Computational Modelling of Synaptic Plasticity in the Dentate Gyrus Granule Cell

Nicholas Hananeia

15 December 2014

A thesis submitted in fulfilment of the Master of Science in Computer Science

Abstract

After more than 30 years of study, the dynamics of synaptic plasticity in neurons still remain somewhat a mystery. By conducting a series of simulations on a simulated version of the rat dentate gyrus granule cell using the Izhikevich spiking neuron model, we compare and contrast several potential synaptic plasticity rules' applicability to the same experiment. Based on a 2001 experiment (Abraham et al., 2001), our simulations find that spike timing dependent plasticity (STDP), a more recent (Markram et al., 1997) theory of synaptic plasticity, is insufficient to replicate the heterosynaptic LTD shown in the experiment without including aspects of the significantly older Bienenstock-Cooper-Munro (BCM) (Bienenstock et al., 1982) theory. A combination of the history-independent STDP model and the history-dependent BCM model seems most likely to be an accurate candidate for reproducing the greatest variety of cell dynamics. We also find that in simpler nearest-neighbour STDP rules, the choice of pairing scheme is critical in achieving the greatest concordance with experiment.

Acknowledgements

I would like to thank all that gave me their wholehearted support and assistance in researching for and writing this thesis in the past year.

First and foremost, I thank my supervisor. Dr. Lubica Benuskova, for her advice, wisdom and insight, which granted me a deeper understanding of not only the subject matter in this project, but also a much deeper insight into neuroscience in general.

Secondly, I thank the Department of Computer Science at the University of Otago for their hospitality and assistance throughout this project, special mention to their facilitating my presentation of results at the NeuroEng 2014 workshop.

Finally, I give my heartfelt thanks to the friends and family that gave me moral support and guidance throughout this endeavour.

1. Introduction

In this thesis we conduct a series of simulations of synaptic plasticity in the mammalian dentate gyrus granule cell, which serve to examine the robustness and applicability of several different models of synaptic plasticity. This will be accomplished by qualitative comparison of simulation results and experimental results, along with exploration of parameter spaces.

Compared with basic cell mechanics, synaptic plasticity is a poorly understood phenomenon in the field of neuroscience, with several competing theories aiming to provide a mathematical framework to describe when and how much by the connection strengths between cells vary. In this study, we will simulate the same experiment with several different methods and examine what theories or syntheses of theories are best for description of a cell undergoing normal LTP induction protocols.

An understanding of synaptic plasticity is critical for our understanding of the brain as a whole - the brain has an amazing ability to dynamically change its connectivity, which has ramifications for the computational nature of any cell, or, indeed, the brain as a whole. Given how much biological inspiration for technology has occurred in the world of computer science, a proper understanding of synaptic plasticity could result in granting us the ability to build better artificial intelligences and learning machines, and perhaps implement more biologically realistic ways of training artificial neural networks.

We examine five different models of synaptic plasticity by running a series of simulations based on the same experiment while using the same neuron model. If a model is a good description of synaptic plasticity in the granule cell, we expect the results to be qualitatively similar to those of the real experiment. In addition to examining these four models, for one of them, we will examine further details of synaptic plasticity, namely the effect of choice of pairing of spikes, and additionally we conduct a preliminary examination into the robustness of the Benuskova & Abraham plasticity model when generalised to a more biologically realistic cell with more than two inputs.

1.1. Goals

In this project, we will start with an implementation of the Izhikevich neuron with the Benuskova & Abraham synaptic plasticity rule using the presynaptically centred pairing scheme. This will serve as a baseline for all following simulations.

The first goal will be to expand the Benuskova & Abraham rule to cover other STDP pairing schemes - the symmetric, reduced symmetric and nearest spike pairing schemes are those to be considered. Once these are implemented and the parameter optimisation complete, we will compare and contrast the results from these implementations to each other and to the initial presynaptically centred pairing scheme, focussing on finding which candidate is the best fit for the data in Abraham et al., 2001.

With this done, we will work on increasing the robustness of our Benuskova & Abraham simulation. Currently the simulation only has two inputs - increasing this to a larger number (20 to start, then investigating moving to a more biologically realistic number) would be a good avenue of investigation. Once this is done, we will investigate whether increasing the biological realism in this way improves or changes our simulation's results.

The second goal will be to broaden our investigations beyond the Benuskova & Abraham rule, to other STDP rules. There are a wide variety of STDP rules to choose from, and not all are applicable for or feasible to implement for our simulation. Selection of a variety of rules to consider will come first, followed by implementation of a wide selection of rules. Ideally, we will cover enough of a range to be able to observe the strengths and weaknesses of different models in reproducing the same experiment that we examined with our original implementation.

In these simulations we will conduct, our main hypotheses that we will test are that, first, the way in which spikes are paired in nearest-neighbour STDP has an effect on the outcome of plasticity, and second, in order to reproduce experimental data, some form of metaplasticity must be in the synaptic plasticity rule.

2. Background

In this section we provide an overview of the biophysical and computational background that underlies the experiments conducted during this study.

2.1 Biological Background

2.1.1. The Neuron

Neurons are the fundamental computational unit of the animal central nervous system, and differ very little in their function or mechanism across a great variety of species. Although there are several different types of neuron in the brain, differing in size, shape and connectivity, all of them share a fundamentally similar basic structure, having the same characteristic features.

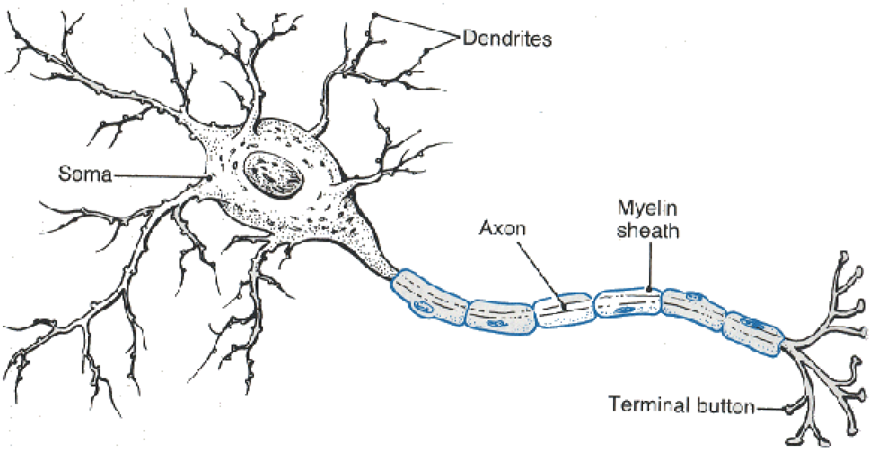

Fig.1: A neuron, showing main components. Source: Carlson, 1992

As can be seen in Fig.1, the neuron has three principal regions of interest: The axon, the dendrites and the soma. The soma is the body of the neuron, and contains the cell nucleus as well as the bulk of the cell's mass. This is the computational centre of the cell, and is where summation of inputs is carried out.

The dendrites are the cell's input region; these are where most of the connections from other cells are made. Dendrites are branched like a tree (indeed, the word "dendrite" comes from the Greek dendron meaning tree), and a typical neuron contains thousands of connections to other cells, making synapses with smaller features on the surface of the dendrite called dendritic spines. (Synapses need not always connect to dendritic spines; they can also connect to the dendrite's main surface, the soma, or the axon).

Likewise, the axon is the cell's output region. The axon carries action potentials away from the soma, to synapses with other cells at the end of the axon terminal. The axon is coated with a fatty sheath called myelin, which functions as an electrical insulator to counter any attenuation of the action potential. There are periodic breaks in the myelin sheath called nodes of Ranvier, which act as electrical repeaters, further boosting action potentials as they pass down the axon, acting as another counter to attenuation of the action potential. This is an important function - degradation of the myelin sheath causing attenuation of action potentials is the cause of the symptoms of multiple sclerosis.

The presence of the fatty myelin is responsible for the distinction between the regions of the brain called grey matter and white matter, simply named for their physical appearance. Grey matter is brain tissue consisting of mostly neuronal somas and dendrites, whereas white matter is brain tissue dominated by axons and glial cells, white in colour because of the heavy presence of myelin. The bulk of the brain's energy consumption is spent in grey matter regions. Glial cells, such as astrocytes, are cells that support the nervous infrastructure, doing tasks such as maintaining myelination, supporting the physical arrangement of the neurons, and providing them nutrition.

At the very end of the axon, the axon branches, forming synapses with many other cells. Each branch ends in a synapse, connecting this presynaptic cell and the postsynaptic cell on the other side of the synapse.

The surface of any cell is a phospholipid bilayer called the cell membrane. This consists of two stacked layers of phospholipid molecules in a tight lattice that separates the extra-cellular fluid from the interior cytoplasm. Each phospholipid molecule consists of a polar head with two hydrocarbon tails. In the cell membrane, the hydrocarbon tails of the two layers point inwards, leading to a surface on both sides that is a tightly packed array of the phospholipids' heads.

Embedded within the cell membrane are various proteins. They can be periphery proteins, which are only embedded within one of the two phospholipid layers, surface proteins that lie on the surface of the bilayer, or transmembrane proteins that are completely embedded into the membrane, spanning both layers. A particular class of transmembrane proteins, the transport proteins, which allow ions to pass through the otherwise impermeable membrane, are of particular interest in neuroscience.

Transport proteins that are of interest in neuroscience include gates, pumps and receptors. The sodium-potassium pumps are responsible for a large portion of the brain's total energy use, and serve to maintain a constant potential gap between the interior and exterior of the cell. This potential gap is in the region of -65 millivolts (Sterratt et al., 2011) and is created by the pumps selectively bringing K^+ ions into the cell while pushing Na^+ ions out of the cell. Although both ions are positive, the ions are pumped at differing rates - two K^+ ions for every three Na^+ ions (Bear et al., 2007, p.66) This difference is sufficient to maintain the resting potential.

When the potential of the cell rises above a threshold level which varies between cells, an action potential is generated (it is worth noting that some uncommon examples of cells such as thalamo-cortical cells (Sterratt et al., 2011) also generate action potentials from highly negative potentials). This process is facilitated by voltage-gated ion channels, also on the surface of the cell. These channels are said to be gated because they will not allow any ions through when they are closed. Once the potential reaches threshold, these channels open, allowing Na^+ ions from outside the cell to enter. This causes the potential of the cell to rise even further, increasing to peaks of up to +90mV (Sterratt et al., 2011). Because of the presence of voltage-gated ion channels on the cell membrane in the soma and axon (but not the dendrites - the soma and axon are termed actively conducting because of this, in contrast to the dendrites' passive conductance), the action potential will spread through the soma and down the axon. Once the potential is sufficiently high, the sodium channels will close and the potassium channels will open, causing the cell's potential to drop back to below the resting value, where it will remain for some time until resting potential is achieved again. This is the refractory period - another action potential will not be able to be generated until the refractory period has passed.

2.1.2. The Synapse

Synapses are structures that permit neurons to communicate with each other, and almost always exist at axon terminals (rare exceptions do exist, as in the case of synapses between the dendrites of two cells (Morest, 1971)). Synapses can connect from the presynaptic cell to a cell's dendrites, soma or axon (axodendritic, axosomatic and axoaxonic synapses respectively), but are usually associated with the dendrites of the postsynaptic cell. When an action potential arrives at a synapse, it causes an electrical response on the other side - a postsynaptic potential.

There are two categories of synapse: Chemical synapses and electrical synapses (otherwise known as gap junctions). In the mammalian nervous system, chemical synapses play a far greater role than their electrical counterparts. The two types of synapses have different roles in the nervous system owing to the advantages and disadvantages inherent in their structure.

Electrical synapses occur where the membranes of the presynaptic and postsynaptic cell are directly in contact, and are essentially a place where charge is able to freely flow between the cells. Because of this, they are much faster than chemical synapses, and are commonly seen in organisms that need exceptionally fast reaction times, such as in the escape reflexes of simple invertebrates (Edwards et al., 1999).

However, chemical synapses are much more prevalent than their electrical counterparts for two simple reasons - firstly, postsynaptic potential of any given synapse is the same regardless of the intensity of the action potential that caused the synapse to activate - chemical synapses are all or nothing. This makes communication between neurons more akin to a digital signal than an analog one, with meaning encoded in the spike train. However, at this point, whether the timing or rate of the spike train determines the meaning of whatever information is being communicated between neurons is a relative unknown.

Secondly, chemical synapses are capable of plasticity - the strength of the postsynaptic response of a given synapse can vary over time. This gives them the ability to exhibit a much greater range and complexity of behaviours than their electrical cousins. As this study focuses on the phenomenon of synaptic plasticity, careful attention must thus be paid to the structure and function of the chemical synapse.

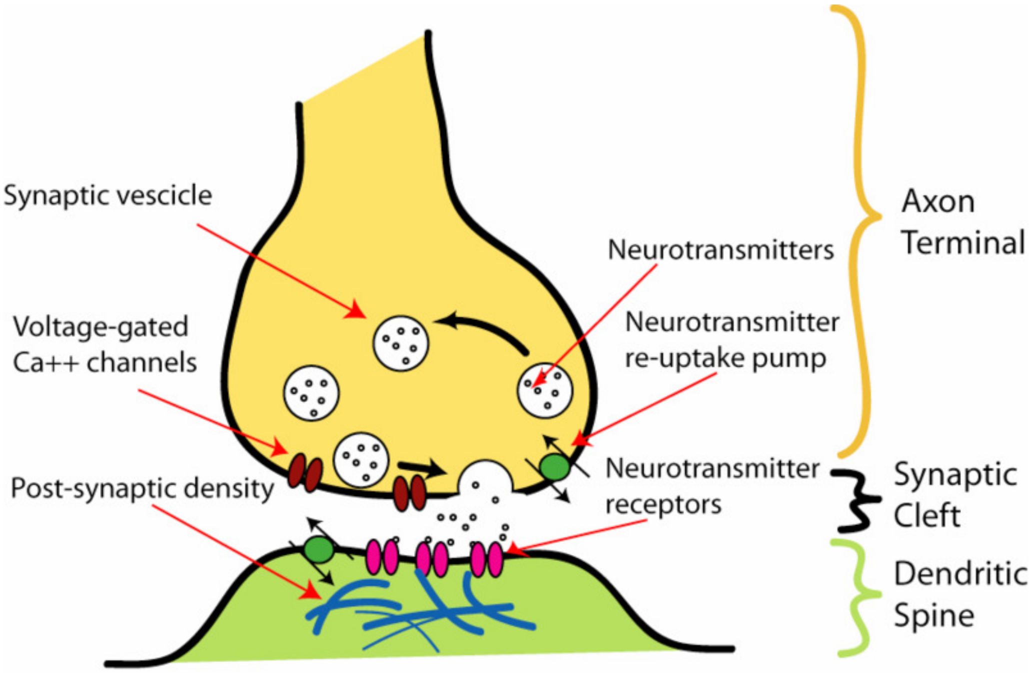

Fig. 2: A chemical synapse. Source: http://commons.wikimedia.org/wiki/File:SynapseIllustration2.png

As can be seen in Fig. 2, the chemical synapse is not actually a point of contact between the two cells, in contrast to the electrical synapse. The cells are separated by the synaptic cleft, a gap of around 20-40nm. Because of this, the presynaptic action potential cannot actually cross the synapse - this explains why the postsynaptic potential of the chemical synapse is independent of the action potential's magnitude.

When an action potential arrives at the axon terminal, it causes synaptic vesicles in the terminal to fuse with the cell wall in the synaptic cleft, releasing the neurotransmitters they contain into the gap. What neurotransmitter is contained within the vesicles depends on the presynaptic neuron in question - each synapse contains only one type of neurotransmitter, allowing a synapse to exhibit, excitatory, inhibitory, modulatory, or a combination of excitation and modulation or inhibition and modulation (Bear et al., 2011). The neurotransmitters then cross the synaptic cleft and interact with receptors on the postsynaptic cell membrane.

Each neurotransmitter can interact with many types of receptor, usually accomplishing one of three effects - excitation, inhibition and modulation. Glutamate, gamma-aminobutyric acid (GABA) and 5-hydroxytryptamine (serotonin) are examples of these, respectively. The receptors are microscopic structures similar to the aforementioned ion channels, and accomplish a similar function. In a receptor-gated ion channel, the neurotransmitter interacting with the receptor will cause an ion channel to open, allowing Na^+ ions to enter, causing a small excitatory postsynaptic potential. The mechanism is similar for an inhibitory receptor.

In a modulatory receptor, a variety of actions are possible. A prototypical modulatory receptor is a G-protein coupled receptor - interaction with a neurotransmitter will release a signalling chemical within the cell that can accomplish many effects, generally longer-term in nature than a short-term postsynaptic potential. This sort of receptor is important for changing neurotransmitter release rates in the postsynaptic cell, and on a whole brain level, they play in important part in an individual's psychological state. This is exemplified by successful treatment of mood disorders via artificially increasing serotonin levels in the brain.

Although each receptor is named for the neurotransmitter that interacts with it, there is a level of classification below this - there is more than one type of receptor that specifically interacts with a given neurotransmitter. A good example of this is acetylcholine (ACh) receptors. Acetylcholine is the neurotransmitter that is responsible for muscle movements, and was indeed the first discovered neurotransmitter, by Otto Loewi in 1921, winning him the 1936 Nobel Prize in Physiology or Medicine. There are two types of acetylcholine receptor - the nicotinic and muscarinic receptors. The nicotinic ACh receptor is so named because it is activated by nicotine. In contrast, the muscarinic ACh receptor is named because it is activated by muscarine, a toxin that is found in the hallucinogenic mushroom *Amanita muscaria.*

Fig.3: Amanita muscaria*, the fly agaric mushroom, where muscarine was originally isolated. Source: Nicholas Hananeia*

These two receptor types are both excitatory, although accomplish excitation via different means. The nicotinic ACh receptor is a neurotransmitter-gated ion channel, whereas the muscarinic receptor is a G-protein coupled receptor that causes other ion channels to open via a mechanism known as a second messenger cascade - in this way the muscarinic ACh receptor is an intermediary to opening other channels. These receptors are also antagonised by separate toxins - atropine (the active toxin in deadly nightshade, *Atropa belladonna*) in the case of the muscarinic receptor and curare in the case of the nicotinic receptor (Bear et al., 2007, p. 139). The action of these toxins on these receptors is what causes their main toxic effects. In addition, there are multiple types of nicotinic and muscarinic ACh receptors, differing in their morphology. In this way, designations based on interactions with chemicals usually denote a specific subclass of receptor.

After the neurotransmitter has interacted with the receptor and has been subsequently released, it will either diffuse away from the synaptic cleft due to random particle motions or be reuptaken by the presynaptic terminal, where it can be incorporated into a vesicle and used for another release. The rate of reuptake is an important mechanism by which the synapse's release rate can be adjusted.

2.1.3. Synaptic Plasticity

Chemical synapses are capable of synaptic plasticity, a function that is largely absent from electrical synapses. In short, synaptic plasticity is the ability of a synapse to change in efficacy - that is, a more efficient synapse will cause a larger postsynaptic potential. Because excitatory chemical synapses are the primary mode of communication between brain cells, synaptic plasticity is an extremely powerful ability for the synapses to have, and, indeed, alteration of the efficacy of synapses is the brain's primary way of storing information.

These changes need not be positive - synaptic plasticity works in both directions. When the changes of the efficacy (or weight) of the synapse are long-lasting, they are known as long-term potentiation or long-term depression (LTP or LTD). Understanding of the mechanisms underlying LTP and LTD has been a major focus area of neuroscience in the last few decades, with several rival theories still under consideration.

Physically, synaptic plasticity is accomplished via a complicated sequence of chemical reactions inside the postsynaptic cell. Here we will focus on the mechanisms of synaptic plasticity in glutamate synapses, which are responsible for excitatory activity in the hippocampus (Benuskova & Kasabov, 2007).

The two subclasses of glutamate receptor that are of interest for synaptic plasticity are the AMPA and NMDA receptors, simply named for the chemicals that selectively activate them. They are both transmitter-gated ion channels, both producing a positive postsynaptic potential when opened. However, their behaviour differs in a few crucial ways.

When glutamate activates the AMPA receptor, it opens as normal, letting Na^+ ions cross the membrane and trigger a small excitatory postsynaptic potential as a result. The NMDA receptor, in contrast, has a Mg^{2+} ion bound to the interior side of its channel, blocking any ions from passing through when it is activated by glutamate. Only when the Mg^{2+} ion is removed from the channel by a sufficiently high membrane potential, a condition brought about by the action of the AMPA receptors, does the NMDA receptor open. This allows the NMDA receptor to function as a co-incidence detector, opening only in the presence of both presynaptic and postsynaptic activity.

In addition to allowing Na^+ ions through the membrane, the NMDA receptor's ion channel also allows the passage of Ca^{2+} ions. This means that the NMDA receptors trigger a much larger postsynaptic potential than the AMPA receptors, one that lasts for significantly longer. This gives the glutamate synapse a two-staged response, increasing significantly after the NMDA receptors' activation threshold is passed.

Once they cross the membrane through the channel gated by the NMDA receptor, the Ca^{2+} ions bind with a messenger protein, calmodulin, the presence of which triggers a cascade of chemical reactions. Firstly, the presence of the Ca^{2+}/calmodulin complex activates Ca^{2+}/calmodulin -dependent protein kinase (CaMKII), which in turn phosphorylates dormant AMPA receptors, activating them. The presence of many dormant AMPA receptors allows the strength of the synapse to be quickly adjusted by this method for low-magnitude, quick, easily reversible LTP. Ca^{2+}/calmodulin can also trigger a release of a retrograde messenger to the presynaptic terminal, increasing the amount of neurotransmitter released.

LTP that requires more than simply activating dormant receptors is also accomplished by the presence of Ca^{2+}/calmodulin. If a sufficient amount of this is present, caused by a sustained train of activation of the synapse, the Ca^{2+}/calmodulin activates an adenylyl cyclase, causing a series of biochemical reactions in the cell nucleus leading to a gene expression. In response, the nucleus will manufacture and transport more AMPA and NMDA receptors to the synapse in question, thus increasing the available pool of usable receptors and a more permanent increase in the synapse's strength. The nucleus can also release chemicals that cause the construction of additional synapses if needed.

