shared between different operating systems and easily archived, either by converting them into a database or by storing them directly.

- The structure of some markup languages can be validated without prior knowledge of their form.

- Once you have a highly structured representation of a machine's mental states and a methodology for analysing them for phenomenal consciousness, it is possible to see how a machine's conscious states can be extended or enhanced.

[28] Coward and Sun (2007) claim that this type of hierarchical description is necessary for a science of consciousness.

- Markup languages are a good foundation for other techniques for representing non-conceptual mental states, such as the suggestions made by Chrisley (1995) about content realization, ability instantiation and self instantiation (see Section 4.5), which depend to some extent on a precise specification of states of oneself and the environment
- Markup languages can be very flexible. For example, in addition to tags and data, XML can contain references to external files, pieces of code and equations, which enables it to include features that cannot be precisely described in natural language.

Whilst a number of markup languages, such as JSON, LMNL, YAML and OGDL, would have been appropriate for synthetic phenomenology, the popularity of the eXtensible Markup Language (XML) and the availability of good parsers in most programming languages made it a good choice for illustrating this approach. In the future it might be necessary to change to a more sophisticated markup language, such as LMNL, which supports overlapping elements and structured attributes.[29]

4.4.4 Example XML Description

This section outlines the XML structure that will be used to describe the phenomenology of an artificial neural network in Chapter 7. This is only an example, rather than a fully fledged standard, because it is tailored to an approach in which individual neurons are interpreted as individual representational states, and the mutual information shared between each of the internal neurons and neurons in the input and output layers is calculated using the methodology described in Section 7.3.3. If XML is found to be a useful way describing the phenomenology of artificial systems, then it is hoped that a more general specification can be developed. This example does

[29] A good XML tutorial can be found at: http://www.w3schools. com/ xml/default.asp. More information about LMNL can be found here: http://lmnl.net/.

not include non-representational mental states and mental states that represent other mental states. As Chapter 7 shows, at the current stage of research it is hard enough to identify and describe mental states that are systematically related to states of the world, without trying to include mental states that are almost impossible to articulate in human language.

```
<!-- Standard XML header. -->[30]
<?xml version="1.0" encoding="ISO-8859-1"?>

<!-- Start of the analysis. -->
<analysis>

  <!-- General description of the contents of the file. -->
  <description>Synthetic phenomenology of the SIMNOS virtual robot.
                                                    </description>

  <!-- Author(s) of the file and date on which the analysis was generated. -->
  <author>David Gamez</author>
  <date>Mon Jan 28 14:44:27 2008</date>

  <!-- The system that is being analysed along with its version number. A full description of the
     system should be included in the source files. -->
  <system>SIMNOS version 1.0; SpikeStream version 0.1</system>

  <!-- Source files for the analysis. These include the files for the neural network (if there is
     one, since the system may not be neural) and the analysis files. Source files should always
     be included with the phenomenological description to enable other researchers to validate
     the predictions and generate their own description of the synthetic phenomenology. -->
  <source_files>
    <file>TrainedNeuralNetwork_version1.sql.tgz</file>
    <file>AnalysisRun1_NoiseRun1_NeuralArchive.sql.tar.gz</file>
  </source_files>

  <!--The archive that is being described. -->
  <archive>Analysis Run 1 [ 2007-12-18 20:42:55 ]</archive>

  <!-- The time step of the archive that is being analyzed or the time at which the data was
     captured from the system. -->
  <time_step>13194</time_step>

  <!-- Start of the phenomenological description. -->
  <phenomenology>
```

[30] XML comments start with "<!-- " and end with "-->".

```
<!-- The next part of the file lists the system’s mental states. These may be representational
  and they may be predicted to be conscious according a type II theory of consciousness. -->

  <!-- A mental state of the system. -->
  <mental_state>

    <!-- The OMC rating of the part of the system in which this mental state is instantiated,
      along with the version of the scale that is being used. -->
    <omc_scale>
      <rating>0.427</rating>
      <version>0.6</version>
    </omc_scale>

    <!-- In this example mental states are active neurons. -->
    <physical_description>
      <firing_neuron>
        <id>120811</id>
      </firing_neuron>
    </physical_description>

    <!--The cluster tag is used to indicate the functional or effective connectivity between
      this mental state and other mental states. Different methods can be used to measure
      this, such as information integration (Tononi and Sporns 2003). -->
    <cluster>
      <id>200809</id>
      <type>phi</type>
      <amount>75.1173</amount>
    </cluster>

    <!-- List of the states of the world that are functionally or effectively connected to this
      mental state. In this example, representational states are identified using the mutual
      information that is shared with neurons in the input or output layers – see Section
      7.3.3. -->
    <representations>

      <!-- This mental state is effectively connected to data leaving the system. -->
      <output>
        <neuron>
          <id>127936</id>
        </neuron>
        <mutual_information>0.993765</mutual_information>
        <human_description>Proprioception / motor output
                                              </human_description>
        <physical_description>N/A</physical_description>
      </output>

      <!-- This mental state is effectively connected to data entering the system. -->
      <input>
        <neuron>
```

```
          <id>104327</id>
        </neuron>
        <mutual_information>1.00854</mutual_information>
        <human_description>Red / blue visual input
                                                </human_description>
        <physical_description>700/450 nm electromagnetic waves
                                             </physical_description>
      </input>

      <!-- Further input and outputs can be added here. -->

    <!-- The end of the list of representations. -->
    </representations>

    <!-- Type II theories of consciousness are used to predict whether phenomenal
      consciousness is associated with this mental state. In this example, the predictions
      are made using Tononi's (2004), Aleksander's (2005) and Metzinger's (2003)
      theories. -->
    <phenomenal_predictions>

      <!-- Whether this mental state is part of the conscious part of the system according
        to Tononi's theory of consciousness (see Section 7.5 for the criteria for this). -->
      <tononi>0</tononi>

      <!-- Whether this mental state is part of the conscious part of the system according
        to Aleksander's theory of consciousness (see Section 7.6.2 for the criteria for
        this). -->
      <aleksander>0.993765</aleksander>

      <!-- Whether this mental state is part of the conscious part of the system according
        to Metzinger's theory of consciousness (see Section 7.7.3 for the criteria for this).
        -->
      <metzinger>75.1173</metzinger>

      <!-- Other phenomenal predictions can be added here. -->

    <!-- The closing tag of the phenomenal predictions. -->
    </phenomenal_predictions>

  <!-- The closing tag of the mental state. -->
  </mental_state>

  <!-- Any number of mental states can be added here. -->

<!-- The end of the description of the phenomenology of the system. -->
</phenomenology>
```

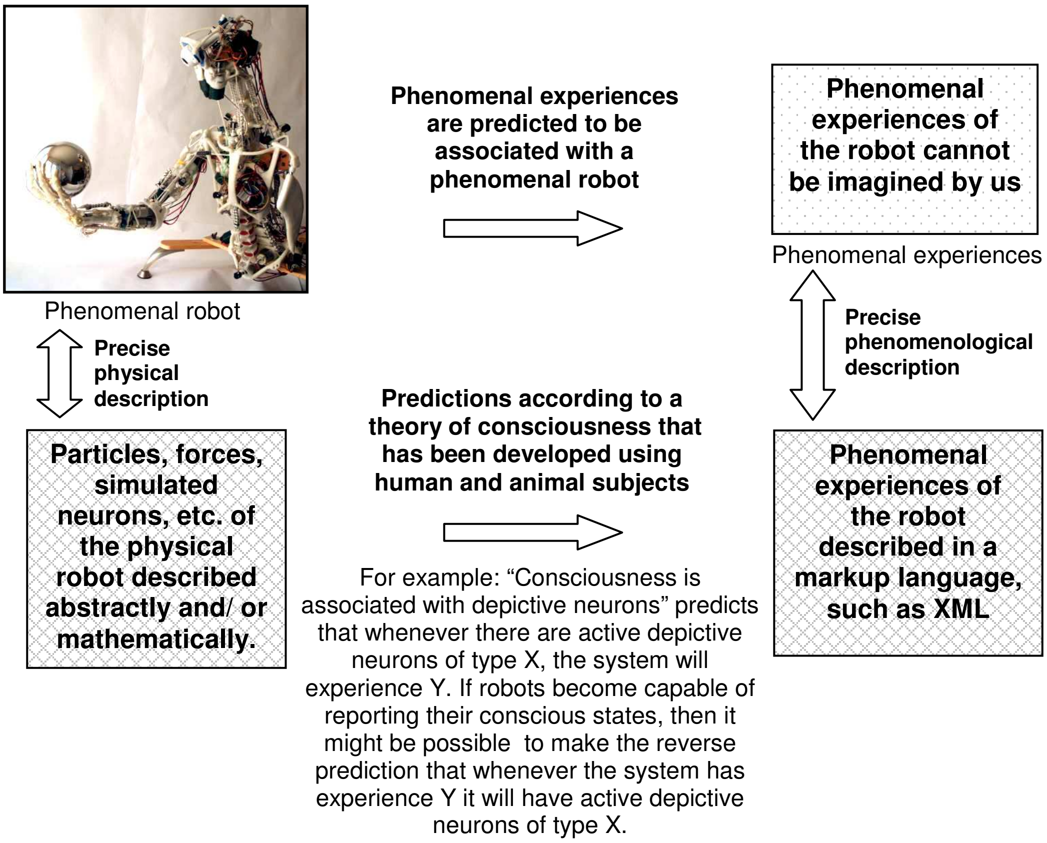
```
<!-- This final closing tag ends the analysis of the system. -->
</analysis>
```

4.4.5 A Description of the Synthetic Phenomenology?

Given the history of phenomenology, we might expect that the final outcome of synthetic phenomenology would be a natural language description. Even if we cannot achieve this at present, it might be thought that this should be the final goal of the procedures outlined in this chapter. Viewed from this perspective, the markup language would only be a preparatory stage that would help us to prepare a traditional phenomenological account of the experiences of COG, CRONOS or IDA. However, the problems discussed in Section 4.4.2 make it unlikely that we are ever going achieve fluid natural language descriptions of the phenomenology of non-human systems. Instead, it might be much better to treat the XML as the best description that we are going to get of the phenomenology of an artificial system. We don't have adequate words in human language to describe a system that can only experience vertical lines, but we can represent such a system accurately using XML, and by looking at the XML we can start to understand how much and how little we can imagine what it is like to be such a system. Some of the issues raised by the use of XML in synthetic phenomenology are covered in Section 7.9.9.

4.4.6 Synthetic Phenomenology and Science

This section takes a brief look at how this approach to synthetic phenomenology fits in with the approach to the science of consciousness that was put forward in Section 2.4.5. The main difference between the study of human consciousness and synthetic phenomenology is that robots are currently unable to describe their conscious states, and so we can only make *predictions* about their consciousness based on theories that have been developed using humans and animals.


Phenomenal robot
Phenomenal experiences are predicted to be associated with a phenomenal robot
Phenomenal experiences of the robot cannot be imagined by us
Phenomenal experiences
Precise physical description
Precise phenomenological description
Particles, forces, simulated neurons, etc. of the physical robot described abstractly and/ or mathematically.
Predictions according to a theory of consciousness that has been developed using human and animal subjects
Phenomenal experiences of the robot described in a markup language, such as XML

For example: “Consciousness is associated with depictive neurons” predicts that whenever there are active depictive neurons of type X, the system will experience Y. If robots become capable of reporting their conscious states, then it might be possible to make the reverse prediction that whenever the system has experience Y it will have active depictive neurons of type X.

Figure 4.1. How synthetic phenomenology fits in with the approach to the science of consciousness that was put forward in Section 2.4.5. With artificial systems, it is only possible to make predictions about the phenomenal experiences that are associated with them, and so there are unidirectional arrows from the phenomenal robot to the robot’s phenomenal experiences and from the description of the physical system to the description of the robot’s phenomenology. This diagram should be contrasted with Figure 2.4 in Chapter 2, where the horizontal arrows are bidirectional because the association between phenomenal experiences and the phenomenal brain is the starting point for experiments on the correlates of consciousness and systematic relationships are being identified between the phenomenal and physical descriptions.

This situation is illustrated in Figure 4.1, in which the arrows between the robot and its phenomenal states and between the physical and phenomenal descriptions are only one way to indicate that phenomenal states are predicted to be associated with the robot. If we can develop robots that can report their conscious states, then it will be possible to validate these predictions and speak about an association between the phenomenal states and the robot.

4.5 Previous Work in Synthetic Phenomenology

This approach to synthetic phenomenology has been substantially influenced by the previous work in traditional phenomenology, such as Husserl (1960, 1964), Merleau-Ponty (1989, 1995) and Heidegger (1995a), which attempted to describe different aspects of human conscious experience from a first person perspective. These descriptions were carried out in natural language and generally took the position that the physical world is a secondary interpretation of our phenomenal experiences and not something to which our phenomenal experiences should be reduced. Although Heidegger (1995b) made some attempts to understand animal consciousness, the main emphasis of traditional phenomenology is on human phenomenal experience.

The question whether artificial systems are capable of conscious states has been extensively discussed in the literature on consciousness and the contributions roughly divide into those who accept the difficulties with behaviour-based attribution of phenomenal states, and those who have a theory of consciousness that enables them to make definite claims about which machines are phenomenally conscious. In the first group, Moor (1988) sets out the arguments against knowing for certain whether robots have qualia, but claims that we need to attribute qualia to robots in order to understand their actions. A similar position is set out by Harnad (2003), who claims that the other minds problem limits us to attributing consciousness on the basis of behaviour, and so any robot that passes the T3 version of the Turing test for a lifetime must be acknowledged to be conscious. Prinz (2003) is closest to the position of this thesis since he does not think that we can identify the necessary and sufficient conditions for consciousness and does not suggest other grounds for attributing consciousness to machines.

People who claim to know exactly what the causes or correlates of consciousness are can say precisely which machines are capable of phenomenal states - replacing the OMC scale set out in this chapter with a dividing line dictated by their theory of consciousness. One of the most liberal of these theories is Chalmers (1996), whose link between consciousness and information leads him to attribute phenomenal states to machines as simple as thermostats. At the other extreme, Searle (1980) believes that his Chinese room argument excludes the possibility that any of the functional levels could be simulated and Searle (2002) rather vaguely ties consciousness to a causal property of matter, so that only biological humans, animals and possibly aliens could be conscious. In between these positions are people like Aleksander and Morton (2007a), who set out two criteria that a system must conform to if it is to be a candidate for synthetic phenomenology: "To be *synthetically phenomenological*, a system S must contain machinery that represents what the world and the system S within it *seem* like, from the point of view of S." (Aleksander and Morton 2007a, p. 110). An unpacked version of this definition is used by Aleksander and Morton to argue that their own kernel architecture is synthetically phenomenological, whereas the global workspace architecture is not.

Once it has been decided which artificial systems are capable of phenomenal states (if any) the second question faced by synthetic phenomenology is how artificial phenomenal states can be described. One approach to this was put forward by Chrisley (1995), who set out a number of techniques for representing non-conceptual content. These include content realization, in which content is referred to by listing "perceptual, computational, and/or robotic states and/or abilities that realize the possession of that content" (Chrisley, 1995, p. 156), ability instantiation, which involves the creation or demonstration of a system that instantiates the abilities involved in entertaining the concept, and two forms of self instantiation, in which the content is referred to by pointing to states of oneself or the environment that are linked to the presence of the content in oneself. More recently Chrisley and Parthemore (2007) used a SEER-3 robot to specify the non-conceptual content of a model of perception based on O'Regan and Noë's (2001) sensorimotor contingencies. Initially the robot had no expectations about what it was going to see and as it moved its eye around it built up expectations about what it would see if it were to move its eye to a particular position. These expectations were plotted for each position in visual space to generate a graphical representation of the robot's visual experience. Chrisley and Parthemore used this representation to evaluate some aspects of O'Regan and Noë's (2001) theory, such as their interpretation of change blindness and how visual experience appears to be coloured at the periphery despite the lack of colour receptors outside the fovea. Other graphical representations of a robot's inner states have been produced by Holland and Goodman (2003) and Stening et al. (2005), who plotted the sensory and motor information stored in a Khepera's concepts. More details about this work are given in Section 3.5.5.

Synthetic phenomenology has a number of overlaps with the description of human phenomenology from a third person perspective. This type of research is commonly called "neurophenomenology", although this term is subject to two conflicting interpretations. The first interpretation of "neurophenomenology" was put forward by Varela (1996), who used it to describe a reciprocal dialogue between the accounts of the mind offered by science and phenomenology.[31] This type of neurophenomenology emphasises the first person human perspective and it has little in common with synthetic phenomenology. However, neurophenomenology can also be interpreted as the description of human phenomenology from a third person perspective using measurements of brain activity gathered using techniques, such as fMRI, EEG or electrodes. Good examples of this type of work are Kamitani and Tong (2005), Haynes and Rees (2005a,b) and Kay et al. (2008), who used the patterns of intensity in fMRI voxels to make predictions about the phenomenal states of their subjects. In some ways neurophenomenology is easier than synthetic phenomenology because it does not have to decide whether its subjects are capable of consciousness and the description of non-conceptual states is considerably simpler in humans. However, both disciplines are attempting to use external data to identify phenomenal states in a system and there is considerable potential for future collaboration between them.

[31] A review of this interpretation of neurophenomenology can be found in Thompson et al. (2005) and it had a substantial influence on the analysis of consciousness in Chapter 2.

4.6 Conclusions

This chapter has set out an approach to synthetic phenomenology that can be used to describe a machine's predicted phenomenal states. Since the link between type I PCCs and consciousness cannot be empirically established, the first part of this chapter outlined an OMC scale, which models our subjective judgement about the relationship between type I PCCs and consciousness. The next part of this chapter developed concepts of a mental state and a representational mental state and outlined how these could be identified in a system and used to make predictions about phenomenal states using type II theories of consciousness. Problems with the description of artificial phenomenal states in human language were then discussed and it was suggested how a markup language, such as XML, could be used to describe the phenomenal states of artificial systems.

The next chapter outlines the design and implementation of a neural network that is based on the some of the theories of consciousness set out in Chapter 2. The approach to synthetic phenomenology that has just been described is used to make predictions about the consciousness of this network in Chapter 7.

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:

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

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:

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

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

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.

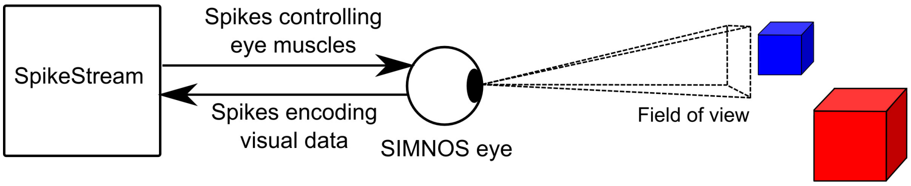


Figure 5.1. Experimental setup with the eye of SIMNOS in front of red and blue cubes

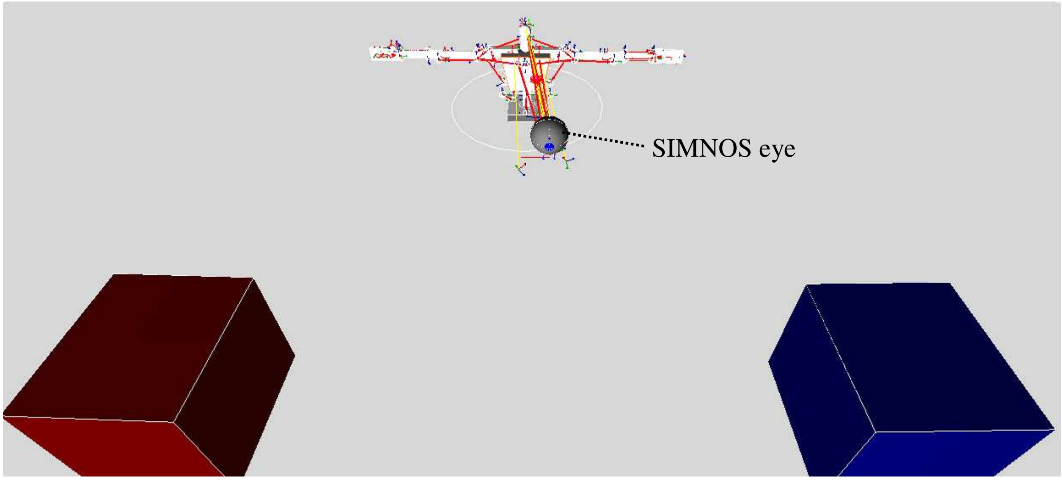


Figure 5.2. Screenshot of SIMNOS in front of the red and blue cubes

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.

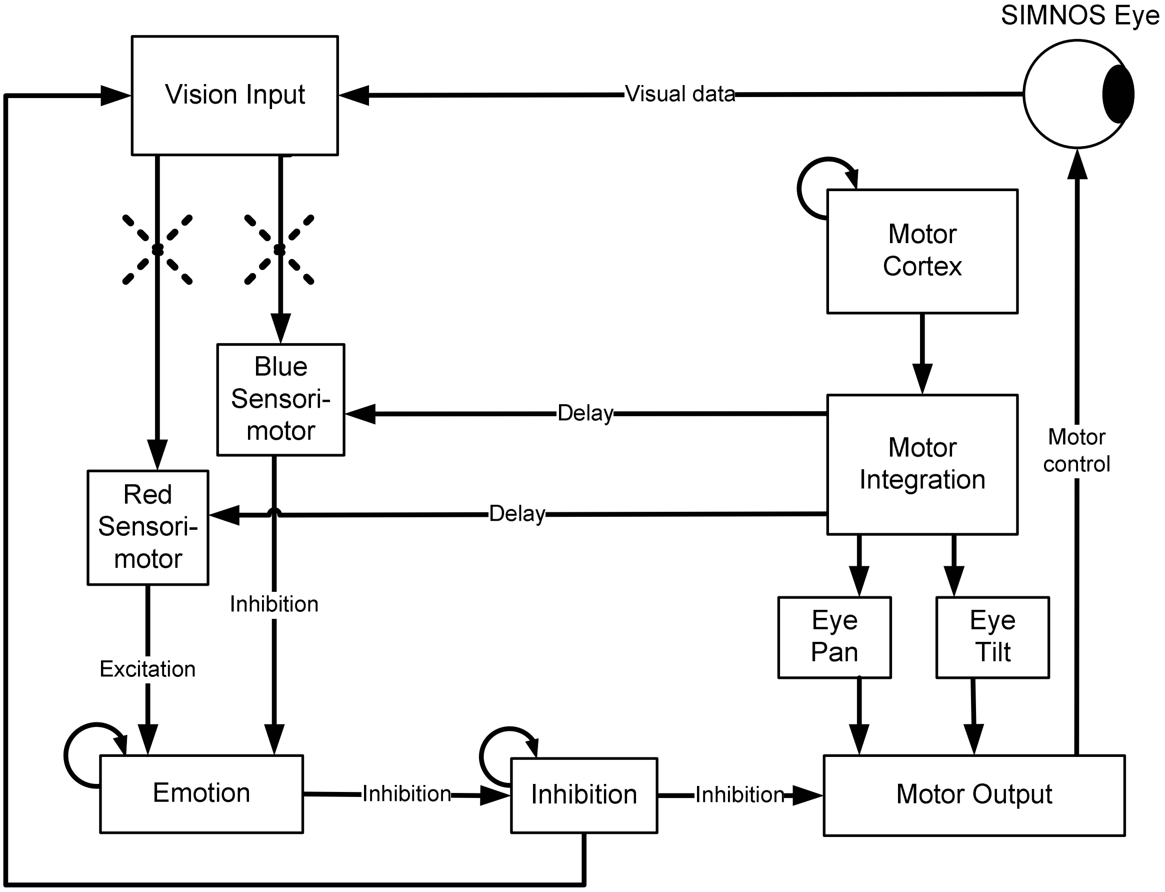


Figure 5.3. Neural network with SIMNOS eye. Arrows indicate connections within layers, between layers or between the neural network and SIMNOS. The connections marked with dotted crosses were disabled for the imagination test in Section 5.5.2.

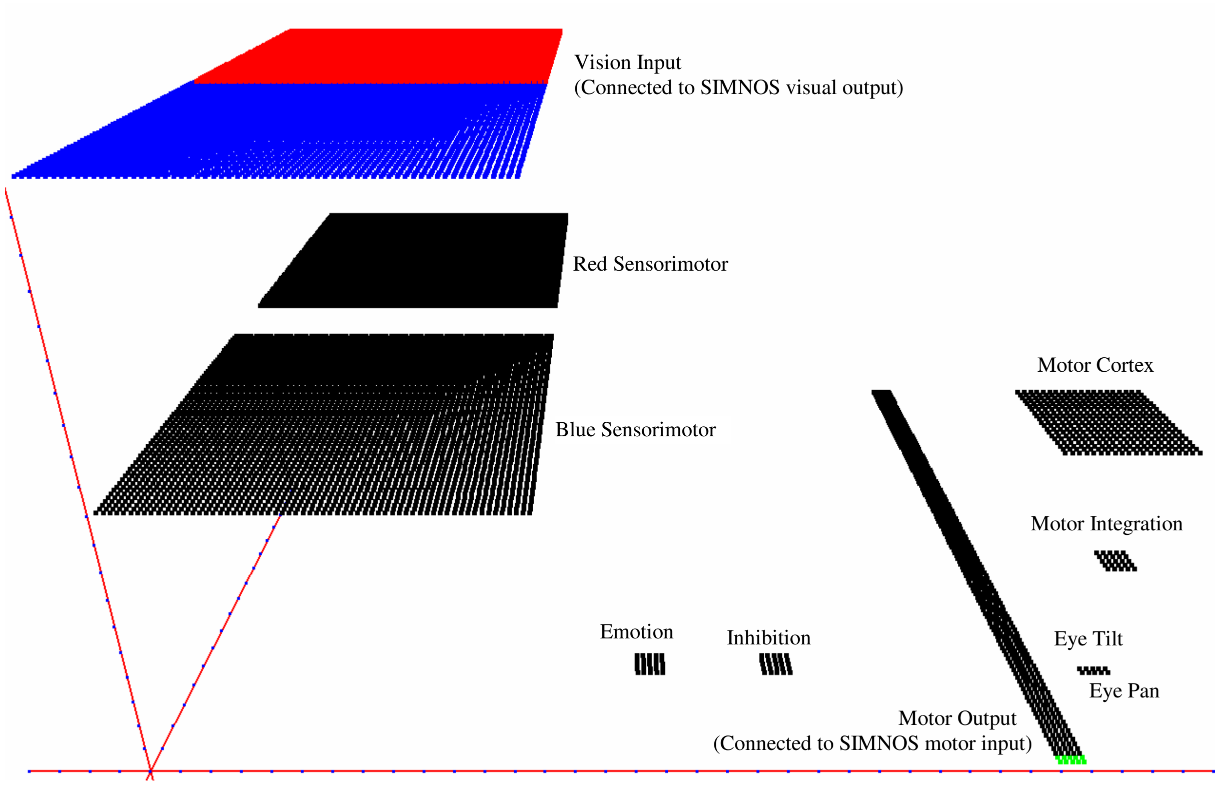


Figure 5.4. The network in SpikeStream. The red and blue sensitive parts of Vision Input are highlighted in red and blue. The neurons in Motor Output that set the pan and tilt of SIMNOS's eye are highlighted in green.

	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.

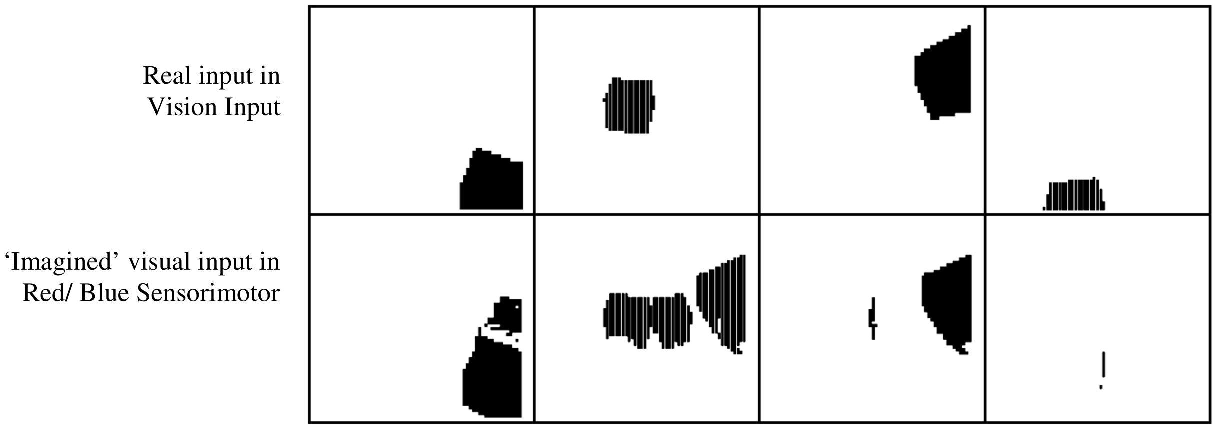


Figure 5.5. Examples of the contrast between real visual input (top row) and imagined visual input (bottom row)

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.

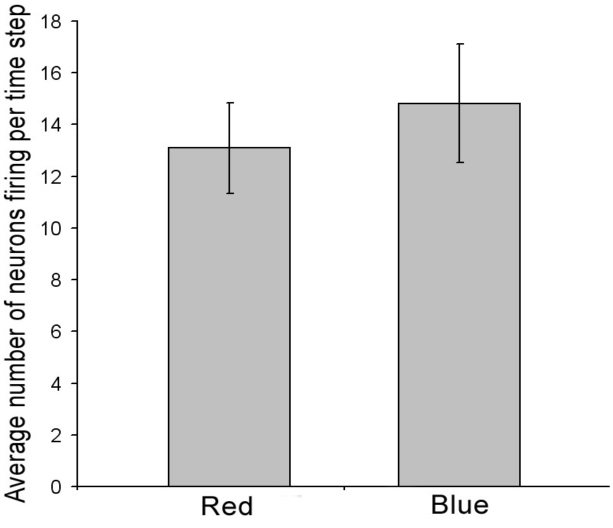


Figure 5.6. Average number of neurons firing per time step in the red and blue sensitive parts of Vision Input when the cognitive mechanisms associated with consciousness were disabled [15]

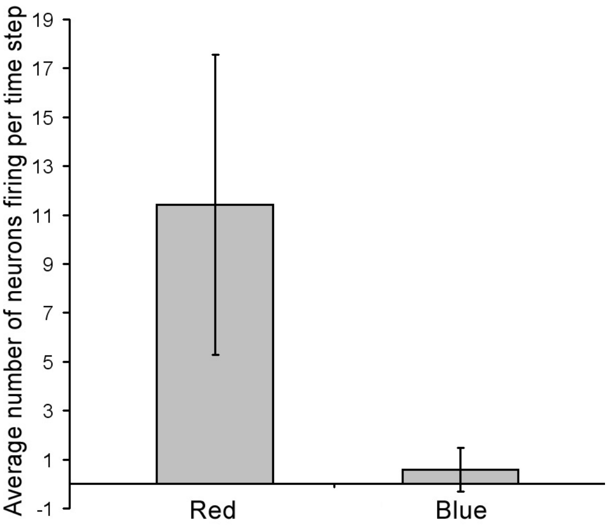


Figure 5.7. Average number of neurons firing per time step in the red and blue sensitive parts of Vision Input when the cognitive mechanisms associated with consciousness were enabled [16]

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.

6. SpikeStream[1]

6.1 Introduction

This chapter outlines the spiking neural simulator that was developed to model the 'conscious' neural network described in Chapter 5. This simulator had to be fast, it had to exchange spikes with the SIMNOS robot, it needed to support different neuron and synapse models, and the ability to record a network's activity was essential for the synthetic phenomenology in Chapter 7. A substantial amount of fine tuning of the network was required, and so an intuitive graphical interface with good monitoring facilities was also desirable.

Since none of the available simulators met these requirements (see Section 5.2.4), I developed a new spiking neural simulator, SpikeStream, that can be used to edit, display and simulate up to 100,000 neurons. This simulator uses a combination of event-based and synchronous simulation and stores most of its information in databases, which makes it easy to run simulations across an arbitrary number of machines. A comprehensive graphical interface is included and SpikeStream can send and receive spikes to and from real and virtual robots across a network. The architecture is highly modular, and so other researchers can use its graphical editing facilities to set up their own simulations or use their own code to create networks in the SpikeStream databases.

The first part of this chapter outlines the different components of the SpikeStream architecture and sets out the features of the graphical interface in more detail. Next, the performance of SpikeStream is documented along with its communication with external devices. The last part of this chapter suggests some applications for SpikeStream, describes its limitations and gives details about the SpikeStream release under the terms of the GPL license. Much more

[1] An earlier version of this paper was published as Gamez (2007b).

detailed information about the features and operation of SpikeStream is given in the SpikeStream Manual, which is included as the first appendix to this thesis.

6.2 Architecture

SpikeStream is built with a modular architecture that enables it to operate across an arbitrary number of machines and allows third party applications to make use of its editing, archiving and simulation functions. The main components of this architecture are a number of databases, the graphical SpikeStream Application, programs to carry out simulation and archiving functions, and dynamically loaded neuron and synapse classes.

6.2.1 Databases

SpikeStream is organized around a number of databases that hold information about the network model, patterns and devices. This makes it easy to launch simulations across a variable number of machines and provides a great deal of flexibility in the creation of connection patterns. The SpikeStream databases are as follows:

- *Neural Network.* Each neuron has a unique ID and connections between neurons are recorded as a combination of the presynaptic and postsynaptic neuron IDs. The available neuron and synapse types along with their parameters are also held in this database.
- *Patterns*. Holds spatiotemporal patterns that can be applied to the network for training or testing.
- *Neural Archive*. Stores archived neuron firing patterns. Each archive contains an XML description of the network and data in XML format.
- *Devices*. The devices that SpikeStream can exchange spikes with over the network.

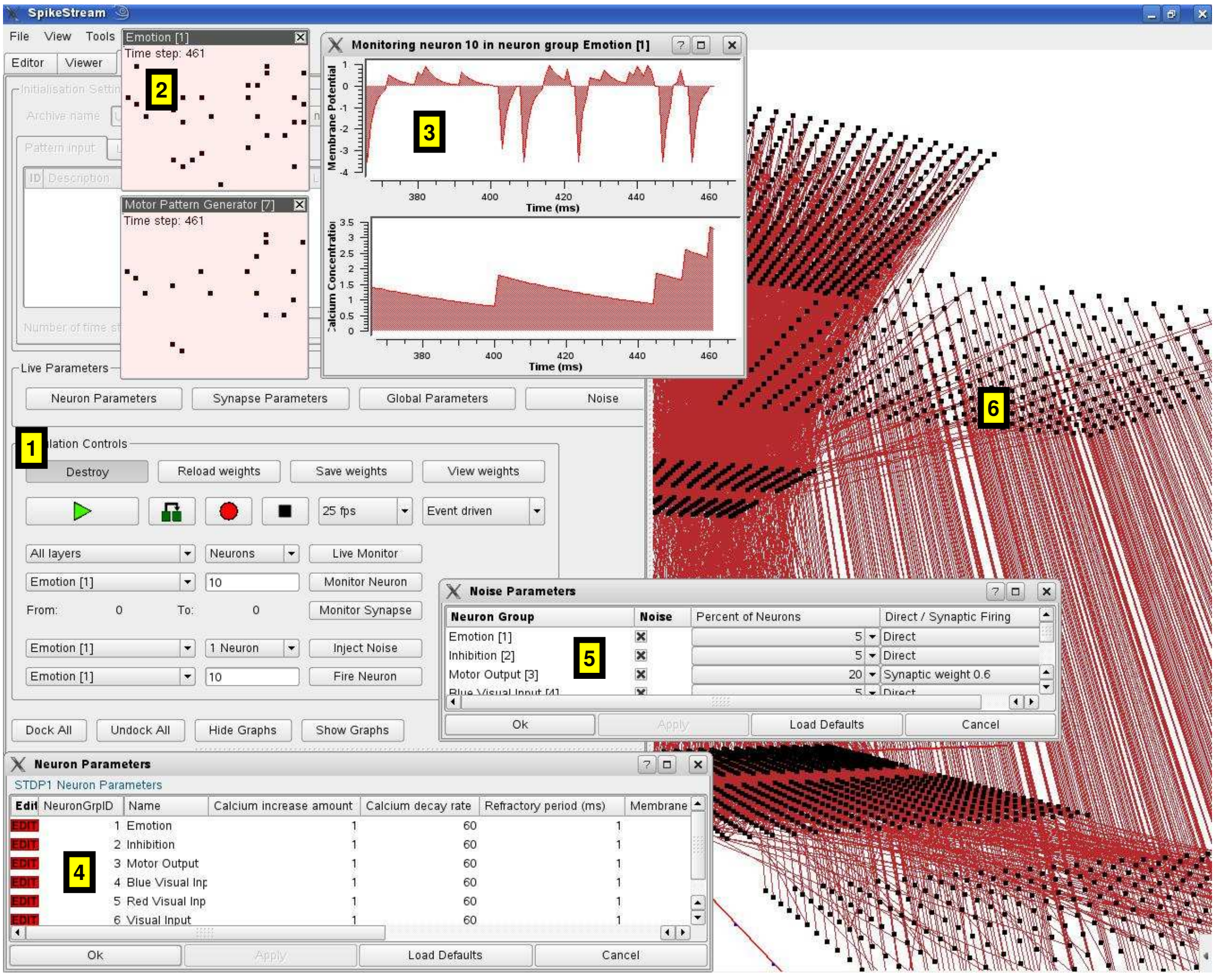
These databases are edited by SpikeStream Application and used to set up the simulation run. They can also be edited by third party applications - to create custom connection patterns or neuron arrangements, for example - without affecting SpikeStream's ability to visualize and simulate the network.

6.2.2 SpikeStream Application

An intuitive graphical user interface has been written for SpikeStream (see Figure 6.1) with the following features:

- *Editing.* Neuron and connection groups can be created and deleted.
- *3D Visualisation.* Neuron and connection groups are rendered in 3D using OpenGL and they can be rotated, selectively hidden or shown, and their individual details displayed. The user can drill down to information about a single synapse or view all of the connections simultaneously.
- *Simulation.* The simulation tab has controls to start and stop simulations and vary the speed at which they run. Neuron and synapse parameters can be set, patterns and external devices connected and noise injected into the system.
- *Monitoring.* Firing and spiking patterns can be monitored and variables, such as a neuron's voltage, graphically displayed.
- *Archiving.* Archived simulation runs can be loaded and played back.

Much more information about the graphical features of SpikeStream can be found in Appendix 1.



Noise Parameters

Neuron Group	Noise	Percent of Neurons	Direct / Synaptic Firing
Emotion [1]	☒	5	Direct
Inhibition [2]	☒	5	Direct
Motor Output [3]	☒	20	Synaptic weight 0.6
Blue Visual Input [4]	☒	5	Direct

Ok Apply Load Defaults Cancel

Neuron Parameters

STDP1 Neuron Parameters

Edit	NeuronGrpID	Name	Calcium increase amount	Calcium decay rate	Refractory period (ms)	Membrane
EDIT	1	Emotion	1	60	1	
EDIT	2	Inhibition	1	60	1	
EDIT	3	Motor Output	1	60	1	
EDIT	4	Blue Visual Inp	1	60	1	
EDIT	5	Red Visual Inp	1	60	1	
EDIT	6	Visual Input	1	60	1	

Ok Apply Load Defaults Cancel

Figure 6.1. SpikeStream graphical user interface. The numbers highlighted in yellow indicate the following features: (1) Simulation controls, (2) Dialog for monitoring the firing of neurons in a layer, (3) Dialog for monitoring variables inside the neurons, such as the calcium concentration and voltage, (4) Dialog for viewing and setting neuron parameters (5) Dialog for viewing and setting the noise in the network, (6) 3D network view.

6.2.3 SpikeStream Simulation

The SpikeStream simulator is based on the SpikeNET architecture (Delorme and Thorpe 2003) and it consists of a number of processes that are launched and coordinated using PVM, with each process modelling a group of neurons using a combination of event-based and synchronous simulation.[2] In common with synchronous simulations the simulation period is divided into steps with an arbitrarily small time resolution and each neuron group receives lists of spikes from

[2] One difference between SpikeStream and SpikeNET is that messages are sent rather than requested at each time step.

other connected layers at each time step. However, only the neuron and synapse classes that receive a spike are updated, which substantially cuts down on the amount of processing required. Since the main overhead is calculating the neurons' state and sending the spikes, the simulator's update speed depends heavily on the level of network activity, and at high levels the performance becomes the same as a synchronous simulator. In theory, SpikeStream's run speed should be relatively independent of the time step resolution, since the calculation of each time step is efficient and the network should emit the same number of spikes per second independently of the time step resolution. In practice, the setting of this value can affect the number of spikes emitted by the network because higher values reduce the number of spikes that arrive during a neuron's refractory period and alter the network dynamics (see Table 6.2).

The spikes exchanged between neurons are a compressed version of the presynaptic and postsynaptic neuron IDs, which enables each spike to be uniquely routed to a class simulating an individual synapse. Variable delays are created by copying emitted spikes into one of 250 buffers, which enables them to be delayed for up to 250 time steps. This number of buffers was chosen to minimize the space required to store the delays in the database and it was found to offer enough resolution and length of delay for the time step values that were used in the experiments.[3]

Unlike the majority of neural simulation tools, SpikeStream can operate in a live mode in which the neuron models are calculated using real time instead of simulation time. This live mode is designed to enable SpikeStream to control robots that are interacting with the real world and to process input from live data sources, such as cameras and microphones. Although SpikeStream is primarily an event-driven simulator, it can also be run synchronously to accommodate neuron models that generate spontaneous activity.

[3] This value could be changed by editing the SpikeStream code if longer delays or higher delay resolution was required.

6.2.4 SpikeStream Archiver

During a simulation run, the firing patterns of the network can be recorded by SpikeStream Archiver, which stores lists of spikes or firing neurons in XML format along with a simple version of the network model.

6.2.5 Neuron and Synapse Classes

Neuron and synapse classes are implemented as dynamically loaded libraries, which makes it easy to experiment with different neuron and synapse models without recompiling the whole application. Each dynamically loadable class is associated with a parameter table in the database, which makes it easy to change parameters during a simulation run. The current distribution of SpikeStream includes neuron and synapse classes implementing the Spike Response Model and Brader et al.'s (2006) STDP learning rule (see sections 5.3.2 and 5.3.3), which were developed for the work in this thesis.

6.3 Performance

6.3.1 Tests

The performance of SpikeStream was measured using three test networks put forward by Brette et. al. (2006). The main network specified by this paper has 3,200 excitatory neurons and 800 inhibitory neurons that are randomly interconnected with a 2% probability. Larger networks of 10,000 and 20,000 neurons with a similar excitatory/ inhibitory ratio were also put forward by Brette et al., and for the performance tests of SpikeStream the networks were divided into four layers to enable them to be distributed across multiple machines. The neuron and synapse models for these networks could be implemented in four different ways:

- *Benchmark 1*. A network of conductance-based integrate and fire neurons, equivalent to the COBA model described in Vogels and Abbott (2005).
- *Benchmark 2*. A network of integrate and fire neurons connected with current-based synapses, which is equivalent to the CUBA model described in Vogels and Abbott (2005).
- *Benchmark 3*. Conductance-based Hodgkin-Huxley network.
- *Benchmark 4*. Integrate and fire network with voltage-jump synapses.

For these performance tests, Benchmark 4 was chosen because it was the easiest to implement using event-driven simulation strategies.

At the beginning of each simulation run the networks were driven by a random external current until their activity became self sustaining and then their performance was measured over repeated runs of 300 seconds. A certain amount of fine tuning was required to make each network enter a self-sustaining state that was not highly synchronized and the final parameters for each size of test network are given in Table 6.1.[4] The neuron and synapse models that were used for these tests were the same as those described in Section 5.3.2.

The first two networks were tested on one and two Pentium IV 3.2 GHz machines connected using a megabit switch with time step values of 0.1 and 1.0 ms. The third network could only be tested on two machines because its memory requirements exceeded that available on a single machine. All of the tests were run without any learning, monitoring or archiving.

[4] Most of the initial values of these parameters were taken from Brette et. al. (2006).

Parameter	Small network	Medium network	Large network
Neurons	4000	10,000	19,880
Connections	321985	1,999,360	19,760,878
ω_{ij} (excitatory)	0.11	0.11	0.11
ω_{ij} (inhibitory)	-1.0	-0.6	-0.6
Threshold	0.1	0.15	0.25
τ_m	3	3	3
m	0.8	0.8	0.8
n	3	3	3
Connection delay	1	1	1
ρ	3	3	3

Table 6.1. Parameters of test networks

6.3.2 Results

The amount of time taken to simulate one second of biological time for each of the test networks is plotted in Figure 6.2. In this graph the performance difference between 0.1 and 1.0 ms time step resolution is partly due to the fact that ten times more time steps were processed at 0.1 ms resolution, but since SpikeStream is an event-based simulator, the processing of a time step is not a particularly expensive operation. The performance difference between 0.1 and 1.0 ms time step resolution was mainly caused by changes in the networks' dynamics that were brought about by the lower time step resolution, which reduced the average firing frequency of the networks by the amounts given in Table 6.2.

Time step resolution	Small network	Medium network	Large network
0.1 ms	109 Hz	72 Hz	40 Hz
1.0 ms	79 Hz	58 Hz	30 Hz

Table 6.2. Average firing frequencies in simulation time at different time step resolutions

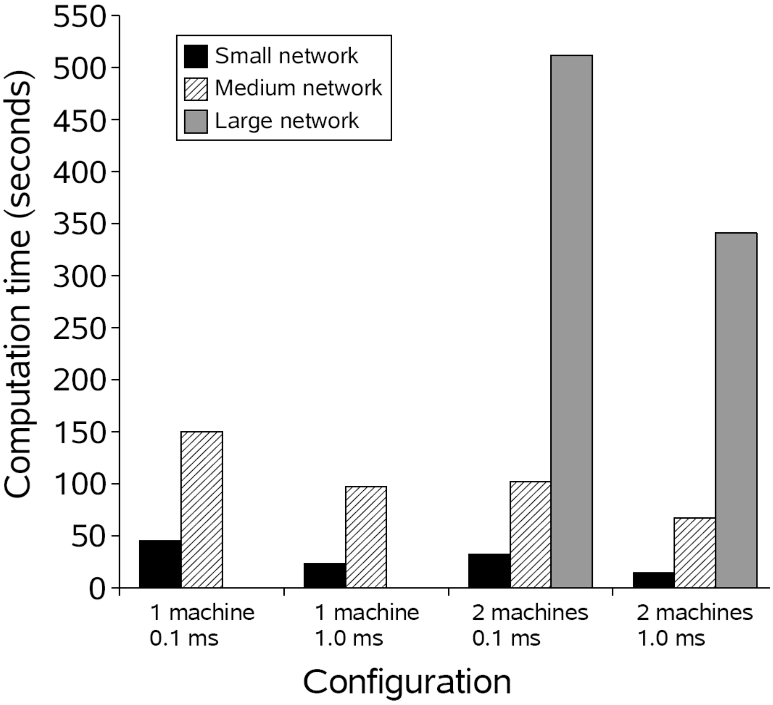


Figure 6.2.. Time taken to compute one second of biological time for one and two machines using time step resolutions of 0.1 and 1 ms

The differences in average firing frequency shown in Table 6.2 suggest that the relationship between real and biological time needs to be combined with other performance measurements for event-based simulators. To address this issue, the number of spikes processed in each second of real time was also measured and plotted in Figure 6.3. This graph shows that SpikeStream can handle between 800,000 and 1.2 million spike events per second on a single machine and between 1.2 million and 1.8 million spike events per second on two machines for the networks that were tested. Figure 6.2 and Figure 6.3 both show that the performance increased when the processing load was distributed over multiple machines, but with network speed as a key limiting factor, multiple cores are likely to work better than multiple networked machines.

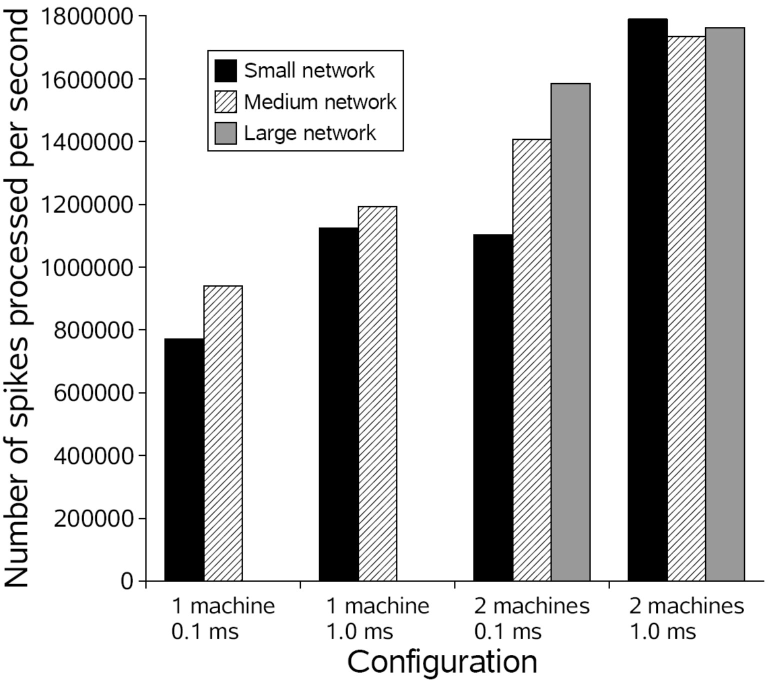


Figure 6.3. Number of spikes processed per second of real time for one and two machines using time step resolutions of 0.1 and 1 ms

Most of the performance measurements in Brette et. al. (2006) are for the neuron and synapse models specified by benchmarks 1-3, which cannot be meaningfully compared with the SpikeStream results for Benchmark 4. The only results that are directly comparable are those for NEST, which are given by Brett et al. (2006, Figure 10B) for two machines. On the 4,000 neuron network NEST takes 1 second to compute 1 second of biological time when the synapse delay is 1 ms and 7.5 seconds to compute 1 second of biological time when the synapse delay is 0.125 ms. Compared with this, SpikeStream takes either 14 or 30 seconds to simulate 1 second of biological time, depending on whether the time step resolution is 1.0 or 0.1 ms, and these SpikeStream results are independent of the amount of delay that is used.

The other point of comparison for the performance of SpikeStream is SpikeNET. The lack of a common benchmark makes comparison difficult, but Delorme and Thorpe (2003) claim that SpikeNET can simulate approximately 400,000 neurons firing at 1Hz real time with 49 connections per neuron and 1 ms time step. This works out as 19.6 million spike events per second, whereas SpikeStream can only handle a maximum of 1.2 million spike events per second on a single PC for the networks tested (see Figure 6.3). This measurement for SpikeNET was obtained using a substantially slower machine, and so its performance would probably be at least 800,000 neurons firing at 1 Hz in real time today.

6.4 External Devices

SpikeStream can pass spikes over a network to and from external devices, such as cameras and real and virtual robots, in a number of different ways:

- *Synchronized TCP.* Spikes are exchanged with the device at each time step; SpikeStream and the external device only move forward when they have both completed their processing for the time step.
- *Loosely synchronized UDP.* Spikes are sent and received continuously to and from the external device with the rate of the simulation determined by the rate of arrival of the spike messages.
- *Unsynchronized UDP.* Spikes are sent and received continuously from the external device. This option is designed for live work with robots.

The main external device that has been used and tested with SpikeStream is the SIMNOS virtual robot created by Newcombe (see Section 1.2.3 and Figure 6.4). Visual data (available with different types of pre-processing), muscle lengths and joint angles are encoded by SIMNOS into spikes using a variety of methods and passed across the network to SpikeStream using the synchronized TCP method. When the spikes are unpacked by SpikeStream they are used to directly fire neurons or to change their voltage. SIMNOS also receives muscle length data from SpikeStream in the form of spiking neural events, which are used to control the virtual robot. Together SIMNOS and SpikeStream provide an extremely powerful way of exploring sensory and motor processing and integration.

