

NEURAL MODELLING OF A JELLYFISH

JAMES YEOMANS

BSc (Hons) Computer Information Systems

University of Bath

May 2005

**PROJECT SUPERVISOR: Dr DANIEL
RICHARDSON**

This thesis may be made available for consultation within the University Library and may be photocopied or lent to other libraries for the purpose of consultation.

Signed

Abstract

Neural Networks are becoming increasingly important in the world of computing. They are considered to be a potentially powerful concept with many uses currently under research. The idea of a Neural Network is to imitate and utilise the functionality of a biological nervous system for computing purposes.

Jellyfish possess a seemingly very primitive nervous system as they have no brain however they display characteristics with complexity not normally associated with a creature believed to date back some 600 million years.

This project will investigate the jellyfish nervous system and attempt to produce a model of at least part of the system. The model developed displays some of the functionality associated with the nervous system and the functionality and limitations of the model are discussed. There is also a discussion of possible further work that might take place and how it might be undertaken.

Acknowledgments

I would like to thank the project supervisor Dr Daniel Richardson for his supervision and support throughout the project, the unit lecturer Dr Alwyn Barry for providing guidance on the writing of the dissertation, Dr William Megill for providing a great deal of insight into the jellyfish and Andrew Carnell for his help in using the CSIM software.

Neural Modelling of a Jellyfish

Submitted by James Yeomans

COPYRIGHT

Attention is drawn to the fact that copyright of this thesis rests with its author. The Intellectual Property Rights of the products produced as part of the project belong to the University of Bath (see <http://www.bath.ac.uk/ordinances/#intelprop>).

The copy of this thesis has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with its author and that no quotation from the thesis and no information derived from it may be published without the prior written consent of the author.

Declaration

This dissertation is submitted to the University of Bath in accordance with the requirements of the degree of Bachelor of Science in the Department of Computer Science. No portion of the work in this dissertation has been submitted in support of an application for any other degree or qualification of this or any other university or institution of learning. Except where specifically acknowledged, it is the work of the author.

Signed

Contents

1)	Introduction.....	7
1.1	Project Overview and Objective.....	7
1.2	Current Research into Jellyfish.....	7
1.3	The Animal Concerned.....	7
2)	Literature Survey.....	8
2.1	Structure of the Literature Review.....	8
2.2	Jellyfish.....	8
2.2.1	2 Different Types of Coelenterates.....	8
2.2.2	Basic Structure.....	8
2.3	Nervous Systems.....	9
2.3.1	Neurons.....	9
2.3.2	Cell Membranes.....	10
2.3.3	Cell Potential.....	11
2.3.4	Nerve Impulses.....	12
2.3.4.1	Depolarisation.....	12
2.3.4.2	Repolarisation.....	12
2.3.5	Synapses.....	12
2.3.5.1	Electrical.....	13
2.3.5.2	Chemical.....	13
2.3.5.3	Transmission at the Nerve-Muscle Synapse.....	13
2.4	Neural Networks in Computing.....	14
2.4.1	Spiking Neural Networks.....	15
2.5	Jellyfish Nervous System Overview.....	16
2.5.1	Ocelli.....	16
2.5.2	The INR.....	16
2.5.3	The ONR.....	16
2.5.3.1	The B-System.....	17
2.5.3.2	The O-System.....	17
2.6	Jellyfish Muscular Structure.....	18
2.7	Swimming Control.....	18
2.8	Swimming Muscle.....	19
2.9	Giants in the Inner Nerve Ring.....	21
2.9.1	Different Sized Jellyfish.....	24
2.10	Jellyfish and Spiking Neural Networks.....	24

2.11	Discussion.....	25
3)	The Model.....	26
3.1	The Concept.....	26
3.2	The Goal.....	26
3.3	Model Construction.....	26
3.3.1	Output.....	28
3.4	Experimentation.....	30
3.4.1	Varying Synaptic Time Delay.....	30
3.4.2	Varying Refractory Period.....	31
3.4.3	Varying Synaptic Time Delay Once Again.....	32
4)	Conclusions.....	33
4.1	Analysis of Model.....	33
4.1.1	Accuracies.....	33
4.1.2	Discrepancies.....	33
4.2	Possible Further Development.....	34
4.2.1	Two Directions for Further Development.....	34
4.3	Concluding Remarks.....	35
5)	Bibliography.....	36
5.1	Papers.....	36
5.2	Books.....	36
5.3	Web Sites.....	37
5.4	Further Reading.....	37
6)	Appendix A.....	38
A.1	Test Simulations.....	38
A.1.1	Model Construction.....	38
A.1.2	Varying Synaptic Time Delay.....	56
A.1.3	Varying Refractory Period.....	64
A.1.4	Varying Synaptic Time Delay Once Again.....	71
7)	Appendix B.....	76
B.1	Code for the model.....	76

1 Introduction

1.1 Project Overview and Objective

The initial idea for this project was to conduct some detailed research into the nervous system of a primitive jellyfish and to build a model, of at least part, of this nervous system. The literature review would help find and evaluate all the necessary information I would need to do this but also to refine the project idea from a very general one into an idea that was more specific i.e. it would bring the focus to one part of the nervous system in particular and would provide the foundation for the project. It was important to bear in mind the scale of the project with respect to the resources that were available and also the time that had been allocated to it.

At the beginning it was obvious I would have to research Jellyfish and nervous systems. This provided the starting point and as the project went on I was able to narrow down the scope of the project and define the clear goals that it was going to try to achieve.

1.2 Current Research in to Jellyfish

There has been a lot of research into the neural system of jellyfish, probably because of the simplicity of the creature and their suitability for biological experiments. As far as I am aware there are no computer models available that emulate their neural behaviour and so the aim of this project is to produce one.

1.3 This Animal Concerned

There are too many species of jellyfish to produce a universal neural model so I have narrowed it down to 1 specific species, *Polyorchis penicillatus*. This appears to be the jellyfish which has been involved in the most research and I was able to find considerable information about this particular species. *Polyorchis penicillatus* is pictured below:

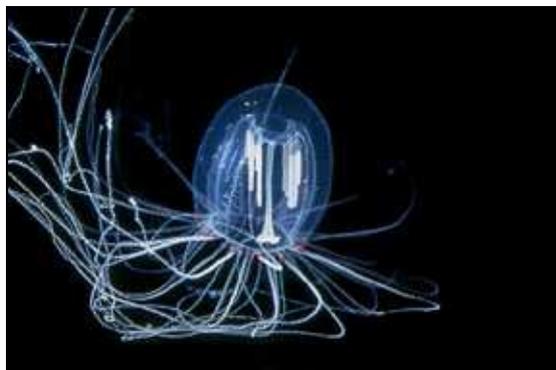


Fig 1. Polyorchis penicallatus. The main structures are clearly visible.

2 Literature Survey

2.1 Structure of the Literature Survey

I have tried to give the literature review a structure consistent with that of my views about the project i.e. the important areas to investigate and the order in which to research them. After a little bit of background of information about the jellyfish the next section is concerned with the biology of nervous systems. I then talk about neural networks in computing as that is what the project is really concerned about, building a computer based neural model, and then the nervous system of the jellyfish itself and how its structure relates to its functionality. Then the idea of using computer based neural networks to build a neural model of a jellyfish and how I intend to go about this.

2.2 Jellyfish

I have come across various names given to jellyfish. They are also known as Coelenterates, Cnidarians, and Hydro-medusae. Coelenterates and Cnidarians are basically different words for the same family.

2.2.1 Two Different Types of Coelenterates

The 2 members of the Coelenterate family are Polyps and Medusa. This project is not concerned with the polyps but only with the medusa side of the family which covers the jellyfish.

Fossils have shown that jellyfish have been in existence for over 600 million years. They have a bad reputation with a lot of people as they are associated with attacking and stinging, however this is limited to only a few species from the Hydrozoa, Scyphozoa and Siphonophora families (Bayer and Owre 1968). Probably the best known of these is the infamous Portuguese Man of War belonging to the Siphonophora family.

2.2.2 Basic Structure

In order to understand how the neural system of a jellyfish works I think it is first important to understand their basic structure and how it relates to their functionality.

Polyorchis is an example of a radially symmetrical coelenterate.

The body of a coelenterate is a continuous smooth surface. They lack cephalization, that is they have no central nervous system or brain. They are composed of 3 systems, digestive, nervous and sensory which are made up from 3 different types of tissues, muscular, epithelial and connective. They have tentacles which are used for catching food and it is these that contain the thread capsules that are used for stinging (nematocysts). Medusa are a free living animal usually composed of a transparent bell shaped umbrella inside which can be found the basic organs that make up their primitive structure (Bayer and Owre 1968).

The picture below illustrates the structure of a jellyfish:

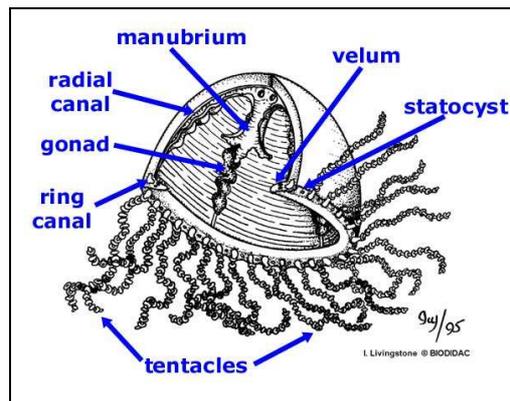


Fig 2. Diagram to show the main structures in a jellyfish like the one being investigated. Picture from: <http://www.esu.edu/>

In between the exumbrellar epithelium (outer surface of the bell) and the gastrodermal lamella is the bell mesoglea. This is a thick, stiff jelly like substance that gives the jellyfish its structure. There is another inner layer known as the joint mesoglea in between the gastrodermal lamella and the subumbrellar epithelium (the inner surface of the bell), (Megill 2002). The manubrium hangs from the center of the bell and is the stomach like structure of the jellyfish, this terminates in the mouth at the base of the bell (Bayer and Owre 1968). The 4 radial canals extend from the gastric cavity to the marginal ring canal at the base of the velum (Megill 2002). A statocyst is a small fluid filled sac located around the base of the bell that are used for balance. At the base of the bell are 2 nerve rings, the Inner (INR) and Outer (ONR) nerve ring (Spencer and Arkett 1983).

2.3 Nervous Systems

The primitive nervous system found in a hydrozoan jellyfish is different to that found in most other animals mainly because it has no brain and because it is an invertebrate, no spinal cord. It does, however, have a lot in common.

2.3.1 Neurons

Nervous systems in both vertebrates and invertebrates are made up of specialised cells called neurons. I decided not to conduct a lot of research into cell biology as it was not necessary to this project but I felt it was important to look at the structure of neurons as they must certainly play a part in the neural pathways of the jellyfish. A basic neuron is shown on the next page:

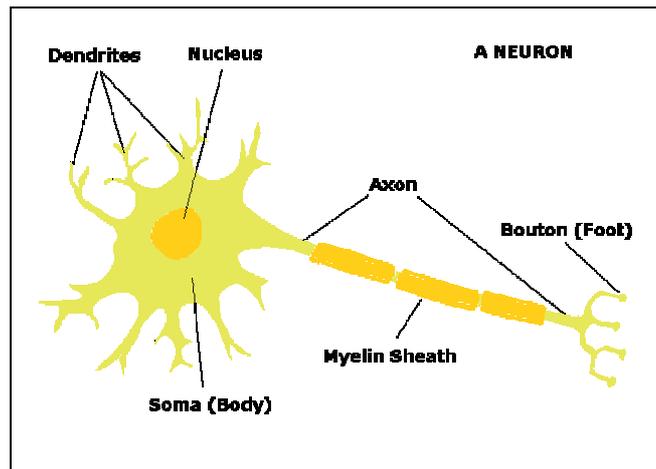


Fig 3. The diagram shows a simplified view of a neuron. In *Polyorchis* the axon is unmyelinated. Picture from <http://www.ship.edu/>

The picture is of one neuron cell. A nervous system is made up of many neurons. Raven and Johnson (1992) describe the structure and function of a neuron: The cell body, or Soma, is the metabolic center of the cell, it contains the nucleus, which is the organelle in which the genes are contained as well as all the other microstructures that would be found in any cell body. The soma also gives rise to the 2 other important parts of a nerve cell, the dendrites and the axon. The dendrites branch out in a tree like fashion and they receive signals from other nerve cells. The axon that extends from the cell body is the structure that passes on signals to other cells, although there is only 1 axon per cell it may split into several branches and form junctions with a variety of other cells. Nerve impulses can travel along the axon at up to 100m/s in some animals. In most creatures they would then arrive at the brain where they would be interpreted, however jellyfish do not contain a brain so it was not necessary to research this part of the nervous system. Large axons (they can grow up to 2 meters long, although not in jellyfish) are sometimes insulated by a fatty insulating material, the cover that this forms is known as the myelin sheath. It does not completely cover the axon, instead it leaves gaps at intervals along the axon known as Nodes of Ranvier. These Nodes of Ranvier allow the impulse being conducted to travel at a greater velocity than they otherwise could. The neurons in a jellyfish however, are not covered by a myelin sheath.

2.3.2 Cell Membranes

Before I could look at how nerve impulses were propagated along and through nerve cells it was necessary to look at the structure and properties of cell membranes. The structure of a cell membrane is shown on the next page and the important features are described in Vilee (1977):

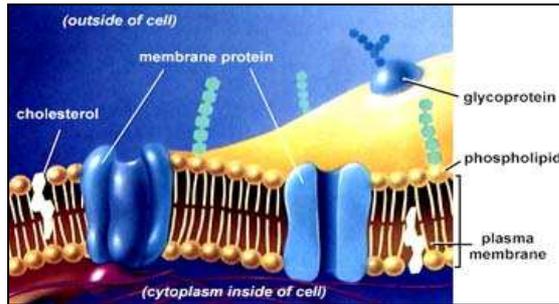


Fig 4. The picture shows the basic components of a cell membrane including the very important ion channels. Picture from www.emc.maricopa.edu

The important part as far as the project is concerned is the membrane protein as it is these that play a vital role in the conduction of nerve impulses. Membrane proteins form channels through phospho-lipid layer that makes up most of the membrane and it is through these channels that certain molecules and compounds can move. Different proteins allow different molecules to move through them, they are selectively permeable, this is very important with respect to the propagation of nerve impulses.

2.3.3 Cell Potential

Neurons and in fact a lot of cells are able to maintain an electrical gradient between the inside of the cell compared to its surroundings, this is known as the cells resting potential. Curtis (1975) describe how cells are able to maintain this electrical gradient: They do this by keeping a larger concentration of positive ions on the outside of the cell compared to the inside, effectively making the inside of the cell negatively charged. Cell membranes have many protein channels including ones that are selectively permeable to Potassium (K^+) ions, protein channels that are permeable to Sodium (Na^{++}) ions and they also have protein channels known as sodium-potassium pumps. These sodium-potassium pumps are responsible for maintaining the resting potential of the cell. They move Na^{++} ions out of the cell and pump K^+ in. Na^{++} ions are kept at about a concentration about 10 times higher outside the cell compared to the inside and K^+ ions are kept at a concentration of 20 times higher inside compared to outside (Kandel et al. 1992). The K^+ permeable protein channels in the membrane remains open allowing the natural diffusion of K^+ ions out of the cell down its concentration gradient and this combined with the pump maintains a positive charge outside the cell. The electrical potential difference of the inside of the cell compared to the outside is maintained at about $-70mv$. The diagram below shows how the balance is maintained by the pump:

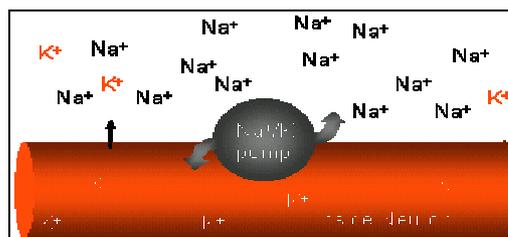


Fig 5. Diagram to show how the resting potential of a cell membrane is maintained by the sodium potassium pump. As well as the pump the natural diffusion of K^+ out of the cell maintains the resting potential. Picture from www.faculty.clintoncc.suny.edu

2.3.4 Nerve Impulses

It is the ability of neurons to change their resting potential that gives them the ability to conduct nerve impulses (Kandel et al. 1992). A nerve impulse travelling through the axon is known as an action potential. For an action potential to be created the cell must depolarise to the threshold value at which an action potential can occur. If this threshold value is not reached then an action potential will not occur (Curtis, H 1975). The propagation of the nerve impulse can be described as the following phases (from Raven and Johnson 1992):

2.3.4.1 Depolarisation

- 1) The membrane suddenly becomes highly permeable to Na^{++} causing a fast influx down the concentration gradient.
- 2) This causes the inside of the cell to become positively charged with respect to the outside. The action potential is then propagated down the axon because the increase in positive charge causes Na^+ channels to open in the neighbouring region of the axon and the depolarisation process starts there.
- 3) The efflux of K^+ continues but it is largely masked by the Na^{++} influx. This will become an important feature later when looking at the conduction of impulses in *Polyorchis*.
- 4) The balance of Na^{++} reaches a state of equilibrium and the potential difference across the membrane is now at about +40mv (inside compared to outside).

2.3.4.2 Repolarisation

- 1) Na^{++} channels gradually become inactivated.
- 2) K^+ channels open after a very slight delay which causes a net efflux of positive charge from the cell.
- 3) The Sodium-Potassium pump gradually restores the balance of the cell resting potential.
- 4) The repolarisation process overshoots the resting potential of the cell slightly, temporarily causing that part of the axon to become inactivated.

Action potentials involve the expenditure of energy, it costs energy in order to restore the resting potential of the cell and I will show later on one instance where I think the jellyfish conserves energy by the way in which it produces action potentials in response to repetitive stimulation. The duration of an action potential is only fractions of a second, although the exact length varies between different types and species of animal.

2.3.5 Synapses

The point at which the axon of a neuron joins with a dendrite of a neighbouring cell is called a synapse, and is shown in the following picture:

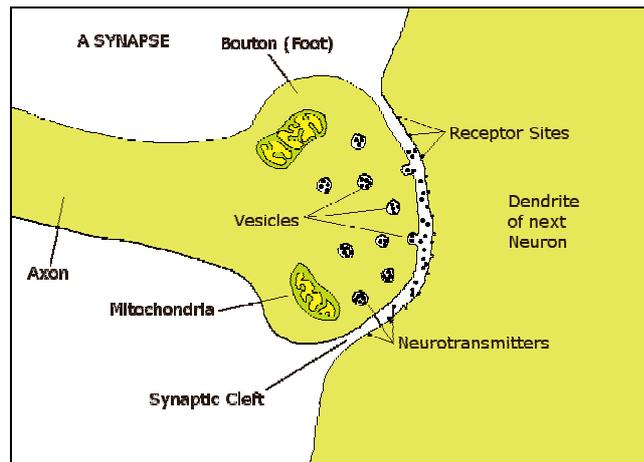


Fig 6. The diagram shows a simplified structure of a chemical synapse. Picture from www.ship.edu

There are 2 types of transmission between nerve cells, electrical and chemical and they are described in Kandel et al. (1992).

2.3.5.1 Electrical

This is the simplest type of transmission. It is much faster than chemical transmission. The junctions between the 2 cells are known as gap junctions. There are ion channels in the pre and post-synaptic membrane and the action potential is transmitted across the synaptic cleft in the same way that it is propagated down the axon of a neuron. There may also be some cytoplasmic continuity between the cells. It is electrically coupled networks of neurons that are of great importance in the nervous system of *Polyorchis*, although there are chemical connections as well, it is the electrical coupling that provides the most important functionality.