These two phases of activation are called early and late phase LTP, and correspond to a more short-term mechanism and a longer one more suited for permanent storage of information. LTD is accomplished by a reversal of this process - dephosphorylation of AMPA receptors to render them dormant for early-phase LTD, or re-absorption and recycling by the nucleus of redundant receptors for late-phase LTD.

A third class of glutamate receptor, the metabotropic receptor, serves to modulate the rate at which these processes happen, but the AMPA and NMDA receptors remain primarily responsible for synaptic plasticity in glutamate synapses.

2.1.4. The Hippocampus

The hippocampus is a structure located deep within the brain, in the temporal lobe under the cortex. It is responsible for memory consolidation - that is, the conversion of short-term to long term memories, and, as such, is of significant clinical and theoretical interest. The hippocampus' distinctive shape led to its name - taken directly from the Greek word for "seahorse." It consists of two interlocking features - the dentate gyrus and the Ammon's horn (referred to as CA, for cornu ammonis, and its regions named CA1, CA2, CA3 and CA4.)

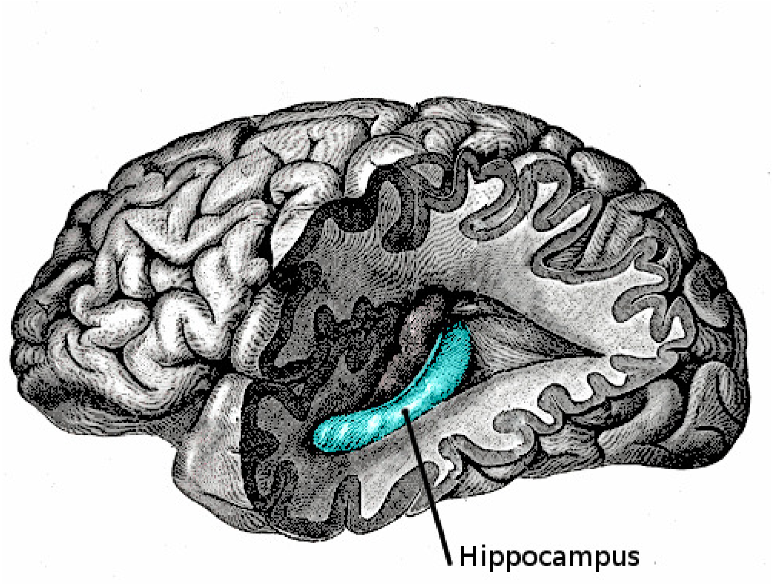

Fig.4: Location of hippocampus in human brain. Source: Gray's Anatomy

As long-term memories are stored distributed across the cortex, the hippocampus is very strongly connected with it.

Hippocampal cells are also strongly connected to spatial memory, with the CA1 pyramidal cell firing only sparsely and then only so when the organism is in a certain spatial orientation (Ahmed & Mehta, 2009) and, as such, CA1 pyramidal cells are often called place cells. Experiments in rats have shown that bilateral hippocampal destruction gives a complete inability for the animal to solve even basic spatial memory related tasks (Bear et al, 2007).

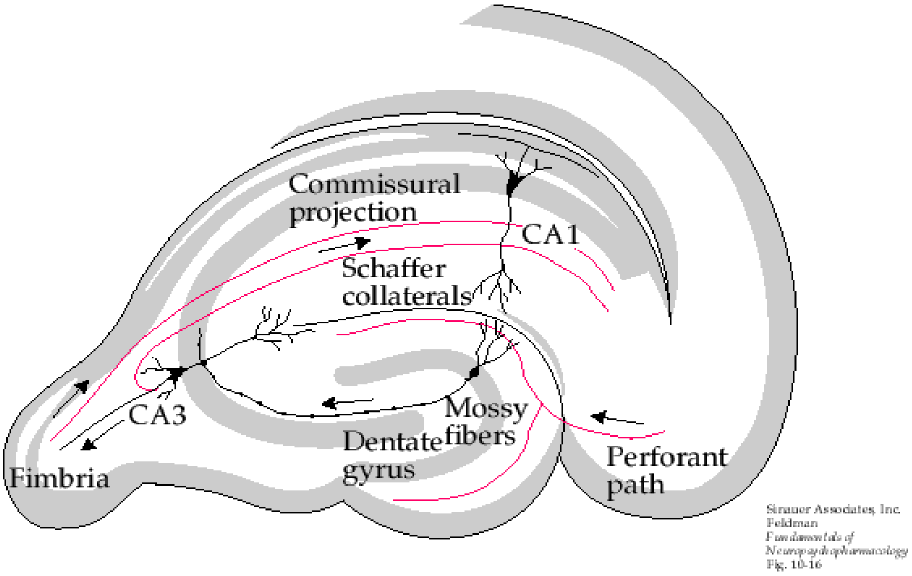

Fig.5 Diagram of hippocampus showing major anatomical sub-areas and pathways.

2.1.5. The Dentate Granule Cell

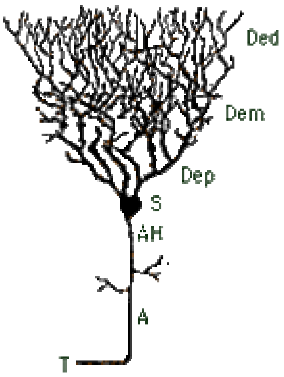

Fig.6: A dentate granule cell. T: terminal, A: axon, AH: axon hillock, S: soma, Dep: proximal dendrites, Dem: medial dendrites, Ded: distal dendrites.

Source: http://neurolex.org/wiki/Category:Dentate_gyrus_granule_cell#tab=Advanced

The cell on which this study will be taken is the dentate granule cell, a cell in the dentate gyrus which occupies the first input layer in the hippocampus (Förster et al., 2006). This cell is called a granule cell because it is a small, circular cell (indeed, granule cells are amongst the smallest kinds of neuron). The cell has a large dendritic tree (relative to the size of the soma) and receives two main excitatory inputs in addition to recurrent inputs, inputs from other granule cells, and inhibitory cells (Ahmed & Mehta, 2009). The granule cells themselves sit side by side in a layer.

The two primary excitatory inputs are the medial and lateral perforant paths (MPP and LPP), which connect the granule cells to the entorhinal cortex, a structure located outside the hippocampus. The dentate granule cell's axons connect to the pyramidal cells in CA3. Each of these two perforant paths contains thousands of individual axons, each of which makes a connection with many of the dentate granule cells.

The MPP and LPP synapse upon adjacent but separated regions of the granule cell's dendritic tree. The MPP synapses upon the medial region of the tree, with the LPP synapsing upon the distal regions. However, the small differences in transmission between the two are generally thought not to result from the differences in synapse site, but from the subtly different transmission properties of the paths (Abraham & McNaughton, 1984).

2.2. Physical Experiment

In this report, we will conduct repeated simulations of the same experiment using different plasticity models. As such, a thorough understanding of what was involved in this experiment is necessary. The experiment (Abraham et al., 2001), was conducted on live rats to take measurements of the results of various LTP induction protocols.

This paper detailed various experiments testing different protocols - we will be focusing on the basic one. For all experiments, rats had a measuring/stimulating electrode directly implanted into their brains, which is connected to an external device via a wire. After the rats recovered from the surgery, the experiments were performed, the rats being allowed to freely roam within the test area for the duration.

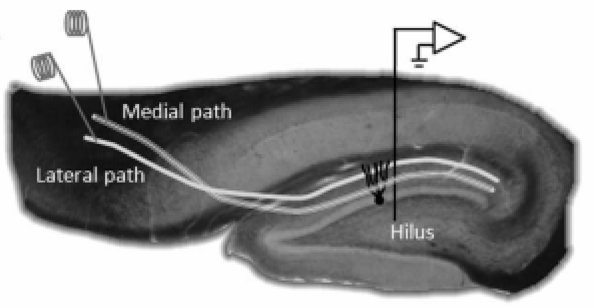

Fig. 7: Position of the stimulating (far left, end of medial and lateral paths) and recording (centre right) electrodes in the rat hippocampus. Source: Bowden et al., 2012

In the experiment, a brief series of pulses of high frequency stimulation (HFS) was applied directly to the medial perforant path (MPP) by the electrode. Following this, after a few hours, an identical stimulus was applied to the lateral perforant path (LPP). In this experiment, the HFS consisted of a series of ten groups of five 10-pulse trains at 400Hz, but other protocols have been used in other experiments (Bowden et al., 2012).

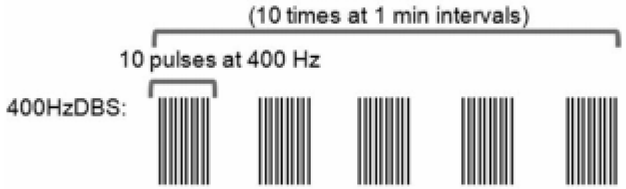

Fig. 8: Schematic of high-frequency stimulation used in Abraham et al., 2001.

The weights, and the changes to them, were inferred by a series of test pulses through the pathways with a frequency of one per minute. These pulses are of low enough intensity that their impact on the weights themselves would be minimal, but enough so that the resulting postsynaptic potential could be measured and used to infer any change in the weight.

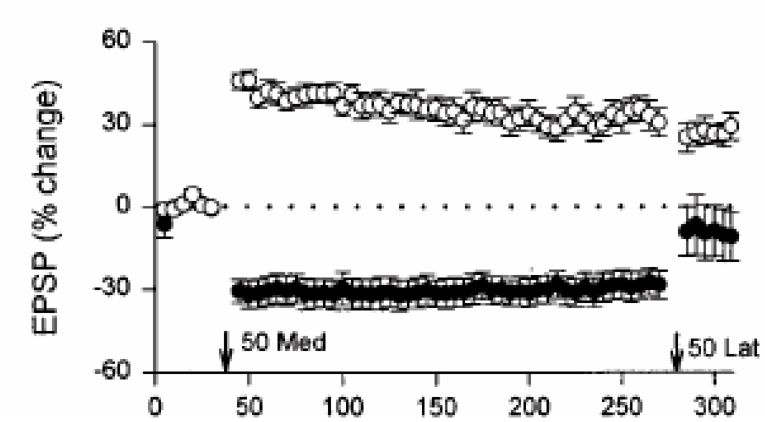

Fig. 9: Weight changes elicited by HFS in Abraham et al. 50 Med indicates 50 pulses of HFS applied to MPP; 50 Lat indicates 50 pulses of HFS applied to LPP. Source: Abraham et al., 2001

The results, seen in Fig. 9, show several effects of note. Firstly, shortly after upon the application of the HFS to the MPP, the MPP's weight increased, the change persisting for hours - this is the intended result of the LTP induction protocol. However, at the same time, the LPP's weight decreased by a lesser but still significant amount, this change also persisting. Note that we cannot know the values of the weights during HFS - the test pulses are not applied during HFS and as such these periods are not shown in the graph.

This effect is called heterosynaptic plasticity - heterosynaptic LTD in this case. Heterosynaptic plasticity occurs when a synapse adjacent to a stimulated synapse that is not itself being stimulated is also subject to LTP or LTD. Heterosynaptic LTP can also occur too, however it happens under different circumstances (Wöhrl et al., 2007).

The second round of HFS at 270 minutes partially reversed the changes, heterosynaptic LTD also happening here.

2.3. Neuron models

In computational neuroscience, there is no one unified approach to simulating the behaviours of a biological neuron. Because the neuron is a complicated biochemical and biophysical system, there are of course compromises that need to be made for ease of computation - much in the same way as a physicist simplifies a complicated object to a point mass. Neuron models are roughly categorised into two groups - the biophysical models and the phenomenological models. Biophysical models seek to simulate the physical details of the cell - such as ion channels, action potentials and postsynaptic potential, whereas phenomenological models seek only to reproduce the cell's overall input-output characteristics via a simplification of the cell's internal workings.

2.3.1. Hodgkin-Huxley

The Hodgkin-Huxley model is one of the earliest and still one of the most used neuron models. This is a set of differential equations which simulate the ion channels and pumps of the neuronal membrane with a set of conductance variables. As such, it is considered a biophysical model. The original form (Hodgkin & Huxley, 1952) is:

$$I = C\frac{dV_m}{dt} + g_K n^4 (V_m - V_K) + g_{Na} m^3 h (V_m - V_{Na}) + g_l (V_m - V_l) \tag{1}$$

$$\frac{dn}{dt} = \alpha_n (1-n) - \beta_n n \tag{2}$$

$$\frac{dm}{dt} = \alpha_m (1-m) - \beta_m m \tag{3}$$

$$\frac{dh}{dt} = \alpha_h (1-h) - \beta_h h \tag{4}$$

$$\alpha_n (V_m) = \frac{0.01(V_m - 10)}{\exp\left(\frac{V_m - 10}{10}\right) - 1} \tag{5}$$

$$\alpha_m (V_m) = \frac{0.1(V_m - 25)}{\exp\left(\frac{V_m - 25}{10}\right) - 1} \tag{6}$$

$$\alpha_h(V_m) = 0.07\exp\left(\frac{V_m}{20}\right) \tag{7}$$

$$\beta_n(V_m) = 0.125\exp\left(\frac{V_m}{80}\right) \tag{8}$$

$$\beta_m(V_m) = 4\exp\left(\frac{V_m}{18}\right) \tag{9}$$

$$\beta_h(V_m) = \frac{1}{\exp\left(\frac{V_m - 30}{10}\right) + 1} \tag{10}$$

where g_K, g_{Na} and g_l are the potassium, sodium and leak conductances, C is the membrane capacitance, V_m is the membrane voltage, and V_K, V_{Na} and V_l are the potassium, sodium and leak voltages, n, m and h are the so-called gating variables, and the α and β rate variables. More recent implementations of this model have a different, more general formulation of the α and β variables (Nelson, 2005). With these values for the constants, the voltages are all in millivolts.

This set of differential equations, although providing a rich picture of the biological neuron, has no analytical solution, and as such, is very computationally costly to implement, rising to extreme computation times for even moderately complicated simulations.

This model was and is of such importance to neuroscience as a whole that Hodgkin and Huxley were jointly awarded the 1963 Nobel Prize in Physiology or Medicine with Sir John Eccles for their discoveries concerning the ionic mechanisms involved in excitation and inhibition in the peripheral and central portions of the nerve cell membrane.

2.3.2. Compartmental models

Although the Hodgkin-Huxley model well describes the behaviour of the cell's soma and axon, it is not a complete description for the dendrites. If we want to accurately model the complete behaviour of a cell with a large dendritic tree, a compartmental model can be used. These models split the cell up into "compartments" with a subtly different type of neuron model applicable in each compartment. A typical compartmental model may use separate compartments for the soma, the distal dendrites, the medial dendrites, and the proximal dendrites.

This is necessary because the conductance behaviour of the dendrites is different to that of the soma or axon - while the action potential is constantly reinforced by voltage gated ion channels in the "active" membrane of the soma and axon, no such channels exist in the dendrites. Instead the dendritic membrane has a "passive" behaviour where the action potential decays as it travels up the dendritic tree. To accomplish this, terms for passive conductance are added to the Hodgkin-Huxley model.

2.3.3. Spiking neuron models

Spiking neuron models are, in contrast to the deep biophysical richness of the Hodgkin-Huxley model or a compartmental model, very simple. These models are often described by a single differential equation, and, as such, are often described as integrate-and-fire models, but still see widespread use in computational neuroscience because of their simplicity and associated low computational complexity. A general integrate-and-fire model has the following form:

$$I(t) = C\frac{dV}{dt} + I_{ext} \tag{11}$$

Here, the neuron is treated simply as a capacitor, and will fire with a delta function spike whenever the voltage hits a certain threshold from below. While an extremely simple representation, this is sufficient for some uses. More complicated integrate-and-fire models add extra terms such as leak currents, refractory periods, and exponential spike generation.

2.3.4. Izhikevich model

This model is a phenomenological model which attempts to achieve the same biological plausibility of the Hodgkin-Huxley model, while remaining computationally inexpensive (Izhikevich, 2003). This is necessary because as computational neuroscience simulations become more and more complicated, both in scale (large networks of neurons) and in complexity of experimental protocol, the

Hodgkin-Huxley model becomes less and less ideal due to the computational demands.

The Izhikevich model aims to have a biophysically rich model while also being efficient enough to be used on a desktop computer. On an identical simulation, an Izhikevich implementation may complete hundreds of times faster than a Hodgkin-Huxley implementation, making its applicability to large-scale or long-duration simulations obvious.

The model is based on the full Hodgkin-Huxley model, and uses bifurcation methods (Izhikevich 2010) to reduce the complex system of differential equations down to a mere two-dimensional system, given by the equations:

$$\frac{dv}{dt} = 0.04v^2 + 5v + 140 - u + I \tag{12}$$

$$\frac{du}{dt} = a(bv - u) \tag{13}$$

with the conditions

$$\text{If } v \geq 55\text{mV then } v \leftarrow c \text{ and } u \leftarrow u + d \tag{14}$$

Here, *v* is the membrane voltage and *u* is a membrane recovery variable, accounting for activation of K^+ channels and inactivation of Na^+ channels. Once the spike reaches peak, the membrane voltage and the recovery variable are reset to a pre-spike state.

The constants *a*, *b*, *c*, *d* are parameters that describe the nature of the cell. The parameter *a* is a rate constant that describes the recovery time of variable *u*. A high *a* will lead to a faster recovery. The parameter *b* describes the sensitivity of the recovery variable to fluctuations in potential that fail to trigger a spike. The parameter *c* is the potential at which the cell resets to after a spike, the parameter *d* describes the after-spike reset of the membrane recovery variable.

Although the variables and parameters in the model are all dimensionless, time is given in milliseconds, and *v* and *c* are given in millivolts. This is because of the constants in equation 12. These were obtained by fitting the spike initiation dynamics of a cortical neuron such that *v* was in mV and *t* in ms.

Although the fit was for a cortical cell, Izhikevich notes (Izhikevich, 2003) that other choices would be feasible. Because of the large parameter space in *a*, *b*, *c*, and *d*, many types of cell dynamics can be accurately reproduced, in spite of the model's fit being for a specific type of cell. By adjusting these parameters, the type of cell that the model is reproducing can be altered.

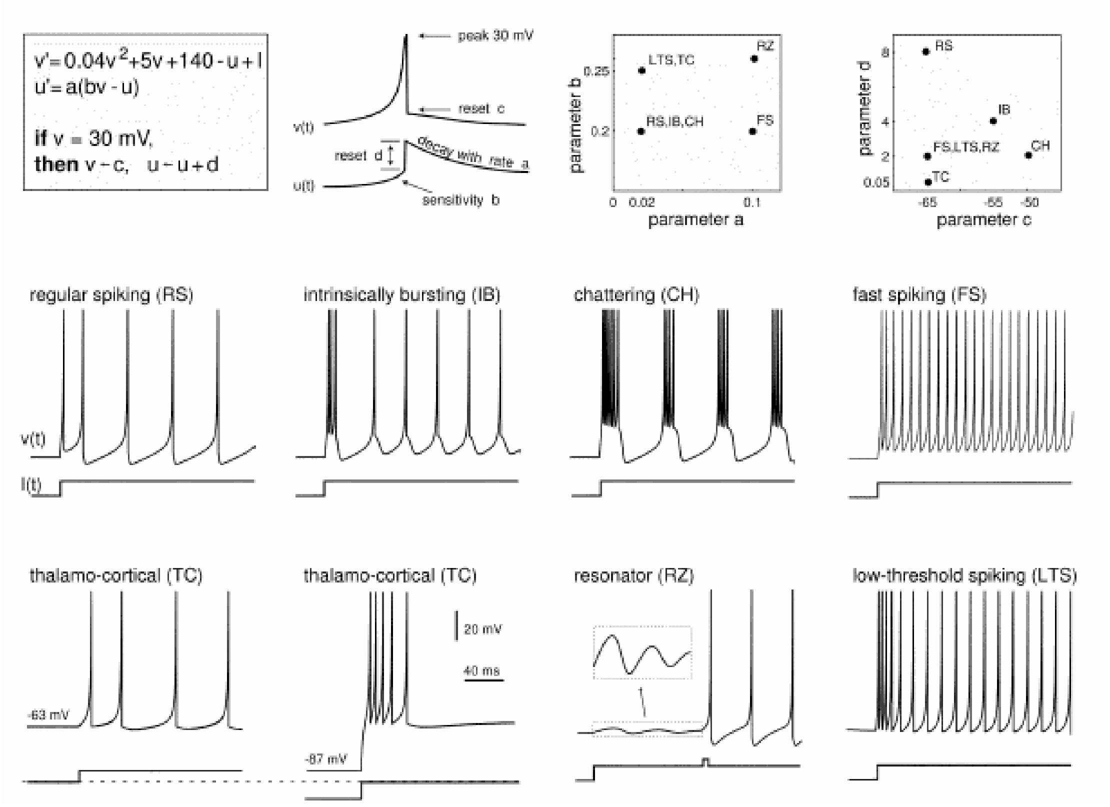

Fig. 10: Different cell behaviours yielded by alteration of a, b, c, d. Source: Izhikevich, 2003.

As can be seen in Fig. 10, the Izhikevich model is capable of reproducing a wide variety of cell dynamics. The cells' behaviour is simply dependent on the configuration of *a*, *b*, *c*, and *d*, as shown in the two plots at the top-right of Fig. 10. The model can reproduce both excitatory and inhibitory cells' behaviour, and of note is the model's ability to simulate the thalamo-cortical cell, which spikes on both hyperpolarisation and depolarisation. The fact that these cells are able to be accurately simulated with the model is evidence that the model's initial fit being for a cortical cell (shown here as the prototypical cortical cell, a regularly spiking or RS cell) is irrelevant in its overall robustness.

If, as claimed (Izhikevich, 2003), the model is indeed as biologically realistic as the Hodgkin-Huxley model, because its computational complexity is on par with a simple integrate-and-fire model, it is by far an excellent choice for a great myriad of computational neuroscience projects.

2.4. Synaptic plasticity models

Although synaptic plasticity is an extremely important brain function, it is still quite poorly understood, in spite of years of research and many different approaches in modelling (Mayr et al., 2010). There is currently no consensus on exactly how this phenomenon works, and there are multiple viable theories on the matter.