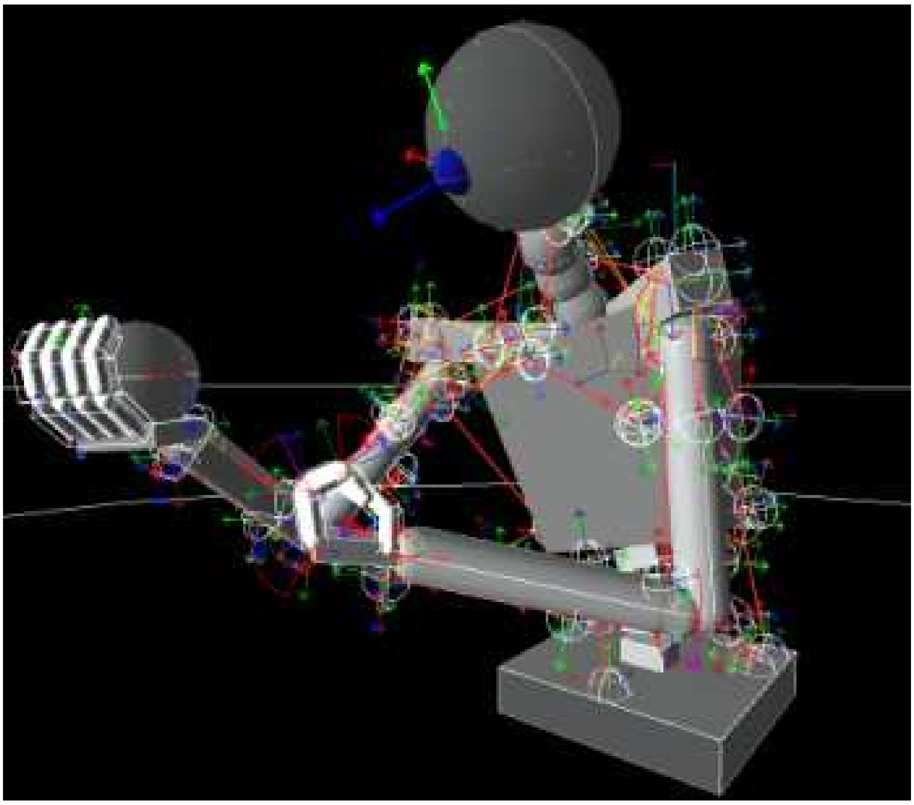


Figure 6.4. SIMNOS virtual robot. The red lines are the virtual muscles consisting of damped springs whose lengths are sent as spikes to SpikeStream. The outlines of spheres with arrows are the joint angles, which are also sent as spikes to SpikeStream.

6.5 Applications

Some potential applications of SpikeStream are as follows:

Biologically inspired robotics

Spiking neural networks developed in SpikeStream can be used to process sensory data from real or virtual robots and generate motor patterns. A good example of this type of work is that carried out by Krichmar et. al. (2005) on the Darwin series of robots (see Section 5.6).

Genetic algorithms

The openness of SpikeStream's architecture makes it easy to write genetic algorithms that edit the database and run simulations using PVM.

Models of consciousness and cognition.

Dehaene et al. (1998, 2003, 2005) and Shanahan (2008) have built models of consciousness and cognition based on the brain that could be implemented in SpikeStream (see Section 3.5.6). The neural network in Chapter 5 is also an example of this type of work.

Neuromorphic engineering

SpikeStream's dynamic class loading architecture makes it easy to test neuron and synapse models prior to their implementation in silicon. Initial work has already been done on enabling SpikeStream to read and write AER events, which would enable it to be integrated into AER chains such as those developed by the CAVIAR project.[5]

Teaching

Once installed SpikeStream is well documented and easy to use, which makes it a good tool for teaching students about biologically structured neural networks and robotics.

6.6 Limitations and Future Work

The flexibility and speed of SpikeStream come at the price of a number of limitations:

- Neurons are treated as points. Each connection can have a unique delay, but there is none of the complexity of a full dendritic tree.
- The connection delay is a function of the time step, not an absolute value, and there is a maximum of 250 delayed time steps. This limitation makes it more complicated to change the time step resolution, but it does not affect the accuracy or the performance of the simulator. The number of buffers could be changed in a future SpikeStream release if higher resolution of the delay or a longer delay was required.

[5] CAVIAR project: http://www.imse.cnm.es/caviar/.

- The full functionality of SpikeStream is only available for layers of rectangular neurons. The main work that would be needed for three-dimensional neuron groups is the extension of SpikeStream Application to enable it to monitor three-dimensional firing patterns. The editing and visualisation have already been partly extended to deal with three-dimensional neuron groups, the simulation and archiving code will work with any shape of neuron group, and the databases will also support any shape of neuron group.
- Any two neurons can only have a single connection between them. This restriction exists because the ID of each connection in the database is formed from a combination of the presynaptic and postsynaptic neuron IDs. This limitation has little impact on the ability of SpikeStream to model point neurons. Multiple connections between two neurons would only make sense if the full dendritic tree was being modelled - when a simulator, such as NEURON or GENESIS, would be more appropriate.
- Although SpikeStream's performance was adequate for the network developed by this thesis, it is likely that it could be substantially improved. Whilst the performance advantage of SpikeNET was achieved at the cost of many important features, it would be worth looking more closely at NEST to see if some its optimization strategies could be incorporated into the SpikeStream simulator. It might also be possible to use the SpikeStream databases to set up simulation runs in NEST, which lacks a graphical user interface.
- SpikeStream currently uses mysqldump to save and load its databases. In the future it would be worth extending the saving and loading functions of SpikeStream to

support the standard XML formats that have been developed for neural networks, such as NeuroML[6] and BrainML.[7]

6.7 Release

SpikeStream is available for free download under the terms of the GPL license. The current (0.1) release has 25,000 source lines of code,[8] full source code documentation, a mailing list for SpikeStream users, and a comprehensive 80 page manual, which has been included in this thesis as Appendix 1. SpikeStream is also available pre-installed on a virtual machine, which works on all operating systems supported by VMware and can be run using the free VMware Player.[9] More information about this release is available at the SpikeStream website: http://spikestream.sf.net. At the time of writing SpikeStream 0.1 has had 140 downloads from the Sourceforge website.

6.8 Conclusions

This chapter has outlined the architecture and performance of SpikeStream, which can simulate medium sized networks of up to 100,000 neurons and is available for free download under the terms of the GPL licence. This simulator is modular, flexible and easy to use and can interface with real and virtual robots over a network. SpikeStream was used to model the neural network described in Chapter 5, and the next chapter analyzes this network for representational mental states and information integration, and makes predictions about its phenomenal states.

[6] NeuroML website: http://www.neuronml.org.

[7] BrainML website: http://www.brainml.org.

[8] This was calculated using Wheeler's SLOCCount software. More information about Wheeler's measure can be found here: http://www.dwheeler.com/sloc/.

[9] VMware Player: http://www.vmware.com/products/player/.

7. ANALYSIS

7.1 Introduction

This chapter describes how the neural network in Chapter 5 was analyzed for consciousness using the approach to synthetic phenomenology set out in Chapter 4. The first section in this chapter covers the calculation of the OMC rating of the network, Section 7.3 explains the method that was used to identify the representational mental states, and then Section 7.4 describes the analysis of the system for information integration using Tononi and Sporns' (2003) approach. Sections 7.5 - 7.7 look at whether the network is capable of consciousness according to Tononi's, Aleksander's and Metzinger's theories and definitions are formulated that enable the network to be automatically analyzed for phenomenal states. The final part of this chapter describes how the network's activity was recorded and combined with the analysis data to produce a sequence of XML files that predict the phenomenology of the system according to the three theories of consciousness.

All of this analysis was carried out on two 3.2 GHz Pentium IV computers with 2 GB RAM. The code for this analysis is all part of the Network Analyzer software, which was written as part of this PhD and is briefly covered in Appendix 2. No official release of Network Analyzer is planned, but the source code for the current version is included in the supporting materials.

7.2 OMC Rating

In this network all of the mental states are implemented in the same way, and so they all have the same rating on version 0.6 of the OMC scale described in Section 4.2. The system is a biologically inspired simulated neural network running on two single processor computers at a speed that is significantly slower than the human brain, and so its factors are S1, R2, F1, FN4, TS2, AD3, giving a total weighting of 3.025 x 10^{-3}. This needs to be multiplied by 0.1 to compensate for the missing level of molecules, atoms and ions, leading to a final weighting of 3.0 x 10^{-4}, which is an OMC position of 111 out of 192 on the scale, and an OMC rating of 0.43. This OMC rating makes intuitive sense because the final arrangement of atoms and electrons in the system is substantially different from that in a human brain, but not to the extent that it is impossible to conceive that it has conscious states. This OMC rating is incorporated into the XML description of the phenomenology in Section 7.9.

7.3 Identification of Representational Mental States

7.3.1 Definition of a Mental State for this System

In this analysis a simulated neural network is being analysed for consciousness, and so the mental states are states of the simulated network.[1] Depending on how a neural network is modelled, there are many different ways of defining its states – for example, the spiking activity of a population of neurons, the voltages in the neurons, the average neuron firing rates, changes in memory addresses or activity in the processor and RAM – and in this analysis, it was decided to treat the firing of a neuron as a mental state. Although this is fairly basic, the main purpose of this analysis is to illustrate how synthetic phenomenology can be carried out, and it would have been unnecessarily complicated to use population codes or memory addresses to make predictions about the network's phenomenal states.

[1] See Section 4.3.2 for the definition of a mental state that is being used in this thesis.

7.3.2 Selection of Method

To identify the representational mental states of the network a method was needed that could identify the functional or effective connections between the input and output data and the internal states (see Definition 4.2 in Chapter 4). In this network the input and output pass through Vision Input and Motor Output, and so I decided to look at the functional and effective connections between these layers, which had known response characteristics, and the internal layers whose responses were not known. The first problem that had to be addressed was that a complete map of the representational states of the network was required for the XML description, and yet the network only activated a small selection of its possible states during normal activity. To get around this problem it was decided to inject noise into the layers that had known response characteristics, and use an algorithm or mathematical method to identify the functional or effective relationships between activity in the neurons with known response characteristics and activity in the internal neurons whose representational characteristics were being measured.

One of the first algorithms that I considered was the backtracing method developed by (Krichmar et. al. 2005), which examines the firing rate of a reference neuron at a specific time step and identifies the neurons connected to the reference neuron that were active during the previous time step. Whilst it might have been possible to trace the spikes back through the network in this way, the recurrent loops and delays in the network would have made this process extremely complicated. Another method that was considered was Granger causality (Seth and Edelman 2007), but this would have required conversion of the spiking activity into average firing rates, which I wanted to avoid if possible. Instead, it was decided to use mutual information to measure the relationships between the input/ output and internal neurons, and the next section describes how this can be calculated from the spiking activity. Although mutual information does not directly measure causal relationships, under these experimental conditions a strong case can be made that it is a measure of effective connectivity (see Section 7.3.6).

7.3.3 Identification of Representational Mental States Using Mutual Information

The first step in the analysis for representational mental states was the selection of an input or output layer, which was given one description in natural human language and another in terms applicable to the physical world (when this could be done reasonably easily). Next, noise was injected into the input or output layer and the network activity was recorded. This data was then used to calculate how much mutual information each internal neuron shared with the input or output neurons that had been given the physical and human descriptions. This procedure was repeated separately for each input and output layer that had response characteristics that could be easily described. In theory this noise injection technique could be also used to identify mental states that represent other mental states, but the difficulty of describing internal neuron groups led me to exclude meta representational mental states from this analysis.[2]

The mutual information between each input/output neuron, *X*, and each internal neuron, *Y*, was calculated by recording the number of times that the following combinations occurred for different steps back in time ("1" indicates that the neuron was firing at that time step and "0" indicates that the neuron was quiescent):

$x = 0 \;\&\; y = 0$

$x = 1 \;\&\; y = 0$

$x = 0 \;\&\; y = 1$

$x = 1 \;\&\; y = 1$

These statistics enabled the joint probabilities to be calculated:

[2] Meta representational mental states would have been needed to analyze the network using Rosenthal's (1986) higher order thought theory. However for the reasons discussed in Section 2.3.2 this theory was not used in this analysis.

$p(x = 0,\ y = 0)$

$p(x = 1,\ y = 0)$

$p(x = 0,\ y = 1)$

$p(x = 1,\ y = 1)$

for different steps back in time as well as the marginal probabilities:

$p(x = 0)$

$p(x = 1)$

$p(y = 0)$

$p(y = 1)$.

Using these values, the mutual information between each input/output neuron *X* and each internal neuron *Y* was calculated using the standard formula for mutual information:

$$I(X;Y) = \sum_{x=0}^{1}\sum_{y=0}^{1} p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right). \tag{7.1}$$

Equation 7.1 was also used to work out the maximum possible mutual information under the experimental conditions. With 20% of the neurons being fired randomly at each time step:

$p(x = 0) = 0.8$

$p(x = 1) = 0.2$

$p(y = 0) = 0.8$

$p(y = 1) = 0.2$.

When the mutual information between *X* and *Y* is at a maximum, their state will always be the same, and so:

$p(x = 0,\ y = 1) = 0$

$p(x = 1,\ y = 0) = 0$,

and the remaining joint probabilities can be derived from the noise:

$p(x = 0,\ y = 0) = 0.8$

$p(x = 1,\ y = 1) = 0.2.$

Putting these figures into Equation 7.1 yields a maximum possible mutual information of 0.72.

During the recording of the data a time step value of 10 ms was used to avoid complications caused by the refractory period[3] and all other sources of network noise were switched off. The injection of 20% noise into each input/ output layer[4] and approximately 10,000 time steps of recorded activity were found to give mutual information values that matched expectations based on the known connectivity of the network. Since the mutual information between two neurons is rarely zero, a threshold was used to eliminate low mutual information values that would have been superfluous in the final XML description. The results for Vision Input and Motor Integration are covered in the next two sections.

7.3.4 Visual Representational Mental States

Vision Input was an obvious choice of input layer for the visual analysis because it could be easily labelled and had strong forward connections to the rest of the network. With layers of several thousand neurons the analysis for representational mental states consumes a lot of time and memory because the mutual information has to be calculated for each combination of input/ output and internal neurons. This problem can be reduced by excluding layers from the analysis that are unlikely to have any systematic link with the layer that is being used as input or output For the visual analysis Motor Cortex was excluded because it did not have any input connections from other layers, and Motor Integration, Eye Pan and Eye Tilt were also left out because they did not have any direct or indirect connections from Vision Input. The mutual information between the input and internal neurons was calculated for between zero and five

[3] The total refractory period of the neurons is approximately 10 ms.

[4] Noise injection in this part of the analysis was done by firing a random selection of 20% of the neurons at each time step.

steps back in time because activity in Vision Input took four time steps to propagate to Motor Output.

The next stage in the visual analysis was to decide on appropriate labels for the neurons in Vision Input. Since half of Vision Input carries red visual information and half carries blue, it was decided to include both red and blue in the human description and to use the corresponding wavelengths of light in the physical description.[5] The parameters for the identification of visual representational mental states are summarised in Table 7.1.

Parameter	**Value**
Input Neuron Group	Vision Input
Internal Neuron Groups	Emotion, Red Sensorimotor, Blue Sensorimotor, Inhibition, Motor Output
Human Description	“Red / blue visual input”
Physical Description	“700 nm / 450 nm electromagnetic waves”
Steps back in time	0 - 5
Mutual Information Threshold	0.1
Input Neuron Group Noise	20%

Table 7.1. Parameters for the analysis of visual representational mental states

The data structures were too large to fit in memory, and so the input/output and internal layers were split into five groups and the mutual information calculations were run on the 25 possible combinations between them, which took several days to complete.[6] A high level summary of the average mutual information shared between Vision Input and the internal layers is plotted in Figure 7.1.

[5] A more sophisticated analysis could have distinguished between light wavelengths and perceived colours when assigning the human and physical labels.

[6] This separation into separate groups did not have any effect on the final result.

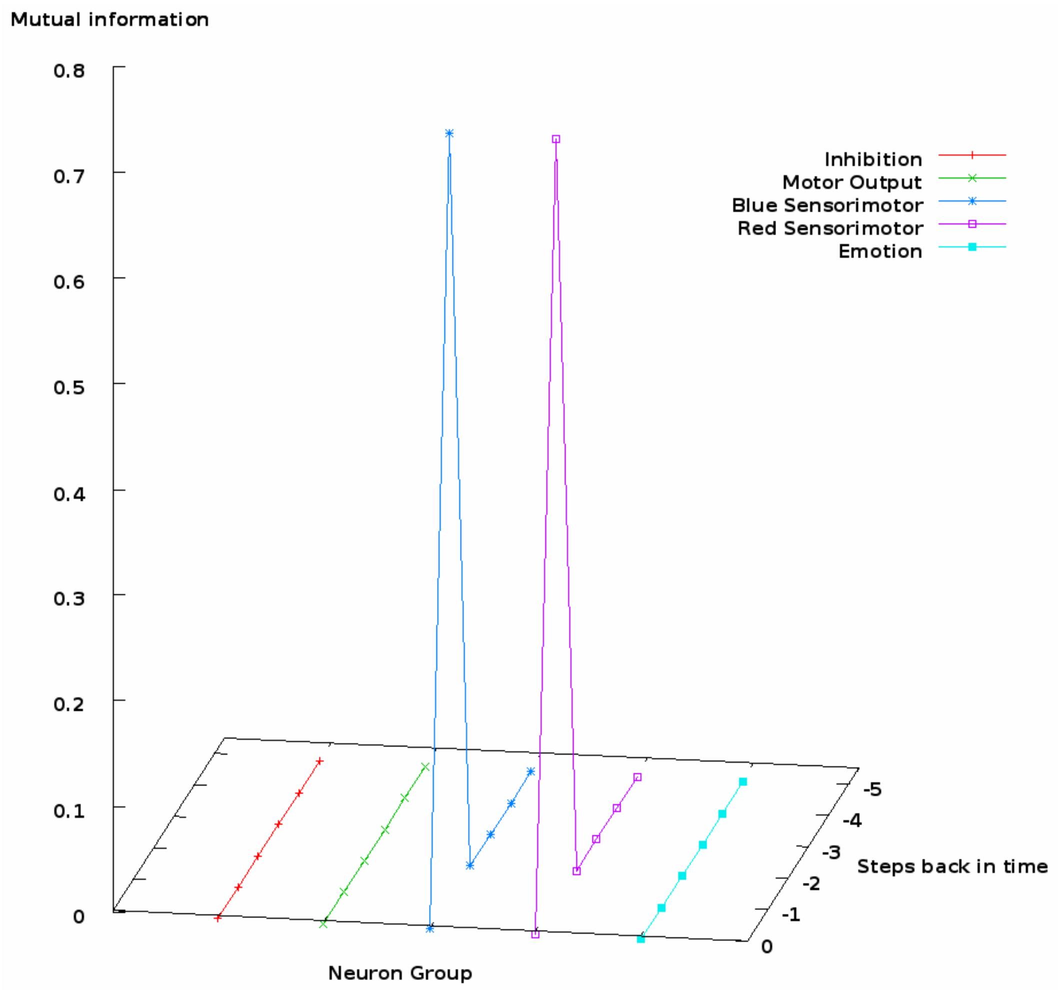


Figure 7.1. Average mutual information shared between Vision Input and the internal layers during the analysis for representational mental states

The results in Figure 7.1 show that the mutual information shared between Red Sensorimotor and Blue Sensorimotor and Vision Input was close to the theoretical maximum of 0.72, which matched expectations because of the strong topological connections between Vision Input and Red/ Blue Sensorimotor. Although Emotion is indirectly connected to Vision Input, it shared no mutual information above the threshold, which was probably due to the large number of internal connections within Emotion that made its self-sustaining activity largely independent of Red Sensorimotor. The other neuron groups downstream of Emotion, such as Inhibition and Motor Output, also shared no mutual information with Vision Input.

7.3.5 Proprioception/ Motor Output Representational Mental States

Motor Output would not have been a good choice of output layer for the motor analysis because it does not have any forward connections to the other layers, and no representational mental states would have been found by injecting noise into it. A better choice was Motor Integration, which has forward connections to other layers, contains a complete map of all possible motor combinations and plays a key role in action selection through its connections to Red Sensorimotor and Blue Sensorimotor. Motor Integration can also be given a clear human description because it maps directly down to Motor Output through Eye Pan and Eye Tilt. Motor Cortex and Vision Input were excluded from this part of the analysis because they did not have any incoming connections from other layers.

Although the neural network does not receive sensory data from SIMNOS's joints or muscles, the motor control signals sent from Motor Integration accurately predict the position of the eye after a delay of a few time steps, and so activity in this layer encodes both proprioceptive and motor information. To reflect this dual role, "Proprioception / motor output" was chosen as the human description of the neurons in Motor Integration and the physical description was set to "N/A" because it would have been too complicated to describe the physical movements of the eye in response to activity in this layer. There is an 11 time step delay from Motor Integration to Red Sensorimotor and Blue Sensorimotor, and so the mutual information between the input and internal neurons was calculated for between zero and twelve steps back in time. Although this had the effect of excluding potential representational links between Motor Integration and Emotion via Red/ Blue Sensorimotor, the visual analysis strongly suggested that there was no representational relationship between Emotion and Red/ Blue Sensorimotor. The parameters for the identification of proprioception/ motor output representational mental states are summarized in Table 7.2.

Parameter	**Value**
Input Neuron Group	Motor Integration
Internal Neuron Groups	Emotion, Red Sensorimotor, Blue Sensorimotor, Inhibition, Eye Tilt, Eye Pan, Motor Output.
Human Description	"Proprioception / motor output"
Physical Description	"N/A"
Steps back in time	0 - 12
Mutual Information Threshold	0.1
Input Neuron Group Noise	20%

Table 7.2. Parameters for the analysis of proprioception/ motor output representational mental states

The data structures for the proprioception/ motor output analysis fitted comfortably in memory and the calculations took less than an hour to complete. A high level summary of the average mutual information shared between Motor Integration and the internal layers is plotted in Figure 7.2.

In the results shown in Figure 7.2 Motor Output shows a small response with a peak of 0.01 at minus two time steps, which might have been expected to be higher since there is an indirect link between Motor Integration and Motor Output. However, the value of 0.01 represents the *average* mutual information between Motor Integration and Motor Output, and only 10 out of 675 neurons in Motor Output are indirectly connected to Motor Integration. The highest average mutual information is shared between Motor Integration and Eye Pan and Eye Tilt at -1 time steps. This is close to the theoretical maximum and it is due to the topographic connections between Motor Integration and Eye Pan and Eye Tilt. There are also average mutual information peaks for Red Sensorimotor and Blue Sensorimotor at -12 time steps, which matched expectations since there is a connection with a delay of 11 time steps from Motor Integration to Red/ Blue Sensorimotor and it takes one time step for a spike to be emitted from one group and processed in another.

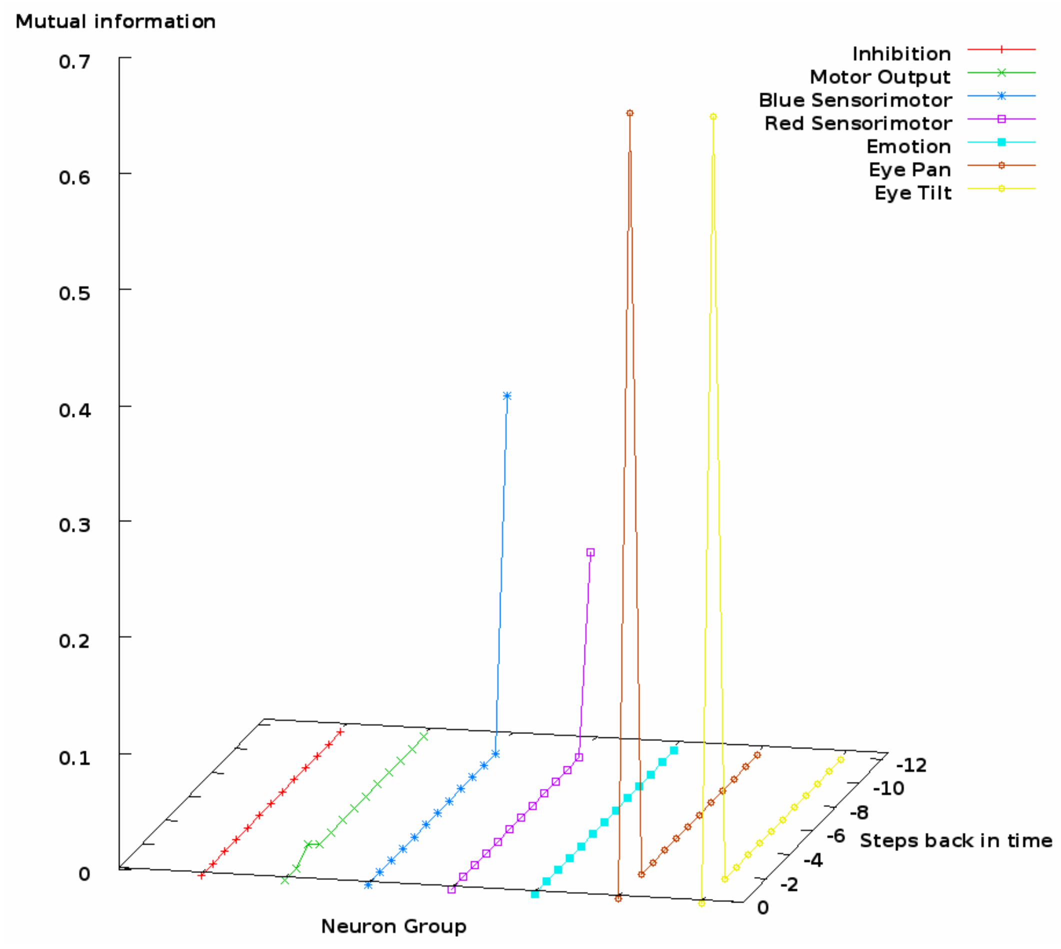


Figure 7.2. Average mutual information shared between Motor Integration and internal layers during the analysis for representational mental states

The graphs in Figure 7.1 and Figure 7.2 only give the average mutual information that is shared between the input/output and internal layers. The detailed results can be found in the VisualRepresentationalMentalStates.xml and MotorRepresentationalMentalStates.xml files, which are included in the supporting materials.

7.3.6 Representational Mental States: Discussion and Future Work

One limitation of mutual information is that by itself it is not a measure of the causal relationships between neurons. If two neurons, *A* and *B*, share mutual information, then it could be because *A* is causally influencing *B*, *B* is causally influencing *A* or because *A* and *B* are under the causal influence of a third neuron, *C*. However, in the method described Section 7.3.3 there is good reason to believe that the activity of the internal neurons is due to their causal dependencies on the input/ output layers because the input/ output layers are individually put into a high (but not maximum) state of entropy and there is no other source of spontaneous activity within the network. A second reason why this method is likely to measure causal dependencies is because the mutual information is calculated for different numbers of steps back in time. If the mutual information between *A* and *B* peaked at time step zero, for example, then this would suggest that *A* and *B* were under the causal influence of a third neuron, *C*, but if *B* shares maximum mutual information with *A* at -2 time steps, then it is more likely that there is a causal relationship between *A* and *B* - although *A* and *B* could still be subject to a common cause *C* that is connected to A and B with different delays. Finally, the close match between the mutual information relationships and the structure and delays of the network makes it reasonable to assume that the internal neurons sharing high mutual information with input/output neurons are causally dependent on these input/output neurons.

This analysis did not attempt to identify mental states that represent other mental states because the descriptions would have been too complicated to define at both the human and physical levels. Future work in this area might be able to track the processing of data through the network by repeating the analysis a number of times at different levels. For example, mental states that responded to a combination of motor output and blue visual information could be injected with noise to discover representational mental states that respond to more abstract higher level information. In this way meta representational mental states could be described as combinations of more basic mental states that are linked to states of the world. Mental states representing more complex features of the world could also be identified using more specific test data.

The visual and motor systems of this network were extremely basic and on such a simple system the injection of noise into Vision Input and Motor Integration was a good way of identifying representational mental states. However, a more subtle approach would be needed to identify representational mental states in more complex systems. If the system is *designed* with layers that respond to different aspects of the input signal - for example the visual input layers in Krichmar et. al. (2005) – then the layers could be individually labelled and injected with noise to identify the representational mental states. However, when the system's responses to complex aspects of the world are not known – for example, in self-organizing networks, such as the hippocampus in Krichmar et al. (2005) – then it might be possible to use the statistical methods developed by Lungarella et. al. (2005) and Lungarella and Sporns (2006) to identify regularities in the input and output signals, which could be used to label the representational mental states.

In the future it would be worth exploring whether other techniques, such as transfer entropy (Schreiber 2000, Sporns et al. 2006), backtracing (Krichmar et. al. 2005) and Granger causality (Seth and Edelman 2007), make different predictions about the representational mental states of the network. It would also be worth investigating how the definition of a system's mental states affects its representations. For example, if mental states were defined in terms of populations of neurons, then Kohonen (2001) or one of Grossberg's (1976) neural networks could be used to identify patterns in the neuron populations, and the mutual information shared between these patterns and the input/ output data could then be measured using the noise injection method.

7.4 Information Integration Analysis

7.4.1 Introduction

This section describes how the neural network was analyzed for information integration using Tononi and Sporns' (2003) method. The main motivation for this analysis was to make predictions about the consciousness of the network using Tononi's (2004) information integration theory of consciousness. The phenomenology of a system also depends on the integration between the different pieces of conscious information (see Section 4.3.6), and since information integration is a measure of effective connectivity (Sporns 2007), it made sense to use Tononi and Sporns' method to identify the integration between the mental states in the network. Information integration is also used in the predictions about the consciousness of the network based on Metzinger's (2003) theory (see Section 7.7.3).

The central difficulty with Tononi and Sporns' (2003) method is that the analysis time increases factorially with the number of subsets and bipartitions, which makes it impossible to exhaustively analyse systems with more than fifty elements. To find out the scale of this problem, Section 7.4.3 gives an estimate of how long the full analysis would take on a network with 17,544 neurons. Since this is of the order of 10^{9000} years, optimisation strategies had to be developed for large networks, which are documented in Section 7.4.4, and Section 7.4.5 gives the result of testing these optimizations on Tononi and Sporns' (2003) sample networks. The remaining information integration sections present the results and some background and future work. Further details about the information integration results are included in Appendix 3.

7.4.2 Tononi and Sporns' Information Integration Calculation

As explained in Section 2.6.2, the complexes of a system are identified by considering every possible subset S of m elements out of the n elements of the system, starting with $m = 2$ and finishing with $m = n$. For each possible bipartition of the subset, the effective information integrated across the bipartition, EI($A \leftrightarrows B$), is calculated and the minimum normalized effective information, min{ EI($A \leftrightarrows B$) / H^{MAX}($A \leftrightarrows B$)}, is identified. The non-normalized minimum effective information is the Φ value of the subset, and a complex is a subset with $\Phi > 0$ that is not included in a larger subset with greater Φ. At the centre of this method is the calculation of EI($A \leftrightarrows B$), which is repeated a large number of times during the analysis. The stages in the calculation of EI($A \leftrightarrows B$) are as follows.

Normalization

The starting point for the EI(A⇆B) calculation is the connection matrix, CON(X), which is an *m* x *m* matrix representing all of the connections between the *m* elements of the subset. In this analysis all of the weights were made positive by multiplying them by -1 on the grounds that a connection is passing information regardless of whether it is excitatory or inhibitory.[7] Without this normalization of negative weights it is conceivable that the positive and negative connections between the two bipartitions of a subset would have partially cancelled each other out, leading to a value of EI(A⇆B).that did not reflect the amount of information that was exchanged between the two bipartitions.[8]

Tononi and Sporns (2003) normalized the connection matrix by multiplying the weights so that the absolute value of the sum of the afferent synaptic weights per element was a constant value, *w*, which they set to 0.5 for their analysis Whilst this normalization method was appropriate for Tononi and Sporns' task of comparing different architectures that have been artificially evolved, it substantially distorts the relationships between the weights and does not correctly measure the information integrated by the system. For example, two neurons connected with a weight of 0.00001 have very little effective information between them, but the constant value weight normalization changes the connection weight to 0.5 and substantially alters the information exchanged between the two neurons. To avoid these problems, this analysis normalized the connection matrix by summing each neuron's afferent weights, finding the maximum value and calculating the factor that would reduce this maximum to less than 1.0. All of the weights in the network were then multiplied by this factor, which ensured that the sum of

[7] The alternative method of adding a constant to all of the weights was rejected because it would have made positive connections count for more, when in fact positive and negative connections with the same weight were transmitting the same amount of information

[8] I have not been able to find any discussion of negative weights in Tononi and Sporns (2003) or Tononi (2004) and their examples are all based on positive weights.

each neuron's afferent weights was always less than one without distorting the relationships between them.

Covariance Matrix

In each effective information calculation one part of the subset, A, is put into a state of maximum entropy and the entropy of the response of B is used to calculate EI(A→B). Since A is being substituted by independent noise sources, all of the self connections within A and the connections afferent to A are set to zero within CON(X). Under Gaussian assumptions, the elements in the system can be represented by a vector X of random variables that are subject to independent Gaussian noise R of magnitude c. When the elements settle under stationary conditions, the final state of the system is given by Equation 7.2:

$$X = X * \mathrm{CON}(X) + cR. \tag{7.2}$$

Using standard algebra and averaging over the states produced by successive values of R, this equation can be rearranged as:

$$X = R\ (1\text{-}\mathrm{CON}(X))^{-1}, \tag{7.3}$$

and a substitution of:

$$Q = (1\text{-}\mathrm{CON}(X))\text{-}1 \tag{7.4}$$

into Equation 7.3 gives the formula:

$$X = RQ. \tag{7.5}$$

In Equation 7.5, the elements of R that correspond to the A bipartition of the subset are set to 1.0 to put A into a state of maximum entropy, and the elements of R that correspond to the B bipartition are set to a value that corresponds to the background noise, which is typically 0.00001. Using the standard covariance formula:

$$\mathrm{COV(X) = X^{T}X}, \tag{7.6}$$

and substituting in Equation 7.5, we obtain:

$$\mathrm{COV(X) = (RQ)^{T}RQ}, \tag{7.7}$$

which is equivalent to Equation 7.8:

$$\mathrm{COV(X) = Q^{T}R^{T}RQ}, \tag{7.8}$$

and can be calculated from CON(X) using standard matrix operations.

Entropy

EI(A⇋B) depends on the entropies H(A), H(B) and H(AB), which can be calculated by extracting the sub matrices COV(A), COV(B) and COV(AB) from the covariance matrix and putting their determinants into Equation 7.9:

$$H(X) = \frac{\ln(\,(2\pi e)^{n}\,|\,COV(X)\,|\,)}{2}, \tag{7.9}$$

where | COV(X) | is the determinant of COV(X).[9]

EI(A⇋B)

The effective information from A to B, EI(A→B), is given by the mutual information between A and B when A is in a state of maximum entropy:

[9] This standard formula for calculating the entropy from the determinant of a covariance matrix can be found in Papoulis (1984, p. 541).

$$\mathrm{MI}(A^{HMAX}{:}B) = H(A^{HMAX}) + H(B) - H(A^{HMAX}B), \tag{7.10}$$

which can easily be calculated from the entropy values. The process is then repeated in the opposite direction by putting B into a state of maximum entropy to calculate EI(B→A), and the final value of effective information is given by:

$$\mathrm{EI}(A \leftrightarrows B) = \mathrm{EI}(A \rightarrow B) + \mathrm{EI}(B \rightarrow A). \tag{7.11}$$

This is normalized by $H^{MAX}(A \leftrightarrows B)$ to enable different bipartitions to be compared, and the information integration for subset S, or Φ(S), is the non-normalised value of $\mathrm{EI}(A \leftrightarrows B)$ for the minimum information bipartition.

The C++ code for these calculations was based on Tononi and Sporns' Matlab toolkit.[10] The most substantial change was that the Matlab code calculates $Q^T R^T RQ$ on the whole connection matrix and then extracts the A, B and AB sub matrices to work out the entropy. Since the complete connection matrix has 17,544 rows and columns, this approach would have been impossible with the computer resources available in this project. To get around this problem, the connection matrix was generated for each bipartition and then the determinants of A, B and AB were extracted. This yielded nearly identical results to the Matlab code on the validation tests (see Section 7.4.5) and can be justified by assuming that the effect of A on B when A is in a state of maximum entropy is much greater than the effect of the rest of the system on B. A brief overview of the Network Analyzer software is given in Appendix 2.

[10] The Matlab complexity toolbox is available at: http://tononi.psychiatry.wisc.edu/informationintegration/toolbox.html.

7.4.3 Time for the Full Information Integration Analysis

The full information integration analysis is computationally expensive because the EI(A⇋B) calculations are processor-heavy matrix operations that have to be run on every bipartition of every possible subset of the network. The first part of the full analysis is the extraction of all the possible subsets of the network, with the number of ways of selecting *m* elements out of the *n* elements of the system being given by:

$$\frac{n!}{m!(n-m)!}, \tag{7.12}$$

which has to be summed over all subset sizes from *m* = 2 to *m* = *n*.

The next part of the full analysis is the calculation of EI(A⇋B) on every possible bipartition of each subset in order to find the minimum information bipartition. A bipartition is created by selecting *k* elements out of the *m* elements in the subset, where k ranges from 1 to *m* / 2. Putting the subset selections together with the bipartition selections gives:

$$t_{analysis} = \sum_{m=2}^{\mathrm{n}} \sum_{k=1}^{m/2} \frac{n!}{m!(n-m)!} \frac{m!}{k!(m-k)!} f(m), \tag{7.13}$$

where $t_{analysis}$ is the full analysis time and *f*(*m*) is the time taken to calculate EI(A⇋B) on a single bipartition of a subset of size *m*. By cancelling out *m*! Equation 7.13 can be rearranged as follows:

$$t_{analysis} = n! \sum_{m=2}^{\mathrm{n}} \sum_{k=1}^{m/2} \frac{1}{(n-m)!} \frac{1}{k!(m-k)!} f(m). \tag{7.14}$$

Equation 7.14 omits the fact that when the number of neurons in each half of the bipartition is exactly the same, the number of possible bipartitions has to be divided by two, because the selection of all possible combinations in one half results in the selection of all possible combinations in the other half. This adjustment was included in the code that was used to estimate the full analysis times.

The time taken for each EI(A⇋B) bipartition calculation depends on a number of factors, including the efficiency of the code and the speed of the computer, and an estimate of this value was obtained by recording the average time that each EI(A⇋B) calculation took on subsets of different sizes (see Figure 7.3).

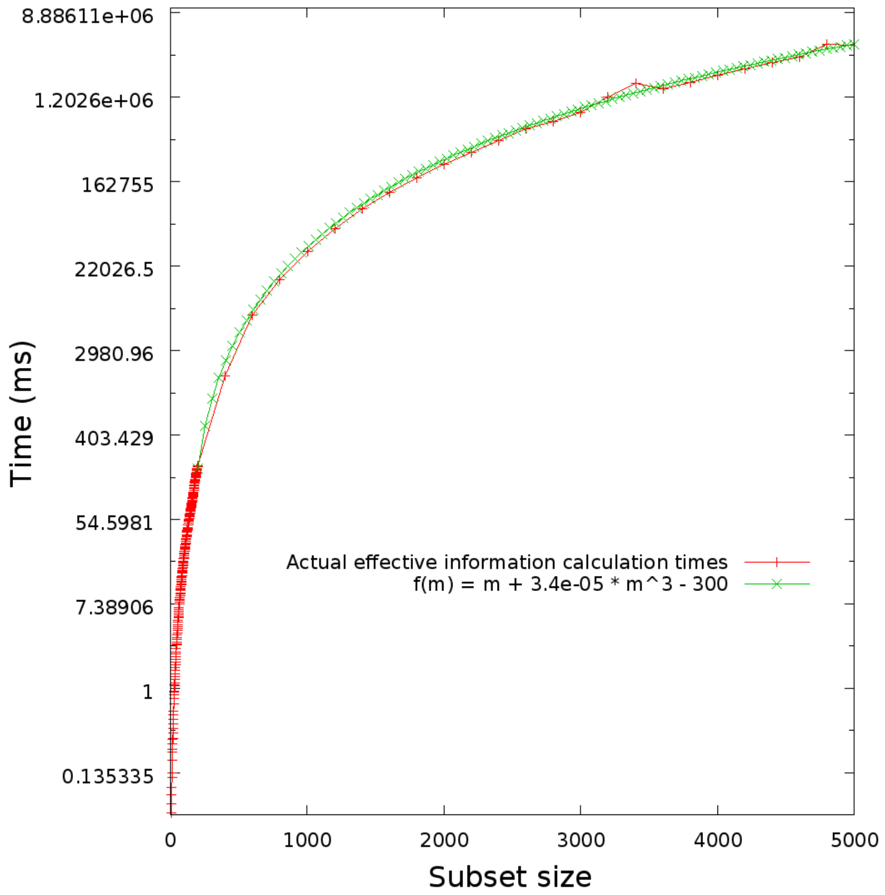


Figure 7.3. Actual and predicted times for each EI(A⇋B) bipartition calculation on subsets of different sizes

The curve fitting functions of gnuplot suggested that:

$$f(m) = m + 3.4e^{-5} \text{ x } m^{3} - 300 \tag{7.15}$$

was a good approximation to the actual values for $m > 200$ and Equation 7.15 was combined with the actual EI(A⇋B) calculation times to predict the bipartition calculation times for subsets with 2 - 5000 neurons. A short piece of code was then written that used the bipartition calculation times, Equation 7.14 and the adjustment for equal bipartitions to calculate $t_{analysis}$ for networks of different sizes, and the results from this calculation are plotted in Figure 7.4.[11]

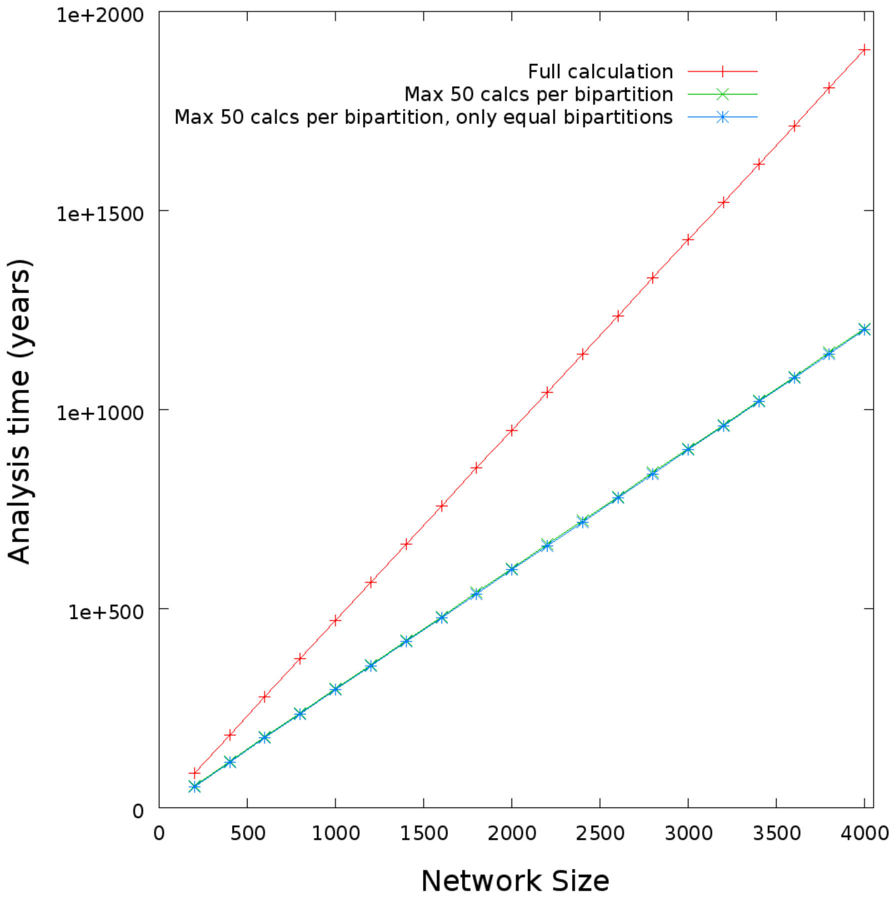


Figure 7.4. Predicted full information integration analysis times for networks of different sizes

Figure 7.4 only shows the predicted times for networks up to 4000 neurons because the factorial calculations took an increasingly long time to run as the network size increased and it was unclear whether it would reach 18,000 neurons within a reasonable time. However, the linear relationship between network size and the log of the calculation time can be extrapolated up to 18,000 neurons to give a predicted full analysis time of around 10^{9000} years. This shows that a full information integration analysis would have been completely impossible on a 17,544

[11] The data in Figure 7.4 was generated by the TimeCalculator class in Network Analyzer, which is included in the Supporting Materials.

neuron network with my current equipment. This difficulty is acknowledged by Tononi and Sporns (2003), who admit that “Practically, the procedure for finding complexes can only be applied exhaustively to systems composed of up to two dozen elements” (p. 18). The optimization and approximation strategies that were used to address this problem are discussed next.

7.4.4 Optimizations and Approximations

Given the extremely large amount of time that would have been required for the full analysis, it was necessary to develop optimizations and approximations that could identify some of the complexes in the network with the limited time and computer resources that were available to this project.

Sub-sampling

One approximation suggested by Tononi and Sporns (2003) is to evaluate EI(A⇆B) on a random selection of the possible bipartitions at each subset level. For example, to take 15 samples at each level for a 200 neuron subset, one would evaluate EI(A⇆B) for 15 samples of the 1:199 bipartition, 15 samples of the 2:198 bipartition, and so on up to 15 samples of the 100:100 bipartition. Although Tononi and Sporns suggested using 10,000 sub samples per level, the duration of each bipartition calculation suggested that orders of magnitude less sub-samples would have to be used if the calculation was going to complete in a reasonable time.

The impact of this approximation strategy is shown in Figure 7.4, where the blue line plots the predicted analysis times when the number of bipartition calculations per level is limited to 50, and the timings for the group analyses in Table A3.12 demonstrate that this approximation strategy is effective in practice. The disadvantage of this approximation is that it can dramatically reduce the proportion of bipartitions that are examined for the minimum information bipartition, which leads to a substantial loss of accuracy. In Network Analyzer, this approximation is implemented as the *Max number of bipartitions per level* parameter. The current version of the code examines the permutations of each bipartition in an ordered sequence up to the maximum limit; in the future it might be better to select the permutations at random.[12]

Another way of reducing the number of bipartition calculations is to sub-sample the levels. For example, in a subset of 200 neurons, this could involve sampling the 20:180, 40:160 … 100:100 bipartitions instead of every possible level. Although this option was included in the Network Analyzer code as the *Percentage of bipartition levels* parameter, it was rarely used in practice.

Seed expansion method

A second strategy, developed in collaboration with Richard Newcombe at Essex, is to grow each complex incrementally from a seed. To begin with a single neuron is selected at random, either from the entire network or from one of the neuron groups in the network. Next, one or a number of neurons connected to this seed are added to the subset and the Φ of the subset is calculated. If the new Φ is greater than the old one, the neurons are left in the subset and the process is repeated again. On the other hand, if the new Φ is less than the old one, then the new neurons are removed from the subset and the process is repeated with a different set of connected neurons. When all of the connections to and from the seed have been explored, the connections to and from other neurons in the subset are tried until the subset cannot be expanded any further. The remaining subset is likely to be a complex because any larger subset with greater Φ that includes the subset would have to be connected to it, and it has been shown that the addition of further connected neurons decreases the subset's Φ. The steps in the seed method are summarised in Figure 7.5:

[12] Random selection was not done in the current analysis because of the extra processing that would have been required to calculate the full range of permutations and make a random selection from it.

```
1.  Start a new subset by choosing a neuron to act as the seed.
2   Select numNeur neurons connected to neurons in the subset that have
    not been selected before.
3.  if ( numNeur > 0 ) //Have found neurons connected to the subset
4.        Add neurons to the subset.
5.  else //No neurons connected to the subset – it is a complex
6.        Store details about the complex and return to step 1
7.  Calculate the new $\Phi$ of the subset, newPhi.
8.  if ( newPhi < oldPhi ) //Adding the neurons has reduced the $\Phi$
9.        Delete the added neurons and return to step 2.
10. else //Adding the neurons has increased the $\Phi$
11.       Leave the neurons in the subset, set oldPhi equal to newPhi and
          return to step 2.
```

Figure 7.5. Seed expansion algorithm

One advantage of the seed method is that it avoids all subsets with disconnected neurons and a Φ value of 0, whereas Tononi and Sporns' full analysis checks all subsets regardless of whether they contain disconnected neurons. The seed method also provides a way of identifying small complexes in large networks and it enables a limit to be set on the maximum size of the complexes, which is very useful for controlling the analysis duration.

The seed method does suffer from a number of potential and actual problems. To begin with, it can miss complexes that include subsets with higher Φ – for example the large complex in Tononi and Sporns (2003, Figure 7) was missed by this method (see Section 7.4.5). However, this was not a problem in the current analysis, which only aimed to identify the highest Φ complex that each neuron was involved in. A second disadvantage of the seed method is that the order of expansion may affect the final complex and in future work it would be worth doing some experiments to see if this is a significant effect. Finally, the seed expansion algorithm can lead to multiple calculations of Φ on the same subset, particularly when the neurons are highly connected together. Although this did occasionally happen during the analysis, it was not found to be a major issue.

A number of parameters were included in Network Analyzer to control the speed and accuracy of the seed expansion method:

- *Expansion rate per connection group*. This controls the number of neurons that are added to the subset at step 2 of the algorithm. Higher values of this parameter enable larger complexes to be identified in a shorter time, but smaller complexes may be missed when the expansion rate is greater than 1.
- *Maximum subset size*. The subset is discarded if it expands beyond this limit. This parameter is useful for searching for small complexes within a large neuron group and it was used extensively in this analysis because many seeds expanded into subsets that exceeded the available time and processing power.
- *Maximum number of consecutive expansion failures per connection group*. Some neural networks have large homogenous connections and the effect of adding one neuron from a homogenous connection group is likely to be the same as adding another neuron from the same group. When the number of failed attempts exceeds this limit, the entire connection group is discarded. For example, the network in this thesis has over 8000 connections with identical weights from each neuron in Inhibition to Vision Input. If the first twenty connections cannot be used to expand the subset, there is little reason to think that the next 8000 will, and it is more efficient to abandon the attempt to expand the connection group.[13]
- *Store Φ calculations*. When several neurons in the subset connect to the same external neuron, the same Φ calculation may be repeated several times and it might be thought that storing the results would be a good way to speed up the analysis.

[13] A variation of this approximation is to sample a random selection of neurons from a homogenous connection group. This option is available in Network Analyzer, but it was superseded by the consecutive expansion failures parameter.

However, this approach was not used in practice because the number of repeated calculations was not that large and it took a significant amount of processing to compare the neuron IDs in the current subset with the neuron IDs in each of the stored calculations.

Equal bipartitions

Another optimization strategy suggested by Tononi and Sporns (2003) is that "the bipartitions for which the normalized value of EI will be at a minimum will be most often those that cut the system in two halves, i.e. midpartitions" (p. 17). To evaluate how often mid partitions yield the minimum normalized effective information, seeds were selected from six of the layers and allowed to expand into a complex or up to a maximum subset size of 200 neurons. The percentage of times that the each bipartition had the minimum normalized effective information is plotted in Figure 7.6, which shows that mid partitions most often had the minimum normalized EI, but this was by no means always the case and during one of the seed expansions the mid partition only accounted for 40% of the minimum information bipartitions. When this approximation was applied in combination with the seed expansion method it was found that the occasional wrong expansion had a substantial effect on the final complex, and so this approximation was not used in the final analysis - although the timings presented in Section A3.3 show that the equal bipartition approximation can speed up the analysis by a factor of ten.

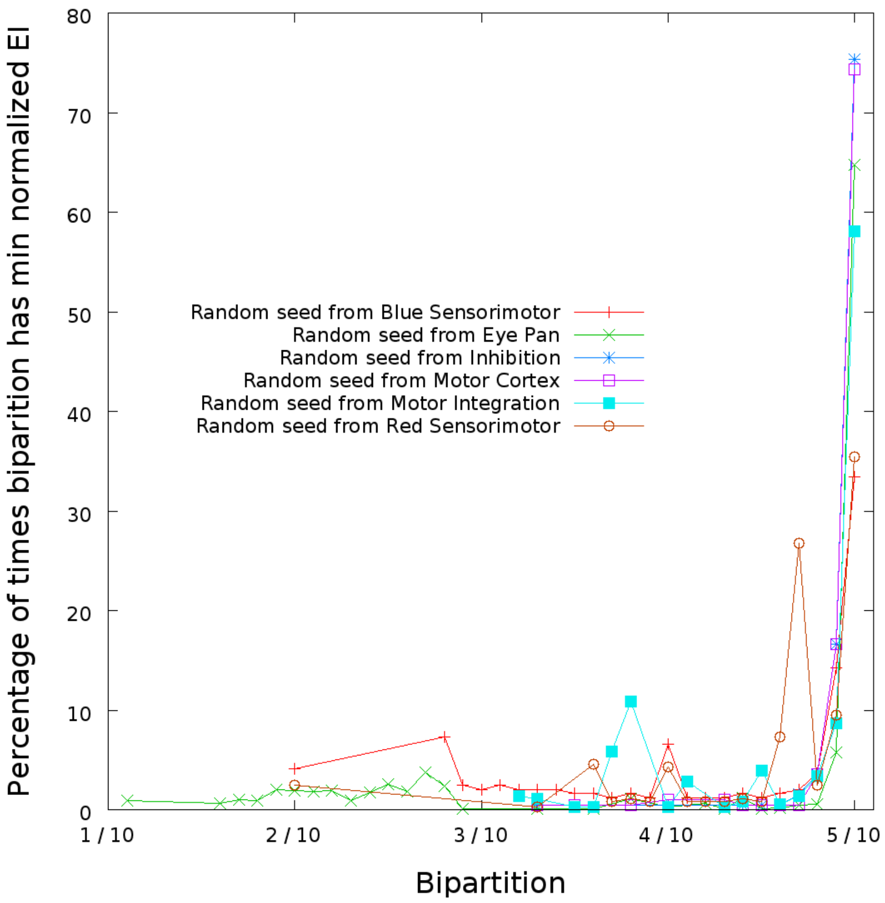


Figure 7.6. Percentage of times that different bipartitions had the minimum normalized effective information

Final strategy

The time taken to expand the seeds from each layer depends heavily on the complexes that are present in the network. For example, although Vision Input has 8,192 seed neurons, the analysis could be completed in 4.5 days because it identified a large number of complexes of approximately 30 neurons that were relatively quick to analyze. On the other hand, Inhibition has only 25 neurons, but it took 3.5 days to analyze because each seed neuron in this group had to be expanded up to the maximum subset size of 150 neurons. Since the complexes in the network were unknown at the start of the analysis, one or two test runs had to be carried out on each neuron group to identify the parameters that would enable the analysis to complete in a reasonable time. The seed expansion was then restarted on the neuron group and allowed to run to completion.