2.3.5.2 Chemical

This is the most common type of synapse and occurs across larger synaptic clefts. It is more complex than electrical gap junctions as it can alter the amplitude of the action potential, this property is an important one to consider when looking at the jellyfish nervous system. When the action potential arrives at the pre-synaptic membrane it causes vesicles containing a transmitter molecule to fuse with the membrane and release the transmitter molecule into the cleft. The molecules bind with receptor sites on the post-synaptic membrane and cause depolarisation and the creation of an action potential.

It is important to bear in mind that it is not only neurons that are capable of propagating a nerve impulse. There are 3 types of cell that are capable of being excited: Muscular, Neuronal and Epithelial (Spencer 1977). This became an important fact as the study of the jellyfish went on, most importantly it was important to look at the transmission of an impulse between nerve and muscle cells.

2.3.5.3 Transmission at the Nerve-Muscle Synapse

This is done chemically and is described in Bayer and Owre (1968). The action potential arrives at the pre synaptic terminal and causes calcium (Ca^{++})

channels to open and calcium ions flood in. This causes the vesicles with the transmitter molecule to fuse with the membrane. The molecule is consequently released in to the synaptic cleft where it binds with receptor sites on the muscle cell causing a depolarisation of the cell which can then be propagated across all the cells of the muscle sheet. The important fact to remember here is the amount of transmitter molecules released is proportional to the peak rate of influx of Ca^{++} ions into the pre-synaptic terminal (Spencer 1993).

Thousands of neurons can link together via thousands of synapses to form a giant integrated network. Neurons do not just link together in a linear way, indeed they may form connections taking impulses from thousands of other neurons and passing them on to thousands more. A spike in such a network is the response produced after a stimulus is received (what would also be called an action potential) which may then propagate around the network.

The neural network in a jellyfish is considered quite simple due to the relatively small amount of neurons it contains, however it does show some very interesting and significant properties, that would not normally be expected from such a primitive creature.

2.4 Neural Networks in Computing

The purpose of neural networks in computing is to simulate parts of the central nervous system, in particular the brain and the functions that it performs such as learning and pattern recognition (amygdala.sourceforge.net). It is common to refer to these “Artificial Neural Networks” as a neural network or neural net (www.wikipedia.org). Neural networks have been built that exhibit some of the functions of the brain but they cannot be considered to be anywhere near as complex as the brain with the largest being made up of about 10^3 artificial neurons compared to the 10^{10} neurons believed to make up the human brain (amygdala.sourceforge.net).

In the first neural networks, neurons were modelled as binary neurons which could have the value active or inactive, and the value was calculated by summing the values of all the neurons connected to it (S.Bohte 2003). Even with this relatively simple model it was still possible to implement simple learning algorithms (S.Bohte 2003). In neural networks neurons are known as nodes and the network is made up of many of these connected together. The nodes are either input, hidden or output nodes and there are connections between them(www.wikipedia.org). It seems that these connections are representing synapses. The following diagram shows how the different types of nodes are interconnected:

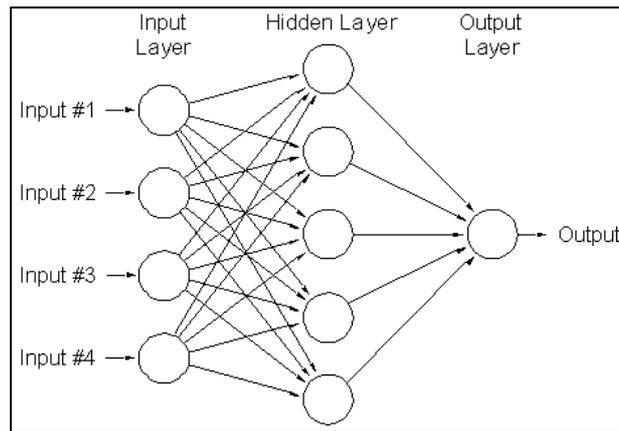


Fig 7. Each circle in the diagram represents a node of one type. The inputs pass into an input node and are passed onto the next node until the output is reached. Picture from www.wikipedia.org.

The basic technique for calculating the output from an input is by passing the input on to all the nodes that the input node is connected to multiplying by the weight placed on the connections between the nodes and so on until the output is reached. This is very similar to the original binary model of a neural network as described above. Uses for neural nets include being used to forecast costs and sales and they can also act as expert systems by simulating the thinking of an expert (www.promland.com). In order to build a simulation of part of a jellyfish neural system it will be necessary to use a more advanced type of neural net known as a spiking neural network.

2.4.1 Spiking Neural Networks

The problem with standard neural network models is that they are not very dynamic. They use a timing function that only allows signals to be sent between certain time intervals and they assume that the amplitude of the signal stays the same in between these timings (amygdala.sourceforge.net). This is of course not the way in which the brain behaves, it is much more complex than that. Spiking Neural Networks try to create a more accurate representation (amygdala.sourceforge.net). The transformation that occurs between nodes is usually done in a non-linear way, usually by a sigmoidal curve, which is necessary for solving complex problems, again, however, this is not sufficient for representing neurons in a real nervous system as transformation do not occur in that way, a neuron is not capable of performing complex multiplication and normalisation methods, it is spiking neurons that perform a more realistic operation (<http://www.ivorix.com>). So what is a spiking neuron? Firstly the way in which they are different from the other artificial neurons:

- 1) A spiking neuron will only fire if an input/stimulus crosses a certain threshold (<http://www.ivorix.com>). This is much more like real neurons as they are required to depolarise to a threshold value before they initiate an action potential.
- 2) They use an optimum time delay in between receiving an input and giving an output (<http://www.ivorix.com>). This again seems more like real neurons as chemical synapses do have a slight delay in transmitting the signal between neurons.

- 3) After spiking the neuron will resist any input signals for a short period of time so as to inhibit the creation of another spike very quickly after the previous one (<http://www.ivorix.com>). Just like neurons making up a central nervous system when the action potential repolarisation very briefly overshoots the normal resting potential inhibiting the creation of another action potential.
- 4) Also spiking neurons do not fire continuously like previous artificial neurons do (<http://www.ivorix.com>) they fire and then wait for the threshold value to be reached again by another stimulus, just like a real neuron.

With an understanding of neural networks and how a nervous system works I next decided to look at the structure of the nervous system in *Polyorchis penicillatus*.

2.5 Jellyfish Nervous System Overview

Hydro-medusae have 2 nerve rings at the base of the velum and sub-umbrella known as the Inner Nerve Ring (INR) and Outer Nerve Ring (ONR). The INR is located on the sub-umbrella side (inside) of the bell at the base of the velum (Spencer 1980) and the ONR is located on the ex-umbrella side of the bell (outside) (Spencer and Satterlie 1982).

2.5.1 Ocelli

These are the photic receptors of the jellyfish and are located towards the top of the tentacles. They are normally red in colour and provide the stimulus to neurons that connect into the nerve rings. It was beyond the scope of this project to investigate exactly how the ocelli work.

2.5.2 The INR

Within the INR are 2 networks of neurons, one of these networks is made of giant neurons that form a large anastomosing network of electrically coupled neurons (Spencer 1978). It is believed that these giant neurons are those that control the swimming function of the jellyfish (Spencer and Satterlie 1979). Within the INR there are normally between 6 and 15 giant neurons and up to 500 smaller ones (Spencer and Satterlie 1979). The function of the remaining neurons in the INR is still unclear to me. The radial nerves contain a network of neurons connected to the INR (Spencer and Arkett 1983). There is an unidentified excitatory pathway between the INR and the O-System of the ONR.

2.5.3 The ONR

The ONR also contains 2 separate networks known as the B-system, located in the center of the nerve ring, and the O-system (Spencer and Arkett 1983).

2.5.3.1 The B-System

The B-system is a network of electrically coupled neurons and is directly connected to the ocelli as stimulation of these, by turning off a light source, produces rapid bursts of impulses. These impulses can have a maximum amplitude of 10mv, occur within 300ms of the stimulus and produce multiple excitatory post-synaptic potentials (epsp's) within 5-8ms in neurons and epithelial cells overlying the ONR (Spencer and Arkett 1983). It is possible, although not conclusive that it is this system that performs the pacemaker function of the nervous system by regularly producing spikes that stimulate the swimming motor neurons (Spencer and Arkett 1983 reference Passano 1965, Mackie 1975 and Spencer 1978). Between 1 and 6 spikes are produced at regular intervals which as well as stimulating the motor neurons can be correlated with longitudinal muscle contraction in the tentacles (Spencer and Arkett 1983). The neurons have a mean resting potential of -40mv and the action potentials, which have a mean duration of 5 seconds, have an amplitude between 75mv and 80mv (Spencer and Arkett 1983). The spikes are followed by a 12mv to 20mv hyperpolarizing phase that can delay further action potentials for up to 70ms (Spencer and Arkett 1983). Pairs of spikes can arise spontaneously however larger numbers of spikes are usually associated with photic stimulation from the ocelli.

2.5.3.2 The O-System

The O-system is an anastomosing network of neurons that spread across the width of the outer nerve ring and has projections extending out towards the ocelli at the top of the tentacles (Spencer and Arkett 1983), although I have found no confirmation that they actually reach the ocelli. Spontaneous electrical activity occurs within the O-system although action potentials have not actually been recorded (Spencer and Arkett 1983). The activity of the network is not always the same. There are periods when the membrane can be seen to be producing regular oscillations and other periods when the membrane potential remains fairly constant. Membrane potentials between -40mv and -75mv have been recorded and it is these that affect the oscillations that occur. If the resting potential is more negative than -55mv then oscillations are of a lower amplitude than if it is more positive than -55mv (Spencer and Arkett 1983). The mean frequency of the oscillations is 0.9Hz with an average duration of 770ms (Spencer and Arkett 1983). Spontaneous changes do occur within the oscillation pattern but they are usually due to changes in the lighting, for example a rapid light reduction completely hyperpolarizes the entire network and the oscillations stop, however it normally returns to its regular firing pattern between 10s and 2 minutes afterwards even if the light is not returned to normal (Spencer and Arkett 1983). An increase in light will cause the oscillations to alter to a very high frequency and amplitude for a period of about 3s to 8s (Spencer and Arkett 1983). This looks to me like it could cause swimming activity in the jellyfish but there is no confirmation that the O-system actually links to the swimming motor neurons (SMN's). Despite the uncertainty of the connection between the O-system and the SMN's there is sometimes a direct correlation between the oscillations in the O-System and the activity of the SMN's, however the activity of the O-system cannot be essential for swimming because they have been observed at rest whilst the SMN's have been firing (Spencer and Arkett 1983). If these neurons do connect with the

ocelli then this is not their only source of input as removing the ocelli does not cause any change in the way they react to light (Spencer and Arkett 1983).

It was obviously not possible, with the time-scale and resources available, to build a model of the entire nervous system of the jellyfish. Before deciding which part to focus on I had a look at the muscles of the jellyfish.

2.6 Jellyfish Muscular Structure

The swimming muscle of a jellyfish is a single layer of epithelial cells that line the cavity of the bell (Spencer 1993). There are varying views regarding the structure of this muscle, it can be viewed as 4 separate muscle sheets separated by the 4 radial canals (Spencer and Satterlie 1981) or as 1 single continuous muscle sheet (Dr Megills thesis references Spencer 1979). This swimming muscle is an example of striated muscle, which is known to have evolved in muscles that require fast twitch like contractions (Spencer and Satterlie 1979). Polyorchis also has muscles in the velum at the base of the bell. These are also involved in the swimming action of the jellyfish. Longitudinal muscles are located in the tentacles and are used in the contraction of the tentacles during feeding (Megill 2002). Radial muscles along the radial canals and are used in the crumpling response of the jellyfish (Megill 2002). Endodermal muscles are associated with the manubrium and the digestive system (Megill 2002).

Because there appears to be most information about, and it is what interests me the most, I have decided to look at the group of neurons associated with controlling the swimming muscle of the jellyfish.

In order to model the structure and function of these neurons I decided to first take an overall look at how the swimming is controlled and how the muscles work.

2.7 Swimming Control

Propulsion in a jellyfish is achieved by the contraction of the subumbrella muscle sheet and consequently ejection of water from the subumbrella cavity (Spencer and Satterlie 1981). More power is achieved by the contraction of the muscles in the velum during the jet stroke (Megill 2002 references Gladfelter 1972).

The swimming muscle sheet is required to contract throughout simultaneously in order to achieve a smooth stroke (Spencer 1980).

The swimming muscle is controlled by the giant neurons in the INR (Spencer 1979). Neuromuscular synapses are found between these neurons and the muscle sheet throughout the margin of the bell and also along the radial canals (Spencer 1978). This means that the muscle sheet can be innervated at multiple points around the margin and the radii (Spencer 1977).

I find it interesting, but not surprising, that swimming is controlled by a group of electrically coupled neurons that have a distinct speed advantage over chemically coupled neurons when it comes to the transmission of action potentials. At first it seemed likely that a faster nerve impulse would result in a shorter delay to contraction of the swimming muscles thus allowing the jellyfish to respond quicker to any

stimulus, i.e. get out of the way of any perceived threat, but considering it would probably only be a matter of milliseconds quicker this does not really provide any advantage, so they probably evolved like this for another reason. There must be some other advantage to having this nerve impulse conducted so quickly around the swimming motor neurons.

2.8 The Swimming Muscle

The swimming muscle in the subumbrella cavity and the velar muscle do not have a nerve net of their own (Spencer 1977), however muscle cells themselves are excitable and capable of conducting impulses throughout the cells making up the muscle sheet (Spencer 1977) once they have been stimulated at the margin by the swimming motor neurons. Excitation spreads from cell to cell via gap junctions electrically coupling them and so it is fairly fast (Spencer 1993, Spencer and Satterlie 1981). The cells in the subumbrella swimming muscle are arranged in a brick like pattern which enhances the spread of the impulse in the circular direction (Spencer 1993), as the impulse has less gap junctions to cross.

This project is not directly concerned with the structure and functionality of the swimming muscle so will not go into any more detail.

Diagram of Neuronal Integration in Polyorchis

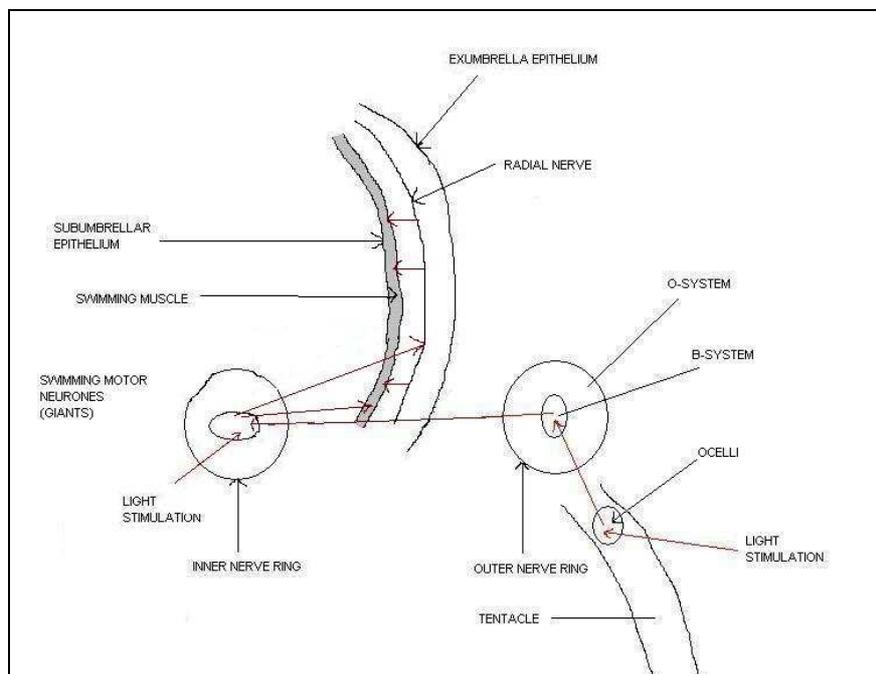
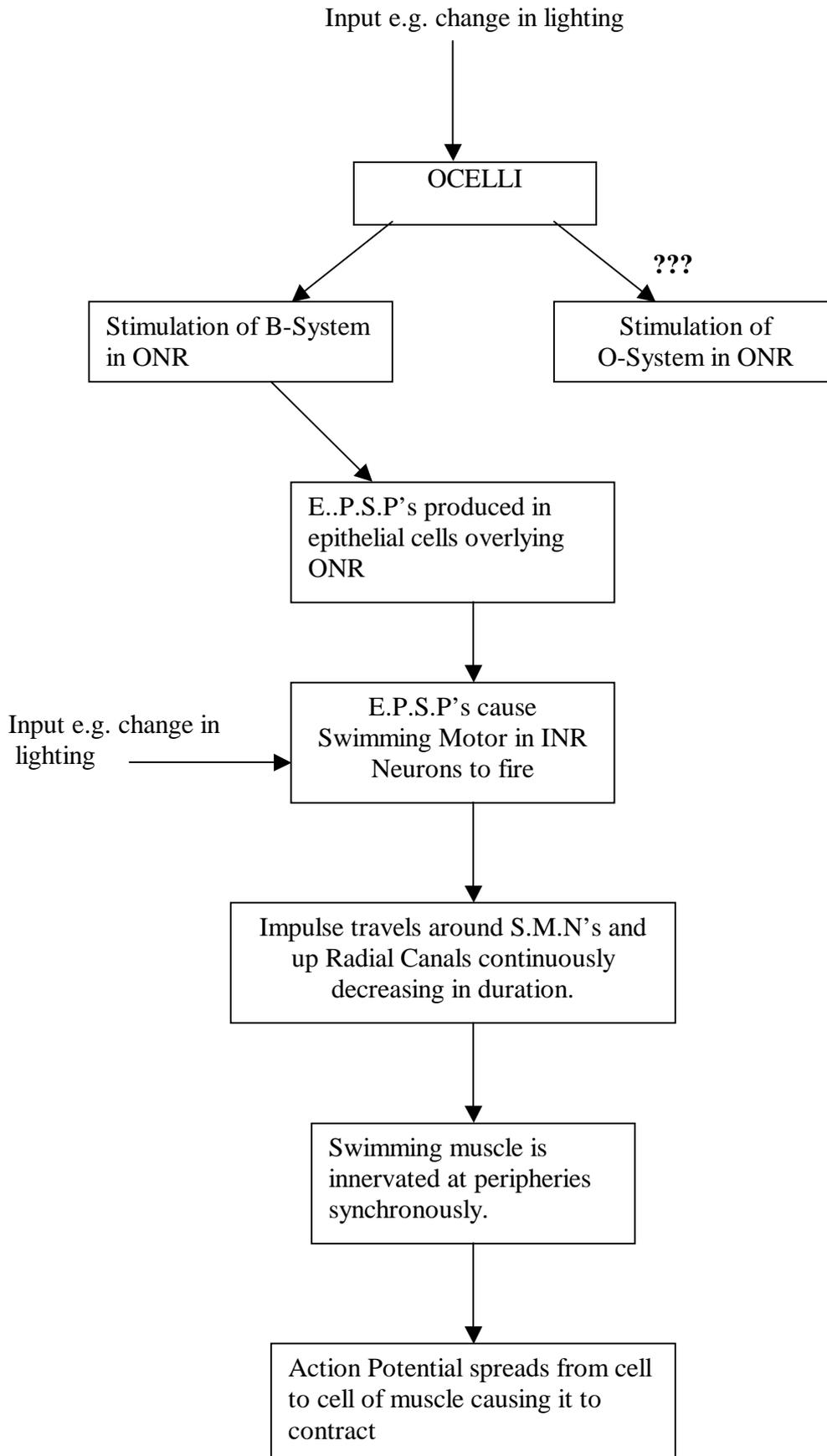


Fig 8. Simplified diagram to show neuronal pathway of muscle excitation, not to scale. The separate structure have been pulled apart in a cross sectional view. The only stimulation shown here is from a light source and the spread of excitation is shown by the red arrows. The swimming muscle is shaded in grey and is on the inside of the bell.