2.4.1. Hebb rule

One of the earliest and still most influential models of synaptic plasticity is the Hebb rule. Stated simply, "neurons that fire together wire together", or, when the a presynaptic cell fires and is (partially) responsible for a postsynaptic cell's firing, the synaptic weight increases. This is also called associative learning - a strong temporal association between the presynaptic and postsynaptic cell's firing causes the connection to become stronger (Hebb, 1949). This is described in a network of neurons by the simple equation

$$\frac{dw_{ij}}{dt} = x_{ij} \tag{15}$$

where w_{ij} is the weight from neuron i to neuron j, and x_{ij} is the input from neuron i to neuron j. However, this theory has its limitations - there is no mechanism for weights to decrease in a Hebbian system, leading to problems in any implementation. Implementations of this rule normally use some form of decay term or re-normalisation of weights to stop any weight from increasing to unreasonable levels. Also, many experimental results in biological neurons, such as the observance of depression of neuronal weights, run against the Hebb postulate - this led to the Bienenstock - Cooper - Munro (BCM) theory. However, it retained its influence later, as many aspects of the more recent theory of spike timing dependent plasticity (STDP) are in fact Hebbian.

2.4.2. BCM model

BCM, named after its theorists, Elie Bienenstock, Leon Cooper, and Paul Munro, is one of the earliest theories of synaptic plasticity (Bienenstock et al., 1982), and is still a commonly used framework in recent years (Cooper & Bear, 2012). BCM proposes a threshold level of postsynaptic activity, θ_M, below which LTD will occur, and above which LTP will occur. This modification threshold changes over time based on the overall activity of the neuron. This phenomenon can be referred to as *metaplasticity* (Abraham, 2008).

Under BCM theory, if the cell has a high level of activity, the threshold will increase, making further LTP more and more difficult, and vice-versa for a low activity cell. This manifests as suppressing plasticity in times of high activity, and facilitating it in times of low activity.

This effect can be seen in an experiment (Kirkwood, 1996), in which the visual cortices of kittens raised in a dark environment were compared to those of kittens raised in a normal environment. The dark-reared kittens showed more potential for plasticity in their visual cortex than those raised in a normal environment, providing credence to the BCM theory - under BCM, the visual cortex neurons of the dark-reared kittens would have a significantly lower modification threshold.

BCM theory can be expressed as the following equations, where y is the postsynaptic activity, w_i is the weight of the ith synapse to the cell, x_i is the presynaptic activity at the ith synapse, and θ_0 is a scaling constant:

$$y = \sum_i w_i x_i \tag{16}$$

$$\frac{dw_i}{dt} = y(y - \theta_M)x_i - \varepsilon w_i \tag{17}$$

$$\theta_M = E^P[(y\theta_0)] \tag{18}$$

This results in an approximately parabolic curve that intersects with the x-axis at θ_M before ceasing resemblance to a parabola as x increases further past θ_M, where the x-axis is the amount of postsynaptic activity and the y-axis is the weight change, as shown in Fig. 11:

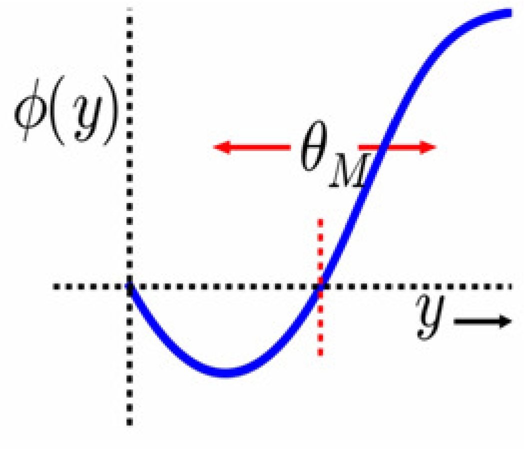

Fig. 11: BCM response curve, where y is postsynaptic activity, $\phi(y)$ the magnitude of weight change, and θ_M the BCM modification threshold, which shifts in the direction of either arrow in response to presynaptic activity. Source: http://www.scholarpedia.org/article/BCM/

Normally, in an implementation of BCM, provided appropriate choices of scaling constant θ_0 and window length (the period over which the previous activity of the cell is set to influence the current modification threshold) are made, there is no need for any hard-coded limits on weights as may be necessary in other synaptic plasticity models. This is because the dynamically changing BCM threshold acts as a soft cap - as the weight increases, the resulting increased activity of the neuron would cause a corresponding increase in the threshold. With a higher threshold, any further increases to the weight would become more and more difficult.

2.4.3. STDP model

STDP, or spike timing dependent plasticity, is a much more recent model of synaptic plasticity (Markram et al., 1997) that proposes a completely different framework than that of BCM. However, the theoretical foundation of STDP is older, based on simple Hebbian associative learning, where the connection between two neurons that simultaneously fire is made stronger as a result (Taylor, 1973). Under STDP, it is not the overall rate of activity in the postsynaptic neuron that dictates whether the weight will change, but the relative timing of spikes in the presynaptic and postsynaptic cells. If a presynaptic spike occurs before a postsynaptic spike, LTP of the synapse will occur, and LTD if the postsynaptic spike occurs before the presynaptic spike. If two spikes are simultaneous, the effect is ignored since this is equivalent to a 50% chance of LTP or LTD, which is what we would expect in a biological cell if the underlying theory holds true. This is expressed with the equations:

$$\Delta W_+ = A_+ e^{-t/\tau_+} \text{ if } t > 0 \tag{19}$$

$$\Delta W_- = A_- e^{t/\tau_-} \text{ if } t < 0 \tag{20}$$

where A_+ and A_- are constants (A_- negative) depending on the nature of the neuron, and τ_+ and τ_- are decay constants, and t is the time between the presynaptic and postsynaptic spikes. When plotted as a piecewise function, shows a pair of exponential curves centred on the origin - the positive one for LTP and the negative one for LTD.

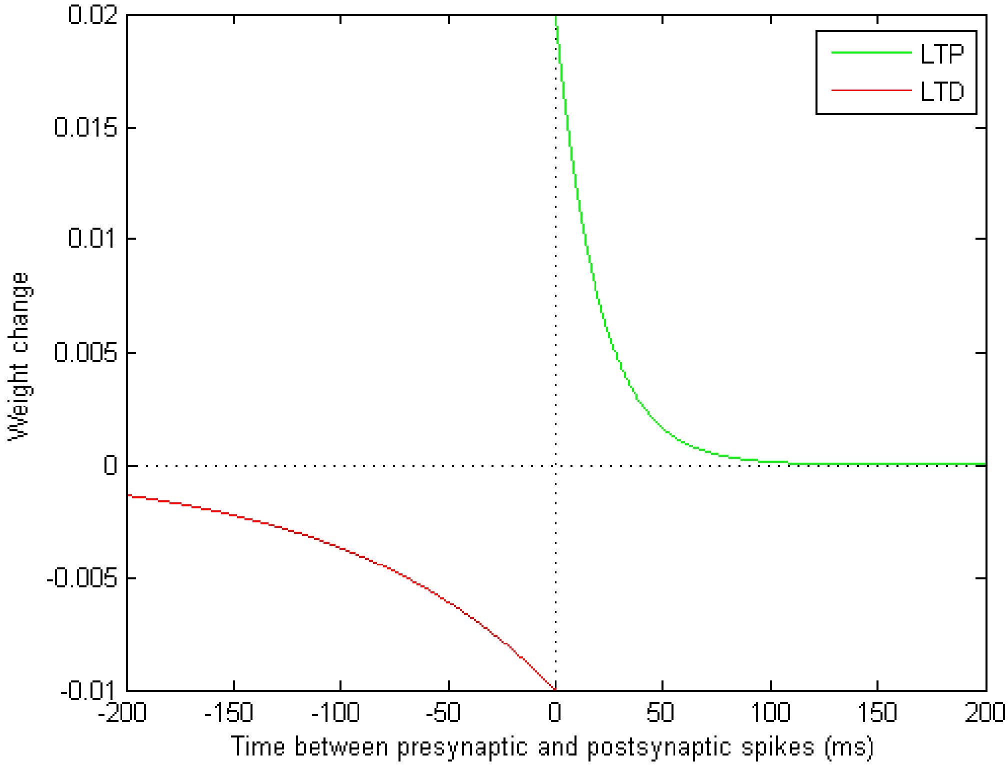

Fig. 12: STDP timing curves for $A_+ = 0.02$, $A_- = -0.01$, $\tau_+ = 20ms$, $\tau_- = 100ms$

2.4.4. Pairing schemes in STDP

While STDP is a powerful framework for describing synaptic plasticity, the theory is formulated in terms of relative spike timings. Since a postsynaptic cell may receive many input spikes and spike itself many times, the method of pairing two or more spikes together is unknown. There are many ways that this can be done, and whether any one method is better than another is still unknown.

The naive way to accomplish STDP would be the all-to-all scheme, wherein every presynaptic spike is paired with every postsynaptic spike, and the contributions to the weights of every single pairing summed. Although this would result in every possible contribution being counted, it has obvious drawbacks that limit its usefulness. In quickly-firing cells with large amounts of inputs, the computational load of having to sum every contribution from the entire history of the simulation would generate an enormous computational load.

Secondly, and perhaps more importantly, considering every possible contribution like this simply isn't biologically realistic. Although long-term synaptic modification effects do exist, STDP is strictly a short-duration effect. As the magnitudes of the modification curves of STDP quickly approach zero as the duration between spikes increases, all-to-all necessitates considering spikes with negligible impact, even if we do institute a cut off time after which spikes are no longer counted.

At the other extreme, we can consider the nearest spike scheme, where only the nearest spike to the currently considered spike is considered when finding the STDP contribution. Although simple, it is possible that this may be sufficient to account for the changes caused by STDP in some cases.

Three alternative schemes (Morrison et al., 2008) are proposed, which may provide a less computationally intensive and more biologically realistic pairing scheme. These are the symmetric, presynaptic centred, and the reduced symmetric pairing schemes (see Fig. 13).

The symmetric scheme pairs each postsynaptic spike with the immediately preceding presynaptic spike, regardless of how long ago it occurred, and similarly each presynaptic spike is paired with the most recent postsynaptic spike. This is the only scheme mentioned here which guarantees that every spike will contribute to weight change.

The reduced symmetric scheme is identical, except it only pairs immediately neighbouring spikes - if there is an additional postsynaptic spike between a given postsynaptic spike and its potential presynaptic partner, this pairing will not contribute to any weight change. This guarantees that each spike contributes to only to one pairing.

The presynaptic centred scheme simply pairs each presynaptic spike with both the preceding and the succeeding postsynaptic spikes.

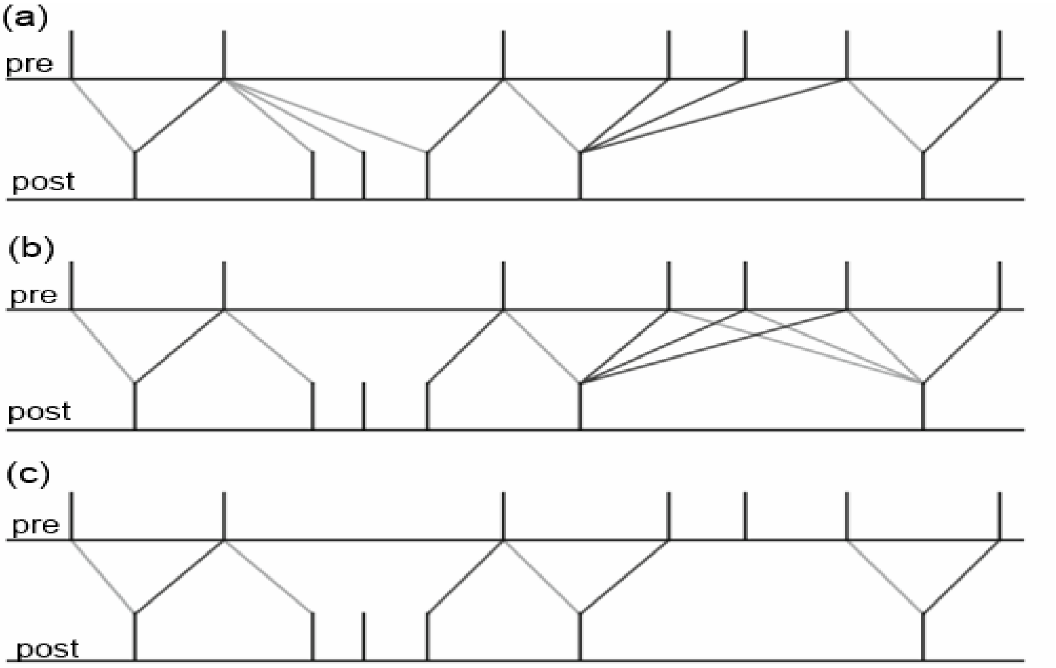

Fig. 13: Illustration of different nearest neighbour spike interactions: (a): Symmetric scheme; (b): Presynaptic centred scheme; (c): Reduced symmetric scheme. Adapted from Morrison et al. (2008).

2.4.5. Benuskova & Abraham rule

The Benuskova & Abraham implementation of STDP relies on earlier work (Izhikevich & Desai, 2003) in which for nearest-neighbour pairing schemes, a form of BCM can be shown to follow from STDP, given some assumptions. Under this rule, a time average of previous activity of the cell modifies the amplitudes of the STDP timing curves, combining the short term plasticity of STDP with the long term plasticity of BCM.

Upon first glance, BCM and STDP seem to be completely unrelated and unrelatable - STDP is a Hebbian rule based entirely on the relative timing of spikes, whereas BCM is a rate-based plasticity rule where the average firing rate of the cell over time dictates synaptic modification. The study in Izhikevich & Desai, 2003 was motivated by the fact that in isolation, STDP can be demonstrated with clear pairs of postsynaptic and presynaptic spikes, in a natural cell with a natural spike train, where many possible pairings can be considered, and BCM effects become more and more prevalent. Since evidence for both BCM and STDP has been obtained from the same cells in the same regions of the brain, it can be presumed that both theories are describing different aspects of the same biophysical phenomenon. In this case, a unifying theory would be desirable.

To examine this, several different implementations of STDP were compared to simple BCM in a biologically realistic system - that is one with weakly correlated presynaptic and postsynaptic neurons, firing according to a randomly generated presynaptic input with a Poisson distribution.

After considering eight different models of synaptic plasticity (additive and multiplicative forms of classical STDP, several forms of nearest-neighbour STDP, and STDP with suppression (§2.4.6)), one was found to be the most suitable for making STDP and BCM compatible - that is a nearest-neighbour implementation, where only presynaptic and postsynaptic spikes which are immediately temporally adjacent are considered when calculating contributions to STDP. This is biologically realistic, since effects caused by the most recent spike may almost completely override any lingering effects from previous ones, due to back-propagation of the postsynaptic spike up the dendritic tree. This occurs because the dendrites conduct in all directions, with each cell spike sending an action potential up the dendrites as well as down the axon. Although this is not constantly reinforced as the action potential travels up the dendrites since dendrites lack voltage gated ion channels, it is sufficient to reach the synapses in question and "reset" any residual effects from previous spikes by overwhelming them.

Given that we assume only nearest neighbour interactions apply, along with a Poisson spike train input with average firing rate *x*, the average synaptic modification per presynaptic spike can be written, with the first integral representing average potentiation, and the second representing average depression, as:

$$C(x) = \int_{0}^{\infty} A_{+} \exp(-t/\tau_{+}) x \exp(-xt)\,dt + \int_{-\infty}^{0} A_{-} \exp(t/\tau_{-}) x \exp(xt)\,dt \qquad (21)$$

$$= x\left(\frac{A_{+}}{\tau_{+}^{-1} + x} + \frac{A_{-}}{\tau_{-}^{-1} + x} \right) \qquad (22)$$

When formulated this way, as an average modification per spike, STDP appears much more BCM-like, with a high activity resulting in potentiation, and a low activity resulting in depression. Finding the root of *C(x)* allows us to find the LTP/LTD threshold, which is:

$$\theta = \frac{A_{+}/\tau_{-} + A_{-}/\tau_{+}}{A_{+} + A_{-}} \qquad (23)$$

This demonstrates a link between STDP and BCM, and allows us to devise models of timing-dependent plasticity that include rate-based effects, using this modification threshold as a starting point, so long as we restrict ourselves to nearest-neighbour pairings. This relationship will also hold for semi-nearest neighbour parings, where we consider up to two spikes in the past (Izhikevich & Desai, 2003). This allows us to also consider triplet models.

It was later proposed (Benuskova & Abraham, 2007) that this synthesis of BCM and STDP could be implemented by dynamically changing the amplitudes A_+ and A_- of the STDP timing curves:

$$A_{+}(t) = \frac{A_{+}(0)}{\langle c(t) \rangle} \quad \text{and} \quad A_{-}(t) = A_{-}(0)\langle c(t) \rangle \qquad (24)$$

Here, <*c*> is an average of postsynaptic activity in the recent past, and scales the initial amplitudes $A_+(0)$ and $A_-(0)$. The length of time that is considered for this average is called the sliding window, and thus, a form of BCM-like plasticity is achieved - more activity leads to a smaller amplitude and thus less potential for LTP and more for LTD. This average is calculated as follows:

$$\langle c(t) \rangle = \frac{c_0}{\tau_{\mathrm{M}}} \int_{-\infty}^{t} c(t')\exp(-(t-t')/\tau_{\mathrm{M}})\, dt' \qquad (25)$$

with scaling constants c_0 and τ_M and c a postsynaptic spike count . This system of modifying the amplitudes of the STDP curves is then combined with normal STDP, and in the original implementation used a presynaptically centred pairing scheme with the weights updating as per this equation:

$$w(t) = w(1 + \Delta w_+ - \Delta w_-) \qquad (26)$$

with Δw_+ and Δw_- taken from equations (19) and (20). As the Benuskova & Abraham rule uses an implementation of BCM, no weight limits or re-normalisation are necessary.

2.4.6. Froemke rule

Froemke et al. (2005) proposed a "suppression model" for STDP that implements a sort of metaplasticity. In this paper, they compared the basic "history-independent" STDP model (i.e. one without any form of metaplasticity) to their suppression model. The suppression model is implemented as a dynamic scaling factor to the original STDP rule, which Froemke calls *F(Δt)*. The weights are changed as follows:

$$\mathrm{F}(\Delta \mathrm{t}) = \begin{cases} A_+ \exp(-\mid \Delta t \mid / \tau_+) & \text{if } \Delta t > 0 \\ A_- \exp(-\mid \Delta t \mid / \tau_-) & \text{if } \Delta t < 0 \end{cases} \qquad (27)$$

$$\Delta w = \varepsilon^{pre} \varepsilon^{post} F(\Delta t) \qquad (28)$$

$$\varepsilon_k = 1 - \exp(-(t_k - t_{k-1}) / \tau_s \qquad (29)$$

where τ_s is a decay constant for the suppression effect, and ε^{pre} and ε^{post} are the values of ε^k (which is a placeholder for the contents of equation (29)) for the presynaptic and postsynaptic spikes that are being paired at each step. As can be seen, the scaling factors for the presynaptic and postsynaptic neurons are calculated separately based on when the neurons spiked, and then multiplied with the basic STDP weight change. The more spikes there are in a given period, the smaller the suppression factors will get, and the smaller the weight changes become. The duration of this effect is adjusted with $\tau_{s.}$

Although this effect is metaplastic in nature (that is, it adjusts the rate at which the plasticity itself occurs), it is by no means BCM-like. No hard-coded limits on weight are necessary here, since the suppression effect limits weight increases.

2.4.7. Pfister rule

This rule considers not pairs of spikes, but triplets, and, as such, is completely different in form to the aforementioned ones. The Pfister & Gerstner (2006) rule proposes that in addition to the pairs of spikes that have been considered in the previous rules, there is also a lesser effect on the overall weight change coming from a potential third spike in this association.

This rule updates the weights according to the equations:

$$w(t+1) = w(t) - o_1(t)[A_2^- + A_3^- r_2(t-\varepsilon)] \text{ if } t = t_{pre} \tag{30}$$

$$w(t+1) = w(t) + r_1(t)[A_2^+ + A_3^+ o_2(t-\varepsilon)] \text{ if } t = t_{post} \tag{31}$$

As can be seen, in addition to the amplitudes A_2 and A_3, inside these two weight change equations are embedded four functions which are described by the four differential equations:

$$\frac{dr_1(t)}{dt} = -\frac{r_1(t)}{\tau_+}; \quad \text{if } t = t_{pre} \text{ then } r_1 \rightarrow r_1 + 1 \tag{32}$$

$$\frac{dr_2(t)}{dt} = -\frac{r_2(t)}{\tau_x}; \quad \text{if } t = t_{pre} \text{ then } r_2 \rightarrow r_2 + 1 \tag{33}$$

$$\frac{do_1(t)}{dt} = -\frac{o_1(t)}{\tau_-}; \quad \text{if } t = t_{post} \text{ then } o_1 \rightarrow o_1 + 1 \tag{34}$$

$$\frac{do_2(t)}{dt} = -\frac{o_2(t)}{\tau_y}; \quad \text{if } t = t_{post} \text{ then } o_2 \rightarrow o_2 + 1 \tag{35}$$

All four of these variables behave similarly, and each decays exponentially according to its own decay constant. t_{pre} is any time when a presynaptic spike occurs, likewise t_{post} is any time when a postsynaptic spike occurs. Thus, these variables track the behaviour with the cell, with the value of the relevant variables being incremented by 1 whenever a presynaptic or postsynaptic spike occurs. This serves to facilitate the triplet interactions. The Pfister rule does not use any form of weight limit.

2.4.8. Clopath rule

The Clopath rule (Clopath et al., 2010) is unlike the other models mentioned, in that it does not consider neuronal spikes as discrete events - instead it makes changes to the weights based on the cell's membrane voltage at each time step.