To fill in the gaps left by the seed-based analysis the Φ calculations were also run on combinations of neuron groups up to a maximum size of 700 neurons – a number that was found to be a reasonable compromise between the information gained about the network and the time available. These group analysis results are not complexes because it has not been shown that they are not included within a subset of higher Φ, and to make this distinction clear they will be referred to as *clusters*.

Although the seed and group analyses were carried out with a high level of approximation, enough information was gathered about the complexes and clusters of the network to allow predictions to be made about the network's phenomenology in Section 7.9. In the future if more accurate information about the complexes of the network could be obtained, then it would be easy to re-generate the predictions about consciousness using the improved information integration data.

7.4.5 Validation on Tononi and Sporns' Examples

The Network Analyzer code and the seed expansion method were tested on the examples supplied by Tononi and Sporns (2003) using the parameters given in Table 7.3.[14] These tests were mainly intended to establish that the seed expansion method could find the same complexes as the full analysis, and so the approximations were disabled by setting *Maximum number of consecutive expansion failures per connection group* to 1000 (greater than the number of connections in any of the examples) and *Max number of bipartitions per level* to 5000 (greater than the maximum number of possible bipartitions for this network). The results for this validation are given in Table 7.4.

[14] The connection matrices for the validation analysis were downloaded from: http://tononi.psychiatry.wisc.edu/ informationintegration/toolbox.html.

Parameter	Value
Max number of bipartitions per level	5000
Percentage of bipartition levels	100
Expansion rate per connection group	1
Maximum subset size	20000
Maximum number of consecutive expansion failures per connection group	1000
Only examine equal bipartitions	false

Table 7.3. Parameters for the validation on Tononi and Sporns' examples

Example Network	Seed Expansion Algorithm		Tononi & Sporns (2003) Analysis	
	Neurons	Φ	Neurons	Φ
Figure 2	1,2,3,4	20.8	1,2,3,4	21
	5,6,7	20.1	5,6,7	20
	1,2,3,4,5,6,7,8	7.4	1,2,3,4,5,6,7,8	7
Figure 3	1,2,3,4,5,6,7,8	73.9	1,2,3,4,5,6,7,8	73
	1,4	19.1	-	-
	3,5	19.6	-	-
Figure 4	1,2,3,4,5,6,7,8	5.8	1,2,3,4,5,6,7,8	5.8
	3,6	1.8	-	-
Figure 5	1,2,3,4,5,6,7,8	60.8	1,2,3,4,5,6,7,8	60
	1,2,3,4,6,7	40.5	-	-
	5,8	20.3	-	-
Figure 6	1,2,3,4,5,6,7,8	20.5	1,2,3,4,5,6,7,8	20.5
Figure 7	1,2	20.3	1,2	20.5
	3,4	20.3	3,4	20.5
	5,6	20.3	5,6	20.5
	7,8	20.3	7,8	20.5
	-	-	1,2,3,4,5,6,7,8	19.5

Table 7.4. The complexes found in Tononi and Sporns' (2003) example networks by the full analysis and using the seed expansion algorithm. The quoted Φ values for Tononi and Sporns' (2003) analysis are approximate readings from the graphs in their figures.

The results in Table 7.4 show that the seed expansion algorithm finds most of the complexes that were identified in Tononi and Sporns (2003) and that the Network Analyzer code performed accurate Φ calculations. However the seed expansion algorithm did identify a number of false complexes in figures 3, 4 and 5, and since none of the other approximations were being used, the most likely explanation is that the order of expansion of the neurons altered the complexes.[15] The only results from the information integration analysis that are used in the predictions about consciousness are the highest Φ complexes that each neuron is involved in (see Section 7.4.6). From this perspective the identification of false complexes is not a problem as long as the larger complexes with higher Φ that incorporate the smaller complexes are also found. On these examples, all of the highest Φ complexes were correctly identified by the seed expansion algorithm and the false complexes could have been easily eliminated by post-processing the seed analysis results.[16]

The only other disparity between the results from the seed algorithm and the full analysis are that the expansion algorithm can miss complexes that include smaller complexes with higher Φ – see the last row of the results in Table 7.4. This was also not a problem in an analysis which is only looking for the highest Φ complex that is associated with each neuron.

7.4.6 The Information Integration of the Network

Since there was a great deal of overlap between the different complexes and clusters, the results from the seed and group analyses were integrated together to identify the main complex, the independent complexes and the information integration between different parts of the network. More detailed results from the seed and group analyses and illustrations of some of the

[15] For example, suppose that the subset contains two neurons, A and B, and A is connected to another two neurons, C and D. It might be the case that adding C before D reduces the Φ of the subset, whereas adding C after D causes the Φ value of the subset to increase. It is also possible that adding C or D individually to the subset reduces its Φ, whereas adding both together increases it.

[16] For example, in the figure 3 example in Table 7.4, the seed method claims that neurons 1 and 4 form a complex with a Φ value of 19.1 and that these neurons are also part of another complex with Φ = 73.9. According to the definition of a complex, it is easy to see that the complex containing only neurons 1 and 4 is a false complex.

complexes are given in Appendix 3, and the results are also available in XML format in the Supporting Materials. To present the results as clearly as possible the neuron groups in figures 7.7 - 7.15 are labelled using the IDs in Table 7.5, which correspond to the IDs that were used for these neuron groups in the database.

ID	**Neuron Group**
24	Vision Input
28	Red Sensorimotor
29	Blue Sensorimotor
62	Emotion
34	Inhibition
61	Motor Cortex
60	Motor Integration
54	Eye Pan
55	Eye Tilt
53	Motor Output

Table 7.5. Neuron group IDs

According to Tononi and Sporns (2003) the *main* complex of the network is the one with the highest Φ. In this network the main complex has 91 neurons, a Φ value of 103 and it includes all of Inhibition, most of Emotion and small numbers of neurons from Vision Input, Red Sensorimotor, Motor Output, Eye Tilt and Motor Integration (see Figure 7.7). Tononi (2004) claims that the main complex is the conscious part of the network.

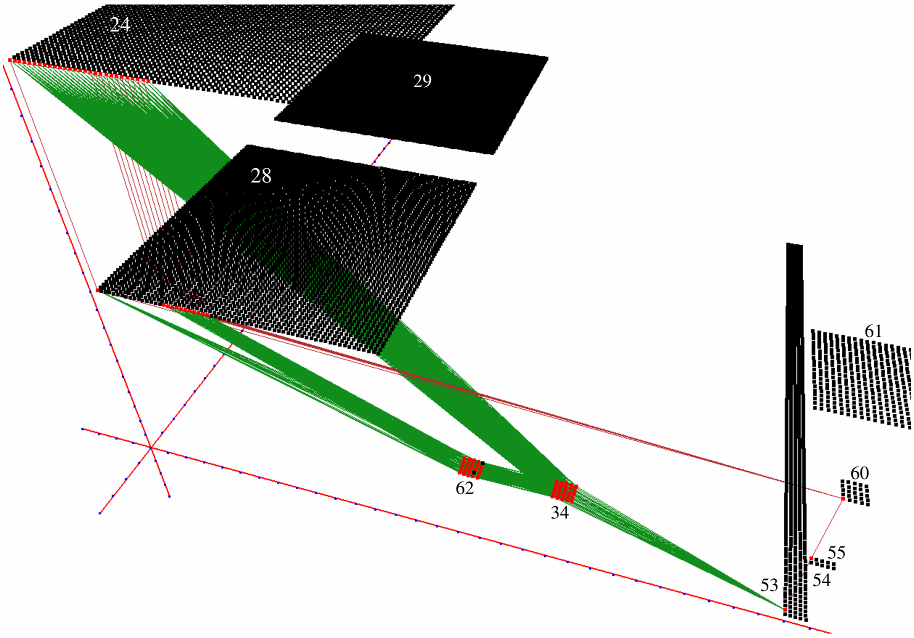


Figure 7.7. The main complex of the network.

A second aspect of information integration is whether different parts of the network integrate their information in isolation from each other (see Section 4.3.6). In this analysis, the notion of an *independent* complex is defined as follows:

None of the neurons in an independent complex, A, are part of another complex, B, that has higher Φ than A. (7.1)

This definition was used to search for independent complexes in the network, and it turned out that the main complex was the only independent complex, with all of the other complexes and clusters having some overlap with the main complex and thus not being independent by this definition.

In order to understand the information integration between different parts of the network, ten neurons were selected at random from each neuron group and the complex(es) with the highest Φ that each neuron was involved in were identified. Only the highest Φ complexes were considered because the phenomenal predictions in sections 7.5 and 7.7 are based on the maximum information integration of each mental state, and the most significant information relationships of each neuron will be with other neurons in its highest Φ complex. The results from this analysis were as follows:

Vision Input

All of the sampled neuron's highest Φ complexes included Inhibition and different combinations of Blue Sensorimotor, Red Sensorimotor, Emotion and Motor Output. Amongst the sampled neurons, the typical highest Φ complex contained 29-31 neurons, with Φ ranging from 75-93.

Red Sensorimotor

All of the sampled neuron's highest Φ complexes included Inhibition and Vision Input, along with different combinations of Blue Sensorimotor, Emotion, Motor Integration and Motor Output. Amongst the sampled neurons, the typical highest Φ complex contained 29-31 neurons, with Φ ranging from 75-93.

Blue Sensorimotor

All of the sampled neuron's highest Φ complexes included Inhibition and Vision Input, along with different combinations of Blue Sensorimotor, Emotion, Motor Integration and Motor Output. Amongst the sampled neurons, the typical highest Φ complex contained 29-31 neurons, with Φ ranging from 75-93.

Emotion

Although this neuron group was strongly integrated with itself, higher values of Φ were found in complexes that included Inhibition and other layers. The sampled neurons' highest Φ complex was the main complex.

Inhibition

All of the sampled neuron's highest Φ complexes were part of the main complex. The Inhibition layer is a key part of many high Φ complexes because of its recurrent connections and its large number of strong connections to Vision Input and Motor Output. On its own Inhibition has a Φ of 77.3 and this increases to more than 129 when it is combined with a small number of neurons from other layers.[17]

Motor Cortex

Despite a large number of recurrent connections, Motor Cortex only had a Φ value of 17.9 when it was measured by itself. The sampled neurons had two highest Φ clusters: one with Φ = 59 and 425 neurons from Motor Cortex and Motor Integration, and another with Φ = 59 and 435 neurons from Motor Cortex, Motor Integration, Eye Pan, and Eye Tilt.

Motor Integration

One of the sampled neurons in Motor Integration had 129 highest Φ complexes with Φ=75 and 25 neurons from other layers. Some of the other highest Φ complexes of the sampled neurons had 75-91 neurons and Φ ranging from 84-103. Motor Integration also had sampled neurons that were not included in any of the complexes identified by the seed-based analysis. These had two highest Φ clusters: one with Φ=58.7 and 425 neurons from Motor Cortex and Motor Integration, and another with Φ=58.7 and 435 neurons from Motor Cortex, Motor Integration, Eye Pan and Eye Tilt.

Eye Pan

One of the seeds in this layer expanded beyond the maximum subset size of 150 and its highest Φ value came from the group analysis, which identified two highest Φ clusters: one with Φ=58.7

[17] Some of the subsets expanded from Motor Output included Inhibition and achieved a Φ value of 129 before the maximum subset size was exceeded.

and 425 neurons from Motor Cortex and Motor Integration, and another with Φ=58.7 and 435 neurons from Motor Cortex, Motor Integration, Eye Pan and Eye Tilt. The other four neurons in this group had highest Φ complexes with 75-79 neurons from all of the other layers and Φ ranging from 84–102.

Eye Tilt

The sampled neuron's highest Φ complexes had 71-91 neurons from some or all of the other layers and Φ ranging from 80–103.

Motor Output

The sampled neuron's highest Φ complexes had Φ=57 and 22 neurons from Inhibition. Ten of the neurons in Motor Output, which were not included in the random sample, are connected through Eye Pan and Eye Tilt into complexes with Φ up to 103.

These results show that the highest Φ complexes of neurons in different layers have a consistent level of information integration that typically ranges from 58 - 103. The most important neuron group for information integration was Inhibition, which played a central role in many of the complexes with higher Φ.

7.4.7 Previous Work on Information Integration

Evidence for a link between information integration and consciousness was provided by Lee et al. (2007), who made multi-channel EEG recordings from 8 sites in conscious and unconscious subjects and constructed a covariance matrix of the recordings on each frequency band that was used to identify the complexes within the 8 node network using Tononi and Sporns' (2003) method. This experiment found that the information integration capacity of the network in the gamma band was significantly higher when subjects were conscious.

Theoretical work on information integration has been carried out by Seth et al (2006), who identified a number of weaknesses in Tononi and Sporns' (2003) method and criticized the link between information integration and consciousness. To begin with, Seth et al. showed that simple Hopfield-type networks can be designed to have arbitrary values of Φ, which suggests that Φ may not be an adequate sole measure of the consciousness of a system. A second problem identified by Seth et al. is that the value of Φ depends on arbitrary measurement choices made by the observer. Different descriptions of the system lead to different predictions about its information integration, and Seth et al. demonstrate that a simple continuous system consisting of two coupled oscillators can generate arbitrary and even infinite values of Φ depending on the measurement units that are used. Both of these criticisms highlight the fact that Tononi and Sporns' (2003) method is at an early stage of development and needs further refinement to increase the accuracy of its predictions about real biological networks. Seth et al. also point out that Φ is essentially a static measure of consciousness, which makes it unable to distinguish between a conscious and an unconscious brain, and they discuss the difficulties of calculating the information integration of a realistic system.

Tononi and Sporns' (2003) Φ measure is based on their earlier work on neural complexity (Tononi et al. 1994, 1998). Neural complexity is defined as the average mutual information that is shared between a subset of the network and the rest of the system, where this average is taken over all subset sizes. Whilst Tononi and Sporns' (2003) method looks for the minimum information bipartition of the subset and introduces the concept of a complex, neural complexity is calculated once for the whole network without searching for the most integrated part. The computation cost of calculating neural complexity increases factorially in a similar way to effective information, but it can be approximated by limiting the analysis to bipartitions between a single element and the rest of the network (Seth et al. 2006). Since neural complexity depends solely on mutual information it is only a measure of functional and not effective connectivity.[18]

Another way of measuring effective connectivity is the causal density measure put forward by Seth et. al. (2006), which identifies the causally significant interactions amongst a network's elements using Granger causality, and then calculates the causal density using Equation 7.16:

$$cd = \frac{\alpha}{n(n-1)}, \tag{7.16}$$

where *cd* is the causal density, α is the total number of significant causal interactions and *n(n - 1)* is the total number of directed edges in a fully connected network with *n* nodes.[19] Causal density depends on a comprehensive set of test data because it is calculated using the actual activity of the network, and it also has scaling problems since the multivariate regression models become difficult to estimate accurately as the number of variables increases. However, these scaling problems are substantially less serious than the factorial dependencies associated with neural complexity and Φ.

There has also been a substantial amount of analysis of the anatomical, functional and effective connectivity of biological networks, either using scanning or electrode data, or based on large-scale models of the brain. For example, Honey et al. (2007) used transfer entropy to study the relationship between anatomical and functional connections on a large-scale model of the macaque cortex, and demonstrated that the functional and anatomical connectivity of their model coincided on long time scales. Other examples of this type of work are Brovelli et al. (2004), who used Granger causality to identify the functional relationships between recordings made from different sites in two monkeys as they pressed a hand lever during the wait discrimination

[18] See Sporns et al. (2004) for the difference between anatomical, functional and effective connectivity.

[19] Granger causality has also been used by Seth and Edelman (2007) to identify causal cores within a large network.

task, and Friston et al. (2003), who modelled the interactions between different brain areas and made predictions about the coupling between them. There is also the work by Massimini et. al. (2005) who measured the cortical effective connectivity during non-REM sleep and waking. An overview of this type of research can be found in Sporns et. al. (2004) and Sporns (2007).

7.4.8 Information Integration: Discussion and Future Work

The seed expansion method was found to be an effective way of speeding up the calculations and offered a valuable way of controlling the analysis time by limiting the maximum subset size. However, this method did have the problem that errors introduced by other approximations could lead to erroneous expansions of the subset, and it is also probable that the order of expansion of the connected neurons significantly altered the final complex. Future work in this area could evaluate the effect of different expansion orders on the complexes found in the network.

One possible improvement to this analysis would be to use a shuffling algorithm to randomly select different neurons from homogenous connections, in order to identify complexes with similar Φ and connection patterns. For example, the high information integration of the main complex partly depends on connections to Vision Input that are selected from a large uniform set, and a different selection of these connections could be used to identify a different complex with similar Φ.

In this analysis, the main compromise between speed and accuracy was the limitation on the number of calculations per bipartition, which had a big effect on the calculation time (see Figure 7.4 and Table A3.12 in Appendix 3), and a proportionally greater impact on larger networks. On most calculations this approximation would have made the final Φ higher than it actually was by reducing the number of bipartitions that were examined for the minimum normalized effective information. However, in some circumstances this approximation might have artificially reduced the Φ by changing the way in which the subset expanded.

Although the equal bipartition approximation speeded up the analysis considerably, the results in Figure 7.6 show that a significant number of other bipartitions had the minimum normalised effective information. When this approximation was combined with the seed method it led to substantially different complexes, and so it was not used in the final analysis. In future work it would be worth investigating the strengths and limitations of this approximation in more detail and it might be possible to use the structure of the network to decide when the equal bipartition approximation is most likely to be accurate.

The main limitation of this analysis was the extremely long time that was required to calculate Φ. One way of addressing this problem would be to use graphics cards for the matrix calculations - for example, using the NVIDIA CUDA system.[20] Although this analysis did run partly in parallel by expanding the seeds from different neuron groups on different computers, the code could be rewritten to automatically distribute itself across an arbitrary number of processors. This would enable it to run on supercomputers and address some of the memory limitations that were encountered with large neuron groups.[21]

Work is already in progress on the simulation of networks with a billion spiking neurons (see Section 5.6) and on networks of this size even supercomputing power will not be enough to identify the complexes of the network. Future work should investigate other methods of estimating the effective connectivity of neural networks, such as Seth et. al.'s (2006) causal density measure, and it would also be worth investigating whether Φ can be estimated on the basis of sub-samples of each bipartition.

A further limitation of Tononi and Sporns' (2003) method is that it is essentially static and ignores the fact that complexes in a real network might change over time. In future work, it would be much better to record the network as it interacts with the world and use transfer

[20] NVIDIA CUDA: http://www.nvidia.com/object/cuda_home.html.

[21] For example, the Φ of Vision Input could not be calculated because it used more than 2GB of RAM, which was the maximum that could be installed on the computers used for this analysis.

entropy (Schreiber 2000) or a similar method to identify the effective information that is integrated across different bipartitions of each subset. It would also be worth analyzing the system at a number of different levels - for example, using populations of neurons, ion channels and memory addresses as well as neurons – to increase our understanding of the difference between simulated and physical systems.

In the next three sections definitions based on Tononi's, Aleksander's and Metzinger's theories of consciousness are developed, which are used make predictions about the phenomenology of the network in Section 7.9.

7.5 Phenomenal Predictions based on Tononi's Information Integration Theory of Consciousness

Tononi (2004) makes an explicit connection between the consciousness of a system and its capacity to integrate information: "consciousness corresponds to the capacity to integrate information. This capacity, corresponding to the quantity of consciousness, is given by the Φ value of a complex." This link between Φ and consciousness is independent of the material that the system is made from, but there is not a simple proportional relationship between Φ and consciousness because only the main complex is capable of consciousness according to Tononi's theory – parts of the system that are outside the main complex are completely unconscious.

When complexes overlap it seems reasonable to follow Tononi (2004) and only allocate consciousness to the one with the highest Φ.[22] However, when complexes do not overlap and exchange relatively little information, it seems more sensible to attribute two consciousnesses to the system, rather than saying rather arbitrarily that the one with slightly higher Φ is conscious and the other not conscious at all. To accommodate this type of case without including all of the independent complexes of the system, this analysis will consider a firing neuron to be conscious

[22] The problems with this are discussed in Section 7.9.4.

according to Tononi's theory if it is part of the main complex or if it is part of an independent complex whose Φ value is at least 50% that of the main complex. The explicit definition is as follows:

> *A mental state will be judged to be included in the phenomenally conscious part of the system according to Tononi if it is part of the main complex or if it is part of an independent complex whose Φ is 50% or more of the Φ of the main complex. The* ***amount*** *of consciousness will be indicated by the Φ of the complex.* (7.2)

The results from the information integration analysis showed that the main complex was the only independent complex, and so Tononi's theory predicts that the 91 neurons in the main complex will be the only parts of the network that are associated with conscious states. Tononi (2004) claims that the amount or quantity of consciousness in the conscious part of the network is given by the Φ value of the main complex, which is 103.

7.6 Phenomenal Predictions based on Aleksander's Axioms

7.6.1 Is the System Synthetically Phenomenological?

In earlier work, Aleksander and Dunmall (2003) set out five axiomatic mechanisms and claimed that these are minimally necessary for consciousness (see Section 2.6.3). Objects that did not possess these mechanisms were not considered to be conscious according to this theory. Over the last few years Aleksander's thinking has evolved and he now emphasises the importance of depiction over the other axioms, as illustrated in the following quotation:

> **Def 1:** To be **synthetically phenomenological**, a system S must contain machinery that represents what the world and the system S within it *seem* like, from the point of view of S. …
>
> **Def 2:** A **depiction** is a state in system S that represents, as accurately as required by the purposes of S the world, from a virtual point of view within S.

> **Assertion 1:** A depiction of Def. 2 is the mechanism that is necessary to satisfy that a system be synthetically phenomenological according to Def. 1.
>
> Aleksander and Morton (2007a, p. 72)

This section will take a brief look at whether the neural network developed in this thesis conforms to all five of Aleksander's axioms, but it will only consider the network to be capable of consciousness (or synthetically phenomenological) if it includes depiction.

1. Depiction

Although the network described in this paper does not have gaze locked cells, the neurons in Red Sensorimotor and Blue Sensorimotor are connected to both Vision Input and Motor Integration, and respond to both visual data and the motor signals sent to control the eye, which contain proprioceptive information. These observations are confirmed by the measurements of representational mental states in Section 7.3, which showed that neurons in Red Sensorimotor and Blue Sensorimotor share mutual information with Vision Input and Motor Integration. It also appears to be consistent with the interpretation of depiction in this thesis that it could be implemented as a population code in which the *combined* activity of the motor and visual layers represents the presence of an out there world. In this case some kind of binding or integration between the motor and visual layers would be all that was needed for depiction.

2. Imagination

The network has an offline mode in which it can 'imagine' the consequences of different motor actions without carrying them out.

3. Attention

This network's 'imagination' is used to select the part of the world that is looked at by the system.

4. Volition

When Vision Input and Motor Output are inhibited, the 'imagination' circuit decides which part of the world to look at and then executes the selected motor action based on the response of its 'emotion' layer.

5. Emotion

The neural network has an 'emotion' layer, which responds in a hardwired way to different characteristics of the world with a high impact low information signal that is characteristic of the neuromodulatory aspect of emotion (Arbib and Fellous 2004). However, it could be argued that this 'emotion' layer does not directly represent the state of SIMNOS's body, and so it is at best something like the 'as if' circuit discussed by Damasio (1995). Other limitations of the 'emotional' response are that it does not modulate the way in which neurons and synapses compute and it lacks the detail that we sense when our viscera and skeletal muscles are changed by an emotional state, such as fear or love (Damasio 1995, p. 138). These limitations do not completely exclude the possibility that the 'emotional' response of the network can be counted as an emotion, and so it will be provisionally accepted as a very primitive emotion that is much simpler than our basic human emotions.

This discussion suggests that the neural network in this thesis is capable of depiction and minimally conforms to Aleksander's other axioms, and so it is likely to possess a very simple form of consciousness according to this theory. Since the network is simulated and operates very differently from a real biological network on a much smaller scale, the contents and qualitative character of this consciousness will be very different from the consciousness of biological creatures that have the axiomatic mechanisms.[23]

[23] These differences are likely to be much greater than those identified by Nagel (1974) between human and bat consciousness.

7.6.2 What are Aleksander's Predictions about Phenomenal States at Time *t* ?

In this analysis, predictions about phenomenal states according to Aleksander's theory are based on his link between depiction and consciousness. The depictive neurons are identified by using the method set out in Section 7.3 to look for representational relationships between input/ output neurons and internal states of the system. Under these experimental conditions, high mutual information between an input/ output and internal neuron indicates a strong representational relationship, and so an internal neuron that shares a high level of mutual information with both visual and proprioceptive data is likely to be depictive. Since depictive neurons are defined by the fact that they respond to both sensory and proprioceptive data, the amount of depiction will be limited by whichever of these is smallest. This leads to the following definition:[24]

> *A mental state will be judged to be within the phenomenally conscious part of the system according to Aleksander if it shares mutual information with both sensory and proprioceptive layers. The* ***amount*** *of consciousness will be measured by the minimum mutual information that is shared with sensory and proprioceptive layers. So, for example, if the neuron has 0.4 mutual information with an auditory input layer, 0.2 mutual information with a visual input layer and 1.0 mutual information with a proprioception layer, then its amount of consciousness would be judged to be min{0.4, 0.2, 1.0} = 0.2, according to Aleksander's theory.* (7.3)

Based on this definition, the only parts of the network that share mutual information with both visual input and proprioception/ motor output are Red Sensorimotor and Blue

[24] It might be thought that the sensory and motor mutual information values could be added or multiplied together to get the amount of depiction. However, consider two neurons: neuron A that has 1000 mutual information with visual input and 0.1 mutual information with motor output, and neuron B that has 10 mutual information with visual input and 10 mutual information with motor output. A's strong response to visual information makes it much more like a photographic representation, whereas neuron B is much closer to the gaze-locked neurons discovered by Galletti and Battaglini (1989) that respond to a particular combination of sensory and muscle information, and are cited by Aleksander (2005) as a key example of depictive neurons. In this example, addition of the mutual information values gives 1000.1, for neuron A and 20 for neuron B, which erroneously suggests that neuron A is more depictive than neuron B. The product of the mutual information values gives 100 for neuron A and 100 for neuron B, which is also an incorrect measure of their relative levels of depiction. In this example, the minimum of the two values, which is 0.1 for neuron A and 10 for neuron B most accurately predicts which neuron is most depictive.

Sensorimotor, and so activity in these parts of the network will be conscious according to Aleksander's theory. For the phenomenal predictions in Section 7.9 the mutual information values were normalized to the range 0-1, and so the maximum amount of consciousness is 1. In the future, methods such as transfer entropy (Schreiber 2000, Sporns and Lungarella, 2006), backtracing (Krichmar et. al. 2005) and Granger causality (Seth and Edelman 2007) could be used to identify the depictive parts of the network.

7.7 Phenomenal Predictions based on Metzinger's Constraints

7.7.1 Is Artificial Subjectivity Possible?

Although Metzinger (2003) believes that machines are capable of consciousness, he points out that our current simulations and robotic models are too coarse to replicate the extremely fine levels of detail of biological systems:

> The subtlety of bodily and emotional selfhood, the qualitative wealth and dynamic elegance of the *human* variety of having a conscious self, will not be available to any machine for a long time. The reason is that the microfunctional structure of our emotional self model simply is much too fine-grained, and possibly even mathematically intractable. … Self-models emerge from elementary forms of bioregulation, from complex chemical and immunological loops—and this is something machines don't possess.
>
> Metzinger (2003, p. 619)

One way of developing machines with a fine-grained biological structure is to use biological neurons to control a real or virtual robotic body, as was done in the work of DeMarse et al. (2001). Metzinger also points out that consciousness is a *graded* phenomena and that there are degrees of constraint satisfaction and phenomenality: "just as with animals and many primitive organisms surrounding us on this planet, it is rather likely that there will soon be artificial or postbiotic systems possessing simple self-models and weaker forms of conscious experience in our environment." (Metzinger 2000, p. 620). If consciousness is graded, then systems as simple as the neural network developed by this project may be capable of an extremely limited form of consciousness if they can satisfy Metzinger's minimal set of constraints.

7.7.2 Does the Network Conform to Metzinger's Constraints?

Although Metzinger describes his constraints on conscious experience at a number of different levels, these descriptions remain at a fairly high level of abstraction and in some cases it is quite difficult to say whether the network developed by this thesis matches them or not. This section is a general discussion about the degree to which the network conforms to the constraints; a more precise definition of what it would mean for the network to conform to Metzinger's minimal definition of consciousness is given in the next section.

1. Global availability

The network can access information in different parts of its real and imaginary environment and this information is available for the control of action, and so the network does possess a limited form of global availability. Metzinger links global availability with Tononi's earlier work on information integration, and so it might be possible to use Φ to measure this constraint.

2. Window of presence

Activity within the network does exist in a single now and there is a certain amount of temporal integration along the connections with different delays. The reverberatory activity within Emotion, Inhibition and Motor Cortex also stores a limited amount of information about earlier states of the system. Taken together these observations suggest that the window of presence of the network is very thin, but not completely non-existent.

3. Integration into a coherent global state

Global availability (constraint 1) is a functionalist-level description of global integration, which Metzinger links to Tononi's earlier work on information integration. This suggests that Φ could be used to measure the degree to which the network integrates information into a coherent global state.[25]

4. Convolved holism

The visual processing of the neural network is too basic to identify wholes at different levels of scale, and so it does not even minimally conform to this constraint. In the future, more complex processing could be added to the network to enable it to identify part-whole relationships.

5. Dynamicity

The network can sustain the activation of a neuron group over time, but it has a very limited ability to integrate information between points in time and it is not sensitive to the part-whole structure of temporal information.

6. Perspectivalness

This constraint has a certain amount of overlap with Aleksander's depiction axiom and the network's integration between sensory and proprioceptive information should give it some kind of rudimentary sense of seeing the world from somewhere. Since the size of objects changes with distance and the network only perceives part of the world at any one time, there is also some sense to the idea that it has a perspective.

[25] Metzinger (2003) was published in the same year as Tononi and Sporns (2003), and so it is unlikely that Metzinger (2003) knew about Tononi's work on Φ. Metzinger's more recent work, such as Metzinger and Windt (2007) and Metzinger (2008), has focused on the phenomenal self model and the phenomenal model of the intentional relation.

7. Transparency

Since the network lacks internal sensors there is some basis to the claim that it is as transparent as a biological neural network, such as the brain. However, Metzinger distinguishes between conscious and unconscious transparency and claims that almost nothing is known about the neural basis of phenomenal transparency. This suggests that we have no reason to believe that the neural network is *less* transparent than the human brain, but much more research needs to be done on transparency.

8. Offline activation

This constraint is similar to Aleksander's second axiom of imagination and the system is capable of inhibiting its sensory input and motor output whilst it 'imagines' an eye movement that would look at a red or blue object.

9. Representation of intensities

Information in the network is held as neurons that spike at different rates, and so this constraint is implemented by the system.

10. "Ultrasmoothness": the homogeneity of simple content.

Although individual neurons represent individual areas of colour, there is no representation within the system of the gaps between neurons, and so the network cannot access the graininess of the neurons' spatial firing patterns that is visible to us as outside observers. The network is also unable to represent the graininess of its temporal representations, and so it is probably reasonable to claim that its mental states are ultrasmooth.

11. Adaptivity

This network did not come about through natural selection, and so it does not conform to this constraint.

7.7.3 What are Metzinger's Predictions about Phenomenal States at Time *t* ?

The discussion in the previous section demonstrated that the network is likely to conform to a number of Metzinger's constraints, including a coherent global model of reality (*constraint 3*), a window of presence (*constraint 2*) and transparency (*constraint 7*), which are sufficient for Metzinger's minimal notion of consciousness. In this analysis, the degree to which a mental state is involved in a coherent global model of reality will be indicated by the Φ value of the highest Φ complex that it is involved in. Since recurrency is a key way in which information can be integrated over time, a window of presence will be attributed to neurons whose highest Φ complex includes a recurrent part of the system. Transparency will be left out of this analysis because it cannot be directly identified, and it has been argued that we do not have any reason for believing that the network is less transparent than the human brain. The final definition is as follows:

> *A mental state will be judged to be minimally conscious according to Metzinger if the highest Φ complex that it is involved in includes one or more recurrent layers. The* ***amount*** *of consciousness will be indicated by the Φ of this complex.* (7.4)

According to this definition, the conscious parts of the network will be the complexes that include Motor Cortex, Emotion and Inhibition. The amount of consciousness will be the Φ of these complexes.

7.8 Other Phenomenal Predictions

For the reasons discussed in Section 2.6.1, only three theories of consciousness are being used to make predictions about the consciousness of the network in this thesis. However, to provide more context for this work I will make brief remarks about some other theories that make fairly explicit predictions about the consciousness of the network. None of these predictions were used to generate the final XML description in Section 7.9.

Pantheism

Pantheists, such as Spinoza (1992), believe that all matter is conscious to some degree, and so the physical computer running the simulation is conscious even when it is switched off. From this perspective, the task of synthetic phenomenology is to determine the amount of consciousness in the system and the qualitative character of this consciousness at different points in time. Pantheism is a type I theory because the behaviour of the system does not affect the attribution of consciousness to it.

Information states

Chalmers (1996, p. 292) claims that conscious experiences are realizations of information states, and so systems as simple as thermostats are conscious because they contain information. Since the neural network contains a large number of information states, it is conscious according to this hypothesis. This link between consciousness and information states is a type I theory because every object in the universe interacts to some degree and stores ‘information’ about the particles and forces affecting it.

Non-biological systems cannot be conscious

A number of people would argue that the neural network developed by this project can never become conscious because it is a simulated artificial system (Searle 2002) or because the calculations that are used to simulate it are all algorithmic (Penrose 1990, 1995). These theories are discussed in detail in Section 3.4.

Internal models

Holland (2007) claims that internal models play an important role in our conscious cognitive states and may be a cause or correlate of consciousness in humans (see Section 3.5.2). In this network, the activity in Motor Cortex, Motor Integration, Eye Pan, Eye Tilt and Motor Output accurately reflects the position of SIMNOS's eye 'muscles', and this could be interpreted as an internal model in an extremely limited sense. This internal modelling could be made more realistic by making the activity in Emotion reflect the internal states of SIMNOS's body, which would also link the network more closely to Damasio's (1995) work.

7.9 XML Description of the Phenomenology of the Network

7.9.1 Introduction

This section explains how the data about representational mental states and complexes was integrated with definitions 7.2 - 7.4 to generate a sequence of XML files that predicts the phenomenology of the network at each time step. The first parts of this procedure were two recordings of the network, which are documented in Section 7.9.2. The next section explains how the XML files were generated, and then sections 7.9.4 – 7.9.6 examine the predictions that were made about the consciousness of the network using Tononi's, Aleksander's and Metzinger's theories of consciousness. After discussing what these results show about the relationship between consciousness and action, some extensions and enhancements of the consciousness of the network are suggested in Section 7.9.8, and the analysis concludes with a discussion and suggestions for future work.

7.9.2 Analysis Data

The main data for this analysis was recorded as the neural network moved SIMNOS's eye and used its 'imagination' to avoid looking at the blue cube, as described in Section 5.5.1. The recording starts at time step 13100 with the network in its online perception mode and an empty visual field. At time step 13138 a red object starts to appear in the top left corner of the visual field and this moves in and out of view until time step 13503, when a blue object appears in the bottom left corner of the visual field. This leads to the activation of Inhibition after time step 13520 and the system switches into its offline 'imagination' mode. At time step 13745 the system 'imagines' a blue blob in the left half of its visual field and eventually it 'imagines' a red object at time step 13945, which activates Emotion and returns the system to online perception. Finally at time step 13966 the network starts to perceive a red object in the top left corner of its visual field. This recording of data from time steps 13100 to 14004 will be referred to as "Analysis Run 1", and a video of Analysis Run 1 is included in the supporting materials.

The average number of times that each neuron fired during Analysis Run 1 was recorded and the results were normalized to the range 0-1 and used to illustrate the activity of the network in Figure 7.8. This shows that Inhibition was the most active part of the network, followed by Emotion. Traces of motor and visual activity can also be seen in Figure 7.8.

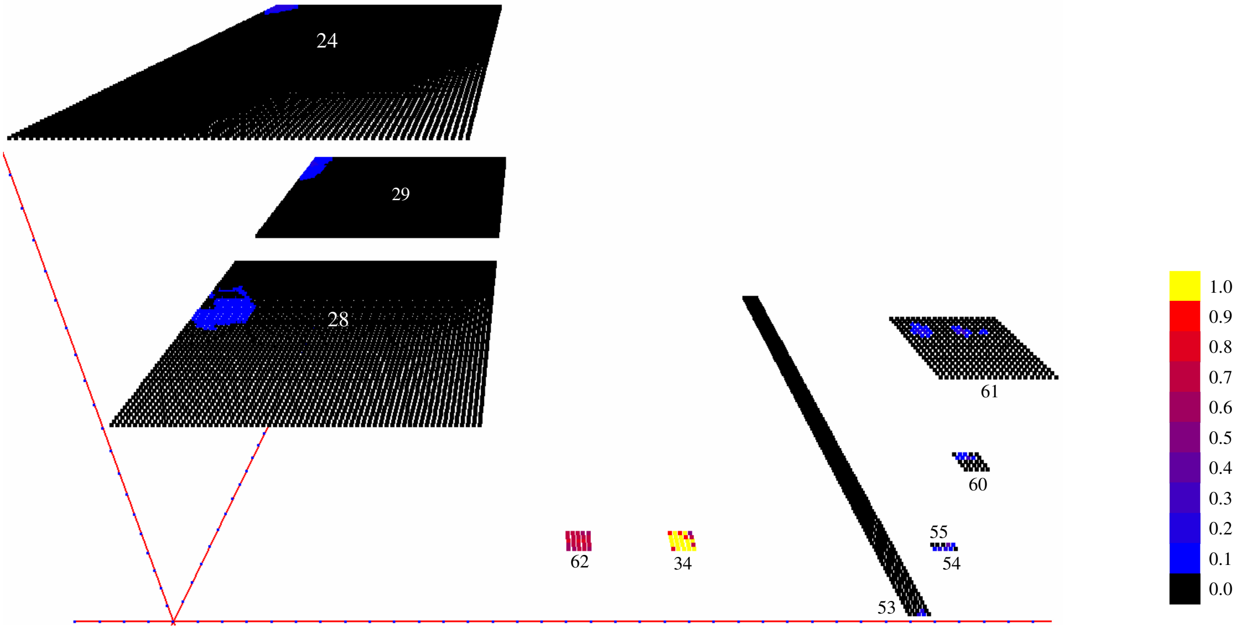


Figure 7.8. Normalized average firing frequency of neurons during Analysis Run 1

A recording was also made in which the neuron groups were disconnected from each other and themselves and 5% noise was injected into each layer at each time step for 100 time steps. The normalized average firing frequency of each neuron was used to illustrate the activity of the network in Figure 7.9, which shows that there was a reasonably even spread of activity across the layers. This noise recording will be referred to as "Noise Run 1".

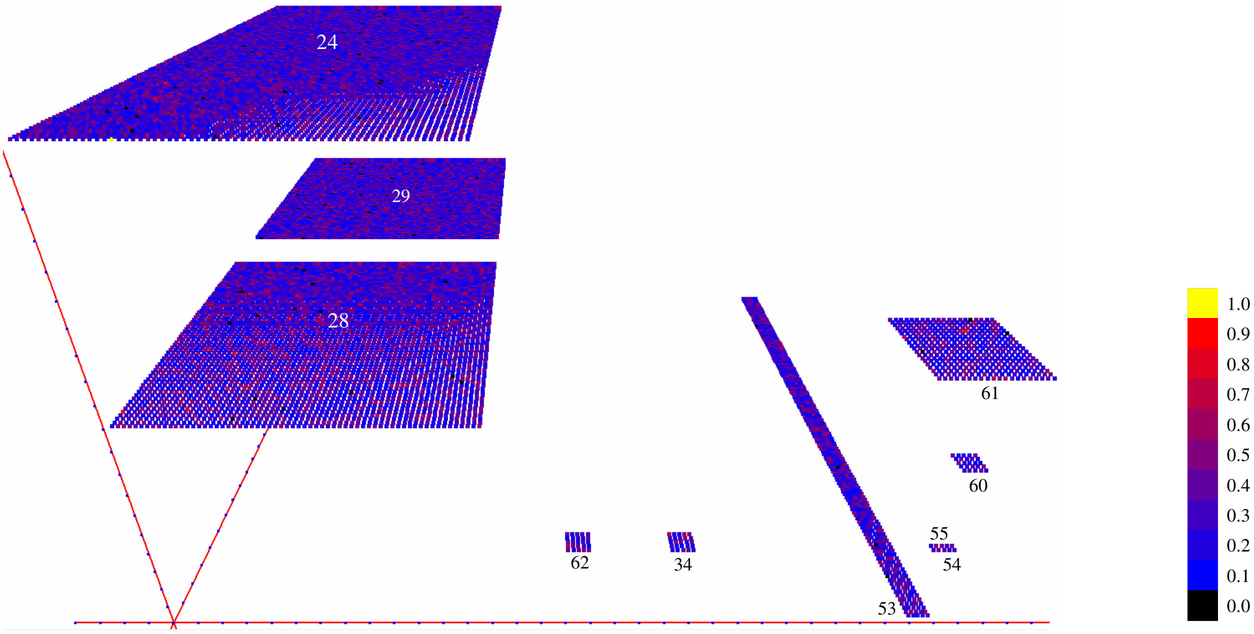


Figure 7.9. Normalized average firing frequency of neurons during Noise Run 1

The data from Analysis Run 1 can be used to predict the *actual* consciousness that was experienced by the network as it interacted with the world. However, in this recording only a small part of the network was active, and so it does not tell us about the consciousness that might be predicted to be associated with the other parts of the network. On the other hand, the noise data has an even spread of activity that includes all of the neurons, but it was recorded with the layers disconnected from themselves and each other, and so the predictions about the consciousness of the network during Noise Run 1 are made *as if* the noise patterns had been present when the network was fully connected. In other words, the noise data provides a useful way of understanding the *potential* for consciousness of the different parts of the network.

7.9.3 Generation of the XML Description

To generate the XML files, the recordings of the network's activity were combined with the OMC rating, representational mental states, complexes, clusters and definitions to produce a sequence of XML files describing the phenomenology of the system at each time step. As discussed in Section 7.3.1, firing neurons are being treated as mental states for this analysis and the predictions about the consciousness associated with each mental state are given by definitions 7.2 – 7.4. It was decided not to normalize the predictions based on Tononi's and Metzinger's theories of consciousness, both because Φ does not have a maximum value and because Tononi interprets Φ as an absolute measure of a system's consciousness. The predictions based on Aleksander's theory were normalized to the range 0-1 by dividing the mutual information by the maximum possible mutual information of 0.72.[26]

In addition to the representational mental states identified in sections 7.3.4 and 7.3.5, the neurons in the input and output layers were also treated as representational mental states in the final XML description and assigned a mutual information value of 1 to reflect the fact that they shared the maximum amount of mutual information with themselves. The other mutual information values for the representational mental states were normalized by the maximum possible mutual information. In the integration part of the description, neurons that were not included in any complex were assigned a Φ value of zero.

In order to compare the different theories' predictions about the distribution of consciousness associated with the network, the amount of predicted consciousness per neuron was averaged over Analysis Run 1 and Noise Run 1, normalized to the range 0-1 and used to highlight the network in figures 7.10 - 7.15. I have only shown the *relative* distribution of consciousness in the network because the assignment of absolute values to predicted

[26] See Section 7.3.2 for the calculation of this value. In practice the normalized values occasionally strayed over 1.0 due to noise in the data.

consciousness is largely meaningless without some form of *calibration* on humans – a problem that is discussed in Section 7.9.9.

7.9.4 Predictions about the Consciousness of the Network According to Tononi's Theory

Tononi's theory predicts that the main complex is the only conscious part of a system and that the amount of consciousness in the main complex can be measured by its Φ value. In this network the main complex has a Φ of 103 and it includes all of the neurons highlighted in Figure 7.7. The predicted consciousness of the network at each point in time is therefore the intersection of the neuron activity with the main complex. In Noise Run 1 there is fairly uniform activity across the network, and so the distribution of consciousness for Noise Run 1 is an extract from the average activity shown in Figure 7.9 that is shaped like the main complex (see Figure 7.10).

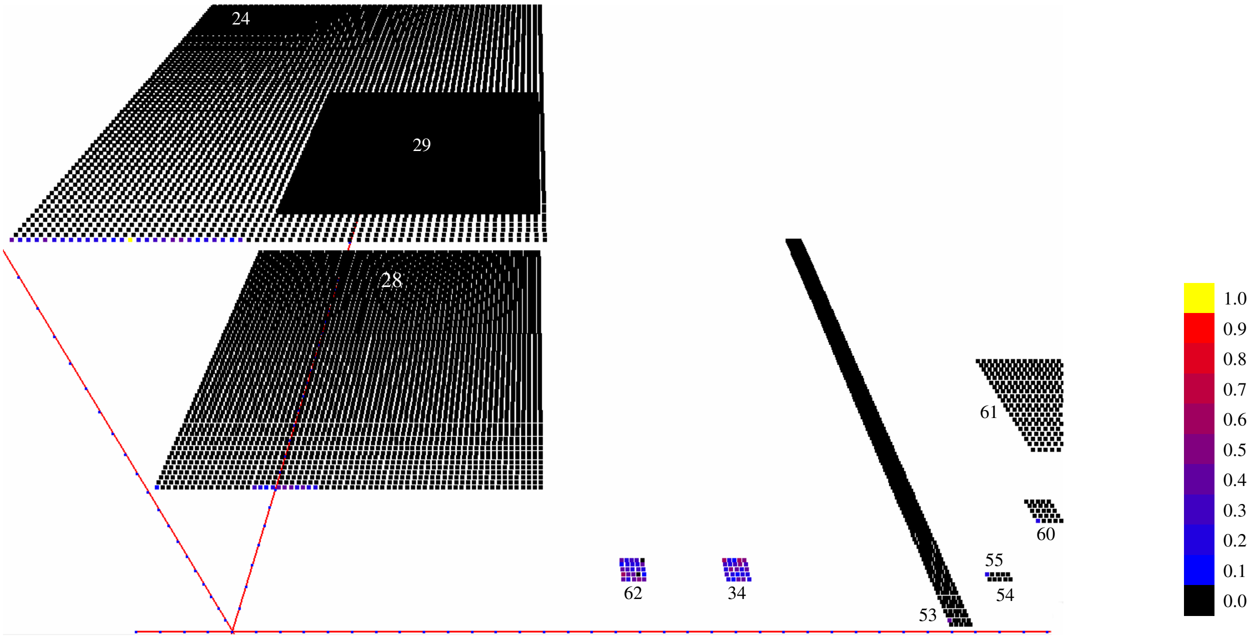


Figure 7.10. Predicted distribution of consciousness during Noise Run 1 according to Tononi's theory

The more specific neuron activity during Analysis Run 1 did not include any of the main complex neurons outside of Emotion and Inhibition, and so the predicted distribution of consciousness in Figure 7.11 only includes neurons from Emotion and Inhibition, with the pattern closely matching the average firing frequencies shown in Figure 7.8. The network would not have been conscious *of* anything during Analysis Run 1 because none of the conscious mental states were representational.[27]

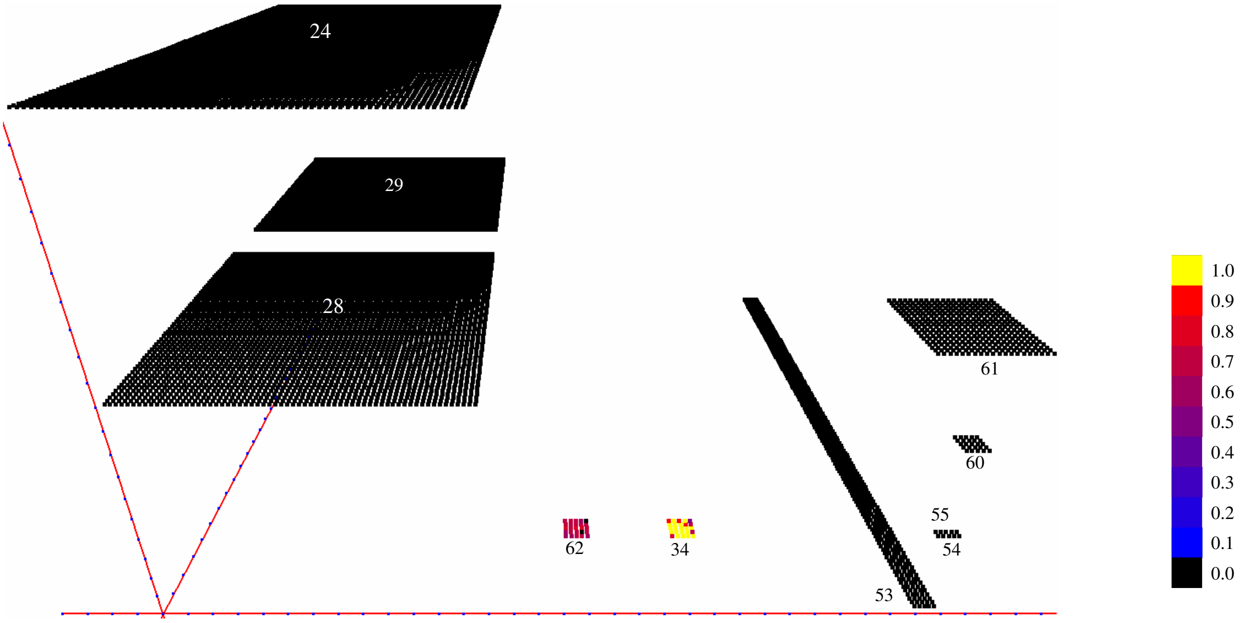


Figure 7.11. Predicted distribution of consciousness during Analysis Run 1 according to Tononi's theory

These results highlight a major problem with a simplistic link between the main complex and consciousness. This network has a number of overlapping complexes with approximately the same value of Φ and it seems somewhat arbitrary to interpret just one of these as the main complex, when it is also conceivable that several overlapping complexes could be part of the same consciousness. In such a consciousness, there would be strong integration between the neurons in Inhibition and Vision Input, but low integration between the different neurons in Vision Input. This appears to reflect our own phenomenology since we seem to be most conscious of our intentional relationship with the world and much less conscious of the relationships that different parts of the world have to each other. One way in which overlapping complexes could be combined would be to look at the rate of change of Φ between adjacent

[27] Tononi's (2004) suggestion that the qualitative character of mental states is determined by their informational relationships might lead to different predictions about what the network was conscious of during Analysis Run 1.

overlapping complexes: a high rate of change of Φ could be used indicate a boundary between the conscious and unconscious parts of the system.

7.9.5 Predictions about the Consciousness of the Network According to Aleksander's Theory

Aleksander's emphasis on depiction led to a prediction about phenomenal states that was based on the minimum amount of mutual information shared with both sensory input and proprioception/ motor output. In this network only Red Sensorimotor and Blue Sensorimotor share mutual information with both Vision Input and Motor Integration, and so these were the only layers that were capable of consciousness according to Aleksander's theory. Whilst there are homogenous connections between Vision Input and Red/ Blue Sensorimotor, the connections between Motor Integration and Red/ Blue Sensorimotor reflect the learnt associations between motor output and visual input, which are stronger whenever motor output consistently resulted in red or blue visual input. This variation in connection strength affects the mutual information between Motor Integration and Red/ Blue Sensorimotor, producing a pattern in the predicted distribution of consciousness for Noise Run 1, which is shown in Figure 7.12.