Simple Model of Swimming Control in Polyorchis



To understand how simultaneous and synchronised innervation of the muscle sheet is achieved it is important to understand the structure of the network of giant neurons and how it relates to its functionality, the aim of this project

2.9 Giants in the INR

The giants in the INR (also known as the swimming motor neurons, SMN's) form a network of very tightly electrically coupled neurons in which membrane potential oscillations and action potentials are synchronized throughout the network (Spencer 1979), the network effectively acts as a single neuron. This is very important when considering the importance of innervating multiple regions of the swimming muscle simultaneously because electrical synapses are a lot faster at conducting impulses than chemical ones. The SMN's receive excitatory input from the ocelli via the B-system of neurons in the ONR which are directly photosensitive and has also been suggested to have neural connections with the O-system (Spencer and Arkett 1983). The giants do not synapse directly with the muscle sheets of the subumbrella and velum, instead they form chemical synapses with overlying epithelial cells that form a continuous sheet with the muscle cells (Spencer 1993). If the neurons are firing at almost exactly the same time then the muscle sheet will be innervated around the margin at almost the same time, however there is a much more sophisticated approach to ensure that the muscle contracts in a synchronised fashion. There is a compensation for the time taken for the impulse to conduct around the margin to the opposite side of the bell (Spencer 1993). As the impulse travels through the network there is a constant decrease of the duration of the action potential and it has been shown that a shorter duration action potential in the pre-synaptic terminal (the SMN) leads to a greater amplitude of potential in the post-synaptic muscle cell and so consequently the time to muscle contraction will be quicker (Spencer 1993). The time delay to muscle contraction is directly proportional to the duration of the pre-synaptic action potential (Spencer 1993). So basically as the impulse travels round the network it becomes more efficient at stimulating the muscle sheet. In post-synaptic terminals the amplitude of the action potential generated is proportional to the number of transmitter molecules that are released from the pre-synaptic terminal, and the number of transmitter molecules released is proportional to the peak rate of influx of Ca^{++} ions in the pre-synaptic terminal (Spencer 1993). An influx of Ca^{++} does not happen until the repolarisation phase of the action potential (Spencer 1993). Because shorter duration action potentials have a more rapid repolarising phase there is a very large inward driving force on the Ca^{++} ions (Spencer 1993), consequently more transmitter molecules are released and the potential in the post synaptic cell is increased. The following diagram shows how the change in duration affects the rate at which calcium flows in, and consequently how many transmitter molecules are released.

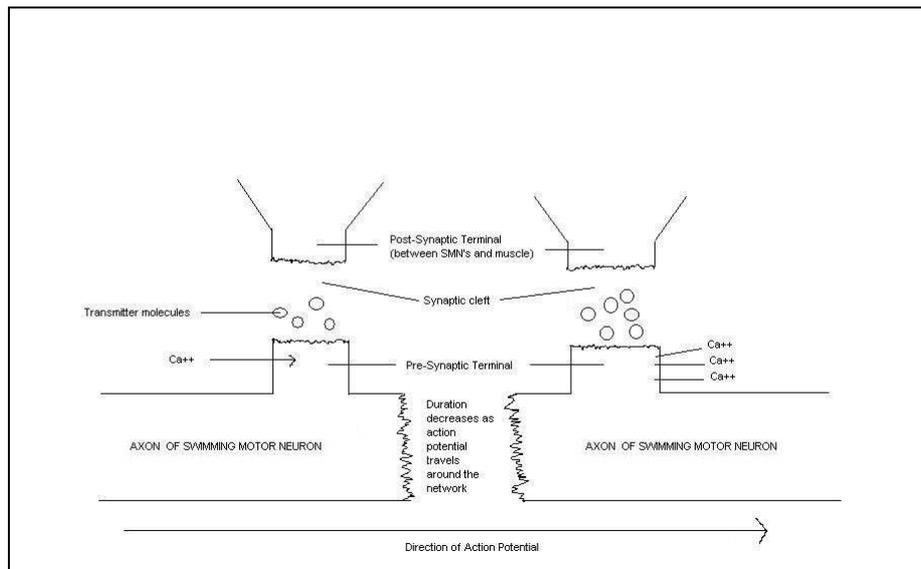


Fig 9. Diagram to show how the number of transmitter molecules released from the pre-synaptic junction is proportional to the peak rate of calcium influx. In this diagram the Ca^{++} does not relate to the amount of calcium that flows in, but to the peak rate in which it flows in.

This explains how larger amplitude post-synaptic action potentials are produced by shorter duration pre-synaptic action potentials but it is important to understand how the duration of the action potential constantly decreases as it travels round the margin.

At the point of origin in the INR the duration of the action potential is usually between 25-40ms, but as it propagates around the margin it gradually decreases until at its furthest point from the origin, at the opposite side of the bell, it is about 7ms Spencer (1993). So how does it do this? Spencer et al. (1989) give an explanation: The rate at which K^+ leaves the cell during repolarisation is proportional to the degree of polarisation in the cell membrane. This current is known as $I_{(k-fast)}$. It follows that the greater the polarisation of the membrane then the faster $I_{(k-fast)}$ will flow and the quicker the polarisation will be i.e. the shorter the duration of the action potential. The steady-state inactivation curve for $I_{(k-fast)}$ shows how its rate increases as the polarisation increases and is shown below.

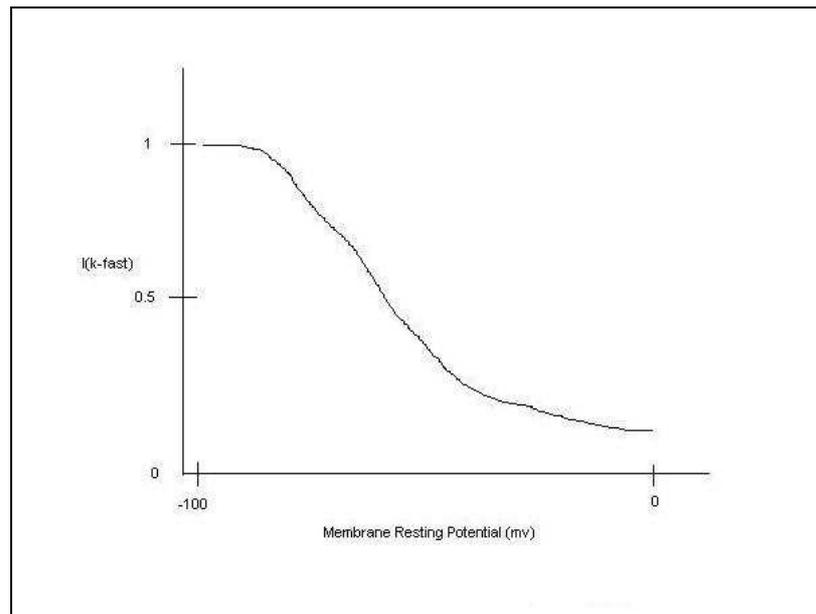


Fig 10. Diagram to show how the degree of $I_{(k-fast)}$ varies with the size of the polarisation in the membrane. This graph is also known as the state inactivation curve for $I_{(k-fast)}$. Redrawn from Spencer et al. (1989).

When the action potential is stimulated the resting potential is relatively depolarised, that is towards the right hand side of the graph and so $I_{(k-fast)}$ is relatively slow and the action potential has a longer duration. As further more polarised regions are stimulated by the action potential the degree of $I_{(k-fast)}$ increases and the duration decreases, hence the duration is continually decreasing as the action potential travels around (Spencer et al. 1989). This offered up the question of why further regions of the network tend to be more polarised, i.e. have a more negative resting potential. The answer is related to the electrical coupling of the network. The place in which the action potential initiates will be the area that is most depolarised, and therefore closer to the firing threshold required for an action potential to initiate. Because of the nature of electrically coupled networks the most hyperpolarized part of the network must be the place that is exactly opposite this initiating part of the ring, and because the action potential travels around the ring in both directions it is constantly travelling into regions that are more hyperpolarized until it reaches this point (A.Spencer, pers. comm. 12th December 2004). This leads me to wonder why not all neural networks act in this way, I suspect it is due to the nature of chemical synapses that are more common than electrically coupled cells where the electrical activity of the network is not synchronised in the same way. Maybe electrically coupled networks evolved precisely for situations that require a change in the duration of the action potential as it travels, i.e. where synchronous muscle contractions are required.

So it can be seen that the action potential duration is constantly reduced as it travels through the network and consequently how this produces a greater amplitude of action potential in the cells which it stimulates. This feature allows simultaneous contraction throughout the regions the swimming muscle and it is for this reason that it is so important that action potentials are conducted at a fast rate around the swimming motor neurons, were they conducted at a slower rate then the synchronisation would probably be harder to achieve and this is probably why electrically coupled neurons are present as opposed to chemically coupled ones. This

is an incredible feature of a jellyfish such as this, for an animal believed to be hundreds of millions of years old it appears to be a very advanced characteristic, certainly not in line with the primitiveness normally associated with such a creature. Without the ability to swim in a synchronised manner the swimming abilities of the animal would have been severely limited making it a much easier target for any predators. It is possible it may have had to develop such a functionality in order for the species to survive.

It is interesting that repetitive stimulation of the giants causes a longer duration action potential, and lower amplitude, in the epithelial cells overlying the ring canal (Spencer 1977). This presumably leads to longer duration action potentials in the swimming muscle and hence longer contractions of the swimming muscle. I suspect that these longer contractions are more efficient than multiple shorter contractions both in the expulsion of water from the sub-umbrella cavity and also in the expenditure of energy as fewer muscle contractions will result in a smaller energy loss. If the amount of stimuli received by the jellyfish are proportional to the danger it is in, which would make sense, then this means they can respond efficiently to a dangerous situation, a good feature for survival.

2.9.1 Different Size Jellyfish

There is no direct correlation between the size of jellyfish and the amplitude of the action potential it generates (Spencer and Satterlie 1981). However there is a direct correlation between the size of jellyfish and the duration of the action potential, larger jellyfish produce a longer duration action potential (Spencer and Satterlie 1981). As I have already discussed longer duration action potentials mean a longer time to peak tension in the swimming muscle, i.e. a longer slower muscle contraction. This makes sense to me as I can see that larger jellyfish would have to have longer muscle contractions in order to displace more water from the subumbrella cavity.

2.10 Jellyfish and Spiking Neural Networks

As far as I am aware there are no current models available to demonstrate the neural behaviour of a jellyfish. That's not to say they haven't been made but any that have don't appear to have been made available. On the advice of the project supervisor, Dr Daniel Richardson, I will use the CSIM software package, which is used for building neural networks, to build a model of the swimming motor neurons in *Polyorchis penicillatus*. CSIM is an extension to Matlab and is based on a particular model of spiking neuron known as a "Leaky Integrate and Fire Neuron" (D.Richardson pers. comm. 7th December 2004). From the CSIM user manual (<http://www.lsm.tugraz.at/csim/reference/html/classLifNeuron.html>) LIF neurons make use of the following equation and parameters:

$$\tau_m \frac{dV_m}{dt} = -(V_m - V_{resting}) + R_m \cdot (I_{syn}(t) + I_{inject} + I_{noise})$$

Equation 1. The algorithm for the calculation inside LIF Neurons.

where $\tau_m = C_m \cdot R_m$ is the membrane time constant.

- float C_m . The membrane capacity C_m [range=(0,1); units=F].
- float R_m . The membrane resistance R_m [units=Ohm; range=(0,1e30)].
- float V_{thresh} . If V_m exceeds V_{thresh} a spike is emitted. [units=V; range=(-10,100)].
- float V_{resting} . The resting membrane voltage. [units=V; range=(-1,1)].
- float V_{reset} . The voltage to reset V_m to after a spike. [units=V; range=(-1,1)].
- float V_{init} . The initial condition for V_m at time $t=0$. [units=V; range=(-1,1)].
- float T_{refract} . The length of the absolute refractory period. [units=sec; range=(0,1)].
- float I_{noise} . The standard deviation of the noise to be added each integration time constant. [range=(0,1); units=A].
- float I_{inject} . A constant current to be injected into the LIF neuron. [units=A; range=(-1,1)].
- double V_m . The membrane voltage V_m [read only; units=V].
- float I_{syn} . synaptic input current.

From experiments Spencer (1980) calculated values for some of these variables in the giant network of neurons in a hydrozoan jellyfish (e.g. *Polyorchis penicillatus*):

$$R_m = 98 \text{ k}\Omega \text{ cm}^2$$

$$C_m = 1.52 \text{ }\mu\text{F/cm}^2$$

$$\tau_m = 149\text{ms}$$

$$V_{\text{resting}} = -55\text{mv} \quad (\text{Spencer } 1977).$$

2.11 Discussion

The biology behind nervous systems is very complex and could probably go into a lot more detail than I have in this review. Jellyfish possess a relatively simple nervous system compared to many other creatures but it shows incredible functionality, especially when considering it may have evolved over 600 million years ago.

Neural networks are being increasingly used in computing for the implementation of complex algorithms and the future of this must surely be models that more closely imitate the reality of nervous systems, in particular the brain.

The next stage in this project is to produce a model of the Swimming Motor Neurons located within the Inner Nerve Ring of *Polyorchis penicillatus*.

3 The Model

3.1 The Concept

The idea was to build a computer based model of a spiking neural network to simulate the behaviour of the Swimming Motor Neurone's (SMN's) inside the jellyfish *Polyorchis penicillatus*. From undertaking the Literature Survey I had become aware that the most important feature of the SMN's was the synchronicity with which these neurones fired. As was discussed in the Literature Survey the SMN's are much more complex than all simply firing at the same time, they have a much more sophisticated mechanism built in to achieve this synchronicity producing a consistently changing wave pattern that travels around the nerve ring. Modelling this complex wave pattern, if possible with a spiking neural net, would be exceptionally difficult and not necessary to achieve the goals of the project. The important concept was to have a ring of neurones all firing at the same time to give a consistent spiking pattern around the entire ring. This would still emulate the behaviour of the SMN's in the jellyfish but in a slightly different way that was appropriate for a computer based neural network. So it was not going to be behaving in exactly the way a jellyfish would be expected to but the output would achieve the same results.

3.2 The Goal

As CSIM does not provide any form of electrical coupling the goal was to use the available components of CSIM to model the electrical coupling in the SMN's. The final model was intended to be a ring of neurons that displayed the synchronous spiking behaviour associated with the SMN's of the jellyfish. The other side to the project was to investigate how suitable CSIM was for this type of modelling and for future work on modelling the nervous system in the jellyfish.

3.3 Model Construction

The code for the model is included in the appendix at the back of the report. A diagram and description of the model follows:

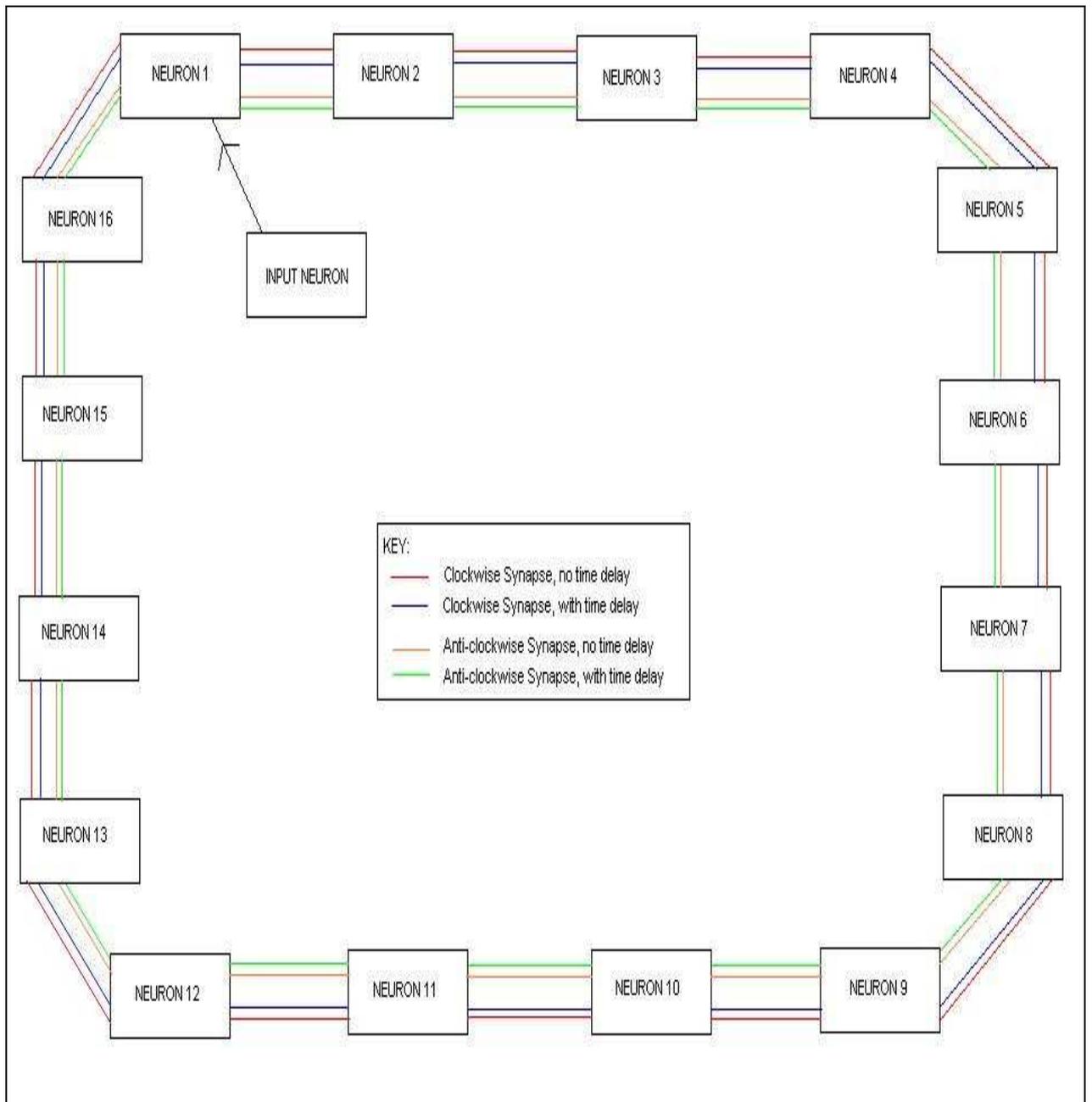


Fig 11. Diagram to show the structure of the final model.