Here, LTP and LTD are described by separate equations:

$$\frac{dw^-}{dt} = -A_-(\langle u \rangle)X(t)(u_-(t) - \theta_-) \qquad \text{if } w > w_{min} \tag{36}$$

$$\frac{dw^+}{dt} = A_+ x(t)(u - \theta_+) \qquad \text{if } w < w_{max} \tag{37}$$

In these equations are contained more equations. *u* is the membrane voltage taken directly from the current value in the simulation (and is for each plasticity application treat as a constant at that instant), θ_- and θ_+ are adjustable parameters, A_+ and A_- are the LTP and LTD amplitudes, and *X(t)* is a variable which is set to 1 when there is a presynaptic spike and 0 when there is not one. The rest of the variables are described by the equations:

$$\tau_- \frac{du_-(t)}{dt} = -u_-(t) + u \tag{38}$$

$$\tau_x \frac{dx(t)}{dt} = -x(t) + X(t) \tag{39}$$

$$A_-(\langle u \rangle) = A_-(0)\frac{\langle u \rangle^2}{\langle u \rangle_{ref}^2} \tag{40}$$

Here, τ_- and τ_x are decay constants, $A_-(0)$ is the base LTD amplitude, and <*u*> is a moving average of the membrane voltage *u*, with a set reference value as a simulation parameter.

It is worth noting in this model that metaplasticity for LTP and LTD are handled differently. As seen in equation (40), metaplasticity for LTD is BCM-like, whereas metaplasticity for LTP is handled in equations (37) and (39), and is very similar to the suppression effect in the Froemke rule.

Consisting of four coupled differential equations, this STDP rule is by no means simple, but according to the authors cannot be simplified any further (Clopath 2010).

In contrast to the other rules discussed here, the Clopath rule does insist on upper and lower bounds for the weights. Without these, it is possible for a weight to become negative and give absurd results.

3. The Model

3.1. Basic Design

For all simulations discussed in this work, we use the same basic model, with modifications for exploring the effects of different methods. At the core, we use an implementation of the Izhikevich spiking neuron model in the C programming language, for a dentate granule cell with two inputs, corresponding to the MPP and LPP.

As it stands, two inputs is a gross oversimplification of the complexity of the biological system in question. In addition to both the MPP and the LPP consisting of thousands of axons making thousands of synapses on each of the granule cells, we also ignore input from inhibitory cells as well as recurrent connections from the granule cell and its neighbours. However, previous work on this system has shown that a simplified treatment may well be sufficient (Benuskova & Abraham, 2007).

In this simplified approach, instead of simulating each of the thousands of axons in each of the MPP and LPP, each with their own weight, we will treat the MPP and the LPP each as a single synapse with a single weight. To counter for the expected low level of activity this would yield, each input to the cell will be multiplied by a scaling factor which will be a crude approximation to there being many fibres in the pathways. As a default, we set this scaling factor to 250 "fibres" per pathway.

The simulations are conducted with a time resolution of 1ms, and, since there is no easy way of gaining analytical solutions for the differential equations involved here, at each time step, the variables are updated by solving their equations using the forward Euler method, which is as follows:

$$y_{n+1} = y_n + \Delta t \cdot f(t_n, y_n) \tag{41}$$

As all of the differential equations we will use are in the form

$$\frac{dy}{dt} = f(t, y) \tag{42}$$

with a step size Δt of 1ms, which is equal to the simulation's time resolution, implementation of the Euler rule is trivial - we must only choose a starting value for each variable and add *f(t, y)* at each time step.

It is a worthwhile possible criticism to state that the Euler method is a primitive and inaccurate method for solving differential equations, and in general this criticism is valid. However, this will be sufficient for this simulation for a few reasons. Firstly, the length of the simulation is significantly greater than our chosen time step - with a time step of 1ms and a simulation time of several hours, any accumulated errors may cancel each other out. Secondly, given there is a repeat discontinuity in the voltage for every spike on account of the Izhikevich neuron's spike reset, higher order methods would be difficult to implement, whereas the first-order Euler method is able to work on the same time step as the simulation.

3.2. Input Modes

A cell in a live brain does not exist in isolation - and since the experiment we are modelling was indeed carried out in vivo, we will need to account for this natural activity. In a real granule cell, the MPP and LPP carry inputs from the entorhinal cortex, whose theta rhythm of 6 - 10 Hz is the most prominent oscillatory activity (Deshmukh et al, 2010). This however is an average rate - as the brain is communicating meaningful information, this is not a constantly periodic input.

For simulating this input activity, we have three options ranging from the simple to the most biologically realistic. At the simplest, we could have a perfectly periodic spike train with a frequency of 8Hz - this would be the simplest to implement, requiring no additional coding work. However, an earlier preliminary investigation into this (Hananeia, 2012) found that the cell did not behave as expected in this mode.

A more realistic option would be to use a Poisson spike train with an average frequency of 8Hz. This is very simple to implement, as we need only sample from the Poisson distribution:

$$P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!} \tag{43}$$

where P is the probability that there are k events happening in a given time, and λ is the mean number of times said event will occur in the given time. In our case, the λ used is the mean amount of spikes that will occur in 1 millisecond, and for a frequency of 8Hz, that is 0.008. Implementing this was simple – at this point, all that was needed to be done was to draw a uniformly distributed random number between $0<n<1$ using native C libraries and then check to see if this random number was less than 0.008. If it was, a spike was sent (Hananeia, 2012). Although random, the spontaneous activity occurs across both pathways simultaneously.

The most biologically realistic input method would be to simulate a quasi-periodic spike train - that is a spike train with a frequency of 8Hz, plus or minus some random "dither" with each spike. This however brings about its own problems in that difficulty of implementation would be high for a result that would, given the tendency of the Poisson distribution to approximate a periodic spike train over long intervals, likely very closely resemble the input we already have.

In addition to the natural random input from the entorhinal cortex, we also must simulate the artificial sources of input that were present in the experiment. There are two sources of this - the HFS used for LTP induction, as well as the test input.

The test input, although insignificant compared to the random input and the HFS by virtue of its low frequency relative to the random input (once per 10 seconds versus 8 Hz), must still be implemented for the sake of completeness. Although in experiment it is supposed that the test input is low enough in intensity to not cause any LTP or LTD, it is still worth implementing so that any effect it may have is still reproduced. This is done with a single spike to both the MPP and LPP every 10 seconds, with an intensity lower than any other spike. This is accomplished by it only engaging 150 of the 250 "virtual fibres".

The HFS is somewhat more complicated. In the experiment, the HFS consisted of 10 bursts of stimulation, each consisting of 5 trains of spikes. Each of these spike trains consisted of 25 spikes at 400Hz. Implementing this exactly is however impossible given our simulation resolution of 1ms. This is because the period of a 400Hz spike train is 2.5ms. Since this is not an integer multiple of the simulation resolution and this cannot be simulated properly, we must either set the resolution to 0.5ms or use an approximation.

Adjusting the simulation resolution to 0.5ms would have been difficult given how many functions in our implementation work on the assumption of a 1ms time step, so the approximation was chosen for the sake of ease of implementation. We generated the trains using a Poisson random process to give 25 spikes with an average frequency of 400Hz, using an identical process to that which was used for our random input. This may seem excessively unrealistic, but it is worth considering that the stimulating electrode used in the experiment is a mechanical device with its own innate imperfections, and is very unlikely to deliver a consistent output of exactly 400Hz. Combined with the aforementioned cancellation of accumulated errors over the duration of the simulation, this should make the approximation here adequate.

During the HFS, the correlated spontaneous activity is de-correlated, that is it occurs with the mean frequency of 8Hz at both MPP and LPP but is generated independently according to the Poisson distribution. As has been seen in earlier work (Benuskova & Abraham, 2007), any heterosynaptic plasticity will not occur unless spontaneous activity is maintained on the adjacent pathway during LTP induction.

3.3. Cell Parameters

In the Izhikevich model (eqs. 12, 13, 14), there are four parameters *a*, *b*, *c*, and *d* which give the model much of its wide spectrum of applicability. The choice of these four parameters will determine which kind of cell the model is simulating - and so, getting these correct is of great importance before any kind of experiment is undertaken.

To ensure that our implementation of the Izhikevich model is working correctly, we conducted several preliminary tests on a variety of parameter sets for different cell types in response to both constant and random inputs to the cell (Hananeia, 2012). In this investigation we were able to obtain reasonable reproductions of some of Izhikevich's results:

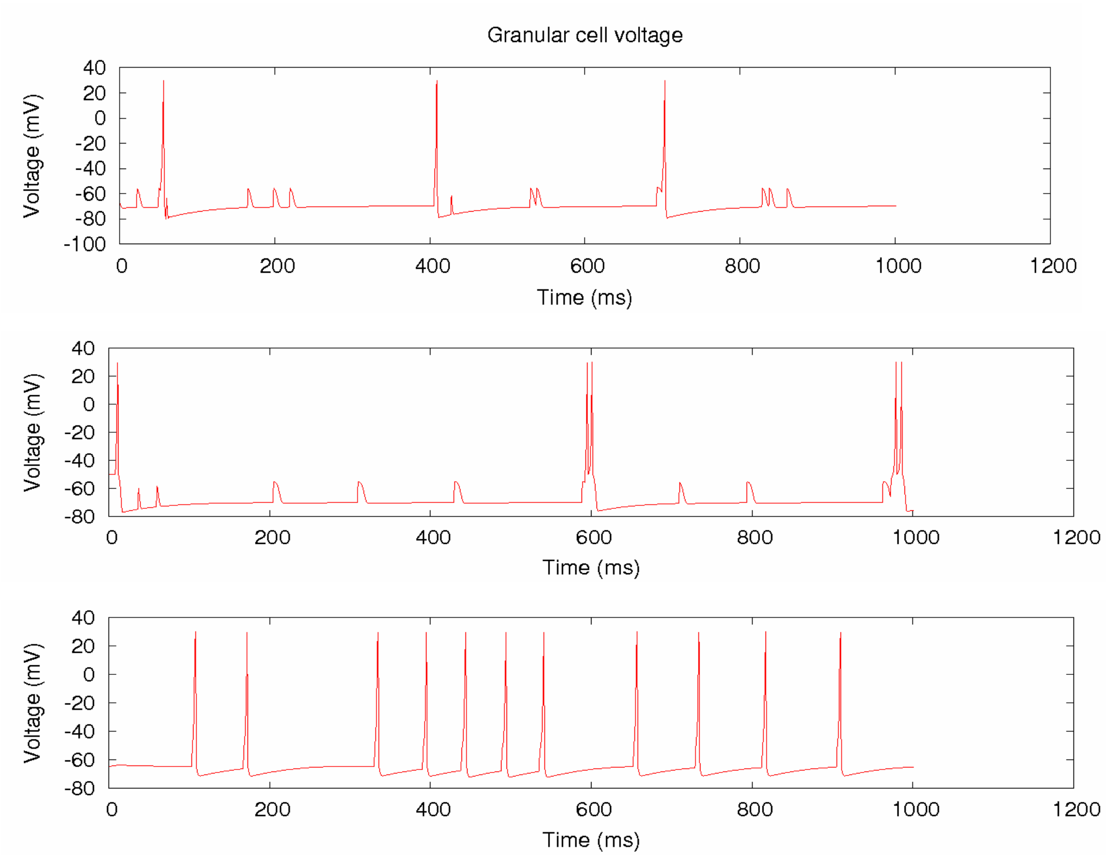

Fig. 14:

(top) : Membrane voltage of regularly spiking cell ([a, b, c, d] = [0.02, 0.2, -65, 8]) in response to random input;

(middle): Membrane voltage of chattering cell ([a, b, c, d] = [0.02, 0.2, -50, 2]) in response to random input;

(bottom): Membrane voltage of low-threshold spiking cell([a, b, c, d] = [0.02, 0.25, -65, 2]) in response to random input.

As can be seen in Fig. 14, we were able to obtain reasonable reproductions of a few of the cell types with our Poisson-type random input using the parameters given in Fig. 10 (Izhikevich, 2003). In addition to this, we also simulated the same cell types with a constant input, which gave us results more in line with Izhikevich's, in particular for the chattering cell, as shown in Fig. 15:

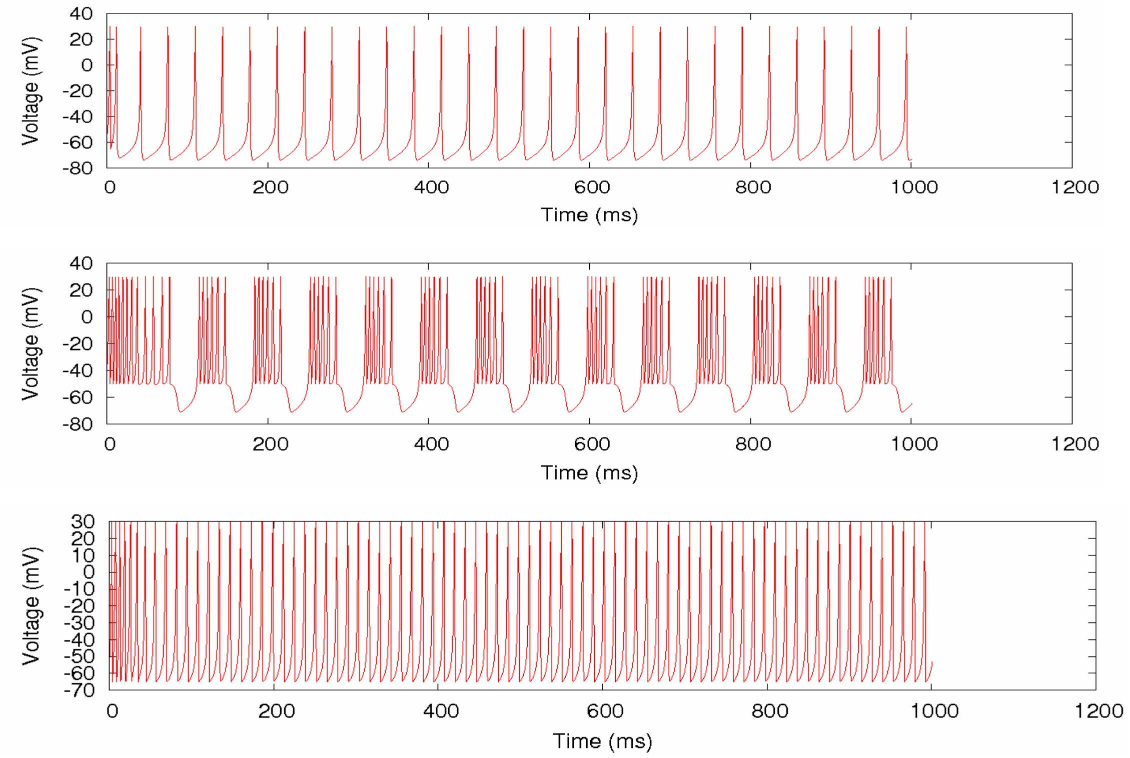

Fig. 15:

(top): Membrane voltage of regularly spiking cell ([a, b, c, d] = [0.02, 0.2, -65, 8]) in response to constant input;

(middle): Membrane voltage of chattering cell ([a, b, c, d] = [0.02, 0.2, -50, 2]) in response to constant input;

(bottom): Membrane voltage of low-threshold spiking cell([a, b, c, d] = [0.02, 0.25, -65, 2]) in response to constant input.

As the dentate granule cell is a regularly spiking cell (Rose et al., 1983), for the rest of our simulations here, we use the parameters for such a cell: [0.02, 0.2, -69, 2], with a slightly lower c value, which corresponds to the cell's resting potential.

3.4. STDP parameters

Each of the STDP rules discussed has its own set of parameters, describing various aspects such as the shape of the STDP curves, or decay constants for other associated functions, with some of the models having as many as six independent parameters. These models are crafted for different cells in different areas of the brain, and each of the parameters in these corresponds to some unknown physical quantity. However, the actual values of the parameters cannot be directly measured, and they do not remain the same between different cells, and in fact, differ for the same cell for even as trivial a change as spike pairing scheme.

Because of these inherent unknowns, every time we implement a new model, we have no immediately accurate way of knowing what parameters to choose. A good analogy for this are the constants in physical laws, such as for Newton's theory of universal gravitation:

$$F = G\frac{m_1 m_2}{r^2} \tag{44}$$

Although Newton was able to theorise gravity was an inverse-square relationship, this was not enough - he had to find a value for *G* which made the theory come into line with physical observation. Any value of *G* except the valid one would not produce reliable results.

Much the same happens here - if incorrect parameters are chosen, the model will not produce accurate simulations, the results of simulations with bad parameters verging on nonsensical - for example, the weights increasing without bound to enormous values, or all cell activity stopping. In this study, we have opted to slowly explore the parameter space manually, changing the various parameter values until an end result that resembles the experiment is achieved. Since we are only doing a qualitative rather than a quantitative investigation, mere resemblance to the experimental result is sufficient.

Unfortunately, automating this task of finding the correct parameters is very difficult in this case. While time-consuming, refining the parameters "by hand" relies on educated guesses and visual recognition of whether or not a final graph is closer to a "good" result than a previous attempt. Given the enormous space that must be explored (for six independent parameters, this would entail finding a small region in 6-dimensional space), any automated optimisation method would take a long time to implement and a similarly unacceptably long time to run, although such methods, such as evolutionary or genetic algorithms and Monte Carlo sampling do exist and could potentially be used. In the case of a genetic algorithm, for example, finding a suitable fitness function would be very difficult, and would require a thorough study of what leads to stability or instability in the simulations. Finding the correct parameters is thus the most time-consuming part of each experiment, but manual trial and error is still by far the fastest method to find optimal parameter values.

4. Results

For all of our simulations in this study, the same sequence of stimuli is used. The simulation will last 420 simulated minutes (the actual execution of a run taking between 10 seconds and a few minutes), with the first 60 minutes being treated as "negative time" - that is, time in which the simulation is run with only spontaneous input so that the system can stabilise. These 60 minutes will not be displayed on any of the results, the system being treated as if it runs from 0 to 360 minutes. At 0 minutes, the test input is applied for the remainder of the simulation, except for the 10 minutes following HFS onset, and then at 30 minutes, a single round of HFS is applied to the MPP. We will neglect the second round of LPP HFS that is shown in the physical experiment.

Testing of this basic protocol with the Benuskova & Abraham rule (Benuskova & Abraham, 2007)(§2.4.5) was done in earlier work (Hananeia, 2012), laying the groundwork for the experiments we conduct here.

Because there is no reliable way of tuning the cell's firing rate in our implementation of the Pfister, Clopath or Froemke rules, we will only put focus on the short-term effects of the HFS for the various implementations of the Benuskova & Abraham rule. In these other rules, the cell often exhibited firing rates in excess of 8Hz, rendering our examination of the effectiveness of these rules qualitative at best. Attempts we have made to change the firing rate of the cell by altering other parameters generally resulted in instability or a complete failure of the cell to fire at all, meaning that likely a completely different parameter set would be needed for stability under a different cell firing rate. Certainly, this is an avenue for further investigation.

4.1. Benuskova & Abraham rule

As the Benuskova & Abraham rule was the one used for the entirety of earlier development of this simulation, we pay special attention to it, conducting deeper experiments to test its overall robustness under different pairing schemes. Here, we test four pairing schemes - Presynaptic centred, symmetric, reduced symmetric and nearest spike (see Fig. 13 for illustration).

4.1.1. Presynaptic centred scheme

The original implementation of this simulation was built using the presynaptic centred scheme (Benuskova & Abraham, 2007) so the results here should be treated more as a baseline than a test *per se*. The particulars of this pairing scheme are illustrated in Fig. 13b. The STDP parameters used for our results in Fig. 16 are: A_+ = 0.02, A_- = 0.01 τ_+ = 20ms, τ_- = 100ms, c_0 = 2000.

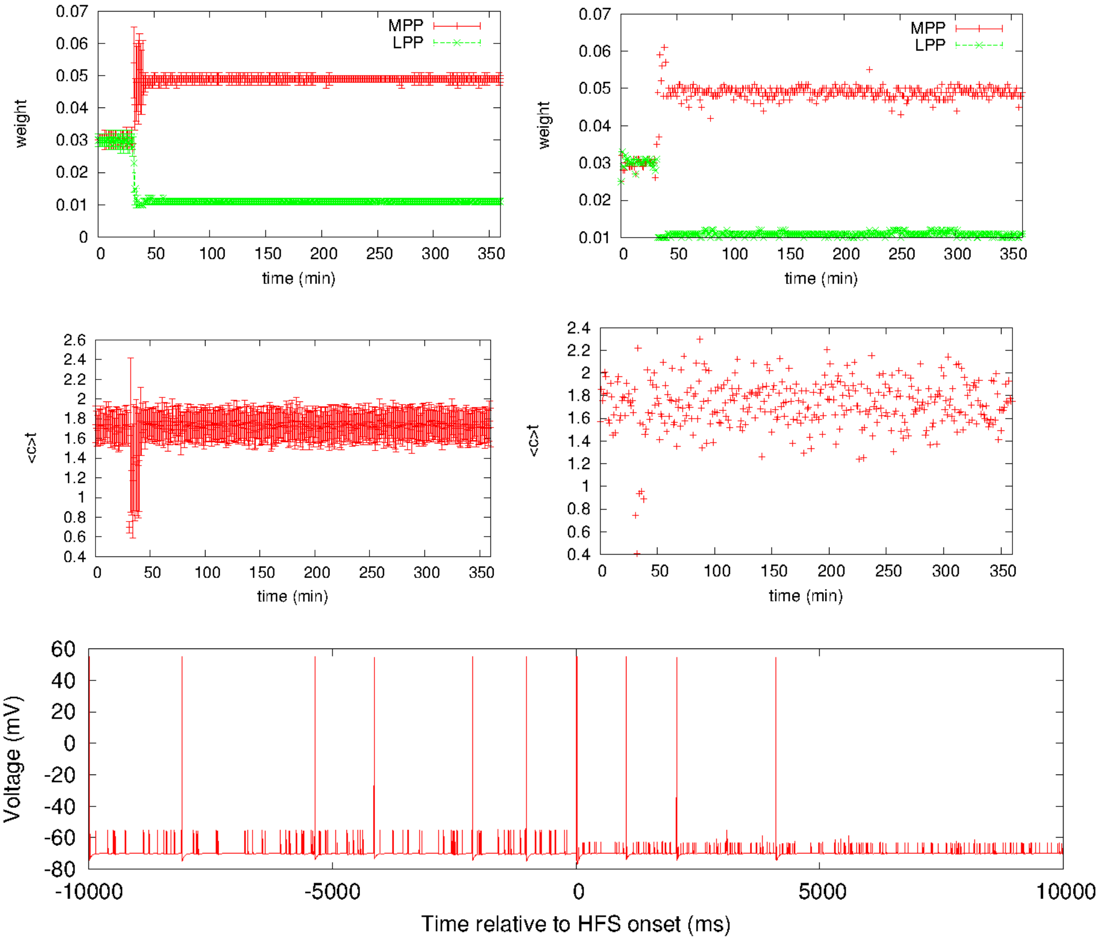

Fig. 16: Characteristics of the presynaptic centred pairing scheme with the Benuskova & Abraham rule. Top: Evolution of MPP(red) and LPP(green) weights over time; left: 30 run average with standard deviations, right: single run. Middle: Average cell activity over time; left: 30 run average with standard deviations, right: single run. Bottom: Membrane potential of cell around HFS onset.

