



THE UNIVERSITY OF  
**WAIKATO**  
*Te Whare Wānanga o Waikato*

Research Commons

<http://waikato.researchgateway.ac.nz/>

## Research Commons at the University of Waikato

### Copyright Statement:

The digital copy of this thesis is protected by the Copyright Act 1994 (New Zealand).

The thesis may be consulted by you, provided you comply with the provisions of the Act and the following conditions of use:

- Any use you make of these documents or images must be for research or private study purposes only, and you may not make them available to any other person.
- Authors control the copyright of their thesis. You will recognise the author's right to be identified as the author of the thesis, and due acknowledgement will be made to the author where appropriate.
- You will obtain the author's permission before publishing any material from the thesis.

# Theoretical Investigation of NMDA Effect on the Cerebral Cortex

A thesis submitted in partial fulfilment  
of the requirements for the degree of

**Master of Science**

in Physics

University of Waikato

by

**Waleed Hamdan Khalfan AL Saidi**



THE UNIVERSITY OF  
**WAIKATO**  
*Te Whare Wānanga o Waikato*



# Dedication

I dedicate this thesis:

To my mother Shaika

To my father Hamdan

To my wife Fadeela

To my sons Khalid and AL Rabee

To my daughters AL Shaima, AL Balja and AL Nouwar

To my other children who are not yet with us

Thank you for all love, support, patience and inspiration.



# Acknowledgements

I would like to acknowledge with considerable gratitude all those who have helped me with my research. At first, a special thanks to Moira Steyn-Ross and Alistair Steyn-Ross for your supervision and helpful advice, working with you has been an honour and a privilege. Thanks Moira again for your guidance and encouragement. Alistair I have to admit that I was lucky to be supervised by you, your precise corrections in writing my thesis would not be forgettable. Thank you Dr Michael Cree for assistance with  $\text{\LaTeX}$  whenever it was required. Hono Kayano I am glad to have a friend like you, you were always there when I need any assistance in MATLAB. Hood my close friend your great help and support has a great impact on me. Thank you my Mum and Dad for your encouragement and prayers. Last but far from least, my beloved wife and children thanks for your love and being patient when I was studying overseas.



# Abstract

This thesis examines the dynamical behaviour of incorporating NMDA (an excitatory neurotransmitter) for the electrodynamic model of the cerebral cortex. The model used is the mean-field model developed by Steyn-Ross *et al.* (2005) which describes the behaviour of the cortex in terms of parameters averaged over spatially localised populations. The behaviour of the model is determined by the four control parameters: inhibitory effect  $\lambda_i$ , subcortical drive  $s$ , and NMDA neurotransmitter effect set by an excitatory factor  $\lambda_e$  and the magnesium concentration  $C$ . Adopting this model could give a better understanding of the cortex functionality and the anaesthetic mechanism.

The model predicts that there are either one or three stationary states available to the cortex. We identify two of these with “highly activated” state and a “quiescent” state and focus on the transition between the two. Theoretical stability predictions (eigenvalue analysis) verified by a numerical simulation show that the system is unstable between the two Hopf bifurcations. In addition, in the stable region the steady states remains stable under a small perturbation, while in the unstable region either a transition between states or a limit cycle (oscillation) occurs depending on the position of the steady state.



# Contents

<b>Dedication</b>	<b>iii</b>
<b>Acknowledgements</b>	<b>v</b>
<b>Abstract</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Thesis overview . . . . .	2
1.2 Original work . . . . .	2
<b>2 Neurophysiology</b>	<b>3</b>
2.1 Brain and neuron . . . . .	3
2.2 Electrical signals in neurons . . . . .	4
2.3 Neuron communication . . . . .	5
2.3.1 Excitatory and inhibitory post-synaptic potential . . . . .	8
2.4 Anaesthesia . . . . .	9
2.5 Anesthetic mechanism . . . . .	10
2.5.1 Indirect effects of anaesthetics on the NMDA synapse . . . . .	13
<b>3 Modelling the cerebral cortex</b>	<b>15</b>
3.1 Mean-field model . . . . .	15
3.2 Brief history of mean-field models . . . . .	15
3.3 Neural equivalent circuit . . . . .	17
3.4 Hodgkin-Huxley model . . . . .	19
3.4.1 Resting voltage . . . . .	19
3.4.2 Action potential . . . . .	19
3.5 Tuckwell subthreshold neuron . . . . .	20
3.6 The Liley neuron . . . . .	23
3.7 Mapping from Tuckwell $\rightarrow$ Liley . . . . .	24
3.8 Steyn-Ross model . . . . .	26
3.8.1 Integral equations for the macrocolumn . . . . .	26
3.8.2 Differential equations for the macrocolumn . . . . .	30
3.8.3 Synaptic flux densities . . . . .	31
3.8.4 Subcortical flux: White-noise driving . . . . .	33
3.8.5 Summary of stochastic DEs for cortex . . . . .	33
<b>4 NMDA-modulated mean-field cortical model</b>	<b>35</b>
4.1 Modelling NMDA . . . . .	35
4.2 Stationary states . . . . .	36
4.3 Numerical solution . . . . .	38
4.3.1 Isocline method . . . . .	39

4.3.2	Root finding . . . . .	39
4.4	Distribution of roots . . . . .	41
4.4.1	Variations in inhibitory neurotransmitter, $\lambda_i$ . . . . .	41
4.4.2	Variations in subcortical driving $s$ . . . . .	42
4.4.3	Variations in excitatory neurotransmitter $\lambda_e$ . . . . .	42
4.4.4	Variations in excitatory neurotransmitter $C$ . . . . .	43
4.5	Significance of the steady-states . . . . .	44
4.6	Stability analysis . . . . .	44
4.6.1	Linear stability analysis: Theory . . . . .	44
4.6.2	Linear stability analysis for cortical model . . . . .	47
4.6.3	Elements of the $8 \times 8$ Jacobian matrix . . . . .	49
4.7	Stability predictions . . . . .	51
4.7.1	Stability changes with anesthetic effect $\lambda_i$ . . . . .	52
4.7.2	Stability changes with subcortical driving, $s$ . . . . .	53
4.7.3	Stability changes with NMDA effect, $\lambda_e$ and $C$ . . . . .	53
4.8	Predicted effect of NMDA on cortical stability . . . . .	53
<b>5</b>	<b>Numerical simulation</b> . . . . .	<b>59</b>
5.1	Discrete approximation for white noise . . . . .	59
5.2	Difference equations . . . . .	60
5.3	Verification of macrocolumn steady states . . . . .	61
5.4	Validation of the stable branches . . . . .	61
5.5	Examination of the Hopf bifurcation . . . . .	62
5.6	Simulation on the unstable region . . . . .	63
<b>6</b>	<b>Conclusions and further work</b> . . . . .	<b>69</b>
6.1	Overview . . . . .	69
6.1.1	Equilibrium solutions . . . . .	69
6.1.2	Theoretical stability predictions of the steady states . . . . .	70
6.1.3	Numerical simulation of the steady states . . . . .	70
6.1.4	Further work . . . . .	71
<b>A</b>	<b>Matlab code</b> . . . . .	<b>73</b>
A.1	Sigmoid functions and their derivatives . . . . .	73
A.2	Normalised conductance for NMDA . . . . .	75
A.3	Global initial constants of the cortical model . . . . .	75
A.4	NMDA steady states finder . . . . .	77
A.5	Locus of equilibrium states . . . . .	94
A.6	Eigenvalues of the cortical model . . . . .	99
A.7	System stability and Hopf . . . . .	110
A.8	Euler numerical simulation of the cortical model . . . . .	114
	<b>References</b> . . . . .	<b>121</b>

# List of Figures

2.1	The neuron's structure . . . . .	4
2.2	Graded potential . . . . .	6
2.3	Action potential (AP) or impulse . . . . .	7
2.4	Signal transmission at a chemical synapse . . . . .	8
2.5	A GABAergic model for anaesthetic action . . . . .	11
2.6	A block-diagram of the GABA receptor . . . . .	12
2.7	Classification of the ionotropic glutamate receptors . . . . .	13
2.8	A magnesium ion blocks the channel within the NMDA receptor (in-activated configuration) . . . . .	13
2.9	The Flohr model of the NMDA synapse as a target for anaesthetics . . . . .	14
3.1	Neural equivalent circuit . . . . .	18
3.2	Neural equivalent circuit with variable conductances. . . . .	20
3.3	Prediction from the Hodgkin and Huxley model for the formation of an action potential . . . . .	21
3.4	The $K^+$ , $Cl^-$ and $Na^+$ batteries and conductances replaced by $V_{rest}$ in series with $g_{rest}$ . . . . .	22
3.5	Tuckwell subthreshold neuron . . . . .	23
3.6	Schematic wiring diagram for a cortical macrocolumn. . . . .	24
3.7	Excitatory and inhibitory neural populations for Liley's generalisation of Tuckwell subthreshold neuron . . . . .	26
3.8	Tuckwell $\rightarrow$ Liley neural model with local feedbacks coupling the two populations together . . . . .	27
3.9	Full circuit diagram of Tuckwell $\rightarrow$ Liley neural model. . . . .	28
3.10	Sigmoidal function relating firing rate to average soma potential. . . . .	32
4.1	Voltage dependence of NMDA conductance $g$ . . . . .	36
4.2	Isocline showing 1 stationary state and the curve is going down . . . . .	40
4.3	Isocline showing 1 stationary state and the curve is going up . . . . .	40
4.4	Isocline showing 3 stationary states . . . . .	41
4.5	A model predictions for the stationary states for $V_{e,i}$ and $Q_{e,i}$ as a function of $\lambda_i$ . . . . .	42
4.6	A prediction for the stationary states for $V_{e,i}$ and $Q_{e,i}$ as a function of sub-cortical driving $s$ . . . . .	43
4.7	Locus for the stationary states for $V_{e,i}$ and $Q_{e,i}$ as a function of $\lambda_e$ . . . . .	43
4.8	A model predictions for the stationary states for $V_{e,i}$ and $Q_{e,i}$ as a function of magnesium concentration $C$ . . . . .	44
4.9	The regions in S-bend of the cortical steady states . . . . .	45
4.10	S-bend and dominant eigenvalues prediction of the cortical model stability for $\lambda_i$ . . . . .	54

---

4.11 S-bend and dominant dominant eigenvalues of the cortical model for the subcortical driving $s$ . . . . .	55
4.12 S-bend and dominant dominant eigenvalues of the cortical model for the NMDA effect $\lambda_e$ . . . . .	56
4.13 S-bend and dominant dominant eigenvalues of the cortical model for the NMDA effect $C$ . . . . .	57
4.14 The NMDA effect on the macrocolumn stability, varying $\lambda_i$ . . . . .	58
4.15 The NMDA effect on the macrocolumn stability, varying $s$ . . . . .	58
5.1 Simulation on the stable region . . . . .	62
5.2 Simulating the Hopf points . . . . .	63
5.3 Transition between states . . . . .	64
5.4 Limit cycle for the subcortical drive, $s$ . . . . .	65
5.5 Mid-branch point leads to transition and limit cycles . . . . .	66
5.6 Mid-branch divergence result in a transition between states . . . . .	67
5.7 Mid-branch divergence result in a limit cycle . . . . .	67

# List of Tables

3.1	Typical ion concentration (in mmol/L) and Nernst potentials (mV) for a resting neuron. . . . .	18
3.2	Symbol definitions and model constants for the Liley equations. . . . .	27



# Chapter 1

## Introduction

In 1997 Dr Moira Steyn-Ross (a theoretical physicist at Waikato University) was given a new research by Dr Jamie Sleigh (a senior anaesthetist at Waikato Hospital). Dr Sleigh noticed that the anaesthetised patients went from consciousness to unconsciousness suddenly rather than gradually as the anaesthetic concentration is increased. Since then the Waikato cortical modelling group (Dr Moira Steyn-Ross, Dr Alistair Steyn-Ross, Dr Jamie Sleigh, and Dr Marcus Wilson) started developing a mean-field model to have a better understanding of the anaesthetic effects on the human cerebral cortex. Their main aim was to theoretically describe the behaviour of the electrical activity in the cortex of a person undergoing anaesthesia. They predict that the cortex can be either “highly activated” or “quiescent”. As the anaesthetic concentration is increased, a sudden phase transition occurs at a critical concentration, from the conscious state to unconscious state.

The awake  $\longrightarrow$  asleep switchover can be understood as being analogous to the physical phase transition that happens as water freezes to form ice. There is no intermediate state between liquid water and solid ice, instead the transition occurs abruptly.

Measuring electroencephalogram (EEG) signals from the brain determines whether a patient is conscious or unconscious. However, consciousness-monitoring devices can be inaccurate, in some cases an anaesthetized patient woke up during the operation while he was reported unconscious. But that patient might not be able to tell the surgeon that he is awake because the drugs used to induce unconsciousness have the effect of stopping muscles “flinching”, paralysing the body during the surgery. Also, anaesthetics are known to cause loss of memory around the operation time. Therefore, an awake patient most likely would forget this traumatic situation before reporting it.

An accurate cortical model (such as the Waikato model) could be used to understand how anaesthetics work. Ideally, such a model could identify when a patient is about to wake up during the operation, alerting theatre staff that more anaesthetic should be administered to keep the patient unconscious. Hence, more anaesthetics might be given to him to stay unconscious until the operation is finished. This might

improve patient safety and reduce the nightmare risk of inadequate anaesthesia.

## 1.1 Thesis overview

The aim of this thesis is to examine the dynamical behaviour of incorporating NMDA (an excitatory neurotransmitter) into the Waikato cortical model.

A brief review of the neurophysiology is given in Chapter 2 in order to understand the biology behind our model. I start this Chapter by introducing the brain parts and defining the neuron and its structure. Then the electrical activity generated by a typical neuron was discussed, as well as neural communication. The categories and mechanism of anaesthesia are presented. Since NMDA effect our main concern in this thesis, it is discussed in the end of this Chapter.

Chapter 3 is about modelling the cerebral cortex. It starts by describing the mean-field model and giving a brief history of mean-field models, how these models improved over the time and what contribution each neuroscientist has made. Before we discuss the Steyn-Ross *et al.* (1999) model we give more detail about three relevant models: Hodgkin and Huxley (1952), Tuckwell (1988), and Liley *et al.* (1998, 1999, 2002). Then the equations of the Steyn-Ross *et al.* (1999) model are derived and explained in detail.

In Chapter 4 the effects of NMDA is introduced to our cortical model by following Jahr and Stevens (1990). After introducing the NMDA to the system, the stationary states are numerically computed. The main four input parameters [inhibitory neurotransmitter  $\lambda_i$ , subcortical drive  $s$ , NMDA excitatory neurotransmitters ( $\lambda_e$  and magnesium concentration  $C$ )] were varied to examine their effects on the steady states. We conclude this chapter by a theoretical stability prediction.

In Chapter 5 I simulate the system numerically using the one-step stochastic Euler method. Finally I discuss the results of the numerical simulation and compare them with the theoretical predictions of Chapter 4. In Chapter 6 a brief summary is given, as well as suggestions for further work.

## 1.2 Original work

The majority of this thesis is my own original work (Chapters 4, 5 and 6) while Chapters 2 and 3 give a summary of other people's work. However, I used sometimes other people's work for the sake of completeness. I received a beneficial guide from Dr Moira Steyn-Ross in formulating the model equations and their analysis. The `nmda_ss_finder.m` code is written by D. A. Steyn-Ross to locate the steady states. Some other codes are written by me with some help from Hono Kayano. Apart from that I did all needed calculations and analysis.

## Chapter 2

# Neurophysiology

### 2.1 Brain and neuron

The human brain is the most complex organ in the body (Tortora and Grabowski (2000)). The main four parts of the brain are: brain stem, diencephalons, cerebellum and cerebrum. The brain stem is located between the spinal cord and the diencephalons; it includes the midbrain, pons and medulla oblongata. The diencephalon stretches from the brain stem to the cerebrum and consists of the thalamus, hypothalamus, epithalamus and subthalamus. The second largest part of the brain is the cerebellum which is located in the inferior posterior portion of the head. It is divided into two large hemispheres, and made up of ten smaller lobules. Its essential function is to coordinate skilled movements and regulate posture and balance. Cerebrum is considered to be the largest element of the brain which lies in the top of the brain stem. It is also divided into two halves, left and right hemispheres and has four lobes: frontal, parietal, occipital and temporal lobes. It plays an important role in providing people with ability of reading, remembering, arithmetic and creating other significant skills.

Looking at the brain from a wider perspective, some regions look white (white matter) and glistening while others tend to be gray (gray matter). The white matter contains the myelinated axons from many neurons (long distance wiring) where myelin is white looking substance. The gray matter contains neuronal cell body, dendrites and axons where there is little or no myelin so it looks more grayish. It actually forms a layer (roughly 3 mm thick) around the cerebrum called the cerebral cortex. Projections (wiring connections) to and from the cerebral cortex are connected to the white matter which is located below the cerebral cortex.

Neurons (or nerve cells) are tiny data processing units that are thought to be responsible for most of the brain functionality located in cerebral cortex. There are approximately 100 billion neurons in the cerebral cortex and each neuron is connected to many other neurons, generating a vast neural network. Each neuron can either be excitatory or inhibitory, and it is believed that the cerebral cortex consists of 85% excitatory neurons and 15% inhibitory neurons (Braitenberg and

Schüz (1991)).

A single neuron (shown in Fig. 2.1) is composed of three parts: cell body (soma), dendrites and an axon. The soma (or cell body) has a cell nucleus surrounded by cytoplasm. The dendrites (dendritic trees) are generally short, extremely branched projections which are directed outward from the soma. The fundamental role of the dendrite is to sample its chemical environment by receiving inputs from surrounding neurons. The axon has a long slim cylindrical shape that stretches out from the cell body and ends in branched root projections. The role of the axon is to transmit information (nerve pulses) from the soma to the surrounding neurons.

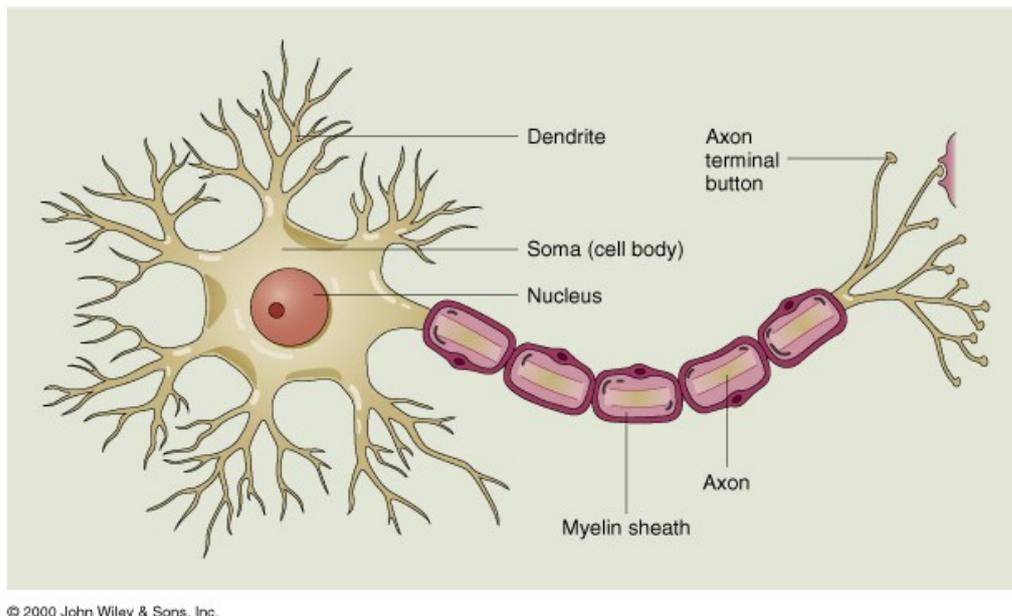


Figure 2.1: Diagram showing neuron structure. Each neuron consists of three main parts: the soma or cell body, the dendrites and the axon. [[http://scienceblogs.com/purepedantry/2006/07/background\\_to\\_the\\_20\\_year\\_coma.php](http://scienceblogs.com/purepedantry/2006/07/background_to_the_20_year_coma.php)]

## 2.2 Electrical signals in neurons

Each neuron, like other cells, contains ionic fluid (intracellular fluid), and is surrounded by ionic fluid outside its membrane (extracellular fluid). These two fluids have differing concentrations of certain ions (e.g.  $K^+$ ,  $Na^+$ ) causing a potential difference to be generated across the neuron's membrane (membrane potential). The extracellular fluid is used as the reference that defines zero potential. The neuron is said to be at rest when it is not receiving stimuli from its dendritic tree. The resting potential is found to be roughly  $-70$  mV relative to the defined zero potential. At resting membrane potential, the buildup of the negative ions in the intracellular fluid equals the buildup of the positive ions in the extracellular fluid.

The neuron's membrane contains ion channels that allow specific ions to diffuse

across the membrane. According to Tortora and Grabowski (2000) there are four main ions in a neuron intracellular and extracellular fluid: potassium ( $K^+$ ), sodium ( $Na^+$ ), calcium ( $Ca^{2+}$ ) and chloride ( $Cl^-$ ). There are two types of channels: *leakage* and *gated* channels. Leakage channels are always open and have higher permeability to some particular ions (typically membrane permeability to  $K^+$  is higher than  $Na^+$ ). In contrast, some sort of stimulus is required to cause the gated channels to open. For example, a change in membrane potential opens  $K^+$  channels during an action potential. In all cases, the diffusion of ions across a membrane channel causes a flow of current that can change the membrane potential.

## 2.3 Neuron communication

There are two types of electrical signals in neural communication: *graded potentials* and *action potentials*. The action potentials allow short and long distance communication within the body, while the graded potentials only allow short distance communication (Tortora and Grabowski (2000)).

When a stimulus causes the gated ion channels to open or close in a membrane, a neuron produces the graded potential which is a small deviation from the resting voltage that makes the membrane either more polarised (more negative) or less polarised (less negative). The less polarised case is called depolarising graded potential, while the more polarised case is called the hyperpolarising graded potential as shown in Fig.2.2.

If the depolarisation reaches a certain threshold value (approximately  $-60$  mV), the gated ion channels open, allowing the neuron's membrane to become less negative generating a pulse of voltage called an action potential (neuron's membrane fires). The action potential mainly consists of depolarising and re-polarising phases as shown in Figure 2.3. The action potential cycle starts at the resting membrane potential ( $K^+$  and  $Na^+$  are closed). The stimulus causes depolarisation to threshold ( $Na^+$  and  $K^+$  channels open), followed by a re-polarising phase in which  $Na^+$  channels are inactivating while  $K^+$  are still open, then an after-hyperpolarising phase where  $Na^+$  are in resting state while  $K^+$  are still open, and finally back to the resting state.