As can be seen the model is a ring of 16 neurons with a single input acting as the stimulus. Each neuron has 2 synaptic connections towards each neighbour and also receives 2 synaptic connections from each neighbour totalling 4 synapses feeding into the neuron and 4 synapses leaving the neuron. The refractory period on each neuron was set to 0.04 seconds so that each round of spiking could easily be identified from the rest. Originally there were 2 rings of synapses, one in each direction, with zero time delays, this provided the starting point for the model. An impulse from the input Neuron causes a spike in Neuron 1 which results in spikes travelling around the ring in both the clockwise and anti-clockwise direction. It travels through the synapses with zero time delay causing each neuron to spike 0.0002 seconds after the previous neuron in the ring (presumably due to time taken for propagation and calculation in the neuron). Although a delay of 0.0002 seconds may seem negligible we must

consider the complexity and accuracy of the mechanism within the jellyfish that effectively causes synchronous spiking where such a delay would be unacceptable and indeed does not occur. By the time the spike had propagated around to the other side of the ring, i.e. to Neuron 9 where the refractory period would effectively cause the impulse to die and no longer be propagated the delay between that neuron spiking and the first neuron spiking would be $(0.0002 * 8 \text{ (synapses)}) 0.0016$ seconds, a delay that is even more unacceptable. Without providing input to all of the neurons in the circuit this time delay will be inevitable thus not accurately modelling the neural behaviour associated with the SMN's. Obviously inputting to all of the neurons would destroy the concept of having a network behaving synchronously. A mechanism was needed that would cause the network to synchronise over a period of time (preferably as short a time as possible) from an original input. It was necessary to create extra synapses linking the network with parameters that would cause this behaviour. Originally a set of extra synapses were added in a clockwise direction around the ring with a time delay. The time delay decreased by 0.0002 seconds with each consecutive synapse so that they would all eventually fire at the same time. The anticlockwise ring of synapses with no time delay were removed for simplicity. This worked to a certain extent with the result being a single synchronous spike throughout the ring. However the network was then out of synch due to the different time delays on all the synapses and so the model was ineffective. It was questioned as to whether or not a time delay that was the same on each synapse would cause the network to synchronise over time, a time delay long enough to cause spiking well after the original wave had travelled around the ring. With the anticlockwise ring of synapses with zero time delay re-added the time delay on the delay variable synapses was changed to 0.1 seconds and straight away a pattern emerged that showed the spiking pattern moving towards a state of synchrony. However as this appeared to be taking a significant length of time, in fact the network had not even synchronised after 3 seconds by which time 26 rounds of spiking had taken place, so another set of time-delayed synapses was constructed in the anti-clockwise direction of the ring, this also made the ring more symmetrical, a feature certainly important to the nerve ring in the jellyfish and so consequently to this model. The model now looked like the model in the above diagram with all 4 rings present. Now the network firing pattern synchronised much more quickly than it had done before, from an original spike the network began firing synchronously by the 13th round of spiking, 1.2027 seconds after the first spike had been received. The data from these simulations can be found in the appendix at the end of the report.

3.3.1 Output

The output from a run of the simulation was a series of graphs, one for each neuron to show the time of spiking. The graphs at first were misleading, neurons that appeared to be spiking at the same time in fact were not so a printout of the spiking time matrix for each neuron was obtained. These were a lot easier to compare than the graphs as time differences of 0.0002 seconds were hard to see on a graph but could be physically seen in a list of data. The graphs produced were as follows:

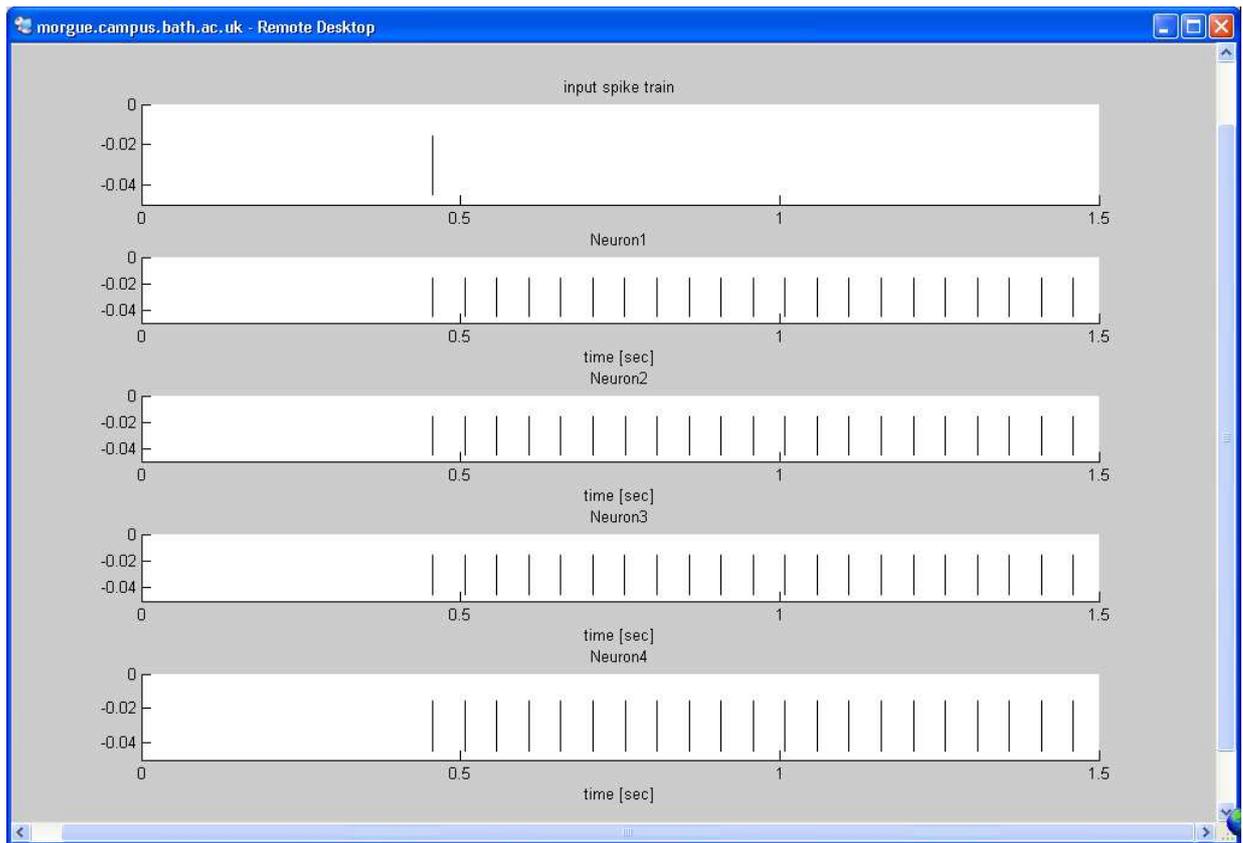


Fig 12. The output for Neurons 1-4 in the ring. The input spike is also shown.

4 frames would be produced for each run of the simulation, each one containing the spiking output from 4 neurons. It is easy to see that exact spiking times cannot be obtained from looking at the graphs. The printout of the spiking time matrices was simple and fairly easy to compare and had the following structure (the bold type set was added later to highlight the first synchronous spike):

```
st =
Columns 1 through 5
0.4567  0.5067  0.5567  0.6071  0.6571

Columns 6 through 10
0.7075  0.7575  0.8079  0.8579  0.9081

Columns 11 through 15
0.9583  1.0085  1.0587  1.1089  1.1591

Columns 16 through 20
1.2093  1.2595  1.3097  1.3599  1.4101
```

Column 21

1.4603

st =

Columns 1 through 5

0.4565 0.5065 0.5569 0.6069 0.6573

Columns 6 through 10

0.7073 0.7577 0.8077 0.8581 0.9081

Columns 11 through 15

0.9583 1.0085 1.0587 1.1089 1.1591

Columns 16 through 20

1.2093 1.2595 1.3097 1.3599 1.4101

Column 21

1.4603

The exact time of synchronisation is easy to see, in this case it was after 1.0587 seconds of the simulation, however this is not the time to synchronisation it was important to bear in mind that the input spike was not applied until 0.4560 seconds in so the exact time to synchronisation was $(1.0587 - 0.4560)$ 0.6027 seconds. These 2 neurons also appear to be synchronised with each other much earlier as well, they were, the reason for this being they are next to each other, in fact they are neurons no.15 and no. 16, however it was not until spike 13 where the entire ring was synchronised.

3.4 Experimentation

3.4.1 Varying Synaptic Time Delay

The idea of decreasing the synaptic time delay so that synchronisation would be achieved faster was investigated. At this point the refractory period was still set at 0.04 seconds. The time delay was halved from 0.1 seconds to 0.05 seconds with the assumption that this would make synchronisation significantly faster. It did. Now the time to synchrony was 0.6027 seconds, $(1.2027 - 0.6027)$ 0.6 seconds faster than a synaptic time delay 0.05 seconds longer. It was also interesting that again 13 rounds of spiking were required before synchronisation was achieved. If the first round of spiking is ignored as that is going to happen anyway regardless of the time delays on the synapses then there are another 12 rounds of spiking before synchronisation will occur, i.e. 12 rounds of spiking where the time delays play a part. If each round of

spiking is now 0.05 seconds faster (because the time delay is that much shorter) then it makes sense that the time to synchronisation will be $(12 * 0.05)$ 0.6 seconds faster. From this a hypothesis that “if the synaptic time delay was reduced from 0.1 seconds to 0.075 seconds then there would be 13 spikes produced in each neuron before synchronisation was achieved and that the reduction in time to synchronisation would be $(0.025*12)$ 0.3 seconds” was tested. It was expected that the time to synchronisation would be $(1.2027 - 0.3)$ 0.9027 seconds. It was exactly 0.9027 seconds. The data for all of the simulations run can be found in the appendix at the end of the report. So theoretically by altering the synaptic time delay appropriately the time to synchronisation could be set as required. A formula for achieving the required time to synchronisation is given as follows:

$$\text{SynapticDelay} = 0.1 - ((1.2027 - \text{TTS}) / 12)$$

TTS = Time to Synchronisation

Equation 2. Formula for calculating the Synaptic Delay when a specific Time to Synchronisation is required.

The formula appears to imply that 1.2027 is the longest time to synchronisation possible, this seems strange but having a synaptic delay greater than 0.1 seconds does not produce any resultant spikes in the neurons. I suspect that after such a duration the impulse has decayed to such an extent that the signal passed on by the synapse is not large enough for the neuron to fire.

It also has to be remembered that the time delay must be at least as large as the refractory period as otherwise the spikes produced by the time delayed synapses will merely be inhibited by a neuron in the refractory state.

3.4.2 Varying Refractory period

The refractory period is the length of time in which a neuron will resist any spiking input which it receives from incoming synapses after it has spiked itself. The idea was that by varying this period the time taken to get through the necessary number of spiking rounds before synchronisation was achieved would also be varied, i.e. by reducing the refractory period synchronisation would be achieved faster. The time delay was set to its default value of 0.1 seconds. At first the refractory period was reduced from 0.04 to 0.02 seconds. No difference was seen in the spiking behaviour of the neurons in any way. See the appendix for the data obtained from the simulation. This further validates Equation 1 as being a general formula for calculating the desired synaptic delay when the TTS is required to be set as it does not have to take the refractory period into account. By altering the refractory period to 0.01 seconds a dramatic change was recorded, however it was not as intended. The length of the time to synchronisation was dramatically increased and in fact the network had still not synchronised after almost 3 seconds, see results in appendix. The idea that this inefficiency to synchronise was thought to be due to the length of the time delay on the synapses being set at a relatively high value of 0.1 seconds, this meant that as spikes were allowed to propagate more freely around the ring they would probably “get in the way” of the spikes then produced by the time delayed synapses, i.e. by the time a time delayed synapses fired, the neuron it was spiking into would be in the refractory period having been spiked a fraction of a second earlier so the spike would have no effect. The overall result of this would be that the time

delayed synapses were not allowed to have an effect on the spiking pattern of the ring and the hypothesis that “under these conditions the ring would never actually synchronise” was proposed. An extended simulation was run over a 6 second period and indeed the network had not synchronised after this time. To see if the network could be made to synchronise earlier the time delay on the synapses was reduced from 0.1 seconds to 0.05 seconds, it was hoped that this would allow the spikes from the time delayed synapses to have an impact on the network, causing it to synchronise. This had no effect with the network still not being synchronised after 3 or even 6 seconds. The time delay was further reduced to 0.01 seconds in line with the refractory period to make sure the spike from the time delayed synapse would be the first one to hit the neuron after it had previously been spiked, this made sense to me because if a neuron is spiked then the time delayed synapse feeding into that neuron will be “loaded” with a spike which, after the time delay, should be the next spike to affect the neuron. This however had no effect on the pattern produced and the network did not synchronise after the 6 seconds.

It was concluded that varying the refractory period, even with an optimised time delay on the synapses could not be made to influence the network in a positive way, i.e. make it synchronise faster, in fact if any change was seen it was only that of the network being unable to synchronise. These results I believe can be considered accurate as the time delays used were tested and proved to improve the synchronisation speed, with the only change now being to the refractory period, therefore it must be the refractory period contributing to the inability of the network to synchronise.

3.4.3 Varying Synaptic Time Delay Once Again

It was noticed that although the change in the refractory period from 0.04 seconds to 0.02 seconds had no change in the spiking pattern when the time delay was 0.1 seconds, the opportunity for a shorter time delay now existed, as the time delay has to be at least as large as the refractory period. To prove that there was no change in the spiking patterns of the simulations with these 2 refractory periods a simulation was run with a time delay of 0.05 seconds and the refractory period set at 0.02 seconds and compared to the simulation run earlier when the time delay was the same and the refractory period 0.04 seconds. No differences were observed, so it can be concluded that these varying refractory periods do not alter the simulation. The hypothesis that “by reducing the synaptic time delay to 0.03 seconds when the refractory period was set to 0.02 seconds would produce the fastest synchronisation yet” was tested. This was an obvious statement and doesn’t prove anything new about the network but if proved correct would show that a shorter refractory period could indirectly decrease the time to synchronisation because it allowed a shorter time delay to be used, a feature that has been proved to reduce the time to synchronisation. It worked and the network synchronised in 0.3627 seconds, by far the shortest time taken so far. See the appendix for the actual data obtained. To validate equation 1.1 further, the data from this simulation can be put in and it should prove to be correct:

$$0.1 - ((1.2027 - 0.3627) / 12) = 0.03$$

0.03 was the synaptic time delay that was used so this proves the equation.

It must be realised that this equation is only valid when the refractory period is set appropriately, i.e. in the range where synchronisation is achieved. It cannot, for example, be used when the refractory period is set to 0.01 seconds as the network does not achieve synchronisation when this is the case.

4 Conclusions

4.1 Analysis of Model

Overall I feel that the model produced shows some of the fundamental characteristics of the SMN's in the jellyfish, certainly the ability to synchronise and to continue to fire synchronously. The model is dynamic in the sense that by varying some of the parameters of the network the behaviour can be influenced, more specifically the spiking behaviour and the time taken for the network to synchronise.

4.1.1 Accuracies

The important concept to integrate into the model was that of synchronous behaviour between the neurons, more specifically synchronous spiking behaviour. This has been achieved and the network reaches a state where every neuron in the ring is in a synchronous state spiking indefinitely, or for as long as the simulation is run. It was not possible with the neural modelling software used to produce a ring that would display synchronous behaviour instantaneously and the best result achieved was for the network to synchronise in 0.3627 seconds, a figure which could almost certainly be improved with some modifications of the parameters in the neurons and synapses. It is important to remember the aim was not to imitate exactly the behaviour of the SMN's in the jellyfish such as the complex wave pattern that emerges as an action potential travels around the ring, indeed spikes produced by neurons in computer based neural nets have a specific value they do not differ in amplitude or duration, a spike is a spike with no variation. The model produced should be thought of as the computer based version of the jellyfish displaying the important characteristics without having to emulate all of the complex functionality.

4.1.2 Discrepancies

As I have already stated the network of neurons takes time to synchronise a characteristic not seen with in the SMN's of the jellyfish. Time taken to synchronisation is inevitable and has been reduced to a very small value but never the less is still an unwanted feature. The SMN's in the Inner Nerve Ring of the jellyfish are not as simple as a simple ring with each neuron connected to the next in sequence, Spencer (1979) describes the SMN's as "an anastomosing network of giant neurons," i.e. there are many connections throughout the ring and they are not arranged just linearly. However although the structure of the ring produced is not like this it does not affect the output that it produces. A model where there were many interconnections between the neurons would certainly result in the same synchronous spiking behaviour of the neurons, there is although a possibility that extra connections

might facilitate the faster synchronisation of the network, an idea that is certainly worth investigating.

4.2 Possible Further Development

4.2.1 Two Directions for Further Development

There are 2 directions in which further development of the model could proceed.

Firstly in the development of the model itself in which the behaviour of the nerve ring would be improved. The time to synchronisation could undoubtedly be reduced further by more careful setting of the neuron and synapse parameters. Also the response of the network to different patterns of spike inputs as opposed to the one spike that it has been tested with so far, a real jellyfish is certain to receive many inputs that stimulate its SMN's not just one at a time. Spencer (1980) states that "a constant depolarising current injected into 1 SMN will cause the whole network to fire repeatedly at very high frequencies," a characteristic that has not yet been tested on this network. Another obvious modification to the ring would be the addition of further synaptic connections between some of the neurons in the ring, this could be done in a structured pattern throughout the ring or in a more random manner that would probably more accurately represent the SMN's in the jellyfish.

The second direction for development to proceed would be to add further structures onto and around the nerve ring such as some of the muscle tissue or epithelial cells that overly the SMN's and are directly stimulated by the spiking of the ring. The goal of this would be to see the different regions of the muscle sheet spiking at the same time imitating the synchronous contractions seen in the sub-umbrella muscle sheet inside the bell of the jellyfish. This concept would be very similar to that seen in the jellyfish because the system controlling the contraction of the muscle sheet is designed to cause the regions of the muscle sheet to contract at the same time, however this differs from the spiking pattern in the SMN's where in actuality all of the neurons do not fire at the same time instead they have the complex time delay mechanism built in that makes them behave as if they were all spiking at the same time. Structures resembling the ocelli could be attached around the ring as currently the Input Neuron is playing this role, however an input neuron would still be required to input to these. I am unsure as to the ability of CSIM to model structures such as muscle and epithelial tissue, this would have to be investigated and there is the possibility different software would have to be used however this is not a conclusion.

4.3 Concluding Remarks

The original aim of the project was to investigate the neural structure of the jellyfish and attempt to build a model of at least part of the system. The literature review was instrumental in providing a much more focused aim in that after it had been undertaken it was decided that the Swimming Motor Neurons would be the area that would be concentrated on with the aim to be to build a model of these. The reason for concentrating on this group of neurons in particular was because they appeared to be the most significant and important neurons the jellyfish possess. Their structure and functionality are very complex and are closely related to one another and I see them as being the 'driving force' behind swimming control. It was felt that if this project was to be the basis for future work then this was the best place to begin as much of the structure and functionality of the jellyfish appears to be built around these neurons. The software, CSIM, that I was going to use was also to be evaluated in its appropriateness for such a project and for possible future work.

There were no real expectations as to what would be produced at the start of the project as the outline and aim was vague. After the literature survey the goal became to produce a model that would show the behaviour of the SMN's. This has been achieved to a certain extent and although the model is not a strictly accurate representation of what goes on in the SMN's it displays some of the properties.

If this model is going to be built upon then the nerve ring will surely have to be improved to more accurately model the SMN's probably by obtaining more realistic values for the variables in the Neurons but it should be a good starting point.

The project has left some questions that have not been answered. Firstly it is unclear, from the literature reviewed, as to whether or not the action potential in the SMN's actually synchronises over time like the model does. I have talked about the complex mechanism by which swimming muscle innervation is achieved but I am unsure as to whether or not the neurons will continue to innervate the muscle sheet in this complex manner or if the neurons will eventually become synchronised in their firing pattern. I have not found any references to the neurons actually becoming synchronised however that is not to say it does not happen after a period of time. It is possible that the constant decreasing of the action potential is only there to mask the fact that the neurons do not initially fire simultaneously and eventually they will reach a point where they do and the varying shape of the action potential will not be seen. This is a hypothesis that may be worth investigating.

As for the appropriateness of CSIM, it was certainly able to show some results however I am unsure just how much further it will be able to take this project as the lack of electrical coupling available might be an issue. It might be good enough as synchronisation was achieved however a conclusion cannot be drawn and the possibility of further investigation would be recommended.