Here, we see a very stable weight evolution, and a very clear homosynaptic LTP (i.e. LTP on the stimulated synapse, here the MPP) and heterosynaptic LTD (LTD on a synapse adjacent to one stimulated, but not directly stimulated itself, in this case the LPP) after HFS onset. During HFS, we can see the average activity of the cell decreasing sharply, before returning to pre-HFS levels afterwards. If we examine the membrane potential of the cell as shown in Fig. 16, we can see the associated reduction in cell activity.

As soon as the HFS induction period starts, we can see the base level of spontaneous activity (shown by the smaller spikes) halve - this is because we only apply spontaneous activity to the unstimulated pathway during HFS. The HFS's first, most immediate effect is a temporary silencing of the cell - for a period extending beyond 10 seconds after HFS start, the cell is only spiking in response to the HFS - we can see four of the five HFS pulses in the first train represented by a postsynaptic spike each. The missing HFS pulse is a random result of the nondeterministic nature of the simulation.

Occasionally, we see a case where the LTP induction protocol causes heterosynaptic LTP and homosynaptic LTD - exactly the opposite of what we would expect. The characteristics of these "inverted" runs are shown in Fig. 17:

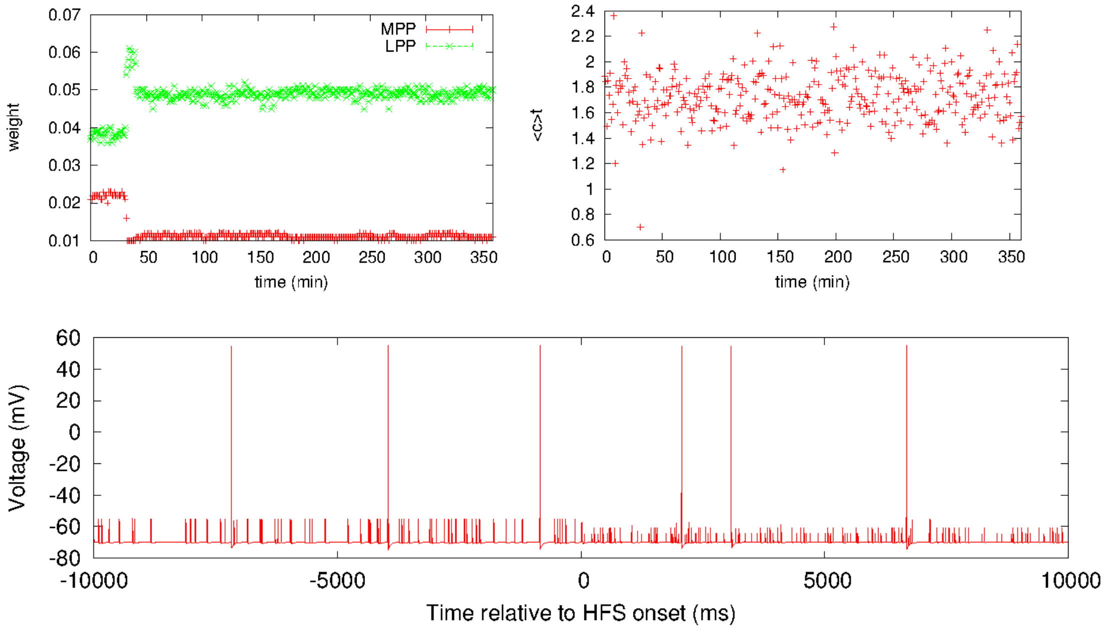

Fig. 17: Characteristics of an anomalous "inverted" run. Top left: Evolution of MPP(red) and LPP(green) weights over time. Top right: Average cell activity over time. Bottom: Membrane potential of cell around time of HFS onset.

This behaviour occurs very rarely - in over 1000 simulations, only 31 of them showed this pattern, 969 of them correctly reproducing the direction of the LTP. Of note is the MPP and LPP weights stabilising to different values before HFS start, and the lack of complete silencing of the cell after HFS, as seen in Fig. 16. However, as we can see in Fig. 17(b), there is still an immediate reduction in average cell activity during HFS.

Because of its extremely consistent behaviour (aside from the inverted runs of which there is a very low incidence, indeed the lowest of any of our pairing schemes), the presynaptic-centred scheme is well in agreement with experimental data.

4.1.2. Symmetric scheme

This pairing scheme is illustrated in Fig. 13a. To obtain the results in Fig. 18, we used the parameters: $A_+ = 0.002$, $A_- = 0.001$ $\tau_+ = 70$, $\tau_- = 150$, $c_0 = 2500$.

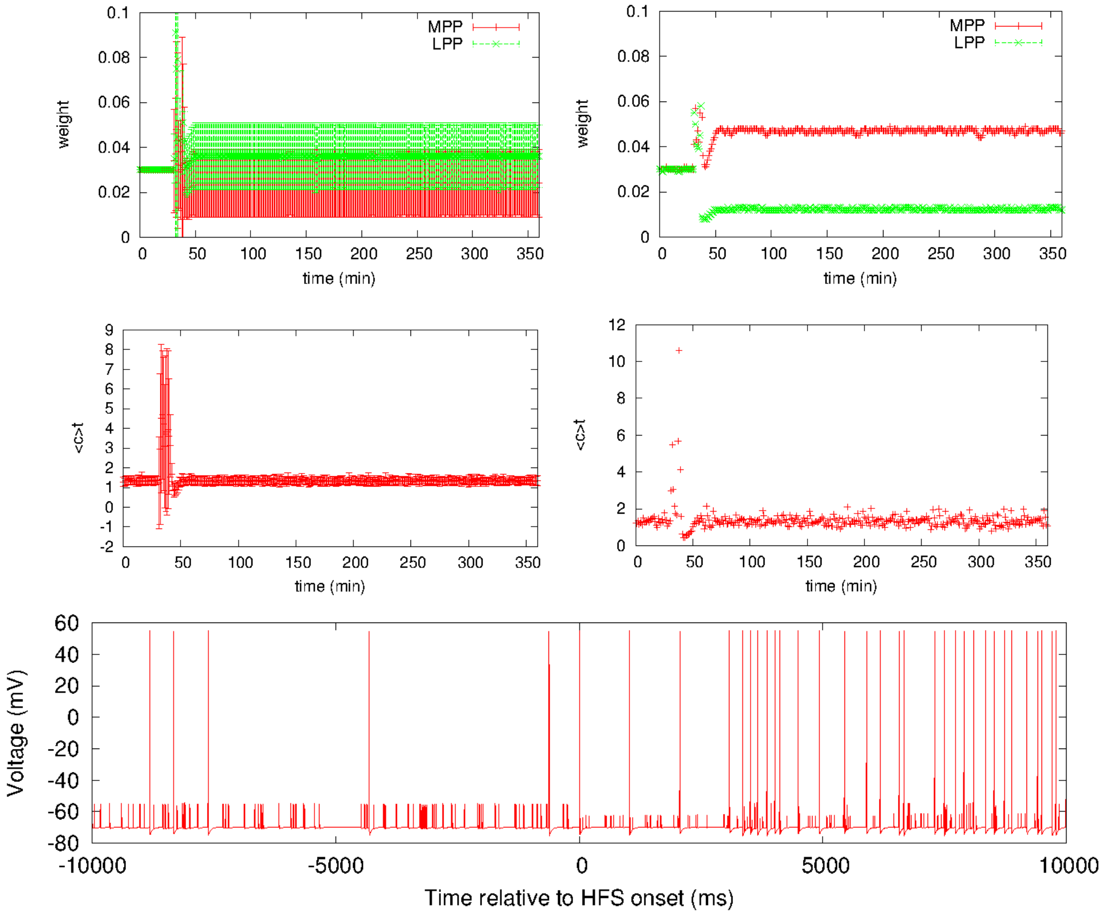

Fig. 18: Characteristics of the symmetric pairing scheme with the Benuskova & Abraham rule. Top: Evolution of MPP(red) and LPP(green) weights over time; left: 30 run average with standard deviations, right: single run. Middle: Average cell activity over time; left: 30 run average with standard deviations, right: single run. Bottom: Membrane potential of cell around HFS onset.

As can be seen in Fig. 18, the LTP of the MPP synapse and the LTD of the LPP synapse are no longer clear as they were under the presynaptic centred scheme. In fact, this is due to a much greater incidence of the "reversed" behaviour noted in §4.1.1; over 1000 simulations, the MPP weight was potentiated 403 times and the LPP weight was potentiated 597 times.

In addition, the behaviour of the cell during HFS is extremely strange - the cell acts as if there is no refractory period, spiking extremely quickly during the HFS. The "silencing" of the cell seems to come well after HFS, as we can see a reduction in $<c>$ after the HFS in Fig.18.

Thus, since the "wrong" weight is potentiated more often, and in addiction the cell shows strange spiking behaviour during HFS, the symmetric scheme is not in agreement with experimental data.

4.1.3. Reduced symmetric scheme

This scheme is illustrated in Fig. 13c. To obtain these results, shown in Fig. 19, we use the parameters $A_+ = 0.002$, $A_- = 0.001$ $\tau_+ = 70$, $\tau_- = 150$, $c_0 = 2500$, identical to those used for the symmetric scheme.

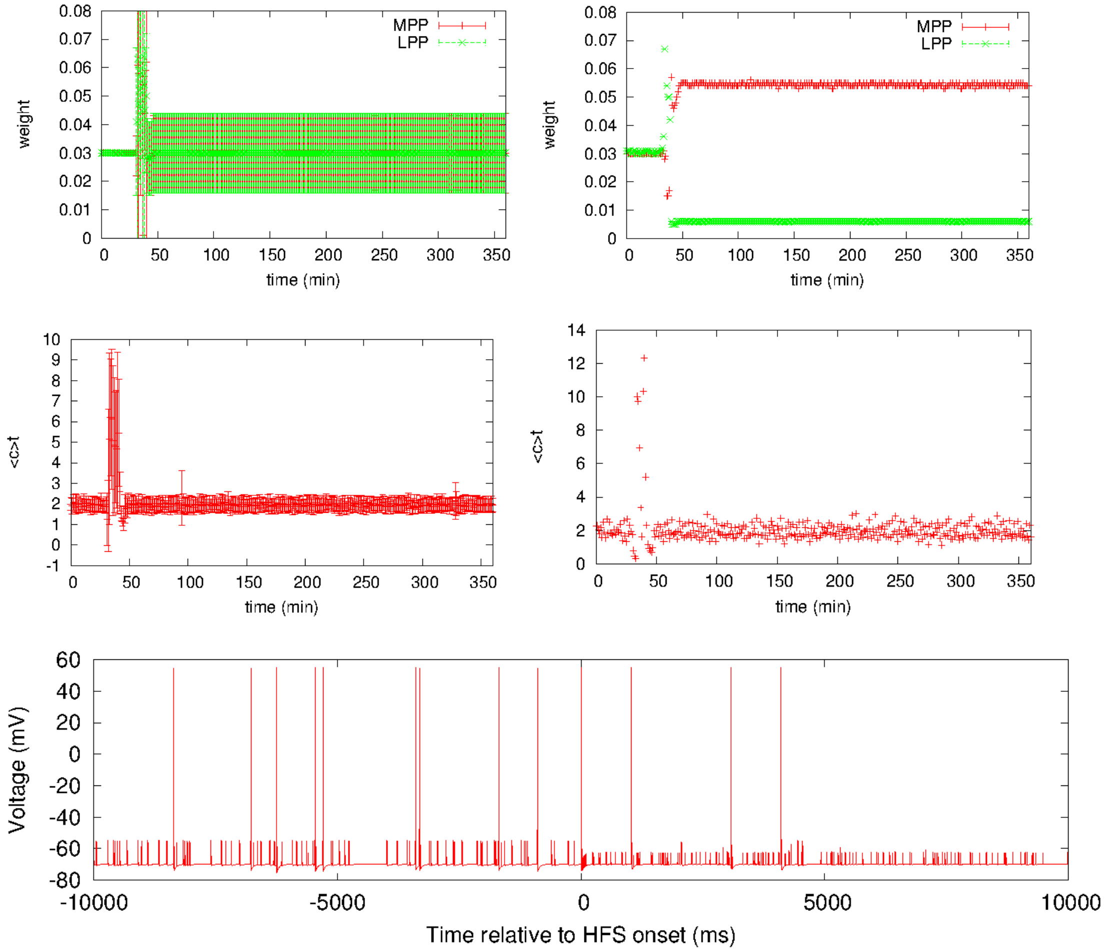

Fig. 19: Characteristics of the reduced symmetric pairing scheme with the Benuskova & Abraham rule. Top: Evolution of MPP(red) and LPP(green) weights over time; left: 30 run average with standard deviations, right: single run. Middle: Average cell activity over time; left: 30 run average with standard deviations, right: single run. Bottom: Membrane potential of cell around HFS onset.

As would be expected from a scheme that uses a subset of the pairings used in the symmetric scheme, this implementation produces very similar results. Similar to the symmetric scheme, heterosynaptic LTP/homosynaptic LTD is significantly more common than with the presynaptic centred scheme, this time the "reversed" behaviour happening nearly equally as often as the expected behaviour when tested over 1000 runs - 430 MPP potentiations and 570 LPP potentiations. This is not in agreement with experiment.

If we examine the average activity of the cell, we however find much more stable behaviour than in the symmetric scheme. There is an immediate silencing of the cell which is clearly seen in the voltage trace. This is however followed by a massive increase in cell activity, but this is however not immediate as it was in the symmetric scheme.

It is worth noting that despite lengthy parameter optimisation, both the symmetric and reduced symmetric models occasionally displayed the absurd behaviour mentioned in §3.4 - for approximately 1 in 20 runs, the weights would increase without bound. In all cases, stability of the simulation is achieved only thanks to the stabilising action of the dynamic adjustments of potentiation and depression amplitudes - we do not use any form of hard caps on weights in the Benuskova & Abraham model.

4.1.4. Nearest spike scheme

This scheme can be visualised as a modification of the scheme in Fig. 13b where only the very closest interactions are considered. To obtain these results, shown in Fig. 20, we used the parameters: $A_+ = 0.01$, $A_- = 0.01$ $\tau_+ = 20$, $\tau_- = 40$, $c_0 = 3500$.

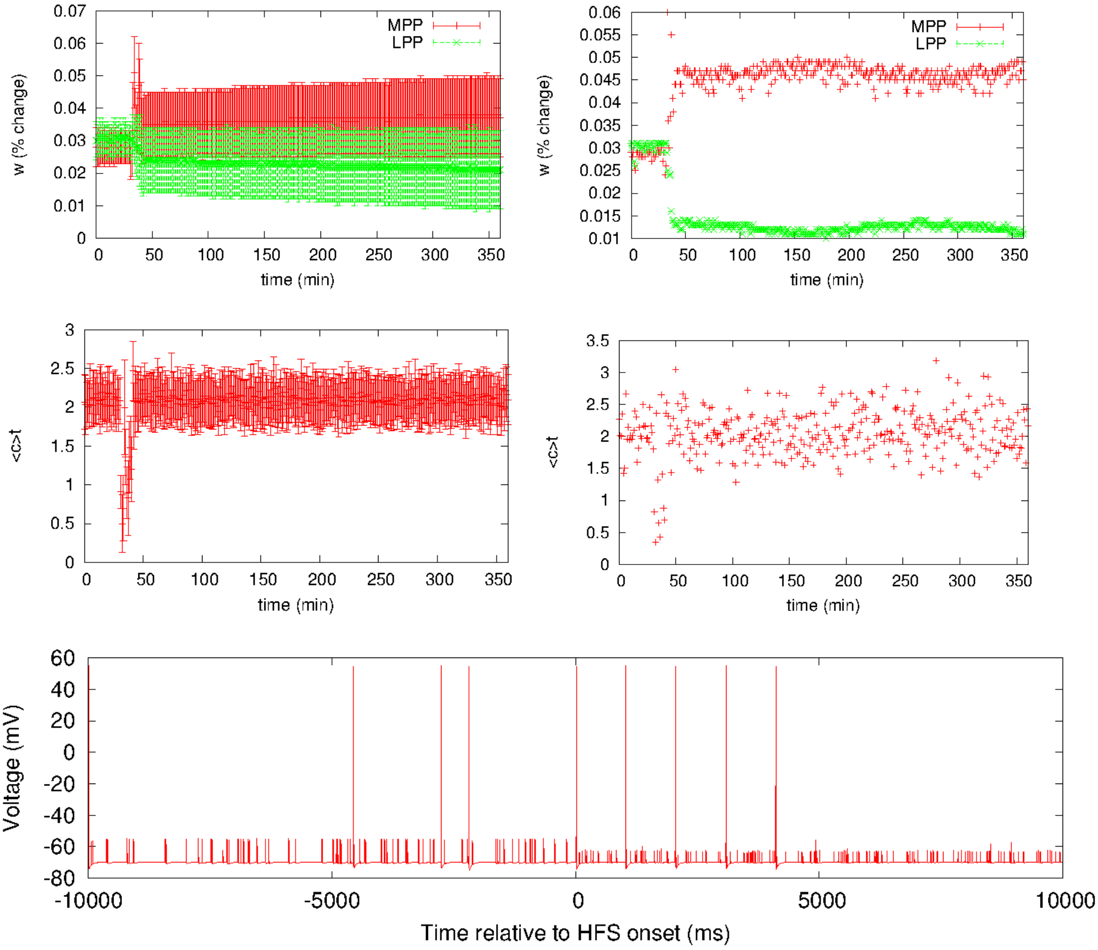

Fig. 20: Characteristics of the nearest spike pairing scheme with the Benuskova & Abraham rule. Top: Evolution of MPP(red) and LPP(green) weights over time; left: 30 run average with standard deviations, right: single run. Middle: Average cell activity over time; left: 30 run average with standard deviations, right: single run. Bottom: Membrane potential of cell around HFS onset.

As with the presynaptic centred scheme, the symmetric scheme also reproduces the general form of the experimental results. However, it is worth noting that the magnitude of the changes elicited by the HFS here is less than those with the presynaptic centred pairing, even when the smaller size of the A_+ parameter is considered. Also the changes are nowhere near as predictable - the standard deviations on the weights after HFS are much wider than they were under the presynaptically centred scheme.

Similarly to the other schemes, this scheme will also produce the "reversed" result occasionally. Here, for 1000 runs, the MPP was potentiated 782 times and the LPP 218 times. Although still in concordance with experiment when these "inverted" runs do not happen, the nearest spike scheme is less consistent than the presynaptic centred one.

4.2. Conventional STDP

Here we attempted to use naïve STDP without any modifications. Without a “decay” term of the following sort:

$$w_{t+1} = w_t - \lambda w_t \tag{34}$$

both weights would slowly increase linearly without bound. This simple decay term countered this linear increase allowing us to see the underlying behaviour. It is not unimaginable that such a decay may have a biological analogue in the nature of short-term synaptic plasticity (Benuskova & Kasabov, 2007). In this simulation we retain the presynaptic centred pairing scheme.

To obtain the results shown in Fig. 21, we used the parameters $A_+ = 0.02$, $A_- = 0.01$ $\tau_+ = 20$ms, $\tau_- = 70$ms, $\lambda = 0.00001$.

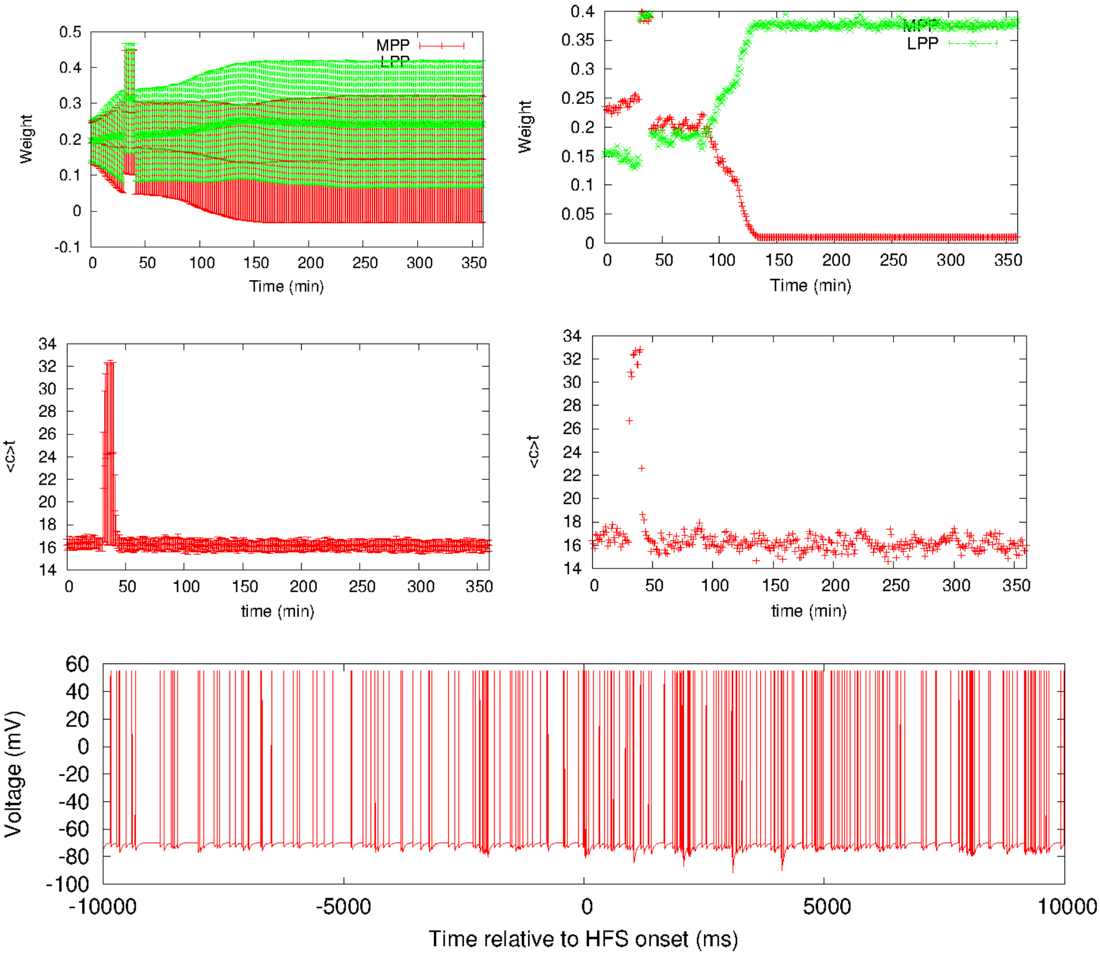

Fig. 21: Characteristics of the conventional STDP rule. Top: Evolution of MPP(red) and LPP(green) weights over time; left: 30 run average with standard deviations, right: single run. Middle: Average cell activity over time; left: 30 run average with standard deviations, right: single run. Bottom: Membrane potential of cell around HFS onset.