In order to understand the communication in the neural nervous system, it is essential to first understand the synaptic communication between two neurons. The neuron sending the signal is called pre-synaptic, while the neuron receiving the signal is called post-synaptic. Between these two pre-and post-synaptic neurons there exists a gap called the synaptic cleft. The communication is done via electrical and chemical signals, where the electrical activity is accountable for the flow of information around the neuron while the actual communication between neurons is achieved by a chemical process. The overall communication process starts when the pre-synaptic electrical signal (nerve impulse) is converted into a chemical signal (called neurotransmitter), that gets diffused across the synaptic cleft and acts on

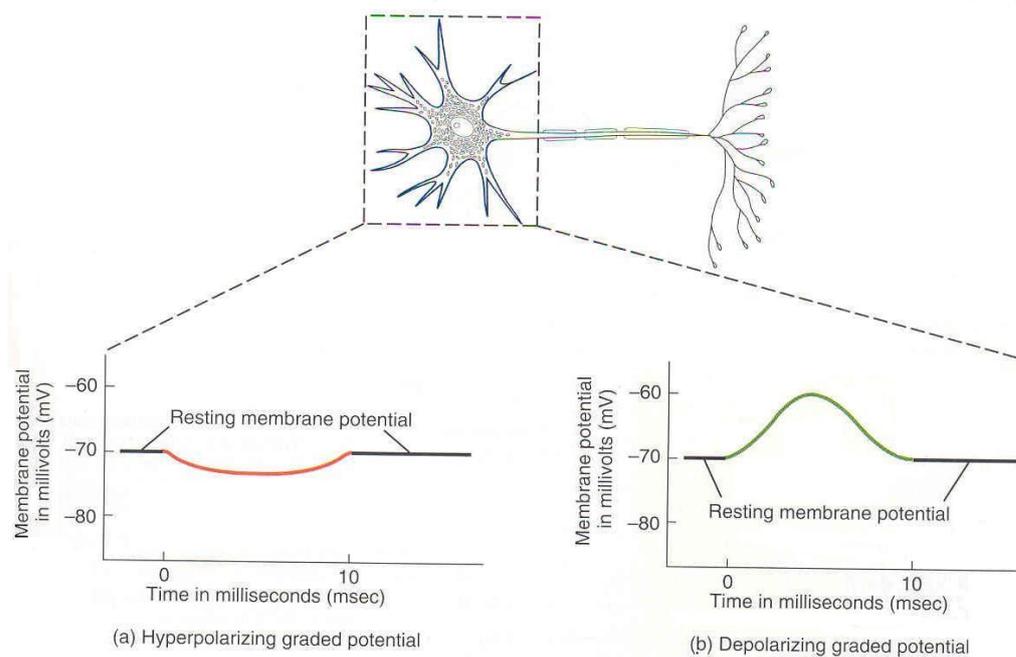


Figure 2.2: Graded potential; most graded potentials appear in the cell body and dendrites (dashed outline). (a) The membrane polarisation becomes more negative than the resting level (hyperpolarising graded potential). (b) The membrane polarisation becomes less negative than the resting level (depolarising graded potential). [Source: Fig. 12.10 of Tortora and Grabowski (2000)]

receptors in the post-synaptic neuron's membrane, causing a post-synaptic potential and finally converted back to an electrical signal again. Figure 2.4 explains the steps of the synaptic communication as following:

1. Action potential arrives at the pre-synaptic axon;
2. Depolarising phase of the action potential in which both  $\text{Ca}^{2+}$  and  $\text{Na}^{+}$  channels are normally opened;
3. Increase in the concentration of  $\text{Ca}^{2+}$  inside the presynaptic neuron triggers some chemical process that releases neurotransmitter molecules into the synaptic cleft;
4. Neurotransmitter molecules diffuse across the synaptic cleft and bind to neurotransmitter receptors in the post-synaptic neurons's membrane;
5. Binding of neurotransmitter to their receptors opens the channels and allows particular ions to flow across the membrane;
6. Ionic flow causes depolarisation or hyperpolarisation of the postsynaptic membrane depending on which ions the channels admit;
7. If depolarisation reaches threshold (roughly  $-60$  mV), one or more action potentials are generated.

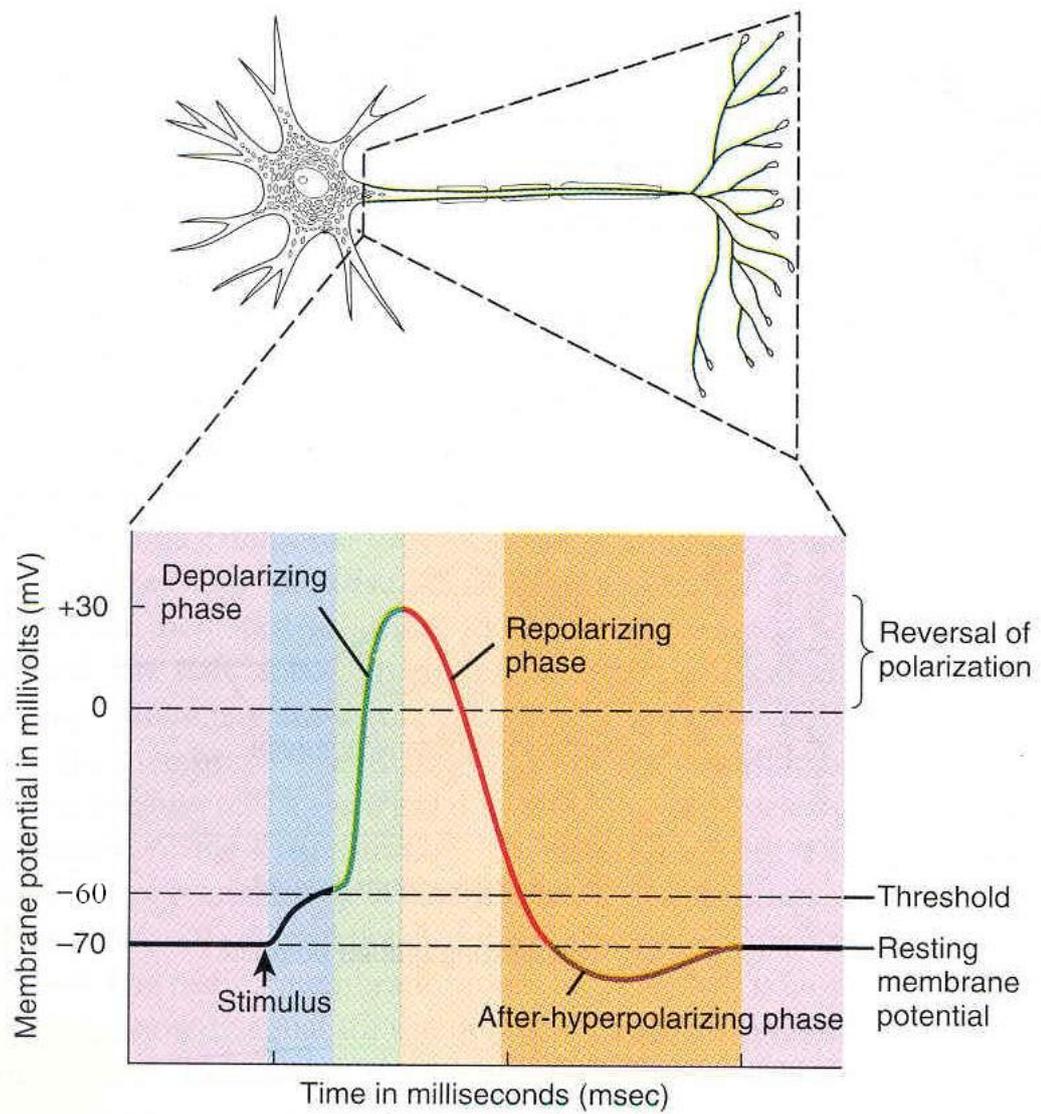


Figure 2.3: Action potential (AP) or impulse. When the neuron membrane is depolarised by a stimulus, an action potential is generated. [Source: Fig. 12.11 of Tortora and Grabowski (2000)]

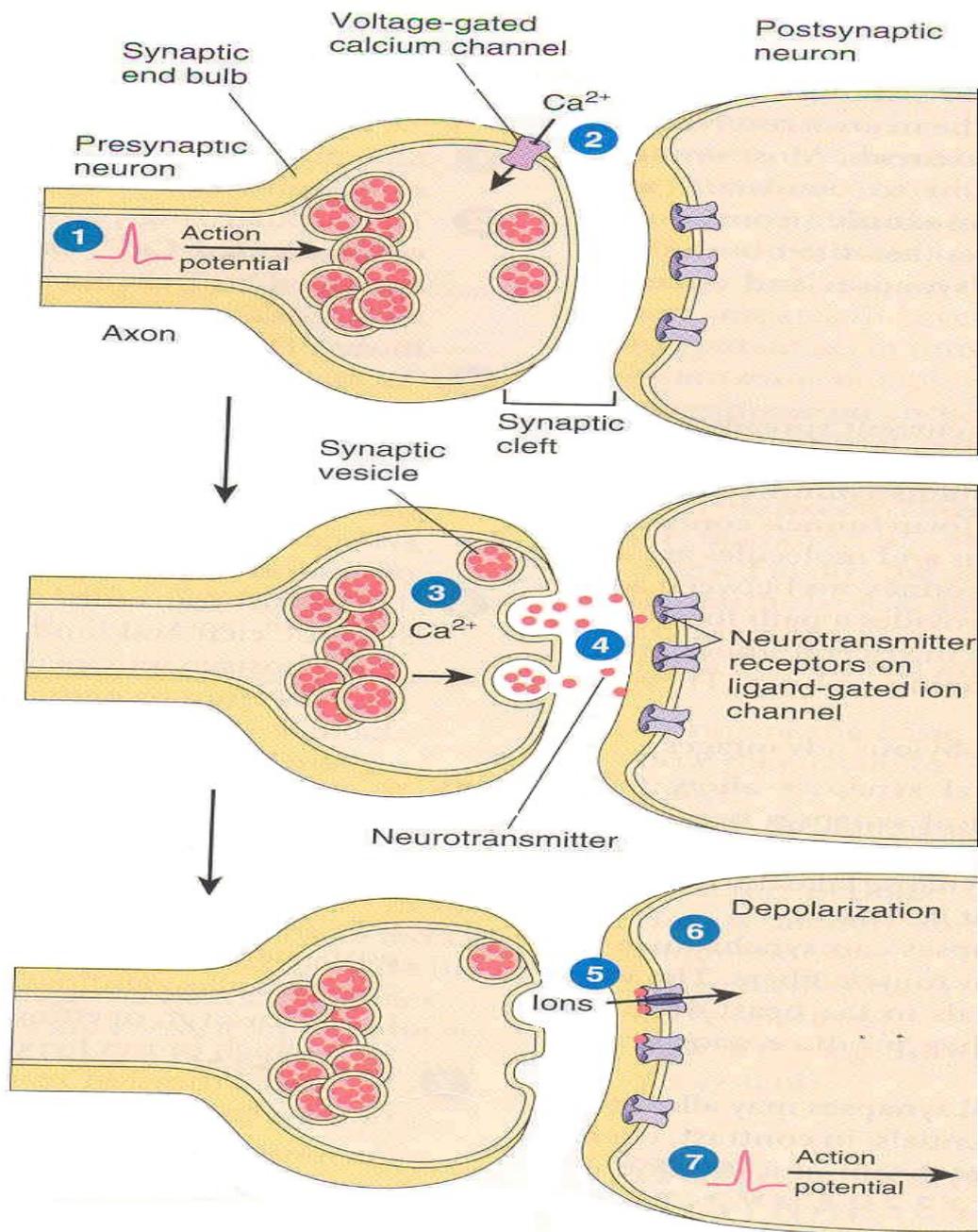


Figure 2.4: Signal transmission at a chemical synapse. Neurotransmitter molecules are released from a presynaptic neuron and bind to receptors in the membrane of the postsynaptic neuron, then a postsynaptic potential is produced (may eventually lead to an action potential). [Source: Fig. 12.14 of Tortora and Grabowski (2000)]

### 2.3.1 Excitatory and inhibitory post-synaptic potential

The post-synaptic potential generated by the neurotransmitter can be either excitatory or inhibitory. If the neurotransmitter causes the post-synaptic membrane to depolarise, then the potential generated is called an excitatory post-synaptic potential (EPSP). In contrast, if it causes the post-synaptic membrane to become

hyperpolarised, then the potential generated is called an inhibitory post-synaptic potential (IPSP).

The EPSP occurs when the gated channels open allowing  $\text{Na}^+$ ,  $\text{K}^+$  and  $\text{Ca}^+$  ions to pass through, but the inflow of  $\text{Na}^+$  is greater than either  $\text{K}^+$  outflow or  $\text{Ca}^+$  inflow. The generation of the EPSP does not necessarily mean that a firing will occur but it makes the neuron more excitable and hence more likely to fire in the next EPSP.

On the other hand, the IPSP occurs when the gated  $\text{Cl}^-$  or  $\text{K}^+$  channels open. When  $\text{Cl}^-$  channels open, the membrane permeability to  $\text{K}^+$  increases causing more diffusion of  $\text{K}^+$  outwards and leading to a more negative membrane and hence less likely to fire.

## 2.4 Anaesthesia

Anaesthesia is categorised as a crucial clinical and surgical medication as it has the ability to safely and reversibly render a patient unconscious. This surgical revolution, started in the 1840s by Oliver Wendell Holmes, allowed many advanced and complicated surgeries to be performed. The anaesthetised patient may have the following effects: lack of pain sensation, lack of memory and muscle relaxation.

Depending on the type of surgery and the patient's medical condition, one of four broad categories of anaesthesia can be used:

1. **Local anaesthesia:** This type of medication is given to block sensation at a particular part of the body while the patient remains conscious. Usually an injection of local anaesthetic is given in the area of nerves that provide feeling to that part of the body;
2. **Sedation:** This medication makes a patient "drowsy" in an effort to make him/her more comfortable. It does not result in complete unconsciousness but can result in amnesia (lack of memory) and the patient may "fall asleep" for a period of time;
3. **General anaesthesia:** This medication causes the patient to be totally unconscious yet his vital physiological functions, such as breathing and maintenance of blood pressure, continue to function.

Throughout this thesis, the general anaesthesia is the type that is studied and assumed for the modelling. According to Guedel (1937), there are four stages in the induction of general anaesthesia:

- 1 **Analgesia and amnesia:** Patient remains conscious while experiencing pain relief and dreamy disorientation;
- 2 **Delirium:** Consciousness is lost with possible signs like: rise in blood pressure and irregular breathing, breath holding, swallowing, vomiting and uncontrolled violent movements;

- 3 **Surgical anaesthesia:** At this stage the surgery is safe where breathing returns to regular, skeletal muscles are relaxed, eye movements slow;
- 4 **Respiratory paralysis:** This is the anaesthetic crisis stage where the patient's respiration stops; death from circulatory collapse will follow without assisted ventilation and circulatory support.

It has been found that it is possible for an apparently anaesthetized patient to wake up during an operation, turning the operation into a nightmare. To prevent this, the anaesthetist should monitor the movements and the behaviour of the patient throughout the operation's duration and adjust the drug amount and concentration as required. Patient behaviour is an indirect and imprecise measure of the anaesthetic depth, and may lead to under or overdosing the anaesthetic drug, Kuizenga (2001).

To be more precise in determining brain activity, a device was invented to indirectly measure the electrical activity inside the cerebral cortex via electrodes placed on the scalp. This device draws a voltage-vs-time traces known as EEG (electroencephalogram) that the anaesthetist can use to determine if changes in drug concentration are needed.

According to Steyn-Ross (2002), an observer of a patient under anaesthetic effect would notice a change in the inhibition/excitation of the brain at particular stages of the anaesthesia. There is a reduction in brain activity at stage 1 after the drug is given. This brain inhibition is followed by an abrupt excitation state found in stage 2. The brain then goes back to a full inhibition state in stage 3 where surgeries are performed. The conscious-to-unconscious transition (stage 2 to 3) looks similar to a classical first-order thermodynamic phase transition in physics. Therefore he coined the phrase "*Anaestheto-dynamic phase transition*" for this abrupt change in the brain state induced by a general anesthetic drug.

## 2.5 Anesthetic mechanism

Despite the use of general anesthetics for more than 150 years, its mechanism of action inside the brain remained a mystery for a long time. However, with the use of modern advanced scientific knowledge and experimental techniques, scientists have made good progress in understanding the brain's activity and the anesthetic mechanism. Understanding the types of synaptic transmission would give a possible explanation about changes in brain activity during anaesthesia.

There are two types of synaptic transmissions: excitatory and inhibitory transmission. The type of the transmission depends on the type of neurotransmitters and the corresponding ion channel receptors that they interact with. The inhibitory transmission mainly uses a neurotransmitter called GABA ( $\gamma$ -aminobutyric acid) which interacts with GABA receptors that are permeable to negatively charged chloride ions ( $\text{Cl}^-$ ). The main effect of GABAergic anaesthetics (e.g. propofol)

is to prolong GABA channel opening, increasing the postsynaptic inhibition. This would lead in opening these channels for longer time generating a hyperpolarising wave and hence making the neurons less likely to fire, and causing unconsciousness. Franks and Lieb (1997) argue that this explains the change in brain activity caused by general anaesthetics. Fig. 2.5 illustrates the interaction between chloride ions and the GABA receptor.

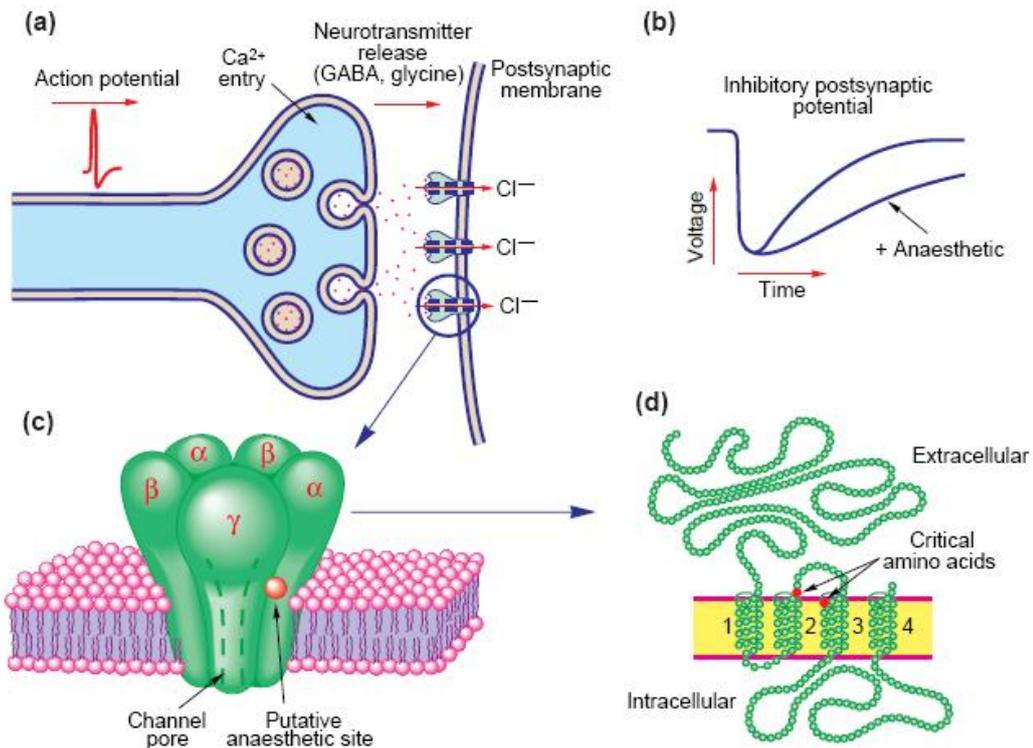


Figure 2.5: A GABAergic model for anaesthetic action: (a) An action potential at the membrane of an inhibitory presynaptic neuron causing GABA neurotransmitter to be diffused across the synaptic cleft. The GABA receptors in postsynaptic neuron's membrane open to the chloride ions  $\text{Cl}^-$  resulting in hyperpolarisation. (b) The effect of prolonged opening of the postsynaptic GABA channels leading to an increase in inhibition. (c) Structure of each channel with its five protein subunits. (d) The structure of each subunit of each channel. [Source: Fig. 1 of Franks and Lieb (1997)]

The GABA receptor shown in Fig. 2.5(c) is simplified as a block diagram in Fig. 2.6. Inhibition occurs when the GABA neurotransmitter binds with the GABA receptor molecule causing hyperpolarisation. Overall, this enhancement of inhibitory effect is thought to be the basis of the so-called GABAergic anaesthetics that lead to fully unconsciousness state.

The second type of synaptic transmission is excitatory. According to Waters and Machaalani (2004), the L-glutamate (amino acid) neurotransmitter interacts with glutamate receptors to allow the ingress of positively charged sodium ions ( $\text{Na}^+$ ), causing depolarisation waves in the postsynaptic neuron, making it more likely to fire. Thus an alternative method for inducing anaesthesia would be to reduce the

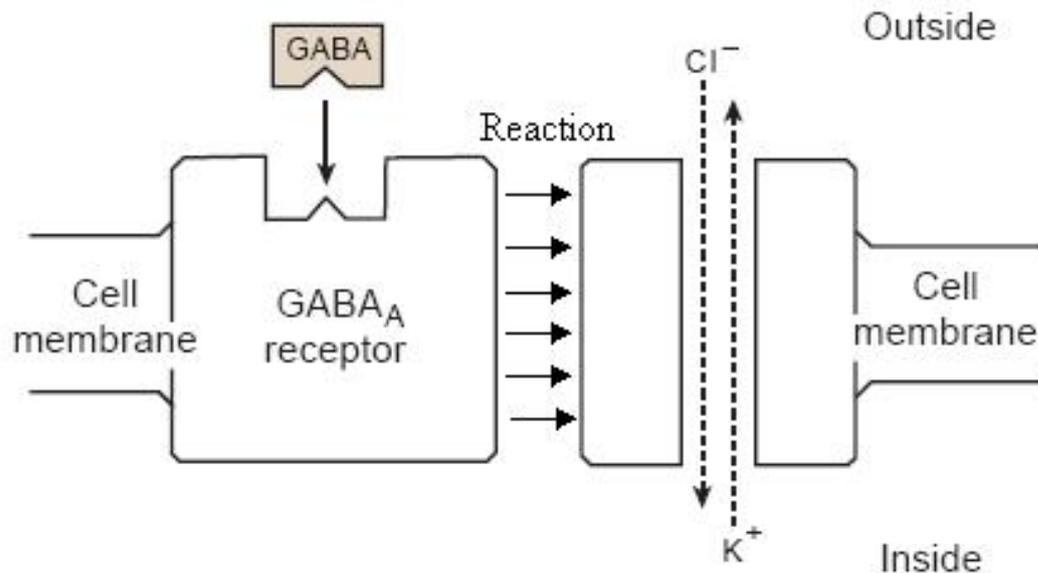


Figure 2.6: A block-diagram of the GABA receptor: Inhibition occurs when the GABA neurotransmitter binds with the GABA receptor molecule causing hyperpolarisation. [Source: Fig. 1.3 of Steyn-Ross (2002)]

effectiveness of these excitatory synaptic events.

The excitatory synaptic transmission activates two types of channel receptor: metabotropic and ionotropic. The metabotropic receptors are not ionic channels but pass signals into the cell via other accessory proteins. The ionotropic channel receptors have three types: AMPA, NMDA (*N*-methyl-D-aspartate) and kainate. As NMDA is our main concern in this thesis we will go into more detail. The NMDA receptors are composed of assemblies of three subunits: NR1, NR2A-D and NR3A as shown in Fig. 2.7. NMDA action is unique because it requires simultaneous binding of two subunits, NR1 subunit combined with one or more NR2 or NR3 subunits. It is permeable to  $\text{Na}^+$ ,  $\text{K}^+$  and  $\text{Ca}^{2+}$  ions and regulated by  $\text{Mg}^{2+}$ , which serves to block  $\text{Ca}^{2+}$  influx in a voltage-dependent manner. NMDA is important for many neural process such as neuronal migration, proliferation, injury and synaptic plasticity (where brain structure can change to better cope with the environment), Waters and Machaalani (2004).

The NMDA neurotransmitter increases excitatory activity. It is possible to induce anaesthesia by suppressing the excitatory effectiveness of the NMDA receptor on the postsynaptic neuron's membrane. This NMDA-antagonist class of agent is called a *dissociative anaesthetic*. These dissociative agents do not usually result in full unconsciousness (Franks *et al.* (1998); Jevtovic-Todorovic *et al.* (1998)). The duration of the the excitatory postsynaptic response to the presynaptic action potential depends on both anaesthetic concentration (greater concentration is inversely proportional to the duration of the EPSPs) and the voltage state of the receiving neuron (the magnesium block in the ion channel is only removed when the neuron is depolarised). This fact makes modelling the action of the NMDA receptor compli-

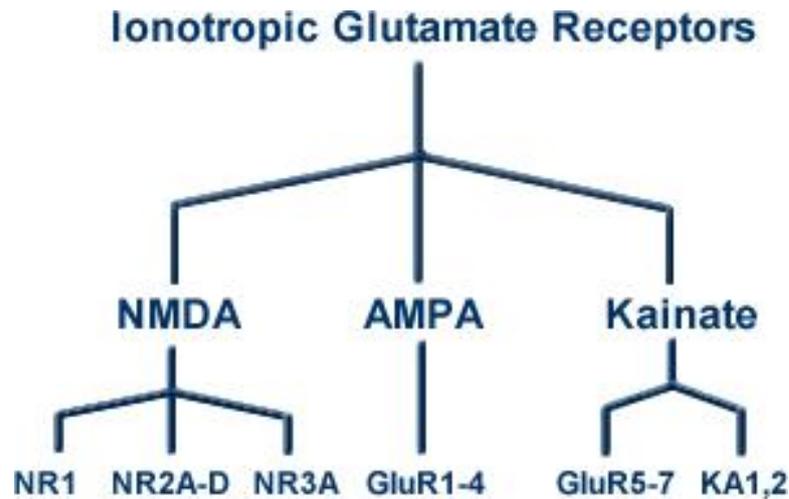


Figure 2.7: Classification of the ionotropic glutamate receptors. [source: [http://www.bris.ac.uk/synaptic/public/basics\\_ch1\\_1.html](http://www.bris.ac.uk/synaptic/public/basics_ch1_1.html)]

cated. However, the focus of this thesis is to develop a model for NMDA-antagonist effect in general anaesthetic. Fig. 2.8 shows a simplified block diagram of the NMDA receptor.