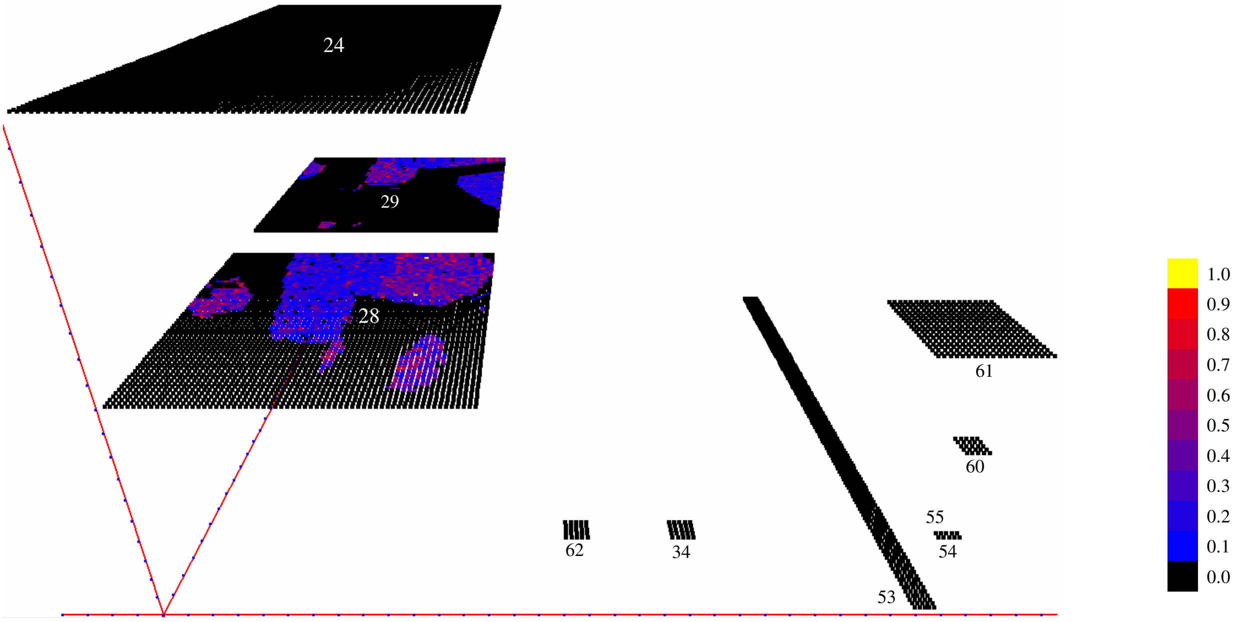


Figure 7.12. Predicted distribution of consciousness during Noise Run 1 according to Aleksander's theory

The predicted distribution of consciousness for Analysis Run 1 reflects the fact that visual activity was concentrated in the top left corner of the red visual field with the occasional ‘imagined’ blue image (see Figure 7.13). According to Aleksander’s definition of depiction, the red and blue data that is represented by these conscious mental states would have been experienced by the system as part of an out there world.

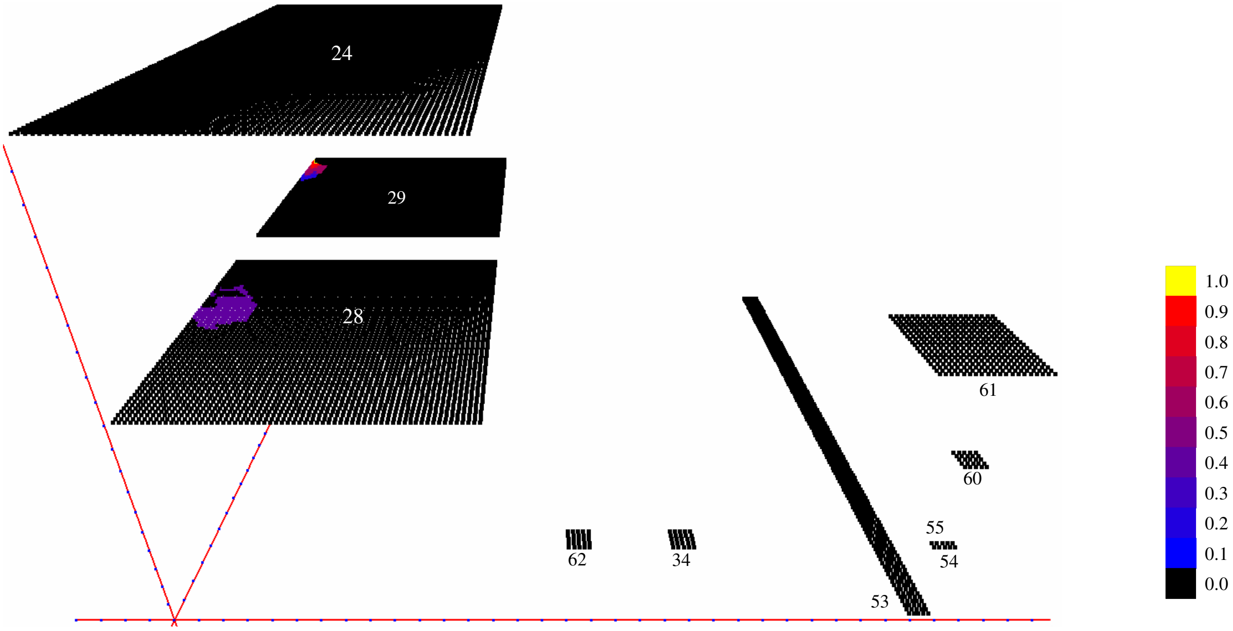


Figure 7.13. Predicted distribution of consciousness during Analysis Run 1 according to Aleksander’s theory

7.9.6 Predictions about the Consciousness of the Network According to Metzinger’s Theory

Predictions about consciousness based on Metzinger’s theory used a combination of spatial and temporal integration, with the former measured using Φ and the latter marked by the presence of a recurrent neuron group in the highest Φ complex. It turned out that almost all of the neurons’ highest Φ complexes included one of the three recurrent layers (Motor Cortex, Emotion and Inhibition), and so almost all of the network was predicted to be minimally conscious according to Metzinger. This is shown in the predicted distribution of consciousness for Noise Run 1 (Figure 7.14) and Analysis Run 1 (Figure 7.15), which closely match the distribution of firing frequencies depicted in Figure 7.9 and Figure 7.8.

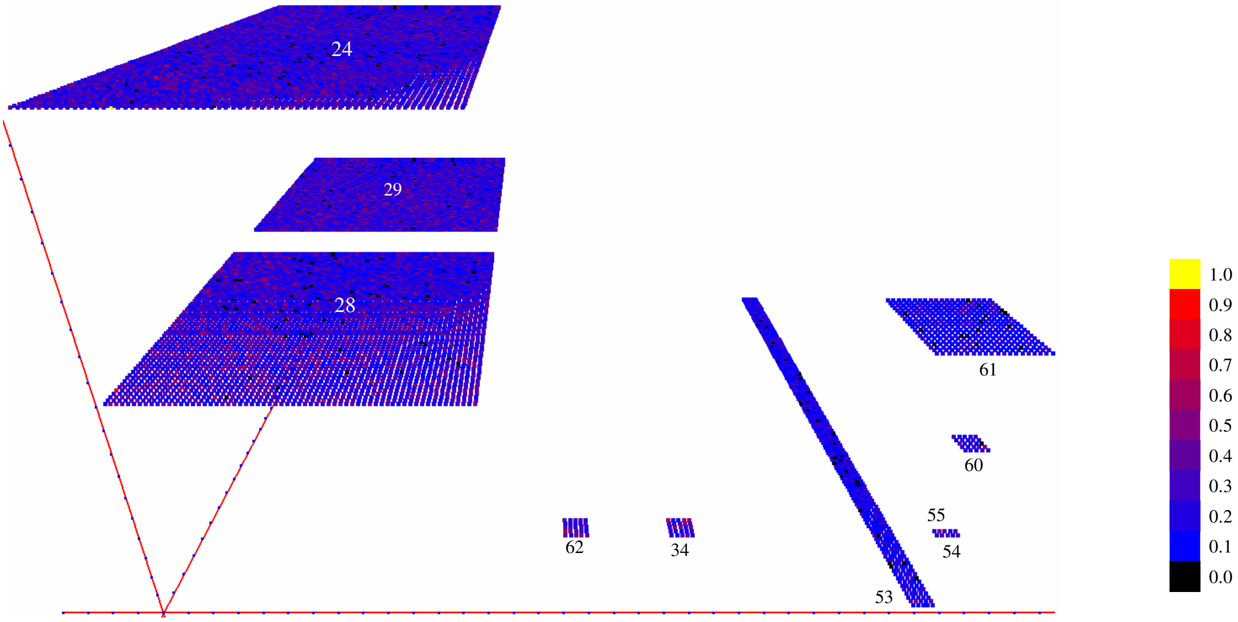


Figure 7.14. Predicted distribution of consciousness during Noise Run 1 according to Metzinger's theory

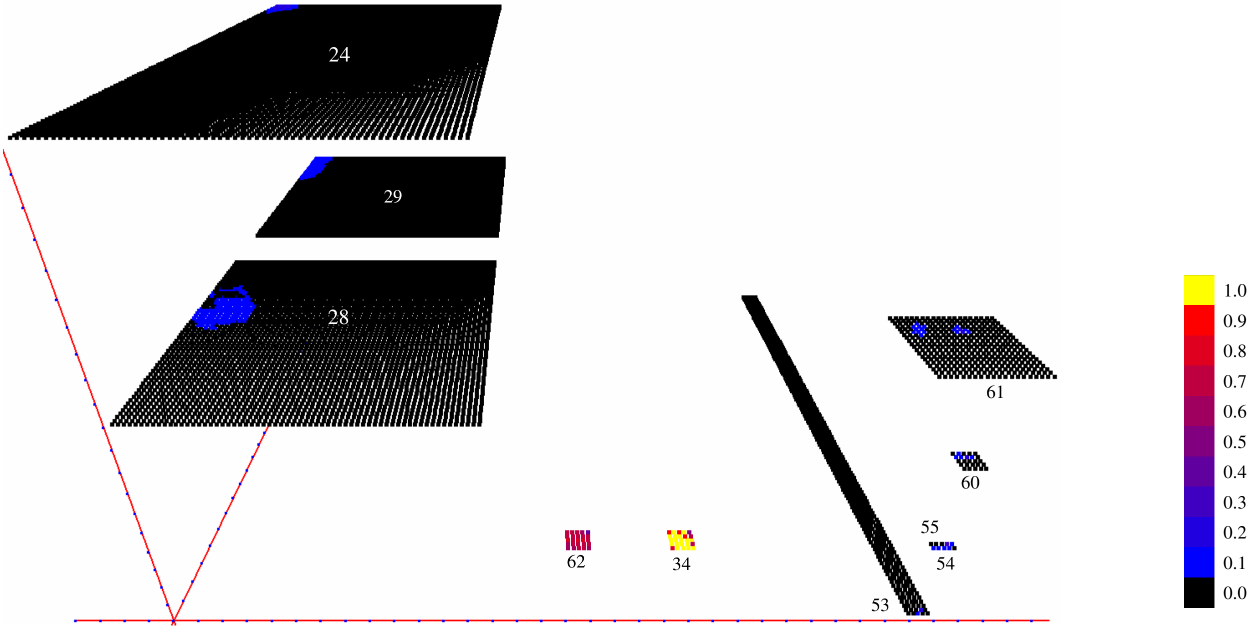


Figure 7.15. Predicted distribution of consciousness during Analysis Run 1 according to Metzinger's theory

During Analysis Run 1 the network would have been conscious of all the active visual and proprioception/ motor output information. This prediction of uniform potential for consciousness throughout the network is likely to change if more of Metzinger's constraints were taken into account. For example, if the mental states associated with consciousness had to be capable of offline activation (constraint 8), then the neurons in Vision Input and Motor Output would no longer be predicted to be associated with consciousness.

7.9.7 Predictions about Conscious and Unconscious Action

This section looks at how the predictions made about the consciousness of the network stand in relation to the discussion of consciousness and action in Section 2.7. As discussed in Section 5.7, the absence of a reactive layer in the network makes it incapable of conscious will, and this discussion focuses on whether it is capable of discrete conscious control according to the different theories of consciousness.

Tononi

The main complex includes only a small number of neurons from Vision Input, Blue Sensorimotor, Motor Integration, Eye Pan, Eye Tilt and Motor Output, and all of these were predicted to be unconscious during Analysis Run 1. However, under very specific conditions it is possible that these sensory and motor parts of the main complex could become active and 'imagine' an action prior to carrying it out, but this is unlikely to happen during normal operation, and most of the time it will be the unconscious parts of the network that decide an action, initiate it and unconsciously carry it out.

Aleksander

Aleksander's theory predicts that there will not be any conscious activity in Vision Input, Motor Integration, Eye Pan, Eye Tilt, Motor Output, Inhibition or Emotion during Analysis Run 1. Whilst the network might be experiencing red and blue in an out there world, the conscious parts do not have any way of differentiating between real and imagined visual input, and so the system cannot tell whether it is deciding to perform an action or actually carrying it out. If the network cannot consciously differentiate between planning and execution, then it cannot be said to be making a conscious decision - it may be conscious of parts of the planning process, but it is not conscious *that* it is planning and it is unable to remember whether it is planning or executing an action. So this interpretation of Aleksander's theory predicts that the network unconsciously chooses an action, unconsciously initiates it and then consciously carries it out.

Metzinger

According to this interpretation of Metzinger's theory, the network is conscious of its planned motor actions and their 'imagined' sensory consequences, and when an action is chosen and initiated, the system becomes conscious of the actual sensory consequences. This suggests that the network is capable of discrete conscious control, in which it consciously plans actions that are initiated immediately and consciously carried out.

7.9.8 Extensions and Enhancements to the Predicted Consciousness of the Network

These predictions about the consciousness of the network suggest a number of ways in which it could be extended or enhanced.

Tononi

Before any thought can be given to extending the consciousness that was predicted to be associated with the network according to Tononi's theory, it is essential to get a more plausible picture of its consciousness by improving the way that consciousness is analyzed to take account of overlapping complexes in a more flexible way (see Section 7.9.4). Once this has been done, it might be possible to design a network in which the main complex has enough representational mental states for conscious decision making. The network's consciousness could also be increased by evolving connection patterns that give the main complex a higher value of Φ.

Aleksander

The mutual information between Vision Input and Red/ Blue Sensorimotor cannot be increased because it is already at its theoretical maximum, but it might be possible to increase the mutual information between Motor Integration and Red/ Blue Sensorimotor by fine tuning the training. The main direction of improvement for this network would be to extend the range of consciousness by making more parts depictive. For example, Emotion and Inhibition could become depictive if they were connected to proprioceptive data and internal sensory data from virtual organs in SIMNOS's body. Red/ Blue Sensorimotor could then change the state of the virtual organs, and when the system sensed this change it would become conscious of the difference between 'positive' and 'negative' body states.

However, consciousness of 'positive' and 'negative' states would not be enough for the network to differentiate between imagination and online perception – it would be conscious of seeing red and feeling good or conscious of seeing blue and feeling bad, but it would not know if it was imagining or perceiving the red or blue stimuli.[28] One solution to this problem would be to use a remembered context or image intensity to indicate whether the network is imagining or not, and in Aleksander's kernel architecture (see Section 3.5.1), the memory module in the awareness area could perform this function by remembering which state is the real world.

Metzinger

The entire network was predicted to be minimally conscious according to Metzinger's theory, and so it would not be possible to extend this predicted consciousness. The qualitative characteristics of the consciousness in the network could be greatly improved by extending the

[28] This problem is closely related to Metzinger's discussion of the world zero hypothesis: "one of both world-models has to be defined as the *actual* one for the system. One of both simulations has to be represented as the *real* world, in a way that is functionally nontranscendable for the system itself. One of both models has to become indexed as the *reference model,* by being internally defined as real, that is, as *given* and not as constructed." (Metzinger 2003, p. 61).

visual processing, adding other senses, such as touch and audition, and increasing the complexity of the actions.

7.9.9 Phenomenal Predictions: Discussion and Future Work

The XML format that was used in these experiments is intended to be a simple example that illustrates the main ideas and a great deal more work is needed to turn this starting point into a usable method. As this approach develops there are likely to be a large number of changes and ambiguities, and although this might initially appear to be a weaknesses of the method, it is actually a strength because it indicates that synthetic phenomenology has the potential to become a paradigmatic science that can move forward by asking questions and resolving ambiguities. At the moment synthetic phenomenology is so unclear that even its lack of clarity is unclear to it, and this XML-based approach will enable synthetic phenomenology to ask and answer precise questions and move forward in a sustainable manner. As has been shown, different theories generate different predictions about the phenomenal states of a system and as brain scanning improves and robots become able to report their conscious states, we will be able to test these predictions and eliminate inaccurate theories.

This analysis presented the final results as the normalized average distribution of consciousness in the network during Noise Run 1 and Analysis Run 1. Whilst this did provide useful predictions about the consciousness of the network and suggestions for enhancing it, it did not address the question about how *much* consciousness was present. Ideally, this analysis would have stated that this network exhibited 5% of the consciousness of the average waking human brain, for example, but without *calibration* of the measurement scales it is impossible to say how much consciousness was associated with the system. Although Tononi (2004) claims that Φ is an absolute measure of the amount of consciousness, he has made no attempt, as far as I am aware, to calculate or measure the Φ of the main complex in an average waking human brain, and without this reference point, the Φ values quoted in this analysis are without absolute meaning. The values of mutual information that were used to measure depiction are equally problematic because we have no idea about how much mutual information is needed to make a mental state depictive.

In order to address this problem urgent work is needed to measure or estimate the Φ and mutual information of a waking human brain, in order to have some way of comparing the measurements of other systems with a system that can (at least to begin with) be taken as a reference standard of consciousness. Without such a 'platinum bar', it is impossible to measure the amount of consciousness in a system using numerical methods. A first step towards obtaining these figures would be to measure Φ and mutual information on more realistic simulations, such as the networks created by the Blue Brain project (Markram 2006). This would give some idea about the Φ and mutual information values that might be found in a real biological system and help us to understand what level of consciousness might be associated with the Φ value of 103 that was found in this network. Better ways of quantifying the amount of consciousness in the system will also go some way towards addressing the "small networks" argument put forward by Herzog et al. (2007), which suggests that many influential theories of consciousness can be implemented by very small networks of less than ten neurons, which we would unwilling to attribute much consciousness to.

In the future it might make sense to multiply the predicted levels of consciousness by the OMC rating to compensate for the type I differences between each system and the human brain. However, in this analysis it would have been pointless to multiply the uncalibrated predictions by a constant factor that would not have appeared in the relative distributions plotted in figures 7.10 - 7.15. Once calibration has been done on Φ and on the use of mutual information to measure depiction, it will be possible to use the OMC scale to compensate for the differences between the system and the human brain, and say how much consciousness the network experienced during Analysis Run 1.

The sequence of XML files is a reasonably accurate description of the predicted phenomenology of the network that makes minimal assumptions about the nature of the phenomenal states. However, large XML files are almost impossible to read and digest and it is difficult to understand how the predicted consciousness of the system changes over time A logical extension of this work would be to investigate ways of presenting the content of these XML files in a more intuitive manner. If the system was experiencing a red spot in the left hand corner of its visual field, then it would be much easier to use virtual reality, for example, to show this to a human observer, instead of asking him or her to read an XML description. Such a 'debugger' for conscious states would also have applications in neurophenomenology.

Another direction of future work would be to move towards a common XML standard for neuro- and synthetic phenomenology that would facilitate collaboration between people working on machine consciousness and people from neuroscience and experimental psychology. This would enable phenomenal prediction methods that were developed in the biological sciences to be tested on artificial systems, and the methodology developed for synthetic phenomenology could be applied to fMRI data and used to make predictions about the consciousness of live human subjects.

Finally, in future work it would be worth making predictions about the consciousness of the network using other theories. For example, it would be particularly interesting to use some of the neural correlates of consciousness, such as neural synchronization (Crick 1994).

7.10 Conclusions

This chapter has demonstrated how the approach to synthetic phenomenology developed in Chapter 4 can be used to make predictions about the consciousness of an artificial neural network. This analysis led to a number of suggestions about how the network's consciousness could be extended and enhanced and it showed how different theories of consciousness make different predictions about the relationship between consciousness and action. This work is at an extremely early stage and a great deal of research is needed to improve the accuracy of our predictions about phenomenal states. It is hoped that this will eventually lead to a more systematic science of consciousness that includes both natural and artificial systems within a single conceptual and experimental framework

8. Conclusions

> I shall certainly admit a system as empirical or scientific only if it is capable of being tested by experience. These considerations suggest that not the verifiability but the falsifiability of a system is to be taken as a criterion of demarcation. In other words: I shall not require of a scientific system that it shall be capable of being singled out, once and for all, in a positive sense; but I shall require that its logical form shall be such that it can be singled out, by means of empirical tests, in a negative sense: it must be possible for an empirical system to be refuted by experience.
>
> Popper (2002, p.18)

8.1 Achievements

One of the key achievements of this thesis was the development and demonstration of a synthetic phenomenology framework that provides a way of predicting and describing the conscious states of artificial systems using different theories of consciousness. This methodology works entirely from a third person perspective and it does not rely on implicit assumptions about biological neurons being necessary for consciousness. Systematic falsifiable predictions about artificial conscious states could help machine consciousness to become more scientific, and this methodology may also contribute to the science of consciousness more generally since it enables predictions to be made about the consciousness of biological systems. The work on synthetic phenomenology also offered a number of significant innovations:

- An OMC scale that models our intuitions about the consciousness of artificial systems.
- A clear definition of mental states and representational mental states.
- A method for the identification of representational mental states that uses noise injection and mutual information.

- New approximation methods for measuring the information integration of systems with more than a few dozen elements.
- The use of a markup language to describe artificial phenomenal states.
- Detailed predictions about the distribution of consciousness in a neural network according to Tononi's, Metzinger's and Aleksander's theories.

A second achievement of this project was the development of a neural network that used some of the cognitive characteristics associated with consciousness to control the eye movements of the SIMNOS virtual robot. This network is a novel contribution to the field and differs from the networks developed by Aleksander (2005), Shanahan (2006, 2008), Dehaene et al. (1998, 2003, 2005) and Krichmar et al. (2005). This network exhibited a limited form of conscious behaviour (MC1), had cognitive characteristics associated with consciousness (MC2) and was predicted to be phenomenally conscious (MC4) according to three theories of consciousness, and so this thesis can lay reasonable claim to have created an extremely limited form of consciousness for SIMNOS, and thus to have fulfilled one of the key aims of the CRONOS project.

A further significant achievement of this project was the development of the SpikeStream neural simulator. This has good performance and its simulation features and graphical interface were a substantial advance over Delorme and Thorpes's (2003) implementation of the SpikeNET architecture. The source code of SpikeStream is fully documented and SpikeStream has been released both as source code and pre-installed on a VMWare virtual machine running SUSE Linux. The close integration between SpikeStream and SIMNOS makes them an extremely powerful toolset for carrying out research into all aspects of perception, muscle control, machine consciousness and spiking neural networks.

Finally, this thesis makes a number of theoretical contributions to the study of natural and artificial consciousness, which include the discussion of the relationship between the phenomenal and physical, the distinction between type I and type II potential correlates of consciousness, and the analysis of conscious will and conscious control. The distinction between the different MC1-4 areas of machine consciousness was also original and the review of work in machine consciousness, published as Gamez (2007a), received a positive response from other people working in the field.

8.2 General Discussion and Future Work

This thesis has emphasized the importance of scientific experimentation in machine consciousness research. Whilst theoretical discussion is needed to establish a framework within which empirical work can take place, machine consciousness will only become fully scientific when it can make falsifiable predictions about the consciousness of artificial systems.[1] Key requirements for this are more formal definitions of each theory that can be used to make predictions about the consciousness associated with different systems. These definitions can be mathematical equations, algorithms or pieces of code – their only requirement is that they take the states of an arbitrary system as input and generate predictions about its phenomenal states. The work of Tononi (2004) is a good example of how a theory of consciousness can be formalized in this way, and the definitions offered in Section 7.6.2 and Section 7.7.3 were a first attempt at a formalization of Aleksander's and Metzinger's theories.

To compare predicted distributions of consciousness with first 'person' reports, more work needs to be done on how artificial systems can be given the ability to speak about their conscious states – perhaps using the work of Steels (2001, 2003). More theoretical work is also needed to understand how the reporting of conscious states fits into the framework of conscious control and how this works at a phenomenal and physical level. Formalized theories of consciousness could also be used to make predictions about the consciousness of biological

[1] This view is shared by Crick and Koch (2000) – see the quotation in Section 2.6.1.

systems that can report their conscious states, which could be tested through collaborations with people working in experimental psychology and neuroscience. The current lack of low level access to biological systems' states means that this work is not likely to progress very fast until scanning technologies experience breakthroughs in their temporal and spatial resolution.

Many parts of the approach to synthetic phenomenology in this thesis are based on numerical methods that need to be tested and calibrated on real data. To begin with, the OMC scale could be tested by using psychophysical methods to establish how accurately it models our subjective assessment about the link between type I PCCs and consciousness. Second, we need to measure how much mutual information is necessary for a state to become representational in a real biological system, and the link between mutual information and depiction needs to be validated and calibrated by estimating the amount of depiction in humans. Finally, the information integration of real biological systems needs to be measured to establish a connection between information integration and consciousness. This process faces many problems, such as the size of real biological neural networks, the fact that noise injection cannot be practiced on humans and the low spatial and/ or temporal resolution of scanning data.

The neural network developed by this project was very basic and could be improved in many ways. One direction of improvement would be to use SIMNOS's visual pre-processing to add layers sensitive to movement, edges and other data, which could work in a similar way to the visual input layers in the network developed by Krichmar et al. (2005). A reactive layer could also be included to improve the performance of the network and to make it capable of conscious will. In this thesis the lack of a viable software interface for CRONOS and delays in the production of the final robot meant that it was not possible to test the network on a real system, and this is something that could be attempted in future work. The learning of the network could also be improved and more research needs to be done on how learning can be implemented on different time scales.

In the future a well documented biologically inspired test network could be developed that would enable people to validate their predictions about consciousness on a commonly agreed standard and compare different methods of measuring functional and effective connectivity. Although a common project or series of meetings might be needed to design such a network, the previous work on machine consciousness (Chapter 3) and on the simulation of biologically inspired neural networks (Section 5.6) suggests that enough work has been done to design an initial test system.

The interpretation of consciousness put forward in Chapter 2 will not be popular with people who believe that some kind of reduction of the phenomenal to the physical is the only way in which a science of consciousness can proceed. However, if a non-reductive interpretation is correct, then it could provide a more secure framework for a science of consciousness, and in the future more work needs to be done to clarify this approach and work through 'use cases' that examine the relationship between the phenomenal and the physical in as much detail as possible. One major problem is how independent causal chains within the phenomenal and the physical should be understood, and it may need some reworking of the concept of causation to deal with the crossover that occurs when a conscious decision leads to changes in the physical world.[2]

The main focus of this thesis was on the development of a systematic framework for analyzing systems for conscious states. Since current theories could be used to illustrate this approach, it was not necessary to develop a new type II theory of consciousness in this thesis, and little attempt was made to criticize or improve existing theories. As robots and scanning technologies improve, we will be able to make more accurate comparisons between predictions about consciousness and reports of conscious states, which should enable us to develop better type II theories of consciousness.

[2] Hume's (1983) interpretation of causation as a constant conjunction between cause and effect might be applicable here.

APPENDIX 1
SPIKESTREAM MANUAL

Note on the Text

This appendix is the manual that was included with the 0.1 distribution of SpikeStream and it has been included in this thesis to give a better idea about the functionality of the SpikeStream simulator, which was developed as part of this PhD. The documentation in this manual is complementary to the comprehensive source code documentation, which is also part of the SpikeStream distribution and is included in the Supporting Materials. The text of this manual is largely the same as the version that was included in the SpikeStream 0.1 release, with a few minor improvements. The numbers and formatting have been changed to match the rest of the thesis.

A1.1 Introduction

SpikeStream is a simulator that has been tested on medium sized networks of up to 100,000 spiking neurons. It works in a modular distributed manner and can run in parallel across an arbitrary number of machines. SpikeStream exchanges spikes with external devices over a network and it comes ready to work with the SIMNOS virtual humanoid robot (see Section A1.9.4). More information about the architecture of SpikeStream can be found in Gamez (2007b). This manual covers the installation of SpikeStream and use of its key features.

I have tried to make the installation of SpikeStream on Linux as painless as possible using four scripts that set the necessary variables, build SpikeStream, install SpikeStream and create the databases. However, these depend on third party software and a database, and so a certain amount of work is required to get the whole system running. For other operating systems a virtual machine distribution has been prepared, which is covered in Section A1.2.8.

SpikeStream is a complex piece of software with many useful features and it is stable enough to run experiments. However, it is still at an early stage of development and subject to a number of bugs and limitations. Occasionally it will crash, but most of the time no data will be lost because all changes are immediately stored in the database and restarting most often solves the problem. If you let me know about any undocumented bugs and limitations, I will do my best to solve them and any offers of help with SpikeStream are extremely welcome. If there is enough interest, I will turn it into a collaborative open source project.

This manual is targeted at the user of SpikeStream who wants to use the simulation functions and may want to extend the Neuron or Synapse classes to create their own neuron and synapse models. I have tried to make the information in this manual as accurate as possible and apologize for any errors and omissions. Documentation of the source code is available in the doc folder of the distribution and at http://spikestream.sourceforge.net.

Feel free to get in touch if you have any problems building and running SpikeStream. You can reach me at david@davidgamez.eu or on +44 (0) 7790 803 368. I have also set up a mailing list for SpikeStream at spikestream-user@lists.sourceforge.net.

A1.2 Installation

A1.2.1 Overview

Before installing SpikeStream it is recommended that you read the paper covering its architecture and operation (Gamez, 2007b). Sections A1.2.2 - A1.2.7 give full instructions for installing SpikeStream on Linux and other UNIX-based systems. If you just want to try SpikeStream out or use it on a different operating system, it is available pre-installed on a SUSE 10.2 virtual machine, which can be run using the VMware Player (see Section A1.2.8).

A1.2.2 System Requirements for Linux Installation

Operating System

SpikeStream has been written and tested on SUSE 10.0 and SUSE 10.2. SpikeStream Simulation and SpikeStream Archiver have also been tested on Debian 3.1. A few adjustments may be required to get it working on other Linux and UNIX operating systems. It should be possible to get SpikeStream running on Cygwin under Windows, but I have not attempted this yet.

Hardware

SpikeStream can run on a single machine or across a cluster. On the main workstation, hardware graphics acceleration will speed up the visualization of large networks. A megabit network is useful if you want to run SpikeStream across several machines.

A1.2.3 Dependencies

SpikeStream depends on a number of other libraries, which must be installed first. Some of these are only needed on the main workstation to compile and run SpikeStream Application. Others are needed by all modules.

Google Sparse Hash

Fast and efficient dense and sparse hash maps developed by Google. Available at http://goog-sparsehash.sourceforge.net/.

Install on all machines.

MySQL Database and Development Libraries

May form part of your Linux distribution. Otherwise available at www.mysql.org. You need the development parts of MySQL as well as the server.

The development libraries need to be installed on all machines. The server only needs to be *installed on the machine(s) that are hosting the databases.*

MySQL++

C++ wrapper for MySQL. Available at: http://tangentsoft.net/mysql++/ .

Install on all machines.

Qt

Provides a graphical user interface and many useful functions. Likely to come with your distribution of Linux.

IMPORTANT NOTE: *SpikeStream only compiles and runs using Qt version 3.*.**. It will not compile using Qt 4.*.*. If 4.*.* is your default version of Qt, you need to install Qt 3.*.* in a separate location to compile SpikeStream. In SUSE 10.2 the default Qt version is 4, but Qt 3 is also installed and you can make Qt 3 the default by adding the Qt 3 directory to the start of your path in your .bashrc file using: `export PATH=$QTDIR/bin:$PATH.` You can also directly invoke this version of qmake on the command line by using **`$QTDIR/bin/qmake`** instead of **`qmake`** when you generate the makefiles.

Qt is only needed on the main workstation.

PVM (Parallel Virtual Machine)

Used for distributed message passing and spawning of remote processes. Included with some Linux distributions, otherwise install manually. Available at: http://www.netlib.org/pvm3/index.html.

IMPORTANT NOTES:

1. The build of PVM may break with recent versions of gcc. If it breaks with the error:

```
... src/global.h: 321: error: array type has incomplete element type
... src/global.h: 323: error: array type has incomplete element type
```

   Replacing PVM_ROOT/src/global.h with global.h from the 'extras' folder of the

SpikeStream distribution should fix the problem.

2. It can be useful to give user accounts permission to write to $PVM_ROOT/bin/LINUX. This makes it easier when you have to manually install spikestreamarchiver and spikestreamsimulation, which have to be installed in this directory to be launched by pvm.
3. On SUSE 10.2 (and perhaps elsewhere) you may get an error along the lines of : ”netoutput() sendto: Invalid argument” when adding a second host in pvm. This can be fixed by adding an entry to your hosts file along the lines of:

   ```
   [machine ip address] [machine name]
   ```

   for example:

   ```
   192.168.1.22    desktopmachine
   ```

If you install pvm yourself, don't forget to create a link to PVM_ROOT/lib/pvm from your bin folder so that it can be run from anywhere. You may also want to install xpvm, which can be very helpful for debugging processes and messages when things go wrong. Getting pvm to run successfully across several machines can be tricky and is beyond the scope of this manual.

Install on all machines.

Qwt

Graph drawing libraries available at: http://qwt.sourceforge.net/.

Only needed on the main workstation.

A1.2.4 Build and Installation Using Scripts

This section covers the installation of SpikeStream using scripts that set the variables, build the modules and install the libraries. These are the quickest and easiest way to install SpikeStream on Linux. If anything goes wrong with these scripts, Section A1.2.5 covers manual installation of the individual modules.

Unpack Distribution

When you have downloaded SpikeStream, you need to unpack it using the command:

```
tar -xzvf spikestream-0.1.tar.gz
```

This will extract it to a directory called spikestream-0.1. This will be the root directory for building and running the application, so move this directory to its final location before moving on to the next step.

Set SPIKESTREAM_ROOT

SpikeStream depends on a shell variable called SPIKESTREAM_ROOT, which is *essential* for building and running the application. This variable should be set to the root of the spikestream-0.1 directory. The best place to set this is in your .bashrc file by adding, for example:

```
export SPIKESTREAM_ROOT=/home/davidg/spikestream-0.1
```

This needs to be done on all machines that you build and run SpikeStream on and you need to make sure that the remote shell invoked by pvm (which may be different from your default bash shell) also has SPIKESTREAM_ROOT set correctly.

Set Build Variables

To keep everything as simple as possible, the locations of the libraries needed for building SpikeStream are set by the SetSpikeStreamVariables script, which can be found in the scripts folder of the distribution. Open this script up and check that the library and include locations match those on your system:

```
#Location of MySQL
export MYSQL_INCLUDE=/usr/include/mysql
# Location of MySQL++
export MYSQLPP_INCLUDE=/usr/local/include/mysql++
# Location of Qwt files. Not needed for simulation builds
export QWT_ROOT=/usr/local/qwt
```

```
# Location of Google hash map include files.
export GOOGLE_INCLUDE=/usr/local/include/google
```

When you are installing SpikeStream across several machines, the Qt and Qwt libraries are only needed on the machine running SpikeStream Application. In this case, run the script with the option “-s”.

SpikeStream cannot be built unless these variables have been set correctly for the type of build. When you have checked the locations, save the script and try running it from the scripts folder using:

`./SetSpikeStreamVariables` (Main workstation)

`./SetSpikeStreamVariables -s` (Other machines used in the simulation)

If it exits without errors, you can move on to the next stage of the installation. If you get errors setting the variables, make sure that all of the required libraries are in the places set by the script and SPIKESTREAM_ROOT and PVM_ROOT are set correctly.

Run Build Script

SpikeStream comes with a build script that compiles all of the modules and copies the ones that are installed in the SPIKESTREAM_ROOT directory to their correct locations. This is not guaranteed to work on every occasion, but it can speed up the installation process considerably. If you have problems running this script it is worth taking a look inside it for the list of commands that are needed to build and install the parts of the application. To run this script, change to the scripts folder and type:

`./BuildSpikeStream` (Main workstation)

`./BuildSpikeStream -s` (Other machines used in the simulation)

If all goes well you should end up with the following output on the main workstation:

```
---------------------------------------------------------------------
---------------            Build Results             ----------------
---------------------------------------------------------------------
SpikeStreamApplication: Built ok.
SpikeStreamSimulation: Built and installed ok.
SpikeStreamArchiver: Built and installed ok.
STDP1 Neuron: Built ok.
STDP1 Synapse Built ok.
SpikeStream built successfully.
```

If one of the libraries or applications does not build, you will have to track down the error by looking at the configure and make output and either re-run the build script or install the missing component(s) individually. Instructions for installing each of the components individually are given in Section A1.2.5.

Install SpikeStream

This script installs spikestreamsimulation and spikestreamarchiver in the $PVM_ROOT/bin/LINUX directory, which often requires root privileges. Some neuron and synapse libraries also need to be installed as root to enable dynamic linking and the install script creates symbolic links between one of the default library locations on your system and the neuron and synapse libraries in $SPIKESTREAM_ROOT/lib. The use of symbolic links is suggested because it is anticipated that you will be recompiling the neuron and synapse libraries to implement your own learning algorithms and the use of symbolic links saves you the trouble of installing them as root each time you do this. If you are planning to use only the supplied neuron and synapse classes, then copies of these can be placed in the specified locations. More information about this can be found in Section A1.12.3.

IMPORTANT NOTE: You should only install links to these libraries as root if you are the sole user of SpikeStream on the system. Otherwise you may end up dynamically loading another user's libraries!

To run the install script, get a root shell, make sure that SPIKESTREAM_ROOT is defined in the root shell ("`echo $SPIKESTREAM_ROOT`" should return the correct location) and run:

```
$SPIKESTREAM_ROOT/scripts/InstallSpikeStream
```

If everything has worked up to this point you can move on to set up the databases, as described in Section A1.3. If the build has broken for some reason, take a look at some of the common build and installation problems covered in Section A1.2.7. Instructions for manually building each component are given in the next section.

A1.2.5 Manual Installation Procedure

Once your have unpacked the distribution and set the SPIKESTREAM_ROOT variable (Section A1.2.4), you are ready to manually build and install the SpikeStream components. You should only install SpikeStream this way if you have run into problems with the build and installation scripts.

SpikeStream Library

This contains classes that are common to many parts of the system and should be compiled first.

- Check the locations in the SetSpikeStreamVariables script and run it using "**`. ./SetSpikeStreamVariables`**" (don't miss the *second* dot before the slash!).
- Change to directory $(SPIKESTREAM_ROOT)/spikestreamlibrary/
- Run the command: **`./configure -libdir=$SPIKESTREAM_ROOT/lib`**

- Type **`make`**
- If everything goes ok, type **`make install`**. There should be a file called libspikestream.a in the $SPIKESTREAM_ROOT/lib directory.

SpikeStream Application

This is the graphical application for editing neuron groups and launching simulations and it only needs to be built on the main workstation. It is a Qt project, so installation is a little different from the other parts of the system.

- Check your version of Qt is correct by typing **`qmake --version`**. The output should contain the version of Qt that qmake is using, for example Qt 3.3.7. If your version is greater than 3.*.*, you need to install Qt 3 on your system and make sure that qmake uses this version of Qt. See Section A1.2.3 for more on this.
- Check the locations and debug flags in the SetSpikeStreamVariables script and run it using: **`. ./SetSpikeStreamVariables`** (don't miss the *second* dot before the slash!).
- Change to the $SPIKESTREAM_ROOT/spikestreamapplication directory and use qmake to create the makefiles: **`qmake spikestreamapplication.pro`**
- Type **`make`**
- If everything goes ok, there should be a program called spikestreamapplication in the $SPIKESTREAM_ROOT/spikestreamapplication/bin directory.
- If you want, create a symbolic link to $SPIKESTREAM_ROOT/bin or your local bin directory using: **`ln -s $SPIKESTREAM_ROOT/spikestreamapplication/bin/spikestreamapplication $SPIKESTREAM_ROOT/bin/spikestream`**.

You can try to run spikestreamapplication, but it will not work properly until the database has been configured – see Section A1.3.

SpikeStream Simulation

This is the program that simulates a neuron group. It is launched using pvm, so it has to be installed in the $PVM_ROOT/bin/LINUX directory on every machine that you want to run the simulation on. If you are running SpikeStream across several different Linux versions, this program will have to be recompiled for each architecture.

- Check the locations and debug flags in the SetSpikeStreamVariables script and run it using: **`. ./SetSpikeStreamVariables`** (don't miss the *second* dot before the slash!).
- Change to the $SPIKESTREAM_ROOT/spikestreamsimulation directory.
- Run the command: **`./configure --bindir=$PVM_ROOT/bin/LINUX --libdir=$SPIKESTREAM_ROOT/lib`**
- Type **`make`**
- If all goes well type **`make install`**. You will need to have write permission to the $PVM_ROOT/bin/LINUX directory or change to superuser for this step.
- If everything goes ok, there should be a file called libspikestreamsimulation.a in the $SPIKESTREAM_ROOT/lib directory and an executable file called spikestreamsimulation in the $PVM_ROOT/bin/LINUX directory.

SpikeStream Archiver

This program stores firing patterns in the database. It is launched using pvm, so it has to be in the $PVM_ROOT/bin/LINUX directory of every machine that you want to run a simulation on. If you are running SpikeStream across several different Linux versions, this program will have to be recompiled for each architecture.

- Check the locations and debug flags in the SetSpikeStreamVariables script and run it using **`. ./SetSpikeStreamVariables`** (don't miss the *second* dot before the slash!).

- Change to the $SPIKESTREAM_ROOT/spikestreamarchiver directory.
- Run the command: **`./configure --bindir=$PVM_ROOT/bin/LINUX`**
- Type **`make`**
- If all goes well type **`make install`**. You will need to have write permission to the $PVM_ROOT/bin/LINUX directory or change to superuser for this step.
- If everything goes ok, there should be an executable file called spikestreamarchiver in the $PVM_ROOT/bin/LINUX directory.

Neuron and Synapse Classes

Neuron and Synapse classes are stored as libraries that are dynamically loaded at runtime and the name of each library should be added to NeuronTypes or SynapseTypes in the database. Some neuron and synapse libraries may need to call methods on each other and they need to be placed in the $SPIKESTREAM_ROOT/lib directory to enable cross linking. Copies also need to be placed in /user/local/lib to enable dynamic loading. Section A.1.12 gives detailed information about adding your own neuron and synapse classes to SpikeStream. Installation instructions are given here for STDP1Synapse, which should be followed for each of the neuron and synapse libraries.

- Check the order in which the neuron and synapse classes need to be built. Some neuron and synapse classes depend on each other so the build order may be important. For example, STDP1Synapse must be built before STDP1Neuron.
- The neuron and synapse classes depend on the spikestreamsimulation library, so make sure that this is installed correctly before commencing installation.
- Check the locations and debug flags in the SetSpikeStreamVariables script and run it using: **`. ./SetSpikeStreamVariables`** (don't miss the *second* dot before the slash!).

- Change to the $SPIKESTREAM_ROOT/STDP1Synapse directory.
- Run the command: **`./configure`**
- Type **`make`**
- If all goes well copy the libstdp1synapse.so library to $SPIKESTREAM_ROOT/lib directory.
- Log in as root and change to your system's library location: **`cd /usr/local/lib`**
- Create a link from your system's library location to the neuron library: **`ln -s -f ${SPIKESTREAM_ROOT}/lib/libstdp1synapse.so libstdp1synapse.so.1`**
- Add the information about the neuron class that you have installed to the database – see Section A1.12.4.

A1.2.6 Cleaning Up and Uninstalling SpikeStream

CleanSpikeStream Script

SpikeStream can cleaned up using the CleanSpikeStream script. This removes all of the files in SPIKESTREAM_ROOT that were created by the build script and runs make clean in each of the directories. It also removes the "makefile" files created by qmake in the spikestreamapplication directory. The clean script does not remove spikestreamsimulation, spikestreamarchiver or the symbolic links to libstdp1neuron.so and libstdp1synapse.so that are created by the InstallSpikeStream script. You need to run the uninstall script to delete these components of SpikeStream.

UninstallSpikeStream Script

This script uninstalls spikestreamsimulation, spikestreamarchiver and deletes the symbolic links to the neuron and synapse libraries. Use this when you want to remove all SpikeStream files from the system except for those at SPIKESTREAM_ROOT.

IMPORTANT NOTE: *This script must be run as root.*

A1.2.7 Common Build and Installation Problems

Some common build and installation problems are as follows.

1. When building SpikeStream application you are likely to get the warning "has virtual functions but non-virtual destructor". This is a known issue, which should be ignored. See: http://lists.trolltech.com/qt-interest/2005-10/msg00342.html.

2. You may get some strange Qt errors that break the build, such as:

   ```
   In file included from NetworkDataXmlHandler.h:27,
                    from ArchiveManager.h:28,
                    from ArchiveManager.cpp:24:
   NetworkMonitor.h:33:17: error: qgl.h: No such file or directory
   In file included from ArchiveManager.h:28,
                    from ArchiveManager.cpp:24:
   NetworkDataXmlHandler.h:30:18: error: qxml.h: No such file or
   directory
   In file included from SpikeStreamMainWindow.h:28,
                    from ArchiveManager.cpp:28:
   NetworkViewer.h:33:20: error: qaccel.h: No such file or directory
   In file included from SpikeStreamMainWindow.h:29,
                    from ArchiveManager.cpp:28:
   NetworkViewerProperties.h:38:20: error: qtable.h: No such file or
   directory
   In file included from MonitorArea.h:28,
                    from SimulationWidget.h:29,
                    from SpikeStreamMainWindow.h:31,
                    from ArchiveManager.cpp:28:
   MonitorWindow.h:32:25: error: qdockwindow.h: No such file or
   directory
   In file included from SimulationWidget.h:29,
                    from SpikeStreamMainWindow.h:31,
   ```

```
                 from ArchiveManager.cpp:28:
MonitorArea.h:37:23: error: qdockarea.h: No such file or directory
In file included from SpikeStreamMainWindow.h:34,
                 from ArchiveManager.cpp:28:
LayerWidget.h:32:24: error: qpopupmenu.h: No such file or directory
```

These are almost certainly caused by compiling with the wrong Qt version. Check the Qt version by using "qmake --version"**.** *If the Qt version is 4.*.*, it will not work! You must build SpikeStream Application using Qt 3.*.*.* When you have sorted out the correct version of Qt (see Section A1.2.3) you need to remove the "makefile" files from spikestreamappliction and spikestreamapplication/src before running the build script again. This can be done manually or by invoking the CleanSpikeStream script, which will do it for you. A future version of SpikeStream will compatible with Qt 4.

3. Double check that all the libraries are installed in the places specified in the SetSpikeStreamVariables script. If, during manual installation, you run this script without a dot and space before it, then the variables will not be set.

4. Double check that SPIKESTREAM_ROOT and PVM_ROOT are set correctly for your system. Both are crucial to a successful build. A common problem when running SpikeStream across several machines is that the default shell invoked by pvm is different from the one in which SPIKESTREAM_ROOT and PVM_ROOT are set.

5. The error: "**cp: cannot create regular file `/home/davidg/lib/pvm3/bin/LINUX/spikestreamarchiver': Permission denied**" is caused because you do not have permission to access the directory where pvm is installed. Change to root before running the installation script again or give all users write access to this directory. If you lack superuser access you may need to create a local pvm installation.

6. A *build* problem related to permissions may occur if you copy the spikestream-0.1.tar.gz

file as root and then unpack and build it. This can cause errors building STDP1 Neuron and STDP1 Synapse, which gcc attributes to inadequate permission to access the lib file. To solve this problem, set yourself as the owner of spikesream-0.1.tar.gz and set its group to users before unpacking it.

7. If the SpikeStream Application GUI looks like it was built in the 1970's and does not share the look and feel of other KDE applications on your machine, rebooting may solve the problem. Otherwise check that you are not compiling against an old version of Qt (before 3.*.*).
8. If you have database problems when SpikeStream is launched across several machines, make sure that the database configuration is not set to 'localhost' – put the ip address in spikestream.config instead (see Section A1.4.1).

If you cannot find a solution to your problem, see Section A1.1 for further support.

A1.2.8 Virtual Machine Installation

Overview

SpikeStream is also available pre-installed on a SUSE 10.2 virtual machine. This is a much bulkier distribution (around 4GB when uncompressed) that enables it to run on a variety of operating systems with a minimum of installation difficulties. The disadvantages of this are the size, a slightly reduced running speed and the fact that you have to boot up the virtual machine every time that you want to run SpikeStream (although SpikeStream can be restarted any number of times once the virtual machine has booted up). This manual only covers the basics and the VMware documentation should be consulted for full instructions about installing the VMware Player and running virtual machines.

Virtual Machine Files

The virtual machine files are available on DVD (drop me an email if you would like to receive a copy) or for download at: http://csres82.essex.ac.uk/~daogam/.

Install VMware Player

Download and install the free VMware Player from: http://www.vmware.com. If you want to use SpikeStream with SIMNOS (see Section A1.9.4) you need to configure the networking between the SUSE virtual machine and the host operating system so that you can ping each operating system from the other and access the Devices database on the host operating system from SUSE. This is not necessary if you are not using SIMNOS. Support with installation of VMware Player and its networking can be found in the VMware documentation and forums.

Run Virtual Machine

Once SUSE 10.2 is running in your VMware Player, click the SpikeStream icon on the SUSE desktop to start SpikeStream. Some of the devices, such as the DVD drive at location E: and the floppy drive, may not be available on your system. If you want to correct these problems or change the configuration of the virtual machine, you will have to purchase a copy of VMware Workstation, since the free VMware Player does not allow you to edit the virtual machine.

IMPORTANT NOTE: To reduce the size of the virtual machine distribution, the virtual hard drive has been kept as small as possible. There is only about 500MB free space on the drive, so take care not to over fill it or you may not be able to boot the virtual machine.

A1.3 Databases

A1.3.1 Introduction

SpikeStream depends on a number of databases, which can be distributed across different machines. The parameters for these databases are set in the $SPIKESTREAM_ROOT/spikestream.config file. This file is only used on the main workstation since the database parameters are passed to SpikeStream Simulation and SpikeStream Archiver as command line parameters. The SpikeStream databases are as follows:

- *NeuralNetwork*. Stores neurons, synapses and the connections between them. Different types of neuron and synapse classes are also stored here, along with parameters and the amount of noise injected into each of the neuron groups.
- *NeuralArchive*. Stores patterns of spikes or firing neurons that are recorded by the user during a simulation run.
- *Patterns*. Stores patterns that can be applied by the user to a layer during a simulation run. More information about patterns is given in Section A.1.10.
- *Devices*. Lists the devices that are available for SpikeStream to connect to. Also breaks the device layer down into receptors and groups of receptors known as components. See Section A.1.9 for more about SpikeStream and external devices.

More detailed information about the structure and purpose of these databases can be found in the SQL files in $SPIKESTREAM_ROOT/databases, which are used to create and populate the databases. When running SpikeStream with SIMNOS, SIMNOS sets up and updates the Devices and SIMNOSSpikeReceptors tables in the *Devices* database, and the host, username and password of the *Devices* database needs to be coordinated with SIMNOS. This manual assumes that all four databases will be set up using the same host, username and password.

A1.3.2 Setting up MySQL

Introduction

Before SpikeStream can run, the correct databases need to be created and their user, host and password information entered in the $SPIKESTREAM_ROOT/spikestream.config file. This only needs to be done on the main workstation since the database parameters are passed to SpikeStream Simulation and SpikeStream Archiver as command line parameters. You can go straight on to Section A1.3.4 if you already have a MySQL server and an account set up that you want to use with SpikeStream. Details about setting up and running MySQL can be found in many places and there is extensive MySQL documentation online. Only the basics are given here.

Start MySQL Server

When you have installed MySQL (see Section A1.2.3), test to see if it is running using: **`ps -el | grep mysql`**. This should return a line containing "mysqld" as one of the running processes. If this is not listed, use chkconfig to enable the service. As superuser type: **`chkconfig --list mysql`**, which should tell you if mysql is enabled or not. If it is not enabled for your current run level, type: **`chkconfig mysql on`** and make sure that it is enabled.

Even when mysql is enabled, the daemon may not have started. To start the daemon go to /etc/init.d/ and log in as root. Then run the mysql command by typing: **`./mysql start`**, which should start up the daemon. Check that it has started, then you are ready to set up the accounts.

Set Maximum Number of Connections

Each layer is handled by SpikeStream using a separate pvm process, which may have several connections to the database. If you are going to be using a large number of layers it is a good idea to increase the number of allowed connections to the database, which is set by default to 100. You can view the maximum number of connections using:

```
SHOW VARIABLES LIKE 'max_connections';
```

and change the maximum number of connections using, for example:

```
SET GLOBAL max_connections=150;
```

Configure Firewall

You need to allow external access to MySQL if you are running SpikeStream across several machines and your system's firewall may need to be changed to facilitate this. In SUSE this can be done by adding MySQL to the firewall configuration using YAST. If you are communicating with SIMNOS on Windows you will also need to open ports for each device, in addition to the Devices database (if this is on the Windows machine).

A1.3.3 Create Accounts

Root Account

Log in as root using `mysql -u root`

Display the current accounts: `SELECT user, host, password FROM mysql.user;`

Set a password for root: `SET password=PASSWORD(“secretpassword”)`

Get rid of unnecessary users: `DELETE FROM mysql.user WHERE user != “root”;`

Get rid of logins from outside machine: `DELETE FROM mysql.user WHERE host != “localhost”;`

SpikeStream Account

Create accounts with the user ‘SpikeStream’ and the password 'myPassword' that can access the database on localhost or a subnetwork:

```
GRANT ALL ON *.* TO SpikeStream@localhost IDENTIFIED BY “myPassword”;

GRANT ALL ON *.* TO SpikeStream@'192.168.1.0/255.255.255.0'
IDENTIFIED BY “myPassword”;
```

If these have been created successfully it should be possible to log into the database locally or from another machine on the same network using:

`mysql -uSpikeStream -pmyPassword` (local login with password “myPassword”)

`mysql -uSpikeStream -pmyPassword -h192.168.1.9` (remote login with mysql hosted on 192.168.1.9 and password “myPassword”)

You can create a different account for each database or put the databases on different machines. As long as the privileges are set up correctly it should work fine. The details for each database need to be added into the spikestream.config file on the main workstation.