As can be seen, this does not reproduce the experimental results, the MPP increasing and the LPP decreasing after simulation start, with a reset to zero and reversal of directions upon HFS induction, stopping after they reach a “stable” value. Although HFS has an obvious effect, it does not trigger stable LTP induction, and thus is in no way concordant with experiment.

4.3. Froemke rule

Without modification, the Froemke rule had a similar fault to that of conventional STDP – slow linear increase of both weights. To counteract this, a similar decay term was introduced. Also, the simulation here is based on a presynaptic centred pairing, but non nearest-neighbour effects are introduced by the suppression model.

To obtain the results shown in Fig. 22, we used the parameters $A_+ = 0.02$, $A_- = 0.01$ $\tau_+ = 20\text{ms}$, $\tau_- = 70\text{ms}$, $\tau_s^{pre} = 35\text{ms}$, $\tau_s^{post} = 78\text{ms}$, $\lambda = 0.0000085$:

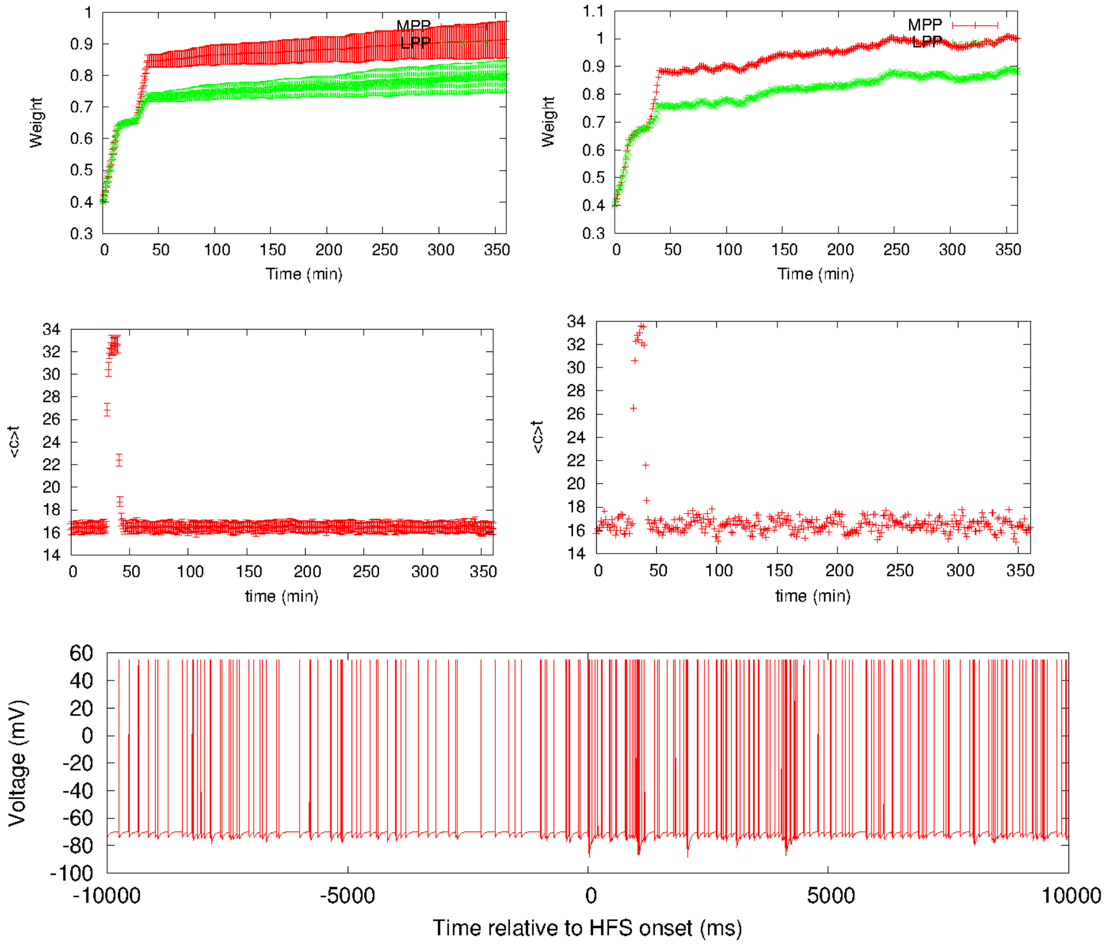

Fig. 22: Characteristics of the Froemke rule. Top: Evolution of MPP(red) and LPP(green) weights over time; left: 30 run average with standard deviations, right: single run. Middle: Average cell activity over time; left: 30 run average with standard deviations, right: single run. Bottom: Membrane potential of cell around HFS onset.

Here, after increasing to a stable value before HFS, the HFS causes both pathways to experience LTP, however the MPP more greatly so. This is in partial concordance with the experiment – we are not seeing any heterosynaptic LTD, however LTP induction is still happening, significantly so on the predicted pathway.

However, the average activity of the cell greatly increases during the HFS period, well above the already extremely high baseline. As mentioned earlier, we have no easy way to control the cell's firing rate with this model, so we cannot make any judgement about how the cell's voltage immediately reacts to the HFS induction - any change is masked by the violent spiking rate. Attempts to lower the cell's firing rate by adjusting the input levels to the cell by means of reducing the amount of "virtual fibres" only led to the cell failing to fire at all, or the firing rate not changing - we were unable to find any middle values in which the rate could be tuned.

It is also of note that this rule takes significantly longer for the weights to stabilise, not occurring until well after the 60 minute stabilisation period that was sufficient for all other rules. To be sure that the cell in fact stabilised, we tested the cell's behaviour over time without any HFS:

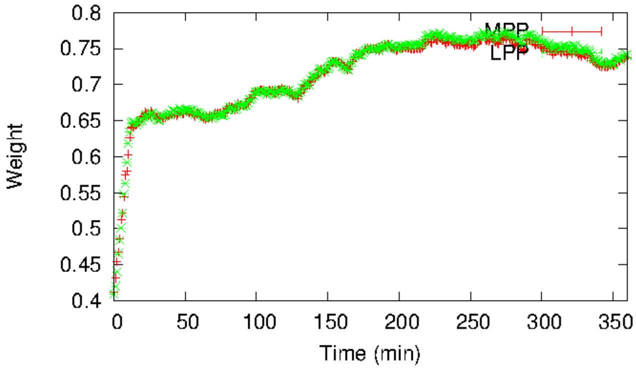

Fig. 23: Evolution of MPP(red) and LPP(green) weights over time for the Froemke rule without any HFS applied.

As we can see in Fig. 23, the cell is reasonably stable after this initial period of increase. While not totally flat as we would expect, the random undulations and remnants of the slow increase mentioned earlier are not of concern.

4.4. Pfister rule

With the Pfister model, we retained the majority of the simulation's framework used earlier in the presynaptic centred "baseline" model, making amendments to facilitate updating of the model's dynamic variables, which are what is responsible for the model's consideration of triplets of spikes.

To obtain the results in Fig. 24, we used the parameters $A_2^+ = 0.02$, $A_2^- = 0.008$, $A_3^+ = 0.0000006$, $A_3^- = 0.0000003$, $\tau_+ = 30$, $\tau_- = 100$, $\tau_x = 50$, $\tau_y = 120$.

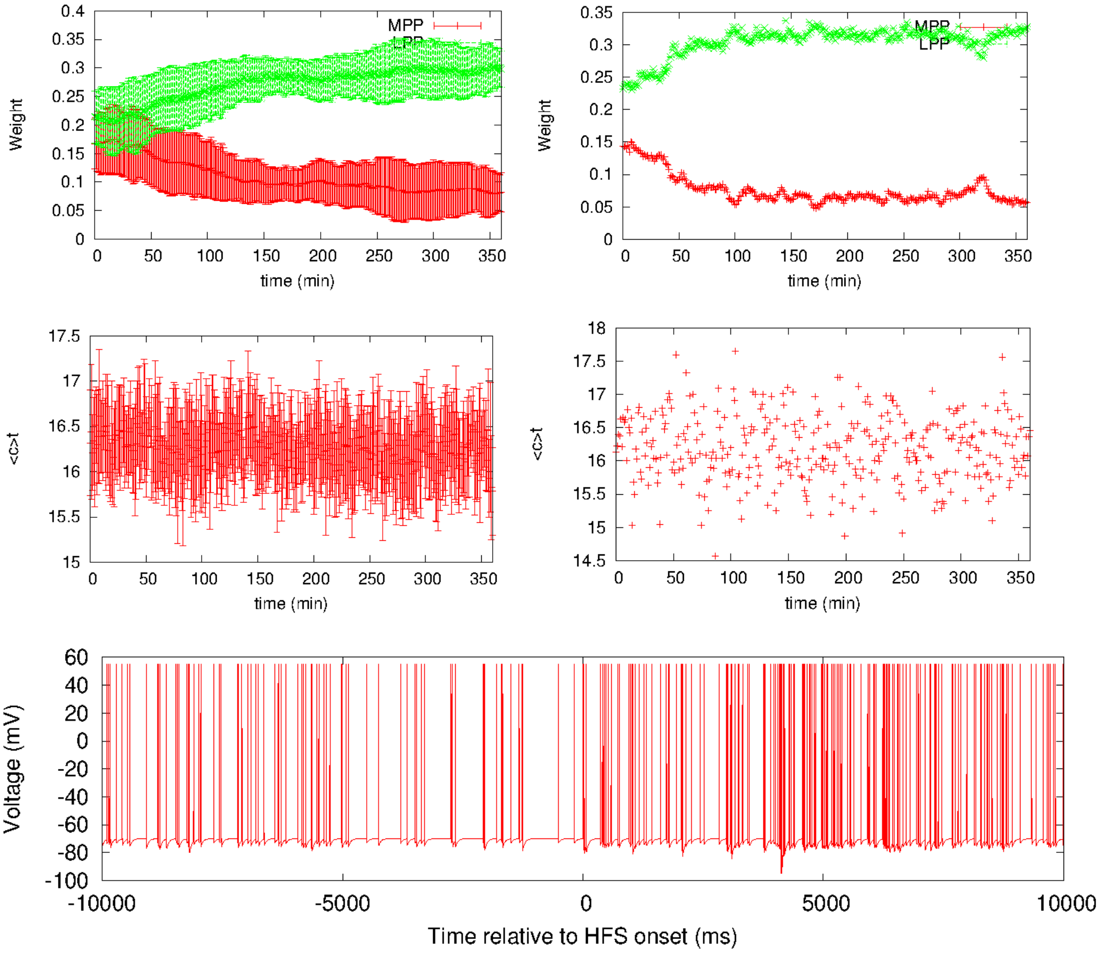

Fig. 24: Characteristics of the Pfister rule. Top: Evolution of MPP(red) and LPP(green) weights over time; left: 10 run average with standard deviations, right: single run. Middle: Average cell activity over time; left: 10 run average with standard deviations, right: single run. Bottom: Membrane potential of cell around HFS onset.

For the Pfister rule, we were unable to obtain any concordance with the experimental results; shown is a result for the parameters we will use in §4.4.1. With this rule, the HFS has either no or a negligible effect, and the weights seem to approach stable values. Testing this rule over a very long amount of simulation time also consistently showed that the weights remain at these “stable” values indefinitely.

4.4.1. Modified Pfister rule with BCM-like metaplasticity

After the Pfister rule failed to reproduce the experiment, we attempt a modification of Pfister’s rule with the BCM-like metaplasticity found in the Benuskova & Abraham rule. The same mechanism is used – dynamic adjustment of the LTP and LTD amplitudes based on a sliding window average of postsynaptic activity.

To obtain the results in Fig. 25, we used the parameters $A_2^+ = 0.02$, $A_2^- = 0.008$, $A_3^+ = 0.0000006$, $A_3^- = 0.0000003$, $\tau_+ = 30$, $\tau_- = 100$, $\tau_x = 50$, $\tau_y = 120$, $c_0 = 500$.

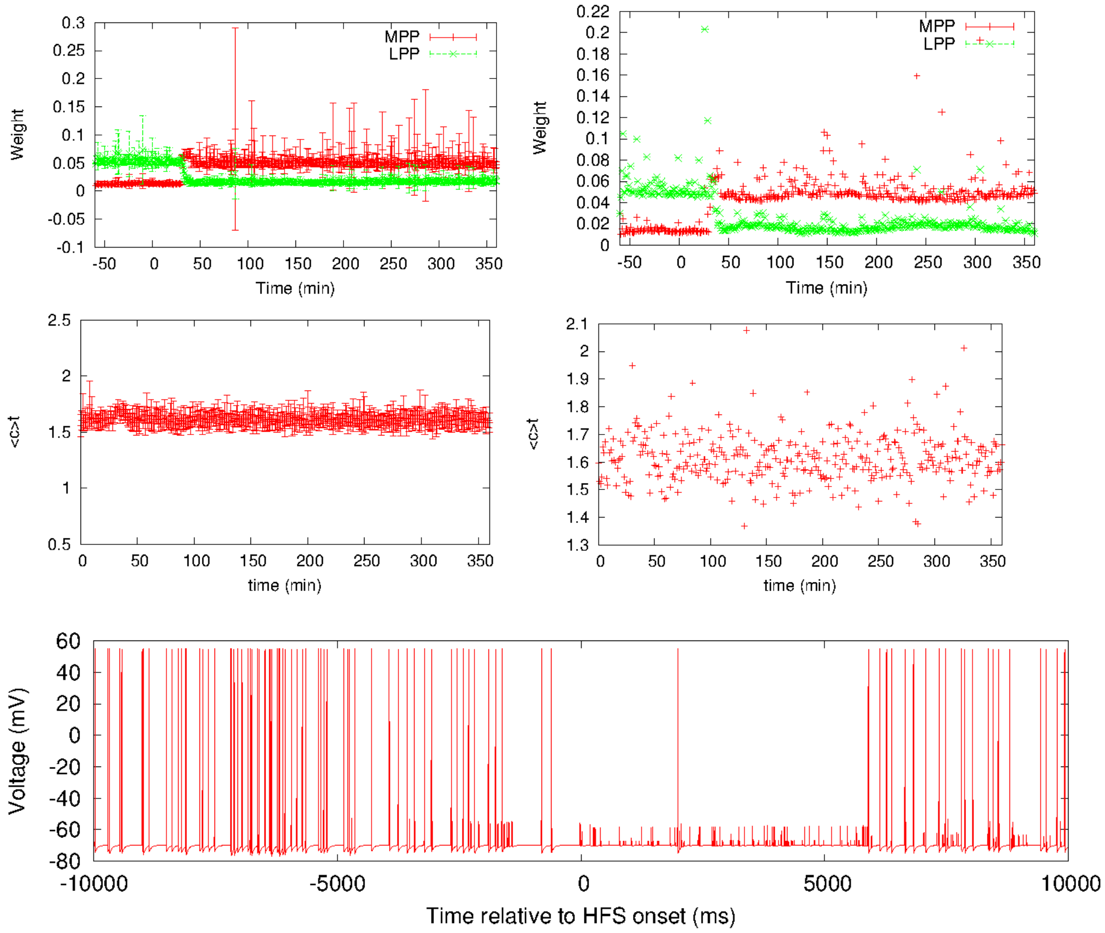

Fig. 25: Characteristics of the Pfister rule with added metaplasticity. Top: Evolution of MPP(red) and LPP(green) weights over time; left: 30 run average with standard deviations, right: single run. Middle: Average cell activity over time; left: 30 run average with standard deviations, right: single run. Bottom: Membrane potential of cell around HFS onset.

With this modification, the Pfister rule achieves partial concordance with experiment. Although the weights stabilise at different values before HFS, after HFS, the MPP is potentiated and the LPP depressed by equal amounts. This is not in concordance with experiment, however the characteristic behaviours exhibited in the experiment are somewhat maintained.

Although the cell has a relatively high spike rate in the region of 4 Hz, we still see a characteristic drop in cell activity immediately during the HFS, with the high spiking rate resuming shortly after the HFS train has finished. Unlike all of the alternative models we tried, this immediate response is in concordance with experiment.

We attempted to lower the spike rate of the cell to the expected 0.8 Hz by adjusting the c_0 parameter, however we were unable to achieve this while maintaining stability of the simulation. Similar efforts to lower the cell's firing rate by decreasing the number of "virtual" fibres and thus the overall input to the cell had an "all or nothing" effect - below a certain amount of fibres, the cell never fired at all, above that, it immediately rose to 4 Hz with no room to tune the firing rate. Similar consequences were seen in using these methods to counteract high fire rates in the Clopath model as well.

4.5. Clopath rule

To obtain the results shown in Fig. 26, we used the parameters $A_+ = 0.00038$, $A_- = 0.00003$, $\tau_+ = 8$ms, $\tau_- = 18$ms, $\tau_x = 16$ms, $\theta_+ = -38$, $\theta_- = -41$, $\lambda = 0.000000005$:

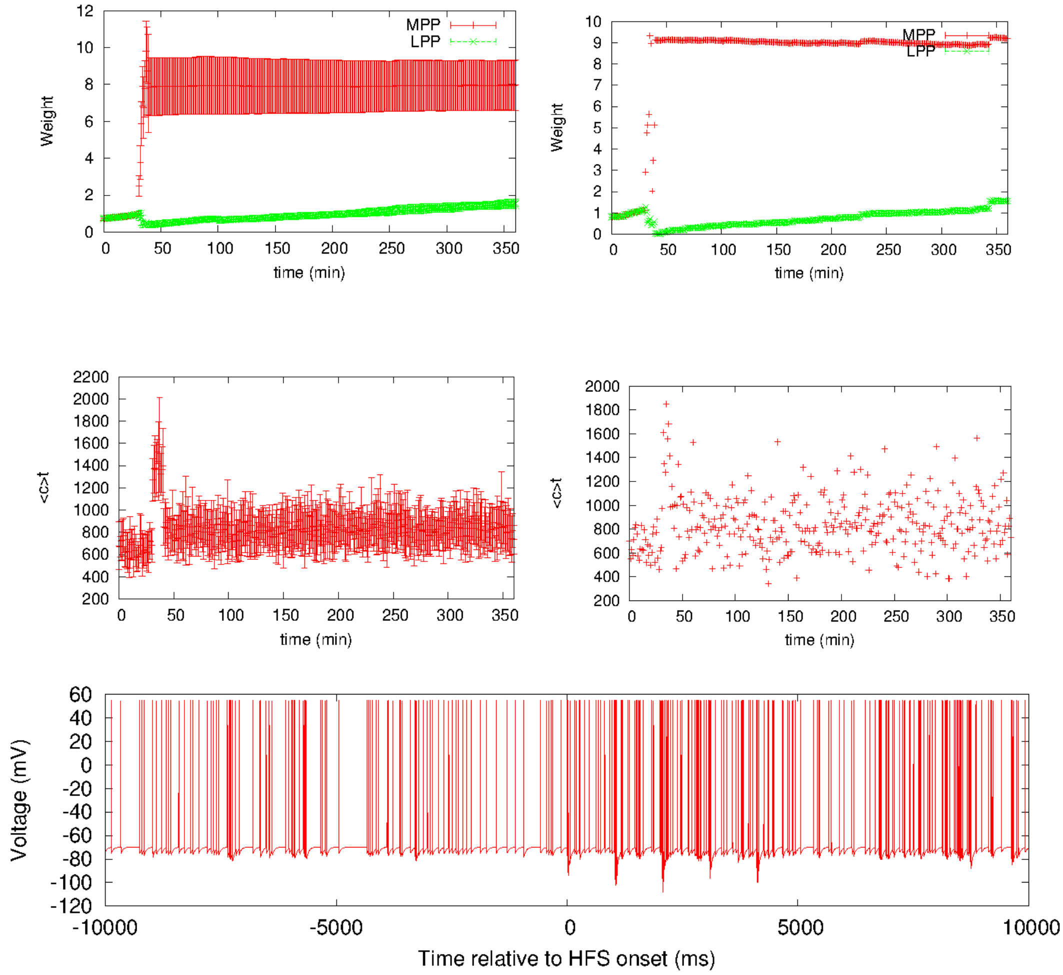

Fig. 26: Characteristics of the Clopath rule. Top: Evolution of MPP(red) and LPP(green) weights over time; left: 10 run average with standard deviations, right: single run. Middle: Average cell activity over time; left: 10 run average with standard deviations, right: single run. Bottom: Membrane potential of cell around HFS onset.

To achieve stability with this rule, we had to use a decay term to counter a steady increase in all weight values, much as we had to for conventional STDP and the Froemke rule. Also, we were only able to obtain 10-run averages without unstable behaviour - the chance of a failure as mentioned in §4.1.3 is much higher here for the most stable parameters we could find than in the other rules.