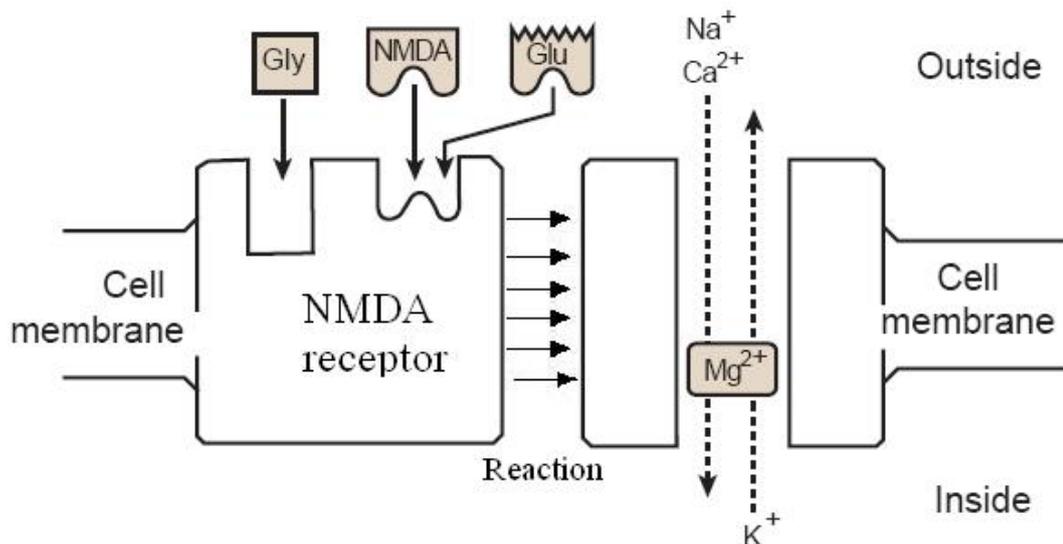


Figure 2.8: In its inactivated (not excited) configuration, a magnesium ion blocks the channel within the NMDA receptor. Once excitation occurs, this magnesium block is removed allowing other ions to diffuse down their concentration gradient:  $\text{Na}^+$  and  $\text{Ca}^{2+}$  ions enter the cell causing depolarisation. The excitation occurs when Gly (glycine) and one of either Glu (glutamate) or NMDA binds to the receptor molecule. [Based on Fig. 1 of Thomas and Riley (1998)]

### 2.5.1 Indirect effects of anaesthetics on the NMDA synapse

The anaesthetic action of anaesthetics that do not primarily affect the NMDA synapse, such as GABA agonists, can be a consequence of indirect effects on the

working conditions of the NMDA receptor (Flohr *et al.* (2000)). In other words, in order for an anaesthetic action to be induced by any agent, the NMDA receptor will be indirectly effected (disrupted). According to Flohr's model of the NMDA synapse as a target of anaesthetics, the activity of the NMDA receptor could be modified by inhibitory (GABA) and excitatory (AMPA) synapses located near the receptor as shown in Fig. 2.9.

This hypothesis was proven by Flohr's experimental evidence showing that propofol (a GABAergic agent) partially blocks the uptake of radio-labelled MK-801 (an NMDA antagonist that binds to the PCP site within the NMDA channel). It was also demonstrated by Steyn-Ross (2002) modelling of GABAergic induction, where the "dissociated state" could also occur when slow NMDA-mediated currents are incorporated into the model.

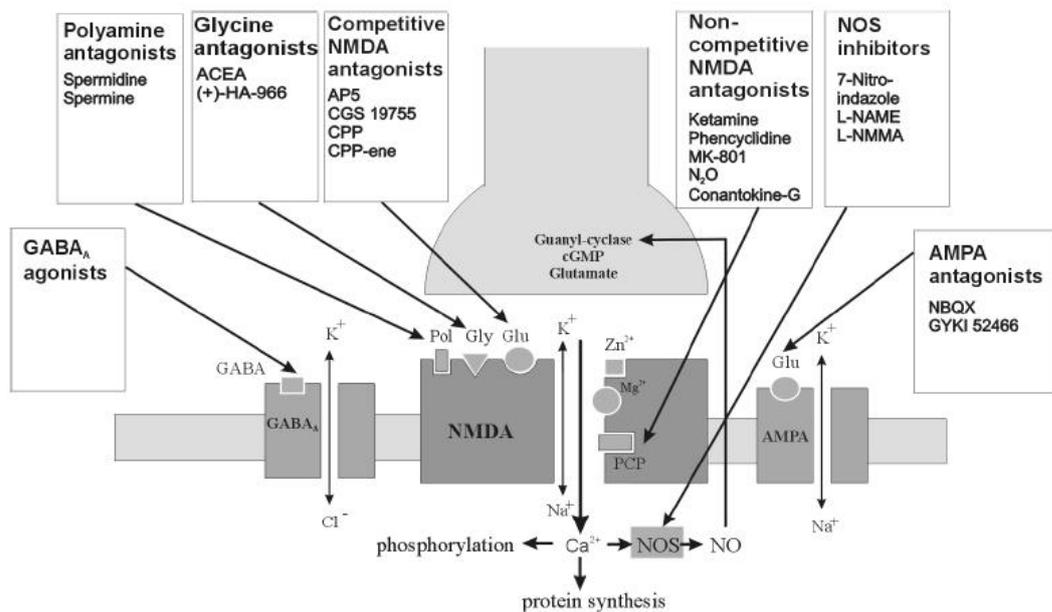


Figure 2.9: The Flohr model of the NMDA synapse as a target for anaesthetics. GABA and AMPA receptors can influence NMDA receptor working condition. [Source: Fig. 2 of Flohr *et al.* (2000)]

## Chapter 3

# Modelling the cerebral cortex

### 3.1 Mean-field model

Modelling the relations between input and output signals for an individual neuron is a hard task to achieve due to the enormous number of contributing factors. Therefore neuroscientists divide neurons into populations of the same type having dense random interconnections, and model these populations. Each group of neurons is called a “neural mass”. The properties of this neural mass (e.g. firing rate, time-dependent membrane potential, etc.) are determined by the spatial averages of all the neurons within this neural mass, so the model is given the name “mean-field”.

Our mean-field model describes the electrical activity inside a small volume of neural tissue called a macrocolumn. The macrocolumn is cylindrical volume ( $\sim 1 \text{ mm}^3$ ) with a diameter of  $\sim 0.3\text{--}1 \text{ mm}$  and length of  $\sim 2\text{--}3 \text{ mm}$ . This column contains roughly 40,000 to 100,000 neurons where approximately 85% are excitatory and 15% are inhibitory (Braitenberg and Schüz (1991)).

Since our modelling is concerned with neural activity during the anaesthetic state, the goal is to determine the level of consciousness of the anaesthetized patient. This modelled level should be consistent with the clinical results achieved by monitoring electroencephalogram (EEG) recordings. The EEG indirectly measures the electrical activity inside the cerebral cortex by electrodes placed on the scalp. By modelling the fluctuation of the excitatory membrane potential, we can make comparisons between theory and the experimentally acquired EEG.

### 3.2 Brief history of mean-field models

The following section will describe in short the development of mean-field models. More elaboration about certain cortical models will be given later depending on their relevance to our research.

The mean-field model was first proposed by Beurle (1956). His model only included an excitatory population of neurons where the connectivity between the neurons decreases exponentially with distance. He assumed a threshold value that

the neuron's membrane voltage must exceed in order for it to fire. Furthermore, he also included a refractory period that occurs after the firing during which the neuron cannot fire again (threshold potential depends on its history, potential is high before decaying to its resting value). As an effect of the refractory period, the neuron temporarily becomes insensitive to further firing, which creates a state of control that prevents any global firing propagation. Yet, this is inadequate to provide the level of stability that a real cortex can maintain. This paradox was overcome later in 1963 by Griffith who introduced the inhibitory synapses that hyperpolarised the post-synaptic neuron making it less likely to fire.

Some years later, Wilson and Cowan (1972, 1973) also modelled both excitatory and inhibitory populations, and assumed that synaptic activity was much slower than the membrane time-constant. They simplified their equations by using a time coarse-graining assuming that the PSPs (post-synaptic potentials) were rectangular functions for both excitatory and inhibitory events, differing only in sign. They introduced two factors that have been used in later mean-field models: considering inhibition acts only locally with a rapid exponential decrease with distance, and the use of nonlinear sigmoid function to map from membrane voltage to firing rate.

Following this, Freeman (1975) formalised the concept of mean-field by introducing the concept of neural mass as populations of neurons having different interconnection properties. He also emphasised the conversion of synaptic input to membrane potential and then to action potential rate (firing rate). In his model, both synaptic and dendritic delays were included, noting that the contribution of delays in feedback loops could lead to oscillatory behaviour in populations of excitatory and inhibitory neurons.

The spatial and temporal variation of cortical voltages generated by a neural mass were described as an integral wave equation in Nunez's model [Nunez (1974, 1981)]. In this model, an oscillation in the cortical voltages was expected to be determined by the relative numbers of excitatory and inhibitory neural mass connections, and by the velocity distribution functions for action potential propagation. For particular choices of boundary conditions, the standing waves can be calculated for certain wave numbers. He then stated that this is the source of cortical rhythms such as the alpha rhythm seen in the EEG. The importance of including boundary conditions was also considered by later modellers.

Jirsa and Haken (1996, 1997) derived mean-field equations for the dendritic currents by generalising the work of Wilson and Cowan and of Nunez. Following Wilson and Cowan, they assumed that the EPSP and IPSP functions have the same magnitude but different sign. They considered the activity of the inhibitory population as a function of the the excitatory neural population activity, meaning that the firing rate of an inhibitory neuron is dependent on the membrane potential of surrounding excitatory neurons, and not on its own membrane potential.

Also in 1996 and 1997, Wright and Liley used anatomical data to derive expressions for the number of synapses between two neurons as a function of their

separation. They used a sigmoidal form for the mapping between membrane voltage and spike rate, and approximated the post-synaptic potential (PSP) as a triangular function. In their model, they followed Wilson and Cowan in assuming that the synaptic activity was much slower than the membrane time-constant.

One year later, Robinson *et al.* (1997) further developed the Wright and Liley model by using a more physiologically realistic biexponential function instead of a rectangular PSP. They studied the steady states of the model and found that it can either have one or three steady states, with the middle state being unstable for the case of three steady states. Robinson *et al.* (1998) represented their steady state solutions as a ratio of  $\ell_i/\ell_e$  where  $\ell_i$  or  $\ell_e$  represents the net post-synaptic response per unit synaptic concentration of inhibitory/excitatory neurotransmitter. The three states only appear when  $\ell_i/\ell_e = 1$  (same magnitude for excitatory and inhibitory responses). However, if the domination of excitatory response over inhibitory is stronger (or vice versa), the system reduces to only one stable state.

The later work of Liley *et al.* (1999), and of Liley *et al.* (2002), improved the earlier models and theories by enhancing the treatment of excitatory and inhibitory neurotransmitter kinetics. For the first time in mean-field models, they incorporated the constraints on depolarisation and hyperpolarisation voltage extremes enforced by cell reversal potentials.

The Steyn-Ross *et al.* (1999) model, a modification of Liley's model, was designed to predict the general behaviour of a patient's brain under general anaesthesia. They found only certain stable stationary states for the soma voltages as they increased the time course of the inhibitory post-synaptic potential (IPSP). Then they introduced stochastic subcortical terms to model the fluctuation seen in an EEG. The results of the stability analysis of the steady states showed that an abrupt change from one state to another must occur as anaesthetic concentration is increased, corresponding to the anaesthetised patient abruptly moving from consciousness to unconsciousness.

### 3.3 Neural equivalent circuit

The neural system can be represented as an electrical circuit with three main components for each ionic species ( $\text{Na}^+$ ,  $\text{K}^+$ ,  $\text{Cl}^-$ ) as shown in Fig. 3.1. A battery is used to represent the difference in concentration between the intracellular and extracellular ionic fluid. For example, the extracellular concentration of  $\text{Na}^+$  is much greater than its intracellular concentration, so the  $\text{Na}^+$  battery is oriented to produce an inwards conventional current. If the membrane becomes more permeable to  $\text{Na}^+$ , more  $\text{Na}^+$  ions would flow inwards into the cell interior. The magnitude of the battery voltage can be calculated using the Nernst potential formula (Eq.3.1) described shortly. The second component is the variable conductance (inverse-resistance) which models the membrane pores (channels). Finally, a capacitor represents the thin accumulations of ions in the water either side of the membrane.

The Nernst potential is the ideal equilibrium voltage for the diffusive force arising

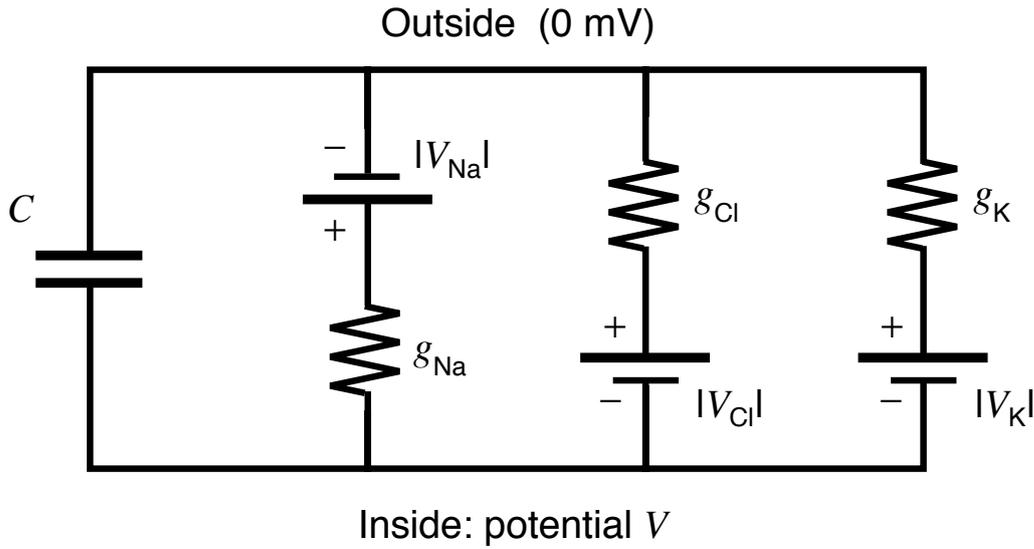


Figure 3.1: Neural equivalent circuit. [Source: Steyn-Ross (2002)]

Table 3.1: Typical ion concentration (in mmol/L) and Nernst potentials (mV) for a resting neuron. Nernst potentials were calculated for a neuron at body temperature (37°C) giving  $\frac{RT}{F} = 26.7$  mV. [Source: Steyn-Ross (2002)]

Ion	Inside Concentration	Outside Concentration	Nernst Potential
Na <sup>+</sup>	10	140	+70
K <sup>+</sup>	140	4	-95
Cl <sup>-</sup>	4	103	-87
Ca <sup>2+</sup>	< 10 <sup>5</sup>	5	> 350

from the concentration gradient across the membrane of a given ion. This Nernst potential ( $V_X$ ) for a given ion (X) can be calculated using the following formula:

$$V_X = V_{in} - V_{out} = \frac{RT}{zF} \log_e \frac{[X]_{out}}{[X]_{in}} \quad (3.1)$$

where:

$T$  is the absolute temperature,

$R = 8.314 \text{ J K}^{-1} \text{ mol}^{-1}$  is the ideal gas constant,

$z$  is the signed valence of the ion (e.g.  $z = -1$  for Cl<sup>-</sup>),

$F = 9.648 \times 10^4 \text{ C mol}^{-1}$  is the Faraday constant.

There are three possible cases of concentration gradient for a positive ion: if the membrane voltage  $V = V_X$ , there is no flow of current; if  $V < V_X$ , current will flow down the concentration gradient; and finally, if  $V > V_X$ , the current direction will reverse. The Nernst potential is also known as the “reversal potential”, since it is the point of voltage balance between normal and reversed current direction.

## 3.4 Hodgkin-Huxley model

### 3.4.1 Resting voltage

A series of experiments on a squid giant axon were carried out by Hodgkin and Huxley (1952) investigating the flow of electrical current through the squid's membrane surface. Their goal was to develop a model using a mathematical description derived from the conduction and excitation behaviour of the membrane. This model has been used as a basis for later ionic current models of excitable tissues.

The neural equivalent circuit described earlier (Fig. 3.1) was derived from the Hodgkin and Huxley model describing the flow of current through the membrane surface. The capacitance and conductance determines the current flow through the membrane surface. The overall current is made up of four sources: the capacitive current ( $I_C$ ), sodium ( $I_{Na}$ ), potassium ( $I_K$ ) and chloride ( $I_{Cl}$ ) ionic channel currents represented by the following formulae:

$$I_C = C \frac{dV}{dt} \quad (3.2)$$

$$I_{Na} = (V - V_{Na})g_{Na} \quad (3.3)$$

$$I_K = (V - V_K)g_K \quad (3.4)$$

$$I_{Cl} = (V - V_{Cl})g_{Cl} \quad (3.5)$$

where the coefficients  $g_{Na}$ ,  $g_K$ ,  $g_{Cl}$  are the channel conductances ( $\Omega^{-1}$ ).

These four formulae can be assembled into a single equation:

$$I = C \frac{dV}{dt} + I_{Na} + I_K + I_{Cl} \quad (3.6)$$

At rest  $I = 0$  and  $C \frac{dV}{dt} = 0$ , thus

$$0 = (V - V_{Na})g_{Na} + (V - V_K)g_K + (V - V_{Cl})g_{Cl} \quad (3.7)$$

Therefore Eq.(3.7) predicts a final resting voltage that is the conductance-weighted sum of Nernst potentials (typically  $-70$  mV):

$$V_{rest} = \frac{g_{Na}V_{Na} + g_KV_K + g_{Cl}V_{Cl}}{g_{Na} + g_K + g_{Cl}} \quad (3.8)$$

### 3.4.2 Action potential

Hodgkin modelled the action potential of the squid giant axon by replacing the constant conductances of  $Na^+$ ,  $K^+$  shown in Fig.3.1 with variable conductances (voltage and time-dependent) as illustrated in Fig.3.2.

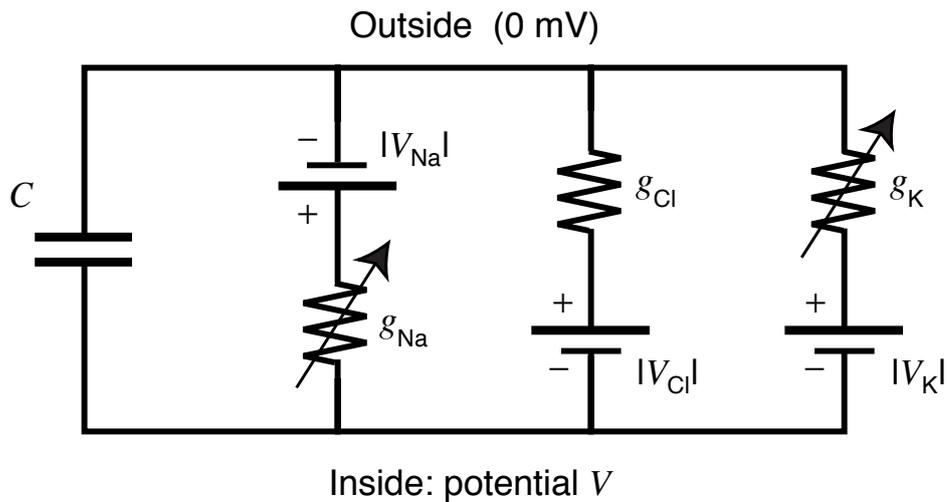


Figure 3.2: Neural equivalent circuit with variable conductances. [Source: Steyn-Ross (2002)]

The voltage-dependent  $\text{Na}^+$  channels are normally closed at resting state. Once the membrane voltage ( $V_m$ ) is driven from its resting value ( $-70$  mV) to the threshold value (roughly  $-60$  mV), the  $\text{Na}^+$  channels activate and abruptly open, increasing the sodium conductance  $g_{\text{Na}}$ , leading to a rapid influx of  $\text{Na}^+$  ions. This drives the membrane voltage towards the sodium reversal potential  $V_{\text{Na}}$  as indicated in Fig.3.3a. The  $\text{Na}^+$  channels begin to close within a fraction of a millisecond. Simultaneously, the voltage-dependent  $\text{K}^+$  channels become activated due to the voltage upswing on the leading edge of the action potential. Hence, the  $\text{K}^+$  channels open, producing a larger outwards current. Combining the effect of the activation and deactivation of the channels results in a rapid downswing to terminate the action potential as drawn in Fig.3.3b.

### 3.5 Tuckwell subthreshold neuron

One of the goals of the Hodgkin-Huxley model is to find a precise mathematical description of the neuron action potential. However, action potentials cannot be seen in scalp EEG records. This is due to the very fast rise and fall times of the action potentials which are strongly low-pass filtered by the cerebrospinal fluid, scalp tissue and skull which are located between the brain surface and scalp electrode. In fact, the EEG records display the much more slowly varying fluctuations in the spatially- and temporally-averaged local potentials of the excitatory neurons.

Modelling the EEG signal requires computing the soma voltage calculated from a membrane integration of all the incoming presynaptic activity, consisting of the inflowing excitatory postsynaptic currents (EPSCs) minus the outflowing inhibitory postsynaptic currents (IPSCs). The averaged effect of a train of action potentials on the neural population can be preserved via a sigmoid (Fig.3.10) mapping from aver-

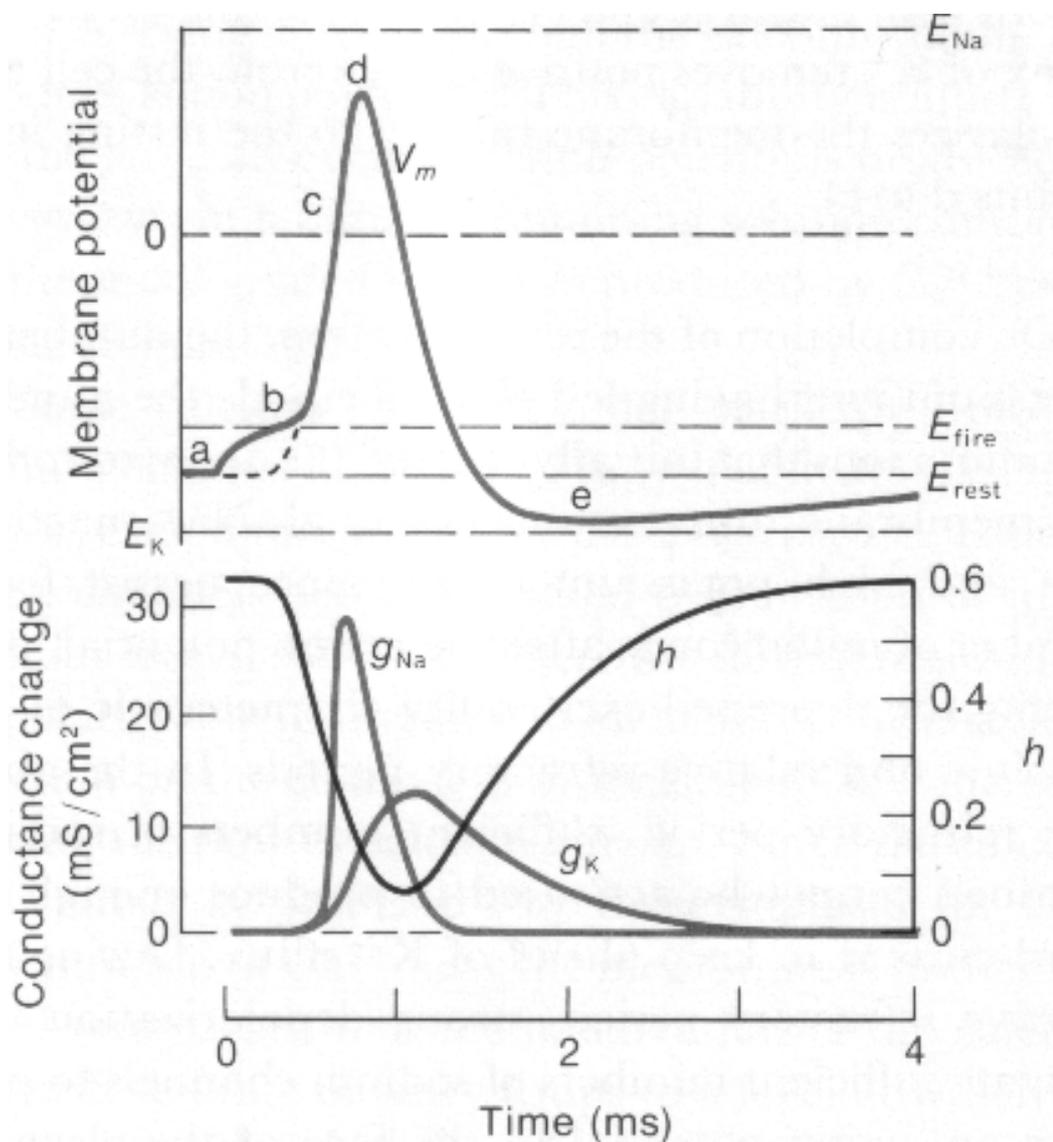


Figure 3.3: a: Prediction from the Hodgkin and Huxley model for the formation of an action potential. b: Action potential is generated by differentially-timed changes in Na<sup>+</sup> and K<sup>+</sup> conductances. [from Halsey and Smith (1970), after Hodgkin and Huxley (1952)]

age soma voltage to average spike-rate. The spike-rate determines the presynaptic activity and hence the average postsynaptic voltage.