A1.3.4 Create Databases and Tables

Create Database Script

Once you have configured the account(s), you can use a SpikeStream script to set up the databases. Open up the script in a text editor and change the user, host and password information to match the details you set earlier. When this information has been set correctly run it using:

```
$SPIKESTREAM_ROOT/scripts/CreateSpikeStreamDatabases
```

IMPORTANT NOTE: This script will overwrite the contents of all SpikeStream databases that are already on the system. It can also be used at a later point to reset all of the databases.

Manual Database Creation

Four SQL files are used to create the databases. These can be found at $SPIKESTREAM_ROOT/ database:

- *NeuralNetwork.sql*
- *NeuralArchive.sql*
- *Patterns.sql*

- *Devices.sql*

Another four SQL files are used to add neuron types, synapse types, probe types and devices to the databases that have been created.

- *AddNeuronTypes.sql*
- *AddSynapseTypes.sql*
- *AddProbeTypes.sql*
- *AddDevices.sql*

Finally, each neuron and synapse type needs an entry in the NeuronTypes and SynapseTypes tables indicating the location of their parameter table and the location of their class library. See the CreateSpikeStreamDatabases script for the commands needed to load these SQL files individually into the database.

IMPORTANT NOTE: The NeuralNetwork SQL sets up the database so that neuron ids start at 10, rather than 0. It is essential for the operation of the system that neuron ids 0-10 remain unused. These ids are generated each time a neuron is added to the system and I am not certain what happens when the automatically generated ids wrap around back to the beginning. It is worth keeping an eye on this and periodically re-initialise the database if necessary.

A1.4 Running SpikeStream

A1.4.1 Configuration

Open up the $SPIKESTREAM_ROOT/spikestream.config file and make sure that the database information is set correctly for the four databases. This only needs to be done on the main workstation since the database parameters are passed to SpikeStream Simulation and SpikeStream Archiver as command line parameters. I recommend leaving the database name untouched. You may also want to set the default location for saving and loading files. Once the config file has been saved you can start SpikeStream Application using the symbolic link “spikestream” in the SPIKESTREAM_ROOT/bin directory.

A1.4.2 PVM

On a single machine SpikeStream should launch pvm and run without problems. If you want to run SpikeStream across several machines, you will need to start pvm and add the other machines as hosts before starting a simulation using SpikeStream. SpikeStream Application can be running whilst you are doing this as long as a simulation is not initialized.

Getting pvm to work across several machines depends on being able to remotely invoke commands on the other machines using rsh (it can also be configured using ssh, but this probably incurs a significant performance penalty). Many Linux clusters are already set up for this, but configuring it from scratch on a new distribution can be a tricky process since rsh is usually disabled by default for security reasons. Finding the right place to set PVM_ROOT and SPIKESTREAM_ROOT so that they is available when pvm is remotely invoked can also cause problems. When pvm has been correctly configured you should be able to start it and add the remote host using the commands:

`pvm` (should return the prompt: “pvm>”)

```
pvm>add newHostName
```

If this works typing **`conf`** should list the new virtual machine configuration. Once the virtual machine has been configured SpikeStream will be able to run a simulation across multiple machines.

A1.4.3 Monitoring and Debugging Information

Some of the monitoring and debugging information that is available when running SpikeStream is as follows:

- The command line output of SpikeStream generally gives more information than is explicitly displayed in error messages. You will need to launch SpikeStream from the command line (rather than a desktop shortcut) to see this information.
- xpvm enables the monitoring of messages sent between the different processes.
- Output of processes started with pvm (all the simulation and archiving tasks) is routed to /tmp/pvml.1000. It can also be picked up using the task output feature of xpvm, although this can cause crashes when there is a large amount of output.
- Most SpikeStream modules have a file called Debug.h, which enables different types of debugging information to be displayed. The relevant part will have to be recompiled for this to take effect.
- pvm has a command line interface that lets you see what processes are running and kill them if necessary. Type pvm and then "help" to find out more about the available commands and look at the online documentation for pvm.

A1.4.4 Common Problems Running SpikeStream

A number of problems can arise when running SpikeStream:

- You will occasionally get an error message "FAILURE TO UPDATE DATABASE WITH TASKID", even when everything is set up correctly between SpikeStream and its databases. This is a bug that has not been sorted out. Restarting the simulation usually fixes the problem.

- When you have built SpikeStream and try clicking on spikestreamapplication with the mouse you may get an error message informing you that SPIKESTREAM_ROOT is not defined and SpikeStream will exit. If SpikeStream runs ok when you type **`./spikestream`** in the SPIKESREAM_ROOT/bin directory, this problem can be solved by logging out of your user account and logging in again. If SpikeStream does not run from the command line either, then you need to make sure that SPIKESREAM_ROOT is defined in the appropriate file for your shell (probably .bashrc). See Section A1.2.4 for more on this.
- Sometimes you will get errors along the lines of "mksocs() connect Connection Refused". This is probably due to a problem with pvm. If this happens, it is most likely due to some old files left over in /tmp from a previous simulation run that crashed. The best solution is to wait 30 seconds until SpikeStream times out, when it will ask you if you want to run the CleanPVM script. Run this script and the problem should go away. Persistent problems can often be solved by deleting all pvm related files from /tmp. The CleanPVM script can also be separately invoked to reset pvm and delete unused files from /tmp.
- SpikeStream will fail to connect with databases and devices on other machines if the firewalls on both machines are not set correctly.
- Simulations will not start if the dynamic neuron and synapse libraries cannot be found by the operating system (see Section A1.12.3). This may generate the message "libstdp1neuron.so: cannot open shared object file" or "libstdp1neuron.so: cannot open shared object file", which can be caused by omitting to run the install script as part of the installation process. It can also be caused by copying a library across from another machine, instead of recompiling it for your system.

- Simulations will not start if pvm is not installed properly. You can check that pvm is working correctly by typing **pvm**, which should return the pvm command prompt: **pvm>.**
- Loading a saved database occasionally creates problems when you have added or removed a neuron or synapse type, since the saved database contains tables with the old information. Similar problems can occur with the Devices database. If SpikeStream generates parameter errors or crashes after loading a database containing different neuron or synapse types, restarting it usually resolves the problem., which is caused by a bug in the parameter dialogs.
- If you have problems adding additional hosts to pvm make sure that you have rsh installed on your system, which may have been left off the default install for security reasons. You will also need to add the main workstation to your list of allowed hosts in .rhosts on the remote machines so that pvm can invoke commands on them without being prompted for the password. Use the IP address if you are working on a local network since the name of the machine may not be resolved (this will have to be set up each time the machines boot if you are using DHCP).
- With more recent versions of qwt you may get the error “libqwt.so.5: cannot open shared object file: No such file or directory”. This linking error arises because the operating system cannot find the qwt library that spikestream was compiled against. One way of solving this problem is to create a symbolic link that points to the appropriate libraries. To solve the qwt problem change to /usr/lib in super user mode, and type

  ```
  ln -s /usr/local/qwt-5.0.2/lib/libqwt.so.5 libqwt.so.5
  ```

  The details of this solution will change depending on the version of qwt that you are using.

- A similar problem can arise with mysqlpp libraries, which can be solved in a similar way by changing to /usr/lib in super user mode and typing:

  ```
  ln -s /usr/local/lib/libmysqlpp.so.2 libmysqlpp.so.2
  ```

  Again, the specific paths and library will change depending on the versions that you are using. Linking problems can also be solved by adding the appropriate locations to the LD_LIBRARY_PATH system variable, which is probably the best bet if you do not have root access to the system.

A1.4.5 Error Messages

When SpikeStream Application detects an error it generally displays an error message. When this error only affects the function that is currently being performed, SpikeStream will not exit, but you will probably want to restart SpikeStream (if possible after sorting the problem out). For example, if you get a database related error when loading a simulation, try to resolve the problem and then restart the simulation. When the error is likely to corrupt the database or make future work impossible, SpikeStream will immediately exit.

When simulation and archive tasks detect an error they will not exit immediately, but enter an error state in which they only respond to exit messages. This is to enable the simulation manager to do an explicit clean up after the end of the simulation without needing to restart and clean pvm. If you get an error message from a task, destroy the simulation, determine the cause of the error if possible and then restart the simulation. Let me know about any persistent problems and I will try to resolve them.

A1.4.6 Known Bugs and Missing Functionality

Known bugs and limitations in SpikeStream 0.1 are as follows:

- The probe feature is still under development and has not been fully implemented.

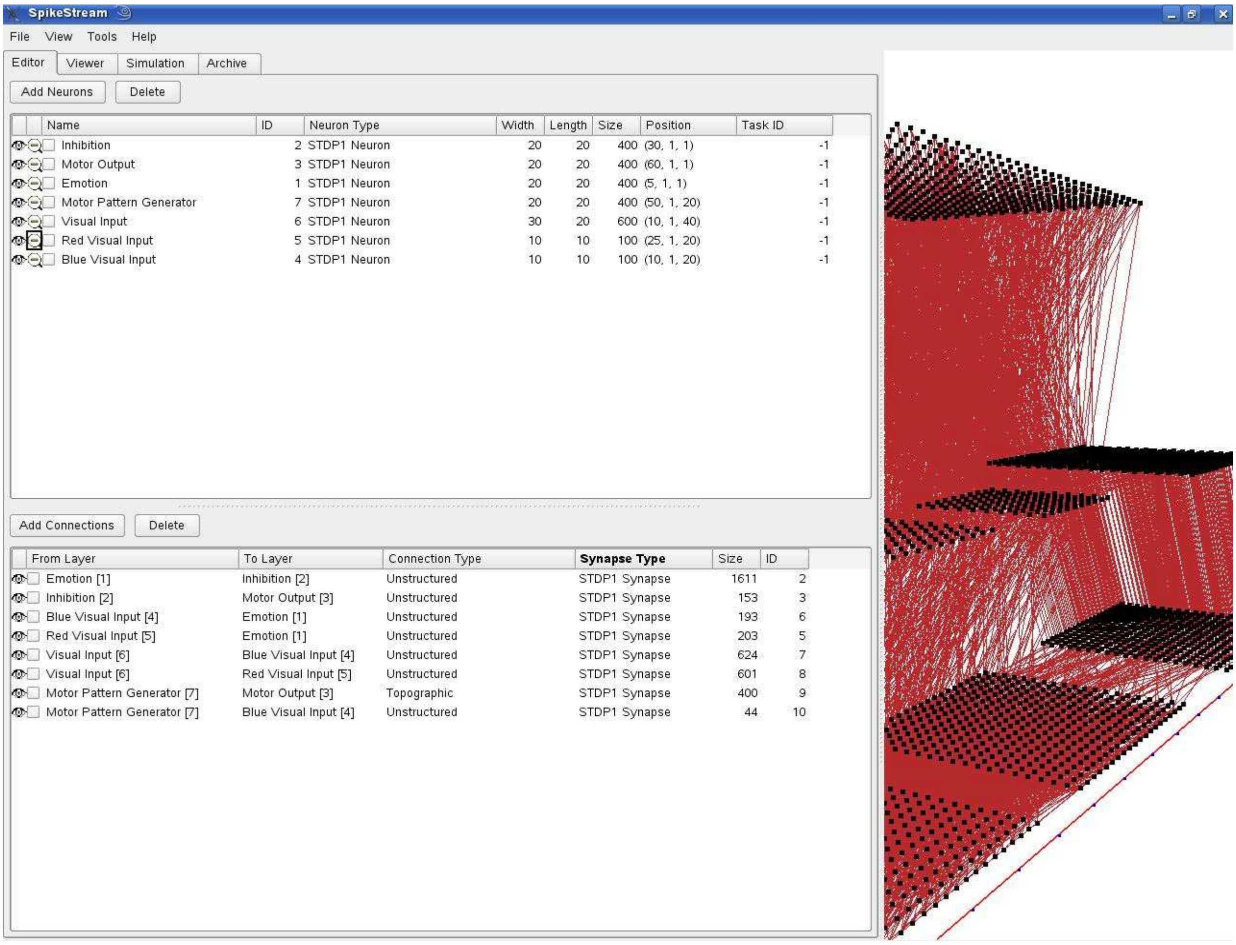
- Rotation of layers for patterns and devices is missing. Although it may be possible to connect a layer with width 10 and length 25 up to a device or pattern with width 25 and length 10, the simulation will not work. You are advised to only connect up layers to patterns or devices that have the same width and height as the layer.
- The ability to set and change the neuron spacing is not well tested. It should work, but it is best left at the default of 1.
- The simulation will only run for 2^{32} time steps, which is around 1000 simulated hours at 1 ms per timestep. After this, the simulation clock will overflow with unknown consequences.
- On later versions of Qt 3, the Network Monitor goes black when resized beyond a certain point. The firing patterns have been made dark red so that they can still be seen, but I have not found a better work around for this problem, which will probably disappear when SpikeStream is rewritten for Qt4.
- There is a limit to the maximum number of network monitors that can be open at once. This is currently 100, which is set using the variable MAX_NUMBER_MONITOR_WINDOWS in SPIKESTREAM_ROOT/include /GlobalVariables.h.
- Off center on surround connections are not implemented in the current version of SpikeStream.
- Make defaults button is not implemented on most of the parameter dialogs.
- Neuron and synapse types can only be changed when SpikeStream Application is not running. If SpikeStream Application is running when they are changed, it is likely to crash, but this will not affect the data in the database and restarting solves the problem.

- The exchange of spikes between SIMNOS and SpikeStream (see Section A1.9.4) is still at the early stages. This feature does work, but expect a certain amount of sweat and hassle to get everything running.
- The “Load Defaults” button is not implemented in the neuron or synapse parameter dialogs.
- The cancelling of operations is not well handled at present and may generate an error message when cancelling the loading of a simulation. A future version of SpikeStream will address this problem by using separate threads to handle heavy operations.
- The recording of network patterns is buggy and currently runs without synchronization to the spikesreamsimulation tasks. This occasionally results in the dropping of recorded time steps, particularly at the beginning or end of the simulation run. You may also get an error: “ArchiveWidget: MYSQL QUERY EXCEPTION MySQL server has gone away", which can be resolved by restarting SpikeStream. These problems will be sorted out in a later version of SpikeStream, which will tightly synchronize spikesreamarchiver with the simulation tasks.

A1.5 Creating Neural Networks

A1.5.1 The Editor Tab

The creation and editing of neural networks is carried out on the Editor tab (see Figure A1.1). The top table in the Editor tab shows information about the current neuron groups; the bottom table contains information about the connections between neuron groups.



Name	ID	Neuron Type	Width	Length	Size	Position	Task ID
Inhibition	2	STDP1 Neuron	20	20	400	(30, 1, 1)	-1
Motor Output	3	STDP1 Neuron	20	20	400	(60, 1, 1)	-1
Emotion	1	STDP1 Neuron	20	20	400	(5, 1, 1)	-1
Motor Pattern Generator	7	STDP1 Neuron	20	20	400	(50, 1, 20)	-1
Visual Input	6	STDP1 Neuron	30	20	600	(10, 1, 40)	-1
Red Visual Input	5	STDP1 Neuron	10	10	100	(25, 1, 20)	-1
Blue Visual Input	4	STDP1 Neuron	10	10	100	(10, 1, 20)	-1

From Layer	To Layer	Connection Type	Synapse Type	Size	ID
Emotion [1]	Inhibition [2]	Unstructured	STDP1 Synapse	1611	2
Inhibition [2]	Motor Output [3]	Unstructured	STDP1 Synapse	153	3
Blue Visual Input [4]	Emotion [1]	Unstructured	STDP1 Synapse	193	6
Red Visual Input [5]	Emotion [1]	Unstructured	STDP1 Synapse	203	5
Visual Input [6]	Blue Visual Input [4]	Unstructured	STDP1 Synapse	624	7
Visual Input [6]	Red Visual Input [5]	Unstructured	STDP1 Synapse	601	8
Motor Pattern Generator [7]	Motor Output [3]	Topographic	STDP1 Synapse	400	9
Motor Pattern Generator [7]	Blue Visual Input [4]	Unstructured	STDP1 Synapse	44	10

Figure A1.1. Editor Tab

Neuron Group Table

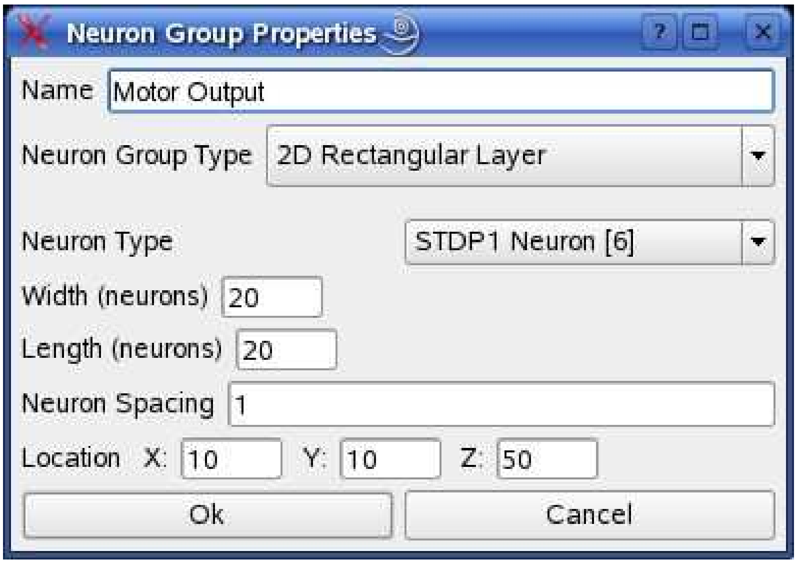
The top half of the Editor tab contains the neuron group table which displays information about the neuron groups in the database. The start of each row has an eye and a magnifier symbol. Clicking on the eye hides or shows a neuron group and you can click on the column header to hide or show all neuron groups. A single click on the magnifying glass zooms to the side of the appropriate neuron group. Click on it again and you are taken to the top of the appropriate neuron group. A third click returns you to a wide view of the entire network.

Connection Group Table

The bottom half of the Editor tab is taken up with the connection group table, which displays information about the connection groups in the database. At the left of each row is an eye symbol that can be used to show or hide the connection groups and you can click on the table header to view or hide all connection groups and to check or uncheck all of the tick boxes. Viewing of connection groups is disabled by default and very large connection groups will only be loaded when you attempt to view them, which may lead to a short delay whilst this is carried out. Virtual connections can never be viewed and are coloured light grey. Clicking on the blue "View" button in the connection group table shows the parameters that were used to create the connection group.

A1.5.2 Adding Neuron Groups

Clicking on the "Add Neurons" button above the neuron group table displays the Neuron Group Properties Dialog, shown in Figure A1.2.



Neuron Group Properties

Name: Motor Output

Neuron Group Type: 2D Rectangular Layer

Neuron Type: STDP1 Neuron [6]

Width (neurons): 20

Length (neurons): 20

Neuron Spacing: 1

Location X: 10 Y: 10 Z: 50

Figure A1.2. Neuron Group Properties Dialog

This dialog allows you to set the following information about the layer.

- **Name**. The name of the new neuron group
- **Neuron Group Type**. This combo box has three options.
  - *2D Rectangular Layer.* Creates a standard 2D layer 1 neuron thick.
  - *3D Rectangular Layer. C*reates a 3D layer. This is not fully implemented yet.

  - *SIMNOS Component*. Uses information from the Devices database to create a layer that connects to a sub-part of an input layer - see Section A1.9.4.
- **Neuron Type**. A list of the neuron classes in the NeuronTypes table.
- **Width**. The width of the neuron group in neurons.
- **Length**. The length of the neuron group in neurons.
- **Neuron Spacing**. Allows you to change the spacing between the neurons. *WARNING: This feature has not been fully tested and it is recommended to leave it at 1.*
- **Location**. The location of the bottom left corner of the neuron group when seen from above. Make sure that your selected location does not clash with an existing layer.

A1.5.3 Editing Neuron Groups

Some of the properties of a neuron group can be changed at a later point in time by right clicking on the neuron group in the neuron group table and selecting “Edit Neuron Group Properties” from the popup menu.

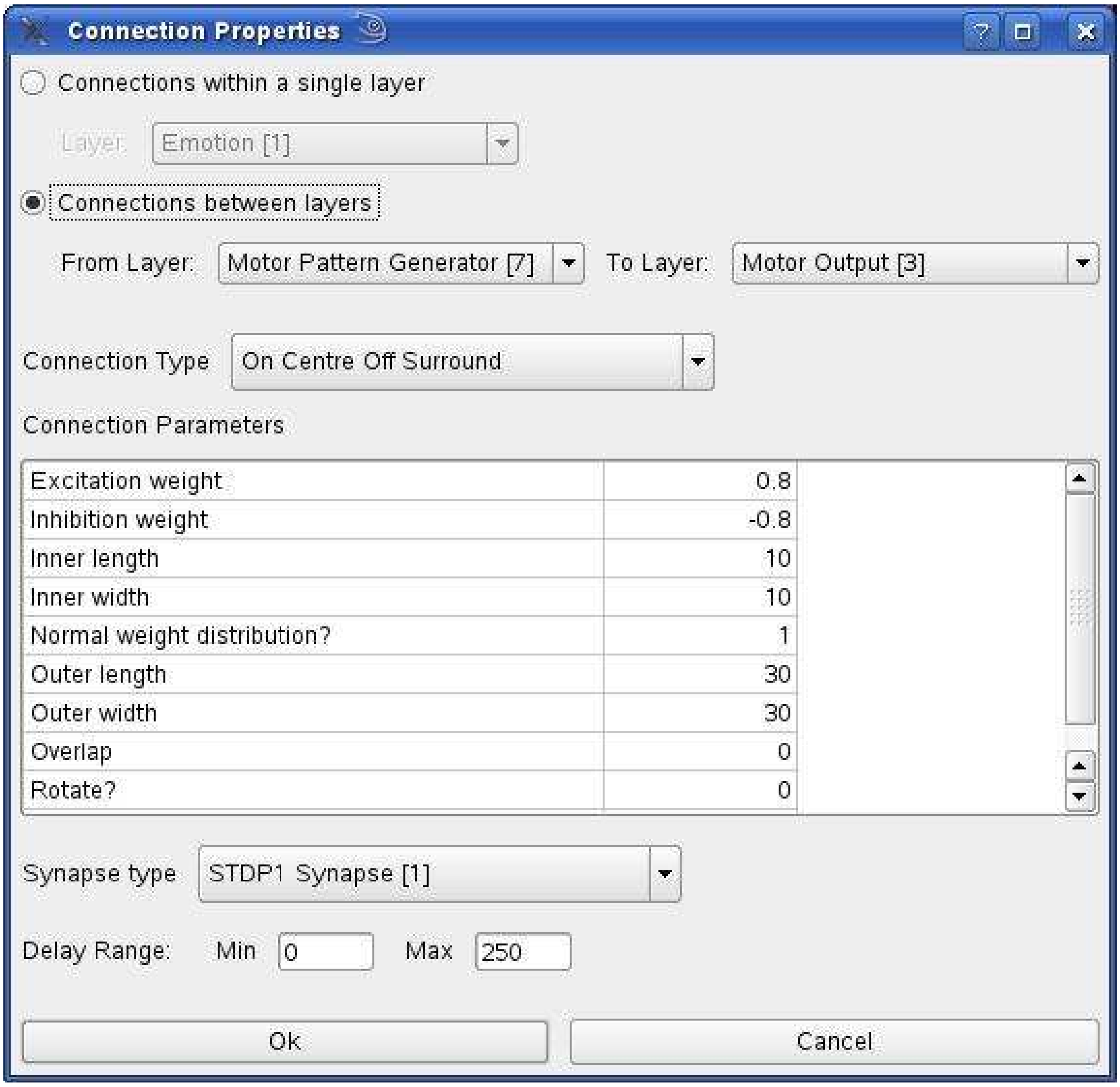
A1.5.4 Deleting Neuron Groups

Check the neuron groups that you want to delete and click on the “Delete” button. A dialog will popup to confirm your decision. Clicking “Ok” will permanently delete the neuron group from the database.

IMPORTANT NOTE: There is no undo function in SpikeStream and no method of reversing this step. Future work on SpikeStream may look into using the MySQL rollback feature to undo transactions.

A1.5.5 Adding Connection Groups

SpikeStream comes with a number of predefined connection patterns. Once you are familiar with SpikeStream you are likely to start creating your own connection patterns by directly editing the database (see Section A1.5.7). To use the built in connection patterns, start by clicking on "Add Connections". This launches the Connection Properties Dialog shown in Figure A1.3.



○ Connections within a single layer

Layer: Emotion [1]

● Connections between layers

From Layer: Motor Pattern Generator [7]   To Layer: Motor Output [3]

Connection Type: On Centre Off Surround

Connection Parameters

Excitation weight	0.8
Inhibition weight	-0.8
Inner length	10
Inner width	10
Normal weight distribution?	1
Outer length	30
Outer width	30
Overlap	0
Rotate?	0

Synapse type: STDP1 Synapse [1]

Delay Range: Min 0 Max 250

Figure A1.3. Connection Properties Dialog

The properties that can be set in this dialog are as follows:

- **Connections within a single layer/ between layers**. These radio buttons select between inter and intra layer connections. Different types of connection are available for each.

- **From layer**. The starting layer for the connection.
- **To layer**. The layer that the connection is made to.
- **Connection Type**. Several different connection types are available in the current version of SpikeStream.
    - **Simple Cortex**. Neurons are connected with short range excitatory connections and long range inhibitory connections. The parameters for this type of connection are given in Table A1.1.

Parameter	**Description**
Excitation connection probability	The number of neurons connected to within the excitation radius. Set to greater than 1 to increase the connection density; set to less than 1 to reduce the connection density.
Excitation radius	Select neurons within this radius for the neuron to connect to.
Excitation weight	The weight of excitation connections +/- the weight range. Weights can range from -1.0 to 1.0.
Inhibition connection density	The proportion of neurons connected to within the inhibition radius. Set to greater than 1 to increase the connection density; set to less than 1 to reduce the connection density.
Inhibition radius	Neurons within this radius, but outside of the excitation radius minus the overlap are selected for inhibitory connections.
Inhibition weight	The weight of inhibitory connections +/- the weight range.
Normal weight distribution	Randomness in the weight is selected using a normal distribution. 1 switches normal distribution on; 0 switches it off.
Overlap	Overlap between the inhibitory and excitatory connections
Weight range	The amount by which the weights can vary randomly.

Table A1.1. Simple cortex connection parameters

- **Unstructured excitatory (inter) and Unstructured excitatory (intra).** Unstructured connections in which each neuron makes all excitatory or all inhibitory connections. The parameters for this type of connection are given in Table A1.2.

Parameter	**Description**
Excitation connection prob	The probability of an excitatory neuron connecting to another excitatory neuron. This parameter can vary between 0 and 1.0.
Excitation weight	The weight of excitation connections +/- the weight range. Weights can range from -1.0 to 1.0.
Excitation weight range	The range of the excitation weight.
Excitation percentage	The percentage of excitatory neurons. Ranges from 0-100.
Inhibition connection prob	The probability of an inhibitory neuron connecting to another inhibitory neuron. This parameter can vary between 0 and 1.0.
Inhibition weight	The weight of inhibitory connections +/- the weight range. Weights can range from -1.0 to 1.0.
Inhibition weight range	The range of the inhibitory weights.

Table A1.2. Unstructured excitatory (inter) and Unstructured excitatory (intra) parameters

- **On Center Off Surround**. Rectangular connection with an excitatory centre and inhibitory surround. The *to* layer must be smaller than the *from* layer for this type of connection to work. The parameters for this type of connection are given in Table A1.3. *WARNING: Some of these parameters are not fully tested.*

Parameter	**Description**
Excitation weight	The weight of excitation connections +/- the weight range. Weights can range from -1.0 to 1.0.
Inhibition weight	The weight of inhibitory connections +/- the weight range. Weights can range from -1.0 to 1.0.
Inner length	The length of the central excitatory connection area.
Inner width	The width of the central excitatory connection area.
Outer length	The length of the inhibitory connection area.
Outer width	The width of the inhibitory connection area.
Overlap	Overlap between the excitatory and inhibitory connection areas.
Rotate	One layer may be rotated relative to the other one.
Weight range	The amount by which the weights can vary randomly.

Table A1.3. On centre off surround connection parameters

- **Off Centre On Surround**. Similar to on centre off surround connections. Note that the to layer must be smaller than the from layer for this type of connection to work. The parameters for this type of connection are given in Table A1.4. *IMPORTANT NOTE*: *Not implemented at present.*

Parameter	**Description**
Excitation weight	The weight of excitation connections +/- the weight range. Weights can range from -1.0 to 1.0.
Inhibition weight	The weight of inhibitory connections +/- the weight range. Weights can range from -1.0 to 1.0.
Inner length	The length of the central inhibitory connection area.
Inner width	The width of the central inhibitory connection area.
Outer length	The length of the excitatory connection area.
Outer width	The width of the excitatory connection area.
Overlap	Overlap between the excitatory and inhibitory connection areas.
Rotate	One layer may be rotated relative to the other one.
Weight range	The amount by which the weights can vary randomly.

Table A1.4. Off centre on surround connection parameters

- **Unstructured**. Each neuron in the from layer is connected to a random number of neurons in the to layer. The parameters for this type of connection are given in Table A1.5.

Parameter	**Description**
Average weight	The weight of connections +/- the weight range. Weights can range from -1.0 to 1.0.
Connection density	The proportion of neurons connected to. This parameter can vary between 0 and 1.0.
Weight range	The amount by which the weights can vary randomly.

Table A1.5. Unstructured connection parameters

- **Virtual**. In order to run the simulation, each neuron group needs to be connected to at least one other neuron group. When there are no functional connections, virtual connections need to be created between neuron groups so that they can be synchronized in the simulation. *NOTE: The simulation may also create temporary virtual connections to enable synchronization between the layers. The creation and destruction of these does not require any intervention by the user.*

- **Topographic**. This creates topographic connections between the layers. The parameters for topographic connections are given in Table A1.6.

Parameter	**Description**
Average weight	The weight of connections +/- the weight range. Weights can range from -1.0 to 1.0.
Overlap	When layers of different size are topographically connected there can be an overlap between each set of connections.
Rotate	One layer can be rotated relative to the other.
Weight range	The amount by which the weights can vary randomly.

Table A1.6. Topographic connection parameters

- **Synapse Type**. Selects one of the currently selected synapse classes for the connection.
- **Delay Range**. Sets the range of delay *expressed in timesteps*. The absolute value of the delay for each connection is the update time per timestep multiplied by the number of timesteps delay.

A1.5.6 Deleting Connection Groups

Select the connection groups that you want to delete and press the “Delete” button above the connections table. Press “Ok” to confirm deletion and the connection groups will be removed from the database.

IMPORTANT NOTE: There is no undo function in SpikeStream and no method of reversing this step. Future work on SpikeStream may look into using the MySQL rollback feature to undo transactions.

A1.5.7 Other Ways to Create Neuron and Connection Groups

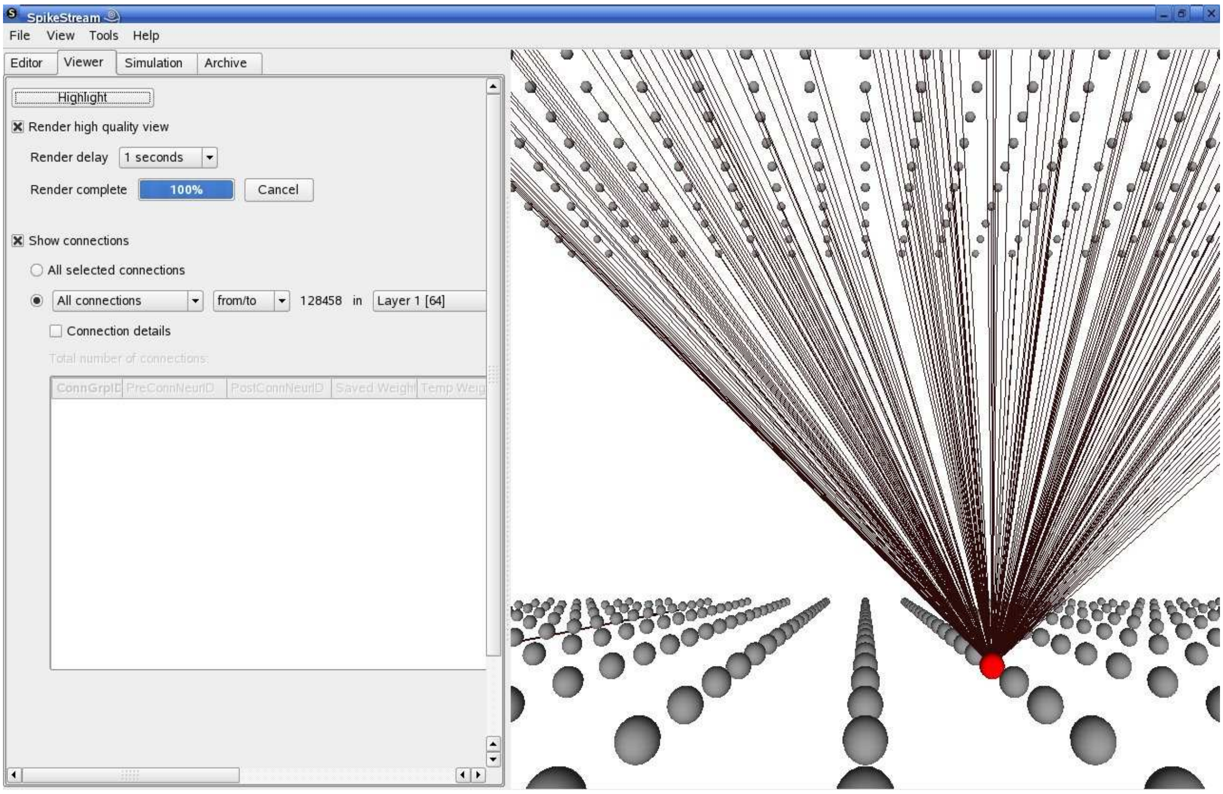
The preset ways of creating and editing neuron and connection groups in SpikeStream Application are hard coded and can only be changed by modifying SpikeStream. However, it is reasonably easy to write your own programs or scripts to add new neurons or connection patterns to the SpikeStream database. The following limitations apply when doing this:

- Any pair of neurons can only have a single connection between them.
- Each neuron group can only have one connection of each type between it. Thus, there can be several connection groups of different types between two layers, but not two connection groups of the same type.
- SpikeStream can visualize neuron groups of any shape, but it is currently unable to connect patterns or devices to non-rectangular neuron groups, or to provide live monitoring of non-rectangular neuron groups.

A1.6 Viewing Neural Networks

A1.6.1 Viewer Tab

The Network Viewer (see Figure A1.4) enables networks to be viewed in three dimensions. This three dimensional window is permanently on the right hand side of the screen and its size can be adjusted by grabbing the dividing bar. The Network Viewer tab has controls that enable you to view different aspects of the connections and set the rendering properties.



☒ Render high quality view

Render delay 1 seconds

Render complete 100%

☒ Show connections

○ All selected connections

● All connections from/to 128458 in Layer 1 [64]

□ Connection details

Total number of connections:

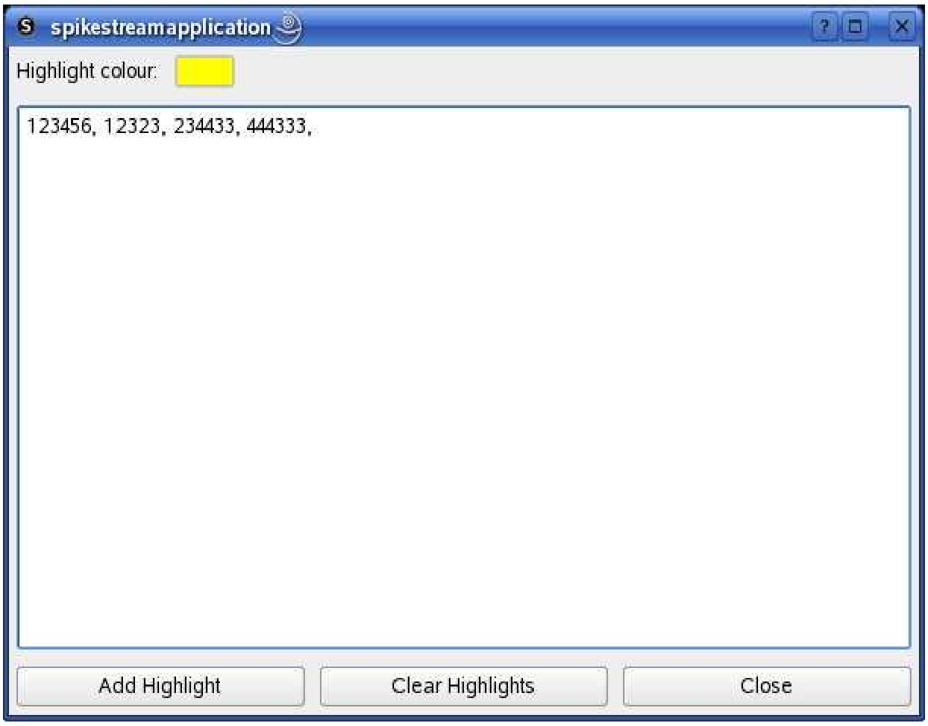
ConnGrpID	PreConnNeurID	PostConnNeurID	Saved Weight	Temp Weig

Figure A1.4. Network Viewer tab (left) and Network Viewer (right)

The controls available in the Network Viewer Tab are covered in the next few sections.

Highlight

Clicking on the highlight button launches the Highlight Dialog shown in Figure A1.5. Type or paste in a list of comma separated neuron IDs that you want to highlight and click on “Add Highlight” to highlight them. The colour can be changed by clicking on the colour field. Multiple groups of neurons can be highlighted in different colours.



Highlight colour: 🟨

123456, 12323, 234433, 444333,

Figure A1.5. Highlight Dialog

Render Settings

Normally neurons are drawn using simple vertices, which considerably speeds up the rendering time. However, if you want a more attractive view, you can check this box to draw neurons as grey spheres. The render delay sets the time between the last navigation event in Network Viewer and the start of the render.

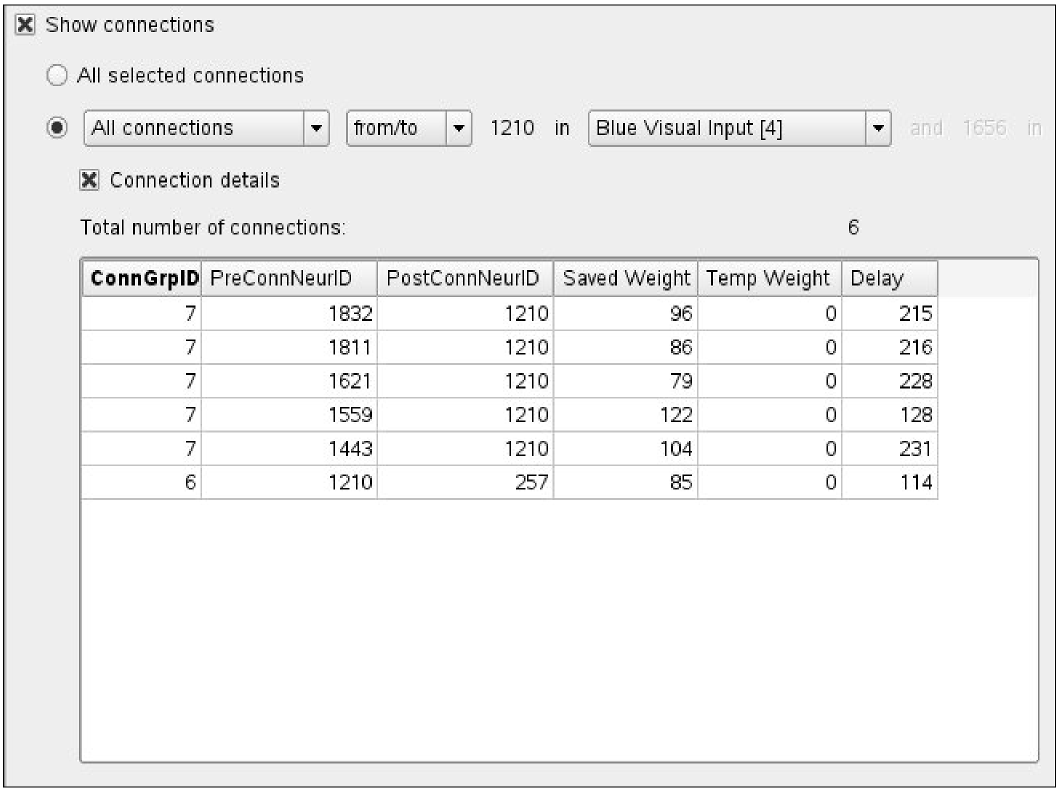
Connection Settings

When the Show Connections check box is selected the Network Viewer displays all of the connections that are set as visible in the Connection Group Table. This part of the Network Viewer tab is very useful for showing different aspects of the connections between neurons and it is also used to select the neurons for monitoring or noise injection in the Simulation tab. If you want to select a subset of the connections for viewing, the following options are available:

- **All connections.** Shows positive and negative connections.

- **Positive connections.** Only connections with positive weights are shown.
- **Negative connections.** Only connections with negative weights are shown.
- **from/to**. Connections from and to the selected neuron in the selected neuron group are shown
- **from**. Connections from the selected neuron in the selected neuron group are shown
- **to** . Connections to the selected neuron in the selected neuron group are shown
- **between**. Connections between the first selected neuron and the second selected neuron are shown. Use this mode to select an individual synapse for monitoring during a simulation.

The connection details check box displays information about the selected connections (see Figure A1.6. In this table, “Saved Weight” is the weight that is loaded up at the beginning of a simulation as the synapse's starting weight. As the simulation progresses, this weight may change and the user can view the current value of the weights by pressing “View Weights” in the Simulation tab. The synapse's current weight is then visible in the “Temp Weight” column of this table. If the user chooses to permanently save the weights during a simulation, their values are written to the Saved Weight field and will become the starting weights when the simulation is next initialised.



☒ Show connections

○ All selected connections

● All connections ▾ from/to ▾ 1210 in Blue Visual Input [4] ▾ and 1656 in

☒ Connection details

Total number of connections: 6

ConnGrpID	PreConnNeurID	PostConnNeurID	Saved Weight	Temp Weight	Delay
7	1832	1210	96	0	215
7	1811	1210	86	0	216
7	1621	1210	79	0	228
7	1559	1210	122	0	128
7	1443	1210	104	0	231
6	1210	257	85	0	114

Figure A1.6. Connection Details Table

A1.6.2 Network Viewer

The Network Viewer shows all of the visible neurons and connections in three dimensions. This display starts out with the Z axis vertical, the X axis horizontal and to the right and the Y axis going into the display away from the viewer. You can navigate around this window using the following controls:

- **Arrow-left**. Moves camera left.
- **Arrow-right.** Moves camera right.
- **Arrow-up**. Moves camera up.
- **Arrow-down**. Moves camera down.

- **Ctrl** + **Arrow-left**. Rotates camera left.
- **Ctrl** + **Arrow-right**. Rotates camera right.
- **Ctrl** + **Arrow-up**. Rotates camera up.

- **Ctrl** + **Arrow-down**. Rotates camera down.

- **Ctrl** + **=**. Zooms in.
- **Ctrl** + **-**. Zooms out.
- **Ctrl + Y**. Zooms out to show all layers.

When viewing connections *from/to*, *from* and *to* an individual neuron, the neuron will be highlighted in red and the selected neuron can be changed using the following controls:

- **ALT + Arrow-right**. Selects the next neuron within the group moving along X positive.
- **ALT + Arrow-left**. Selects the next neuron within the group moving along X negative.
- **ALT + Arrow-up**. Selects the next neuron within the group moving along Y positive.
- **ALT + Arrow-down**. Selects the next neuron within the group moving along Y negative.

When viewing connections *between* two individual neurons, the *from* neuron will be highlighted in red and the selected neuron can be changed using the controls that have just been outlined. The *to* neuron will be highlighted in green and the selected *to* neuron can be changed using the following controls.

- **SHIFT** + **ALT** + **Arrow-right**. Selects the next neuron within the *to* group moving along X positive.
- **SHIFT** + **ALT** + **Arrow-left**. Selects the next neuron within the *to* group moving along X negative.
- **SHIFT** + **ALT** + **Arrow-up**. Selects the next neuron within the *to* group moving along Y positive.
- **SHIFT** + **ALT** + **Arrow-down**. Selects the next neuron within the *to* group moving along Y negative.

WARNING: Occasionally the Network Viewer loses keyboard focus, which may cause the keyboard to control other aspects of SpikeStream. This is rarely serious, but I have accidentally quit the application on occasions by inadvertently navigating through the file menu. Click on the Network Viewer to restore keyboard focus.

A1.6.3 View Menu

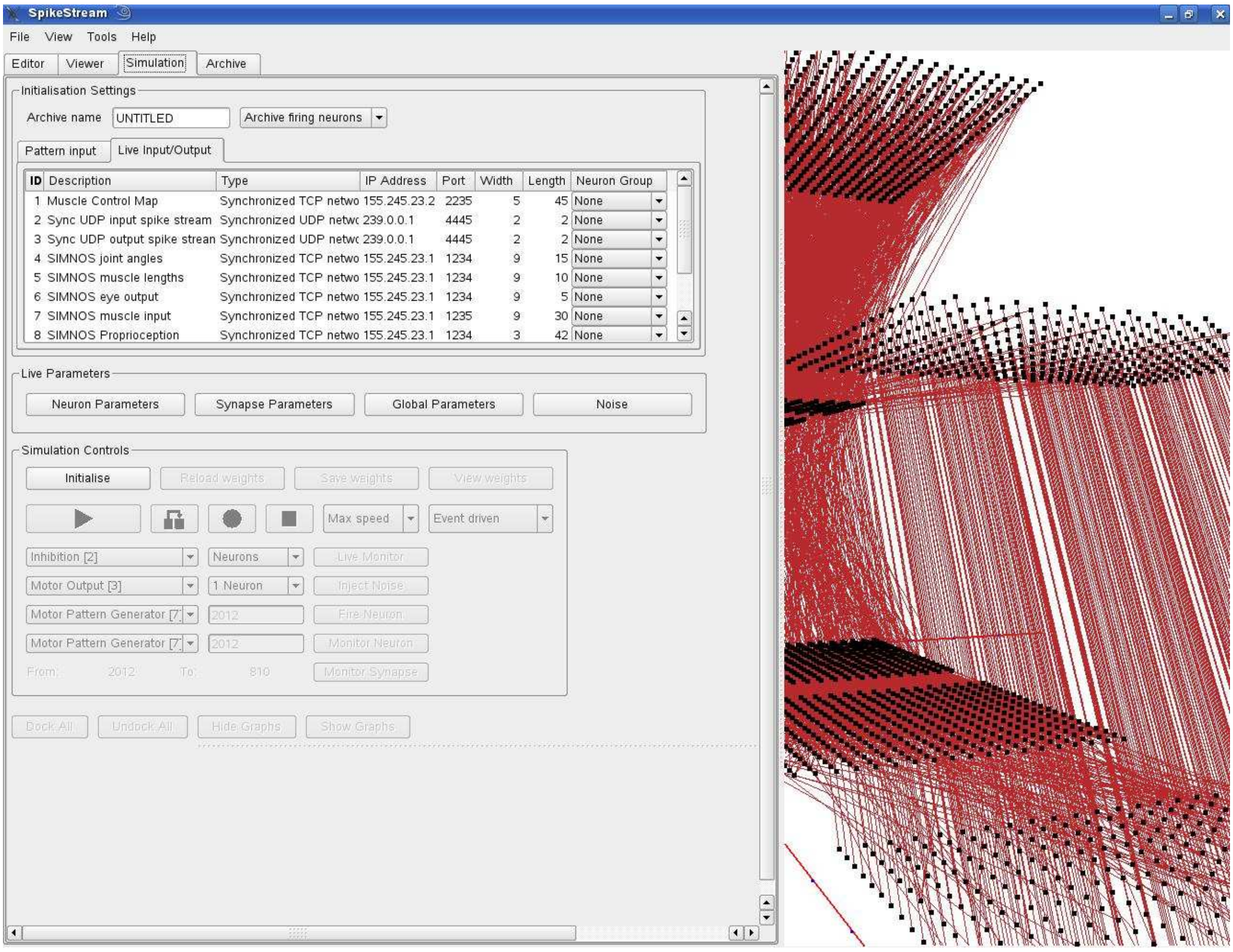
The view menu on the main menu bar allows you to selectively refresh information in SpikeStream:

- **View->Reload devices Ctrl+D**. Reloads the list of devices in the Simulation tab.
- **View->Reload patterns Ctrl+P**. Reloads the list of patterns in the Simulation tab.
- **View->Reload everything Shift+F5**. Reloads everything, including neuron and connection groups, parameters, patterns and devices.

A1.7 Running a Simulation

A1.7.1 Simulation Tab

The Simulation tab (see Figure A1.7) is used to control all aspects of a simulation.



Archive name UNTITLED
Archive firing neurons

ID	Description	Type	IP Address	Port	Width	Length	Neuron Group
1	Muscle Control Map	Synchronized TCP netwo	155.245.23.2	2235	5	45	None
2	Sync UDP input spike stream	Synchronized UDP netwc	239.0.0.1	4445	2	2	None
3	Sync UDP output spike strean	Synchronized UDP netwc	239.0.0.1	4445	2	2	None
4	SIMNOS joint angles	Synchronized TCP netwo	155.245.23.1	1234	9	15	None
5	SIMNOS muscle lengths	Synchronized TCP netwo	155.245.23.1	1234	9	10	None
6	SIMNOS eye output	Synchronized TCP netwo	155.245.23.1	1234	9	5	None
7	SIMNOS muscle input	Synchronized TCP netwo	155.245.23.1	1235	9	30	None
8	SIMNOS Proprioception	Synchronized TCP netwo	155.245.23.1	1234	3	42	None

Max speed
Event driven
Inhibition [2] Neurons
Motor Output [3] 1 Neuron
Motor Pattern Generator [7] 2012
Motor Pattern Generator [7] 2012
From: 2012 To: 810

Figure A1.7. Simulation tab

A1.7.2 Archive Name and Type

At the top of the Simulation tab is a box where you can enter a name for the archive. This archive will only be stored if you record data from the simulation. There is also a combo box that enables you to select between recording the firing neuron patterns from a layer or the spikes emitted from a layer. The firing neurons option is recommended because it has been more thoroughly tested. The archive name can be changed at a later point using the Load Archive Dialog.

A1.7.3 Patterns and Devices

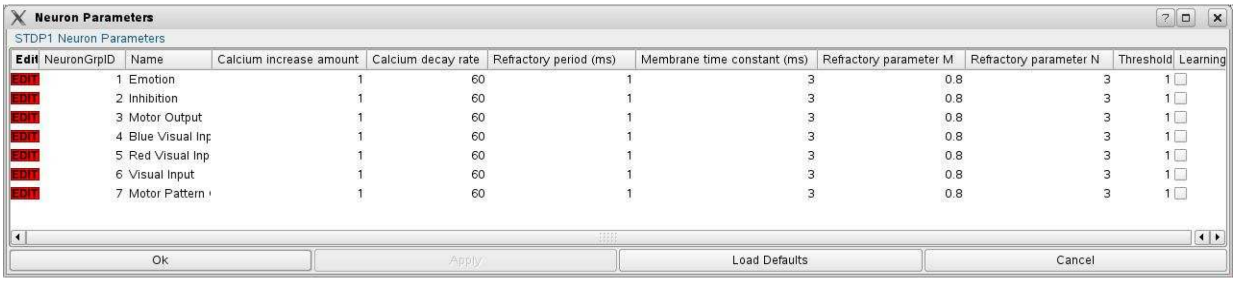
The next part of the Simulation tab is another set of tabs that let you connect patterns and devices up to layers in the simulation. Each of the combo boxes in these tables only displays the layers that are the correct size for the pattern or device. Selecting the layer in the combo box will connect the pattern or device up to the layer when the simulation is initialized. If you add a new device to the Devices table you can refresh the devices table by clicking on "View->Reload devices" or pressing CTRL+D. At the bottom of the pattern table is a text box where you can set the number of time steps between each pattern. For example, if you set this to ten, a pattern will be applied every ten time steps. This is particularly important when you are using patterns that are spread over time. See Section A1.9 for more information on devices and Section A1.10 for more information about patterns.

A1.7.4 Parameters

Parameters for the simulation are set using the four buttons in the "Parameters" section of the Simulation tab.

Neuron Parameters

Clicking on the "Neuron Parameters" button brings up the dialog shown in Figure A1.8, where you can set the parameters for the simulation. This dialog edits the neuron parameters table in the database and these parameters can be changed at any point during a simulation run.