As can be clearly seen, in Fig. 26, the Clopath rule has excellent concordance with the experimental results as far as weight changes are concerned, reproducing both a strong homosynaptic LTP and heterosynaptic LTD reliably without any "inverted" results ever being observed.

However, this comes at a price - in order to achieve this result, we had to use a BCM sliding window of only 1 second, compared to the 60 seconds used in all other experiments. Increasing the duration of the window even modestly above this level results in the concordance with experiment rapidly decreasing - a longer window first removes the heterosynaptic LTD, and with an even longer window, stability of the weights after HFS is lost.

Also, the activity of the cell noticeably increases above the already very high level during HFS, much as we observed for the symmetric rules. This is not in concordance with experiment, but since we have little direct way to control the cell's firing rate here without introducing further instability that would then need to be corrected for, the high firing rate could be masking something deeper. We attempted the same methods for controlling the firing rate that we tried for the modified Pfister rule (§4.4.1), with identical results.

5. Discussion

In exploring the potential of the Izhikevich neuron model for other rules beyond the Benuskova & Abraham rule, we had a mix of successes and failures. Most surprisingly, we notice a particular significance to metaplasticity as defined by Abraham (2008), i.e., a degree of plastic change in the rate or direction of synaptic plasticity over time, in reproducing all of the behaviours we saw in experiment.

In many of the models and pairing schemes, we notice an immediate rise in cell activity following the HFS, in contrast to what we see with our well-tested presynaptically-centred scheme. For the models for which we had no way to control the cell's firing rate, any explanation for this could well be confounded by the high firing rates. These models were made for different parts of the brain and thus different kinds of cells - the Froemke rule for visual cortex cells (Froemke et al., 2006), the Pfister rule for the visual cortex and hippocampus (Pfister &Gerstner, 2006), and the Clopath rule for the cortex (Clopath et al., 2010).

However, when this is observed in the Benuskova & Abraham rule, we can offer a hypothesis to explain this behaviour, which was observed in the symmetric pairing schemes. The symmetric schemes pair much more indiscriminately than the presynaptic-centred or nearest spike (which is in fact a subset of presynaptic centred) schemes, and as we have seen from the statistics on the occurrence of the "inverted" runs, do not favour either the medial or lateral path. During the HFS, the LTP and LTD pairings could become highly synchronised, leading to a brief runaway increase in the weights that is corrected by the time the stimulation period has ended - which would be accompanied by an increase in cell activity and uncertainty in what direction the plasticity would take.

It is also worth considering that the immediate response of the cell to LTP induction is not directly predictable, and cell activity has been observed both decreasing and increasing in response to LTP (Kimura & Pavlides, 2000). The possibility that the symmetric schemes are in fact capturing a natural behaviour thus should not be dismissed.

The heterosynaptic LTD was not observed in all of the models we examined, notably the Froemke rule, and for the Clopath rule with certain parameters (the heterosynaptic LTD disappears when the BCM window is increased too far). It is possible that this is no co-incidence. When we consider that the Pfister rule only reproduced the experimental results when a BCM-like metaplasticity was introduced, it is possible that the explanation lies in metaplasticity itself.

The models that reproduced all the characteristic features in the weight evolution (Benuskova & Abraham, modified Pfister and Clopath) all had some form of BCM-like metaplasticity, whereas the Froemke rule which achieved partial concordance with experiment, had a (although in no way BCM-like) "suppression" mechanism which could be considered metaplasticity. It is possible that this could hint at a deeper link between heterosynaptic plasticity and metaplasticity (Hananeia & Benuskova 2014), and almost definitely shows that STDP, while a powerful theoretical framework, is wholly insufficient as a complete theory of synaptic plasticity. Given that the most successful forms of metaplasticity we experimented with were BCM-like, this gives the older and more established theory of BCM more credence when considered alongside the newer STDP. In fact, later research (Gjorgjieva et al., 2011) into the Pfister triplet model and BCM shows a tight concordance between the two for replicating complex neuron behaviours, again hinting a deeper link between STDP and metaplasticity.

With the Benuskova & Abraham rule, we also made a preliminary investigation into increasing the biological robustness of the model - rather than having 250 "virtual" fibres, we increased the number of actual simulated fibres to 10, each with 25 "virtual" ones. This produced identical results, with the individual weights therein remaining equal across each fibre. This can be interpreted several ways. The lack of any change in the behaviour could be showing a limit to what can be achieved with a simplified neuron model, with the vastly more complicated and computationally expensive multi-compartmental models being a necessary next step to increasing biological realism. Alternatively, the success of our very simple neuron model not changing upon increasing the pathway density could be seen as a testament to the applicability of simplified models, Izhikevich's one in particular.

6. Further Work

There are still many avenues of investigation which remain open after the experiments we have conducted.

The first and most obvious of these is the further exploration of the enormous parameter space that many of these models have, especially the Pfister and Clopath models. Even for Benuskova & Abraham, for which we have done a significant amount of investigation, there is still a very large parameter space, exploration of which might be able to mitigate some of the difficulties found in the symmetric pairing schemes, such as the high incidence of results completely opposite to those that we expected.

Secondly, the firing rates of the cell under all of the STDP rules except for Benuskova & Abraham were not in concordance with experiment, usually being 5-10 times higher. Because of this, thorough examination of the short-term response to HFS was usually impossible, and any comparisons we made were qualitative at best. Two methods of adjusting the firing rate of the cell were attempted - firstly, in the models that use it, adjusting c_0 to tune the fire rate, and secondly, lowering the number of "virtual fibres" in each of the inputs to decrease the input intensity and thus lower the cell's input levels. Both of these failed - Adjusting c_0 introduced instability to the simulation, and lowering the amount of virtual fibres either completely silenced the cell or didn't change the firing rate at all, with no room in between these extremes that could be used for tuning. Finding a way to control the cell firing rate beyond these two methods would thus be a useful endeavour in improving the robustness of our results.

Thirdly, since other types of HFS do exist (Bowden et al., 2012), it could be worthwhile to explore the effects of changing the HFS regime on the simulation's results and comparing to any relevant experiments.

Finally, since each cell type has different dynamics, it would be worthwhile expanding our investigation to cover other kinds of cell, or even simply other experiments on the dentate gyrus granule cell. A viable candidate would be the CA1 pyramidal cell which was also studied in the Abraham (2001) paper that contained the experiment this investigation was based on.

7. Conclusion

In this thesis we examined a background of the biological underpinnings of computational neuroscience as well differing ways to model both the neuron itself and also synaptic plasticity. Then we conducted a series of simulations on the Izhikevich neuron model to test the viability of several different models of synaptic plasticity when applied to a reproduction of an in-vivo experiment.

In our first round of simulations investigating pairing schemes, we found a dramatic variance in results caused by adjustment of a relatively minor detail in the model. While all of the pairing schemes were indeed capable of reproducing the overall form of the experiment for individual runs, when long-term averages are considered, the symmetric and reduced symmetric schemes, in this case, failed. This is because these schemes have a high chance of potentiating the unstimulated pathway sometimes more than 50% of the time - leading to an average of zero modification in some cases. This could well be caused by the symmetric schemes counting pairings that would not have an effect in a natural cell, since the same behaviour is not observed anywhere near as often in the presynaptic-centred and nearest spike schemes.

We found that the choice of synaptic plasticity model can have a profound effect on the accuracy of the results, with some models completely failing to reproduce any of the characteristic behaviours found in experiment, namely conventional STDP and the unmodified Pfister rule. The failure of conventional STDP shows that the theory alone is likely incapable of a complete description of synaptic plasticity, lending credence to the many variations on its implementation. However, the failure of the Pfister rule must be considered in light of the fact that it was not designed for hippocampal cells, nor for a system with random inputs such as ours.

A possible link between exhibition of heterosynaptic LTD and metaplasticity in the model was found, possibly explaining why there was such variance in the success of the models. This is shown by the reproduction of the heterosynaptic LTD only in models with metaplasticity - namely the Benuskova & Abraham rule, the modified Pfister rule, and the Clopath rule. This lends credence to the idea of metaplasticity in general, and specifically to BCM-like metaplasticity, especially when considered alongside STDP such as in Izhikevich & Desai (2003).

Our studies open the door to further investigation of the synaptic plasticity models examined here, perhaps on other experiments with other types of cell, or merely attempting the same exercise under a different neuron model. However, there are a great number of avenues for investigation in this field, more than can be considered in a single work - synaptic plasticity is still a theory in relative infancy and a great deal more research is needed for it in general.

References

Abraham, W. C., Mason-Parker, S. E., Bear, M. F., Webb, S., & Tate, W. P. (2001). Heterosynaptic metaplasticity in the hippocampus in vivo: a BCM-like modifiable threshold for LTP. Proceedings of the National Academy of Sciences, 98(19), 10924-10929.

Abraham, W. C. (2008). Metaplasticity: tuning synapses and networks for plasticity. Nature Reviews Neuroscience, 9(5), 387-387.

Abraham, W. C., & McNaughton, N. (1984). Differences in synaptic transmission between medial and lateral components of the perforant path. Brain research, 303(2), 251-260.

Ahmed, O. J., & Mehta, M. R. (2009). The hippocampal rate code: anatomy, physiology and theory. Trends in Neurosciences, 32(6), 329-338.

Bear, M. F., Connors, B. W., & Paradiso, M. A. (Eds.). (2007). Neuroscience (Vol. 2). Philadelphia: Lippincott Williams & Wilkins.

Benuskova, L., & Abraham, W. C. (2007). STDP rule endowed with the BCM sliding threshold accounts for hippocampal heterosynaptic plasticity. Journal of Computational Neuroscience, 22(2), 129-133.

Benuskova, L., & Kasabov, N. K. (2010). Computational neurogenetic modeling. New York: Springer.

Bienenstock, E. L., Cooper, L. N., & Munro, P. W. (1982). Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. The Journal of Neuroscience, 2(1), 32-48.

Carlson NA (1992). Foundations of Physiological Psychology. Needham Heights, Massachusetts: Simon & Schuster.

Clopath, C., Büsing, L., Vasilaki, E., & Gerstner, W. (2010). Connectivity reflects coding: a model of voltage-based STDP with homeostasis. Nature Neuroscience, 13(3), 344-352.

Cooper, L. N., & Bear, M. F. (2012). The BCM theory of synapse modification at 30: interaction of theory with experiment. Nature Reviews Neuroscience, 13(11), 798-810.

D Deshmukh, S. S., Yoganarasimha, D., Voicu, H., & Knierim, J. J. (2010). Theta modulation in the medial and the lateral entorhinal cortices. Journal of Neurophysiology, 104(2), 994-1006.

Edwards, D. H., Heitler, W. J., & Krasne, F. B. (1999). Fifty years of a command neuron: the neurobiology of escape behavior in the crayfish. Trends in Neurosciences, 22(4), 153-161.

Förster, E., Zhao, S., & Frotscher, M. (2006). Laminating the hippocampus. Nature Reviews Neuroscience, 7(4), 259-268.

Froemke, R. C., Tsay, I. A., Raad, M., Long, J. D., & Dan, Y. (2006). Contribution of individual spikes in burst-induced long-term synaptic modification. Journal of Neurophysiology, 95(3), 1620-1629.

Gjorgjieva, J., Clopath, C., Audet, J., & Pfister, J. P. (2011). A triplet spike-timing–dependent plasticity model generalizes the Bienenstock–Cooper–Munro rule to higher-order spatiotemporal correlations. Proceedings of the National Academy of Sciences, 108(48), 19383-19388.

Hananeia, N. (2012) Computational modelling of synaptic plasticity. Submitted as report for own 1st-year Master's project, covering reproduction of results in Izhikevich (2003) and Benuskova & Abraham (2007).

Hananeia, N., Benuskova L. (2014) Metaplasticity - Integral part of plasticity. NeuroEng 2014, 64.

Hebb, D. O. (1949) The Organization of Behavior: A Neuropsychological Theory. New York.

Hodgkin, A. L., Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. The Journal of Physiology 117 (4): 500–544.

Izhikevich, E. M., & Desai, N. S. (2003). Relating STDP to BCM. Neural computation, 15(7), 1511-1523.

Izhikevich, E. M. (2003). Simple model of spiking neurons. IEEE Transactions on Neural Networks, 14(6), 1569-1572.

Izhikevich, E. M. (2007). Dynamical Systems in Neuroscience. MIT press.

Kimura, A., & Pavlides, C. (2000). Long-term potentiation/depotentiation are accompanied by complex changes in spontaneous unit activity in the hippocampus. Journal of Neurophysiology, 84(4), 1894-1906.

Kirkwood, A., Rioult, M. G., & Bear, M. F. (1996). Experience-dependent modification of synaptic plasticity in visual cortex. Nature, 381(6582), 526-528.

Markram, H., Lübke, J., Frotscher, M., & Sakmann, B. (1997). Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. Science, 275(5297), 213-215.

Mayr, C. G., & Partzsch, J. (2010). Rate and pulse based plasticity governed by local synaptic state variables. Frontiers in Synaptic Neuroscience, 2.

Morest, D. K. (1971). Dendrodendritic synapses of cells that have axons: the fine structure of the Golgi type II cell in the medial geniculate body of the cat. Zeitschrift für Anatomie und Entwicklungsgeschichte, 133(2), 216-246.

Morrison, A., Diesmann, M., & Gerstner, W. (2008). Phenomenological models of synaptic plasticity based on spike timing. Biological Cybernetics, 98(6), 459-478.

Nelson ME (2005) Electrophysiological Models In Databasing the Brain: From Data to Knowledge. (S. Koslow and S. Subramaniam, eds.) Wiley, New York, pp. 285-301

Pfister, J. P., & Gerstner, W. (2006). Triplets of spikes in a model of spike timing-dependent plasticity. The Journal of Neuroscience, 26(38), 9673-9682.

Rose, G., Diamond, D., & Lynch, G. S. (1983). Dentate granule cells in the rat hippocampal formation have the behavioral characteristics of theta neurons. Brain Research, 266(1), 29-37.

Sterratt, D., Graham, B., Gillies, A., & Willshaw, D. (2011). Principles of Computational Modelling in Neuroscience. Cambridge University Press.

Taylor MM (1973). The problem of stimulus structure in the behavioural theory of perception. South African Journal of Psychology 3: 23–45.

Wöhrl, R., Von Haebler, D. and Heinemann, U. (2007). Low-frequency stimulation of the direct cortical input to area CA1 induces homosynaptic LTD and heterosynaptic LTP in the rat hippocampal–entorhinal cortex slice preparation. European Journal of Neuroscience, 25: 251–258.