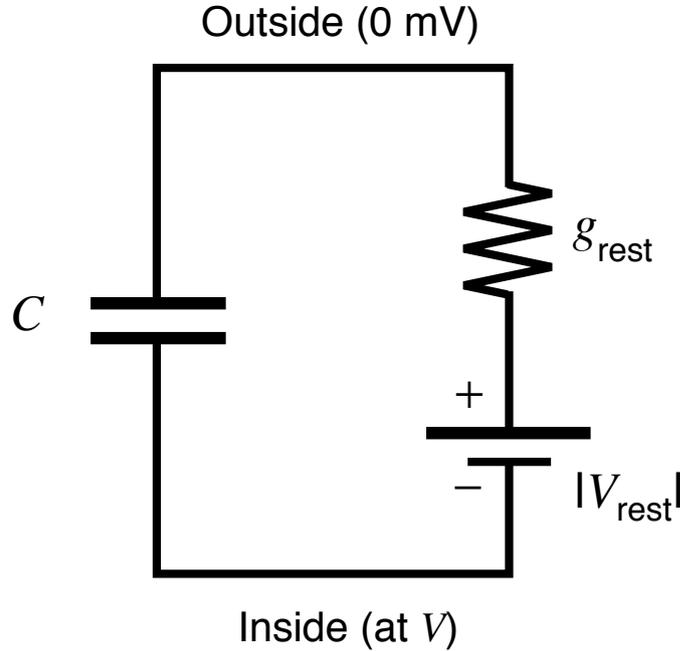


Figure 3.4: The  $K^+$ ,  $Cl^-$  and  $Na^+$  batteries and conductances replaced by  $V_{rest}$  in series with  $g_{rest}$ . [Source: Steyn-Ross (2002)]

Modelling the EEG generated by the neuron population was attempted using mean-field theory developed by Liley *et al* (1998, 1999, 2002). The Liley neuron models the formation of the EEG by the fluctuating potential of populations of excitatory and inhibitory neurons. It is useful to view the Liley neuron as a generalisation of the Tuckwell subthreshold neuron described in Tuckwell (1988).

The Tuckwell subthreshold neuron was produced by modelling a resting neuron coupled with synaptic inputs. Firstly, the sodium, chloride and potassium batteries and conductances are replaced with a single battery  $V_{rest}$  in series with resting conductance  $g_{rest}$  as Fig.3.4 illustrates. As the neuron is in a resting state, the value of the battery  $V_{rest}$  is maintained at  $-70$  mV. The second step is to couple in the synaptic inputs. Tuckwell assumes that there is only one ion species involved in the excitation and similarly only one involved in inhibition. Each synaptic input is then modelled by a variable conductance and a synaptic reversal potential  $V_E$  and  $V_I$  (same as Nernst potentials). The final Tuckwell neuron is drawn in Fig. 3.5.

Total currents sum to zero, giving

$$C \frac{dV}{dt} = (V_{rest} - V)g_{rest} + (V_E - V)g_E + (V_I - V)g_I \quad (3.9)$$

Dividing the previous equation by  $g_{rest}$  yields:

$$\tau \frac{dV}{dt} = (V_{\text{rest}} - V) + (V_E - V) \frac{g_E}{g_{\text{rest}}} + (V_I - V) \frac{g_I}{g_{\text{rest}}} \quad (3.10)$$

where  $\tau = C/g_{\text{rest}}$  is the membrane time constant.

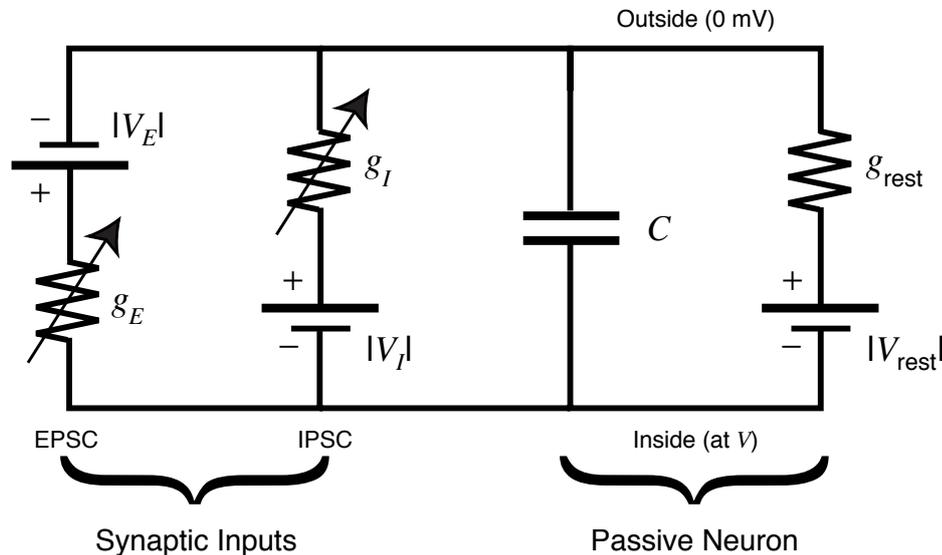


Figure 3.5: Tuckwell subthreshold neuron. The capacitor integrates the synaptic input currents to give a graded local potential  $V$ . [Source: Steyn-Ross (2002)]

### 3.6 The Liley neuron

Liley (2002) developed a population-based neural model in order to reproduce the scalp measured EEG signal generated by a macrocolumn. He assumes that each macrocolumn consists of two “neurons”. The first “neuron” is a spatial average representing the population of excitatory neurons. It has (roughly) a pyramidal shape with axon-dendrites axes aligned parallel to each other and perpendicular to the scalp. The spatially averaged current and potential can now be easily detected by the EEG electrodes because the current dipoles produced by excitatory synaptic events will be reinforced. The second “neuron” represents the population of inhibitory neurons inside the macrocolumn, having roughly spherical symmetry, with random dendrite and axon orientations. The current dipoles produced by this inhibitory “neuron” tend to cancel, so EEG electrodes do not detect inhibitory synaptic events.

The potential activity detected at the scalp using the EEG electrodes arises from the slow fluctuations in the spatial average representing the population of excitatory neurons ( $V_e$ ).  $V_e$  is assumed to be directly proportional to the scalp EEG, while  $V_i$  (inhibitory population average voltage) has no “direct” effect on the EEG. However,  $V_i$  does moderate the behaviour of  $V_e$ , thus having an “indirect” effect on the EEG.

There are two types of connections in Liley neural model: short-range (within

macrocolumn) and long-range (in/out of the macrocolumn). For the short range, all of the synaptic types are allowed:  $e \rightarrow k$  or  $i \rightarrow k$  (where  $k$  is either excitatory  $e$  or inhibitory  $i$  neuron). The long-range however, allows only two synaptic types  $e \rightarrow e$  or  $e \rightarrow i$ . Fig. 3.6 shows the elements of Liley's neural model.

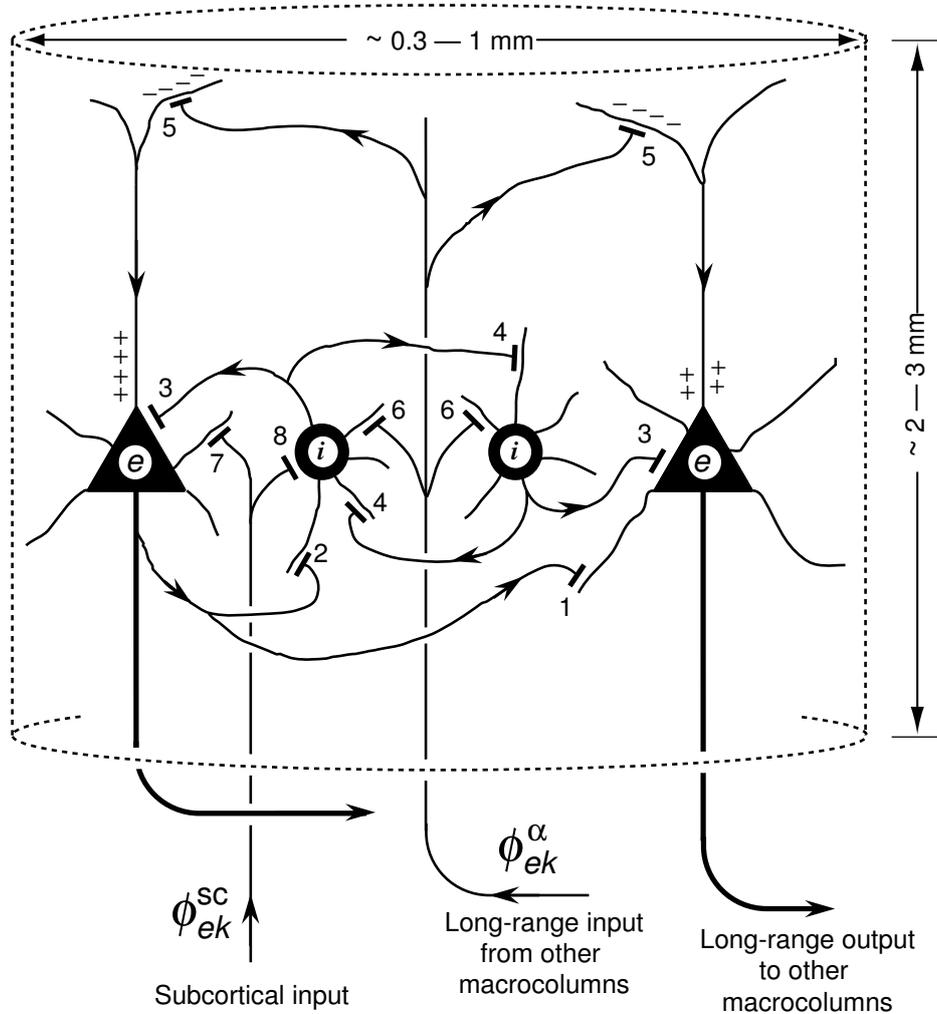


Figure 3.6: Schematic wiring diagram for a cortical macrocolumn showing two excitatory neurons (triangles) and two inhibitory neurons (circles). [Source: Steyn-Ross *et al.* (2005)]

### 3.7 Mapping from Tuckwell $\rightarrow$ Liley

Liley's model can be viewed as a generalisation of the Tuckwell subthreshold neuron. This means the first neuron represents the spatially averaged population of excitatory neurons, while the second one represents the population of inhibitory neurons as drawn in Fig. 3.7. The list below shows how the remappings from Tuckwell to Liley population neurons (Steyn-Ross, 2002):

Tuckwell neuron	→	Liley “neurons”
$V$	→	$h_e, h_i$ membrane voltage
$V_E$	→	$h_e^{\text{rev}}$ excitatory reversal potential = +45mV
$V_I$	→	$h_i^{\text{rev}}$ inhibitory reversal potential = -90mV
$V_{\text{rest}}$	→	$h_e^{\text{rev}}, h_i^{\text{rev}}$ resting potential = -70mV
$C$	→	$C_e, C_i$ membrane capacitance
$g_E$	→	$g_{ee}, g_{ie}$ conductance for excitatory-reversal battery
$g_I$	→	$g_{ei}, g_{ii}$ conductance for inhibitory-reversal battery
$g_{\text{rest}}$	→	$g_e^{\text{rest}}, g_i^{\text{rest}}$ conductance for resting-potential battery

Hence the  $h_{e,i}$  equations of motion for the Liley mapped circuit in Fig. 3.7 are:

$$C_e \frac{dh_e}{dt} = (h_e^{\text{rev}} - h_e)g_{ee} + (h_i^{\text{rev}} - h_e)g_{ie} + (h_e^{\text{rest}} - h_e)g_e^{\text{rest}} \quad (3.11)$$

$$C_i \frac{dh_i}{dt} = (h_e^{\text{rev}} - h_i)g_{ei} + (h_i^{\text{rev}} - h_i)g_{ii} + (h_i^{\text{rest}} - h_i)g_i^{\text{rest}} \quad (3.12)$$

Similarly, the membrane time constant becomes:

$$\tau_e = \frac{C_e}{g_e^{\text{rest}}} \quad (3.13)$$

$$\tau_i = \frac{C_i}{g_i^{\text{rest}}} \quad (3.14)$$

Thus dividing Eqs. (3.11) and (3.12) by  $g_e^{\text{rest}}$  and  $g_i^{\text{rest}}$  yields:

$$\tau_e \frac{dh_e}{dt} = (h_e^{\text{rev}} - h_e) + (h_i^{\text{rev}} - h_e) \frac{g_{ee}}{g_e^{\text{rest}}} + (h_i^{\text{rest}} - h_e) \frac{g_{ie}}{g_e^{\text{rest}}} \quad (3.15)$$

$$\tau_i \frac{dh_i}{dt} = (h_i^{\text{rev}} - h_i) + (h_e^{\text{rev}} - h_i) \frac{g_{ei}}{g_i^{\text{rest}}} + (h_i^{\text{rest}} - h_i) \frac{g_{ii}}{g_i^{\text{rest}}} \quad (3.16)$$

Liley’s final mapping replaced the conductance ratios with voltage ratios as in the following equations:

$$\tau_e \frac{dh_e}{dt} = (h_e^{\text{rev}} - h_e) + (h_i^{\text{rev}} - h_e) \frac{V_{ee}(h_e)}{h_e^{\text{rev}} - h_e^{\text{rest}}} + (h_i^{\text{rest}} - h_e) \frac{V_{ie}(h_i)}{h_i^{\text{rev}} - h_e^{\text{rest}}} \quad (3.17)$$

$$\tau_i \frac{dh_i}{dt} = (h_i^{\text{rev}} - h_i) + (h_e^{\text{rev}} - h_i) \frac{V_{ei}(h_e)}{h_e^{\text{rev}} - h_i^{\text{rest}}} + (h_i^{\text{rest}} - h_i) \frac{V_{ii}(h_i)}{h_i^{\text{rev}} - h_i^{\text{rest}}} \quad (3.18)$$

The double-subscripting on the four synaptic conductances ( $g_{jk}, j, k \in \{i, e\}$ ) appeared in Fig. 3.7, means that the two circuits are very strongly coupled via a local feedback. This feedback is represented in Fig. 3.6 by two nonlinear (sigmoidal) voltage to spike-rate converters (triangles)  $S_e(V_e)$  and  $S_i(V_i)$ . They couple back the excitatory and inhibitory output potentials  $V_{e,i}$  to the four  $g_{jk}(j, k \in \{i, e\})$  synaptic

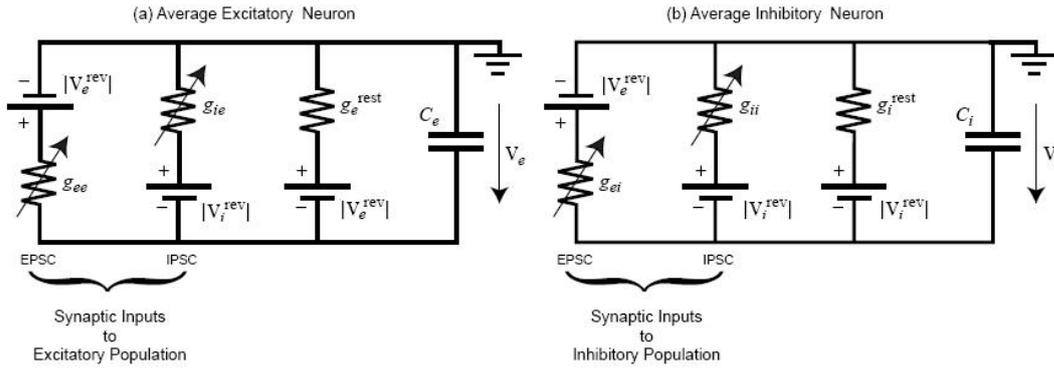


Figure 3.7: Shows excitatory (left) and inhibitory (right) neural populations for Liley’s generalisation of Tuckwell subthreshold neuron. [Source: Steyn-Ross (2002)]

input conductances. Here, the four possible local feedbacks are allowed:  $e \rightarrow k$  or  $i \rightarrow k$  (where  $k \in \{i, e\}$ ). Therefore, four multiplicative scale factors ( $N_{jk}^\beta$  boxes) are needed to indicate the degree of local inter-connectedness between the excitatory and inhibitory populations within the macrocolumn. Liley’s model with the local feedbacks are shown in Fig. 3.8.

We complete Liley’s neural equivalent circuit by considering all of the three inputs to the macrocolumn conductances: local, long-distance cortical, and subcortical. The full equivalent circuit appears in Fig.3.9 as a continuous 1D line of neural mass representing the links inside the macrocolumn. The three inputs are combined at the  $\oplus$  summing point to determine the  $g_{jk}$  synaptic conductances. For simplicity, the “batteries” (reversal and resting potentials) have been omitted in this circuit. Similarly, only the independent subcortical inputs  $P_{jk}$  of the middle macrocolumn were shown instead of drawing them for every macrocolumn.

### 3.8 Steyn-Ross model

The Steyn-Ross model, is an adaption of Liley’s one-dimensional cortical equations was designed to predict the brain reactions to general anaesthetic. In this thesis, we will follow Steyn-Ross *et al.* (2005) in describing the equations.

#### 3.8.1 Integral equations for the macrocolumn

As mentioned previously, at resting state, there is no synaptic input. Therefore the soma voltage of the neuron population  $e$  or  $i$ , will be fixed at  $V_{e,i}^{\text{rest}} \approx -64$  mV (resting voltage). This value remains fixed as the concentration differences of  $\text{Na}^+$  and  $\text{K}^+$  is maintained across the cell membrane. When the dendritic tree receives a spike input, it travels all the way to the soma where it is summed with other synaptic inputs, causing a time-dependent variation about  $V_{e,i}^{\text{rest}}$ . See Table 1 for full list of macrocolumn model constants.

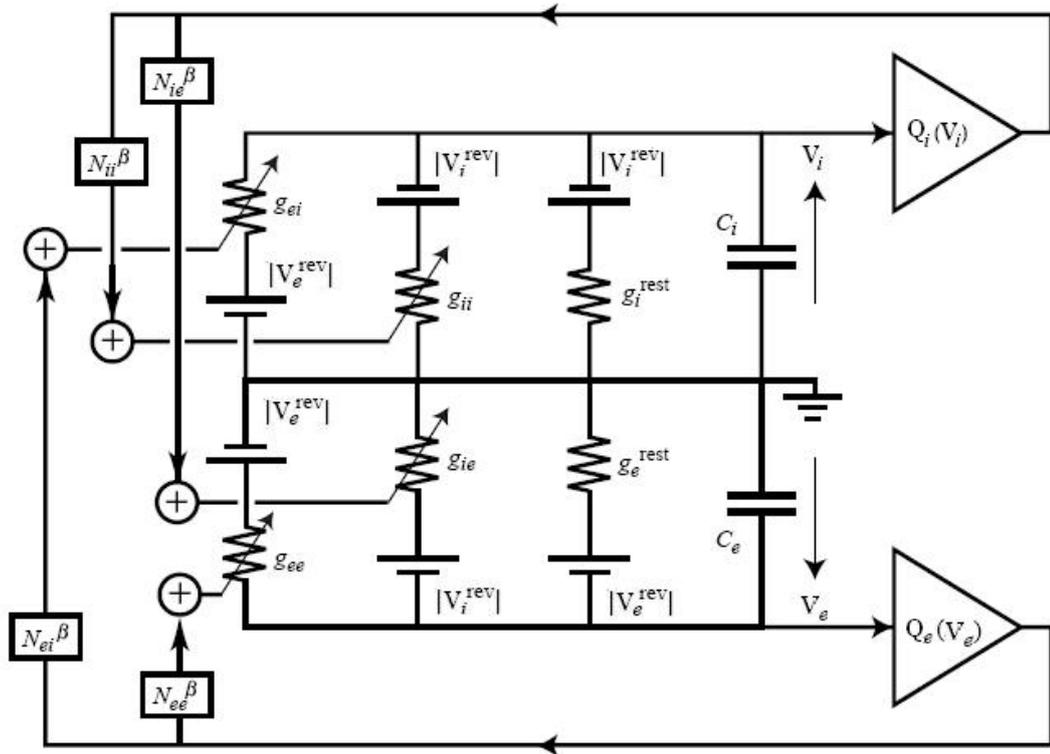


Figure 3.8: Tuckwell → Liley neural model with local feedbacks coupling the two populations together. [Source: Steyn-Ross (2002)]

Table 3.2: Symbol definitions and model constants for the Liley equations. [Source: Steyn-Ross *et al.* (2005)]

Symbol	Description	Value	Unit
$\tau_{e,i}$	membrane time constant	0.050, 0.050	s
$V_{e,i}^{\text{rev}}$	cell reversal potential	0, -70	mV
$V_{e,i}^{\text{rest}}$	cell resting potential	-64, -64	mV
$\rho_{e,i}$	synaptic gain	$(1.00, -1.05) \times 10^{-3}$	mV.s
$N_{ee,ei}^{\alpha}$	long-range $e \rightarrow k$ connections	3710, 3710	-
$N_{ee,ei}^{\beta}$	local $e \rightarrow k$ synaptic connections	410, 410	-
$N_{ie,ii}^{\beta}$	local $i \rightarrow k$ synaptic connections	800, 800	-
$N_{ee,ei}^{\text{sc}}$	subcortical $e \rightarrow k$ synaptic connections	50, 50	-
$N_{ie,ii}^{\text{sc}}$	subcortical $i \rightarrow k$ synaptic connections	50, 50	-
$\epsilon$	axonal conduction speed	9	$\text{ms}^{-1}$
$\Lambda_{ee,ei}$	axonal inverse-length scale	40, 40	$\text{m}^{-1}$
$\gamma_{ee,ei}$	rate constant for $e \rightarrow k$ synaptic input	70, 70	$\text{s}^{-1}$
$\gamma_{ie,ii}$	rate constant $i \rightarrow k$ synaptic connections	58.6, 58.6	$\text{s}^{-1}$
$Q_{e,i}^{\text{max}}$	maximum firing rate for sigmoid	30, 60	$\text{s}^{-1}$
$\theta_{e,i}$	inflection-point voltage for sigmoid	-58.5, -58.5	mV
$\sigma_{e,i}$	sigmoid width	4, 6	mV