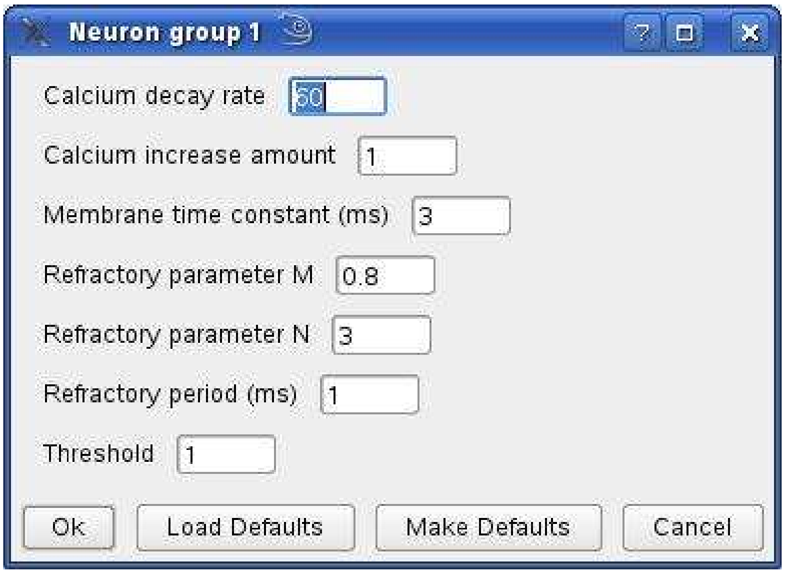


STDP1 Neuron Parameters

Edit	NeuronGrpID	Name	Calcium increase amount	Calcium decay rate	Refractory period (ms)	Membrane time constant (ms)	Refractory parameter M	Refractory parameter N	Threshold	Learning
EDIT	1	Emotion	1	60	1	3	0.8	3	1	□
EDIT	2	Inhibition	1	60	1	3	0.8	3	1	□
EDIT	3	Motor Output	1	60	1	3	0.8	3	1	□
EDIT	4	Blue Visual Inp	1	60	1	3	0.8	3	1	□
EDIT	5	Red Visual Inp	1	60	1	3	0.8	3	1	□
EDIT	6	Visual Input	1	60	1	3	0.8	3	1	□
EDIT	7	Motor Pattern	1	60	1	3	0.8	3	1	□

Figure A1.8. Neuron Parameters Dialog

To change the parameters, click on the edit button for a particular layer and an Edit Neuron Parameters Dialog will be launched that enables you to adjust the parameters (see Figure A1.9).



Calcium decay rate 50

Calcium increase amount 1

Membrane time constant (ms) 3

Refractory parameter M 0.8

Refractory parameter N 3

Refractory period (ms) 1

Threshold 1

Figure A1.9. Edit Neuron Parameters Dialog

Pressing “Ok” in this second dialog updates the Neuron Parameters Dialog, but *will not update the simulation until you press “Ok” or “Apply” within the Neuron Parameters Dialog*. Boolean parameters are set using the check boxes within the Neuron Parameters Dialog.

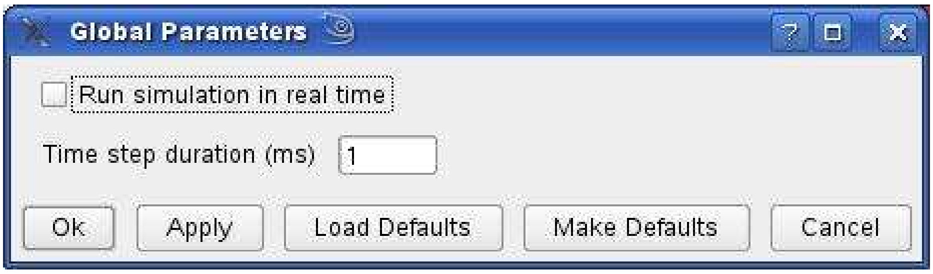
IMPORTANT NOTE: The “Load Defaults” button is not implemented in the Neuron Parameters Dialog and the “Make Defaults” button has not been implemented in the Edit Neuron Parameters Dialog.

Synapse Parameters

The editing of synapse parameters proceeds in a similar way to the editing of neuron parameters.

Global Parameters

This dialog (see Figure A1.10) controls parameters that are global to the simulation. Checking “Run simulation in real time” will update the simulation clock in real time instead of using the time step duration value. “Time step duration” enables you to set the amount of time that is simulated by each time step. Smaller values will lead to a more accurate simulation, but may also increase the amount of time taken to compute the simulation.



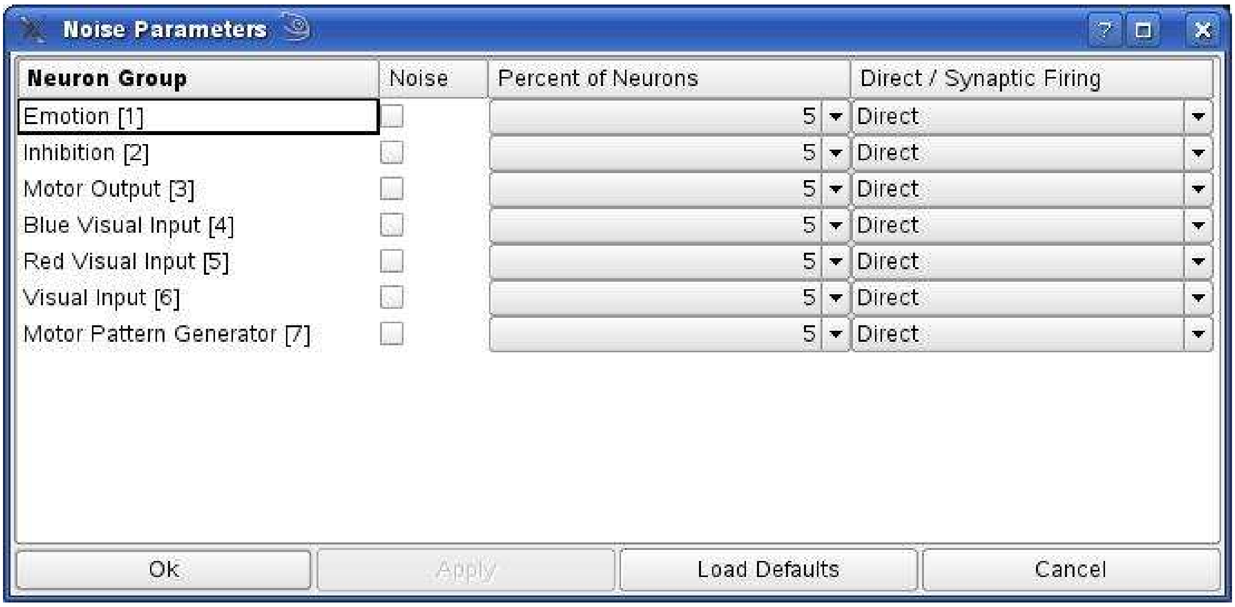
□ Run simulation in real time

Time step duration (ms) 1

Figure A1.10. Global Parameters Dialog

Noise

This dialog (see Figure A1.11) enables you to add random noise to the neuron groups.



Neuron Group	Noise	Percent of Neurons	Direct / Synaptic Firing
Emotion [1]	□	5	Direct
Inhibition [2]	□	5	Direct
Motor Output [3]	□	5	Direct
Blue Visual Input [4]	□	5	Direct
Red Visual Input [5]	□	5	Direct
Visual Input [6]	□	5	Direct
Motor Pattern Generator [7]	□	5	Direct

Figure A1.11. Noise Dialog

The second column enables or disables noise for the neuron group. The third column selects the percentage of neurons that will be randomly selected from each neuron group at each time step. There is also a “random” option that selects a random percentage of neurons at each time step. The last column selects between direct and synaptic firing. In direct noise mode, the selected neurons are directly fired by the simulation. In synaptic noise mode, the specified synaptic current is injected into the neuron at each time step, which may or may not lead to firing.

A1.7.5 Simulation Controls

The next set of controls are for running and monitoring the simulation and for the manual injection of noise. These controls are only enabled when the simulation is initialized (see Figure A1.12).

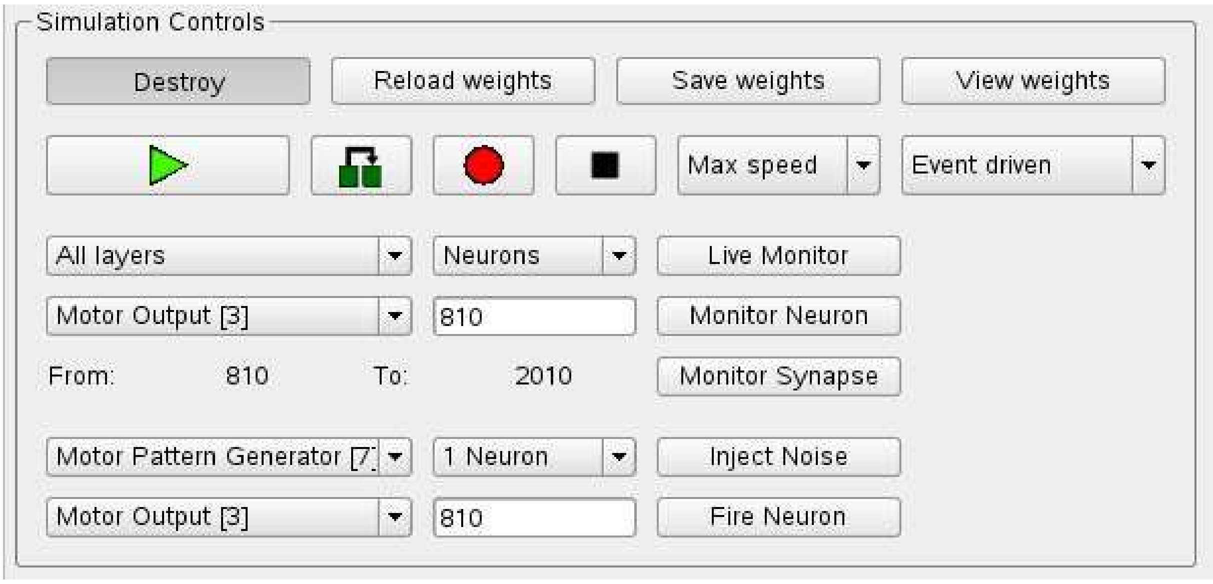


Figure A1.12. Simulation controls

Initialise / Destroy

When initialize is pressed, pvm is used to launch the simulation across all the hosts that have been added to the virtual machine. These are created as separate tasks running in parallel, with one task per neuron group. An extra task is created for the archiving of the simulation. Pressing “Destroy” causes all of these tasks to exit.

Weight Buttons

During a simulation run these buttons offer the following functions:

- **Reload weights**. Requests each task to reload its weights from the database.
- **Save weights**. Requests each task to save its current weights to the database.
- **View weights.** Requests each task to save its current weights to the "Temp Weight" field in the database. This enables the user to view the weights without permanently changing them.

Transport Buttons

The simulation is run using a standard set of transport buttons:

- **Play**. Plays and stops the simulation.
- **Step**. Advances to the next time step. Strange behaviour with pvm message passing can lead each step to take a second or two.
- **Record**. Records the simulation using the specified archive name.
- **Stop**. Stops the simulation.

The first combo box after the stop button can be used to slow the simulation down, which is extremely useful for monitoring what is going on in. The last combo in this row is used to control the update mode of the simulation:

- **Event driven**. The fastest update mode. Neuron and synapse classes are only updated when they receive a spike.
- **Update all neurons**. All neuron classes are updated at each time step. Synapses are only updated when they receive a spike. Useful for neural models that display spontaneous activity.
- **Update all synapses**. All synapse classes are updated at each time step. Neurons are only updated when they receive a spike.

- **Update everything**. All neuron and synapse classes are updated at each time step. In this mode, SpikeStream operates like a synchronous simulator.

Monitoring

The next set of controls are used to monitor what is going on in the simulation.

- **Live Monitor.** Launches a window displaying the firing state of the selected neuron group or all the neuron groups. This window can display the spikes emitted by the neuron group or the firing of the neurons in the neuron group.
- **Monitor Neuron**. Each neuron class can define its own set of variables for live monitoring. Select a neuron using Network Viewer or type in a neuron ID and click this button to draw a live graph of the monitored variables for the neuron (see Figure A1.13). *NOTE: If this is launched part way through a simulation, it may take a little while to adjust itself.*
- **Monitor Synapse**.. Each synapse class can define its own set of variables for live monitoring. To select a synapse you need to set the Network Viewer tab to 'between' mode. You should have a green neuron and a red neuron highlighted. Select a synapse using the Network Viewer and click “Monitor Synapse” to draw a live graph of the monitoring variables for the synapse. *NOTE: If this is launched part way through a simulation, it may take a little while to adjust itself.*

Closing these windows stops the monitoring data being sent from the tasks simulating the neuron group.

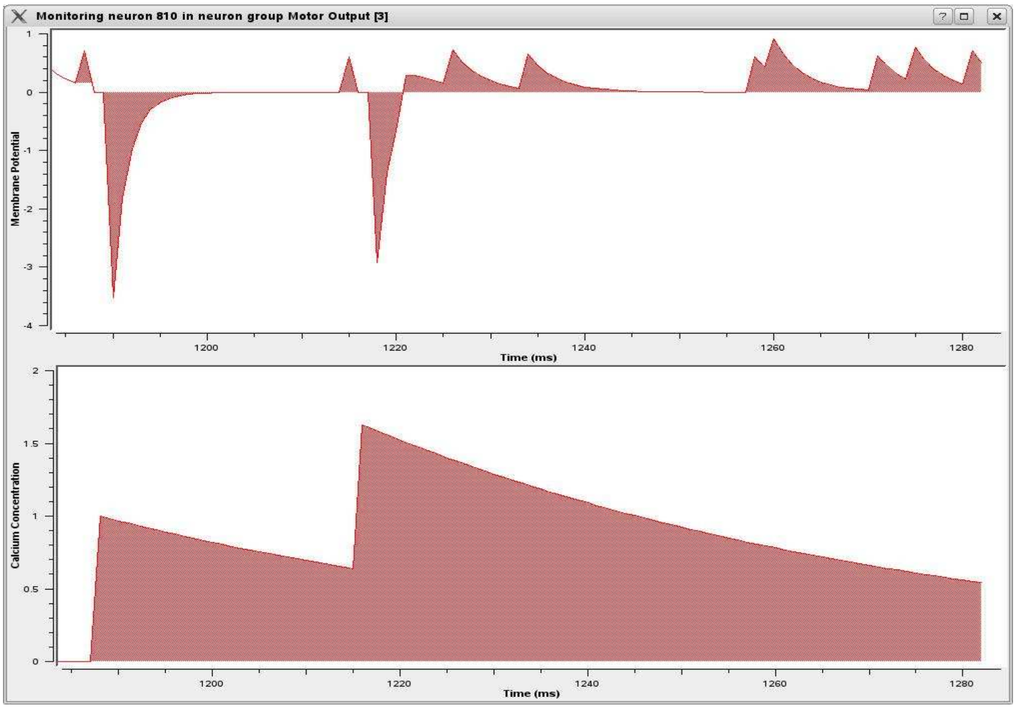


Figure A1.13. Graphs of monitored neuron variables

NOTE: The values in this graph are sampled every time step so with a high time step value of 10ms, for example, you may not see any change on the membrane potential in response to incoming spikes because the neuron will have reset itself to zero by the end oft each time step.

Noise Injection

Controls that can be used to manually inject noise into a neuron group within a single simulation step:

- **Inject Noise.** Fires the specified percentage of neurons once within a simulation step.
- **Fire Neuron**. Fires the specified neuron once within a simulation step. The neuron's id can be typed into the field or selected using the Network Viewer.

Docking Controls

A number of buttons are available to selectively hide and show monitoring information.

- **Dock All**. Places all live monitor windows in the docking area. These windows will continue to display the neuron patterns whilst they are in the docking area and they can be dragged around and rearranged.
- **Undock All**. Restores all live monitor windows to their original location.
- **Hide Graphs**. Makes all graphs invisible and switches their plotting off.
- **Show Graphs**. Makes all the current graphs visible and switches their plotting on.

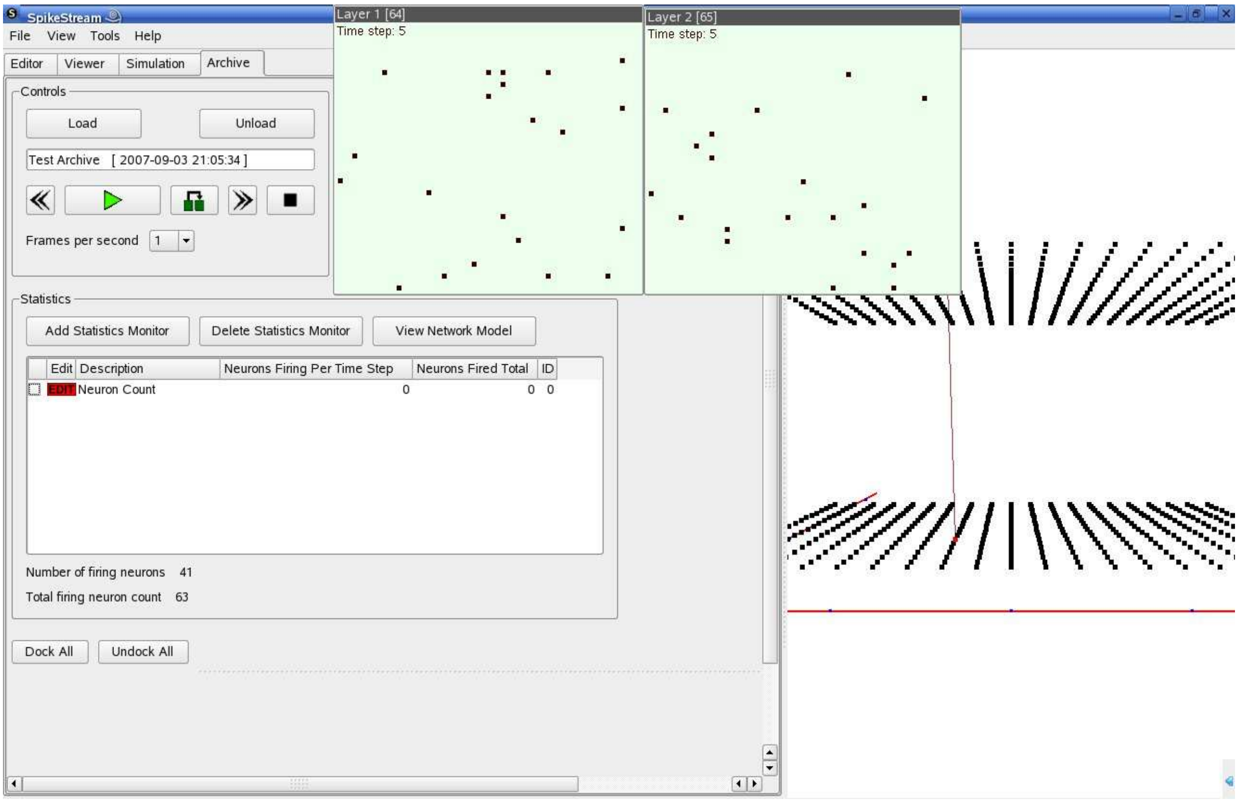
A1.7.6 Network Probes

Clicking on “Tools->Probe manager” launches a dialog to manage the probes. Network Probes are designed to run alongside the simulation and carry out actions on the neural network for testing purposes. For example, a network probe might be created to stimulate parts of the network with noise in order to identify its effective connectivity. *NOTE: This feature is still under development and should be ignored.*

A1.8 Archives

A1.8.1 Archive Tab

The recording of archives is carried out in the Simulation tab. Archives are played back in the Archive tab shown in Figure A1.14.



Test Archive [2007-09-03 21:05:34]

Frames per second 1

	Edit	Description	Neurons Firing Per Time Step	Neurons Fired Total	ID
□	EDIT	Neuron Count	0	0	0

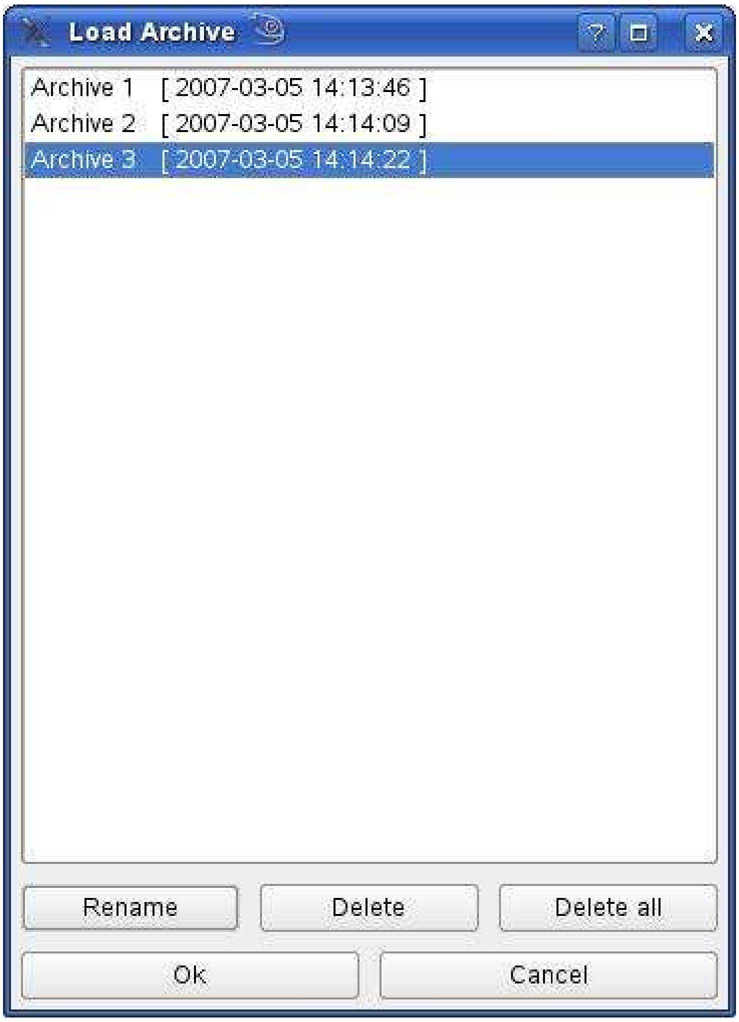
Number of firing neurons 41

Total firing neuron count 63

Figure A1.14. Archive tab

Loading and Playing Back an Archive

To load an archive press the “Load” button, which will open up the Load Archive Dialog, shown in Figure A1.15, which has controls to rename and delete archives. When you have selected your archive and pressed “Ok”, the archive will be loaded and can be replayed, stepped through, rewound etc. using the controls available in the Archive tab.

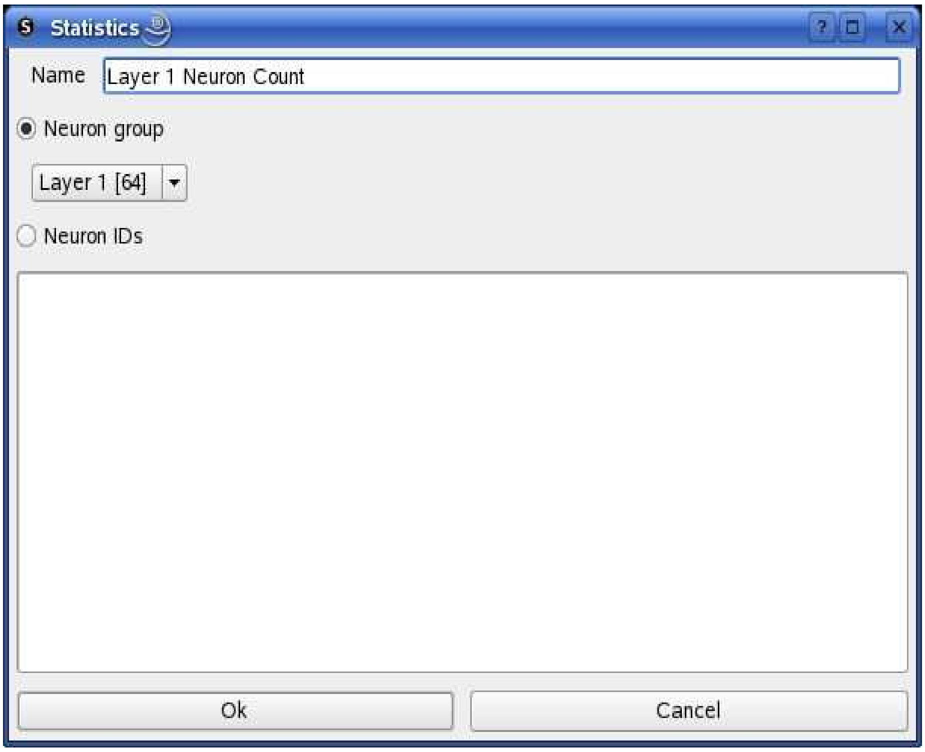


Archive 1 [2007-03-05 14:13:46]
Archive 2 [2007-03-05 14:14:09]
Archive 3 [2007-03-05 14:14:22]

Figure A1.15. Load Archive Dialog

Archive Statistics

Statistics about the archive can be gathered by adding a statistics monitor to count the number of times a neuron fires, the number of times a range of neurons fire, or the number of times neurons fire in a particular neuron group. Clicking on the "Add Statistics Monitor" button launches the dialog shown in Figure A1.16. In this dialog you can choose to monitor the number of times neurons fire in a particular layer or count the number of times one or a number of neuron IDs fire, which is done by adding the neuron IDs as a comma separated list. OR, AND and range operators are supported, for example: 12121 & 12121, 1323 - 56565, 123213 | 098098.



Name Layer 1 Neuron Count

● Neuron group

Layer 1 [64]

○ Neuron IDs

Figure A1.16. Archive Statistics Dialog

There is also a button that allows you to view the XML network model associated with the archive (see next section), which may be different from the network model that is currently loaded into SpikeStream.

A1.8.2 Archive Structure

Each archive contains a summary of the neuron groups stored in XML format in the NetworkModels table. An example of a network model is given below:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<neural_network>
     <neuron_groupid="19">
          <name>Learner</name>
          <start_neuron_id>161429</start_neuron_id>
          <width>1</width>
          <length>1</length>
          <location>10,1,10</location>
          <spacing>1</spacing>
          <neuron_type>6</neuron_type>
     </neuron_group>
     <neuron_group id="17">
          <name>Generator</name>
          <start_neuron_id>161427</start_neuron_id>
          <width>1</width>
          <length>1</length>
          <location>1,1,1</location>
          <spacing>1</spacing>
          <neuron_type>6</neuron_type>
     </neuron_group>
</neural_network>
```

Each network model is associated with one or more rows of firing patterns in the NetworkData table, which are also stored in XML format. An example of NetworkData for one time step is given below:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<network_pattern>
     <neuron_group id="17">161427</neuron_group>
     <neuron_group id="18">161428</neuron_group>
</network_pattern>
```

A1.9 Devices

A1.9.1 Introduction

SpikeStream can send and receive spikes across a network to and from an external device, such as a real or virtual robot, camera, etc. This feature is still under development and only the TCP synchronized method has been fully tested between SpikeStream and the SIMNOS virtual robot.

A1.9.2 Sending and Receiving Spike Messages

A number of different methods exist for sending and receiving spike messages across a network. Not all of them have been implemented and the synchronized TCP methods have been most thoroughly tested. The next few sections outline the general procedure for sending and receiving messages. More detail about this can be found in the SpikeStream Simulation code.

Synchronized TCP Network Input

This method uses TCP to send and receive spike packets across the network. This is designed to work with devices that run in their own simulation time, such as the SIMNOS virtual robot (see Section A1.9.4), and it enables the two devices to remain perfectly synchronized. The procedure for receiving this type of message is as follows:

- Wait to receive packet containing the data.
- Unpack the first four bytes, which contain the number of spikes in the message.
- Unpack the spikes, each of which is four bytes long.
- The first byte is the X position of the spike within the layer.
- The second byte is the Y position of the spike within the layer.
- The third and fourth byte contain the time delay of the spike. *WARNING: This is untested for non-zero values and should be set to zero for the moment.*
- When all spikes have been unpacked send a confirmation message containing a single byte to confirm that the data has been received. This has the value SPIKESTREAM_DATA_ACK_MSG (defined in $SPIKESTREAM_ROOT/include/ DeviceMessages.h), which is currently set to 1, but may change.
- Fire neurons in the layer that received spikes from the device.

Since the layer connected to the device will not complete its simulation step until it has updated itself, this method synchronizes SpikeStream with the external device, which should also wait until it receives the acknowledgment message.

Synchronized TCP Network Vision Input

This method is similar to the previous one, except that no delay is included within the packet and the X and Y positions are defined using two bytes. The procedure for receiving this type of message is as follows:

- Wait to receive packet containing the data.
- Unpack the first four bytes, which contain the number of spikes in the message.
- Unpack the spikes, each of which is four bytes long.
- The first two bytes are the X position of the spike within the layer.
- The next two bytes are the Y position of the spike within the layer.
- When all spikes have been unpacked send a message containing a single byte to confirm that the data has been received. This has the value SPIKESTREAM_DATA_ACK_MSG (defined in $SPIKESTREAM_ROOT /include/DeviceMessages.h), which is currently set to 1, but may change.
- Fire neurons in the layer that receive spikes from the device.

Since the layer connected to the device will not complete its simulation step until it has updated itself, this method synchronises SpikeStream with the external device, which should also wait until it has received the acknowledgment message.

Synchronized TCP Network Output

This method sends spikes in a synchronized manner from SpikeStream to an external device. The procedure is as follows:

- Add the number of spikes as a four byte value to the packet.
- Add the spikes to the packet. The first byte is the X position, the second byte is the Y position and the next two bytes are the delay, currently not used.
- Send the packet.
- Wait to receive a packet containing an acknowledgment that the data has been received. This has the value DEVICE_DATA_ACK_MSG (defined in $SPIKESTREAM_ROOT/ include/ DeviceMessages.h), which is currently set to 3, but may change.

Synchronized UDP Network Input

This method creates a loose synchronization between the external device and SpikeStream by timing the interval between spike packets and slowing the simulator down to match. This method only works if the device can slow itself down as well. This method has been implemented on SpikeStream, but it has not been fully tested and some tweaking of the SpikeStream Simulation code may be necessary to get it working properly. The basic approach is as follows:

- The receive method runs as a separate thread which receives the spike messages and unpacks them into a separate buffer.
- The first two bytes of each packet contain the synchronization information. The first 7 bits are the time step count on the external device. This can overflow without problems since it is there to indicate the rate of increase of the time steps in the external device. The remaining bit is a flag to indicate whether the external device was delaying itself on the previous time step.

- The rest of the packet is filled with spikes, with the first byte being the X position, the second byte the Y position and the next two bytes a delay value, which is not currently used.
- When the packet has been unpacked, the receive method calculates the update time per time step for the external device.
- When SpikeStream Simulation completes a simulation step, it sleeps if its own update time per time step is less than that of the external device and if the external device is not delaying itself.
- The SynchronizationDelay table in the Devices database is used to coordinate delay information between independent SpikeStream tasks.

UDP is a potentially lossy method of transmission and the synchronization is also approximate. This makes this approach a useful halfway step between the loss free TCP synchronization and the potentially highly lossy sending and receiving of information to and from a live hardware device, such as a robot, which is interacting with the real world.

Synchronized UDP Network Output

This method is virtually identical to synchronized UDP network input. SpikeStream needs both input and output connections to a device to make this synchronization method work properly.

Asynchronous UDP Network Input/ Output

This method has been designed for using SpikeStream with a live device, but has not yet been implemented. The procedure is something like the following.

- Input spikes are received by a separate thread that unpacks them into a buffer, which is used to fire the neurons at each time step.
- Output spikes are transmitted at the end of each time step.

When it is implemented, the code will be similar to that used for the synchronized UDP input and output, only without the delay.

A1.9.3 Adding Devices

The Devices table in the Devices database contains a list of available devices that SpikeStream can connect to and details about any new devices should be added here. The communication protocol between SpikeStream and the device is determined by the Type field in this table. Definitions of the different device types can be found in $SPIKESTREAM_ROOT/include/DeviceTypes.h. When a device is selected in the Simulation tab, SpikeStream will attempt to connect to it using the information provided. The “Firing Mode” option in the Devices table in the Simulation tab is used to select whether the spikes from the device fire the neuron directly or inject the specified post synaptic potential into the neuron.

A1.9.4 SpikeStream and SIMNOS

Overview

The main external device that has been used and tested with SpikeStream is the SIMNOS virtual robot created by Richard Newcombe, which is shown in Figure A1.17.

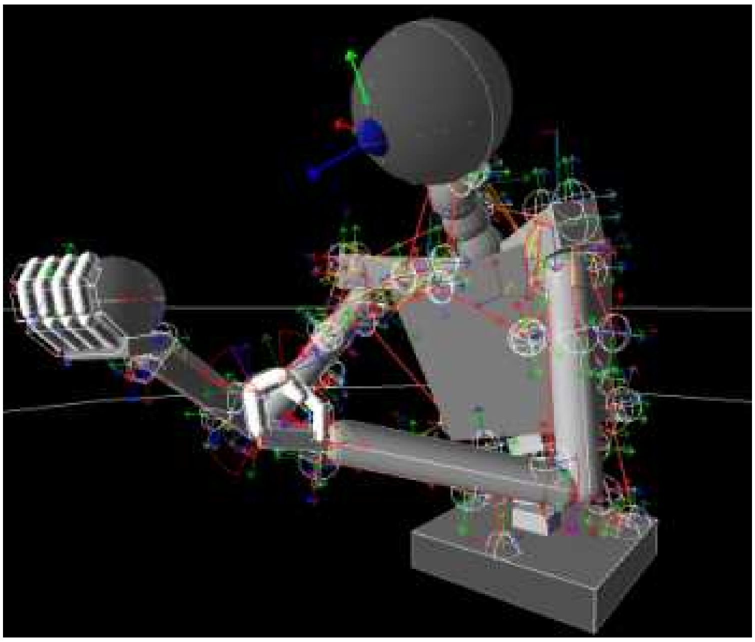
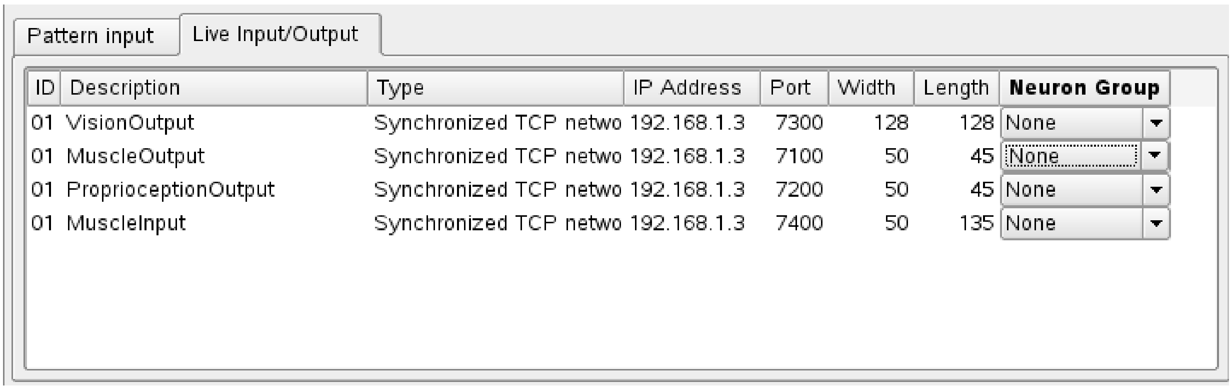


Figure A1.17. SIMNOS virtual robot

SIMNOS is a humanoid anthropomimetic robot whose body is inspired by the human musculoskeletal system. Information about muscle length, joint angles and visual information (available with a wide variety of preprocessing methods) is encoded by SIMNOS into spikes using a selection of methods developed by Newcombe (Gamez et. al. 2006b) and passed across the network to SpikeStream. SIMNOS can also receive muscle length data from SpikeStream in the form of spiking neural events, which are used to control the virtual robot. Together SIMNOS and SpikeStream provide an extremely powerful way of exploring sensory and motor processing and integration. More information about SIMNOS can be found at www.cronosproject.net. SIMNOS will be released soon and anyone interested in using it should contact Richard Newcombe (r.a.newcombe@gmail.com) if they would like a free copy of the current version.

SIMNOS Device Database

The Devices database works a little differently when you are using SIMNOS and SpikeStream together. In this case, the Devices table in the Devices database is created automatically by the SIMNOS spike servers, which enter their information into the Devices and SIMNOSSpikeReceptors tables when they start. To use SIMNOS and SpikeStream you will need to enter the details of the SIMNOS Device database into your spikestream.config file on the main workstation. You will know that you are connecting correctly if you see the four entries in the Devices table shown in Figure A1.18 (the exact entries depend on the configuration of SIMNOS):



ID	Description	Type	IP Address	Port	Width	Length	Neuron Group
01	VisionOutput	Synchronized TCP netwo	192.168.1.3	7300	128	128	None
01	MuscleOutput	Synchronized TCP netwo	192.168.1.3	7100	50	45	None
01	ProprioceptionOutput	Synchronized TCP netwo	192.168.1.3	7200	50	45	None
01	MuscleInput	Synchronized TCP netwo	192.168.1.3	7400	50	135	None

Figure A1.18. SIMNOS device entries

When using SIMNOS, you need to manually create the SynchronizationDelay and SIMNOSReceptors tables in the SIMNOS Devices database by pasting in the appropriate SQL from Devices.sql.

SIMNOS Receptors and Components

Information is exchanged between SIMNOS and SpikeStream in the form of relatively large layers, which connect to layers of equivalent size within the simulator. However, in many cases one wants to connect neuron groups up to part of this incoming information, such as the data from a single arm. It is to solve this kind of problem that the SIMNOS Receptors and Components framework was created. The SIMNOSSpikeReceptors table contains a list of the receptors that are available in SIMNOS, which are associated with a particular device. The SIMNOS Components table consists of lists of receptors, which together constitute a SIMNOS component. These lists of receptor IDs could correspond to the head, neck, arm, part of the visual field or any other abstraction that you want to make of the data from a particular device. Entries in the SIMNOSComponents database have to be created manually by the user and they can then be used to connect a neuron group up to a part of an input or output layer, as explained in the next section.

Using SIMNOSComponents

1. Create a layer that matches the input width and length of the SIMNOS device. For this example we will create a layer to connect to the Muscle Output of SIMNOS, which is currently 50 neurons wide and 45 neurons long. *NOTE: The width varies depends on the spike conversion settings in SIMNOS.*

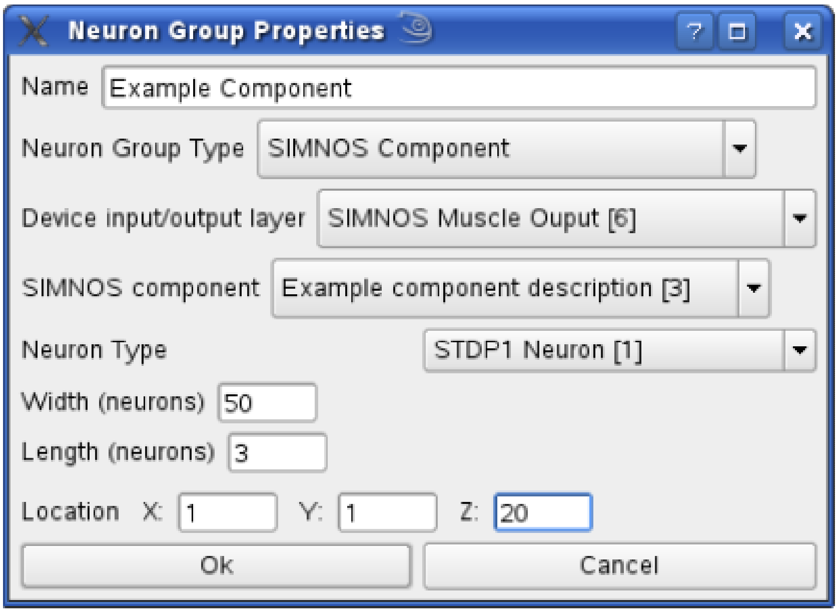
2. Create an entry in the SIMNOSComponents database listing the receptors that you want to connect to in this layer. You need to look in SIMNOSSpikeReceptors table for the receptor IDs, which are associated with a description of the receptor. For example, to connect to the first third and fourth receptor in the SIMNOS muscle output, we need to add an entry as follows: `INSERT INTO SIMNOSComponents (Name, ReceptorIDs, Width, Length) VALUES ("Example component description", "2001,2003,2004", 50, 3);`

3. Click on the “Add Neurons” button to launch the Neuron Group Properties Dialog, enter a name for the layer and select “SIMNOS Component from the “Neuron group type” combo box. The Neuron Group Properties Dialog should look like Figure A1.19.

4. Since there is only one component and one input layer, you don't have any choices in the other combo boxes and you just have to set a location for the new layer.

5. Press “Ok” and you will be presented with a dialog to set the properties for the connection between the device input layer and the component layer that you have just created (see Figure A1.20).

6. When you have set the connection properties, click “Ok” and you should see a new layer with connections to the first third and fourth row of the device muscle output layer (see Figure A1.21).



Name: Example Component

Neuron Group Type: SIMNOS Component

Device input/output layer: SIMNOS Muscle Ouput [6]

SIMNOS component: Example component description [3]

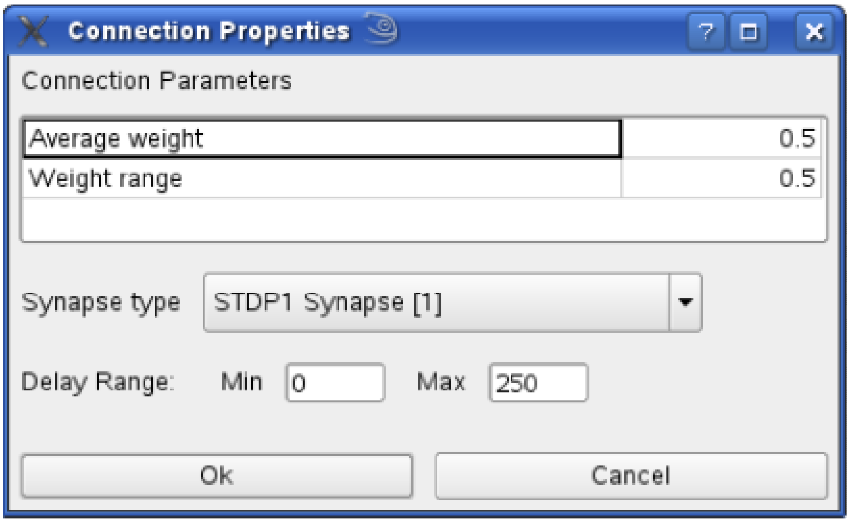
Neuron Type: STDP1 Neuron [1]

Width (neurons): 50

Length (neurons): 3

Location X: 1 Y: 1 Z: 20

Figure A1.19. Creating a SIMNOS component



Connection Parameters

Average weight	0.5
Weight range	0.5

Synapse type: STDP1 Synapse [1]

Delay Range: Min 0 Max 250

Figure A1.20. Setting connection properties for a SIMNOS component

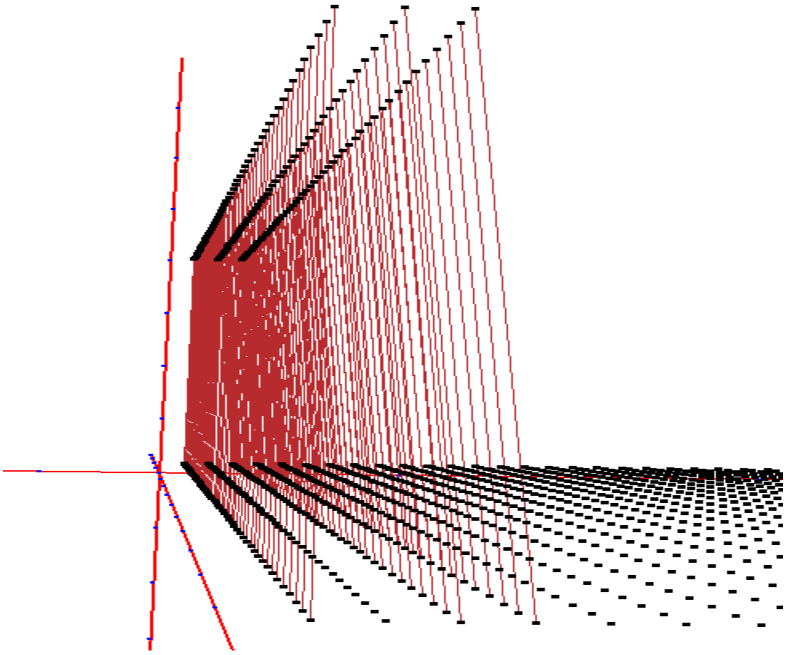


Figure A1.21. SIMNOS component layer connected to device receptors

A1.10 Patterns

A1.10.1 Introduction

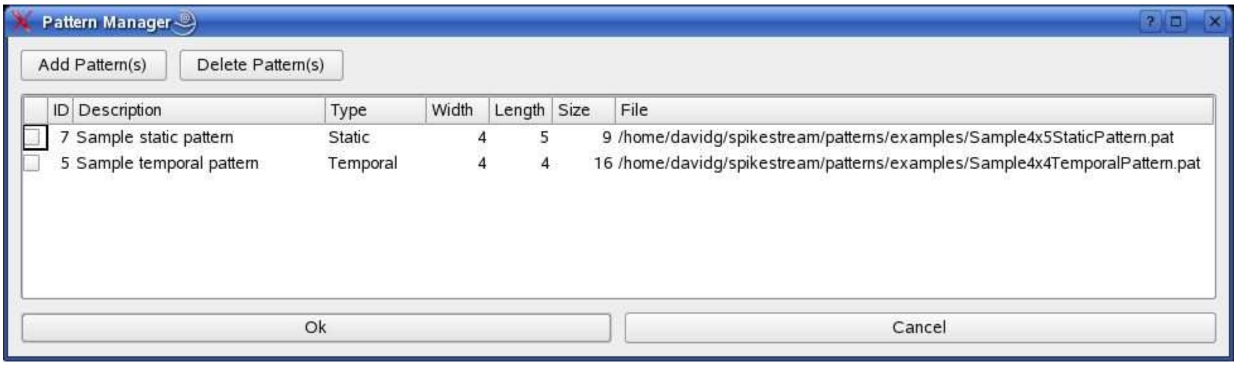
Patterns can be applied to layers in the network for training or testing purposes. Two different types of pattern are available:

- **Static**. A snapshot of a firing pattern in the layer at a single point in time. This pattern will be held for every time step that the pattern is held.
- **Temporal**. The pattern codes a firing pattern that is spread out over several time steps. Each neuron will only be fired once at its specified time.

A1.10.2 Adding Patterns

Pattern Manager

The Pattern Manager (see Figure A1.22) is used to load patterns from a file into the SpikeStream database. Click on Tools->Pattern manager to launch the Pattern Manager, which will display a list of patterns currently stored in the database. Patterns can be deleted by checking their associated box and clicking the “Delete Pattern(s)” button. To load a pattern into the database from a file, click on “Add Pattern(s)”, navigate to the file(s) that you want to add and then click “Ok”. If the pattern file(s) loads up successfully you will see the new pattern(s) listed in the Pattern Manager. Instructions for creating pattern files are given in the next section.



	ID	Description	Type	Width	Length	Size	File
□	7	Sample static pattern	Static	4	5	9	/home/davidg/spikestream/patterns/examples/Sample4x5StaticPattern.pat
□	5	Sample temporal pattern	Temporal	4	4	16	/home/davidg/spikestream/patterns/examples/Sample4x4TemporalPattern.pat

Figure A1.22. Pattern Manager

Pattern Files

The easiest way to create patterns is to manually or programatically generate pattern files and load them into the database using the Pattern Manager. The format is as follows.

- **First lines.** Can contain any information you wish, such as comments, authorship, etc., but must not contain hashes. All lines will be skipped by the parser until the information about the pattern is reached.

- **# Type.** The type of the pattern. This line should either be “# Type: static” or “# Type: temporal”.

- **# Width.** The width of the pattern, for example “# Width: 4”.

- **# Height.** The height of the pattern, for example “# Height: 4”.

- **# Description**. A short description of the pattern that will be added to the pattern database, for example “# Description: Sample static pattern”.

- **# Pattern data**. After the information about the pattern, the file can contain one or more pieces of pattern data. After each "#Pattern data:" heading there should be a width x height matrix of numbers, separated by spaces, containing the pattern at that point in time. For static patterns, these numbers must be either 1 or 0. For temporal patterns, they must be between 0 and 250 (currently the maximum number of time steps). The numbers in temporal patterns code the time that the neuron will be fired after the pattern has been loaded. For example, if you create a pattern containing a number of fives and set the "Number of time steps per pattern" in the Simulation tab to ten, then five time steps after the pattern was loaded, the neurons corresponding to the fives in the pattern will be fired and after another five time steps, the next pattern will be loaded. All of this will become much clearer when you try out the static and temporal sample pattern files given in SPIKESTREAM_ROOT/patterns/examples

NOTE: If your pattern does not behave as expected, make sure that you have the static / temporal field set correctly for your pattern.

Direct Pattern Generation

Whilst the automatic generation of pattern files is probably the easiest way to generate patterns, it is also possible to directly add patterns directly to the Patterns database without using the Pattern Manager. In this case, you need to generate a pattern description and one or more rows of pattern data. When you have added a couple of test patterns to the database using the Pattern Manager, a look at the structure of the data will show you how to directly generate your own patterns.

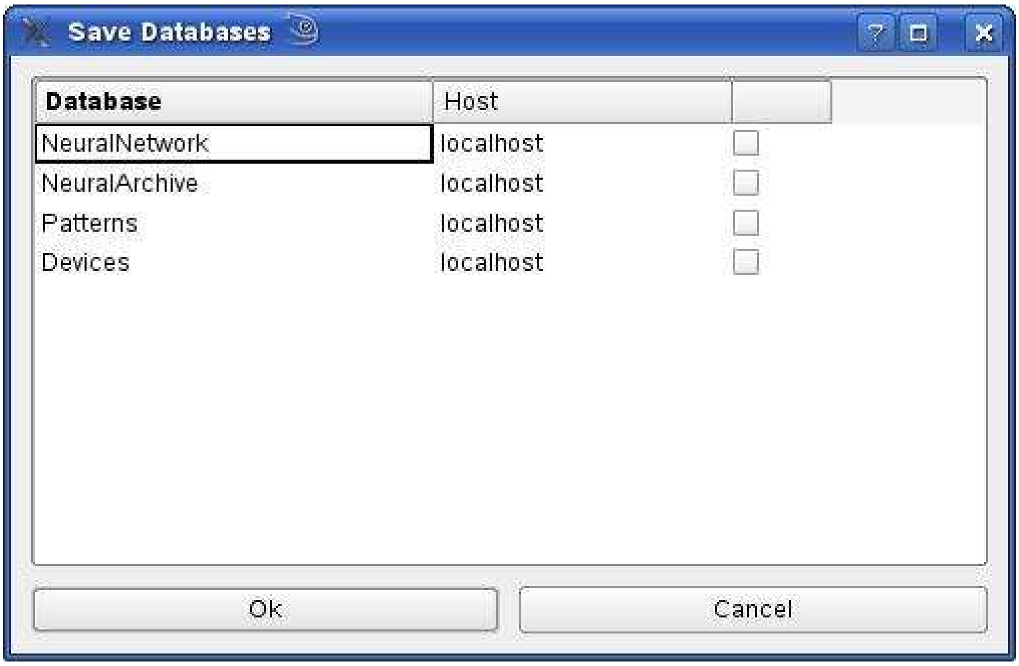
A1.11 Saving and Loading Databases

A1.11.1 Introduction

SpikeStream Application directly edits the database and so there is no need to explicitly save anything when you close it apart from any weights that have been changed during a simulation run. To enable users to save and load different neural networks, SpikeStream can save its databases to a file and reload them at a later point in time.

A1.11.2 Saving Databases

Click on "File->Save database" and you will be prompted to choose the file to save the databases into. When the file is selected you will be presented with the Database Dialog shown in Figure A1.23. This enables you to select which of the databases you want to save – for example, you may only want to save the NeuralNetwork database into the file and leave out the Neural Archive, Patterns and Devices databases. When you have checked the databases that you want to save, press "Ok" and they will be saved into the specified file. Saving and loading of databases is carried out by the SaveSpikeStreamDatabase and LoadSpikeStreamDatabase scripts, which use the mysqldump program.



Database	Host	
NeuralNetwork	localhost	□
NeuralArchive	localhost	□
Patterns	localhost	□
Devices	localhost	□

Figure A1.23 Database Dialog

This operation stores everything associated with each database - for example, neuron types, synapse types, global and noise parameters are all saved when the Neural Network database is saved.

A1.11.3 Loading Databases

Databases can only be loaded when the simulation is *not* initialized and an archive is not currently being played back. The loading of databases follows the reverse procedure to the saving of databases. Click on “File->Load databases”. This will first warn you that the loading operation will overwrite any of the databases that you choose to load. If you want to keep the current database you should cancel the loading operation and save the current database in a separate file. When you are ready to load the database, click “Yes” on this warning and use the file dialog to select the database that you want to load. SpikeStream will then inspect this file to determine which databases are stored inside it and present you with a Database Dialog containing a list of the databases that are available in the file. Select the databases that you want to load and click ok.

IMPORTANT NOTE: In the present implementation, the adding and removing of neuron and synapse types must be done without SpikeStream running. Loading up a database with different neuron and synapse classes from the ones currently loaded will lead to errors. The database should be ok, but you will need to restart SpikeStream to resolve the problem.