5 Bibliography

5.1 Papers

- 1) Spencer A.N. (1977). Neurobiology of Polyorchis 1. Function of Effector Systems. *Journal of Biology*, Vol 9, No.2, pp. 143-157 (1978).
- 2) Spencer A.N. (1978). Neurobiology of Polyorchis 2. Structure of Effector Systems.
- 3) Spencer A.N, Satterlie R.A (1979). Electrical and Dye Coupling in an Identified Group of Neurons in a Coelenterate.
- 4) Spencer A.N. (1980). The Parameters and Properties of a Group of Electrically Coupled Neurons in the Central Nervous System of a Jellyfish. *J.exp.Biol* (1981), 93, 33-50.
- 5) Spencer A.N, Satterlie R.A. (1981). Action Potential and Contraction in Subumbrellar Swimming Muscle in *Polyorchis penicillatus* (Hydromedusae). *J.Biol* (1981) 144:401-407.
- 6) Spencer A.N. (1982). The Physiology of a Coelenterate Neuromuscular Synapse. *J Comp Physiol* (1982) 148:353-363.
- 7) Satterlie R.A, Spencer A.N. (1982). Neuronal Control of Locomotion in Hydrozoan Medusae. *J Physiol* (1983) 150:195-206.
- 8) Spencer A.N, Arkett S.A (1983). Radial Symmetry and the Organisation of Central Neurons in a Hydrozoan Jellyfish. *J.exp.Biol* 110, 69-90 (1984).
- 9) Spencer A.N (1993). Modulatory Mechanisms at a Primitive Neuromuscular Synapse: Membrane Currents, Transmitter Release and Modulation by Transmitters in a Cnidarian Motor Neuron. *Amer. Zool*, 35:520-528 (1995).
- 10) Spencer A.N, Przysieznik J, Acosta-Urquidi J, Basarsky T.A. Presynaptic spike broadening reduces junctional potential amplitude. *Nature*. 1989 Aug 24;340(6235):636-8.
- 11) Megill W.M (2002). The Biomechanics of Jellyfish Swimming. Thesis (P.H.D). University of British Columbia, Canada.
- 12) Bohte S. (2003). Spiking Neural Networks. Thesis. Universiteit Leiden.

5.2 Books

- 13) Curtis, Helena (1975). *Biology* 2nd Edition. Worth.
- 14) Villee, Claude Alvin (1977). *Biology* 7th Edition. Saunders.
- 15) Raven and Johnson (1992). *Biology* 3rd Edition. Mosby-Year-Book.
- 16) Kandel E.R, Schwartz J.H, Jessell T.M (1998). *Essentials of Neuroscience and Behaviour*.
- 17) Bayer F.M, Owre H.B (1968). *The Free Living Lower Invertebrates*.
- 18) Spencer A.N (aspencer@bms.bc.ca), 12th December 2004. Neural Modelling of a Jellyfish. Email to J.A. Yeomans (cs1jay@bath.ac.uk).
- 19) Richardson D (D.Richardson@bath.ac.uk), 12th December 2004. Jellyfish. Email to J.A. Yeomans (cs1jay@bath.ac.uk).

5.3 Web Sites

- 20) <http://amygdala.sourceforge.net>
- 21) <http://www.wikipedia.org/>
- 22) <http://www.promland.com/>
- 23) <http://www.ivorix.com>
- 24) <http://www.esu.edu/>
- 25) <http://www.ship.edu/>
- 26) <http://www.emc.maricopa.edu/>
- 27) <http://www.faculty.clintoncc.suny.edu/>
- 28) <http://www.lsm.tugraz.at/>

5.4 Further Reading

- 29) Ruppert, Barnes. Invertebrate Zoology 6th Edition.
- 30) Guttman, Hopkins (). Understanding Biology.
- 31) Wilmoth J.H (). Biology of Invertebrates.
- 32) Pechenik J.A (). Biology of the Invertebrates 4th Edition.
- 33) Y.-C. James Lin, Spencer A.N (2001). Calcium currents from jellyfish striated muscle cells: preservation of phenotype, characterisation of currents and channel localisation. The Journal of Experimental Biology 204, 3717-3726 (2001).
- 34) Doron T, Schwartz E.L (1996). Computing with the Leaky Integrate and Fire Neuron: Logarithmic Computation and Multiplication. Department of Cognitive and Neural Studies, Boston University.

- 35) <http://www.tvdsb.on.ca/westmin/science/sbioac/homeo/action.htm>
- 36) <http://jellieszone.com/hydromedusae.htm>
- 37) <http://www.cityhillmiddleschool.com/jellyfish.htm>
- 38) <http://faculty.washington.edu/cemills/Hydromedusae.html>
- 39) <http://www.uwm.edu/~fjh/neuron.pdf>
- 40) <http://normandy.sandhills.cc.nc.us/psy150/neuron.html>
- 41) <http://www.ucihs.uci.edu/biochem/steele/Satterlie.pdf>
- 42) <http://www.fortunecity.com/greenfield/buzzard/387/neuronsascircuit.htm>
- 43) <http://www.rwc.uc.edu/koehler/biophys/4d.html>

6 Appendix A

A.1 Test Simulations

The model that was the starting point for each simulation was one that had a set of synapses going in each direction around the ring with no time delays.

Unless stated otherwise the refractory period is 0.04 seconds and the synaptic time delay on the time delayed synapses is 0.1 seconds.

In each simulation the input Neuron was made to spike at 0.4560 seconds into the simulation.

The point at which synapses start spiking synchronously is highlighted in bold text.

A.1.1 Model Construction

1) The anticlockwise ring of synapses with zero time delay was removed and 1 clockwise set of time delayed synapses was added with time delays that decreased from 0.1 seconds by 0.0002 seconds with each consecutive synapse. The 1 synchronous spike that appeared during the simulation was the second spike emitted by each neuron. It is not highlighted as the network was permanently out of synch after this spike.

St =

Columns 1 through 6

0.4563 0.5565 0.6537 0.7511 0.8487 0.9465

Columns 7 through 12

1.0445 1.1427 1.2411 1.3389 1.4367 1.5347

Columns 13 through 18

1.6327 1.7309 1.8291 1.9269 2.0249 2.1229

Columns 19 through 24

2.2209 2.3191 2.4171 2.5151 2.6131 2.7111

Columns 25 through 26

2.8091 2.9073

St =

Columns 1 through 6

0.4565 0.5565 0.6539 0.7513 0.8489 0.9467

Columns 7 through 12

1.0447 1.1429 1.2413 1.3391 1.4369 1.5349

Columns 13 through 18

1.6329 1.7311 1.8293 1.9271 2.0251 2.1231

Columns 19 through 24

2.2211 2.3193 2.4173 2.5153 2.6133 2.7113

Columns 25 through 26

2.8093 2.9075

St =

Columns 1 through 5

0.4567 0.5565 0.6541 0.7515 0.8491

Columns 6 through 10

0.9469 1.0449 1.1431 1.2415 1.3393

Columns 11 through 15

1.4371 1.5351 1.6331 1.7313 1.8295

Columns 16 through 20

1.9273 2.0253 2.1233 2.2213 2.3195

Columns 21 through 25

2.4175 2.5155 2.6135 2.7115 2.8095

Column 26

2.9077

St =

Columns 1 through 5

0.4569 0.5565 0.6543 0.7517 0.8493

Columns 6 through 10

0.9471 1.0451 1.1433 1.2417 1.3395

Columns 11 through 15

1.4373 1.5353 1.6333 1.7315 1.8297

Columns 16 through 20

1.9275 2.0255 2.1235 2.2215 2.3197

Columns 21 through 25

2.4177 2.5157 2.6137 2.7117 2.8097

Column 26

2.9079

St =

Columns 1 through 5

0.4571 0.5565 0.6545 0.7519 0.8495

Columns 6 through 10

0.9473 1.0453 1.1435 1.2419 1.3397

Columns 11 through 15

1.4375 1.5355 1.6335 1.7317 1.8299

Columns 16 through 20

1.9277 2.0257 2.1237 2.2217 2.3199

Columns 21 through 25

2.4179 2.5159 2.6139 2.7119 2.8099

Column 26

2.9081

St =

Columns 1 through 5

0.4573 0.5565 0.6547 0.7521 0.8497

Columns 6 through 10

0.9475 1.0455 1.1437 1.2421 1.3399

Columns 11 through 15

1.4377 1.5357 1.6337 1.7319 1.8301

Columns 16 through 20

1.9279 2.0259 2.1239 2.2219 2.3201

Columns 21 through 25

2.4181 2.5161 2.6141 2.7121 2.8101

Column 26

2.9083

St =

Columns 1 through 5

0.4575 0.5565 0.6549 0.7523 0.8499

Columns 6 through 10

0.9477 1.0457 1.1439 1.2423 1.3401

Columns 11 through 15

1.4379 1.5359 1.6339 1.7321 1.8303

Columns 16 through 20

1.9281 2.0261 2.1241 2.2221 2.3203

Columns 21 through 25

2.4183 2.5163 2.6143 2.7123 2.8103

Column 26

2.9085

St =

Columns 1 through 5

0.4577 0.5565 0.6551 0.7525 0.8501

Columns 6 through 10

0.9479 1.0459 1.1441 1.2425 1.3403

Columns 11 through 15

1.4381 1.5361 1.6341 1.7323 1.8305

Columns 16 through 20

1.9283 2.0263 2.1243 2.2223 2.3205

Columns 21 through 25

2.4185 2.5165 2.6145 2.7125 2.8105

Column 26

2.9087

St =

Columns 1 through 5

0.4579 0.5565 0.6553 0.7527 0.8503

Columns 6 through 10

0.9481 1.0461 1.1443 1.2427 1.3405

Columns 11 through 15

1.4383 1.5363 1.6343 1.7325 1.8307

Columns 16 through 20

1.9285 2.0265 2.1245 2.2225 2.3207

Columns 21 through 25

2.4187 2.5167 2.6147 2.7127 2.8107

Column 26

2.9089

St =

Columns 1 through 5

0.4581 0.5565 0.6551 0.7529 0.8505

Columns 6 through 10

0.9483 1.0463 1.1445 1.2429 1.3407

Columns 11 through 15

1.4385 1.5365 1.6345 1.7327 1.8309

Columns 16 through 20

1.9287 2.0267 2.1247 2.2227 2.3209

Columns 21 through 25

2.4189 2.5169 2.6149 2.7129 2.8109

Column 26

2.9091

St =

Columns 1 through 5

0.4583 0.5565 0.6549 0.7531 0.8507

Columns 6 through 10

0.9485 1.0465 1.1447 1.2429 1.3409

Columns 11 through 15

1.4387 1.5367 1.6347 1.7329 1.8311

Columns 16 through 20

1.9289 2.0269 2.1249 2.2229 2.3211

Columns 21 through 25

2.4191 2.5171 2.6151 2.7131 2.8111

Column 26

2.9093

St =

Columns 1 through 5

0.4585 0.5565 0.6547 0.7531 0.8509

Columns 6 through 10

0.9487 1.0467 1.1447 1.2429 1.3411

Columns 11 through 15

1.4389 1.5369 1.6349 1.7329 1.8311

Columns 16 through 20

1.9291 2.0271 2.1251 2.2231 2.3211

Columns 21 through 25

2.4193 2.5173 2.6153 2.7133 2.8113

Column 26

2.9093

St =

Columns 1 through 5

0.4587 0.5565 0.6545 0.7527 0.8511

Columns 6 through 10

0.9489 1.0467 1.1447 1.2427 1.3409

Columns 11 through 15

1.4391 1.5369 1.6349 1.7329 1.8309

Columns 16 through 20

1.9291 2.0271 2.1251 2.2231 2.3211

Columns 21 through 25

2.4191 2.5173 2.6153 2.7133 2.8113

Column 26

2.9093

St =

Columns 1 through 5

0.4589 0.5565 0.6543 0.7523 0.8505

Columns 6 through 10

0.9489 1.0467 1.1445 1.2425 1.3405

Columns 11 through 15

1.4387 1.5369 1.6347 1.7327 1.8307

Columns 16 through 20

1.9287 2.0269 2.1249 2.2229 2.3209

Columns 21 through 25

2.4189 2.5169 2.6151 2.7131 2.8111

Column 26

2.9091

St =

Columns 1 through 5

0.4591 0.5565 0.6541 0.7519 0.8499

Columns 6 through 10

0.9481 1.0465 1.1443 1.2421 1.3401

Columns 11 through 15

1.4381 1.5363 1.6345 1.7323 1.8303

Columns 16 through 20

1.9283 2.0263 2.1245 2.2225 2.3205

Columns 21 through 25

2.4185 2.5165 2.6145 2.7127 2.8107

Column 26

2.9087

St =

Columns 1 through 5

0.4593 0.5565 0.6539 0.7515 0.8493

Columns 6 through 10

0.9473 1.0455 1.1439 1.2417 1.3395

Columns 11 through 15

1.4375 1.5355 1.6337 1.7319 1.8297

Columns 16 through 20

1.9277 2.0257 2.1237 2.2219 2.3199

Columns 21 through 25

2.4179 2.5159 2.6139 2.7119 2.8101

Column 26

2.9081

2) 1 ring of time delayed synapses added to the original model. Every time delay set to 0.1 seconds. It shows a slow move towards synchronisation although it doesn't actually get there.

St =

Columns 1 through 5

0.4563 0.5567 0.6571 0.7575 0.8579

Columns 6 through 10

0.9583 1.0585 1.1589 1.2591 1.3595

Columns 11 through 15

1.4595 1.5595 1.6595 1.7595 1.8595

Columns 16 through 20

1.9595 2.0595 2.1599 2.2601 2.3605

Columns 21 through 25

2.4607 2.5611 2.6613 2.7617 2.8619

Column 26

2.9623

St =

Columns 1 through 5

0.4565 0.5565 0.6569 0.7573 0.8577

Columns 6 through 10

0.9581 1.0585 1.1587 1.2591 1.3593

Columns 11 through 15

1.4597 1.5597 1.6597 1.7597 1.8597

Columns 16 through 20

1.9597 2.0597 2.1597 2.2601 2.3603

Columns 21 through 25

2.4607 2.5609 2.6613 2.7615 2.8619

Column 26

2.9621

St =

Columns 1 through 5

0.4567 0.5567 0.6567 0.7571 0.8575

Columns 6 through 10

0.9579 1.0583 1.1587 1.2589 1.3593

Columns 11 through 15

1.4595 1.5599 1.6599 1.7599 1.8599

Columns 16 through 20

1.9599 2.0599 2.1599 2.2599 2.3603

Columns 21 through 25

2.4605 2.5609 2.6611 2.7615 2.8617

Column 26

2.9621

St =

Columns 1 through 5

0.4569 0.5569 0.6569 0.7569 0.8573

Columns 6 through 10

0.9577 1.0581 1.1585 1.2589 1.3591

Columns 11 through 15

1.4595 1.5597 1.6601 1.7601 1.8601

Columns 16 through 20

1.9601 2.0601 2.1601 2.2601 2.3601

Columns 21 through 25

2.4605 2.5607 2.6611 2.7613 2.8617

Column 26

2.9619

St =

Columns 1 through 5

0.4571 0.5571 0.6571 0.7571 0.8571

Columns 6 through 10

0.9575 1.0579 1.1583 1.2587 1.3591

Columns 11 through 15

1.4593 1.5597 1.6599 1.7603 1.8603

Columns 16 through 20

1.9603 2.0603 2.1603 2.2603 2.3603

Columns 21 through 25

2.4603 2.5607 2.6609 2.7613 2.8615

Column 26

2.9619

St =

Columns 1 through 5

0.4573 0.5573 0.6573 0.7573 0.8573

Columns 6 through 10

0.9573 1.0577 1.1581 1.2585 1.3589

Columns 11 through 15

1.4593 1.5595 1.6599 1.7601 1.8605

Columns 16 through 20

1.9605 2.0605 2.1605 2.2605 2.3605

Columns 21 through 25

2.4605 2.5605 2.6609 2.7611 2.8615

Column 26

2.9617

St =

Columns 1 through 5

0.4575 0.5575 0.6575 0.7575 0.8575

Columns 6 through 10

0.9575 1.0575 1.1579 1.2583 1.3587

Columns 11 through 15

1.4591 1.5595 1.6597 1.7601 1.8603

Columns 16 through 20

1.9607 2.0607 2.1607 2.2607 2.3607

Columns 21 through 25

2.4607 2.5607 2.6607 2.7611 2.8613

Column 26

2.9617

St =

Columns 1 through 5

0.4577 0.5577 0.6577 0.7577 0.8577

Columns 6 through 10

0.9577 1.0577 1.1577 1.2581 1.3585

Columns 11 through 15

1.4589 1.5593 1.6595 1.7599 1.8601

Columns 16 through 20

1.9605 2.0607 2.1609 2.2609 2.3609

Columns 21 through 25

2.4609 2.5609 2.6609 2.7609 2.8613
Column 26

2.9615

St =

Columns 1 through 5

0.4577 0.5579 0.6579 0.7579 0.8579

Columns 6 through 10

0.9579 1.0579 1.1579 1.2579 1.3583

Columns 11 through 15

1.4587 1.5591 1.6593 1.7597 1.8599

Columns 16 through 20

1.9603 2.0605 2.1609 2.2611 2.3611

Columns 21 through 25

2.4611 2.5611 2.6611 2.7611 2.8611

Column 26

2.9615

St =

Columns 1 through 5

0.4575 0.5579 0.6581 0.7581 0.8581

Columns 6 through 10

0.9581 1.0581 1.1581 1.2581 1.3581

Columns 11 through 15

1.4585 1.5589 1.6591 1.7595 1.8597

Columns 16 through 20

1.9601 2.0603 2.1607 2.2609 2.3613

Columns 21 through 25

2.4613 2.5613 2.6613 2.7613 2.8613

Column 26

2.9613

St =

Columns 1 through 5

0.4573 0.5577 0.6579 0.7583 0.8583

Columns 6 through 10

0.9583 1.0583 1.1583 1.2583 1.3583

Columns 11 through 15

1.4583 1.5587 1.6589 1.7593 1.8595

Columns 16 through 20

1.9599 2.0601 2.1605 2.2607 2.3611

Columns 21 through 25

2.4613 2.5615 2.6615 2.7615 2.8615

Column 26

2.9615

St =

Columns 1 through 5

0.4575 0.5575 0.6579 0.7581 0.8585

Columns 6 through 10

0.9585 1.0585 1.1585 1.2585 1.3585

Columns 11 through 15

1.4585 1.5585 1.6589 1.7591 1.8595

Columns 16 through 20

1.9597 2.0601 2.1603 2.2607 2.3609

Columns 21 through 25

2.4613 2.5615 2.6617 2.7617 2.8617

Column 26

2.9617

St =

Columns 1 through 5

0.4571 0.5575 0.6577 0.7581 0.8583

Columns 6 through 10

0.9587 1.0587 1.1587 1.2587 1.3587

Columns 11 through 15

1.4587 1.5587 1.6587 1.7591 1.8593

Columns 16 through 20

1.9597 2.0599 2.1603 2.2605 2.3609

Columns 21 through 25

2.4611 2.5615 2.6617 2.7619 2.8619

Column 26

2.9619

St =

Columns 1 through 5

0.4569 0.5573 0.6577 0.7579 0.8583

Columns 6 through 10

0.9585 1.0589 1.1589 1.2589 1.3589

Columns 11 through 15

1.4589 1.5589 1.6589 1.7589 1.8593

Columns 16 through 20

1.9595 2.0599 2.1601 2.2605 2.3607

Columns 21 through 25

2.4611 2.5613 2.6617 2.7619 2.8621

Column 26

2.9621

St =

Columns 1 through 5

0.4567 0.5571 0.6575 0.7579 0.8581

Columns 6 through 10

0.9585 1.0587 1.1591 1.2591 1.3591

Columns 11 through 15

1.4591 1.5591 1.6591 1.7591 1.8591

Columns 16 through 20

1.9595 2.0597 2.1601 2.2603 2.3607

Columns 21 through 25

2.4609 2.5613 2.6615 2.7619 2.8621

Column 26

2.9623

St =

Columns 1 through 5

0.4565 0.5569 0.6573 0.7577 0.8581

Columns 6 through 10

0.9583 1.0587 1.1589 1.2593 1.3593

Columns 11 through 15

1.4593 1.5593 1.6593 1.7593 1.8593

Columns 16 through 20

1.9593 2.0597 2.1599 2.2603 2.3605

Columns 21 through 25

2.4609 2.5611 2.6615 2.7617 2.8621

Column 26

2.9623

3) 2 rings of time delayed synapses, 1 in each direction, added to the model. Each delay set to 0.1 seconds. Shows a faster move towards synchronisation than the previous simulation.