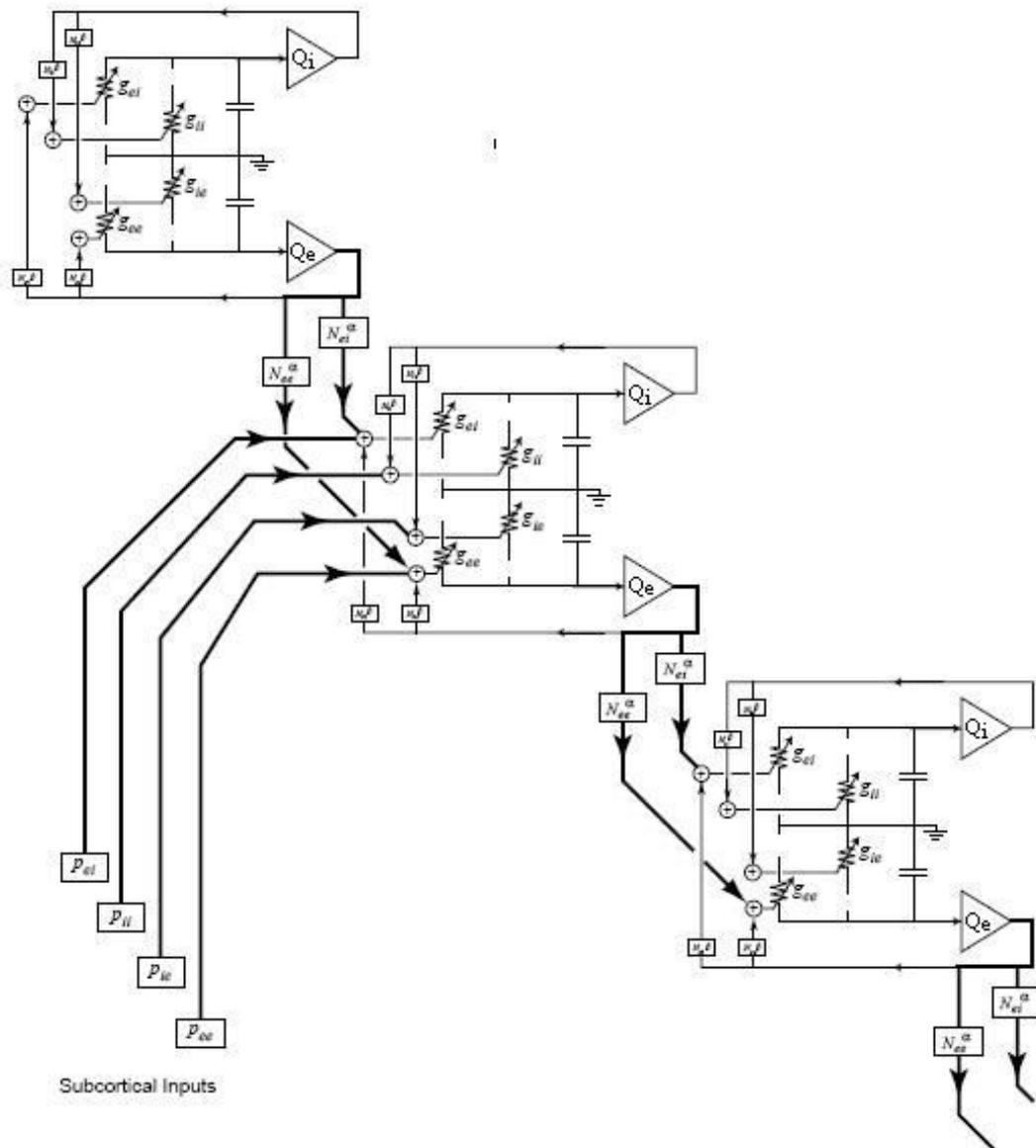


Figure 3.9: Full circuit diagram of Tuckwell  $\rightarrow$  Liley neural model with the three inputs to the macrocolumn: local, long-distance cortical, and subcortical. The two populations are coupled via local feedback. [Source: Steyn-Ross (2002)]

$$V_k(t) = V_k^{\text{rest}} + \int_{-\infty}^t L_k(t-t')[E_k(t') + I_k(t')]dt', \quad \text{where } k = e, i \quad (3.19)$$

Here,  $L_k(t)$  is the impulse response of the soma:

$$L_k(t) = \begin{cases} \frac{1}{\tau_k} \exp[-t/\tau_k], & t \geq 0, \\ 0, & t < 0 \end{cases} \quad (3.20)$$

$E_k(t)$  and  $I_k(t)$  are the excitatory and inhibitory voltage inputs to the soma:

$$E_k(t) = \rho_e \psi_{ek}(t) \Phi_{ek}(t), \quad \rho > 0 \quad (3.21)$$

$$I_k(t) = \rho_i \psi_{ik}(t) \Phi_{ik}(t), \quad \rho < 0 \quad (3.22)$$

where:

$\rho$  is the synaptic strength (mV/s),

$\Phi$  is the input flux rate spikes  $\text{s}^{-1}$ ,

$\psi_{jk}$  are dimensionless weighting function that scale the effectiveness of the  $\Phi_{jk}$  (input flux). It is in proportion to the difference between the cell voltage  $V_k$  of the receiving neuron and the relevant ionic reversal potential  $V_j^{\text{rev}}$ ,

$$\psi_{jk}(t) = \frac{V_j^{\text{rev}} - V_k(t)}{V_j^{\text{rev}} - V_k^{\text{rest}}} \quad (3.23)$$

Here,  $V_e^{\text{rev}} = 0$  mV for excitatory receptors, and  $V_i^{\text{rev}} = -70$  mV for inhibitory receptors.

The total excitatory/inhibitory input flux  $\Phi_{ek}/\Phi_{ik}$  at a neuron  $k$  can be represented as the temporal convolution of the dendrite impulse response  $H(t)$  with the pre-synaptic spike rate  $\phi_{ek}/\phi_{ik}$ , scaled by the number of one-way  $e, i \rightarrow k$  synaptic connections  $N_{ek}/N_{ik}$ ,

$$\Phi_{ek}(t) = \int_{-\infty}^t H_{ek}(t-t')[N_{ek}^{\beta} \phi_{ek}^{\beta}(t') + \phi_{ek}^{\text{sc}}(t') + N_{ek}^{\alpha} \phi_{ek}^{\alpha}(t')]dt' \quad (3.24)$$

$$\Phi_{ik}(t) = \int_{-\infty}^t H_{ik}(t-t')[N_{ik}^{\beta} \phi_{ik}^{\beta}(t') + \phi_{ik}^{\text{sc}}(t')]dt' \quad (3.25)$$

The subscripts identify distinct spike-rate contributions from local ( $\beta$ : within the macrocolumn) distant ( $\alpha$ : cortico-cortical excitatory connections from other macrocolumn) and subcortical (sc: input from subcortical structures such as thalamus and brain-stem). There is no cortico-cortical  $\phi_{ik}^{\alpha}$  term, because there are no distant inhibitory connections from other macrocolumns.

As discussed earlier, each incoming action potential releases a chemical signal into the synaptic cleft called the neurotransmitter. The neurotransmitter changes

the dendritic conductance of the post-synaptic neuron causing a momentary change in voltage as a result of charge transfer through the opened ion channels.

Following the Tuckwell model, the time-course for the momentary postsynaptic potential can be written as an “alpha” function dendrite impulse response  $H_{jk}(t)$ :

$$H_{jk}(t) = \begin{cases} t\gamma_{jk}^2 \exp[-t\gamma_{jk}] & t \geq 0, \\ 0, & t < 0. \end{cases} \quad (3.26)$$

A two-component neuron model is described by these equations: The soma dynamics (in Eq.(3.17)) are determined by the  $L_k(t)$  exponential response function, while the dendrite (synaptic) dynamics in  $\phi_{ek}/\phi_{ik}$  are determined by the  $H_{jk}(t)$  alpha function impulse response Eq.(3.26).

### 3.8.2 Differential equations for the macrocolumn

For the purpose of locating the steady states of the cortex model and determining their stability properties, the integral equations of  $V_k(t)$  and  $\phi_{ek}/\phi_{ik}$  can be transferred to differential equations. To do so, we need to take the time-derivative of  $k = e, i$  convolution indexed by Eq. (3.19). This can be achieved by using the calculus theorem for differentiation under an integral with time-dependent end-points. The result will be a pair of first-order equations of motion for  $V_{e,i}$  (the macrocolumn-average excitatory and inhibitory soma voltages),

$$\tau_e \frac{dV_e}{dt} = V_e^{\text{rest}} - V_e(t) + \rho_e \psi_{ee}(t) \Phi_{ee}(t) + \rho_i \psi_{ie}(t) \Phi_{ie}(t) \quad (3.27)$$

$$\tau_i \frac{dV_i}{dt} = V_i^{\text{rest}} - V_i(t) + \rho_e \psi_{ei}(t) \Phi_{ei}(t) + \rho_i \psi_{ii}(t) \Phi_{ii}(t) \quad (3.28)$$

In order to get differential equations for the rate-change of the  $\Phi_{jk}$  (input flux terms), the convolutions of Eqs. (3.24, 3.25) need to be differentiated twice using the calculus theorem. This implies the following pairs of equations: For excitatory-input flux,

$$\left(\frac{d}{dt} + \gamma_{ee}\right)^2 \Phi_{ee} = [N_{ee}^\alpha \phi_{ee}^\alpha(t) + N_{ee}^\beta Q_e(t) + \phi_{ee}^{sc}(t)] \gamma_{ee}^2 \quad (3.29)$$

$$\left(\frac{d}{dt} + \gamma_{ie}\right)^2 \Phi_{ie} = [N_{ie}^\beta Q_i(t)] \gamma_{ie}^2 \quad (3.30)$$

while the inhibitory-input flux convolutions yield,

$$\left(\frac{d}{dt} + \gamma_{ii}\right)^2 \Phi_{ii} = [N_{ii}^\beta Q_i(t)] \gamma_{ii}^2 \quad (3.31)$$

$$\left(\frac{d}{dt} + \gamma_{ei}\right)^2 \Phi_{ei} = [N_{ei}^\alpha \phi_{ei}^\alpha(t) + N_{ei}^\beta Q_e(t) + \phi_{ei}^{sc}(t)] \gamma_{ei}^2 \quad (3.32)$$

### 3.8.3 Synaptic flux densities

#### Intra-macrocolumn fluxes

The fluxes  $\phi_{jk}(t)$  within the macrocolumn can be written as:

$$\phi_{jk}^\beta(\mathbf{r}, t) = \int f_{jk}^\beta(\mathbf{r} - \mathbf{r}') Q_j(\mathbf{r}', t) d\mathbf{r}' \quad (3.33)$$

where  $f_{jk}^\beta$  is the distribution function that defines the exponential decay over space for the density of synaptic connection inside the macrocolumn:

$$f_{jk}^\beta(\mathbf{r} - \mathbf{r}') = \begin{cases} \frac{\lambda_{jk}}{2} \exp(-\lambda_{jk}|x - x'|) & (1D), \\ \frac{\lambda_{jk}^2}{2\pi} \exp(-\lambda_{jk}|\mathbf{r} - \mathbf{r}'|) & (2D) \end{cases} \quad (3.34)$$

with  $\lambda_{jk}$  being the inverse-length scale for connection strength. Note that these distribution functions are normalised:

$$\int f_{jk}^\beta(\mathbf{r}) d\mathbf{r} = 1 \quad (3.35)$$

The  $Q_j(\mathbf{r}', t)$  mentioned in Eq. (3.33) is the empirical mapping from membrane voltage to neuronal firing rate defined as:

$$Q_j(\mathbf{r}', t) = \frac{Q_j^{\max}}{1 + \exp[-(V_j(\mathbf{r}', t) - \theta_j)C/\sigma_j]} \quad (3.36)$$

where:

$\theta_j$  is the firing population-average threshold voltage,

$\sigma_j$  is the measure of the empirical sigmoidal spread,

$Q_j^{\max}$  is the maximum firing rate for the macrocolumn,

$C = \pi/\sqrt{3}$  allows the  $\sigma_j$  to be interpreted as the standard deviation of the sigmoid-derivative curve  $dQ_j/dV_j$ .

Now assuming a spatial coarse-graining results in the following equation:

$$\phi_{jk}^\beta(\mathbf{r}', t) = Q_j(t) \quad (3.37)$$

#### Long-range, cortico-cortical fluxes

The long range flux is represented by the following equation:

$$\phi_{jk}^\alpha(\mathbf{r}, t) = \int d\mathbf{r}' n_{ek}^\alpha(\mathbf{r} - \mathbf{r}') Q_e(\mathbf{r}', t - |\mathbf{r} - \mathbf{r}'|/v) \quad (3.38)$$

where  $v$  denotes the average axonal conduction speed.

The long-range flux ( $\phi_{jk}^\alpha(\mathbf{r}, t)$ ) is described by a wave equation whose form will depend on the definition for the long-range synaptic distribution function ( $n_{ek}^\alpha(\mathbf{r} - \mathbf{r}')$ ), for example,

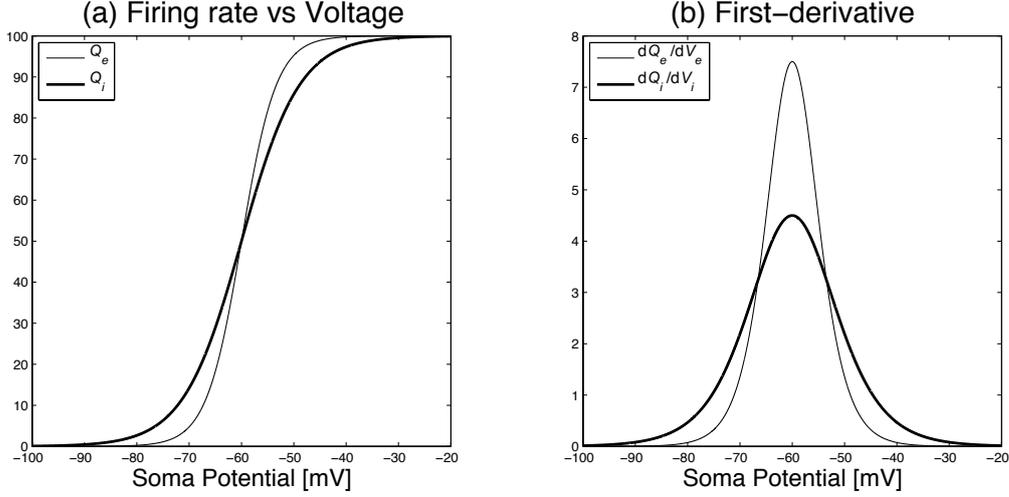


Figure 3.10: Sigmoidal function relating firing rate to average soma potential. (a) Excitatory sigmoid  $Q_e$  (thin curve) and Inhibitory sigmoid  $Q_i$  (thick curve). (b) First derivative of sigmoid functions:  $dQ_e/dV_e$  (thin curve);  $dQ_i/dV_i$  (thick curve).

$$n_{ek}^\alpha(\mathbf{r} - \mathbf{r}') = \begin{cases} \frac{\Lambda_{ek}}{2} \exp(-\Lambda_{ek}|x - x'|) & (1D), \\ \frac{\Lambda_{ek}^2}{2\pi} \exp(-\Lambda_{ek}|\mathbf{r} - \mathbf{r}'|) & (2D) \end{cases} \quad (3.39)$$

From the above equations (Eqs. (3.38) and (3.39)), two different wave equations are derived:

$$\left[ \left( \frac{\partial}{\partial t} + v\Lambda_{ek} \right)^2 - v^2 \frac{\partial^2}{\partial x^2} \right] \phi_{ek}^\alpha = v\Lambda_{ek} \left( \frac{\partial}{\partial t} + v\Lambda_{ek} \right) Q_e(x, t) \quad (1D) \quad (3.40)$$

$$\left[ \left( \frac{\partial}{\partial t} + v\Lambda_{ek} \right)^2 - \frac{3}{2} v^2 \nabla^2 \right] \phi_{ek}^\alpha = v^2 \Lambda_{ek}^2 Q_e(\mathbf{r}, t) \quad (2D) \quad (3.41)$$

For the sake of simplicity, Robinson (1997) applied another 2D synaptic distribution function,

$$n_{ek}^\alpha(\mathbf{r} - \mathbf{r}') = \frac{\Lambda_{ek}^2}{2\pi} K_0(\Lambda_{ek}|\mathbf{r} - \mathbf{r}'|) \quad (2D) \quad (3.42)$$

$$\sim \frac{1}{\sqrt{\Lambda_{ek}R}} \exp(-\Lambda_{ek}R) \quad \text{for large } R \quad (3.43)$$

where  $R = |\mathbf{r} - \mathbf{r}'|$ , and  $K_0$  is a Macdonald function (modified Bessel function of the second kind). This alternative choice leads to a simpler 2D wave equation:

$$\left[ \left( \frac{\partial}{\partial t} + v\Lambda_{ek} \right)^2 - v^2 \nabla^2 \right] \phi_{ek}^\alpha = v^2 \Lambda_{ek}^2 Q_e(\mathbf{r}, t) \quad (2D) \quad (3.44)$$

where the symbol  $\Lambda_{ek}$  indicates the inverse-length scale for long-range connection.

### 3.8.4 Subcortical flux: White-noise driving

There is physiological evidence that the cortex requires a continuous background “wash” of input noise to function normally. This fact motivates us to include white-noise stimulation in this model.

The two excitatory subcortical spike rates  $\phi_{ek}^{\text{sc}}$  appearing in Eqs.(3.29, 3.32) are modelled as a mean level of spike-rate input to which is added Gaussian white noise:

$$\phi_{ek}^{\text{sc}} = s_{ek}Q_e^{\text{max}} + \sqrt{s_{ek}Q_e^{\text{max}}}\xi_m(t) \quad m = 1, 2 \quad (3.45)$$

The  $s_{ek}$  are dimensionless scale-factors that determine the level of subcortical driving. Note that the two stochastic terms  $\xi_m(t)$  of the  $\phi_{jk}^{\text{sc}}$  are independent, zero mean, delta-correlated white-noise sources

$$\langle \xi_m(t) \rangle = 0 \quad (3.46)$$

$$\langle \xi_m(t)\xi_n(t') \rangle = \delta_{mn}\delta(t-t') \quad (3.47)$$

### 3.8.5 Summary of stochastic DEs for cortex

Here are a list of model’s differential equations:

For soma voltages:

$$\tau_e \frac{dV_e}{dt} = V_e^{\text{rest}} - V_e(t) + \rho_e \psi_{ee}(t)\Phi_{ee}(t) + \rho_i \psi_{ie}(t)\Phi_{ie}(t) \quad (3.48)$$

$$\tau_i \frac{dV_i}{dt} = V_i^{\text{rest}} - V_i(t) + \rho_e \psi_{ei}(t)\Phi_{ei}(t) + \rho_i \psi_{ii}(t)\Phi_{ii}(t) \quad (3.49)$$

Synaptic spike-rate inputs:

$$\left(\frac{d}{dt} + \gamma_{ee}\right)^2 \Phi_{ee} = [N_{ee}^\alpha \phi_{ee}^\alpha(t) + N_{ee}^\beta Q_e(t) + s_{ee} \phi_{ee}^{\text{sc}}(t)] \gamma_{ee}^2 \quad (3.50)$$

$$\left(\frac{d}{dt} + \gamma_{ie}\right)^2 \Phi_{ie} = [N_{ie}^\beta Q_i(t)] \gamma_{ie}^2 \quad (3.51)$$

$$\left(\frac{d}{dt} + \gamma_{ii}\right)^2 \Phi_{ii} = [N_{ii}^\beta Q_i(t)] \gamma_{ii}^2 \quad (3.52)$$

$$\left(\frac{d}{dt} + \gamma_{ei}\right)^2 \Phi_{ei} = [N_{ei}^\alpha \phi_{ei}^\alpha(t) + N_{ei}^\beta Q_e(t) + s_{ei} \phi_{ei}^{\text{sc}}(t)] \gamma_{ei}^2 \quad (3.53)$$

The 2D wave equation (Robinson):

$$\left[\left(\frac{\partial}{\partial t} + v\Lambda_{ek}\right)^2 - v^2 \nabla^2\right] \phi_{ek}^\alpha = v^2 \Lambda_{ek}^2 Q_e(\mathbf{r}, t) \quad (2D) \quad (3.54)$$

Voltage-to-firing rate sigmoid:

$$Q_j(\mathbf{r}', t) = \frac{Q_j^{\max}}{1 + \exp[-(V_j(\mathbf{r}', t) - \theta_j)C/\sigma_j]} \quad (3.55)$$

Subcortical fluxes:

$$\phi_{ek}^{\text{sc}} = s_{ek}Q_e^{\max} + \sqrt{s_{ek}Q_e^{\max}} \cdot \xi_m(t), \quad m=1,2 \quad (3.56)$$

In the next chapter, we will search for the number of the macrocolumn steady states after introducing the effect of NMDA. Then a theoretical stability prediction of the stationary states will be discussed by analysing the eigenvalues of the system.

## Chapter 4

# NMDA-modulated mean-field cortical model

### 4.1 Modelling NMDA

Chapter 3 discussed existing mean-field models which are characterised by excitatory neurotransmitters such as AMPA, and inhibitory neurotransmitters such as GABA. We now introduce the effects of NMDA (another excitatory neurotransmitter) by following Jahr and Stevens (1990). They normalise the conductance of NMDA-receptor as follows:

$$g(V, C) = \frac{1}{1 + kC \exp^{-aV}} \quad (4.1)$$

where:

$V$  is the excitatory voltage (mV),

$C$  is the magnesium concentration (mM) which lies in the range  $1 \leq C \leq 10$ ,

$k = \frac{1}{3.57} \text{ (mM)}^{-1}$ ,

$a = 0.062 \text{ (mV)}^{-1}$ .

We thus replace  $\rho_e$  in the cortical equations of Chapter 3 with:

$$\rho_e \longrightarrow \rho_e (1 + kC \exp^{-aV})^{-1} \quad (4.2)$$

Fig. 4.1 illustrates the relationship of NMDA responses between the conductance ( $g$ ) and the voltage ( $V$ ) at 10 magnesium concentrations ( $C$ ). For all 10 values of  $C$ , magnesiums blockade is completely overcome at voltages more positive than about +80 mV.

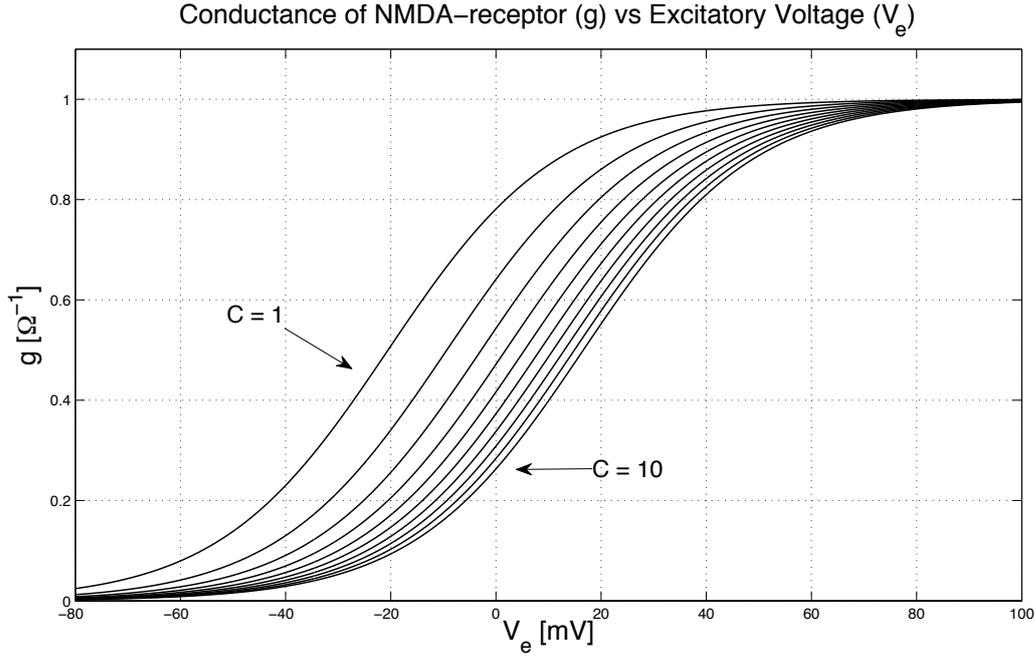


Figure 4.1: Voltage dependence of NMDA conductance  $g$  for 10 values of magnesium concentrations  $C$ . Blockade is completely overcome at membrane voltages more positive than about +80 mV. [ $C = 1, 2, 3, \dots, 10$ ].

## 4.2 Stationary states

Our method relies on the investigation of the behaviour of small perturbations around the stationary states of the cortex. In order to compute the stationary soma voltages  $V_e^0$  and  $V_i^0$  as a function of neurotransmitter effects and subcortical driving, all time and space derivatives in Eqs.(3.48) to (3.54) need to be set equal to zero ( $d/dt = d/dx = 0$ ) and also the noise = 0.