A1.11.4 Clear Databases

The databases can only be cleared when the simulation is *not* initialized and an archive is not currently being played back. Clicking on "File->Clear databases" resets all data in the databases except the neuron, synapse and probe types. This operation is not reversible, so make sure that you do not have any important information or saved simulation runs that you want to keep before pressing "Yes" when the confirm dialog is displayed. If you want to reset everything back to its default state including the neuron, synapse and probe types, use the load database feature (Section A1.11.3) to load the file $SPIKESTREAM_ROOT/database/DefaultDatabase.sql.tar.gz. The CreateSpikeStreamDatabases script can also be used to reset all the databases.

A1.11.5 Import Connection Matrix

This feature is at an early stage of development and it is used to create a neuron group and set of connections based on a connection matrix in which the x and the y axes are the neuron IDs and the values are the weights. After you have clicked "File->Import connection matrix" and selected the file containing the connection matrix it will create the new layer at (0, 0, 0) using the default neuron and synapse types. Before running this function you will need to create enough space at (0, 0, 0) for the new layer.

A1.12 Neuron and Synapse Classes

A1.12.1 Introduction

The dynamic class loading features of SpikeStream make it relatively easy to change the neuron and synapse models without modifying the whole application. However, a certain amount of work is required to get a new neuron or synapse class recognized by SpikeStream so that it can run in a distributed manner.

IMPORTANT NOTE: Adding and removing synapse classes should be done without SpikeStream running or you will get errors from the Neuron and Synapse parameters dialogs, which only load up the Neuron and Synapse type information once during initialization of SpikeStream. Errors can also occur when you load a database with different neuron and synapse types or with a different TypeID from the existing types. Restarting SpikeStream usually resolves the problem.

A1.12.2 Creating Neuron and Synapse Classes

Extend the Neuron or Synapse Class

The first stage is to write the code for the neuron or synapse classes, which should extend the Neuron or Synapse classes in $SPIKESTREAM_ROOT/spikestreamsimulation/src. More information about these classes can be found in the online source documentation, available on the project website http://spikestream.sourceforge.net/pages/documentation.html

The easiest place to start when writing your own neurons or synapses is to look at STDP1Neuron and STDP1Synapse and tweak these to match your own neuron or synapse model or learning rule. These examples also illustrate some of the areas that need to be handled carefully by a neuron or synapse class. The methods that you need to extend are now covered.

Synapse.h

- **virtual const string* getDescription() = 0;** Returns a descriptive name for the synapse, which can be useful for debugging class loading. The class that invokes this method is responsible for cleaning up the string.
- **virtual short getShortWeight() = 0;** Returns the weight as a short between MIN_SHORT_WEIGHT and MAX_SHORT_WEIGHT (defined in Synapse.h). This is a virtual method because some implementations may need the state of the weight to be calculated retrospectively.
- **virtual double getWeight() = 0;** Returns the weight as a double between MIN_DOUBLE_WEIGHT and MAX_DOUBLE_WEIGHT. This is a virtual method because some implementations may need the state of the weight to be calculated retrospectively.
- **virtual bool parametersChanged() = 0;** Called when the parameters of the synapse have changed. The parameters of the synapses are held as references to parameter maps and when these are reloaded this method is called.
- **virtual void processSpike() = 0;** Called when a spike is routed to this synapse. In event-based simulation the synapse should be updated by this method.
- **virtual void calculateFinalState() = 0;** Called to update synapse class when all synapses are being updated at each time step. This method is never called during event based simulation. In this mode, the synapse class is only updated whenever it processes a spike.
- **virtual string getMonitoringInfo();** This method returns a string containing an XML description of the variables that are available for monitoring within this class. Overload this method and getMonitoringData() if you want to send monitoring information back to the main application. This will enable you to view a graph of the weight, for example, as described in Section A1.7.5.

- **virtual MonitorData* getMonitoringData();** Returns a monitor data struct (defined in GlobalVariables.h) containing the data that is being monitored. This returned data must match that defined in the string returned by getMonitoringInfo();

Neuron.h

- **virtual void calculateFinalState() = 0;** Calculates the final state of the neuron after all spikes have been received. In synchronous simulation mode all neurons have this method called on them at the end of each simulation step.
- **virtual void changePostSynapticPotential(double amount, unsigned int preSynapticNeuronID) = 0;** This method is called when a synapse changes the membrane potential of the neuron. The neuron should update itself when this method is called by calling calculateFinalState().
- **virtual const string* getDescription() = 0;** Returns a description of this neuron class for debugging only. Destruction of the new string is the responsibility of the invoking method.
- **virtual bool setParameters(map<string, double> paramMap) = 0;** Sets the parameters of the neuron. These should be defined in their own database, whose name is listed in the NeuronTypes database. This method is called on only one instance of the neuron class with the parameters being set and held statically. The parametersChanged() method is called after the static setting of the parameters to inform each neuron class that the parameters have changed.
- **virtual void parametersChanged() = 0;** Called after the parameters have been statically changed to inform each neuron class that the parameters have been changed. This enables them to update their weights, for example, after learning has been switched off.
- **virtual string getMonitoringInfo();** This method returns a string containing an XML description of the variables that are available for monitoring within this class. Overload

this method and getMonitoringData() if you want to send monitoring information back to the main application. This will enable you to view a graph of the membrane potential, for example, as shown in Section A1.7.5.

- **virtual MonitorData* getMonitoringData();** Returns a monitor data struct (defined in GlobalVariables.h) containing the data that is being monitored. This returned data must match that defined in the string returned by getMonitoringInfo();

A1.12.3 Build and Install Library

When you have created your neuron and synapse classes, compile them as .so libraries and copy them to $SPIKESTREAM_ROOT/lib. They need to have the standard library name format, such as libstdp1neuron.so for a "stdp1neuron" library. More information about this procedure can be found at: http://www.linux.org/docs/ldp/howto/Program-Library-HOWTO/shared-libraries.html. When your neuron class calls methods that are unique to the synapse class – i.e. methods that are not present in Synapse.h – you need to link against the synapse library to build the neuron class. This can be done by passing information about the dynamic synapse library to gcc when you build the neuron class. However, to *run* a simulation using the neuron class, the dynamic library that you have linked against needs to be accessible by the operating system in one of the known locations.[1] This can be done in one of three ways, which have to be carried out on every machine that you run the simulation on.

Method 1: Change the LD_LIBRARY_PATH Environment Variable

One way to ensure that the operating system can find the dynamic libraries is to add the location of your neuron and synapse libraries to the system path. This can be done by adding the following line to your .bashrc file:

[1] This step could probably be avoided by linking the neuron or synapse class against a static version of the other neuron or synapse class. However, I have not tried this yet and it is probably more memory efficient to use a dynamic library.

```
 LD_LIBRARY_PATH=$LD_LIBRARY_PATH:${SPIKESTREAM_ROOT}/lib
```

This can work fine if you are running SpikeStream on a single workstation, but it is likely to cause problems running across multiple machines and is not recommended anyway.

Method 2: Add Links to Library in /usr/local/lib

This method creates a link from /usr/local/lib to the location of your libraries. For example, to install STDP1Synapse, change to /usr/local/lib, log in as root and create the links using the following command:

```
      ln -s /home/davidg/spikestream/lib/libstdp1synapse.so
 libstdp1synapse.so.1
```

This may have to be done using the full address of the library if SPIKESTREAM_ROOT has only been defined for the user shell. The advantage of this approach is that it makes it easy to update the libraries when developing the neuron and synapse classes and it is more portable across systems. This approach is implemented by the InstallSpikeStream script, which is used to install the neuron and synapse classes included in the SpikeStream distribution (see Section A1.2.4).

IMPORTANT NOTE: You should only install links to these libraries as root if you are the sole user of SpikeStream on the system. Otherwise you may end up dynamically loading another user's libraries!

Method 3: Copy Library to /user/local/lib

If your dynamic libraries are rarely going to change, it makes more sense to install them permanently by copying them to /usr/local/lib, rather than linking from /usr/local/lib to somewhere else on the system. This approach only makes sense if the other parts of SpikeStream were installed in /usr/local/bin as well. Since SpikeStream is still in the process of development, this option is not recommended at this stage.

IMPORTANT NOTE: You should only install these libraries as root if you are the sole user of SpikeStream on the system. Otherwise you may end up dynamically loading another user's libraries!

A1.12.4 Update Database

The final stage is to add appropriate entries and tables to the Neural Network database so that networks can be created and simulated using the new neuron classes. This involves updating the neuron and synapse types and adding tables for the neuron and synapse parameters. In these examples, the neuron and synapse classes will be called Example Neuron and Example Synapse.

Add Neuron and Synapse Types

The NeuronTypes and SynapseTypes tables in the NeuralNetwork database hold information about all of the available neuron and synapse types. To use your new neuron and synapse classes in SpikeStream, they must have an entry in these tables. Before adding a new neuron type, select a TypeID. This is a unique identifier for your neuron type which must not conflict with any of the existing types. In this example, I have selected a TypeID of 2 since the only neuron class currently in the database is an STDP1Neuron with a TypeID of 1. To add a new neuron type, use the following SQL:

```
USE NeuralNetwork;

INSERT INTO NeuronTypes(TypeID, Description, ParameterTableName,
ClassLibrary) VALUES (1, "Example Neuron", "ExampleNeuronParameters",
"libexampleneuron.so");
```

The SQL for adding a new synapse type is similar:

```
USE NeuralNetwork;

INSERT INTO SynapseTypes(TypeID, Description, ParameterTableName,
ClassLibrary) VALUES (1, "Example Synapse", "ExampleSynapseParameters",
"libexamplesynapse.so");
```

Add Parameter Tables

Each neuron and synapse class has an associated parameter table in which the parameters for the neuron or synapse model can be set individually for each neuron or connection group, which have entries in the appropriate table. In order for this to work, the parameter table has be set up in a specific fashion. The SQL for the STDP1Neuron and STDP1Synapse parameter tables is given below:

```
USE NeuralNetwork;

CREATE TABLE STDP1NeuronParameters (
NeuronGrpID SMALLINT UNSIGNED NOT NULL,
CalciumIncreaseAmnt_val DOUBLE DEFAULT 1.0,
CalciumIncreaseAmnt_desc CHAR(100) DEFAULT "Calcium increase amount",
CalciumDecayRate_val DOUBLE DEFAULT 60.0,
CalciumDecayRate_desc CHAR(100) DEFAULT "Calcium decay rate",
RefractoryPeriod_val DOUBLE DEFAULT 1.0,
RefractoryPeriod_desc CHAR(100) DEFAULT "Refractory period (ms)",
MembraneTimeConstant_val DOUBLE DEFAULT 3.0,
MembraneTimeConstant_desc CHAR(100) DEFAULT "Membrane time constant (ms)",
RefractoryParamM_val DOUBLE DEFAULT 0.8,
RefractoryParamM_desc CHAR(100) DEFAULT "Refractory parameter M",
RefractoryParamN_val DOUBLE DEFAULT 3.0,
RefractoryParamN_desc CHAR(100) DEFAULT "Refractory parameter N",
Threshold_val DOUBLE DEFAULT 1.0,
Threshold_desc CHAR(100) DEFAULT "Threshold",
Learning_val BOOLEAN DEFAULT 0,
Learning_desc CHAR(100) DEFAULT "Learning",
PRIMARY KEY (NeuronGrpID));
```

```
CREATE TABLE STDP1SynapseParameters (
ConnGrpID SMALLINT UNSIGNED NOT NULL,
Learning_val BOOLEAN DEFAULT 0,
Learning_desc CHAR(100) DEFAULT "Learning",
Disable_val BOOLEAN DEFAULT 0,
Disable_desc CHAR(100) DEFAULT "Disable",
CalciumThreshUpLow_val DOUBLE DEFAULT 30.0,
CalciumThreshUpLow_desc CHAR(100) DEFAULT "Calcium threshold up low",
CalciumThreshUpHigh_val DOUBLE DEFAULT 120.0,
CalciumThreshUpHigh_desc CHAR(100) DEFAULT "Calcium threshold up high",
CalciumThreshDownLow_val DOUBLE DEFAULT 30.0,
CalciumThreshDownLow_desc CHAR(100) DEFAULT "Calcium threshold down low",
CalciumThreshDownHigh_val DOUBLE DEFAULT 40.0,
CalciumThreshDownHigh_desc CHAR(100) DEFAULT "Calcium threshold down high",
WeightChangeThreshold_val DOUBLE DEFAULT 0.8,
WeightChangeThreshold_desc CHAR(100) DEFAULT "Weight change threshold",
WeightIncreaseAmnt_val DOUBLE DEFAULT 0.1,
WeightIncreaseAmnt_desc CHAR(100) DEFAULT "Weight increase amount",
WeightDecreaseAmnt_val DOUBLE DEFAULT 0.1,
WeightDecreaseAmnt_desc CHAR(100) DEFAULT "Weight decrease amount",
PRIMARY KEY (ConnGrpID));
```

As you can see from the examples, each parameter table has a neuron or connection group ID as its primary key. The parameters themselves can either be boolean, which appears as a check box in the parameter dialog, or doubles. Each value is defined using ExampleName_val, which stores the value of the parameter and has the specified default, and ExampleName_desc, whose default is the description of the value. As long as these conventions are adhered to in your parameter tables, you should be able to set the parameters using the Neuron Parameters Dialog and Synapse Parameters Dialog and the simulation should be able to access them without problems.

Appendix 2
Network Analyzer

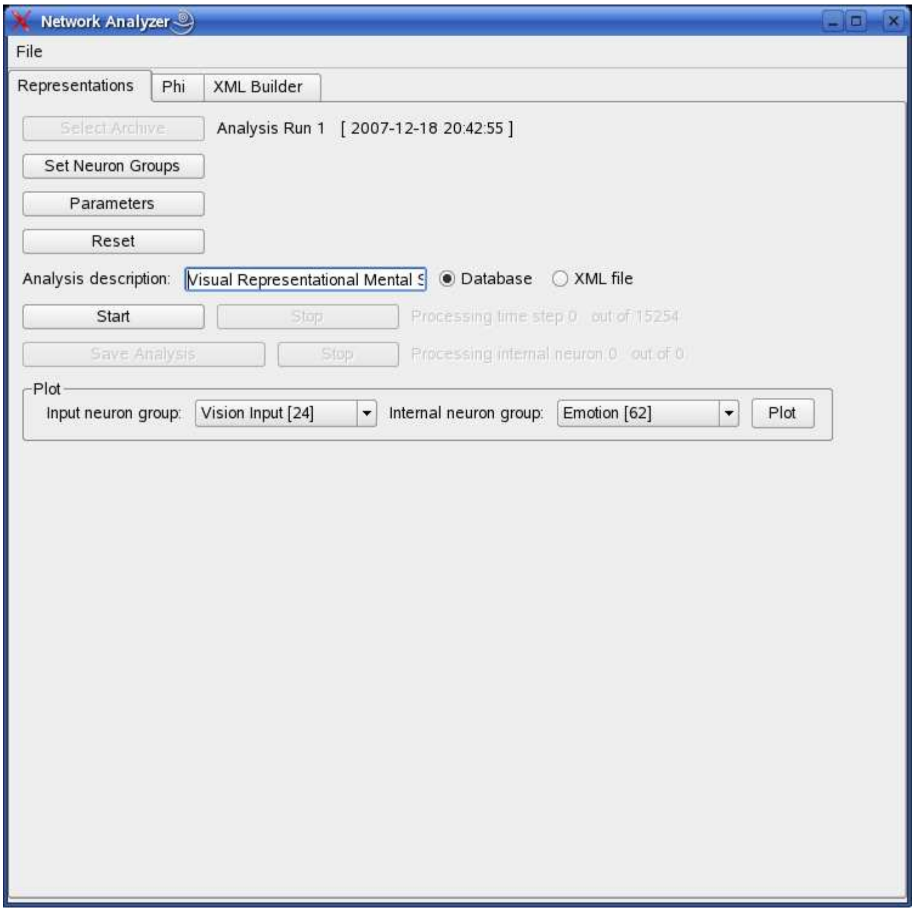
A2.1 Introduction

This appendix gives a brief overview of the Network Analyzer software, which has approximately 10,000 source lines of code[1] and was used for the analysis work in this thesis. There has been no formal release of Network Analyzer, but the source code is included in the Supplementary Materials. A brief overview of the main features of this software now follows.

A2.2 Representation Analyzer

Representation Analyzer identifies representational mental states in the network using the method set out in Section 7.3.3. It includes 2D and 3D plotting tools to display the mutual information between neurons at different steps back in time.

[1] Calculated using Wheeler's SLOCCount software. More information about Wheeler's measure can be found at: http://www.dwheeler.com/sloc/.



Analysis Run 1 [2007-12-18 20:42:55]

Analysis description: Visual Representational Mental S ● Database ○ XML file

Processing time step 0 out of 15254

Processing internal neuron 0 out of 0

Input neuron group: Vision Input [24]

Internal neuron group: Emotion [62]

Figure A2.1. Representation Analyzer

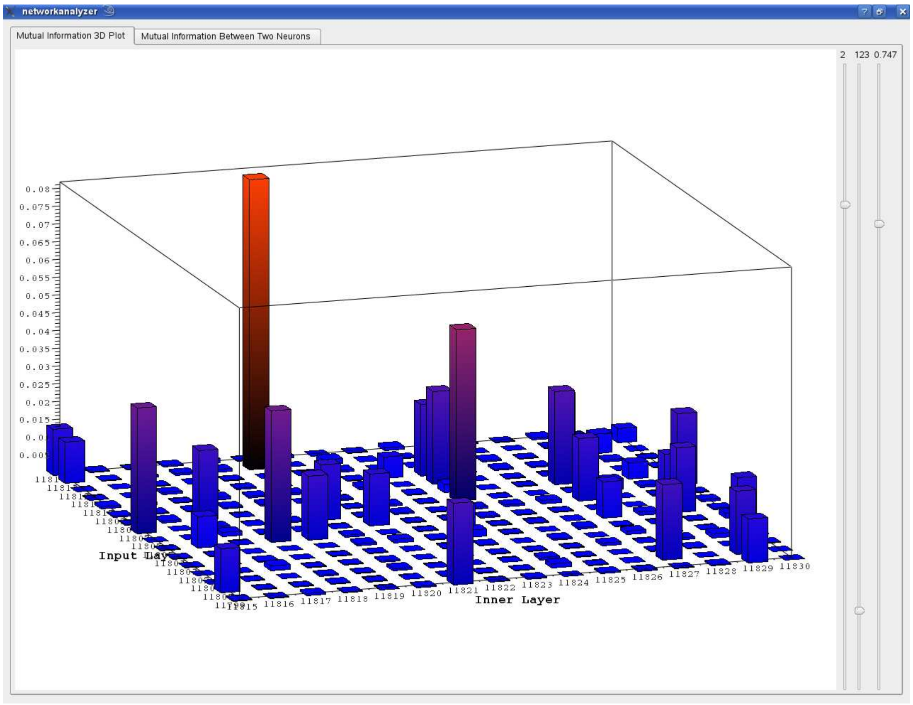


Figure A2.2. 3D mutual information plotter

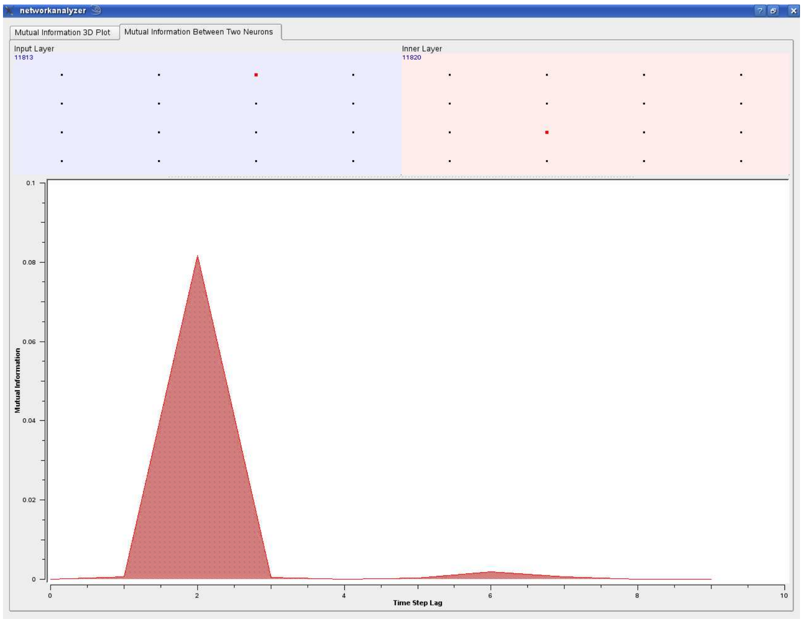
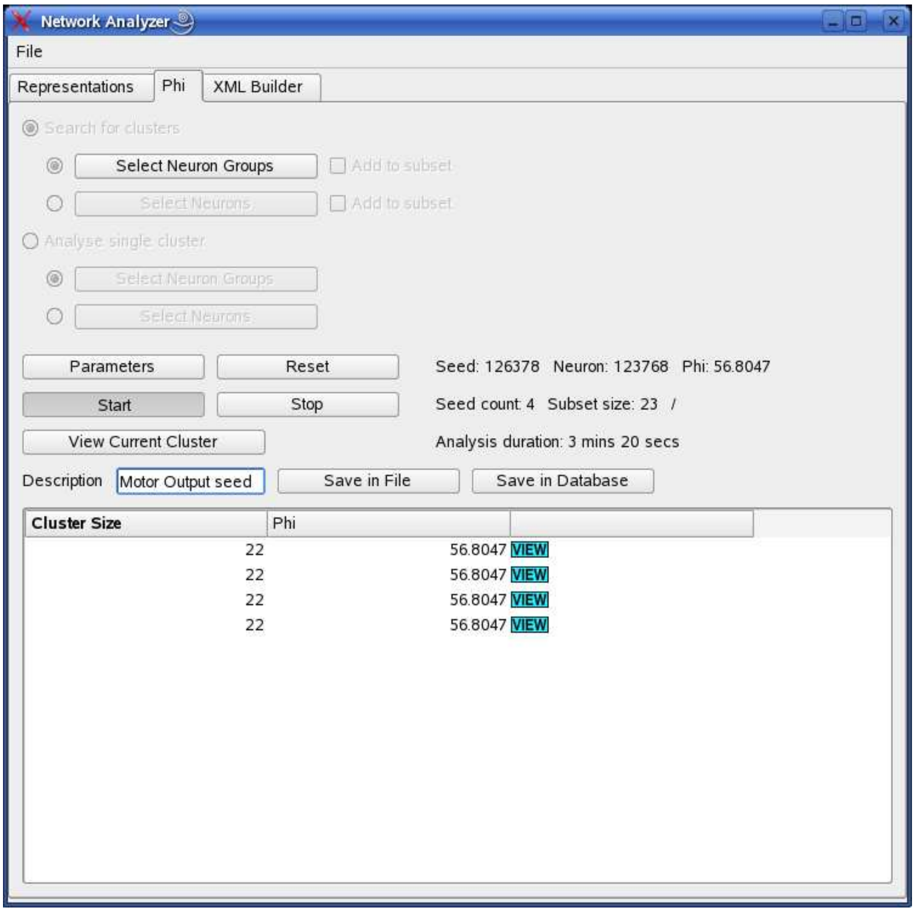


Figure A2.3. 2D mutual information plotter

A2.3 Phi Analyzer

Phi Analyzer identifies the complexes in the network using the method described in Section 7.4.2. The neuron IDs in a current subset or complex can be viewed and used to highlight the SpikeStream network.



Seed: 126378 Neuron: 123768 Phi: 56.8047

Seed count: 4 Subset size: 23 /

Analysis duration: 3 mins 20 secs

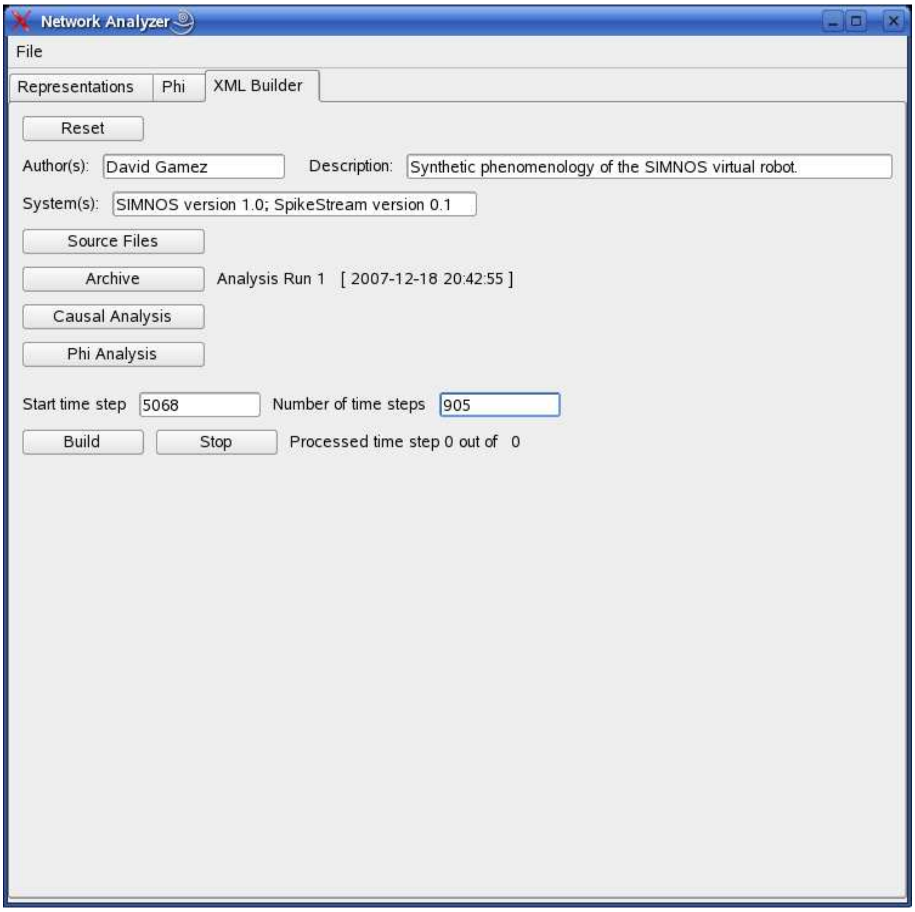
Description: Motor Output seed

Cluster Size	Phi	
22	56.8047	VIEW
22	56.8047	VIEW
22	56.8047	VIEW
22	56.8047	VIEW

Figure A2.4. Phi Analyzer

A2.4 XML Builder

XML Builder was used to construct the final sequence of XML files that describe the predicted phenomenology of the network.



Author(s): David Gamez

Description: Synthetic phenomenology of the SIMNOS virtual robot.

System(s): SIMNOS version 1.0; SpikeStream version 0.1

Analysis Run 1 [2007-12-18 20:42:55]

Start time step 5068

Number of time steps 905

Processed time step 0 out of 0

Figure A2.5. XML Builder

APPENDIX 3
SEED AND GROUP ANALYSES

A3.1 Introduction

This appendix presents the detailed results from the seed and group information integration analyses.

A3.2 Complexes Found using Seed Expansion Method

This section presents the complexes that were found in the network using the seed expansion method. All of these results were brought together in the general discussion of the information integration of the network in Section 7.4.6. To present the results as clearly as possible the neuron groups in the figures are labelled using the IDs in Table A3.1, which correspond to the IDs that were used for these neuron groups in the database. The full results are included as XML files in the Supporting Materials.

ID	Neuron Group
24	Vision Input
28	Red Sensorimotor
29	Blue Sensorimotor
62	Emotion
34	Inhibition
61	Motor Cortex
60	Motor Integration
54	Eye Pan
55	Eye Tilt
53	Motor Output

Table A3.1. Neuron group IDs

A3.2.1 Vision Input

Since this layer contained over 8,000 neurons, it was decided to start with a maximum subset size of 50. All of the seeds in this neuron group expanded to small complexes of approximately 30 neurons with Φ ranging from 75-91. Most of the neurons in these complexes were in Inhibition, as shown in Figure A3.1. The analysis took 4.5 days.

Parameter	Value
Max number of bipartitions per level	5
Percentage of bipartition levels	50
Expansion rate per connection group	1
Maximum subset size	50
Maximum number of consecutive expansion failures per connection group	5
Only examine equal bipartitions	false

Table A3.2. Parameters for seed-based Vision Input analysis

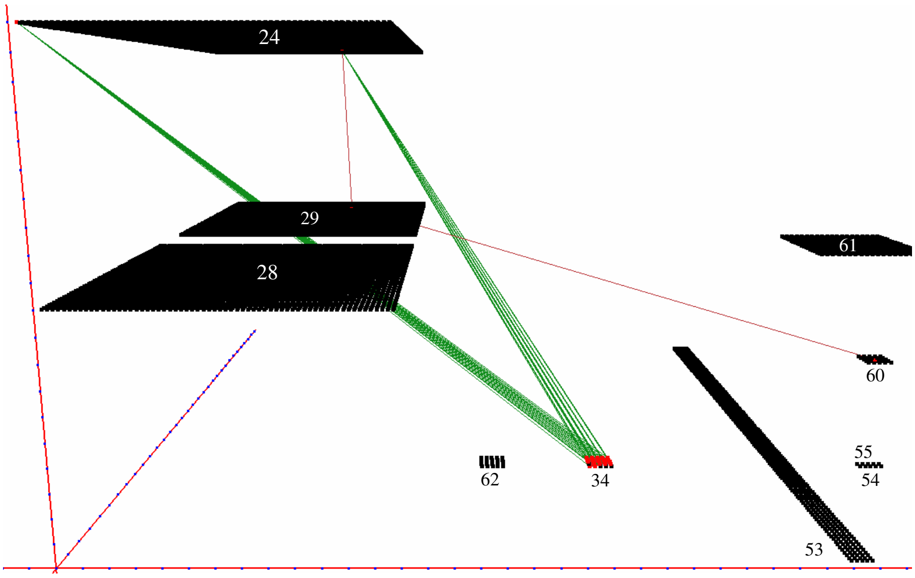


Figure A3.1. Typical complex found during expansion of seeds in Vision Input

A3.2.2 Blue Sensorimotor

This was a large layer with over 4,000 neurons, and so it was decided to start with a maximum subset size of 50. About 2500 of the seeds in this layer expanded into small complexes with Φ ranging from 26-93. Most of the neurons in these complexes were in Inhibition, as shown in Figure A3.2. The analysis took 2 days.

Parameter	**Value**
Max number of bipartitions per level	5
Percentage of bipartition levels	50
Expansion rate per connection group	1
Maximum subset size	50
Maximum number of consecutive expansion failures per connection group	5
Only examine equal bipartitions	false

Table A3.3. Parameters for seed-based Blue Sensorimotor analysis

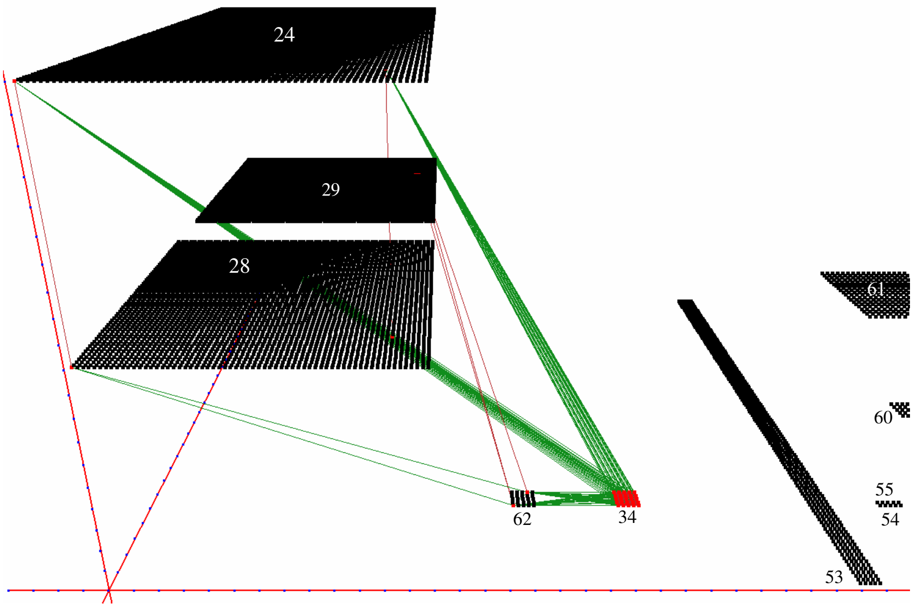


Figure A3.2. Typical complex found during seed-based Blue Sensorimotor analysis

A3.2.3 Red Sensorimotor

This was a large layer with over 4,000 neurons, and so it was decided to start with a maximum subset size of 50. About 3200 of the seeds in this layer expanded into small complexes with Φ ranging from 26-93. Most of the neurons in these complexes were in Inhibition, as shown in Figure A3.3. The analysis took 2.5 days.

Parameter	Value
Max number of bipartitions per level	5
Percentage of bipartition levels	50
Expansion rate per connection group	1
Maximum subset size	50
Maximum number of consecutive expansion failures per connection group	5
Only examine equal bipartitions	false

Table A3.4. Parameters for seed-based Red Sensorimotor analysis

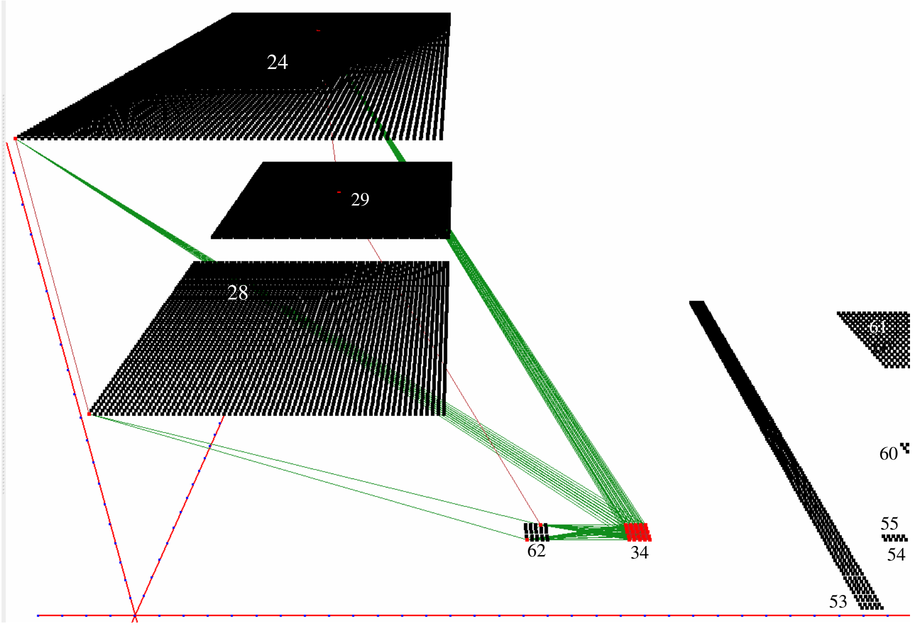


Figure A3.3. Typical complex found during seed-based Red Sensorimotor analysis

A3.2.4 Inhibition

The seeds in Inhibition expanded their connections with Vision Input into a subset that had a relatively low Φ of about 6 (see Figure A3.4). Each expansion increased the Φ value by a small amount, but since there were 8192 connections between each neuron in Inhibition and Vision Input, all of the subsets expanded beyond the maximum subset size of 150. After eleven seeds had been expanded without a complex being found, the expansion rate was changed to 10 to speed up the analysis, which took 3.5 days.

Parameter	Value
Max number of bipartitions per level	25
Percentage of bipartition levels	100
Expansion rate per connection group	10
Maximum subset size	150
Maximum number of consecutive expansion failures per connection group	10
Only examine equal bipartitions	false

Table A3.5. Parameters for seed-based Inhibition analysis

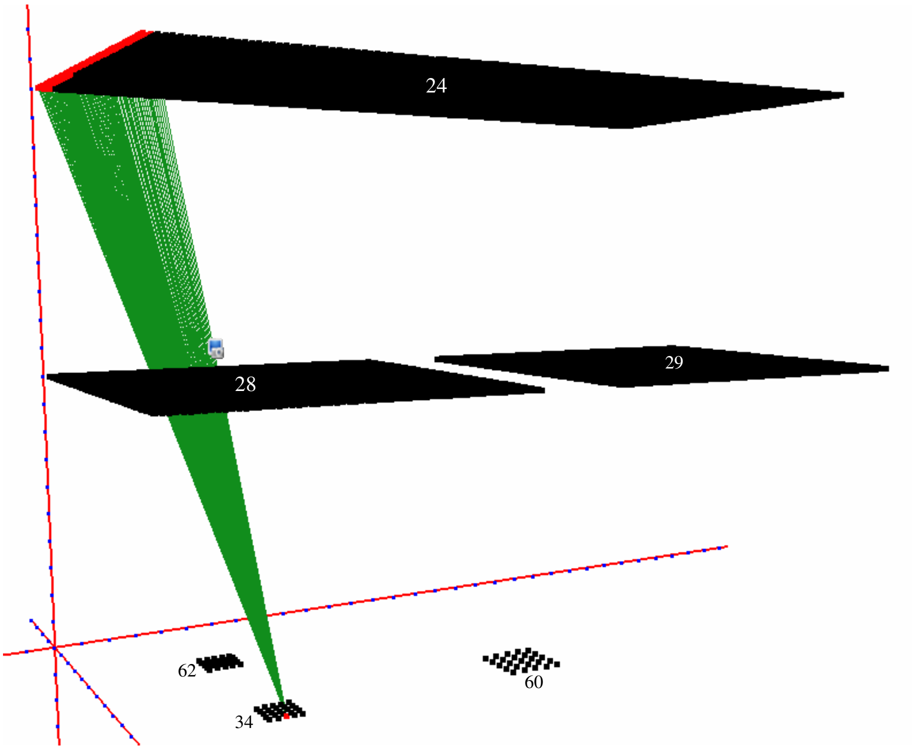


Figure A3.4. Subset during seed-based Inhibition analysis

A3.2.5 Motor Output

Most of the seeds in this layer expanded into a complex with 23 neurons and Φ = 56.8 that included most of Inhibition (see Figure A3.5). A number of seeds also expanded into complexes with 71-91 neurons that included a number of different neuron groups and had Φ ranging from 80-103. One of these turned out to be the main complex, which is shown in Figure A3.6. Only one seed expanded beyond the maximum subset size of 150 neurons. The analysis took 7.5 days.

Parameter	**Value**
Max number of bipartitions per level	25
Percentage of bipartition levels	100
Expansion rate per connection group	1
Maximum subset size	150
Maximum number of consecutive expansion failures per connection group	10
Only examine equal bipartitions	false

Table A3.6. Parameters for seed-based Motor Output analysis

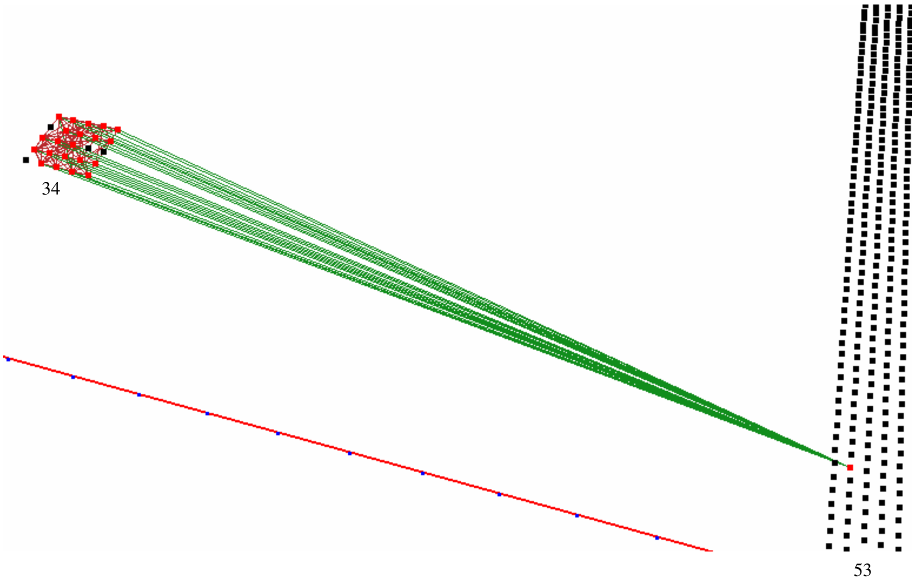


Figure A3.5. Typical small complex found during seed-based Motor Output analysis

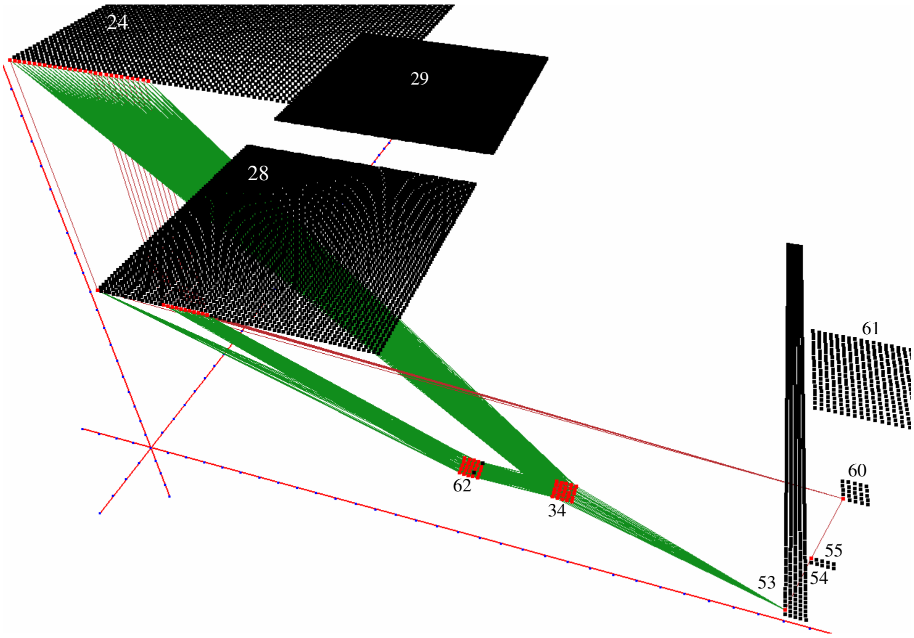


Figure A3.6. Larger complex found during seed-based Motor Output analysis. This is the main complex of the network.

A3.2.6 Eye Pan

One of the seeds in this layer expanded to more than 150 neurons and three seeds expanded to complexes with 12 neurons and Φ = 4.7, an example of which is shown in Figure A3.7. The fifth seed expanded to a complex with 77 neurons and Φ = 59.2, which included neurons from a number of different groups including Inhibition. The analysis took 4 days.

Parameter	**Value**
Max number of bipartitions per level	50
Percentage of bipartition levels	100
Expansion rate per connection group	1
Maximum subset size	150
Maximum number of consecutive expansion failures per connection group	10
Only examine equal bipartitions	false

Table A3.7. Parameters for seed-based Eye Pan analysis

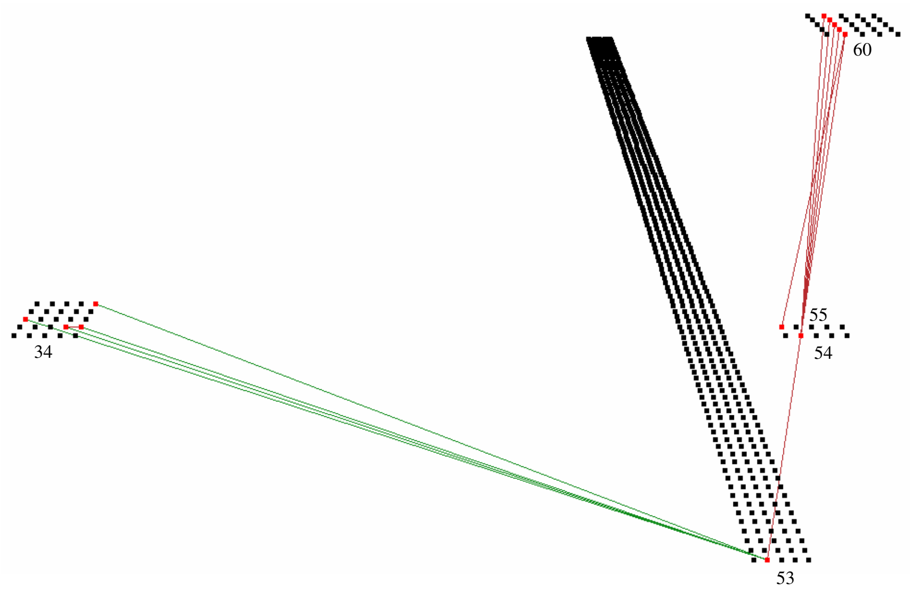


Figure A3.7. Typical small complex found during seed-based Eye Pan analysis

A3.2.7 Eye Tilt

One of the seeds in this layer expanded into a complex with 69 neurons and Φ = 46, which is shown in Figure A3.8. The other four seeds expanded into complexes with 12 neurons and Φ = 4.7, an example of which is shown in Figure A3.9. The analysis took 9 hours.

Parameter	**Value**
Max number of bipartitions per level	50
Percentage of bipartition levels	100
Expansion rate per connection group	1
Maximum subset size	150
Maximum number of consecutive expansion failures per connection group	10
Only examine equal bipartitions	false

Table A3.8. Parameters for seed-based Eye Tilt analysis

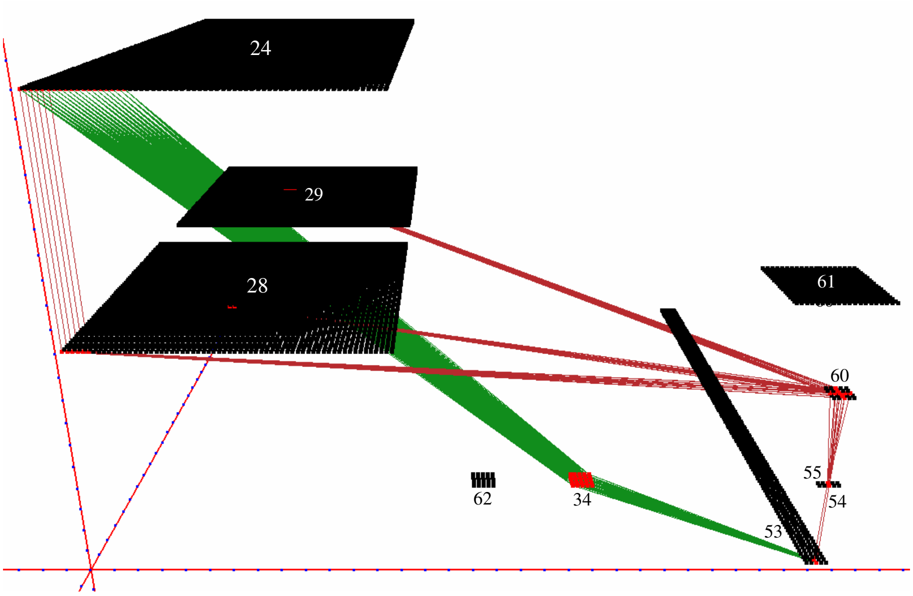


Figure A3.8. Large complex found during seed-based Eye Tilt analysis

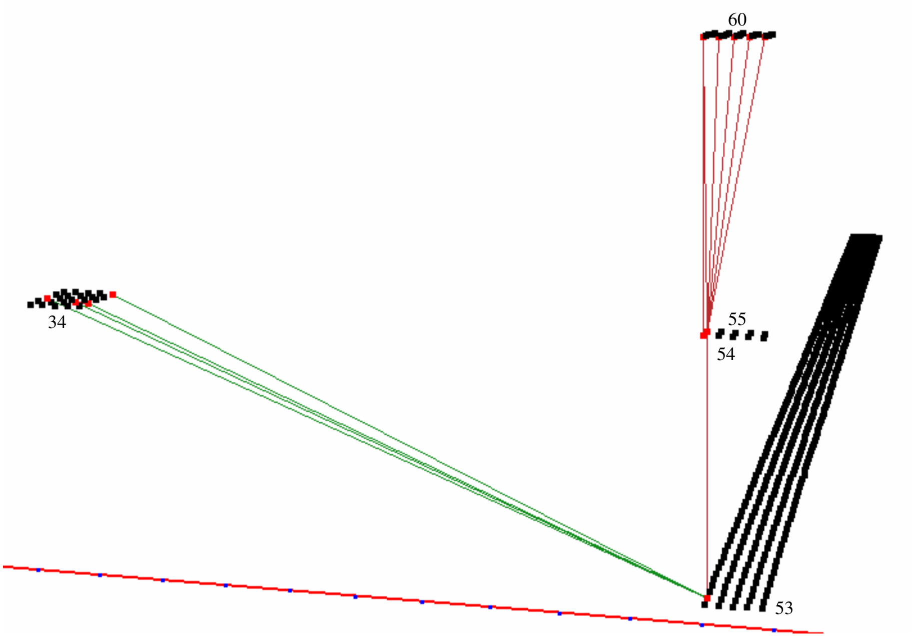


Figure A3.9. Small complex found during seed-based Eye Tilt analysis

A3.2.8 Motor Integration

12 of the seeds expanded into small complexes with 4 neurons and Φ = 4.0, as shown in Figure A3.10. The rest of the seeds expanded into subsets larger than 150 neurons with higher values of Φ, as shown in Figure A3.11. The analysis took 9.5 days.

Parameter	**Value**
Max number of bipartitions per level	25
Percentage of bipartition levels	100
Expansion rate per connection group	1
Maximum subset size	150
Maximum number of consecutive expansion failures per connection group	10
Only examine equal bipartitions	false

Table A3.9. Parameters for seed-based Motor Integration analysis

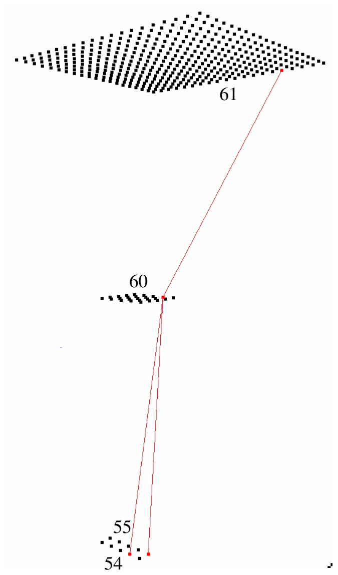


Figure A3.10. Small complex found during seed-based Motor Integration analysis

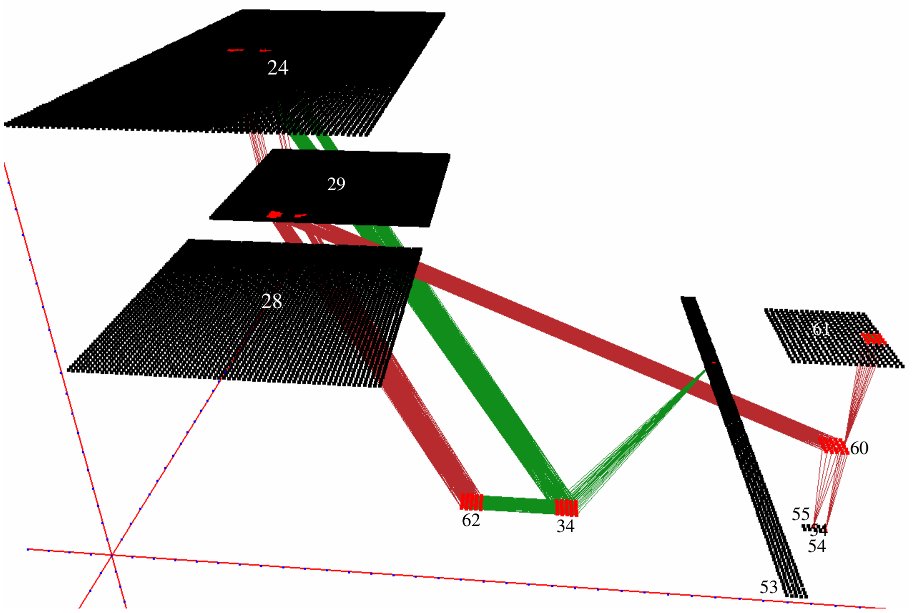


Figure A3.11. Subset during seed-based Motor Integration analysis

A3.2.9 Motor Cortex

This layer has 400 neurons, and so it was decided to start with a maximum subset size of 50. Since this layer has a large number of recurrent connections, it was anticipated that the seeds would expand into complexes that included the whole of Motor Cortex and possibly more. During the analysis all of the seeds in this neuron group expanded into subsets greater than 50 neurons.

Parameter	**Value**
Max number of bipartitions per level	10
Percentage of bipartition levels	100
Expansion rate per connection group	1
Maximum subset size	50
Maximum number of consecutive expansion failures per connection group	10
Only examine equal bipartitions	false

Table A3.10. Parameters for seed-based Motor Cortex analysis

A3.2.10 Emotion

Most of the seeds in this layer expanded into a complex of 25 neurons that included the whole of Emotion and had $\Phi = 79.9$. One seed expanded into a complex with 39 neurons and $\Phi = 74.4$. The analysis took approximately 20 hours.