St =

Columns 1 through 5

0.4563 0.5567 0.6567 0.7571 0.8571

Columns 6 through 10

0.9575 1.0575 1.1579 1.2579 1.3583

Columns 11 through 15

1.4583 1.5585 **1.6587** 1.7589 1.8591

Columns 16 through 20

1.9593 2.0595 2.1597 2.2599 2.3601

Columns 21 through 25

2.4603 2.5605 2.6607 2.7609 2.8611

Column 26

2.9613

St =

Columns 1 through 5

0.4565 0.5565 0.6569 0.7569 0.8573

Columns 6 through 10

0.9573 1.0577 1.1577 1.2581 1.3581

Columns 11 through 15

1.4585 1.5585 **1.6587** 1.7589 1.8591

Columns 16 through 20

1.9593 2.0595 2.1597 2.2599 2.3601

Columns 21 through 25

2.4603 2.5605 2.6607 2.7609 2.8611

Column 26

2.9613

St =

Columns 1 through 5

0.4567 0.5567 0.6567 0.7571 0.8571

Columns 6 through 10

0.9575 1.0575 1.1579 1.2579 1.3583

Columns 11 through 15

1.4583 1.5587 **1.6587** 1.7589 1.8591

Columns 16 through 20

1.9593 2.0595 2.1597 2.2599 2.3601

Columns 21 through 25

2.4603 2.5605 2.6607 2.7609 2.8611

Column 26

2.9613

St =

Columns 1 through 5

0.4569 0.5569 0.6569 0.7569 0.8573

Columns 6 through 10

0.9573 1.0577 1.1577 1.2581 1.3581

Columns 11 through 15

1.4585 1.5585 **1.6587** 1.7589 1.8591

Columns 16 through 20

1.9593 2.0595 2.1597 2.2599 2.3601

Columns 21 through 25

2.4603 2.5605 2.6607 2.7609 2.8611

Column 26

2.9613

St =

Columns 1 through 5

0.4571 0.5571 0.6571 0.7571 0.8571

Columns 6 through 10

0.9575 1.0575 1.1579 1.2579 1.3583

Columns 11 through 15

1.4583 1.5585 **1.6587** 1.7589 1.8591

Columns 16 through 20

1.9593 2.0595 2.1597 2.2599 2.3601

Columns 21 through 25

2.4603 2.5605 2.6607 2.7609 2.8611

Column 26

2.9613

St =

Columns 1 through 5

0.4573 0.5573 0.6573 0.7573 0.8573

Columns 6 through 10

0.9573 1.0577 1.1577 1.2581 1.3581

Columns 11 through 15

1.4583 1.5585 **1.6587** 1.7589 1.8591

Columns 16 through 20

1.9593 2.0595 2.1597 2.2599 2.3601

Columns 21 through 25

2.4603 2.5605 2.6607 2.7609 2.8611

Column 26

2.9613

St =

Columns 1 through 5

0.4575 0.5575 0.6575 0.7575 0.8575

Columns 6 through 10

0.9575 1.0575 1.1579 1.2579 1.3581

Columns 11 through 15

1.4583 1.5585 **1.6587** 1.7589 1.8591

Columns 16 through 20

1.9593 2.0595 2.1597 2.2599 2.3601

Columns 21 through 25

2.4603 2.5605 2.6607 2.7609 2.8611

Column 26

2.9613

St =

Columns 1 through 5

0.4577 0.5577 0.6577 0.7577 0.8577

Columns 6 through 10

0.9577 1.0577 1.1577 1.2579 1.3581

Columns 11 through 15

1.4583 1.5585 **1.6587** 1.7589 1.8591

Columns 16 through 20

1.9593 2.0595 2.1597 2.2599 2.3601

Columns 21 through 25

2.4603 2.5605 2.6607 2.7609 2.8611

Column 26

2.9613

St =

Columns 1 through 5

0.4577 0.5577 0.6577 0.7577 0.8577

Columns 6 through 10

0.9577 1.0577 1.1577 1.2579 1.3581

Columns 11 through 15

1.4583 1.5585 **1.6587** 1.7589 1.8591

Columns 16 through 20

1.9593 2.0595 2.1597 2.2599 2.3601

Columns 21 through 25

2.4603 2.5605 2.6607 2.7609 2.8611

Column 26

2.9613

St =

Columns 1 through 5

0.4575 0.5575 0.6575 0.7575 0.8575

Columns 6 through 10

0.9575 1.0575 1.1579 1.2579 1.3581

Columns 11 through 15

1.4583 1.5585 **1.6587** 1.7589 1.8591

Columns 16 through 20

1.9593 2.0595 2.1597 2.2599 2.3601

Columns 21 through 25

2.4603 2.5605 2.6607 2.7609 2.8611

Column 26

2.9613

St =

Columns 1 through 5

0.4573 0.5573 0.6573 0.7573 0.8573

Columns 6 through 10

0.9573 1.0577 1.1577 1.2579 1.3581

Columns 11 through 15

1.4583 1.5585 **1.6587** 1.7589 1.8591

Columns 16 through 20

1.9593 2.0595 2.1597 2.2599 2.3601

Columns 21 through 25

2.4603 2.5605 2.6607 2.7609 2.8611

Column 26

2.9613

St =

Columns 1 through 5

0.4575 0.5575 0.6575 0.7575 0.8575

Columns 6 through 10

0.9575 1.0575 1.1579 1.2579 1.3581

Columns 11 through 15

1.4583 1.5585 **1.6587** 1.7589 1.8591

Columns 16 through 20

1.9593 2.0595 2.1597 2.2599 2.3601

Columns 21 through 25

2.4603 2.5605 2.6607 2.7609 2.8611

Column 26

2.9613

St =

Columns 1 through 5

0.4571 0.5571 0.6571 0.7571 0.8571

Columns 6 through 10

0.9575 1.0575 1.1577 1.2579 1.3581

Columns 11 through 15

1.4583 1.5585 **1.6587** 1.7589 1.8591

Columns 16 through 20

1.9593 2.0595 2.1597 2.2599 2.3601

Columns 21 through 25

2.4603 2.5605 2.6607 2.7609 2.8611

Column 26

2.9613

St =

Columns 1 through 5

0.4569 0.5569 0.6569 0.7569 0.8573

Columns 6 through 10

0.9573 1.0577 1.1577 1.2579 1.3581

Columns 11 through 15

1.4583 1.5585 **1.6587** 1.7589 1.8591

Columns 16 through 20

1.9593 2.0595 2.1597 2.2599 2.3601

Columns 21 through 25

2.4603 2.5605 2.6607 2.7609 2.8611

Column 26

2.9613

St =

Columns 1 through 5

0.4567 0.5567 0.6567 0.7571 0.8571

Columns 6 through 10

0.9575 1.0575 1.1579 1.2579 1.3581

Columns 11 through 15

1.4583 1.5585 **1.6587** 1.7589 1.8591

Columns 16 through 20

1.9593 2.0595 2.1597 2.2599 2.3601

Columns 21 through 25

2.4603 2.5605 2.6607 2.7609 2.8611

Column 26

2.9613

St =

Columns 1 through 5

0.4565 0.5565 0.6569 0.7569 0.8573

Columns 6 through 10

0.9573 1.0577 1.1577 1.2581 1.3581

Columns 11 through 15

1.4583 1.5585 **1.6587** 1.7589 1.8591

Columns 16 through 20

1.9593 2.0595 2.1597 2.2599 2.3601

Columns 21 through 25

2.4603 2.5605 2.6607 2.7609 2.8611

Column 26

2.9613

A.1.2 Varying Synaptic Time Delay

4) 2 rings of time delayed synapses added, 1 in each direction around the ring. Time delays set to 0.05 seconds.

St =

Columns 1 through 5

0.4563 0.5067 0.5567 0.6071 0.6571

Columns 6 through 10

0.7075 0.7575 0.8079 0.8579 0.9083

Columns 11 through 15

0.9583 1.0085 **1.0587** 1.1089 1.1591

Columns 16 through 20

1.2093 1.2595 1.3097 1.3599 1.4101

Column 21

1.4603

St =

Columns 1 through 5

0.4565 0.5065 0.5569 0.6069 0.6573

Columns 6 through 10

0.7073 0.7577 0.8077 0.8581 0.9081

Columns 11 through 15

0.9585 1.0085 **1.0587** 1.1089 1.1591

Columns 16 through 20

1.2093 1.2595 1.3097 1.3599 1.4101

Column 21

1.4603

St =

Columns 1 through 5

0.4567 0.5067 0.5567 0.6071 0.6571

Columns 6 through 10

0.7075 0.7575 0.8079 0.8579 0.9083

Columns 11 through 15

0.9583 1.0087 **1.0587** 1.1089 1.1591

Columns 16 through 20

1.2093 1.2595 1.3097 1.3599 1.4101

Column 21

1.4603

St =

Columns 1 through 5

0.4569 0.5069 0.5569 0.6069 0.6573

Columns 6 through 10

0.7073 0.7577 0.8077 0.8581 0.9081

Columns 11 through 15

0.9585 1.0085 **1.0587** 1.1089 1.1591

Columns 16 through 20

1.2093 1.2595 1.3097 1.3599 1.4101

Column 21

1.4603

St =

Columns 1 through 5

0.4571 0.5071 0.5571 0.6071 0.6571

Columns 6 through 10

0.7075 0.7575 0.8079 0.8579 0.9083

Columns 11 through 15

0.9583 1.0085 **1.0587** 1.1089 1.1591

Columns 16 through 20

1.2093 1.2595 1.3097 1.3599 1.4101

Column 21

1.4603

St =

Columns 1 through 5

0.4573 0.5073 0.5573 0.6073 0.6573

Columns 6 through 10

0.7073 0.7577 0.8077 0.8581 0.9081

Columns 11 through 15

0.9583 1.0085 **1.0587** 1.1089 1.1591

Columns 16 through 20

1.2093 1.2595 1.3097 1.3599 1.4101

Column 21

1.4603

St =

Columns 1 through 5

0.4575 0.5075 0.5575 0.6075 0.6575

Columns 6 through 10

0.7075 0.7575 0.8079 0.8579 0.9081

Columns 11 through 15

0.9583 1.0085 **1.0587** 1.1089 1.1591

Columns 16 through 20

1.2093 1.2595 1.3097 1.3599 1.4101

Column 21

1.4603

St =

Columns 1 through 5

0.4577 0.5077 0.5577 0.6077 0.6577

Columns 6 through 10

0.7077 0.7577 0.8077 0.8579 0.9081

Columns 11 through 15

0.9583 1.0085 **1.0587** 1.1089 1.1591

Columns 16 through 20

1.2093 1.2595 1.3097 1.3599 1.4101

Column 21

1.4603

St =

Columns 1 through 5

0.4577 0.5077 0.5577 0.6077 0.6577

Columns 6 through 10

0.7077 0.7577 0.8077 0.8579 0.9081

Columns 11 through 15

0.9583 1.0085 **1.0587** 1.1089 1.1591

Columns 16 through 20

1.2093 1.2595 1.3097 1.3599 1.4101

Column 21

1.4603

St =

Columns 1 through 5

0.4575 0.5075 0.5575 0.6075 0.6575

Columns 6 through 10

0.7075 0.7575 0.8079 0.8579 0.9081

Columns 11 through 15

0.9583 1.0085 **1.0587** 1.1089 1.1591

Columns 16 through 20

1.2093 1.2595 1.3097 1.3599 1.4101

Column 21

1.4603

St =

Columns 1 through 5

0.4573 0.5073 0.5573 0.6073 0.6573

Columns 6 through 10

0.7073 0.7577 0.8077 0.8579 0.9081

Columns 11 through 15

0.9583 1.0085 **1.0587** 1.1089 1.1591

Columns 16 through 20

1.2093 1.2595 1.3097 1.3599 1.4101

Column 21

1.4603

St =

Columns 1 through 5

0.4575 0.5075 0.5575 0.6075 0.6575

Columns 6 through 10

0.7075 0.7575 0.8079 0.8579 0.9081

Columns 11 through 15

0.9583 1.0085 **1.0587** 1.1089 1.1591

Columns 16 through 20

1.2093 1.2595 1.3097 1.3599 1.4101

Column 21

1.4603

St =

Columns 1 through 5

0.4571 0.5071 0.5571 0.6071 0.6571

Columns 6 through 10

0.7075 0.7575 0.8077 0.8579 0.9081

Columns 11 through 15

0.9583 1.0085 **1.0587** 1.1089 1.1591

Columns 16 through 20

1.2093 1.2595 1.3097 1.3599 1.4101

Column 21

1.4603

St =

Columns 1 through 5

0.4569 0.5069 0.5569 0.6069 0.6573

Columns 6 through 10

0.7073 0.7577 0.8077 0.8579 0.9081

Columns 11 through 15

0.9583 1.0085 **1.0587** 1.1089 1.1591

Columns 16 through 20

1.2093 1.2595 1.3097 1.3599 1.4101

Column 21

1.4603

St =

Columns 1 through 5

0.4567 0.5067 0.5567 0.6071 0.6571

Columns 6 through 10

0.7075 0.7575 0.8079 0.8579 0.9081

Columns 11 through 15

0.9583 1.0085 **1.0587** 1.1089 1.1591

Columns 16 through 20

1.2093 1.2595 1.3097 1.3599 1.4101

Column 21

1.4603

St =

Columns 1 through 5

0.4565 0.5065 0.5569 0.6069 0.6573

Columns 6 through 10

0.7073 0.7577 0.8077 0.8581 0.9081

Columns 11 through 15

0.9583 1.0085 **1.0587** 1.1089 1.1591

Columns 16 through 20

1.2093 1.2595 1.3097 1.3599 1.4101

Column 21

1.4603

5) 2 rings of time delayed synapses added, 1 in each direction around the ring. Time delays set to 0.075 seconds.

St =

Columns 1 through 5

0.4563 0.5317 0.6067 0.6821 0.7571

Columns 6 through 10

0.8325 0.9075 0.9829 1.0579 1.1333

Columns 11 through 14

1.2083 1.2835 **1.3587** 1.4339

St =

Columns 1 through 5

0.4565 0.5315 0.6069 0.6819 0.7573

Columns 6 through 10

0.8323 0.9077 0.9827 1.0581 1.1331

Columns 11 through 14

1.2085 1.2835 **1.3587** 1.4339

St =

Columns 1 through 5

0.4567 0.5317 0.6067 0.6821 0.7571

Columns 6 through 10

0.8325 0.9075 0.9829 1.0579 1.1333

Columns 11 through 14

1.2083 1.2837 **1.3587** 1.4339

St =

Columns 1 through 5

0.4569 0.5319 0.6069 0.6819 0.7573

Columns 6 through 10

0.8323 0.9077 0.9827 1.0581 1.1331

Columns 11 through 14

1.2085 1.2835 **1.3587** 1.4339

St =

Columns 1 through 5

0.4571 0.5321 0.6071 0.6821 0.7571

Columns 6 through 10

0.8325 0.9075 0.9829 1.0579 1.1333

Columns 11 through 14

1.2083 1.2835 **1.3587** 1.4339

St =

Columns 1 through 5

0.4573 0.5323 0.6073 0.6823 0.7573

Columns 6 through 10

0.8323 0.9077 0.9827 1.0581 1.1331

Columns 11 through 14

1.2083 1.2835 **1.3587** 1.4339

St =

Columns 1 through 5

0.4575 0.5325 0.6075 0.6825 0.7575

Columns 6 through 10

0.8325 0.9075 0.9829 1.0579 1.1331

Columns 11 through 14

1.2083 1.2835 **1.3587** 1.4339

St =

Columns 1 through 5

0.4577 0.5327 0.6077 0.6827 0.7577

Columns 6 through 10

0.8327 0.9077 0.9827 1.0579 1.1331

Columns 11 through 14

1.2083 1.2835 **1.3587** 1.4339

St =

Columns 1 through 5

0.4577 0.5327 0.6077 0.6827 0.7577

Columns 6 through 10

0.8327 0.9077 0.9827 1.0579 1.1331

Columns 11 through 14

1.2083 1.2835 **1.3587** 1.4339

St =

Columns 1 through 5

0.4575 0.5325 0.6075 0.6825 0.7575

Columns 6 through 10

0.8325 0.9075 0.9829 1.0579 1.1331

Columns 11 through 14

1.2083 1.2835 **1.3587** 1.4339

St =

Columns 1 through 5

0.4573 0.5323 0.6073 0.6823 0.7573

Columns 6 through 10

0.8323 0.9077 0.9827 1.0579 1.1331

Columns 11 through 14

1.2083 1.2835 **1.3587** 1.4339

St =

Columns 1 through 5

0.4575 0.5325 0.6075 0.6825 0.7575

Columns 6 through 10

0.8325 0.9075 0.9829 1.0579 1.1331

Columns 11 through 14

1.2083 1.2835 **1.3587** 1.4339

St =

Columns 1 through 5

0.4571 0.5321 0.6071 0.6821 0.7571

Columns 6 through 10

0.8325 0.9075 0.9827 1.0579 1.1331

Columns 11 through 14

1.2083 1.2835 **1.3587** 1.4339

St =

Columns 1 through 5

0.4569 0.5319 0.6069 0.6819 0.7573

Columns 6 through 10

0.8323 0.9077 0.9827 1.0579 1.1331

Columns 11 through 14

1.2083 1.2835 **1.3587** 1.4339

St =

Columns 1 through 5

0.4567 0.5317 0.6067 0.6821 0.7571

Columns 6 through 10

0.8325 0.9075 0.9829 1.0579 1.1331

Columns 11 through 14

1.2083 1.2835 **1.3587** 1.4339

St =

Columns 1 through 5

0.4565 0.5315 0.6069 0.6819 0.7573

Columns 6 through 10

0.8323 0.9077 0.9827 1.0581 1.1331

Columns 11 through 14

1.2083 1.2835 **1.3587** 1.4339

A.1.3 Varying Refractory Period

6) Both rings of time delayed synapses are present with a time delay of 0.1 seconds throughout. Refractory period set to 0.02 seconds