To simplify the set of Eqs.(3.48) to (3.54) we take the advantage of the parameter-value symmetries shown in Table 2:

$$\begin{aligned}
 V_e^{\text{rest}} &= V_i^{\text{rest}} \\
 \tau_e &= \tau_i \\
 \gamma_{ee} &= \gamma_{ei} = \gamma_e \\
 \gamma_{ie} &= \gamma_{ii} = \gamma_i \\
 \Lambda_{ee} &= \Lambda_{ei} = \Lambda_e \\
 N_{ee}^\alpha &= N_{ei}^\alpha = N_e^\alpha \\
 N_{ee}^\beta &= N_{ei}^\beta = N_e^\beta \\
 \phi_{ee}^{\text{sc}} &= \phi_{ei}^{\text{sc}} = \phi_e^{\text{sc}}
 \end{aligned}$$

Thus, as can be seen from Eqs.(3.48) to (3.54) the pairs  $(\Phi_{ee}, \Phi_{ei})$  and  $(\Phi_{ie}, \Phi_{ii})$  obey the same differential equations as do the pair  $(\phi_{ee}^\alpha, \phi_{ei}^\alpha)$ . We thus define:

$$\begin{aligned}\phi_{ee}^\alpha &= \phi_{ii}^\alpha \equiv \phi_e^\alpha \\ \Phi_{ee} &= \Phi_{ei} \equiv \Phi_e \\ \Phi_{ie} &= \Phi_{ii} \equiv \Phi_i\end{aligned}$$

Applying the above simplifications reduces our cortical system to five differential equations, as follows:

Two first-order (3.48 and 3.49) for the soma voltages:

$$\tau_e \frac{dV_e}{dt} = V_e^{\text{rest}} - V_e(t) + \rho_e \psi_{ee}(t) \Phi_e(t) + \rho_i \psi_{ie}(t) \Phi_i(t) \quad (4.3)$$

$$\tau_i \frac{dV_i}{dt} = V_i^{\text{rest}} - V_i(t) + \rho_e \psi_{ei}(t) \Phi_e(t) + \rho_i \psi_{ii}(t) \Phi_i(t) \quad (4.4)$$

two second-order (3.50 and 3.53) synaptic spike-rate inputs:

$$\left(\frac{d}{dt} + \gamma_{ee}\right)^2 \Phi_e = [N_e^\alpha \phi_e^\alpha(t) + N_e^\beta Q_e(t) + s \phi_e^{\text{sc}}(t)] \gamma_e^2 \quad (4.5)$$

$$\left(\frac{d}{dt} + \gamma_i\right)^2 \Phi_i = [N_i^\beta Q_i(t)] \gamma_i^2 \quad (4.6)$$

and one second-order for the 2-D wave equation (Robinson):

$$\left[\left(\frac{\partial}{\partial t} + v \Lambda_e\right)^2 - v^2 \nabla^2\right] \phi_e^\alpha = v^2 \Lambda_e^2 Q_e(r, t) \quad (2D) \quad (4.7)$$

The second-order equations can be reformulated as pairs of first-order DEs. This results in a set of eight ODEs as follows:

$$\tau_e \frac{dV_e}{dt} = V_e^{\text{rest}} - V_e(t) + \rho'_e \psi_{ee}(t) \Phi_e(t) + \rho'_i \psi_{ie}(t) \Phi_i(t) \quad (4.8)$$

$$\tau_i \frac{dV_i}{dt} = V_i^{\text{rest}} - V_i(t) + \rho'_e \psi_{ei}(t) \Phi_e(t) + \rho'_i \psi_{ii}(t) \Phi_i(t) \quad (4.9)$$

$$\frac{d\Phi_e}{dt} = M_e \quad (4.10)$$

$$\frac{dM_e}{dt} = -2\gamma_e M_e + [-\Phi_e + N_e^\alpha \phi_e^\alpha(t) + N_e^\beta Q_e(t) + s \phi_e^{\text{sc}}(t)] \gamma_e^2 \quad (4.11)$$

$$\frac{d\Phi_i}{dt} = M_i \quad (4.12)$$

$$\frac{dM_i}{dt} = -2\gamma_i M_i + [-\Phi_i + N_i^\beta Q_i(t)] \gamma_i^2 \quad (4.13)$$

$$\frac{d\phi_e^\alpha}{dt} = \Omega_e \quad (4.14)$$

$$\frac{d\Omega_e}{dt} = -2v\Lambda_e\Omega_e + [v^2\nabla^2 - v^2\Lambda_e^2]\phi_e^\alpha + v^2\Lambda_e^2Q_e(t) \quad (4.15)$$

where:

$$\begin{aligned} \rho_e' &= \lambda_e\rho_e(1 + kC \exp^{-aV_e})^{-1}, \\ \rho_i' &= \lambda_i\rho_i, \\ M_e, M_i \text{ and } \Omega_e &\text{ are ancillary variables.} \end{aligned}$$

To find the stationary states from the eight ODEs we set all time derivatives in the long time limit to zero ( $d/dt = 0$ ), also we do the same for the space derivatives ( $\nabla^2 = 0$ ) and the noise terms. Then the following equations are obtained:

$$(\phi_e^\alpha)^0 = Q_e^0 \quad (4.16)$$

$$(\Phi_i)^0 = N_i^\beta Q_i^0 \quad (4.17)$$

$$(\Phi_e)^0 = (N_e^\alpha + N_e^\beta)Q_e^0 + s\phi_e^{\text{sc}} \quad (4.18)$$

Now substitute these into Eqs (4.8), (4.9) to find the stationary voltages  $V_e^0, V_i^0$ .

The resulting pair of equations cannot be solved analytically for  $V_e$  and  $V_i$ . Hence a numerical scheme has to be implemented.

### 4.3 Numerical solution

In this implementation we assume full conformal symmetry, thus at steady state  $V_e^0 = V_i^0$ , but  $Q_e^0 \neq Q_i^0$ . We model anaesthetic drug effect by scaling  $\rho_e$ , the synaptic strength for excitatory spike inputs, by a dimensionless scale factor  $\lambda_e$ :  $\rho_e \rightarrow \lambda_e\rho_e$  (EPSP area). Also, we do the same for IPSP area, by scaling  $\rho_i$  (the synaptic strength for inhibitory spike inputs) by  $\lambda_i$  (dimensionless scale factor):  $\rho_i \rightarrow \lambda_i\rho_i$ . There are four parameters used to model the anaesthetic effect: Mg-concentration (mM) determines NMDA scaling of  $\rho_e$  (EPSP area);  $\lambda_e$  gives independent scaling of  $\rho_e$  (EPSP area);  $\lambda_i$  scales  $\rho_i$  (IPSP area);  $s$  scales  $ee$  and  $ei$  subcortical flux.

We use `nmda_ss_finder.m` code (written by D. A. Steyn-Ross) to locate the steady states. The four parameters used to define the anaesthetic effect are:  $C$ ,  $\lambda_e$ ,  $\lambda_i$ ,  $s$  (note that  $C = \text{Mg-conc}$  and  $s = \text{sc}$  in the `nmda_ss_finder.m` code). The first equation  $dV_e/dt$  is written as **F1**, and the second one  $dV_i/dt$  is given the name **F2**. Plotting **F1** vs  $V_{e,i}$  (or  $Q_{e,i}$ ) gives a graph whose zero crossings locate the steady states. By symmetry, the zeros of **F1** will be identical to the zeros of **F2**. This vastly simplifies the search for steady states.

### 4.3.1 Isocline method

The stationary state of the cortex can be determined from the F1 vs  $V_{e,i}$  or F1 vs  $Q_{e,i}$  graphs. According to Wilson and Cowan (1972), these graphs are called nullclines or isoclines (as  $dV_e/dt = 0$ ). The zeros of F1 (or F2) will give the steady states, where the F1 isocline intersects the voltage axis. There can be either one or three stationary states of the cortex depending on the anaesthetic concentration.

In order to observe the shape and behaviour of distribution of steady states for varying anaesthetic effect, one of the four parameters was varied while keeping the other three fixed, resulting in the following observations:

There are three possible isocline shapes, two with one stationary state and one with three as shown in Figures 4.2, 4.3 and 4.4 respectively.

The default values for  $[C, \lambda_e, \lambda_i, s]$  are  $[0.78, 9.0, 1.0, 0.25]$ . Here are the ranges for each parameter and the shape of the isocline for each range:

1. Varying magnesium concentration  $C$  with  $[\lambda_e, \lambda_i, s]$  fixed:  
Three stationary states occur for  $0.65 \lesssim C \lesssim 0.82$  mM (see Fig. 4.4); elsewhere only one stationary state exists (see Figs.4.2 and 4.3)
2. Varying  $\lambda_e$  with  $[C, \lambda_i, s]$  fixed:  
Three steady states can be found for  $8.6 \lesssim \lambda_e \lesssim 10.7$ ; anywhere else only one stationary state exists
3. Varying  $\lambda_i$  with  $[C, \lambda_e, s]$  fixed:  
There are three stationary states for  $0.83 \lesssim \lambda_i \lesssim 1.061$ , and only one stationary state anywhere else
4. Varying subcortical factor  $s$  with  $[C, \lambda_e, \lambda_i]$  fixed:  
There are three stationary states for  $-2.6 \lesssim \lambda_i \lesssim 2.5$ , and only one stationary state anywhere else

Note that negative spike-rates are unphysical, but they were included here to produce a complete S-bend that allows comparison with the other S-bends.

### 4.3.2 Root finding

An alternative way of finding the stationary state of the cortex is to follow an iterative numerical scheme of finding equilibrium soma voltages  $V_{e,i}$  (Steyn-Ross (2002), Section 3.3.1). The graphical results found in the previous section should provide a visual confirmation of the stationary states found by this iterative scheme.

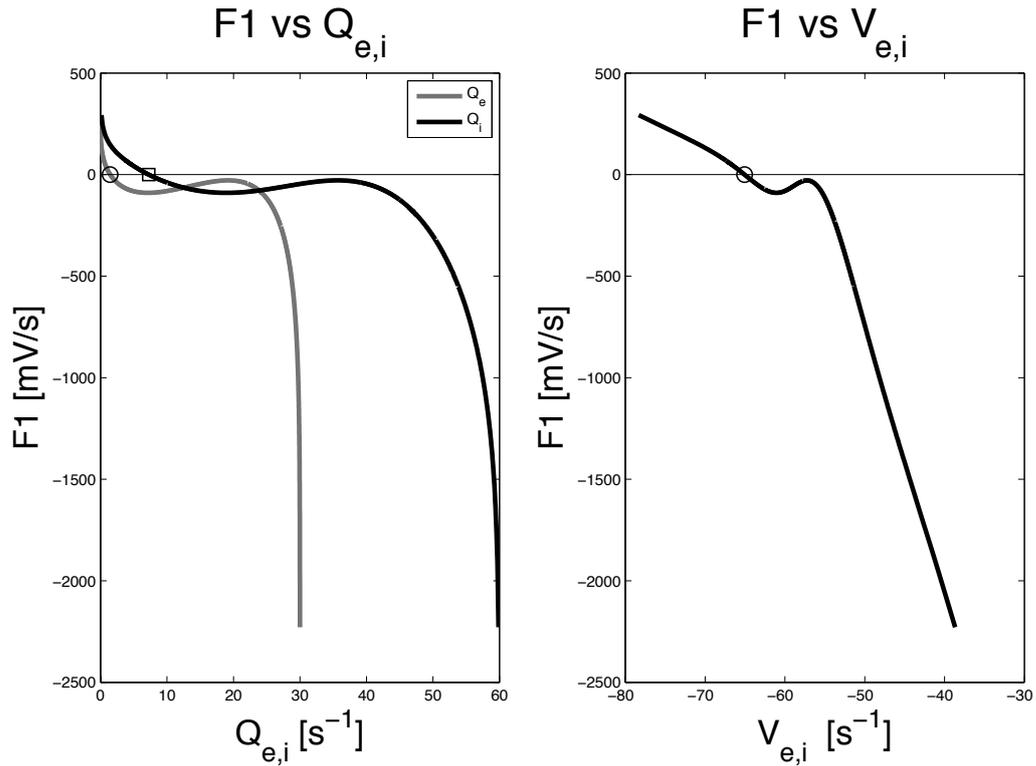


Figure 4.2: Graph of  $F1$  vs  $Q_{e,i}$  (left) and  $F1$  vs  $V_{e,i}$  (right) with isocline showing a single low-firing stationary state. Here,  $[C, \lambda_e, \lambda_i, s] = [0.85, 9.0, 1.0, 0.25]$

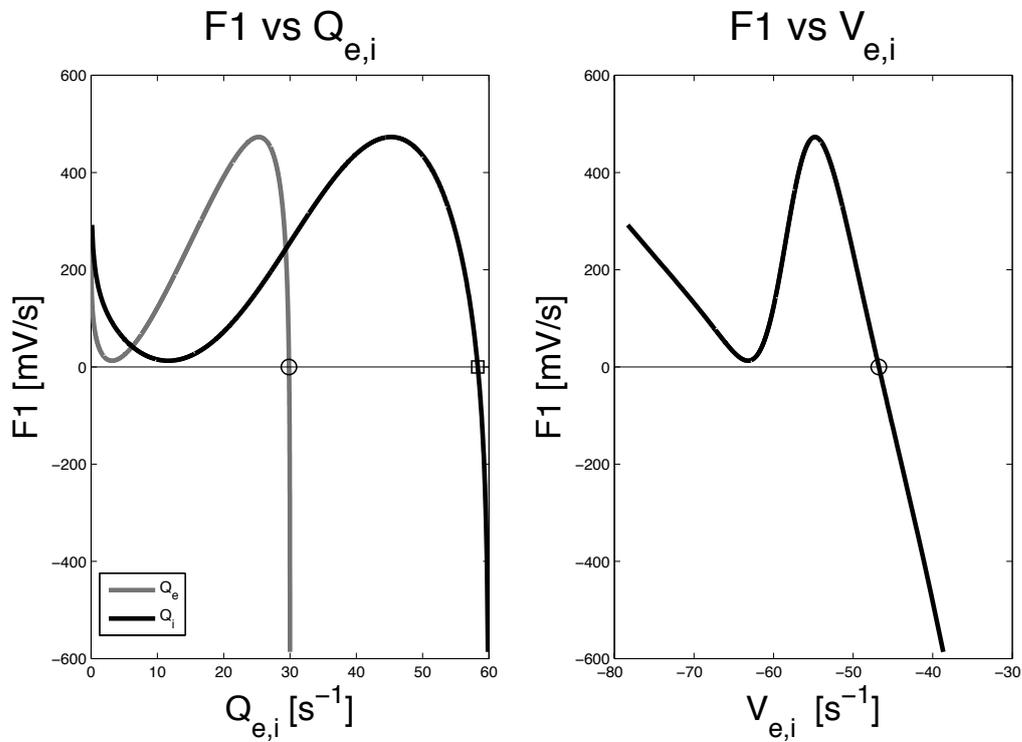


Figure 4.3:  $F1$  vs  $Q_{e,i}$  (left) and  $F1$  vs  $V_{e,i}$  (right); isocline showing a single high-firing stationary state. Here,  $[C, \lambda_e, \lambda_i, s] = [0.6, 9.0, 1.0, 0.25]$

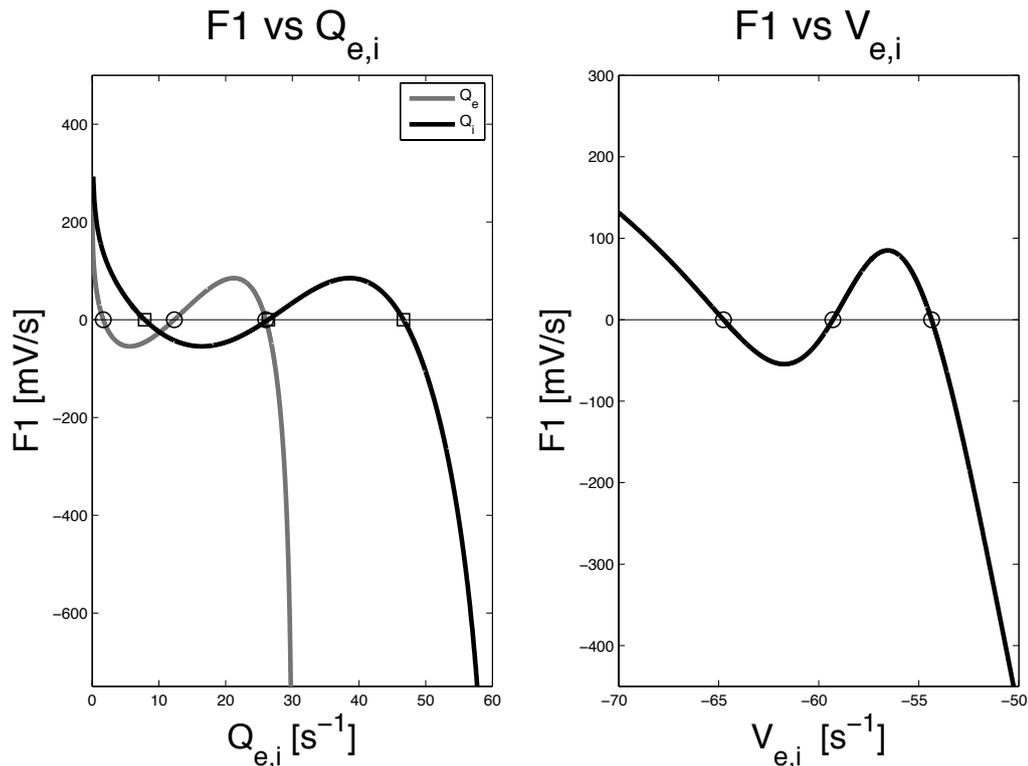


Figure 4.4: F1 vs  $Q_{e,i}$  (left) and F1 vs  $V_{e,i}$  (right) with isocline showing three stationary states. Here,  $[C, \lambda_e, \lambda_i, s] = [0.78, 9.0, 1.0, 0.25]$

In this scheme, the roots are determined by detecting axis crossings as a sign-change in F1. The second step is to form a bracketing interval bounding each root. Finally, each root is “polished” by applying a bisection technique (we used MATLAB’s `fzero` function) to zero-in on the root to within a specified tolerance.

## 4.4 Distribution of roots

In this section we will show the locus of steady-state  $V_{e,i}^0$  soma voltages and  $Q_{e,i}^0$  firing rates as multivalued functions of neurotransmitter effects and subcortical driving, ( $C, \lambda_e, \lambda_i$  and  $s$ ). In each case the roots are distributed in an S-bend shape. Looking vertically at the mid-range of the S-bend, three distinct equilibrium states can be noted, while only one stationary state is located on other regions.

### 4.4.1 Variations in inhibitory neurotransmitter, $\lambda_i$

Fig. 4.5 indicates that for  $\lambda_i \gtrsim 1.06$ , corresponding to strong anaesthetic effect, only one state can be found on the bottom branch with about  $-65$  mV soma voltage  $V_{e,i}$  and around  $1.5$   $s^{-1}$  firing rate for  $Q_e$  and  $7.5$   $s^{-1}$  for  $Q_i$  (low firing rate, *hyperpolarised* state). Also there is a single state on the top branch for  $\lambda_i \lesssim 0.82$ , corresponding to strong anti-anaesthetic effect; here the soma voltage exceeds the sigmoidal inflexion voltage ( $\theta_{e,i} = -58.5$  mV), giving a high firing rate for the neural population of

about  $30 \text{ s}^{-1}$  for  $Q_e$  and  $60 \text{ s}^{-1}$  for  $Q_i$  (*depolarised* state). When the inhibitory effect tends to the zero,  $\lambda_i \rightarrow 0$ , the curve converges at the top-left corner to values close to  $-38 \text{ mV}$  where the neural populations are firing at close to their maximum rates.

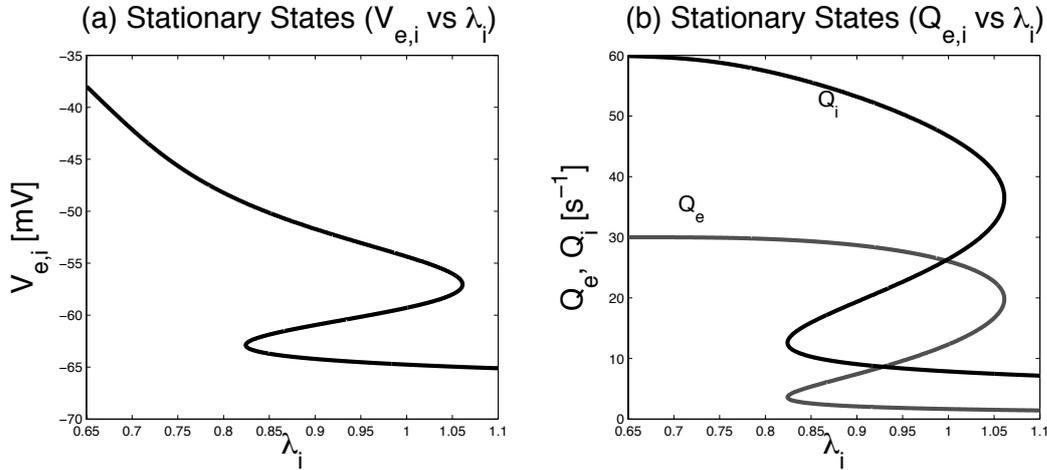


Figure 4.5: (a) Model predictions for the stationary states for soma voltages  $V_{e,i}$  as function of inhibitory neurotransmitter  $\lambda_i$ . (b) Equilibrium firing rates solutions for excitatory  $Q_e$  (grey curve) and inhibitory  $Q_i$  (black curve) as a function of inhibitory neurotransmitter effect. The graph was plotted by varying  $\lambda_i$  and fixing  $[C, \lambda_e, s] = [0.78, 9.0, 0.25]$ .

#### 4.4.2 Variations in subcortical driving $s$

Fig.4.6 shows the steady-state distribution for variable subcortical driving. For  $s \lesssim -2.7$  (bottom branch), where the patient is considered to be strongly anaesthetized, a single state is available with  $V \approx -67 \text{ mV}$  and low firing rate ( $Q_e \approx 0.5 \text{ s}^{-1}$  and  $Q_i \approx 3 \text{ s}^{-1}$ ). When  $s \gtrsim 2.6$  and  $V \approx -52.7 \text{ mV}$ , the firing rate of the patient brain is strong ( $Q_e \approx 28 \text{ s}^{-1}$  and  $Q_i \approx 51 \text{ s}^{-1}$ ). This is on the top-branch where one state existed. For  $s \rightarrow \infty$ , the neural populations tend to their maximum firing rates.

#### 4.4.3 Variations in excitatory neurotransmitter $\lambda_e$

It can be seen from Fig.4.7 that the *hyperpolarised* state occurs when  $\lambda_e \lesssim 8.53$ . This happens on the bottom branch where only a single stationary state is available with approximately the same voltages  $V_{e,i}$  ( $\approx -65 \text{ mV}$ ) and firing rate  $Q_{e,i}$  ( $1.5, 7.5 \text{ s}^{-1}$  respectively) as for  $\lambda_i \gtrsim 1.06$ . The *depolarised* state occurs when  $\lambda_e \gtrsim 10.7$ , corresponding to strong anti-anaesthetic effect. This is on the top branch where we have one steady state and high firing rate ( $Q_e \approx 30 \text{ s}^{-1}$  and  $Q_i \approx 60 \text{ s}^{-1}$ , similar to  $\lambda_i \lesssim 0.82$ ) because the soma voltage is greater than the sigmoid inflexion point. For the excitatory effects, the maximum firing rate of the neural population occurs when  $\lambda_e \rightarrow \infty$ ; this is on the top-right corner where  $V \approx -48.3 \text{ mV}$ .

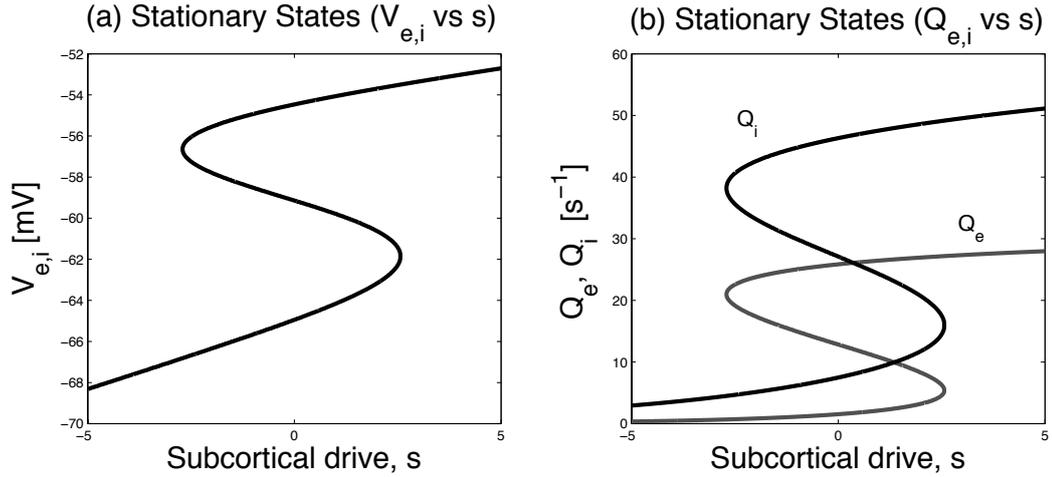


Figure 4.6: (a) Locus of steady states for soma voltages  $V_{e,i}$  as a function of subcortical driving  $s$ . (b) Equilibrium firing rates for excitatory  $Q_e$  (grey curve) and inhibitory  $Q_i$  (black curve). Note that negative subcortical drive values in the curve are unphysical. The graph was plotted by varying  $s$  and fixing  $[C, \lambda_e, \lambda_i] = [0.78, 9.0, 1.0]$ .