Appendix 1: Presynaptic-centred Benuskova & Abraham implementation code

```
   #include<stdio.h>
   #include<stdlib.h>
   #include<math.h>
   #include<sys/time.h>

   typedef struct _neuron *neuron;

   neuron make_neuron_struct();
   void init_neuron_memory(neuron n);
   neuron make_neuron();
   void spontaneous_input(int time, neuron single_cell);
   void test_input(int time, neuron single_cell);
   void Med50(int time, neuron single_cell);
   void Lat50(int time, neuron single_cell);
   void cycle_cell(int time, neuron single_cell);
   void stdp_train_cell(int time, neuron single_cell);
   void bcm_threshold(int time, neuron single_cell);
   void test_sample(int time, neuron single_cell);
   void run_simulation(int time);
   void getTime(long *z);

   #define SIMULATION_PERIOD  25200000 //Number of time steps
   #define SPONT_PERIOD               3600000 //Period before HFS, spontaneous only
   #define MED_ONSET          5400000 //Medial HFS onset
   #define LAT_ONSET          19800000 //Lateral HFS onset
   #define BASELINE_START          2400000 //Start time to calculate baseline weight
   #define BASELINE_STOP           3600000 //Stop time to calculate baseline weight
   #define MIN30                      1800000 //30 minutes
   #define MIN10                      600000 //10 minutes
   #define MIN1               60000 //1 minute

   #define TIME_STEP          1 //Time step in ms
   #define SAMPLE_PERIOD              60000 //Time step between samples
   #define minutes                    60000.0
   #define TEST_PERIOD        10000 //Interval between test pulses

   #define TEST_INTENSITY             150 //number of input fibers engaged by the test
stimulus
   #define TRAIN_INTENSITY            250 //total number of input fibers
   #define AP                 55.0 //Action potential peak
   #define P_HFS                      0.4 //Probability of MPP firing during HFS
   #define P_theta                    0.008//Probability of pp synchronous spont firing
   #define P_pp               0.0001//Probability of pp asynchronous spontaneous
firing
   #define MAX                10000 //Maximum number of presynaptic spikes inbetween
postsynaptic
   #define LENGTH                     420  // length of simulation in minutes

   #define TAU                60000 //Integration period
   #define theta_M0           2000.0 //thetaM scaling parameter
   #define MAX_LOOP           30
   #define BOLTZMANN          0

   FILE *f_data, *f_voltage;
   /***********VARIABLES************/

   //Initialisation
   double MPP_WEIGHT_INIT = 0.03;
   double LPP_WEIGHT_INIT = 0.03;
   double firing_theta = 24.0; //Excitation threshold

   //STDP parameters
   double tau_ltp = 20;
   double tau_ltd = 100;
   double A0_ltp = 0.02;
   double A0_ltd = 0.01;
   double upper_cap = 5;
   double lower_cap = 0.01;

   //Cell Parameters
   double coeff1 = 0.04;
   double coeff2 = 5;
   double coeff3 = 140;
   double coeff4 = 1;

   double a = 0.02;
   double b = 0.2;
   double c = -69.0;
   double d = 2;

   int mpp_count = 0;
   int lpp_count = 0;

   //Other Vars
   int mpp_input, lpp_input, t1_gc, t2_gc, index_lpp, index_mpp;
   int t_lpp[MAX+1], t_mpp[MAX+1];
   int last_mpp_test, last_lpp_test, last_sample;
   int N_MPP, N_LPP;
   int loop, k, burst, train, impulse;
   double input, rnd, av_frequency, mpp_thetaM[MAX+1], lpp_thetaM[MAX+1];
   double av_mpp_weight[LENGTH + 2], av_lpp_weight[LENGTH + 2], av_thetaM[LENGTH + 2],
square_mpp_weight[LENGTH + 2], square_lpp_weight[LENGTH + 2], square_theta_M[LENGTH +
2];


   /**********STRUCTURES************/
   struct _neuron{
        double v;
        double u;
        int output;

        double mpp_weight;
        double lpp_weight;

     double baseline_mpp;
     double baseline_lpp;

        double theta_M;
        double mem;
        double pf;
        double boltzfactor;
   };

   /***********INITIALISATION*********/
   neuron make_neuron_struct(){
        neuron n;

        n = (neuron) malloc(sizeof(struct _neuron));

        n->v = c;
        n->u = b*c;
        n->output = 0;

        n->mpp_weight = MPP_WEIGHT_INIT;
        n->lpp_weight = LPP_WEIGHT_INIT;

        n->baseline_mpp = MPP_WEIGHT_INIT;
        n->baseline_lpp = LPP_WEIGHT_INIT;

        n->theta_M = 1.0;
        n->mem = 0.0;
        n->pf = 1.0;
        n->boltzfactor = 0.0;

        return n;
   }

   void init_neuron_memory(neuron n){
        n->boltzfactor = exp(-((double) TIME_STEP/ (double)TAU));
        if(BOLTZMANN == 0){
                n->pf = 1.0 / (1 - n->boltzfactor);
        }else{
                n->pf = 1.0;
        }
   }

   neuron make_neuron(){
        neuron n;

        n = make_neuron_struct();
        init_neuron_memory(n);

        return n;
   }


   /*******SIMULATION***********/

   void run_simulation(int t){
        int time, i;
        neuron single_cell;
        single_cell = make_neuron();

        for(i = 0; i<MAX; i++){
                t_mpp[i] =0;
                t_lpp[i] = 0;
                mpp_thetaM[i] = 0.0;
                lpp_thetaM[i] = 0.0;
        }

        rnd = 0.0;
        av_frequency = 0;
        last_mpp_test = 0;
        last_lpp_test = 0;
        t1_gc = 0;
        t2_gc = 0;
        index_lpp = 0;
        index_mpp = 0;
        last_sample = 0;
        burst = 1;
        train = 1;
        impulse = 1;
        mpp_input = 0;
        lpp_input = 0;
        k = 0;

        time = 0;

        while(time <= t){
          if(time>= BASELINE_START && time <= BASELINE_STOP){
            if(time == BASELINE_START){
              single_cell->baseline_mpp = single_cell->mpp_weight;
              single_cell->baseline_lpp = single_cell->lpp_weight;
            } else {
              single_cell->baseline_mpp = (single_cell->baseline_mpp + single_cell-
>mpp_weight)/2;
              single_cell->baseline_lpp = (single_cell->baseline_lpp + single_cell-
>lpp_weight)/2;
            }
          }

                if((time < MED_ONSET) || ((time > (MED_ONSET+MIN10)))){// && (time <
LAT_ONSET)) || (time > (LAT_ONSET+MIN10))){
                        spontaneous_input(time, single_cell);
                }

                if(((time > SPONT_PERIOD) && (time < MED_ONSET)) || ((time >
(MED_ONSET+MIN10)))){// && (time < LAT_ONSET)) ||(time > (LAT_ONSET+MIN10))){
                        test_input(time, single_cell);
                }

                if(time >= (MED_ONSET - 10000)&& time <= (MED_ONSET+10000)){
                fprintf(f_voltage, "%d %lf\n", time-MED_ONSET, single_cell->v);
                }

                if((time>=MED_ONSET) && (time <= (MED_ONSET+MIN10))){
                        Med50(time, single_cell);
                }

                if((time>=LAT_ONSET) && (time <= (LAT_ONSET+MIN10))){
                //      Lat50(time, single_cell);
                }

                if(time == 9000000){
                  if(single_cell->mpp_weight > single_cell->lpp_weight){
                    mpp_count = mpp_count + 1;
                  } else {
```

```
                lpp_count = lpp_count + 1;
              }
            }

            cycle_cell(time, single_cell);

            if((time == t2_gc) && (t1_gc > 0)){
              stdp_train_cell(time, single_cell);
            }

            bcm_threshold(time, single_cell);
            test_sample(time, single_cell);
            time  = time + TIME_STEP;
     }

     av_frequency = av_frequency / (double)SIMULATION_PERIOD*1000;
     printf("Average output frequency\t");
     printf("%lf Hz\n", av_frequency);
}

/************SPONTANEOUS INPUT*************/
void spontaneous_input(int time, neuron n){

     mpp_input = 0;
     lpp_input = 0;

     //correlated inputs
     rnd = (double)(random())/(double)RAND_MAX;
     if(P_theta > rnd){
            mpp_input = 1;
            N_MPP = TRAIN_INTENSITY;
            lpp_input = 1;
            N_LPP = TRAIN_INTENSITY;

            if(t1_gc > 0){
                   if((index_mpp <MAX)&&(index_lpp<MAX)&&(time>t2_gc)){
                          t_mpp[index_mpp] = time;
                          t_lpp[index_lpp] = time;
                          mpp_thetaM[index_mpp] = n->theta_M;
                          lpp_thetaM[index_lpp] = n->theta_M;
                          index_mpp = index_mpp + 1;
                          index_lpp = index_lpp + 1;
                   }
            }
     }

     //Asynchronous spontaneous LPP input
     rnd = (double)(random())/(double)RAND_MAX;
     if((lpp_input == 0) && (P_pp > rnd)){
            lpp_input = 1;
            N_LPP = TRAIN_INTENSITY;

            if(t1_gc > 0){
                   if((index_lpp <MAX) && (time>t2_gc)){
                          t_lpp[index_lpp] = time;
                          lpp_thetaM[index_lpp] = n->theta_M;
                          index_lpp = index_lpp + 1;
                   }
            }
     }


     //Asynchronous spontaneous MPP input
     rnd = (double)(random())/(double)RAND_MAX;
     if((mpp_input == 0) && (P_pp > rnd)){
            mpp_input = 1;
            N_MPP = TRAIN_INTENSITY;

            if(t1_gc > 0 ){
                   if((index_mpp < MAX) && (time > t2_gc)){
                          t_mpp[index_mpp] = time;
                          mpp_thetaM[index_mpp] = n->theta_M;
                          index_mpp = index_mpp + 1;
                   }
            }
     }
}
```

```
/******************TEST INPUT******************/
void test_input(int time, neuron n){

     if((time >= (SPONT_PERIOD+10000)) && (time - last_mpp_test >= 2*TEST_PERIOD)){
            last_mpp_test = time;
            if(mpp_input==0){
                   mpp_input = 1;
                   N_MPP = TEST_INTENSITY;
                   if(t1_gc>0){
                          if((index_mpp<MAX) && (time > t2_gc)){
                                 t_mpp[index_mpp] = time;
                                 mpp_thetaM[index_mpp] = n->theta_M;
                                 index_mpp = index_mpp + 1;
                          }
                   }
            }
     }

     if((time >= (SPONT_PERIOD+20000)) && (time - last_lpp_test >= 2*TEST_PERIOD)){
            last_lpp_test = time;
            if(lpp_input == 0){
                   lpp_input = 1;
                   N_LPP = TEST_INTENSITY;

                   if(t1_gc > 0){
                          if((index_lpp < MAX) && (time > t2_gc)){
                                 t_lpp[index_lpp] = time;
                                 lpp_thetaM[index_lpp] = n->theta_M;
                                 index_lpp = index_lpp + 1;
                          }
                   }
            }
     }
}



/***************MPP HFS******************/
void Med50(int time, neuron n){

     mpp_input = 0;
     lpp_input = 0;

     //10 bursts of 5 trains
     if((time >= (MED_ONSET + (burst-1)*MIN1 + (train-1)*1000 + (train-1)*25)) &&
(time <= (MED_ONSET + (burst-1)*MIN1 + (train-1)*1000 + (train-1)*25 + 25))){

            rnd = (double)(random())/(double)RAND_MAX;
            if(P_HFS > rnd){
                   mpp_input = 1;
                   N_MPP = TRAIN_INTENSITY;

                   if(t1_gc >0){
                          if((index_mpp < MAX) && (time >t2_gc)){
                                 t_mpp[index_mpp] = time;
                                 mpp_thetaM[index_mpp] = n->theta_M;
                                 index_mpp = index_mpp + 1;
                          }
                   }
            }

            if(impulse == 26){
                   train = train + 1;
                   impulse = 1;
            }
            if(train == 6){
                   burst = burst + 1;
                   train = 1;
            }
            impulse = impulse + 1;

            //Spontaneous LPP input
            rnd = (double)(random())/(double)RAND_MAX;
            if((P_theta + P_pp) > rnd){
                   lpp_input = 1;
```

```
                    N_LPP = TRAIN_INTENSITY;

                    if(t1_gc > 0){
                            if((index_lpp < MAX) && (time > t2_gc)){
                                    t_lpp[index_lpp] = time;
                                    lpp_thetaM[index_lpp] = n-> theta_M;
                                    index_lpp = index_lpp + 1;
                            }
                    }
            }
    }
    else{
            //Inter-burst spontaneous LPP input
            rnd = (double)(random())/(double)RAND_MAX;
            if((P_theta+P_pp) > rnd) {
                            lpp_input = 1;
                    N_LPP = TRAIN_INTENSITY;
                            if(t1_gc > 0){
                            if((index_lpp < MAX) && (time > t2_gc)){
                                    t_lpp[index_lpp] = time;
                                    lpp_thetaM[index_lpp] = n->theta_M;
                                    index_lpp = index_lpp + 1;
                            }
                    }
                    }
            //Inter-burst spontaneous MPP input
            rnd = (double)(random())/(double)RAND_MAX;
            if((P_theta+P_pp) > rnd) {
                            mpp_input = 1;
                    N_MPP = TRAIN_INTENSITY;
                            if(t1_gc > 0){
                            if((index_mpp < MAX) && (time > t2_gc)){
                                    t_mpp[index_mpp] = time;
                                    mpp_thetaM[index_mpp] = n->theta_M;
                                    index_mpp = index_mpp + 1;
                            }
                    }
                    }
                    test_input(time, n);
    }

    if(time == (MED_ONSET + MIN10)){
            last_mpp_test = MED_ONSET + MIN10 - 10000;
            last_lpp_test = MED_ONSET + MIN10;
            burst = 1;
            train = 1;
            impulse = 1;
    }
}


/***************LPP HFS******************/
void Lat50(int time, neuron n){

    mpp_input = 0; lpp_input = 0;

    //10 bursts of 5 trains
    if((time >= (LAT_ONSET + (burst-1)*MIN1 + (train-1)*1000 + (train-1)*25)) &&
    (time <= (LAT_ONSET + (burst-1)*MIN1 + (train-1)*1000 + (train-1)*25 + 25))){

            rnd = (double)(random())/(double)RAND_MAX;
            if(P_HFS > rnd){
                    lpp_input = 1;
                    N_LPP = TRAIN_INTENSITY;

                    if(t1_gc > 0){
                            if((index_lpp <MAX)&&(time >t2_gc)){
                                    t_lpp[index_lpp] = time;
                                    lpp_thetaM[index_lpp] = n->theta_M;
                                    index_lpp = index_lpp + 1;
                            }
                    }
            }

            if(impulse == 26){
                    train = train+1;
                    impulse = 1;
```

```
                }
                if(train == 6){
                        burst = burst + 1;
                        train = 1;
                }
                impulse = impulse + 1;

                //Spontaneous MPP input
                rnd = (double)(random())/(double)RAND_MAX;
                if((P_theta + P_pp) > rnd){
                        mpp_input = 1;
                        N_MPP = TRAIN_INTENSITY;

                        if(t1_gc > 0){
                                if((index_mpp < MAX) && (time > t2_gc)){
                                        t_mpp[index_mpp] = time;
                                        mpp_thetaM[index_mpp] = n->theta_M;
                                        index_mpp = index_mpp + 1;
                                }
                        }
                }
        }
        else{
                //Inter-burst spontaneous LPP input
                rnd = (double)(random())/(double)RAND_MAX;
                if((P_theta+P_pp) > rnd){
                                lpp_input = 1;
                        N_LPP = TRAIN_INTENSITY;
                        if(t1_gc > 0){
                                if((index_lpp < MAX) && (time > t2_gc)){
                                        t_lpp[index_lpp] = time;
                                        lpp_thetaM[index_lpp] = n->theta_M;
                                        index_lpp = index_lpp + 1;
                                }
                        }
               }
                //Inter-burst spontaneous MPP input
                rnd = (double)(random())/(double)RAND_MAX;
                if((P_theta+P_pp) > rnd){
                        mpp_input = 1;
                        N_MPP = TRAIN_INTENSITY;
                        if(t1_gc > 0){
                                if((index_mpp < MAX) && (time > t2_gc)){
                                        t_mpp[index_mpp] = time;
                                        mpp_thetaM[index_mpp] = n->theta_M;
                                        index_mpp = index_mpp + 1;
                                }
                        }
               }
             test_input(time, n);
        }
        if(time==(LAT_ONSET+MIN10)){
                last_mpp_test = LAT_ONSET+MIN10-10000;
                last_lpp_test = LAT_ONSET + MIN10;
        }
   }


   /*********Update neuron*************/

   void cycle_cell(int time, neuron n){

        n->output = 0;

        if(n->v >= AP){
                n->v = c;
                n->u = n->u + d;
        }else{
                input = mpp_input*n->mpp_weight*N_MPP + lpp_input*n->lpp_weight*N_LPP;
                n->v = n->v + coeff1*(n->v*n->v) + coeff2*n->v + coeff3 - coeff4*n->u +
input;

                n->u = n->u + a*(b*n->v - n->u);

                if(n->v >= firing_theta){
                        n->v = AP;
                        n->output = 1;
```

```
                    av_frequency = av_frequency + 1;

                    if(t1_gc == 0){
                           t1_gc = time;
                           t2_gc = time;
                    } else {
                           t1_gc = t2_gc;
                           t2_gc = time;
                    }
            }
     } // end of else
}

/**************STDP**************/
void stdp_train_cell(int time, neuron n){
     double ltp, ltd, A_ltp, A_ltd;
     int j;

     index_lpp = 0;
     index_mpp = 0;

     while(index_mpp < MAX){
            if(t_mpp[index_mpp] == 0){
                    index_mpp = MAX;
            }else{
                    if(mpp_thetaM[index_mpp] > 0.01 &&
                       mpp_thetaM[index_mpp] < 100){
                      A_ltp = A0_ltp*(1/mpp_thetaM[index_mpp]);
                      A_ltd = A0_ltd*(mpp_thetaM[index_mpp]);
                           }
                    if(t2_gc > t_mpp[index_mpp]){
                           ltp = A_ltp * exp(-(t2_gc - t_mpp[index_mpp])/tau_ltp);
                    }
                    if(t1_gc < t_mpp[index_mpp]){
                           ltd = A_ltd * exp((t1_gc - t_mpp[index_mpp])/tau_ltd);
                    }

                    n->mpp_weight = n->mpp_weight*(1 + ltp-ltd);

                    //Caps. These are bad.
                    if(n->mpp_weight < lower_cap){
                           n->mpp_weight = lower_cap;
                    }
                    if(n->mpp_weight > upper_cap){
                           n->mpp_weight = upper_cap;
                    }
                    index_mpp = index_mpp + 1;
            }
     }

     while(index_lpp < MAX){
            if(t_lpp[index_lpp] == 0){
                    index_lpp = MAX;
            }else{
                      if(lpp_thetaM[index_lpp] > 0.001 &&
                         lpp_thetaM[index_lpp]<1000){
                 A_ltp = A0_ltp*(1/lpp_thetaM[index_lpp]);
                 A_ltd = A0_ltd*(lpp_thetaM[index_lpp]);
                      }
               if(t2_gc > t_lpp[index_lpp]){
                           ltp = A_ltp * exp(-(t2_gc - t_lpp[index_lpp])/tau_ltp);
                    }
                    if(t1_gc < t_lpp[index_lpp]){
                           ltd = A_ltd * exp((t1_gc - t_lpp[index_lpp])/tau_ltd);
                    }

                    n->lpp_weight = n->lpp_weight * (1+ltp-ltd);


                    //Caps. These are bad.
                    if(n->lpp_weight < lower_cap){
                           n->lpp_weight = lower_cap;
                    }
                    if(n->lpp_weight > upper_cap){
                      n->lpp_weight = upper_cap;
                    }
                    index_lpp = index_lpp + 1;
               }
        }

        for(j = 0; j<MAX; j++){
               t_mpp[j] = 0;
               t_lpp[j] = 0;
               mpp_thetaM[j] = 0.0;
               lpp_thetaM[j] = 0.0;
        }

        index_mpp = index_lpp = 0;
        t1_gc = t2_gc;
   }

   /****************BCM*******************/
   void bcm_threshold(int time, neuron n){
        n->mem = (n->mem*(n->pf - 1.0) + (n->output*n->output))/n->pf;
        n->pf = n->pf*n->boltzfactor + 1.0;
        n->theta_M = theta_M0*n->mem;
   }



   /*************Testing*************/
   void test_sample(int time, neuron n){
        double percent_mpp_w, percent_lpp_w;

        if((time ==0) || (time - last_sample >= SAMPLE_PERIOD)){
               last_sample = time;
               percent_mpp_w = 100*(n->mpp_weight-n->baseline_mpp)/n->baseline_mpp;
               percent_lpp_w = 100*(n->lpp_weight-n->baseline_lpp)/n->baseline_lpp;

               av_mpp_weight[k] = av_mpp_weight[k] + n->mpp_weight;
               square_mpp_weight[k] = square_mpp_weight[k] + n->mpp_weight*n-
>mpp_weight; //for S.D.
               av_lpp_weight[k] = av_lpp_weight[k] + n->lpp_weight;
               square_lpp_weight[k] = square_lpp_weight[k] + n->lpp_weight*n-
>lpp_weight;  //for S.D.
               av_thetaM[k] = av_thetaM[k] + n->theta_M;
               square_theta_M[k] = square_theta_M[k] + n->theta_M*n->theta_M;
               k = k+1;
        }
   }

   /************** random seed *****/
   void getTime(long *z){
        struct timeval tp;
        struct timezone tzp;

        gettimeofday(&tp, &tzp);
        *z = tp.tv_usec;
   }


   /*********MAIN**********/
   int main(void){
        int i;
        long z;
        double var_mpp;
        double var_lpp;
        double var_theta;
        getTime(&z);
        srandom(z);

        for(i = 0; i<= LENGTH; i++){
               av_mpp_weight[i] = 0.0;
               av_lpp_weight[i] = 0.0;
               av_thetaM[i] = 0.0;
        }

        f_voltage = fopen("voltage.dat", "w");

        for(i = 0; i < MAX_LOOP; i++){
               run_simulation(SIMULATION_PERIOD);
        }

        f_data = fopen("data.dat", "w");
        for(i = 0; i<= LENGTH; i  = i + 1){
          var_mpp = (square_mpp_weight[i]/MAX_LOOP) -
(av_mpp_weight[i]/MAX_LOOP)*(av_mpp_weight[i]/MAX_LOOP);
          var_lpp = (square_lpp_weight[i]/MAX_LOOP) -
(av_lpp_weight[i]/MAX_LOOP)*(av_lpp_weight[i]/MAX_LOOP);
          var_theta = (square_theta_M[i]/MAX_LOOP) -
(av_thetaM[i]/MAX_LOOP)*(av_thetaM[i]/MAX_LOOP);

          fprintf(f_data, "%d\t%6.3lf\t%6.3lf\t%6.3lf\t%6.3lf\t%6.3lf\t%6.3lf\n", i-60,
av_mpp_weight[i]/MAX_LOOP,sqrt(var_mpp), av_lpp_weight[i]/MAX_LOOP, sqrt(var_lpp),
av_thetaM[i]/MAX_LOOP, sqrt(var_theta));
        }
        printf("MPP potentiated: %d times\nLPP potentiated %d times\n", mpp_count,
lpp_count);
        fclose(f_data);
        fclose(f_voltage);

        return 1;
   }
```

Appendix 2: Synaptic Plasticity routines for other STDP models

Note: This is not complete code, but is extracts taken from the relevant portions of the programs for illustration. Compare with Appendix 2 for details on how these would be implemented; these are very much "drop in" replacements.

Conventional STDP:

```
A_ltp = A0_ltp;
A_ltd = A0_ltd;
if(t2_gc > t_lpp[index_lpp]){
     ltp = A_ltp * exp(-(t2_gc - t_lpp[index_lpp])/tau_ltp);
}
if(t1_gc < t_lpp[index_lpp]){
     ltd = A_ltd * exp((t1_gc - t_lpp[index_lpp])/tau_ltd);
}

n->lpp_weight = n->lpp_weight * (1+ltp-ltd) - decay*n->lpp_weight;
```

Froemke:

```
e_t1 = 1 - exp(-(t1_gc - t3_gc)/sup_post);
e_t2 = 1 - exp(-(t2_gc - t1_gc)/sup_post);


e_pre = 1 - exp(-(t_mpp[index_mpp] - t_mpp[index_mpp - 1])/sup_pre);

A_ltp = A0_ltp;
A_ltd = A0_ltd;
if(t2_gc > t_mpp[index_mpp]){
     ltp =e_t2*e_pre*A_ltp * exp(-(t2_gc - t_mpp[index_mpp])/tau_ltp);
}
if(t1_gc < t_mpp[index_mpp]){
     ltd = e_t1*e_pre*A_ltd*exp((t1_gc - t_mpp[index_mpp])/tau_ltd);
}

n->mpp_weight = n->mpp_weight*(1 + ltp-ltd) - decay*n->mpp_weight;
```

Pfister:

```
//At each MPP spike
r1_mpp = r1_mpp + 1;
r2_mpp = r2_mpp + 1;

//At each LPP spike
r1_lpp = r1_lpp + 1;
r2_lpp = r2_lpp + 1;

//At each postsynaptic spike
o1 = o1 + 1;
o2 = o2 + 1;

//This is done at each cycle of the cell
r1_mpp = r1_mpp - r1_mpp/tau_ltp;
r1_lpp = r1_lpp - r1_lpp/tau_ltp;
r2_mpp = r2_mpp - r2_mpp/tau_x;
r2_lpp = r2_lpp - r2_lpp/tau_x;

o1 = o1 - o1/tau_ltd;
o2 = o2 - o2/tau_y;

//Main STDP routine
if(t2_gc > t_mpp[index_mpp]){
     ltp = r1_mpp*(A2_ltp + A3_ltp*o2*(t2_gc - t_mpp[index_mpp]));
}
if(t1_gc < t_mpp[index_mpp]){
     ltd = o1*(A2_ltd + A3_ltd*r2_mpp*(t1_gc - t_mpp[index_mpp]));
}

n->mpp_weight = n-> mpp_weight *(1 +  ltp - ltd) - n->mpp_weight*decay;
```

Metaplastic Pfister:

```
A2_ltp_eff = A2_ltp*(1/lpp_thetaM[index_lpp]);
A3_ltp_eff = A3_ltp*(1/lpp_thetaM[index_lpp]);
A2_ltd_eff = A2_ltd*(lpp_thetaM[index_lpp]);
A3_ltd_eff = A3_ltd*(lpp_thetaM[index_lpp]);

if(t2_gc > t_lpp[index_lpp]){
     ltp = r1*(A2_ltp_eff + A3_ltp_eff*o2*(t2_gc - t_lpp[index_lpp]));
     }
if(t1_gc < t_lpp[index_lpp]){
     ltd = o1*(A2_ltd_eff + A3_ltd_eff*r2*(t1_gc - t_lpp[index_lpp]));
     }

n->lpp_weight = n->lpp_weight*(1+ ltp - ltd) - n->lpp_weight*decay;
```

Clopath:

```
//At each MPP spike
X_mpp++;

//At each LPP spike
X_lpp++;

//Main STDP procedure
void update_cell(neuron n){
  double A_ltp, A_ltd;
  A_ltp = A0_ltp; // no meta
  A_ltd = A0_ltd*n->theta_M; //Updated according to v. dependence

  u_minus = u_minus + (1/tau_ltd)*(n->v - u_minus);
  u_plus = u_plus + (1/tau_ltp)*(n->v - u_plus);
  x_lpp = x_lpp + (1/tau_x)*(X_lpp - x_lpp);
  x_mpp = x_mpp + (1/tau_x)*(X_mpp - x_mpp);

  //LTP contribs
  if(n->v > theta_plus && u_plus > theta_minus){
    if(n->lpp_weight < upper_cap){
      w_lpp_plus = A_ltp*x_lpp*(n->v - theta_plus)*(u_plus - theta_minus);
    }
    if(n->mpp_weight < upper_cap){
      w_mpp_plus = A_ltp*x_mpp*(n->v - theta_plus)*(u_plus - theta_minus);
    }
  }

  //LTD contribs
  if(u_minus > theta_minus){
    if(n->lpp_weight > lower_cap){
      w_lpp_minus = A_ltd*X_lpp*(u_minus - theta_minus);
    }
    if(n->mpp_weight > lower_cap){
      w_mpp_minus = A_ltd*X_mpp*(u_minus - theta_minus);
    }
  }

  //Update weights
  n->mpp_weight = n->mpp_weight + w_mpp_plus - w_mpp_minus;
  n->lpp_weight = n->lpp_weight + w_lpp_plus - w_lpp_minus;

  n->mpp_weight = n->mpp_weight - decay*n->mpp_weight;
  n->lpp_weight = n->lpp_weight - decay*n->lpp_weight;

  w_mpp_minus = 0;
  w_lpp_minus = 0;
  w_mpp_plus = 0;
  w_lpp_plus = 0;
}
```

Appendix 3: List of Acronyms

AMPA: α-Amino-3-hydroxy-5-methyl-4-isoxazolepropionic acid.

BCM: Bienenstock - Cooper - Munro, used specifically in reference to their synaptic plasticity model.

CA: Cornu ammonis, i.e. Ammon's horn.

HFS: High-frequency stimulation.

LPP: Lateral perforant path.

LTD: Long-term depression.

LTP: Long-term potentiation.

NMDA: *N*-Methyl-D-aspartic acid

MPP: Medial perforant path.

STDP: Spike timing dependent plasticity.