St =

Columns 1 through 5

0.4563 0.5567 0.6567 0.7571 0.8571

Columns 6 through 10

0.9575 1.0575 1.1579 1.2579 1.3583

Columns 11 through 15

1.4583 1.5585 **1.6587** 1.7589 1.8591

Columns 16 through 20

1.9593 2.0595 2.1597 2.2599 2.3601

Columns 21 through 25

2.4603 2.5605 2.6607 2.7609 2.8611

Column 26

2.9613

St =

Columns 1 through 5

0.4565 0.5565 0.6569 0.7569 0.8573

Columns 6 through 10

0.9573 1.0577 1.1577 1.2581 1.3581
Columns 11 through 15

1.4585 1.5585 **1.6587** 1.7589 1.8591

Columns 16 through 20

1.9593 2.0595 2.1597 2.2599 2.3601

Columns 21 through 25

2.4603 2.5605 2.6607 2.7609 2.8611

Column 26

2.9613

St =

Columns 1 through 5

0.4567 0.5567 0.6567 0.7571 0.8571

Columns 6 through 10

0.9575 1.0575 1.1579 1.2579 1.3583

Columns 11 through 15

1.4583 1.5587 **1.6587** 1.7589 1.8591

Columns 16 through 20

1.9593 2.0595 2.1597 2.2599 2.3601

Columns 21 through 25

2.4603 2.5605 2.6607 2.7609 2.8611

Column 26

2.9613

St =

Columns 1 through 5

0.4569 0.5569 0.6569 0.7569 0.8573

Columns 6 through 10

0.9573 1.0577 1.1577 1.2581 1.3581

Columns 11 through 15

1.4585 1.5585 **1.6587** 1.7589 1.8591

Columns 16 through 20

1.9593 2.0595 2.1597 2.2599 2.3601

Columns 21 through 25

2.4603 2.5605 2.6607 2.7609 2.8611

Column 26

2.9613

St =

Columns 1 through 5

0.4571 0.5571 0.6571 0.7571 0.8571

Columns 6 through 10

0.9575 1.0575 1.1579 1.2579 1.3583

Columns 11 through 15

1.4583 1.5585 **1.6587** 1.7589 1.8591

Columns 16 through 20

1.9593 2.0595 2.1597 2.2599 2.3601

Columns 21 through 25

2.4603 2.5605 2.6607 2.7609 2.8611

Column 26

2.9613

St =

Columns 1 through 5

0.4573 0.5573 0.6573 0.7573 0.8573

Columns 6 through 10

0.9573 1.0577 1.1577 1.2581 1.3581

Columns 11 through 15

1.4583 1.5585 **1.6587** 1.7589 1.8591

Columns 16 through 20

1.9593 2.0595 2.1597 2.2599 2.3601

Columns 21 through 25

2.4603 2.5605 2.6607 2.7609 2.8611

Column 26

2.9613

St =

Columns 1 through 5

0.4575 0.5575 0.6575 0.7575 0.8575

Columns 6 through 10

0.9575 1.0575 1.1579 1.2579 1.3581

Columns 11 through 15

1.4583 1.5585 **1.6587** 1.7589 1.8591

Columns 16 through 20

1.9593 2.0595 2.1597 2.2599 2.3601

Columns 21 through 25

2.4603 2.5605 2.6607 2.7609 2.8611

Column 26

2.9613

St =

Columns 1 through 5

0.4577 0.5577 0.6577 0.7577 0.8577

Columns 6 through 10

0.9577 1.0577 1.1577 1.2579 1.3581

Columns 11 through 15

1.4583 1.5585 **1.6587** 1.7589 1.8591

Columns 16 through 20

1.9593 2.0595 2.1597 2.2599 2.3601

Columns 21 through 25

2.4603 2.5605 2.6607 2.7609 2.8611

Column 26

2.9613

St =

Columns 1 through 5

0.4577 0.5577 0.6577 0.7577 0.8577

Columns 6 through 10

0.9577 1.0577 1.1577 1.2579 1.3581

Columns 11 through 15

1.4583 1.5585 **1.6587** 1.7589 1.8591

Columns 16 through 20

1.9593 2.0595 2.1597 2.2599 2.3601

Columns 21 through 25

2.4603 2.5605 2.6607 2.7609 2.8611

Column 26

2.9613

St =

Columns 1 through 5

0.4575 0.5575 0.6575 0.7575 0.8575

Columns 6 through 10

0.9575 1.0575 1.1579 1.2579 1.3581

Columns 11 through 15

1.4583 1.5585 **1.6587** 1.7589 1.8591

Columns 16 through 20

1.9593 2.0595 2.1597 2.2599 2.3601

Columns 21 through 25

2.4603 2.5605 2.6607 2.7609 2.8611

Column 26

2.9613

St =

Columns 1 through 5

0.4573 0.5573 0.6573 0.7573 0.8573

Columns 6 through 10

0.9573 1.0577 1.1577 1.2579 1.3581

Columns 11 through 15

1.4583 1.5585 **1.6587** 1.7589 1.8591

Columns 16 through 20

1.9593 2.0595 2.1597 2.2599 2.3601

Columns 21 through 25

2.4603 2.5605 2.6607 2.7609 2.8611

Column 26

2.9613

St =

Columns 1 through 5

0.4575 0.5575 0.6575 0.7575 0.8575

Columns 6 through 10

0.9575 1.0575 1.1579 1.2579 1.3581

Columns 11 through 15

1.4583 1.5585 **1.6587** 1.7589 1.8591

Columns 16 through 20

1.9593 2.0595 2.1597 2.2599 2.3601

Columns 21 through 25

2.4603 2.5605 2.6607 2.7609 2.8611

Column 26

2.9613

St =

Columns 1 through 5

0.4571 0.5571 0.6571 0.7571 0.8571

Columns 6 through 10

0.9575 1.0575 1.1577 1.2579 1.3581

Columns 11 through 15

1.4583 1.5585 **1.6587** 1.7589 1.8591

Columns 16 through 20

1.9593 2.0595 2.1597 2.2599 2.3601

Columns 21 through 25

2.4603 2.5605 2.6607 2.7609 2.8611

Column 26

2.9613

St =

Columns 1 through 5

0.4569 0.5569 0.6569 0.7569 0.8573

Columns 6 through 10

0.9573 1.0577 1.1577 1.2579 1.3581

Columns 11 through 15

1.4583 1.5585 **1.6587** 1.7589 1.8591

Columns 16 through 20

1.9593 2.0595 2.1597 2.2599 2.3601

Columns 21 through 25

2.4603 2.5605 2.6607 2.7609 2.8611

Column 26

2.9613

St =

Columns 1 through 5

0.4567 0.5567 0.6567 0.7571 0.8571

Columns 6 through 10

0.9575 1.0575 1.1579 1.2579 1.3581

Columns 11 through 15

1.4583 1.5585 **1.6587** 1.7589 1.8591

Columns 16 through 20

1.9593 2.0595 2.1597 2.2599 2.3601

Columns 21 through 25

2.4603 2.5605 2.6607 2.7609 2.8611

Column 26

2.961

St =

Columns 1 through 5

0.4565 0.5565 0.6569 0.7569 0.8573

Columns 6 through 10

0.9573 1.0577 1.1577 1.2581 1.3581

Columns 11 through 15

1.4583 1.5585 **1.6587** 1.7589 1.8591

Columns 16 through 20

1.9593 2.0595 2.1597 2.2599 2.3601

Columns 21 through 25

2.4603 2.5605 2.6607 2.7609 2.8611

Column 26

2.9613

7) Both rings of time synapses are present with a time delay of 0.1 seconds throughout.

Refractory Period changed to 0.01 seconds (these are the final lines of output from the 6 second simulation, the rest of the output is not included as there were 544 spikes produced by each neuron resulting in thousands of lines of output. The point is to show that the network had not synchronised after 6 seconds). These results were exactly the same when the time delay was changed to 0.05 and 0.01 seconds

Columns 541 through 544

St = 5.9645 5.9747 5.9849 5.9951
St = 5.9647 5.9749 5.9851 5.9953
St = 5.9649 5.9751 5.9853 5.9955
St = 5.9651 5.9753 5.9855 5.9957
St = 5.9655 5.9757 5.9859 5.9961
St = 5.9653 5.9755 5.9857 5.9959
St = 5.9655 5.9757 5.9859 5.9961
St = 5.9657 5.9759 5.9861 5.9963
St = 5.9657 5.9759 5.9861 5.9963
St = 5.9655 5.9757 5.9859 5.9961
St = 5.9653 5.9755 5.9857 5.9959
St = 5.9655 5.9757 5.9859 5.9961
St = 5.9651 5.9753 5.9855 5.9957
St = 5.9649 5.9751 5.9853 5.9955
St = 5.9647 5.9749 5.9851 5.9953
St = 5.9645 5.9747 5.9849 5.9951

A.1.4 Varying Synaptic Time Delay Once Again

8) With both rings of time delayed synapses present, the time delays were set to 0.03 seconds throughout and refractory period set to 0.02 seconds.

St =

Columns 1 through 5

0.4563 0.4867 0.5167 0.5471 0.5771

Columns 6 through 10

0.6075 0.6375 0.6679 0.6979 0.7283

Columns 11 through 15

0.7583 0.7885 **0.8187** 0.8489 0.8791

Columns 16 through 19

0.9093 0.9395 0.9697 0.9999

St =

Columns 1 through 5

0.4565 0.4865 0.5169 0.5469 0.5773

Columns 6 through 10

0.6073 0.6377 0.6677 0.6981 0.7281

Columns 11 through 15

0.7585 0.7885 **0.8187** 0.8489 0.8791

Columns 16 through 19

0.9093 0.9395 0.9697 0.9999

St =

Columns 1 through 5

0.4567 0.4867 0.5167 0.5471 0.5771

Columns 6 through 10

0.6075 0.6375 0.6679 0.6979 0.7283

Columns 11 through 15

0.7583 0.7887 **0.8187** 0.8489 0.8791

Columns 16 through 19

0.9093 0.9395 0.9697 0.9999

St =

Columns 1 through 5

0.4569 0.4869 0.5169 0.5469 0.5773

Columns 6 through 10

0.6073 0.6377 0.6677 0.6981 0.7281
Columns 11 through 15

0.7585 0.7885 **0.8187** 0.8489 0.8791

Columns 16 through 19

0.9093 0.9395 0.9697 0.9999

St =

Columns 1 through 5

0.4571 0.4871 0.5171 0.5471 0.5771

Columns 6 through 10

0.6075 0.6375 0.6679 0.6979 0.7283

Columns 11 through 15

0.7583 0.7885 **0.8187** 0.8489 0.8791

Columns 16 through 19

0.9093 0.9395 0.9697 0.9999

St =

Columns 1 through 5

0.4573 0.4873 0.5173 0.5473 0.5773

Columns 6 through 10

0.6073 0.6377 0.6677 0.6981 0.7281

Columns 11 through 15

0.7583 0.7885 **0.8187** 0.8489 0.8791

Columns 16 through 19

0.9093 0.9395 0.9697 0.9999

St =

Columns 1 through 5

0.4575 0.4875 0.5175 0.5475 0.5775

Columns 6 through 10

0.6075 0.6375 0.6679 0.6979 0.7281

Columns 11 through 15

0.7583 0.7885 **0.8187** 0.8489 0.8791

Columns 16 through 19

0.9093 0.9395 0.9697 0.9999

St =

Columns 1 through 5

0.4577 0.4877 0.5177 0.5477 0.5777

Columns 6 through 10

0.6077 0.6377 0.6677 0.6979 0.7281
Columns 11 through 15

0.7583 0.7885 **0.8187** 0.8489 0.8791

Columns 16 through 19

0.9093 0.9395 0.9697 0.9999

St =

Columns 1 through 5

0.4577 0.4877 0.5177 0.5477 0.5777

Columns 6 through 10

0.6077 0.6377 0.6677 0.6979 0.7281

Columns 11 through 15

0.7583 0.7885 **0.8187** 0.8489 0.8791

Columns 16 through 19

0.9093 0.9395 0.9697 0.9999

St =

Columns 1 through 5

0.4575 0.4875 0.5175 0.5475 0.5775

Columns 6 through 10

0.6075 0.6375 0.6679 0.6979 0.7281

Columns 11 through 15

0.7583 0.7885 **0.8187** 0.8489 0.8791

Columns 16 through 19

0.9093 0.9395 0.9697 0.9999

St =

Columns 1 through 5

0.4573 0.4873 0.5173 0.5473 0.5773

Columns 6 through 10

0.6073 0.6377 0.6677 0.6979 0.7281

Columns 11 through 15

0.7583 0.7885 **0.8187** 0.8489 0.8791

Columns 16 through 19

0.9093 0.9395 0.9697 0.9999

St =

Columns 1 through 5

0.4575 0.4875 0.5175 0.5475 0.5775

Columns 6 through 10

0.6075 0.6375 0.6679 0.6979 0.7281
Columns 11 through 15

0.7583 0.7885 **0.8187** 0.8489 0.8791

Columns 16 through 19

0.9093 0.9395 0.9697 0.9999

St =

Columns 1 through 5

0.4571 0.4871 0.5171 0.5471 0.5771

Columns 6 through 10

0.6075 0.6375 0.6677 0.6979 0.7281

Columns 11 through 15

0.7583 0.7885 **0.8187** 0.8489 0.8791

Columns 16 through 19

0.9093 0.9395 0.9697 0.9999

St =

Columns 1 through 5

0.4569 0.4869 0.5169 0.5469 0.5773

Columns 6 through 10

0.6073 0.6377 0.6677 0.6979 0.7281

Columns 11 through 15

0.7583 0.7885 **0.8187** 0.8489 0.8791

Columns 16 through 19

0.9093 0.9395 0.9697 0.9999

St =

Columns 1 through 5

0.4567 0.4867 0.5167 0.5471 0.5771

Columns 6 through 10

0.6075 0.6375 0.6679 0.6979 0.7281

Columns 11 through 15

0.7583 0.7885 **0.8187** 0.8489 0.8791

Columns 16 through 19

0.9093 0.9395 0.9697 0.9999

St =

Columns 1 through 5

0.4565 0.4865 0.5169 0.5469 0.5773

Columns 6 through 10

0.6073 0.6377 0.6677 0.6981 0.7281
Columns 11 through 15

0.7583 0.7885 **0.8187** 0.8489 0.8791

Columns 16 through 19

0.9093 0.9395 0.9697 0.9999

7 Appendix B

Code for the Model

This is the code for the final model of the jellyfish. It has all 4 rings of neurons added.

```
clear all
close all

addpath('.');

% Create Input neuron
inputNeuron=csim('create','SpikingInputNeuron');

% Create input synapse
inputSynapse=csim('create','StaticSpikingSynapse');

% Create the synapses
synapse1=csim('create','StaticSpikingSynapse');
synapse2=csim('create','StaticSpikingSynapse');
synapse3=csim('create','StaticSpikingSynapse');
synapse4=csim('create','StaticSpikingSynapse');
synapse5=csim('create','StaticSpikingSynapse');
synapse6=csim('create','StaticSpikingSynapse');
synapse7=csim('create','StaticSpikingSynapse');
synapse8=csim('create','StaticSpikingSynapse');
synapse9=csim('create','StaticSpikingSynapse');
synapse10=csim('create','StaticSpikingSynapse');
synapse11=csim('create','StaticSpikingSynapse');
synapse12=csim('create','StaticSpikingSynapse');
synapse13=csim('create','StaticSpikingSynapse');
synapse14=csim('create','StaticSpikingSynapse');
synapse15=csim('create','StaticSpikingSynapse');
synapse16=csim('create','StaticSpikingSynapse');
synapse17=csim('create','StaticSpikingSynapse');
synapse18=csim('create','StaticSpikingSynapse');
synapse19=csim('create','StaticSpikingSynapse');
synapse20=csim('create','StaticSpikingSynapse');
synapse21=csim('create','StaticSpikingSynapse');
synapse22=csim('create','StaticSpikingSynapse');
synapse23=csim('create','StaticSpikingSynapse');
synapse24=csim('create','StaticSpikingSynapse');
synapse25=csim('create','StaticSpikingSynapse');
synapse26=csim('create','StaticSpikingSynapse');
synapse27=csim('create','StaticSpikingSynapse');
synapse28=csim('create','StaticSpikingSynapse');
synapse29=csim('create','StaticSpikingSynapse');
synapse30=csim('create','StaticSpikingSynapse');
synapse31=csim('create','StaticSpikingSynapse');
synapse32=csim('create','StaticSpikingSynapse');
synapseA=csim('create','StaticSpikingSynapse');
synapseB=csim('create','StaticSpikingSynapse');
synapseC=csim('create','StaticSpikingSynapse');
synapseD=csim('create','StaticSpikingSynapse');
synapseE=csim('create','StaticSpikingSynapse');
synapseF=csim('create','StaticSpikingSynapse');
```

```

synapseG=csim('create','StaticSpikingSynapse');
synapseH=csim('create','StaticSpikingSynapse');
synapseI=csim('create','StaticSpikingSynapse');
synapseJ=csim('create','StaticSpikingSynapse');
synapseK=csim('create','StaticSpikingSynapse');
synapseL=csim('create','StaticSpikingSynapse');
synapseM=csim('create','StaticSpikingSynapse');
synapseN=csim('create','StaticSpikingSynapse');
synapseO=csim('create','StaticSpikingSynapse');
synapseP=csim('create','StaticSpikingSynapse');
synapseA2=csim('create','StaticSpikingSynapse');
synapseB2=csim('create','StaticSpikingSynapse');
synapseC2=csim('create','StaticSpikingSynapse');
synapseD2=csim('create','StaticSpikingSynapse');
synapseE2=csim('create','StaticSpikingSynapse');
synapseF2=csim('create','StaticSpikingSynapse');
synapseG2=csim('create','StaticSpikingSynapse');
synapseH2=csim('create','StaticSpikingSynapse');
synapseI2=csim('create','StaticSpikingSynapse');
synapseJ2=csim('create','StaticSpikingSynapse');
synapseK2=csim('create','StaticSpikingSynapse');
synapseL2=csim('create','StaticSpikingSynapse');
synapseM2=csim('create','StaticSpikingSynapse');
synapseN2=csim('create','StaticSpikingSynapse');
synapseO2=csim('create','StaticSpikingSynapse');
synapseP2=csim('create','StaticSpikingSynapse');

```

% Create the neurons

```

neuron1=csim('create','LifNeuron');
neuron2=csim('create','LifNeuron');
neuron3=csim('create','LifNeuron');
neuron4=csim('create','LifNeuron');
neuron5=csim('create','LifNeuron');
neuron6=csim('create','LifNeuron');
neuron7=csim('create','LifNeuron');
neuron8=csim('create','LifNeuron');
neuron9=csim('create','LifNeuron');
neuron10=csim('create','LifNeuron');
neuron11=csim('create','LifNeuron');
neuron12=csim('create','LifNeuron');
neuron13=csim('create','LifNeuron');
neuron14=csim('create','LifNeuron');
neuron15=csim('create','LifNeuron');
neuron16=csim('create','LifNeuron');

```

```

csim('set','randSeed',123456);

```

% The refractory period variable

```

refract = 0.02;

```

% Set the parameters of the neurons

```

csim('set',neuron1,'Trefract',refract,'Inoise',50e-9);
csim('set',neuron2,'Trefract',refract,'Inoise',50e-9);
csim('set',neuron3,'Trefract',refract,'Inoise',50e-9);
csim('set',neuron4,'Trefract',refract,'Inoise',50e-9);
csim('set',neuron5,'Trefract',refract,'Inoise',50e-9);
csim('set',neuron6,'Trefract',refract,'Inoise',50e-9);
csim('set',neuron7,'Trefract',refract,'Inoise',50e-9);

```

```

csim('set',neuron8,'Trefract',refract,'Inoise',50e-9);
csim('set',neuron9,'Trefract',refract,'Inoise',50e-9);
csim('set',neuron10,'Trefract',refract,'Inoise',50e-9);
csim('set',neuron11,'Trefract',refract,'Inoise',50e-9);
csim('set',neuron12,'Trefract',refract,'Inoise',50e-9);
csim('set',neuron13,'Trefract',refract,'Inoise',50e-9);
csim('set',neuron14,'Trefract',refract,'Inoise',50e-9);
csim('set',neuron15,'Trefract',refract,'Inoise',50e-9);
csim('set',neuron16,'Trefract',refract,'Inoise',50e-9);

```

```

% The time delay variable

```

```

delay = 0.05;