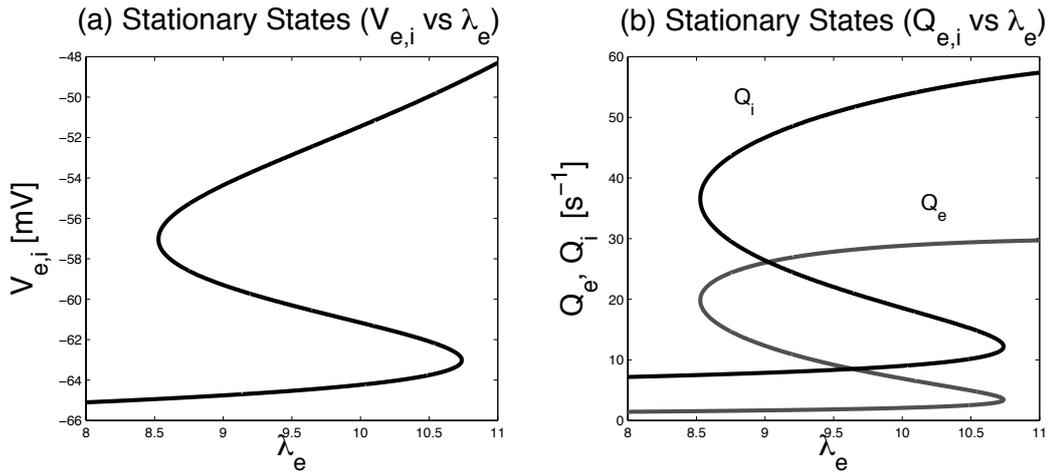


Figure 4.7: (a) Locus of steady states for soma voltages  $V_{e,i}$  as a function of excitatory neurotransmitter effect  $\lambda_e$ . (b) Equilibrium firing-rate solutions for  $Q_e$  (grey curve) and  $Q_i$  (black curve). The graph was plotted by varying  $\lambda_e$  and fixing  $[C, \lambda_i, s] = [0.78, 1.0, 0.25]$ .

#### 4.4.4 Variations in excitatory neurotransmitter $C$

The direction of the S-bend for magnesium concentration effect is the same as the one for  $\lambda_i$  (inverted S-bend); this is shown in Fig.4.8. Looking at the bottom branch when  $C \gtrsim 0.83$  mM, we obtain similar voltage and firing rates as  $\lambda_i \gtrsim 1.06$  Fig.(4.5) and  $\lambda_e \lesssim 8.53$  Fig.(4.7); giving a *hyperpolarised* state  $V_{e,i} \approx -65$  mV with firing rate  $Q_{e,i} = 1.5, 7.5$  s<sup>-1</sup> respectively. Similarly, looking at the top branch (*depolarised* state) when  $C \gtrsim 0.64$  mM, roughly the same values of  $Q_e \approx 30$  s<sup>-1</sup> and  $Q_i \approx 60$  s<sup>-1</sup> are found compared to  $\lambda_i \lesssim 0.82$  and  $\lambda_e \gtrsim 10.7$ . Again a strong firing rate exists on

the top branch when the soma voltages exceed the sigmoid threshold voltage. The maximum firing rate of the neural population happens when  $V \approx -48.3$  mV, but this time on the top-left corner for  $C \rightarrow 0$  compared to  $\lambda_e \rightarrow \infty$ .

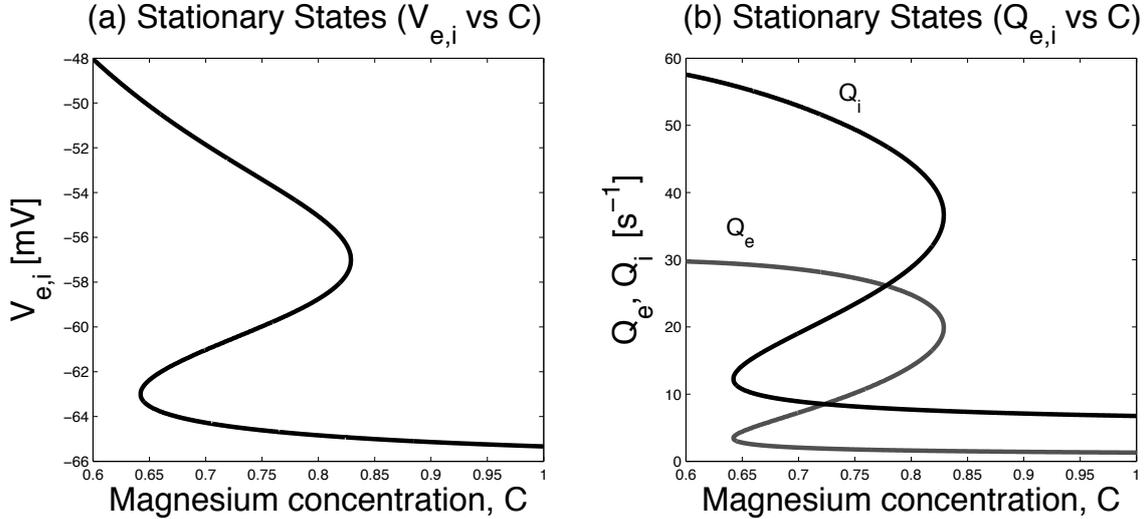


Figure 4.8: (a) Locus of steady states for soma voltages  $V_{e,i}$  as a function of Mg-concentration  $C$ . (b) Equilibrium firing rates  $Q_e$  (grey curve) and  $Q_i$  (black curve). The graph was plotted by varying  $C$  and fixing  $[\lambda_e, \lambda_i, s] = [9.0, 1.0, 0.25]$ .

## 4.5 Significance of the steady-states

Following Steyn-Ross (2002), there are three distinct regions in the manifold of steady-states of the cortex, see Fig.4.9:

Region I: This region is called “quiescent” or anaesthetized state, where the macrocolumn has only a single low-firing quiescent state.

Region II: The macrocolumn has multiple steady states: Unlike the adiabatic anaesthesia modelling reported by Steyn-Ross (2002) which displayed stable top and bottom branches separated by an unstable mid-branch, we will show (in Section 4.7) that the region of instability in the present model extends into the upper and lower branches, terminated by a pair distinct Hopf bifurcation points.

Region III: The firing rate is very strong and a single active state is available; this region is known as “highly activated”.

## 4.6 Stability analysis

Section 4.5 discussed the steady states. We now analyse the linear stability of these steady states, and find predictions of interesting new dynamical behaviours.

### 4.6.1 Linear stability analysis: Theory

Consider a general system of two nonlinear ODEs:

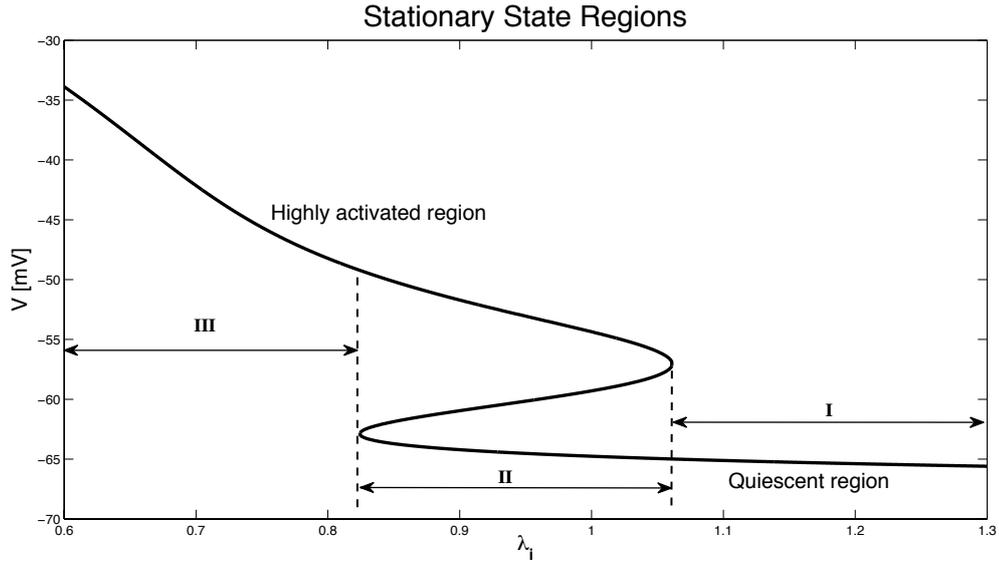


Figure 4.9: The S-bend of cortical steady states has three distinct regions: region I known as “quiescent” where the macrocolumn has only a single low-firing state; region III known as “highly activated” where the firing rate is very strong and a single active state is available; and region II where for a given value of  $\lambda$  there are three possible values for  $V_{e,i}^0$ . In this case  $\lambda_i$  is varied and the other parameters are fixed [ $C, \lambda_e, s$ ] = [0.78, 9.0, 0.25].

$$\dot{x} = f(x, y) \quad (4.19)$$

$$\dot{y} = g(x, y) \quad (4.20)$$

where a fixed point  $(x_0, y_0)$  must exist, such that,

$$f(x_0, y_0) = 0 \quad (4.21)$$

$$g(x_0, y_0) = 0 \quad (4.22)$$

To linearise this system, we expand the nonlinear functions about  $(x_0, y_0)$  as  $x = x_0 + \delta_x, y = y_0 + \delta_y$  using a Taylor series in two variables,

$$\begin{aligned} f(x_0 + \delta_x, y_0 + \delta_y) = & f(x_0, y_0) + [f_x(x_0, y_0)\delta_x] + \frac{1}{2!}(\delta_x)^2 f_{xx}(x_0, y_0) \\ & + 2\delta_x\delta_y f_{xy}(x_0, y_0) + (\delta_y)^2 f_{yy}(x_0, y_0) \dots \end{aligned} \quad (4.23)$$

$$\begin{aligned} g(x_0 + \delta_x, y_0 + \delta_y) = & g(x_0, y_0) + [g_x(x_0, y_0)\delta_x] + \frac{1}{2!}(\delta_x)^2 g_{xx}(x_0, y_0) \\ & + 2\delta_x\delta_y g_{xy}(x_0, y_0) + (\delta_y)^2 g_{yy}(x_0, y_0) \dots \end{aligned} \quad (4.24)$$

From Eqs.(4.19)–(4.20) and (4.23)–(4.24) we obtain,

$$\dot{\delta}_x = f_x(x_0, y_0)\delta_x + f_y(x_0, y_0)\delta_y + f_{xy}(x_0, y_0)\delta_x\delta_y + \dots \quad (4.25)$$

$$\dot{\delta}_y = g_x(x_0, y_0)\delta_x + g_y(x_0, y_0)\delta_y + g_{xy}(x_0, y_0)\delta_x\delta_y + \dots \quad (4.26)$$

Retaining only first-order terms, we find

$$\frac{d}{dt} \begin{bmatrix} \delta_x \\ \delta_y \end{bmatrix} = \begin{bmatrix} f_x(x_0, y_0) & f_y(x_0, y_0) \\ g_x(x_0, y_0) & g_y(x_0, y_0) \end{bmatrix} \begin{bmatrix} \delta_x \\ \delta_y \end{bmatrix} \quad (4.27)$$

These two linear DEs are considered to be the linear transformation of the general system consisting of two nonlinear ODEs. This is a linear homogeneous system with constant coefficients, so can be written in the general form (Boyce and Diprima (1997)):

$$d\mathbf{x}/dt = \mathbf{A}\mathbf{x}, \quad (4.28)$$

where  $\mathbf{A}$  is the  $2 \times 2$  *stability matrix* and  $\mathbf{x}$  is a  $2 \times 1$  vector. The eigenvalues of  $\mathbf{A}$  determine whether the solution of the system will decay exponentially towards or grow exponentially away from its stationary state. When the determinant of  $\mathbf{A}$  is non-zero ( $\det \mathbf{A} \neq 0$ ), then the matrix is nonsingular and hence  $\mathbf{x} = 0$  is the only solution of the system. To find a solution of the form  $\mathbf{x} = \xi e^{\lambda t}$ , substitute  $\mathbf{x}$  in Eq. (4.28):

$$(\mathbf{A} - \lambda\mathbf{I})\xi = 0 \quad (4.29)$$

Note that  $\lambda$  is the eigenvalue of the matrix  $\mathbf{A}$  and  $\xi$  is the corresponding eigenvector. Eq. (4.28) can be used to determine the eigenvectors, and the resulting eigenvalues are the roots of the polynomial equation.

$$\det(\mathbf{A} - \lambda\mathbf{I})\xi = 0 \quad (4.30)$$

Thus, the general solution for the linear homogeneous system in Eq.(4.27) can be written in this form:

$$\mathbf{x}(t) = c_1\xi^{(1)}e^{\lambda_1 t} + c_2\xi^{(2)}e^{\lambda_2 t} \quad (4.31)$$

where  $c_1, c_2$  are constants,  $\lambda_1, \lambda_2$  are eigenvalues and  $\xi^{(1)}, \xi^{(2)}$  are eigenvectors.

The stability of the steady states is determined by both the nature (complex or real) and sign (positive or negative) of the eigenvalues. Following Reichl (1998), various cases of real and imaginary parts of  $\lambda_1$  and  $\lambda_2$  are listed to examine the stability of the system:

1. Both  $\lambda_1$  and  $\lambda_2$  are real and negative with  $\lambda_1 < \lambda_2 < 0$ . This state is *completely stable*. A solution displaced from the steady state will decay exponentially back

to this state.

2. Both  $\lambda_1$  and  $\lambda_2$  are real and positive with  $0 < \lambda_1 < \lambda_2$ . This state is *completely unstable*. A solution displaced from the steady state will exponentially move away.
3. Both  $\lambda_1$  and  $\lambda_2$  are real with  $\lambda_1 < 0 < \lambda_2$ . This state is *unstable*. In this case, for nonzero  $c_2$  and  $t \rightarrow \infty$ , the positive eigenvalue will dominate.
4. Both  $\lambda_1$  and  $\lambda_2$  are purely imaginary. The stability of the steady state is determined by the nonlinear terms, and the system oscillates around the steady state (neither decays nor moves). This case is known as the Hopf bifurcation.
5. Both  $\lambda_1$  and  $\lambda_2$  are complex with negative real part, such that  $\lambda_1 = \alpha + i\beta$ ,  $\lambda_2 = \alpha - i\beta$  where  $\alpha < 0$ . This is a *stable* state, because the solution from the steady state is exponentially damped towards the steady state while at the same time oscillates around it.
6. Both  $\lambda_1$  and  $\lambda_2$  are complex with positive real part, such that  $\lambda_1 = \alpha + i\beta$ ,  $\lambda_2 = \alpha - i\beta$  where  $\alpha > 0$ . This is *unstable* state, where the solution displaced from the steady state will exponentially move away from the steady state while at the same time oscillating around it.

From the above linear analysis, we can determine the steady state-stability of the nonlinear system. The neglected nonlinear terms cannot change the stability of a stationary state. However, a phase transition or bifurcation to non-equilibrium steady state can occur in a nonlinear system (Reichl (1998)). Although the linear stability analysis allows us to know when this phase transition occurs, it cannot determine the form of the new state.

#### 4.6.2 Linear stability analysis for cortical model

In order to establish the stability of the steady states in our cortical model, the eight nonlinear ODEs Eq.(4.8)–(4.15) must be linearised. This can be done by applying the same approach used in linearising two nonlinear ODEs, Section 4.6.1. We assume that all variables can be expressed as a homogenous equilibrium value plus a fluctuating component. Thus, we write:

$$V_e(\vec{r}, t) = V_e^0 + \delta V_e(\vec{r}, t) \quad (4.32)$$

$$V_i(\vec{r}, t) = V_i^0 + \delta V_i(\vec{r}, t) \quad (4.33)$$

$$\Phi_e(\vec{r}, t) = \Phi_e^0 + \delta \Phi_e(\vec{r}, t) \quad (4.34)$$

$$M_e(\vec{r}, t) = M_e^0 + \delta M_e(\vec{r}, t) \quad (4.35)$$

$$\Phi_i(\vec{r}, t) = \Phi_i^0 + \delta \Phi_i(\vec{r}, t) \quad (4.36)$$

$$M_i(\vec{r}, t) = M_i^0 + \delta M_i(\vec{r}, t) \quad (4.37)$$

$$\phi_e^\alpha(\vec{r}, t) = (\phi_e^\alpha)^0 + \delta \phi_e^\alpha(\vec{r}, t) \quad (4.38)$$

$$\Omega_e(\vec{r}, t) = \Omega_e^0 + \delta \Omega_e(\vec{r}, t) \quad (4.39)$$

where  $\vec{r}$  is the 2-D position vector, and  $\delta V_e, \delta V_i, \delta \Phi_e, \delta M_e, \delta \Phi_i, \delta M_i, \delta \phi_e^\alpha, \delta \Omega_e$  are the respective fluctuations about the equilibrium values  $V_e^0, V_i^0, \Phi_e^0, M_e^0, \Phi_i^0, M_i^0, (\phi_e^\alpha)^0, \Omega_e^0$ .

Taking  $d/dt$  of Eq.(4.32)–(4.39) gives the time-rate change of the perturbations. Our linearised system can be represented in matrix form (similar to Eq. (4.27)) as following:

$$\frac{d}{dt} \begin{bmatrix} \delta V_e \\ \delta V_i \\ \delta \Phi_e \\ \delta M_e \\ \delta \Phi_i \\ \delta M_i \\ \delta \phi_e^\alpha \\ \delta \Omega_e \end{bmatrix} = \mathbf{A} \begin{bmatrix} \delta V_e \\ \delta V_i \\ \delta \Phi_e \\ \delta M_e \\ \delta \Phi_i \\ \delta M_i \\ \delta \phi_e^\alpha \\ \delta \Omega_e \end{bmatrix} \quad (4.40)$$

where  $\mathbf{A}$  is the  $8 \times 8$  **Jacobian matrix** of partial derivatives evaluated at the equilibrium point:

$$\mathbf{A} = \begin{bmatrix} J_{11} & J_{12} & J_{13} & J_{14} & J_{15} & J_{16} & J_{17} & J_{18} \\ J_{21} & J_{22} & J_{23} & J_{24} & J_{25} & J_{26} & J_{27} & J_{28} \\ J_{31} & J_{32} & J_{33} & J_{34} & J_{35} & J_{36} & J_{37} & J_{38} \\ J_{41} & J_{42} & J_{43} & J_{44} & J_{45} & J_{46} & J_{47} & J_{48} \\ J_{51} & J_{52} & J_{53} & J_{54} & J_{55} & J_{56} & J_{57} & J_{58} \\ J_{61} & J_{62} & J_{63} & J_{64} & J_{65} & J_{66} & J_{67} & J_{68} \\ J_{71} & J_{72} & J_{73} & J_{74} & J_{75} & J_{76} & J_{77} & J_{78} \\ J_{81} & J_{82} & J_{83} & J_{84} & J_{85} & J_{86} & J_{87} & J_{88} \end{bmatrix} \quad (4.41)$$

Before computing  $\mathbf{A}$ , we link the fluctuation  $(\delta V_e, \delta V_i, \delta \Phi_e, \delta M_e, \delta \Phi_i, \delta M_i, \delta \phi_e^\alpha, \delta \Omega_e)$  to its spatial Fourier transform  $(\widehat{\delta V_e}, \widehat{\delta V_i}, \widehat{\delta \Phi_e}, \widehat{\delta M_e}, \widehat{\delta \Phi_i}, \widehat{\delta M_i}, \widehat{\delta \phi_e^\alpha}, \widehat{\delta \Omega_e})$  (Steyn-Ross *et al.* (2007)), such that:

$$\delta V_e(\vec{r}, t) = \frac{1}{(2\pi)^2} \int_{-\infty}^{\infty} \widehat{\delta V}_e(\vec{q}, t) e^{i\vec{q}\cdot\vec{r}} d\vec{q}, \quad (4.42)$$

$$\delta V_i(\vec{r}, t) = \frac{1}{(2\pi)^2} \int_{-\infty}^{\infty} \widehat{\delta V}_i(\vec{q}, t) e^{i\vec{q}\cdot\vec{r}} d\vec{q}, \quad (4.43)$$

$$\delta \Phi_e(\vec{r}, t) = \frac{1}{(2\pi)^2} \int_{-\infty}^{\infty} \widehat{\Phi}_e(\vec{q}, t) e^{i\vec{q}\cdot\vec{r}} d\vec{q}, \quad (4.44)$$

$$\delta M_e(\vec{r}, t) = \frac{1}{(2\pi)^2} \int_{-\infty}^{\infty} \widehat{M}_e(\vec{q}, t) e^{i\vec{q}\cdot\vec{r}} d\vec{q}, \quad (4.45)$$

$$\delta \Phi_i(\vec{r}, t) = \frac{1}{(2\pi)^2} \int_{-\infty}^{\infty} \widehat{\Phi}_i(\vec{q}, t) e^{i\vec{q}\cdot\vec{r}} d\vec{q}, \quad (4.46)$$

$$\delta M_i(\vec{r}, t) = \frac{1}{(2\pi)^2} \int_{-\infty}^{\infty} \widehat{M}_i(\vec{q}, t) e^{i\vec{q}\cdot\vec{r}} d\vec{q}, \quad (4.47)$$

$$\delta \phi_e^\alpha(\vec{r}, t) = \frac{1}{(2\pi)^2} \int_{-\infty}^{\infty} \widehat{\delta \phi}_e^\alpha(\vec{q}, t) e^{i\vec{q}\cdot\vec{r}} d\vec{q}, \quad (4.48)$$

$$\delta \Omega_e(\vec{r}, t) = \frac{1}{(2\pi)^2} \int_{-\infty}^{\infty} \widehat{\delta \Omega}_e(\vec{q}, t) e^{i\vec{q}\cdot\vec{r}} d\vec{q}, \quad (4.49)$$

where  $\vec{q}$  is the 2-D wavenumber vector. Using Fourier transformation leads to replacement of  $\nabla^2$  by  $-q^2$  (where  $q = |\vec{q}|$ ) in our system. The Fourier-transformed equations are listed for completeness only; we will not use space in our stability analysis or simulations (i.e. we will set  $q = 0$ ).

### 4.6.3 Elements of the $8 \times 8$ Jacobian matrix

In order to obtain the stability characteristics of the new linear system, we need to compute the eigenvalues of the  $8 \times 8$  Jacobian matrix.

Computing the matrix elements from Eqs.(4.8–4.15) (after Fourier transforming

them) results in:

$$J_{11} = -1 + \widehat{\psi}_{ee} \widehat{\Phi}_e \frac{\partial \widehat{\rho}'_e}{\partial \widehat{V}_e} + \widehat{\rho}'_e \widehat{\Phi}_e \frac{\partial \widehat{\psi}_{ee}}{\partial \widehat{V}_e} + \widehat{\rho}'_i \widehat{\Phi}_i \frac{\partial \widehat{\psi}_{ie}}{\partial \widehat{V}_e} \frac{1}{\tau_e} \quad (4.50)$$

$$J_{13} = \widehat{\rho}'_e \widehat{\psi}_{ee} \frac{1}{\tau_e} \quad (4.51)$$

$$J_{15} = \widehat{\rho}'_i \widehat{\psi}_{ie} \frac{1}{\tau_e} \quad (4.52)$$

$$J_{21} = \widehat{\psi}_{ei} \widehat{\Phi}_e \frac{\partial \widehat{\rho}'_e}{\partial \widehat{V}_e} \frac{1}{\tau_i} \quad (4.53)$$

$$J_{22} = -1 + \widehat{\rho}'_e \widehat{\Phi}_e \frac{\partial \widehat{\psi}_{ei}}{\partial \widehat{V}_i} + \widehat{\rho}'_i \widehat{\Phi}_i \frac{\partial \widehat{\psi}_{ii}}{\partial \widehat{V}_i} \frac{1}{\tau_i} \quad (4.54)$$

$$J_{23} = \widehat{\rho}'_e \widehat{\psi}_{ei} \frac{1}{\tau_i} \quad (4.55)$$

$$J_{25} = \widehat{\rho}'_i \widehat{\psi}_{ei} \frac{1}{\tau_i} \quad (4.56)$$

$$J_{34} = 1 \quad (4.57)$$

$$J_{41} = \gamma_e^2 N_e^\beta \frac{\partial \widehat{Q}_e}{\partial \widehat{V}_e} \quad (4.58)$$

$$J_{43} = -\gamma_e^2 \quad (4.59)$$

$$J_{44} = -2\gamma_e \quad (4.60)$$

$$J_{47} = \gamma_e^2 N_e^\alpha \quad (4.61)$$

$$J_{56} = 1 \quad (4.62)$$

$$J_{62} = \gamma_i^2 N_i^\beta \frac{\partial \widehat{Q}_i}{\partial \widehat{V}_i} \quad (4.63)$$

$$J_{65} = -\gamma_i^2 \quad (4.64)$$

$$J_{66} = -2\gamma_i \quad (4.65)$$

$$J_{78} = 1 \quad (4.66)$$

$$J_{81} = v^2 \Lambda_e^2 \frac{\partial \widehat{Q}_e}{\partial \widehat{V}_e} \quad (4.67)$$

$$J_{87} = v^2 (-q^2 - \Lambda_e^2) \quad (4.68)$$

$$J_{88} = -2v\Lambda_e \quad (4.69)$$

Note that all other elements of the matrix are equal to zero.