Parameter	**Value**
Max number of bipartitions per level	50
Percentage of bipartition levels	100
Expansion rate per connection group	1
Maximum subset size	200
Maximum number of consecutive expansion failures per connection group	10
Only examine equal bipartitions	false

Table A3.11. Parameters for seed-based Emotion analysis

A3.3 Calculation of Φ on Neuron Groups(s)

Although 14,528 complexes were identified with the seed expansion method, the limit on subset size meant that many complexes could not be identified and the information integration of many neurons was not known – a problem that was particularly apparent in Motor Cortex and Motor Integration. To close these gaps in the analysis, the Φ calculation was also run on individual neuron groups and on combinations of connected neuron groups, up to a maximum size of about 700 neurons, which was the largest subset that could be analyzed in the time available. Neuron groups without recurrent connections – Blue Sensorimotor, Red Sensorimotor, Vision Input, Eye Pan, Eye Tilt, Motor Integration, and Motor Output - were only analyzed in combination with other neuron groups because they would have had zero Φ on their own.

To measure the effect of the approximations described in Section 7.4.4, these group analyses were also run with five bipartitions per level and using only equal bipartitions. However, only the values with the least approximation were used to generate the XML descriptions in Section 7.9. The results are presented in Table A3.12 and included as XML files in the Supporting Materials. These group analysis results are not complexes because it has not been shown that they are not included within a subset of higher Φ. To make this distinction clear they are referred to as *clusters* in this thesis.

	Neuron Group(s)	Size	Φ	Parameters	Analysis Time
1a	Inhibition	25	77.3	All bipartition levels, 50 bipartitions per level	8 seconds
1b			77.3	Equal bipartitions, 50 bipartitions per level	8 seconds
1c			77.3	All bipartition levels, 5 bipartitions per level	7 seconds
2a	Emotion	25	79.9	All bipartition levels, 50 bipartitions per level	8 seconds
2b			79.9	Equal bipartitions, 50 bipartitions per level	7 seconds
2c			80.2	All bipartition levels, 5 bipartitions per level	7 seconds
3a	Emotion + Inhibition	50	7.1	All bipartition levels, 50 bipartitions per level	17 seconds
3b			7.1	Equal bipartitions, 50 bipartitions per level	7 seconds
3c			7.1	All bipartition levels, 5 bipartitions per level	8 seconds
4a	Inhibition + Motor Output	700	8.4	All bipartition levels, 50 bipartitions per level	3 days
4b			8.4	Equal bipartitions, 50 bipartitions per level	9 minutes
4c			8.4	All bipartition levels, 5 bipartitions per level	6 hours
5a	Motor Cortex	400	17.9	All bipartition levels, 50 bipartitions per level	12 hours
5b			17.9	Equal bipartitions, 50 bipartitions per level	3 minutes
5c			17.9	All bipartition levels, 5 bipartitions per level	1 hour
6a	Motor Cortex + Motor Integration	425	58.7	All bipartition levels, 50 bipartitions per level	16 hours
6b			80.5	Equal bipartitions, 50 bipartitions per level	3.5 minutes
6c			58.7	All bipartition levels, 5 bipartitions per level	1.3 hours
7a	Motor Integration + Eye Pan + Eye Tilt	35	31.8	All bipartition levels, 50 bipartitions per level	8 seconds
7b			36.2	Equal bipartitions, 50 bipartitions per level	7 seconds
7c			31.8	All bipartition levels, 5 bipartitions per level	7 seconds
8a	Motor Cortex + Motor Integration + Eye Pan + Eye Tilt	435	58.7	All bipartition levels, 50 bipartitions per level	16.5 hours
8b			80.7	Equal bipartitions, 50 bipartitions per level	4 minutes
8c			58.7	All bipartition levels, 5 bipartitions per level	1.3 hours
9a	Motor Integration + Eye Pan + Eye Tilt + Motor Output + Inhibition	735	46.8	All bipartition levels, 50 bipartitions per level	7 days
9b			46.8	Equal bipartitions, 50 bipartitions per level	22 minutes
9c			46.8	All bipartition levels, 5 bipartitions per level	13.5 hours

Table A3.12. Neuron group(s) analysis results. The ‘b’ analyses use equal bipartitions, the ‘c’ analyses use only 5 bipartitions per level. Only the more accurate ‘a’ values were used to generate the XML description in Section 7.9.

Appendix 4
Gamez Publications Related to Machine Consciousness

Gamez, D. (2005). An Ordinal Probability Scale for Synthetic Phenomenology. In R. Chrisley, R. Clowes and S. Torrance (eds.), *Proceedings of the AISB05 Symposium on Next Generation approaches to Machine Consciousness*, Hatfield, UK, pp. 85-94.

Gamez, D. (2006). The XML Approach to Synthetic Phenomenology. In R. Chrisley, R. Clowes and S. Torrance (eds.), *Proceedings of the AISB06 Symposium on Integrative Approaches to Machine Consciousness*, Bristol, UK, pp. 128-35.

Gamez, D. (2007a). Progress in Machine Consciousness. *Consciousness and Cognition* doi:10.1016/j.concog.2007.04.005, in press.

Gamez, D. (2007b). SpikeStream: A Fast and Flexible Simulator of Spiking Neural Networks. In J. Marques de Sá, L.A. Alexandre, W. Duch and D.P. Mandic (eds.), *Proceedings of ICANN 2007*, Lecture Notes in Computer Science Volume 4668, Springer Verlag, pp. 370-9.

Gamez, D. (2007c). *What We Can Never Know*. London & New York: Continuum.

Gamez, D., Taffler, S., Delbruck, T. and Ponulak, F. (2006a). A Distributed Saliency System using Ethernet AER. *Report on the 2006 Workshop on Neuromorphic Engineering,* Telluride, pp. 45-52. Available at: http://ine-web.org/fileadmin/templates/_docs /report06_2.pdf.

Gamez, D., Newcombe, R., Holland, O. and Knight, R. (2006b). Two Simulation Tools for Biologically Inspired Virtual Robotics. *Proceedings of the IEEE 5th Chapter Conference on Advances in Cybernetic Systems*, Sheffield, pp. 85-90.

BIBLIOGRAPHY

Aleksander, I. (2005). *The World in My Mind, My Mind in the World: Key Mechanisms of Consciousness in People, Animals and Machines*. Exeter: Imprint Academic.

Aleksander, I. and Dunmall, B. (2000). An extension to the hypothesis of the asynchrony of visual consciousness. *Proceedings of the Royal Society of London B*, 267: 197-200.

Aleksander, I. and Dunmall, B. (2003). Axioms and Tests for the Presence of Minimal Consciousness in Agents. In O. Holland (ed.), *Machine Consciousness*. Exeter: Imprint Academic.

Aleksander, I., Lahnstein, M. and Lee, R. (2005). Will and Emotions: A Machine Model that Shuns Emotions. In R. Chrisley, R. Clowes, and S. Torrance (eds.) *Proceedings of the AISB05 Symposium on Next Generation approaches to Machine Consciousness: Imagination, Development, Intersubjectivity, and Embodiment,* Hatfield, UK, pp. 85-94.

Aleksander, I. and Morton, H. (2007a). Depictive Architectures for Synthetic Phenomenology. In A. Chella and R. Manzotti (eds.), *Artificial Consciousness*. Exeter: Imprint Academic.

Aleksander, I. and Morton, H. (2007b). Phenomenology and digital neural architectures. *Neural Networks* 20(9): 932-7.

Aleksander, I. and Morton, H. (2007c). Why Axiomatic Models of Being Conscious? *Journal of Consciousness Studies* 14(7): 15-27.

Ananthanarayanan, R. and Modha, D.S. (2007). Anatomy of a Cortical Simulator. *Supercomputing 07: Proceedings of the ACM/IEEE SC2007 Conference on High Performance Networking and Computing*, November 10-16, Reno, NV, USA.

Angel, L. (1989). *How to Build a Conscious Machine*. Boulder, San Francisco & London: Westview Press.

Anon. (2006). Artificial Consciousness. Retrieved 6th December 2006 from http://www.v72.org/mind_artificial_consciousness.htm.

Aquila, R.E. (1990). Consciousness as higher-order thoughts: two objections. *American Philosophical Quarterly* 27(1): 81-7.

Arbib, M.A. and Fellous, J.-M. (2004). Emotions: from brain to robot. *TRENDS in Cognitive Sciences* 8(12): 554-61.

Armstrong, D.M. (1981). *The Nature of Mind*. Brighton: The Harvester Press.

Asimov, I. (1952). *I, Robot*. London: Grayson & Grayson.

Baars, B.J. (1988). *A Cognitive Theory of Consciousness*. Cambridge: Cambridge University Press.

Baars, B.J. (2000). There are no known Differences in Brain Mechanisms of Consciousness Between Humans and other Mammals. *Animal Welfare* 10(1): 31-40.

Baird, J.C. and Noma, E. (1978). *Fundamentals of Scaling and Psychophysics*. New York, Chichester, Brisbane and Toronto: John Wiley & Sons.

Bauby, J-.D. (2002). *The Diving Bell and the Butterfly*. Translated by J. Leggatt. London: Fourth Estate.

Berkeley, G. (1988). *Principles of Human Knowledge and Three Dialogues between Hylas and Philonous*. London: Penguin.

Bernoulli, D. (1738). *Hydrodynamica*. Strasbourg: Johannis Reinholdi Dulseckeri.

Bialek, W., Rieke, F., de Ruyter van Steveninck, R.R. and Warland, D. (1991). Reading a neural code. *Science* 252: 1854-7.

Binzegger, T., Douglas, R.J. and Martin, K.A.C. (2004). A Quantitative Map of the Circuit of Cat Primary Visual Cortex. *The Journal of Neuroscience* 24(39): 8441–53.

Blackmore, S.J. (2002). What is consciousness? In H. Swain (ed.), *Big Questions in Science*. London: Jonathan Cape, pp. 39-43.

Block, N. (1978). Troubles with Functionalism. In C. Wade Savage (ed.), *Minnesota Studies in the Philosophy of Science*, *Volume IX, Perception and Cognition Issues in the Foundations of Psychology*. Minneapolis: University of Minnesota Press.

Block, N. (1995). On a confusion about the function of consciousness. *Behavioral and Brain Sciences* 18(2): 227-47.

Bosse, T., Jonker, C.M. and Treur, J. (2005). Simulation and Representation of Body, Emotion, and Core Consciousness. In R. Chrisley, R. Clowes and S. Torrance (eds.), *Proceedings of the AISB05 Symposium on Next Generation approaches to Machine Consciousness*, Hatfield, UK.

Brader, J.M., Senn, W. and Fusi, S. (2006). Learning real world stimuli in a neural network with spike-driven synaptic dynamics. Submitted to *Neural Computation.*

Brette, R., Rudolph, M., Carnevale, T., Hines, M., Beeman, D., Bower, J.M., Diesmann, M., Morrison, A., Goodman, P.H., Harris Jr, F.C., Zirpe, M., Natschlaeger, T., Pecevski, D., Ermentrout, B., Djurfeldt, M., Lansner, A., Rochel, O., Vieville, T., Muller, E., Davison, A.P., El Boustani, S. and Destexhe, A. (2006). Simulation of networks of spiking neurons: A review of tools and strategies. *Journal of Computational Neuroscience*, in press.

Bringsjord, S. (2007). Offer: One Billion Dollars for a Conscious Robot; If You're Honest, You Must. *Journal of Consciousness Studies* 14(7): 28-43.

Brooks, R., Breazeal, C., Marjanovic, M., Scassellati, B. and Williamson, M. (1998). The Cog Project: Building a Humanoid Robot. In C. Nehaniv (ed.), *Computation for Metaphors, Analogy and Agents*, Vol. 1562 of Springer Lecture Notes in Artificial Intelligence. Berlin: Springer-Verlag.

Brovelli, A, Ding, M., Ledberg, A., Chen, Y., Nakamura, R. and Bressler, S.L. (2004). Beta oscillations in a large-scale sensorimotor cortical network: Directional influences revealed by Granger causality. *Proc Natl Acad Sci USA* 101: 9849-54.

Byrne, A. (1997). Some like it HOT: consciousness and higher-order thoughts. *Philosophical Studies* 86: 103-29.

Calverley, D.J. (2005). Towards a Method for Determining the Legal Status of a Conscious Machine. In R. Chrisley, R. Clowes and S. Torrance (eds.), *Proceedings of the AISB05 Symposium on Next Generation approaches to Machine Consciousness*, Hatfield, UK.

Carruthers, P. (2000). *Phenomenal Consciousness: a Naturalistic Theory*. Cambridge: Cambridge University Press.

Chalmers, D. (1996). *The Conscious Mind*. Oxford: Oxford University Press.

Chalmers, D. (1998). On the Search for the Neural Correlates of Consciousness. In S. Hameroff, A. Kaszniak and A. Scott, (eds.), *Toward a Science of Consciousness II: The Second Tucson Discussions and Debates*. Cambridge, Massachusetts: MIT Press.

Chella, A. (2007). Towards Robot Conscious Perception. In A. Chella and R. Manzotti (eds.), *Artificial Consciousness*. Exeter: Imprint Academic, pp. 124-40.

Chella, A. and Macaluso, I. (2006). Sensations and Perceptions in "Cicerobot" a Museum Guide Robot. *Proceedings of BICS 2006*, Lesbos, Greece.

Chella, A. and Manzotti, R. (eds.) (2007). *Artificial Consciousness*. Exeter: Imprint Academic.

Chrisley, R.J. (1995). Taking Embodiment Seriously: Nonconceptual Content and Robotics. In K.M. Ford, C. Glymour and P.J. Hayes (eds.), *Android Epistemology*. Menlo Park, Cambridge and London: AAAI Press/ The MIT Press.

Chrisley, R.J., Clowes, R. and Torrance, S. (eds.) (2007). *Journal of Consciousness Studies* 14(7).

Chrisley, R.J. and Parthemore, P. (2007). Synthetic Phenomenology: Exploiting Embodiment to Specify the Non-Conceptual Content of Visual Experience. *Journal of Consciousness Studies* 14 (7): 44-58.

Churchland, P. (1989). *A Neurocomputational Perspective*. Cambridge, Massachusetts: The MIT Press.

Clark, A. and Chalmers, D.J. (1998). The Extended Mind. *Analysis* 58: 10-23.

Clark T.W. (1999). Fear of mechanism. A compatibilist critique of ‘The Volitional Brain’. *Journal of Consciousness Studies* 6(8-9): 279-93.

Claxton, G. (1999). Whodunnit? Unpicking the ‘seems’ of free will. *Journal of Consciousness Studies* 6(8-9) 99-113.

Cleeremans, A., Timmermans, B. and Pasquali, A. (2007). Consciousness and metarepresentation: A computational sketch. *Neural Networks* 20: 1032–9.

Clore, G.L. (1992). Cognitive Phenomenology: Feelings and the Construction of Judgment. In L.L. Martin and A. Tesser (eds.), *The Construction of Social Judgments*. Hillsdale, New Jersey, Hove, and London: Lawrence Erlbaum Associates, pp. 133–63.

Clowes, R.W. (2006). The Problem of Inner Speech and its relation to the Organization of Conscious Experience: A Self-Regulation Model. In R. Chrisley, R. Clowes and S. Torrance (eds.), *Proceedings of the AISB06 Symposium on Integrative Approaches to Machine Consciousness*, Bristol, UK, pp. 117-26.

Clowes, R.W. (2007). A Self-Regulation Model of Inner Speech and its Role in the Organisation of Human Conscious Experience. *Journal of Consciousness Studies* 14(7): 59-71.

Clowes, R.W. and Morse, A.F. (2005). Scaffolding Cognition with Words. In L. Berthouze, F. Kaplan, H. Kozima, H. Yano, J. Konczak, G. Metta, J. Nadel, G. Sandini, G. Stojanov, and C. Balkenius, (eds.), *Proceedings of the Fifth International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*, Lund University Cognitive Studies 123, Lund.

Cooney, B. (1979). The neural basis of self-consciousness. *Nature and System* 1: 16-31.

Cotterill, R. (2000). *Enchanted Looms*. Cambridge: Cambridge University Press.

Cotterill, R. (2003). CyberChild: A Simulation Test-Bed for Consciousness Studies. In O. Holland (ed.), *Machine Consciousness*. Exeter: Imprint Academic.

Coward, L.A. and Sun, R. (2007). Hierarchical approaches to understanding consciousness. *Neural Networks* 20: 947–54.

Crick, F. (1994). *The Astonishing Hypothesis*. London: Simon & Schuster.

Crick, F. and Koch, C. (2000). The Unconscious Homunculus. In T. Metzinger (ed.), *Neural Correlates of Consciousness: Empirical and Conceptual Questions*. Cambridge, Massachusetts: The MIT Press, pp. 103-10.

Crick, F. and Koch, C. (2003). A framework for consciousness. *Nature Neuroscience* 6(2): 119-26.

Crook, J.H. (1983). On attributing consciousness to animals. *Nature* 303: 11-14.

Cruse, H. (1999). Feeling our body – The basis of cognition? *Evolution and Cognition* 5: 162-73.

Damasio, A.R. (1995). *Descartes' Error: Emotion, Reason and the Human Brain*. London: Picador.

Damasio, A.R. (1999). *The Feeling of What Happens*. New York, San Diego and London: Harcourt Brace & Company.

Daprati, E., Franck, N., Georgieff, N., Proust, J., Pacherie, E., Dalery, J. and Jeannerod, M. (1997). Looking for the agent: an investigation into consciousness of action and self-consciousness in schizophrenic patients. *Cognition* 65: 71–86.

Dautenhahn, K. (2007). Socially intelligent robots: dimensions of human–robot interaction. *Phil. Trans. R. Soc. B* 362: 679–704.

Dawkins, R. (1998). *Unweaving the Rainbow*. London: Penguin.

Dehaene, S., Kerszberg, M. and Changeux, J.P. (1998). A neuronal model of a global workspace in effortful cognitive tasks. *Proceedings of the National Academy of Sciences USA* 95: 14529–34.

Dehaene, S. and Naccache, L. (2001). Towards a cognitive neuroscience of consciousness : Basic evidence and a workspace framework. *Cognition* 79: 1-37.

Dehaene, S., Sergent, C. and Changeux, J.-P. (2003). A neuronal network model linking subjective reports and objective physiological data during conscious perception. *Proceedings of the National Academy of Sciences USA* 100: 8520-5.

Dehaene S., and Changeux, J.-P. (2005). Ongoing Spontaneous Activity Controls Access to Consciousness: A Neuronal Model for Inattentional Blindness. *Public Library of Science Biology* 3(5), e141.

Delorme, A. and Thorpe, S.J. (2003). SpikeNET: An Event-driven Simulation Package for Modeling Large Networks of Spiking Neurons. *Network: Computational in Neural Systems* 14: 613-27.

DeMarse, T.B., Wagenaar, D.A., Blau, A.W. and Potter, S.M. (2001). The Neurally Controlled Animat: Biological Brains Acting With Simulated Bodies. *Autonomous Robots* 11(3): 305-10.

Dennett, D.C. (1988). Quining Qualia. In A. Marcel and E. Bisiach (eds.), *Consciousness in Modern Science*. Oxford: Oxford University Press.

Dennett, D.C. (1992). *Consciousness Explained*. London: Penguin.

Dennett, D.C. (1997). Consciousness in Human and Robot Minds. In M. Ito, Y. Miyashita and E.T. Rolls (eds.), *Cognition, Computation and Consciousness*. Oxford: Oxford University Press.

Descartes, R. (1975). A *Discourse on Method; Meditations on the First Philosophy; Principles of Philosophy*. Translated by J. Veitch. London: Dent.

Diesmann, M. and Gewaltig, M-O. (2002). NEST: An Environment for Neural Systems Simulations. In V. Macho (ed.), Forschung und wissenschaftliches Rechnen, Heinz-Billing-Preis, GWDG-Bericht.

Double, R. (1991). *The Non-Reality of Free Will*. Oxford and New York: Oxford University Press.

Dreyfus, H.L. (1992). *What Computers Still Can't Do: A Critique of Artificial Reason*. Cambridge, Massachusetts: The MIT Press.

Duch, W. (2005). Brain-Inspired Conscious Computing Architecture. *The Journal of Mind and Behavior* 26(1-2): 1-22.

Eccles, J.C. (1994). *How the Self Controls Its Brain*. Berlin, Heidelberg and New York: Springer-Verlag.

Edelman, G.M. and Tononi, G. (2000). *Consciousness: How Matter Becomes Imagination*. London: Penguin.

Evans, E.P. (1987). *The Criminal Prosecution and Capital Punishment of Animals: The Lost History of Europe's Animal Trials*. London: Faber and Faber.

Findlay, J.M. and Gilchrist, I.D. (2003). *Active Vision*. Oxford: Oxford University Press.

Flanagan, O. (1992). *Consciousness Reconsidered*. Cambridge, Massachusetts: The MIT Press.

Flohr, H. (2000). NMDA Receptor-Mediated Computational Processes and Phenomenal Consciousness. In T. Metzinger (ed), *Neural Correlates of Consciousness*. Cambridge, Massachusetts and London, England: The MIT Press.

Franklin, S. (2000). Deliberation and Voluntary Action in 'Conscious' Software Agents. *Neural Network World* 10: 505-21.

Franklin, S. (2001). Automating Human Information Agents. In Z. Chen and L.C. Jain (eds.), *Practical Applications of Intelligent Agents*. Berlin: Springer-Verlag.

Franklin, S. (2003). IDA: A Conscious Artifact. In O. Holland (ed.), *Machine Consciousness*. Exeter: Imprint Academic.

Franklin, S., Baars, B.J., Ramamurthy, U., and Ventura, M.. (2005). The Role of Consciousness in Memory. *Brains, Minds and Media* 1: 1-38.

Friston, K.J., Harrison, L. and Penny, W. (2003). Dynamic causal modelling. *NeuroImage* 19: 1273–302.

Furber, S.B., Temple, S. and Brown, A.D. (2006). High-Performance Computing for Systems of Spiking Neurons. *Proc. AISB'06 workshop on GC5: Architecture of Brain and Mind,* Bristol, Vol.2, pp 29-36.

Galletti, C. and Battaglini, P.P. (1989). Gaze-Dependent Visual Neurons in Area V3A of Monkey Prestriate Cortex. *The Journal of Neuroscience* 9(4): 1112-25.

Gamez, D. (2005). An Ordinal Probability Scale for Synthetic Phenomenology. In R. Chrisley, R. Clowes and S. Torrance (eds.), *Proceedings of the AISB05 Symposium on Next Generation approaches to Machine Consciousness*, Hatfield, UK, pp. 85-94.

Gamez, D. (2006). The XML Approach to Synthetic Phenomenology. In R. Chrisley, R. Clowes and S. Torrance (eds.), *Proceedings of the AISB06 Symposium on Integrative Approaches to Machine Consciousness*, Bristol, UK, pp. 128-35.

Gamez, D. (2007a). Progress in Machine Consciousness. *Consciousness and Cognition* doi:10.1016/j.concog.2007.04.005, in press.

Gamez, D. (2007b). SpikeStream: A Fast and Flexible Simulator of Spiking Neural Networks. In J. Marques de Sá, L.A. Alexandre, W. Duch and D.P. Mandic (eds.), *Proceedings of ICANN 2007*, Lecture Notes in Computer Science Volume 4668, Springer Verlag, pp. 370-9.

Gamez, D. (2007c). *What We Can Never Know*. London & New York: Continuum.

Gamez, D., Taffler, S., Delbruck, T. and Ponulak, F. (2006a). A Distributed Saliency System using Ethernet AER. *Report on the 2006 Workshop on Neuromorphic Engineering,* Telluride, pp. 45-52. Available at: http://ine-web.org/fileadmin/templates/_docs /report06_2.pdf.

Gamez, D., Newcombe, R., Holland, O. and Knight, R. (2006b). Two Simulation Tools for Biologically Inspired Virtual Robotics. *Proceedings of the IEEE 5th Chapter Conference on Advances in Cybernetic Systems*, Sheffield, pp. 85-90.

Gazzaniga, M.S. (1970). *The Bisected Brain*. New York: Appleton-Century-Crofts.

Gennaro, R.J. (ed.) (2004). *Higher-order Theories of Consciousness: An Anthology*. Amsterdam: John Benjamins Publishing Company.

Gerstner, W. and Kistler, W. (2002). *Spiking Neuron Models*. Cambridge University Press, Cambridge.

Gescheider, G.A. (1997). *Psychophysics: The Fundamentals, Third Edition*. Mahwah, New Jersey and London: Lawrence Erlbaum Associates.

Geschwind, D.H., Iacoboni, M., Mega, M.S., Zaidel, D.W., Cloughesy, T. and Zaidel, E. (1995). Alien hand syndrome: Interhemispheric motor disconnection due to a lesion in the midbody of the corpus callosum. *Neurology* 45: 802–8.

Gjertsen, Derek (1986). *The Newton Handbook*. London and New York: Routledge and Kegan Paul.

Goertzel, B. and Pennachin, C. (eds.) (2007). *Artificial General Intelligence*. Berlin: Springer.

Goethe, J.W. (1959). *Faust (Part Two)*. Translated by P. Wayne. London: Penguin.

Goguen, J.A. and Forman, R.K.C. (eds.) (1995). *Journal of Consciousness Studies: Explaining Consciousness - 'The Hard Problem' Part 1*, Volume 2, Issue 3.

Goguen, J.A. and Forman, R.K.C. (eds.) (1996). *Journal of Consciousness Studies: Explaining Consciousness - 'The Hard Problem' Part 2*, Volume 3, Issue 1.

Gomes, G. (1998). The Timing of Conscious Experience: A Critical Review and Reinterpretation of Libet's Research. *Consciousness and Cognition* 7(4): 559-95.

Gomes G. (1999). Volition and the readiness potential. *Journal of Consciousness Studies* 6(8-9): 59-76.

Grand, S. (2003). *Growing up with Lucy*. London: Weidenfeld & Nicolson.

Gray, J.A. (2004). *Consciousness: Creeping up on the Hard Problem*. Oxford: Oxford University Press.

Grossberg, S. (1976). Adaptive pattern classification and universal recoding, I: Parallel development and coding of neural feature detectors. *Biological Cybernetics* 23: 121-34.

Grush, R. (2004). The emulation theory of representation: motor control, imagery, and perception. *Behavioral and Brain Sciences* 27: 377-442.

Grush, R. and Churchland, P.S. (1995). Gaps in Penrose's toiling. *Journal of Consciousness Studies* 2(1): 10-29.

Haikonen, P.O. (2003). *The Cognitive Approach to Conscious Machines*. Exeter: Imprint Academic.

Haikonen, P.O. (2006). Towards the Times of Miracles and Wonder; A Model for a Conscious Machine. *Proceedings of BICS 2006*, Lesbos, Greece.

Haikonen, P.O. (2007). *Robot Brains: Circuits and Systems for Conscious Machines*. Hoboken, New Jersey: John Wiley & Sons.

Hameroff, S. and Penrose, R. (1996). Orchestrated Reduction of Quantum Coherence in Brain Microtubules: A Model for Consciousness? In S.R. Hameroff, A.W. Kaszniak, and A.C. Scott (eds.), *Toward a Science of Consciousness - The First Tucson Discussions and Debates*. Cambridge, MA: MIT Press, pp. 507-40.

Harnad, S. (1990). The Symbol Grounding Problem. *Physica D* 42: 335-46.

Harnad, S. (1994). Levels of Functional Equivalence in Reverse Bioengineering: The Darwinian Turing Test for Artificial Life. *Artificial Life* 1(3): 293-301.

Harnad, S. (2003). Can a Machine be Conscious? How? In O. Holland (ed.), *Machine Consciousness*. Exeter: Imprint Academic.

Hassabis, D., Kumaran, D., Vann, S.D. and Maguire, E.A. (2007). Patients with hippocampal amnesia cannot imagine new experiences. *PNAS* 104(5): 1726-31.

Haxby, J.V., Gobbini, M.I., Furey, M.L., Ishai, A., Schouten, J.L., and Pietrini, P. (2001). Distributed and overlapping representations of faces and objects in ventral temporal cortex. *Science* 293(5539): 2425-30.

Haydon, P. (2000). Neuroglial networks: Neurons and glia talk to each other. *Current Biology* 10(19): R712-R714.

Haynes, J.D. and Rees, G. (2005a). Predicting the orientation of invisible stimuli from activity in human primary visual cortex. *Nature Neuroscience* 8: 686-91.

Haynes, J.D. and Rees, G. (2005b). Predicting the stream of consciousness from activity in human visual cortex. *Current Biology* 15: 1301-7.

Hebb, D.O. (1949). The Organization of Behavior. New York: John Wiley.

Heidegger, M. (1995a). *Being and Time.* Translated by J. Macquarrie and E. Robinson. Oxford: Blackwell.

Heidegger, M. (1995b). *The Fundamental Concepts of Metaphysics.* Translated by W. McNeill and N. Walker. Bloomington and Indianapolis: Indiana University Press.

Herzog, M.H., Esfeld, M. and Gerstner, W. (2007). Consciousness & the small network argument. *Neural Networks* 20: 1054–6.

Hesslow, G. and Jirenhed, D.-A. (2007). The Inner World of a Simple Robot. *Journal of Consciousness Studies* 14(7): 85-96.

Hodgson, D. (2005). A Plain Person's Free Will. *Journal of Consciousness Studies* 12(1): 3-19.

Holcombe, M. and Paton, R.C. (eds.) (1998). *Information Processing in Cells and Tissues*. New York: Plenum Press.

Holland, O. (ed.) (2003). *Machine Consciousness*. Exeter: Imprint Academic.

Holland, O. (2007). A Strongly Embodied Approach to Machine Consciousness. *Journal of Consciousness Studies* 14(7): 97-110.

Holland, O. and Goodman, R. (2003). Robots With Internal Models. In O. Holland (ed.), *Machine Consciousness*. Exeter: Imprint Academic.

Holland, O. and Knight, R. (2006). The Anthropomimetic Principle. In J. Burn and M. Wilson (eds.), *Proceedings of the AISB06 Symposium on Biologically Inspired Robotics*, Bristol, UK.

Holland, O., Knight, R. and Newcombe, R. (2007). A robot-based approach to machine consciousness. In A. Chella and R. Manzotti (eds.) *Artificial Consciousness*. Exeter: Imprint Academic.

Honderich, T. (1993). *How Free are You?* Oxford and New York: Oxford University Press.

Honey, C.J., Kötter, R., Breakspear, M. and Sporns, O. (2007). Network structure of cerebral cortex shapes functional connectivity on multiple time scales. *PNAS* 104(24): 10240–5.

Hubel, D.H. and Wiesel, T.N. (1959). Receptive fields of single neurones in the cat's striate cortex. *Journal of Physiology* 148(3): 574–91.

Hume, D. (1983). *A Treatise of Human Nature*. Oxford: Oxford University Press.

Husserl, E. (1960). *Cartesian Meditations*. Translated by Dorion Cairns. The Hague: Nijhoff.

Husserl, E. (1964). *The Phenomenology of Internal Time-consciousness*. The Hague: Nijhoff.

Ikegami, T. (2007). Subscribed Content Simulating Active Perception and Mental Imagery with Embodied Chaotic Itinerancy. *Journal of Consciousness Studies* 14(7): 111-25.

Izhikevich, E.M. (2003). Simple Model of Spiking Neurons. *IEEE Transactions on Neural Networks* 14: 1569- 72.

Izhikevich, E.M., Gally J.A. and Edelman, G.M. (2004). Spike-Timing Dynamics of Neuronal Groups. *Cerebral Cortex* 14: 933-44.

Jackendoff, R. (1987). *Consciousness and the Computational Mind.* Cambridge, Massachusetts and London: The MIT Press.

Jonker, C.M. and Treur, J. (2002). Compositional Verification of Multi-Agent Systems: a Formal Analysis of Pro-activeness and Reactiveness. *International Journal of Cooperative Information Systems* 11: 51-92.

Jordan, J.S. (1998). Synthetic phenomenology? Perhaps, but not via information processing. Talk given at the Max Planck Institute for Psychological Research, Munich, Germany.

Joy, B. (2000). Why the future doesn't need us. *Wired* 8.04. Retrieved 6th December 2006 from http://www.wired.com/wired/archive/8.04/joy.html.

Julesz, B. (1971). *Foundations of Cyclopean Perception.* Chicago: University of Chicago Press.

Kaczynski, T. (1995). *Industrial Society and Its Future*. Retrieved 6th December 2006 from http://www.thecourier.com/manifest.htm.

Kamitani, Y. and Tong, F. (2005). Decoding the visual and subjective contents of the human brain. *Nature Neuroscience* 8:(5) 679-85.

Kane, R. (1996). *The Significance of Free Will.* Oxford and New York: Oxford University Press.

Kant, I. (1996). *Critique of Pure Reason*. Translated by W.S. Pluhar. Indianapolis: Hackett Publishing Company.

Kay, K.N., Naselaris, T., Prenger, R.J. and Gallant, J.L. (2008). Identifying natural images from human brain activity. *Nature* advance online publication, doi:10.1038/nature06713.

Kent, E.W. (1981). *The Brains of Men and Machines*. Peterborough: BYTE/ McGraw Hill.

Kim, J. (2005). *Physicalism, or Something Near Enough.* Princeton and Oxford: Princeton University Press.

Koch, C. (2004). *The Quest for Consciousness: A Neurobiological Approach.* Englewood, Colorado: Roberts and Company Publishers.

Kohonen, T. (2001). *Self-Organizing Maps, Third Edition*. Berlin, Heidelberg and New York: Springer.

Kossyln, S.M. (1994). *Image and Brain*. Cambridge, Massachusetts and London, England: The MIT Press.

Kouider, S. and Dehaene, S. (2007). Levels of processing during non-conscious perception: a critical review of visual masking. *Phil. Trans. R. Soc. B* 362: 857–75.

Kreiman, G., Koch, C. and Fried, I. (2000). Imagery neurons in the human brain. *Nature* 408: 357-61.

Krichmar, J.L. and Edelman, G.M. (2006). Principles Underlying the Construction of Brain-Based Devices. In T. Kovacs, and J.A.R. Marshall (eds.), *Proceedings of AISB'06: Adaptation in Artificial and Biological Systems*, Bristol, UK, pp. 37-42.

Krichmar, J.L., Nitz, D.A., Gally, J.A. and Edelman, G.M. (2005). Characterizing functional hippocampal pathways in a brain-based device as it solves a spatial memory task. *PNAS* 102(6): 2111-6.

Kruskal, J.B. (1964). Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika* 29: 1-27.

Kurzweil, R. (2000). *The Age of Spiritual Machines*. London: Penguin Putnam.

Laureys, S, Owen, A.M., and Schiff, N.D. (2004). Brain function in coma, vegetative state, and related disorders. *The Lancet Neurology* 3(9): 537-46.

Laureys, S., Antoine, S., Boly, M., Elincx, S., Faymonville, M-E., Berré, J., Sadzot, B., Ferring, M., De Tiège, X., Van Bogaert, P., Hansen, I., Damas, P., Mavroudakis, N., Lambermont, B., Del Fiore, G., Aerts, J., Degueldre, C., Phillips, C., Franck, G., Vincent, J-L., Lamy, M., Luxen, A., Moonen, G., Goldman, S. and Maquet, P. (2002). Brain function in the vegetative state. *Acta neurol. belg*. 102: 177-85.

Lee, U., Kim, S., Noh, G.-J. and Choi, B.-M. (2007). A new dynamic property of human consciousness. Available from *Nature Precedings*: http://hdl.nature.com/10101 /npre.2007.1244.1.

Legrand, D. (ed.) (2005). *Psyche: Thomas Metzinger "Being No One"*, Volume 11, Issue 5.

Lehar, S. (2003). *The World in Your Head.* Mahwah, New Jersey: Lawrence Erlbaum Associates.

Libet, B. (1982). Brain stimulation in the study of neuronal functions for conscious sensory experiences. *Human Neurobiology* 1: 235-42.

Libet, B. (1985). Unconscious cerebral initiative and the role of conscious will in voluntary action. *Behavioural and Brain Sciences* 8: 529-39.

Libet B. (1999). Do we have free will? *Journal of Consciousness Studies*, 6(8-9): 47-57.

Linåker, F. and Niklasson, L. (2000). Time series segmentation using an adaptive resource allocating vector quantization network based on change detection. *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN 2000)*, pp. 323-8.

Locke, J. (1997). *An Essay Concerning Human Understanding*. Edited by R. Woolhouse. London: Penguin Books.

Logothetis, N. (1998). Single units and conscious vision. *Philosophical Transactions of the Royal Society of London B* 353: 1801-18.

Lungarella, M., Pegors, T., Bulwinkle, D. and Sporns, O. (2005). Methods for Quantifying the Informational Structure of Sensory and Motor Data. *Neuroinformatics* 3(3): 243-62.

Lungarella, M. and Sporns, O. (2006). Mapping Information Flow in Sensorimotor Networks. *PLoS Computational Biology* 2(10): 1301-12.

Maas, W. and Bishop, C.M. (eds.) (1999). *Pulsed Neural Networks*. Cambridge, Massachusetts: The MIT Press.

Mach, E. (1976). *Knowledge and Error*. Translated by T.J. McCormack. Dordrecht: D. Reidel Publishing Company.

Marian, I. (2003). A biologically inspired computational model of motor control development. MSc Thesis, Department of Computer Science, University College Dublin, Ireland.

Markram, H. (2006). The Blue Brain Project. *Nature Reviews Neuroscience* 7: 153-60.

Massimini, M., Ferrarelli, F., Huber, R., Esser, S.K., Singh, H. and Tononi, G. (2005). Breakdown of cortical effective connectivity during sleep. *Science* 309: 2228-32.

Matuszek, C., Cabral, J., Witbrock, M. and DeOliveira, J. (2006). An Introduction to the Syntax and Content of Cyc. *Proceedings of the 2006 AAAI Spring Symposium on Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering*, Stanford, CA.

McCauley, L. (2002). Neural Schemas: A Mechanism for Autonomous Action Selection and Dynamic Motivation. *Proceedings of the 3rd WSES Neural Networks and Applications Conference*, Switzerland.

Melzack, R. (1992). Phantom Limbs. *Scientific American* 266: 120-6.

Merleau-Ponty, M. (1989). *Phenomenology of Perception*. Translated by C. Smith. London: Routledge.

Merleau-Ponty, M. (1995). *The Visible and the Invisible*. Edited by C. Lefort, translated by A. Lingis. Evanston: Northwestern University Press.

Metzinger, T. (ed.) (2000). *Neural Correlates of Consciousness: Empirical and Conceptual Questions*. Cambridge, Massachusetts: The MIT Press.

Metzinger, T. (2003). *Being No One*. Cambridge, Massachusetts: The MIT Press.

Metzinger, T. and Windt, J.M. (2007). The philosophy of dreaming and self-consciousness: What happens to the experiential subject during the dream state? In D. Barrett and P. McNamara (eds.), *The New Science of Dreaming*. Estport, CT: Praeger Imprint/Greenwood Publishers.

Metzinger, T. (2008). Empirical perspectives from the self-model theory of subjectivity: A brief summary with examples. *Progress in Brain Research* 168: 215-46.

Milner, A.D. and Goodale, M.A. (1995). *The Visual Brain in Action*. Oxford: Oxford University Press.

Moor, J.H. (1988). Testing robots for qualia. In H.R. Otto and J.A. Tuedio (eds.), *Perspectives on Mind*. Dordrecht/ Boston/ Lancaster/ Tokyo: D. Reidel Publishing Company.

Moravec, H. (1988). *Mind Children*. Cambridge, Massachusetts: Harvard University Press.

Mulhauser, G. (1998). *Mind Out of Matter.* Dordrecht: Kluwer Academic Publishers.

Nagel, T. (1974). What is it like to be a bat? *The Philosophical Review* 83: 435-56.

Newman, J., Baars, B.J. and Cho, S.-B. (1997). A Neural Global Workspace Model for Conscious Attention. *Neural Networks* 10 (7): 1195-206.

Nietsche, F. (1966). *Beyond Good and Evil.* Translated by W. Kaufmann. New York: Random House.

Noë, A. and Thompson, E. (2004). Are There Neural Correlates of Consciousness? *Journal of Consciousness Studies* 11(1): 3–28.

Onians, R.B. (1973). *The Origins of European Thought*. New York: Arno Press.

O'Regan, J.K. and Noë, A. (2001). A sensorimotor account of vision and visual consciousness. *Behavioral and Brain Sciences* 24: 939-1031.

Oxford English Dictionary (1989). Prepared by J.A. Simpson and E.S.C. Weiner. Oxford: Clarendon Press.

Papoulis, A. (1984). *Probability, Random Variables, and Stochastic Processes, Second Edition*. New York: McGraw-Hill.

Paton, R., Bolouri, H., Holcombe, W.M.L., Parish, J.H. and Tateson, R. (2003) *Computation in Cells and Tissues: Perspectives and Tools of Thought*. Berlin and Heidelberg: Springer-Verlag.

Penrose, R. (1990). *The Emperor's New Mind*. London: Vintage.

Penrose, R. (1995). *Shadows of the Mind*. London: Vintage.

Poland, J. (1994). *Physicalism: The Philosophical Foundations*. Oxford: Clarendon Press.

Ponulak, F. and Kasiński, A. (2006). ReSuMe learning method for Spiking Neural Networks dedicated to neuroprostheses control. *Proc. of EPFL LATSIS Symposium 2006, Dynamical principles for neuroscience and intelligent biomimetic devices*, Lausanne, Switzerland, pp.119-20.

Pöppel, E. (1972). Oscillations as possible basis for time perception. In J.T. Fraser, ed., *The Study of Time*. Berlin: Springer-Verlag.

Pöppel, E. (1978). Time perception. In R. Held, H.W. Leibowitz and H.L. Teuber (eds.), *Handbook of Sensory Physiology*, Vol. 8. New York: Springer-Verlag.

Pöppel, E. (1985). *Grenzen des Bewußtein*. Weinheim, Germany: VCH Verlagsgesellschaft.

Pöppel, E. (1994). Temporal mechanisms in perception. *International Review of Neurobiology* 37: 185-202.

Popper, K. (2002). *The Logic of Scientific Discovery*. London and New York: Routledge.

Prinz, J.J. (2003). Level-Headed Mysterianism and Artificial Experience. In O. Holland (ed.), *Machine Consciousness*. Exeter: Imprint Academic.

Quian Quiroga, R., Reddy, L., Kreiman, G., Koch, C. and Fried, I. (2005). Invariant visual representation by single neurons in the human brain. *Nature* 435: 1102-7.

Ramachandran, V.S. and Blakeslee, S. (1998). *Phantoms in the Brain*. London: Fourth Estate.

Reber, A.S. (1967). Implicit learning of artificial grammars. *Journal of Verbal Learning and Verbal Behavior* 6: 855–63.

Revonsuo, A. (1995). Consciousness, dreams and virtual realities. *Philosophical Psychology* 8: 35-58.

Roberts, A. and Bush, B.M.H. (1981). *Neurons without Impulses*. Cambridge: Cambridge University Press.

Rorty, R. (1979). *Philosophy and the Mirror of Nature*. Princeton, New Jersey: Princeton University Press.

Rosenthal, D.M. (1986). Two Concepts of Consciousness. *Philosophical Studies* 49(3): 329-59.

Rowlands, M. (2001). Consciousness and higher-order thoughts. *Mind and Language* 16(3): 290-310.

Russell, B. (1927). *An Outline of Philosophy*. London: Allen and Unwin.

Samsonovich, A.V., and DeJong, K.A. (2005a). Designing a self-aware neuromorphic hybrid. In K.R. Thorisson, H. Vilhjalmsson and S. Marsela (eds.), AAAI-05 Workshop on Modular Construction of Human-Like Intelligence, Pittsburg, PA, *AAAI Technical Report WS-05-08*, pp. 71- 78. Menlo Park, CA: AAAI Press.

Samsonovich, A.V., and DeJong K.A. (2005b). A general-purpose computational model of the conscious mind. In M. Lovett, C. Schunn, C. Lebiere and P. Munro (eds.), *Proceedings of the Sixth International Conference on Cognitive Modeling ICCM-2004*, Mahwah, NJ: Lawrence Erlbaum Associates, pp. 382-3.

Schreber, D.P. (1988). *Memoirs of My Nervous Illness*. Cambridge, Massachusetts: Harvard University Press.

Schreiber, T. (2000). Measuring Information Transfer. *Physical Review Letters* 85(2): 461-4.

Searle, J.R. (1980). Minds, Brains and Programs. *Behavioral and Brain Sciences* 3: 417-57.

Searle, J.R. (1992). *The Rediscovery of the Mind*. Cambridge, Massachusetts: MIT Press.

Searle, J.R. (2002). Why I Am Not a Property Dualist. *Journal of Consciousness Studies* 9(12): 57-64.

Seth, A.K. (2007). Causal networks in simulated neural systems. *Cognitive Neurodynamics,* in press.

Seth, A.K., Baars, B.J. and Edelman, D.B. (2005). Criteria for consciousness in humans and other mammals. *Consciousness and Cognition* 14: 119–39.

Seth, A.K. and Edelman, G.M. (2007). Distinguishing causal interactions in neural populations. *Neural Computation* 19(4): 910-33.

Seth, A.K., Izhikevich, E., Reeke, G.N. and Edelman, G.M. (2006). Theories and measures of consciousness: An extended framework. *PNAS* 103(28): 10799–804.

Shadlen, M.N. and Newsome, W.T. (1994). Noise, neural codes and cortical organization. *Current Opinions in Neurobiology* 4: 569-79.

Shanahan, M.P. (2006). A Cognitive Architecture that Combines Internal Simulation with a Global Workspace. *Consciousness and Cognition* 15: 433-49.

Shanahan, M.P. (2008). A spiking neuron model of cortical broadcast and competition. *Consciousness and Cognition* 17(1): 288-303.

Shanks, D.R. (2005). Implicit learning. In K. Lamberts and R. Goldstone, *Handbook of Cognition.* London: Sage, pp. 202-20.

Shear, J. (ed.) (1997). *Explaining Consciousness - The 'Hard Problem'*. Cambridge, Massachusetts and London: The MIT Press.

Shepard, R.N. (1962a). The analysis of proximities: Multidimensional scaling with an unknown distance function I. *Psychometrika* 27: 125-40.

Shepard, R.N. (1962b). The analysis of proximities: Multidimensional scaling with an unknown distance function II. *Psychometrika* 27: 219-46.

Shepard, R.N. (1966). Metric structures in ordinal data. *Journal of Mathematical Psychology* 3: 287-315.

Shu, Y., Hasenstaub, A., Duque, A., Yu, Y. and McCormick, D.A. (2006). Modulation of intracortical synaptic potentials by presynaptic somatic membrane potential. *Nature* 441: 761-65.

Silver, R., Boahen, K., Grillner, S., Kopell, N. and Olsen, K.L. (2007). Neurotech for neuroscience: Unifying concepts, organizing principles, and emerging tools. *Journal of Neuroscience* 27(44): 11807-19.

Singer, W. (2000). Phenomenal Awareness and Consciousness from a Neurobiological Perspective. In T. Metzinger (ed.), *Neural Correlates of Consciousness*. Cambridge, Massachusetts and London, England: The MIT Press.

Sloman, A. (1999). What Sort of Architecture is Required for a Human-like Agent? In M. Wooldridge, and A.S. Rao. (eds.), *Foundations of Rational Agency*. Dordrecht, Netherlands: Kluwer Academic Publishers.

Sloman, A. (2006). Why Asimov's three laws of robotics are unethical. Retrieved on the 6th December 2006 from http://www.cs.bham.ac.uk/research/projects/cogaff/misc/ asimov-three-laws.html.

Spinoza, B. de (1992). *Ethics*. London: J.M. Dent & Sons Ltd; Rutland, Vermont: Charles E. Tuttle Co., Inc.

Sporns, O. (2007) Brain Connectivity. *Scholarpedia*, Art. #4695. Retrieved 4th December 2007 from: http://www.scholarpedia.org/article/Brain_Connectivity.

Sporns, O., Chialvo, D.R., Kaiser, M. and Hilgetag, C.C. (2004). Organization, development and function of complex brain networks. *TRENDS in Cognitive Sciences* 8(9): 418-25.

Sporns, O., Karnowski, J. and Lungarella, M. (2006). Mapping causal relations in sensorimotor networks. *Proceedings of the 5th International Conference on Development and Learning*.

Steels, L. (2001). Language games for autonomous robots. *IEEE Intelligent Systems and Their Applications* 16(5): 16-22.

Steels, L. (2003). Language Re-Entrance and the 'Inner Voice'. In O. Holland (ed.), *Machine Consciousness*. Exeter: Imprint Academic.

Stening, J., Jacobsson, H. and Ziemke, T. (2005). Imagination and Abstraction of Sensorimotor Flow: Towards a Robot Model. In R. Chrisley, R. Clowes and S. Torrance, (eds.), *Proceedings of the AISB05 Symposium on Next Generation approaches to Machine Consciousness*, Hatfield, UK.

Stevens, S.S. (1946). On the Theory of Scales of Measurement. *Science* 103: 677-80.

Stuart, S. (2003). Artificial Intelligence and Artificial Life - should artificial systems have rights? Retrieved on the 6th December 2006 from http://www.gla.ac.uk/departments/ philosophy/Personnel/susan/NewNightmares.pdf .

Taylor, J.G. (2007). CODAM: A neural network model of consciousness. *Neural Networks* 20: 983–92.

Taylor, J.G. and Fragopanagos, N. (2007). Resolving some confusions over attention and consciousness. *Neural Networks* 20: 993–1003.

Thompson, E., Lutz, A. and Cosmelli, D. (2005). Neurophenomenology: An Introduction for Neurophilosophers. In A. Brook and K. Atkins (eds.), *Cognition and the Brain*. Cambridge: Cambridge University Press, pp. 40-97.

Thompson, E. and Varela, F.J. (2001). Radical embodiment: neural dynamics and consciousness. *Trends in Cognitive Sciences* 5(10): 418-25.

Tononi, G. (2004). An Information Integration Theory of Consciousness. *BMC Neuroscience* 5:42.

Tononi, G., Edelman, G.M. and Sporns, O. (1998). Complexity and coherency: integrating information in the brain. *Trends in Cognitive Sciences* 2(12): 474-84.

Tononi, G. and Sporns, O. (2003). Measuring information integration. *BMC Neuroscience* 4:31.

Tononi, G., Sporns, O. and Edelman, G.M. (1994). A measure for brain complexity: Relating functional segregation and integration in the nervous system. *Proc. Natl. Acad. Sci. USA* 91: 5033-7.

Torrance, S. (2005). Thin Phenomenality and Machine Consciousness. In R. Chrisley, R. Clowes, and S. Torrance (eds.), *Proceedings of the AISB05 Symposium on Next Generation approaches to Machine Consciousness*, Hatfield, UK.

Underwood, G. (1982). Attention and Awareness in Cognitive and Motor Skills. In G. Underwood (ed.), *Aspects of Consciousness, Volume 3, Awareness and Self-awareness.* London and New York: Academic Press, pp. 111-45.

Van Heuveln, B., Dietrich, E. and Oshima, M. (1998). Let's dance! The equivocation in Chalmers' dancing qualia argument. *Minds and Machines* 8: 237-49.

Varela, F. (1996). Neurophenomenology: A Methodological Remedy for the Hard Problem. *Journal of Consciousness Studies* 3(4): 330-49.

Varela, F., Lachaux, J.-P, Rodriguez, E., and Martinerie, J. (2001). The brainweb: Phase synchronization and large scale integration. *Nature Neuroscience* 2: 229-39.

Velmans, M. (1990). Consciousness, Brain and the Physical World. *Philosophical Psychology* 3(1): 77-99.

Velmans, M. (1991). Is human information processing conscious? *Behavioral and Brain Sciences* 14: 651-69.

Vogels, T.P. and Abbott L.F. (2005). Signal propagation and logic gating in networks of integrate-and-fire neurons. *Journal of Neuroscience* 25: 10786-95.

Wegner, D.M. (2002). *The Illusion of Conscious Will*. Cambridge, MA: MIT Press.

Wegner, D.M. (2003). The mind's best trick: how we experience conscious will. *TRENDS in Cognitive Sciences* 7(2): 65-9.

Wegner, D.M. (2004). Precis of The Illusion of Conscious Will. *Behavioral and Brain Sciences* 27: 649-92.

Wegner, D.M. and Wheatley, T.P. (1999). Apparent mental causation: Sources of the experience of will. *American Psychologist* 54: 480-92.

Wilkes, K.V. (1984). Is consciousness important? *British Journal for the Philosophy of Science* 35: 223-43.

Wilkes, K.V. (1988). ———, yìshì, duh, um, and consciousness. In A.J. Marcel, and E. Bisiach, *Consciousness in Contemporary Science*. Oxford: Clarendon Press.

Wilkes, K.V. (1995). Losing consciousness. In T. Metzinger (ed.), *Conscious Experience*. Paderborn: Ferdinand Schöningh.

Wordsworth, W. (2004). I Wandered Lonely as a Cloud. In S. Gill (ed.), *Selected Poems*, London: Penguin.

Yellin, A.M. (1986) Acquired Precise Volitive Cardiac Control. *Annals of the New York Academy of Sciences* 463(1): 362–5.

Zeki, S. (2003). The Disunity of Consciousness. *TRENDS in Cognitive Sciences* 7(5): 214-8.

Zeki, S. and Bartels, A. (1998). The asynchrony of consciousness. *Proceedings of the Royal Society B* 265: 1583-5.

Ziemke, T., Jirenhed, D.A. and Hesslow, G. (2005). Internal simulation of perception: a minimal neuro-robotic model. *Neurocomputing* 68: 85-104.

Zihl, J., Von Cramon, D. and Mai, N. (1983). Selective Disturbance of Movement Vision after Bilateral Brain Damage. *Brain* 106: 313-40.