```

```

% Set the parameters of the synapses

```

```

csim('set',synapse1,'W',1,'delay',0);
csim('set',synapse2,'W',1,'delay',0);
csim('set',synapse3,'W',1,'delay',0);
csim('set',synapse4,'W',1,'delay',0);
csim('set',synapse5,'W',1,'delay',0);
csim('set',synapse6,'W',1,'delay',0);
csim('set',synapse7,'W',1,'delay',0);
csim('set',synapse8,'W',1,'delay',0);
csim('set',synapse9,'W',1,'delay',0);
csim('set',synapse10,'W',1,'delay',0);
csim('set',synapse11,'W',1,'delay',0);
csim('set',synapse12,'W',1,'delay',0);
csim('set',synapse13,'W',1,'delay',0);
csim('set',synapse14,'W',1,'delay',0);
csim('set',synapse15,'W',1,'delay',0);
csim('set',synapse16,'W',1,'delay',0);
csim('set',synapse17,'W',1,'delay',0);
csim('set',synapse18,'W',1,'delay',0);
csim('set',synapse19,'W',1,'delay',0);
csim('set',synapse20,'W',1,'delay',0);
csim('set',synapse21,'W',1,'delay',0);
csim('set',synapse22,'W',1,'delay',0);
csim('set',synapse23,'W',1,'delay',0);
csim('set',synapse24,'W',1,'delay',0);
csim('set',synapse25,'W',1,'delay',0);
csim('set',synapse26,'W',1,'delay',0);
csim('set',synapse27,'W',1,'delay',0);
csim('set',synapse28,'W',1,'delay',0);
csim('set',synapse29,'W',1,'delay',0);
csim('set',synapse30,'W',1,'delay',0);
csim('set',synapse31,'W',1,'delay',0);
csim('set',synapse32,'W',1,'delay',0);
csim('set',synapseA,'W',1,'delay',delay);
csim('set',synapseB,'W',1,'delay',delay);
csim('set',synapseC,'W',1,'delay',delay);
csim('set',synapseD,'W',1,'delay',delay);
csim('set',synapseE,'W',1,'delay',delay);
csim('set',synapseF,'W',1,'delay',delay);
csim('set',synapseG,'W',1,'delay',delay);
csim('set',synapseH,'W',1,'delay',delay);
csim('set',synapseI,'W',1,'delay',delay);
csim('set',synapseJ,'W',1,'delay',delay);
csim('set',synapseK,'W',1,'delay',delay);
csim('set',synapseL,'W',1,'delay',delay);
csim('set',synapseM,'W',1,'delay',delay);
csim('set',synapseN,'W',1,'delay',delay);

```

```

csim('set',synapseO,'W',1,'delay',delay);
csim('set',synapseP,'W',1,'delay',delay);
csim('set',synapseA2,'W',1,'delay',delay);
csim('set',synapseB2,'W',1,'delay',delay);
csim('set',synapseC2,'W',1,'delay',delay);
csim('set',synapseD2,'W',1,'delay',delay);
csim('set',synapseE2,'W',1,'delay',delay);
csim('set',synapseF2,'W',1,'delay',delay);
csim('set',synapseG2,'W',1,'delay',delay);
csim('set',synapseH2,'W',1,'delay',delay);
csim('set',synapseI2,'W',1,'delay',delay);
csim('set',synapseJ2,'W',1,'delay',delay);
csim('set',synapseK2,'W',1,'delay',delay);
csim('set',synapseL2,'W',1,'delay',delay);
csim('set',synapseM2,'W',1,'delay',delay);
csim('set',synapseN2,'W',1,'delay',delay);
csim('set',synapseO2,'W',1,'delay',delay);
csim('set',synapseP2,'W',1,'delay',delay);
csim('set',inputSynapse,'W',1,'delay',0);

```

% Connect the synapses and neurons up to form the ring

% These are the connections going round in the clockwise direction. These are the synapses with no time delays, in the diagram of the network this would be the ring represented by the red line.

```

csim('connect',synapse16,neuron16);
csim('connect',neuron16,synapse15);

```

```

csim('connect',synapse15,neuron15);
csim('connect',neuron15,synapse14);

```

```

csim('connect',synapse14,neuron14);
csim('connect',neuron14,synapse13);

```

```

csim('connect',synapse13,neuron13);
csim('connect',neuron13,synapse12);

```

```

csim('connect',synapse12,neuron12);
csim('connect',neuron12,synapse11);

```

```

csim('connect',synapse11,neuron11);
csim('connect',neuron11,synapse10);

```

```

csim('connect',synapse10,neuron10);
csim('connect',neuron10,synapse9);

```

```

csim('connect',synapse9,neuron9);
csim('connect',neuron9,synapse8);

```

```

csim('connect',synapse8,neuron8);
csim('connect',neuron8,synapse7);

```

```

csim('connect',synapse7,neuron7);
csim('connect',neuron7,synapse6);

```

```

csim('connect',synapse6,neuron6);
csim('connect',neuron6,synapse5);

```

```

csim('connect',synapse5,neuron5);
csim('connect',neuron5,synapse4);

```

```

csim('connect',synapse4,neuron4);
csim('connect',neuron4,synapse3);

csim('connect',synapse3,neuron3);
csim('connect',neuron3,synapse2);

csim('connect',synapse2,neuron2);
csim('connect',neuron2,synapse1);

csim('connect',synapse1,neuron1);
%Close the loop
csim('connect',neuron1,synapse16);

```

% These are the connections going round in the anti-clockwise direction. These are the synapses with no time delays, in the diagram of the network this would be the ring represented by the orange line.

```

csim('connect',synapse17,neuron1);
csim('connect',neuron1,synapse18);

csim('connect',synapse18,neuron2);
csim('connect',neuron2,synapse19);

csim('connect',synapse19,neuron3);
csim('connect',neuron3,synapse20);

csim('connect',synapse20,neuron4);
csim('connect',neuron4,synapse21);

csim('connect',synapse21,neuron5);
csim('connect',neuron5,synapse22);

csim('connect',synapse22,neuron6);
csim('connect',neuron6,synapse23);

csim('connect',synapse23,neuron7);
csim('connect',neuron7,synapse24);

csim('connect',synapse24,neuron8);
csim('connect',neuron8,synapse25);

csim('connect',synapse25,neuron9);
csim('connect',neuron9,synapse26);

csim('connect',synapse26,neuron10);
csim('connect',neuron10,synapse27);

csim('connect',synapse27,neuron11);
csim('connect',neuron11,synapse28);

csim('connect',synapse28,neuron12);
csim('connect',neuron11,synapse29);

csim('connect',synapse29,neuron13);
csim('connect',neuron13,synapse30);

csim('connect',synapse30,neuron14);
csim('connect',neuron14,synapse31);

csim('connect',synapse31,neuron15);

```

```
csim('connect',neuron15,synapse32);
```

```
csim('connect',synapse32,neuron16);  
csim('connect',neuron16,synapse17);
```

% These are the connections going round in the clockwise direction. These are the synapses with time delays, in the diagram of the network this would be the ring represented by the blue line.

```
csim('connect',synapseP,neuron16);  
csim('connect',neuron16,synapseO);
```

```
csim('connect',synapseO,neuron15);  
csim('connect',neuron15,synapseN);
```

```
csim('connect',synapseN,neuron14);  
csim('connect',neuron14,synapseM);
```

```
csim('connect',synapseM,neuron13);  
csim('connect',neuron13,synapseL);
```

```
csim('connect',synapseL,neuron12);  
csim('connect',neuron12,synapseK);
```

```
csim('connect',synapseK,neuron11);  
csim('connect',neuron11,synapseJ);
```

```
csim('connect',synapseJ,neuron10);  
csim('connect',neuron10,synapseI);
```

```
csim('connect',synapseI,neuron9);  
csim('connect',neuron9,synapseH);
```

```
csim('connect',synapseH,neuron8);  
csim('connect',neuron8,synapseG);
```

```
csim('connect',synapseG,neuron7);  
csim('connect',neuron7,synapseF);
```

```
csim('connect',synapseF,neuron6);  
csim('connect',neuron6,synapseE);
```

```
csim('connect',synapseE,neuron5);  
csim('connect',neuron5,synapseD);
```

```
csim('connect',synapseD,neuron4);  
csim('connect',neuron4,synapseC);
```

```
csim('connect',synapseC,neuron3);  
csim('connect',neuron3,synapseB);
```

```
csim('connect',synapseB,neuron2);  
csim('connect',neuron2,synapseA);
```

```
csim('connect',synapseA,neuron1);  
%Close the loop  
csim('connect',neuron1,synapseP);
```

% These are the connections going round in the anti-clockwise direction. These are the synapses with time delays, in the diagram of the network this would be the ring represented by the green line.

```

csim('connect',synapseA2,neuron1);
csim('connect',neuron1,synapseB2);

csim('connect',synapseB2,neuron2);
csim('connect',neuron2,synapseC2);

csim('connect',synapseC2,neuron3);
csim('connect',neuron3,synapseD2);

csim('connect',synapseD2,neuron4);
csim('connect',neuron4,synapseE2);

csim('connect',synapseE2,neuron5);
csim('connect',neuron5,synapseF2);

csim('connect',synapseF2,neuron6);
csim('connect',neuron6,synapseG2);

csim('connect',synapseG2,neuron7);
csim('connect',neuron7,synapseH2);

csim('connect',synapseH2,neuron8);
csim('connect',neuron8,synapseI2);

csim('connect',synapseI2,neuron9);
csim('connect',neuron9,synapseJ2);

csim('connect',synapseJ2,neuron10);
csim('connect',neuron10,synapseK2);

csim('connect',synapseK2,neuron11);
csim('connect',neuron11,synapseL2);

csim('connect',synapseL2,neuron12);
csim('connect',neuron11,synapseM2);

csim('connect',synapseM2,neuron13);
csim('connect',neuron13,synapseN2);

csim('connect',synapseN2,neuron14);
csim('connect',neuron14,synapseO2);

csim('connect',synapseO2,neuron15);
csim('connect',neuron15,synapseP2);

csim('connect',synapseP2,neuron16);
csim('connect',neuron16,synapseA2);

%Connect the input neuron and synapse
csim('connect',neuron1,inputSynapse);
csim('connect',inputSynapse,inputNeuron);

% Create the recorders to be added to the neurons and synapses.
recorder=csim('create','Recorder');
csim('set',recorder,'dt',0.5e-3);
recorder2=csim('create','Recorder');
csim('set',recorder2,'dt',0.5e-3);
recorder3=csim('create','Recorder');

```

```

csim('set',recorder3,'dt',0.5e-3);
recorder4=csim('create','Recorder');
csim('set',recorder4,'dt',0.5e-3);
recorder5=csim('create','Recorder');
csim('set',recorder5,'dt',0.5e-3);
recorder6=csim('create','Recorder');
csim('set',recorder6,'dt',0.5e-3);
recorder7=csim('create','Recorder');
csim('set',recorder7,'dt',0.5e-3);
recorder8=csim('create','Recorder');
csim('set',recorder8,'dt',0.5e-3);
recorder9=csim('create','Recorder');
csim('set',recorder9,'dt',0.5e-3);
recorder10=csim('create','Recorder');
csim('set',recorder10,'dt',0.5e-3);
recorder11=csim('create','Recorder');
csim('set',recorder11,'dt',0.5e-3);
recorder12=csim('create','Recorder');
csim('set',recorder12,'dt',0.5e-3);
recorder13=csim('create','Recorder');
csim('set',recorder13,'dt',0.5e-3);
recorder14=csim('create','Recorder');
csim('set',recorder14,'dt',0.5e-3);
recorder15=csim('create','Recorder');
csim('set',recorder15,'dt',0.5e-3);
recorder16=csim('create','Recorder');
csim('set',recorder16,'dt',0.5e-3);

```

% Connect the recorders to the neurons and synapses.

```

csim('connect',recorder,synapse1,'psr');
csim('connect',recorder,neuron1,'Vm');
csim('connect',recorder,neuron1,'spikes');

csim('connect',recorder2,synapse2,'psr');
csim('connect',recorder2,neuron2,'Vm');
csim('connect',recorder2,neuron2,'spikes');

csim('connect',recorder3,synapse3,'psr');
csim('connect',recorder3,neuron3,'Vm');
csim('connect',recorder3,neuron3,'spikes');

csim('connect',recorder4,synapse4,'psr');
csim('connect',recorder4,neuron4,'Vm');
csim('connect',recorder4,neuron4,'spikes');

csim('connect',recorder5,synapse5,'psr');
csim('connect',recorder5,neuron5,'Vm');
csim('connect',recorder5,neuron5,'spikes');

csim('connect',recorder6,synapse6,'psr');
csim('connect',recorder6,neuron6,'Vm');
csim('connect',recorder6,neuron6,'spikes');

csim('connect',recorder7,synapse7,'psr');
csim('connect',recorder7,neuron7,'Vm');
csim('connect',recorder7,neuron7,'spikes');

csim('connect',recorder8,synapse8,'psr');
csim('connect',recorder8,neuron8,'Vm');

```

```

csim('connect',recorder8,neuron8,'spikes');

csim('connect',recorder9,synapse9,'psr');
csim('connect',recorder9,neuron9,'Vm');
csim('connect',recorder9,neuron9,'spikes');

csim('connect',recorder10,synapse10,'psr');
csim('connect',recorder10,neuron10,'Vm');
csim('connect',recorder10,neuron10,'spikes');

csim('connect',recorder11,synapse11,'psr');
csim('connect',recorder11,neuron11,'Vm');
csim('connect',recorder11,neuron11,'spikes');

csim('connect',recorder12,synapse12,'psr');
csim('connect',recorder12,neuron12,'Vm');
csim('connect',recorder12,neuron12,'spikes');

csim('connect',recorder13,synapse13,'psr');
csim('connect',recorder13,neuron13,'Vm');
csim('connect',recorder13,neuron13,'spikes');

csim('connect',recorder14,synapse14,'psr');
csim('connect',recorder14,neuron14,'Vm');
csim('connect',recorder14,neuron14,'spikes');

csim('connect',recorder15,synapse15,'psr');
csim('connect',recorder15,neuron15,'Vm');
csim('connect',recorder15,neuron15,'spikes');

csim('connect',recorder16,synapse16,'psr');
csim('connect',recorder16,neuron16,'Vm');
csim('connect',recorder16,neuron16,'spikes');

S.spiking = 1;
S.dt = -1;
S.idx = inputNeuron;
% Time of the input spike.
S.data = [0.456];

% Length of time for which the simulation will run.
Tsim=1.5;

% Run the simulation
csim('simulate',Tsim,S);

% Get the data from the recorders.
t=csim('get',recorder,'traces');
t2=csim('get',recorder2,'traces');
t3=csim('get',recorder3,'traces');
t4=csim('get',recorder4,'traces');
t5=csim('get',recorder5,'traces');
t6=csim('get',recorder6,'traces');
t7=csim('get',recorder7,'traces');
t8=csim('get',recorder8,'traces');
t9=csim('get',recorder9,'traces');
t10=csim('get',recorder10,'traces');
t11=csim('get',recorder11,'traces');
t12=csim('get',recorder12,'traces');
t13=csim('get',recorder13,'traces');

```

```

t14=csim('get',recorder14,'traces');
t15=csim('get',recorder15,'traces');
t16=csim('get',recorder16,'traces');

% Draw the graphs for the first 4 Neurones and Spike Train in a new window.
figure(1); clf reset;
% Subplot of the input spike train.
subplot(5,1,1);
st=S.data;
line([st; st],[-0.045; -0.015]*ones(size(st)),'Color','k');
set(gca,'Xlim',[0 Tsim]);
title('input spike train');

% Subplot for the first neuron.
subplot(5,1,2);
st=t.channel(3).data;
line([st; st],[-0.045; -0.015]*ones(size(st)),'Color','k');
set(gca,'Xlim',[0 Tsim]);
xlabel('time [sec]');
title('Neuron1');
% Print out times of spikes at Neuron1
([st])

subplot(5,1,3);
st=t2.channel(3).data;
line([st; st],[-0.045; -0.015]*ones(size(st)),'Color','k');
set(gca,'Xlim',[0 Tsim]);
xlabel('time [sec]');
title('Neuron2');
([st])

subplot(5,1,4);
st=t3.channel(3).data;
line([st; st],[-0.045; -0.015]*ones(size(st)),'Color','k');
set(gca,'Xlim',[0 Tsim]);
xlabel('time [sec]');
title('Neuron3');
([st])

subplot(5,1,5);
st=t4.channel(3).data;
line([st; st],[-0.045; -0.015]*ones(size(st)),'Color','k');
set(gca,'Xlim',[0 Tsim]);
xlabel('time [sec]');
title('Neuron4');
([st])

% Draw the graphs for the second 4 Neurones and Spike Train in a new window.
figure(2); clf reset;
subplot(5,1,1);
st=S.data;
line([st; st],[-0.045; -0.015]*ones(size(st)),'Color','k');
set(gca,'Xlim',[0 Tsim]);
title('input spike train');

subplot(5,1,2);
st=t5.channel(3).data;
line([st; st],[-0.045; -0.015]*ones(size(st)),'Color','k');

```

```

set(gca,'Xlim',[0 Tsim]);
xlabel('time [sec]');
title('Neuron5');
([st])

subplot(5,1,3);
st=t6.channel(3).data;
line([st; st],[-0.045; -0.015]*ones(size(st)),'Color','k');
set(gca,'Xlim',[0 Tsim]);
xlabel('time [sec]');
title('Neuron6');
([st])

subplot(5,1,4);
st=t7.channel(3).data;
line([st; st],[-0.045; -0.015]*ones(size(st)),'Color','k');
set(gca,'Xlim',[0 Tsim]);
xlabel('time [sec]');
title('Neuron7');
([st])

subplot(5,1,5);
st=t8.channel(3).data;
line([st; st],[-0.045; -0.015]*ones(size(st)),'Color','k');
set(gca,'Xlim',[0 Tsim]);
xlabel('time [sec]');
title('Neuron8');
([st])

% Draw the graphs for the third 4 Neurones and Spike Train in a new window.
figure(3); clf reset;
subplot(5,1,1);
st=S.data;
line([st; st],[-0.045; -0.015]*ones(size(st)),'Color','k');
set(gca,'Xlim',[0 Tsim]);
title('input spike train');

subplot(5,1,2);
st=t9.channel(3).data;
line([st; st],[-0.045; -0.015]*ones(size(st)),'Color','k');
set(gca,'Xlim',[0 Tsim]);
xlabel('time [sec]');
title('Neuron9');
([st])

subplot(5,1,3);
st=t10.channel(3).data;
line([st; st],[-0.045; -0.015]*ones(size(st)),'Color','k');
set(gca,'Xlim',[0 Tsim]);
xlabel('time [sec]');
title('Neuron10');
([st])

subplot(5,1,4);
st=t11.channel(3).data;
line([st; st],[-0.045; -0.015]*ones(size(st)),'Color','k');
set(gca,'Xlim',[0 Tsim]);
xlabel('time [sec]');
title('Neuron11');

```

```

([st]
subplot(5,1,5);
st=t12.channel(3).data;
line([st; st],[-0.045; -0.015]*ones(size(st)),'Color','k');
set(gca,'Xlim',[0 Tsim]);
xlabel('time [sec]');
title('Neuron 12');
([st])

% Draw the graphs for the final 4 Neurones and Spike Train in a new window.
figure(4); clf reset;
subplot(5,1,1);
st=S.data;
line([st; st],[-0.045; -0.015]*ones(size(st)),'Color','k');
set(gca,'Xlim',[0 Tsim]);
title('input spike train');

subplot(5,1,2);
st=t13.channel(3).data;
line([st; st],[-0.045; -0.015]*ones(size(st)),'Color','k');
set(gca,'Xlim',[0 Tsim]);
xlabel('time [sec]');
title('Neuron 13');
([st])

subplot(5,1,3);
st=t14.channel(3).data;
line([st; st],[-0.045; -0.015]*ones(size(st)),'Color','k');
set(gca,'Xlim',[0 Tsim]);
xlabel('time [sec]');
title('Neuron 14');
([st])

subplot(5,1,4);

st=t15.channel(3).data;
line([st; st],[-0.045; -0.015]*ones(size(st)),'Color','k');
set(gca,'Xlim',[0 Tsim]);
xlabel('time [sec]');
title('Neuron 15');
([st])

subplot(5,1,5);
st=t16.channel(3).data;
line([st; st],[-0.045; -0.015]*ones(size(st)),'Color','k');
set(gca,'Xlim',[0 Tsim]);
xlabel('time [sec]');
title('Neuron 16');
([st])

drawnow;

```