Thus,

$$\mathbf{A} = \begin{bmatrix} J_{11} & 0 & J_{13} & 0 & J_{15} & 0 & 0 & 0 \\ J_{21} & J_{22} & J_{23} & 0 & J_{25} & 0 & 0 & 0 \\ 0 & 0 & 0 & J_{34} & 0 & 0 & 0 & 0 \\ J_{41} & 0 & J_{43} & J_{44} & 0 & 0 & J_{47} & 0 \\ 0 & 0 & 0 & 0 & 0 & J_{56} & 0 & 0 \\ 0 & J_{62} & 0 & 0 & J_{65} & J_{66} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & J_{78} \\ J_{81} & 0 & 0 & 0 & 0 & 0 & J_{87} & J_{88} \end{bmatrix} \quad (4.70)$$

where:

$$\frac{\partial \hat{\rho}'_e}{\partial \hat{V}_e} = \frac{\lambda_e \rho_e a k C e^{-a \hat{V}_e}}{(1 + k C e^{-a \hat{V}_e})^2} \quad (4.71)$$

The derivatives of the four weighting-functions  $\psi_{jk}$  are constants given by:

$$\frac{\partial \hat{\psi}_{ee}}{\partial \hat{V}_e} = \frac{-1}{(V_e^{\text{rev}} - V_e^{\text{rest}})} \quad (4.72)$$

$$\frac{\partial \hat{\psi}_{ei}}{\partial \hat{V}_e} = \frac{-1}{(V_e^{\text{rev}} - V_e^{\text{rest}})} \quad (4.73)$$

$$\frac{\partial \hat{\psi}_{ie}}{\partial \hat{V}_i} = \frac{-1}{(V_i^{\text{rev}} - V_i^{\text{rest}})} \quad (4.74)$$

$$\frac{\partial \hat{\psi}_{ii}}{\partial \hat{V}_i} = \frac{-1}{(V_i^{\text{rev}} - V_i^{\text{rest}})} \quad (4.75)$$

The firing-rate sigmoids  $Q_{e,i}$  have the following derivatives:

$$\frac{\partial \hat{Q}_e}{\partial \hat{V}_e} = \frac{Q_e^{\text{max}}(C/\sigma_e) \exp[-(\hat{V}_e - \theta_e)C/\sigma_e]}{(1 + \exp[-(\hat{V}_e - \theta_e)C/\sigma_e])^2} \quad (4.76)$$

$$\frac{\partial \hat{Q}_i}{\partial \hat{V}_i} = \frac{Q_i^{\text{max}}(C/\sigma_i) \exp[-(\hat{V}_i - \theta_i)C/\sigma_i]}{(1 + \exp[-(\hat{V}_i - \theta_i)C/\sigma_i])^2} \quad (4.77)$$

## 4.7 Stability predictions

The dominant eigenvalue graphs [Figs 4.10, 4.11, 4.12 and 4.13] all have the following regions in common; a stable part (solid black) and an unstable part (dotted lines). There are two stable regions one belonging to the upper (highly activated) branch and the other to the lower (quiescent) branch. These are joined by an unstable region.

The graph of the real part of the dominant eigenvalue against one of the four input parameters  $[\lambda_i, \lambda_e, C, s]$  with the other three parameters kept constant at near default values, generally looks like a loop. The loop tends to be partly greater

than zero. This is the unstable region. The ‘legs’ of the loop below zero are the stable regions corresponding to parts of the upper and lower branches of the S-bend.

The transition between the stable and unstable regions is marked by a Hopf bifurcation point (circled) where the dominant eigenvalue has zero real part but non-zero imaginary part. Regions on either side of the Hopf point are oscillatory.

The imaginary part of the dominant eigenvalue(s) represents the oscillation component. Hence the purely real dominant eigenvalue of the central part of the unstable region suggests that it may be non-oscillatory. However the other eigenvalues may have non-zero imaginary parts and possibly generate oscillatory behaviour.

The noticeable feature of this model is that the dominant eigenvalue crosses zero while it is still on the top or bottom branches before reaching the turning points (TP1 and TP2) on the S-bend. This means that the regions between the Hopf points and the turning points becomes unstable and a jump between states can occur. The closer the state to the turning point the sooner the jump will occur.

The stability changes caused by variations in each of the four input parameters are now discussed in more detail.

#### 4.7.1 Stability changes with anesthetic effect $\lambda_i$

To demonstrate the effect of anaesthetic on model stability, the values of  $\lambda_i$  were varied while the other parameters remain fixed;  $[C, \lambda_e, s]$  are  $[0.78, 9.0, 0.25]$ .

In Fig.4.10 for low  $\lambda_i$  (minimal inhibition), the system is in a highly activated state. As  $\lambda_i$  is increased the steady state voltage decreases. In the mid-range of  $\lambda_i$  there is a discontinuity between the high and low steady states voltages range where a jump transition may occur; similarly for decreasing  $\lambda_i$  from a quiescent state. Where and how quickly this switching can take place is determined by the system stability.

The system is stable at the extremes of the chosen  $\lambda_i$  range. This is shown by the negative dominant eigenvalues in Fig.4.10b.

Approaching the multiple steady state region from either side, the system loses stability until it is barely stable before the Hopf points. After these critical points the system becomes increasingly unstable up to the turning points (TP1 and TP2). The less stable it is, the longer the system takes to return to the steady states from a given perturbations. The more unstable the system is the sooner the perturbation becomes a large scale event.

Fig.4.10b shows that the upper branch generally has higher oscillation frequency than the lower branch (the magnitude of the imaginary part of the dominant eigenvalue is directly proportional to oscillation frequency). The Hopf point (H1) belonging to the upper branch has approximate frequency of 2.4 Hz, while the lower branch Hopf (H1) has a frequency of approximately 1.3 Hz.

### 4.7.2 Stability changes with subcortical driving, $s$

The parameters  $\lambda_i$ ,  $\lambda_e$  and  $C$  were kept at default values and  $s$  was varied. The negative values of  $s$  are unphysical, we use them as we mentioned before just to show the whole shape of the S-bend. From 0 to H2 only the bottom branch is stable. There are no stable steady states between  $s = 1.196$  and  $s = 4.244$ . These are the Hopf points (H1 and H2). Comparing the S-bend in this case with the other S-bends, we note that there is no overlap between the stable states.

### 4.7.3 Stability changes with NMDA effect, $\lambda_e$ and $C$

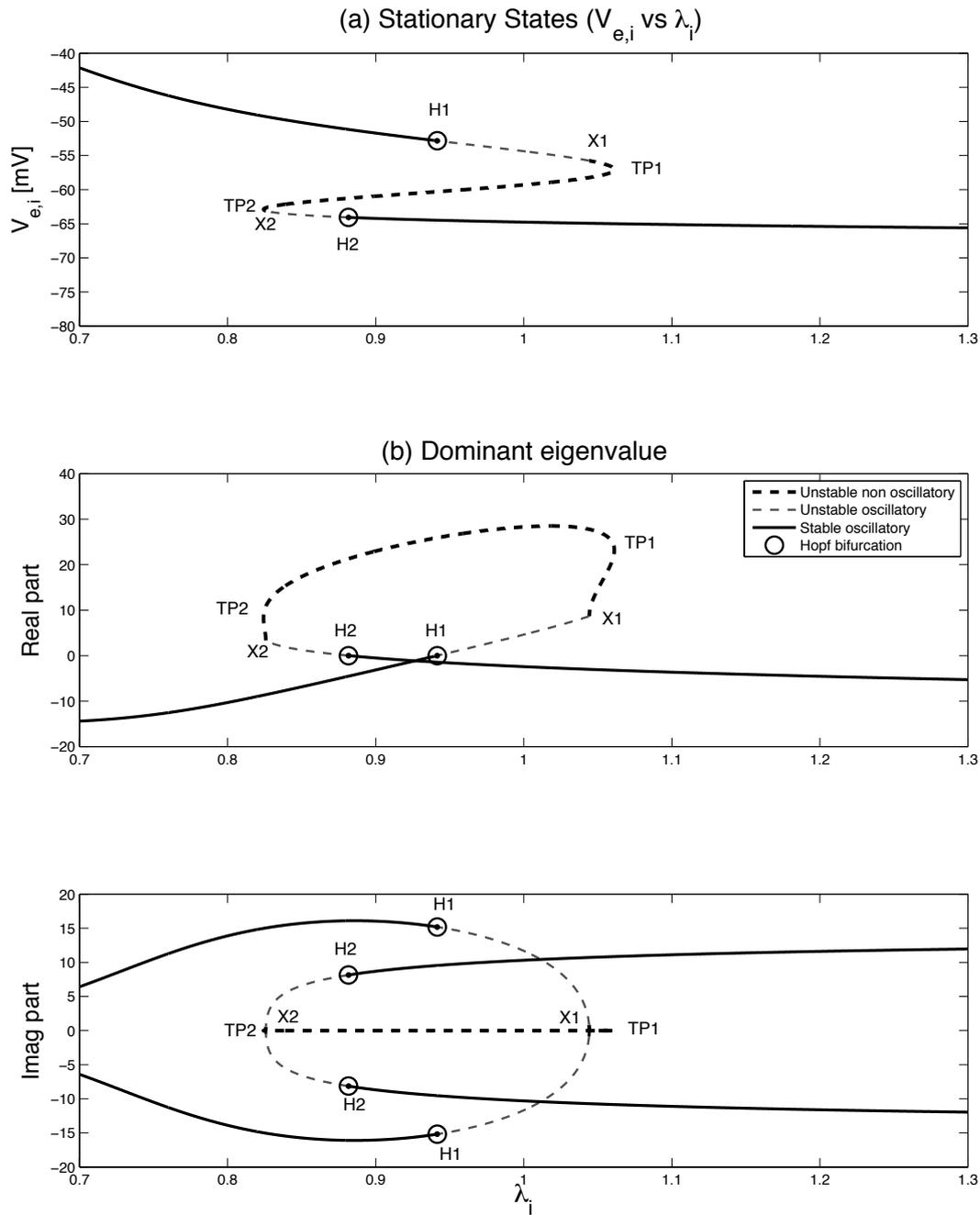
$\lambda_e$  and  $C$  are the excitatory and inhibitory influences respectively and so  $\lambda_e$  produces an S-bend and  $C$  produces a reverse S-bend very similar to that obtained for  $\lambda_i$ . Apart from the range and direction the graphs are almost the same. The Hopf frequencies on the top branch are 1.36 Hz for both NMDA effects. The bottom branch is similar at 2.45 Hz and 2.44 Hz respectively. The maximum frequency is 4 Hz.

## 4.8 Predicted effect of NMDA on cortical stability

For plots of  $\lambda_i$  (such as Fig.4.14), increasing  $\lambda_e$  has the effect of shifting the graph to the right. The  $\lambda_e$  excites the system therefore a larger fixed  $\lambda_e$  will require greater  $\lambda_i$  before the system quiets. On the other hand, the magnesium concentration  $C$  has the opposite effect of aiding the inhibition of  $\lambda_i$  and therefore increasing  $C$  will have the effect of shifting the  $\lambda_i$  graph to the left (i.e., quiet states are more easily achieved with lower value of  $\lambda_i$ ).

The effects of  $C$  and  $\lambda_e$  are opposite to each other (see Fig.4.15). Increased  $\lambda_e$  will require less subcortical drive  $s$  to reach a high firing state and increased  $C$  will cancel out some of the effects of  $s$  and therefore needing more  $s$  to reach a high state.

We discussed above the effect of increasing  $C$  and  $\lambda_e$ , and now we will show what would happen if we decrease these  $C$  and  $\lambda_e$ . Fig.4.14 (for  $\lambda_i$ ) and Fig.4.15 (for  $s$ ) indicate how the steady-state stability changes with decreasing  $C$  and  $\lambda_e$ . It can be seen that decreasing  $C$  shifts the  $\lambda_i$  S-bend to the right and  $s$  S-bend to the left. In contrast, decreasing  $\lambda_e$  shifts the S-bends to the opposite direction of  $C$  effect.



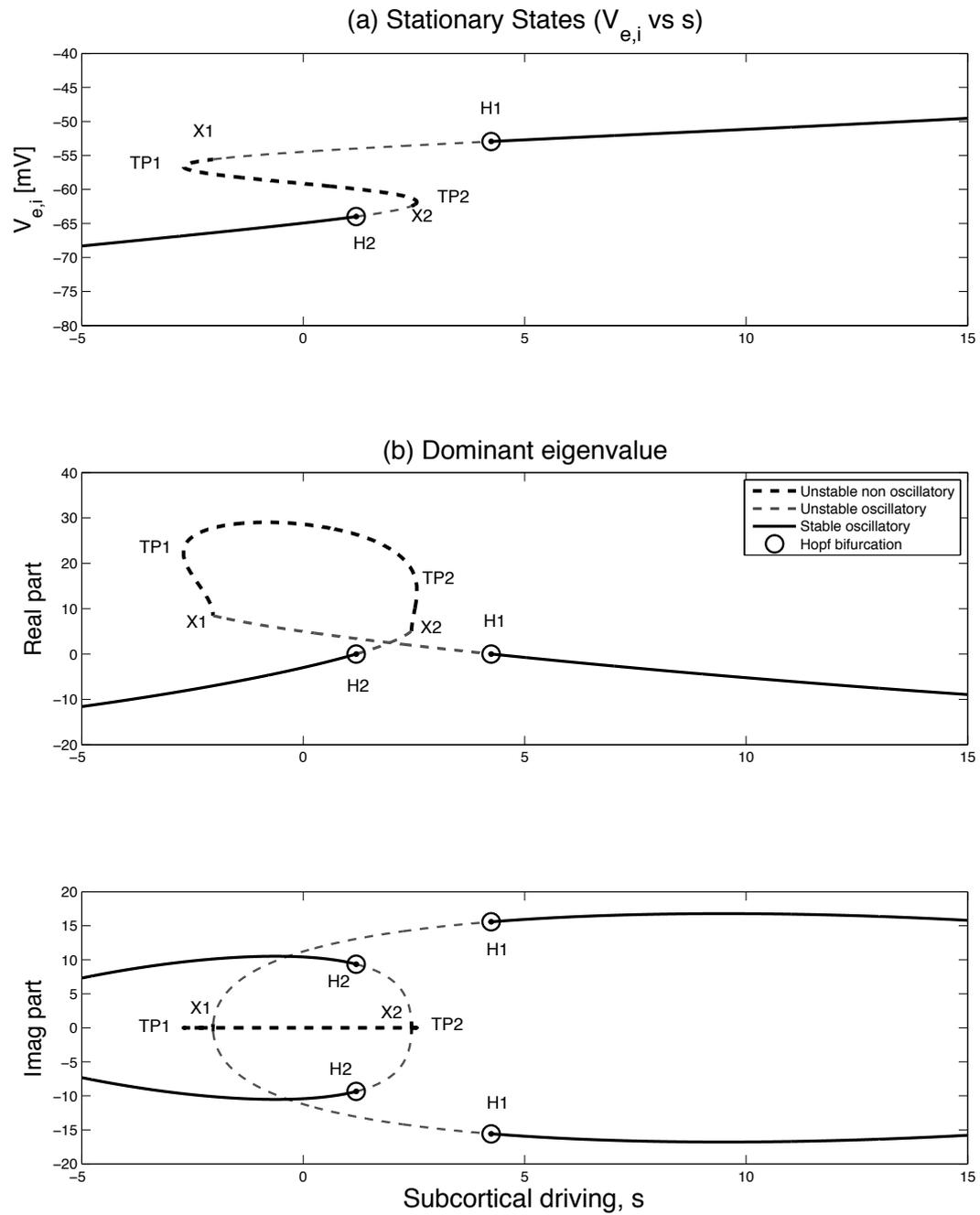


Figure 4.11: (a) Steady states for variable subcortical driving  $s$ . (b) Variation of stability for the subcortical driving  $s$ . Here,  $[C, \lambda_e, \lambda_i] = [0.78, 9.0, 1]$ .

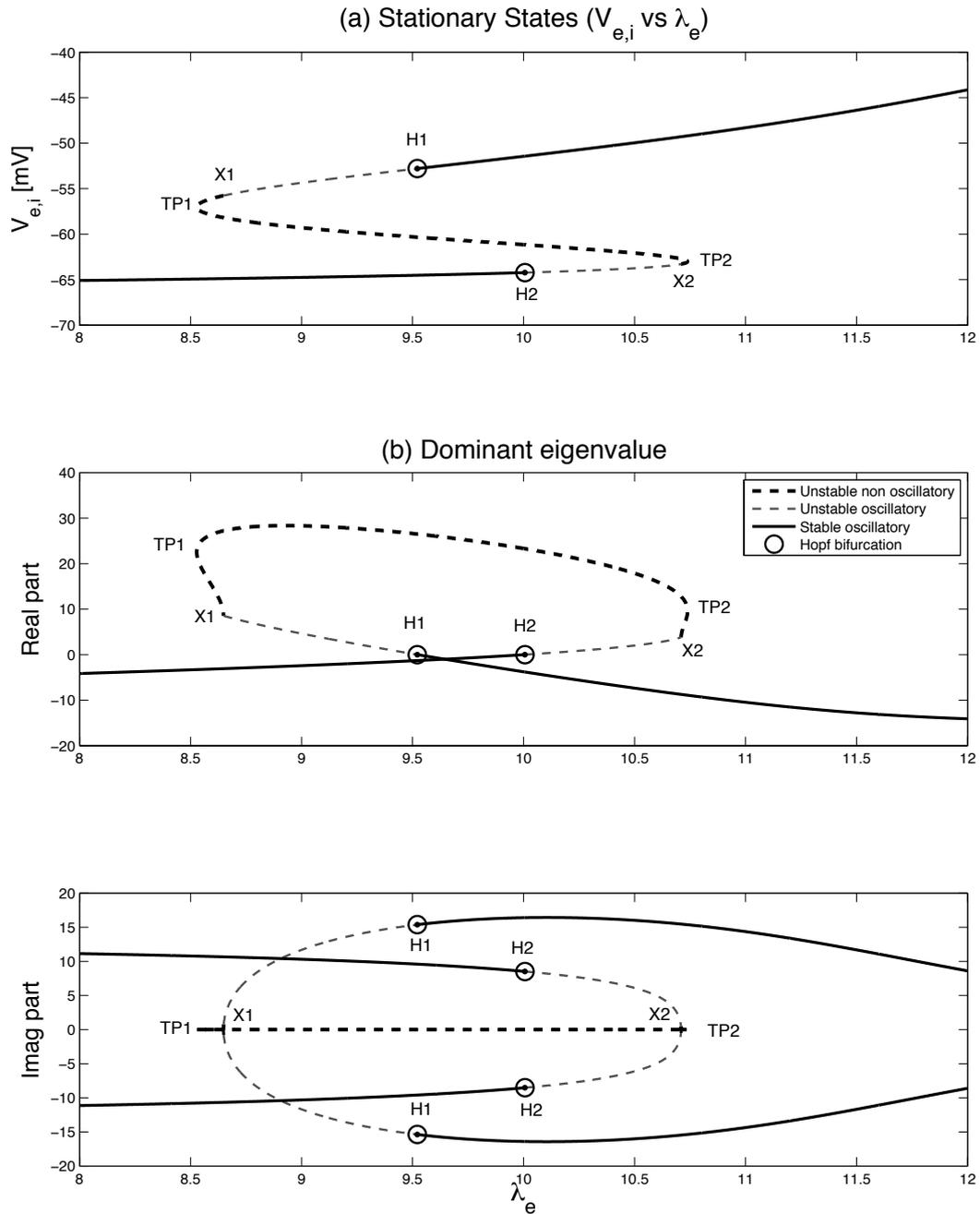


Figure 4.12: (a) Steady states for excitatory effect  $\lambda_e$ . (b) Variation of stability for variable  $\lambda_e$ . Here,  $[C, \lambda_i, s] = [0.78, 1, 0.25]$ .

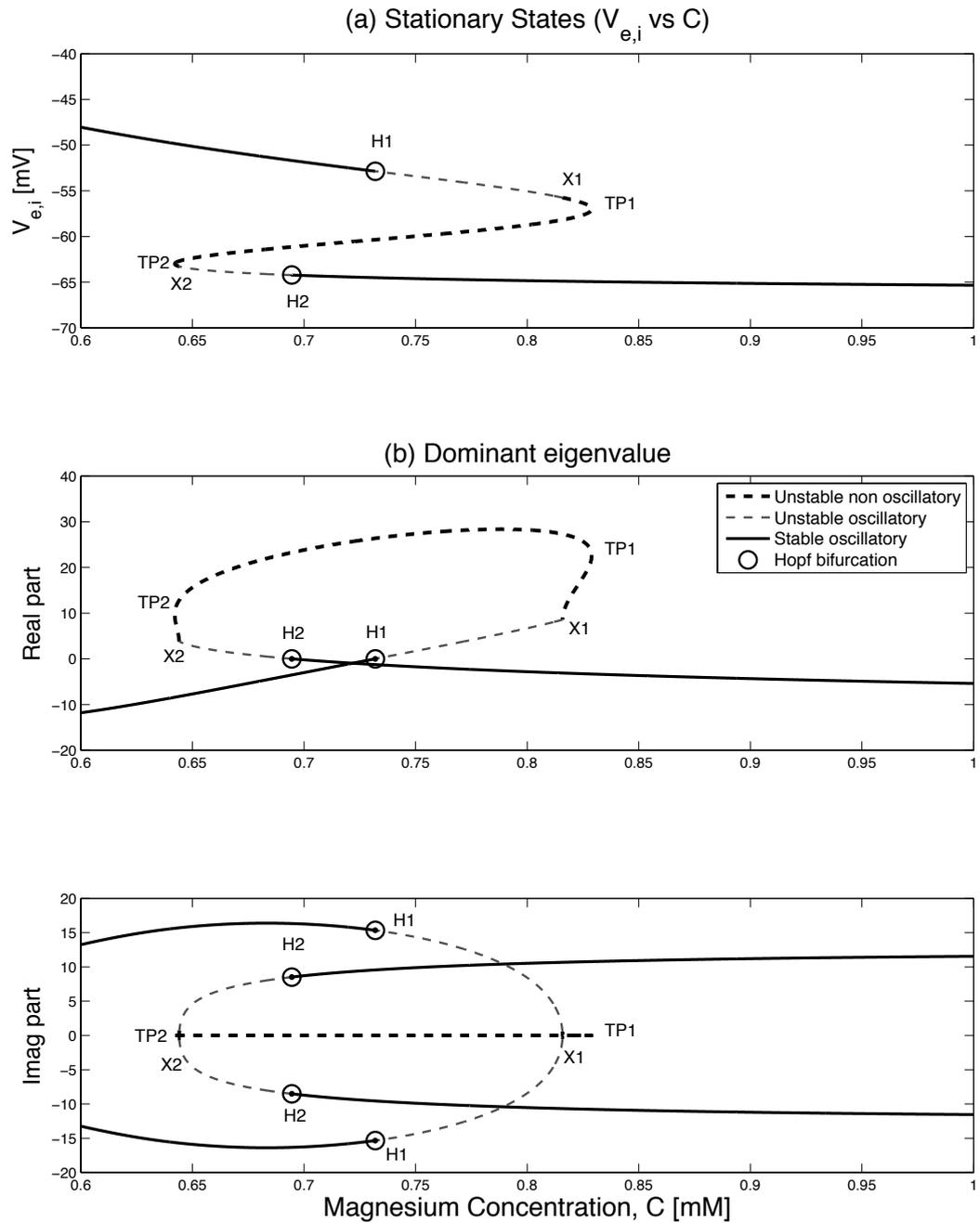


Figure 4.13: (a) Steady states for variation in NMDA effect  $C$ . (b) Variation of stability for the NMDA effect  $C$ .  $[\lambda_e, \lambda_i, s] = [9.0, 1, 0.25]$ .

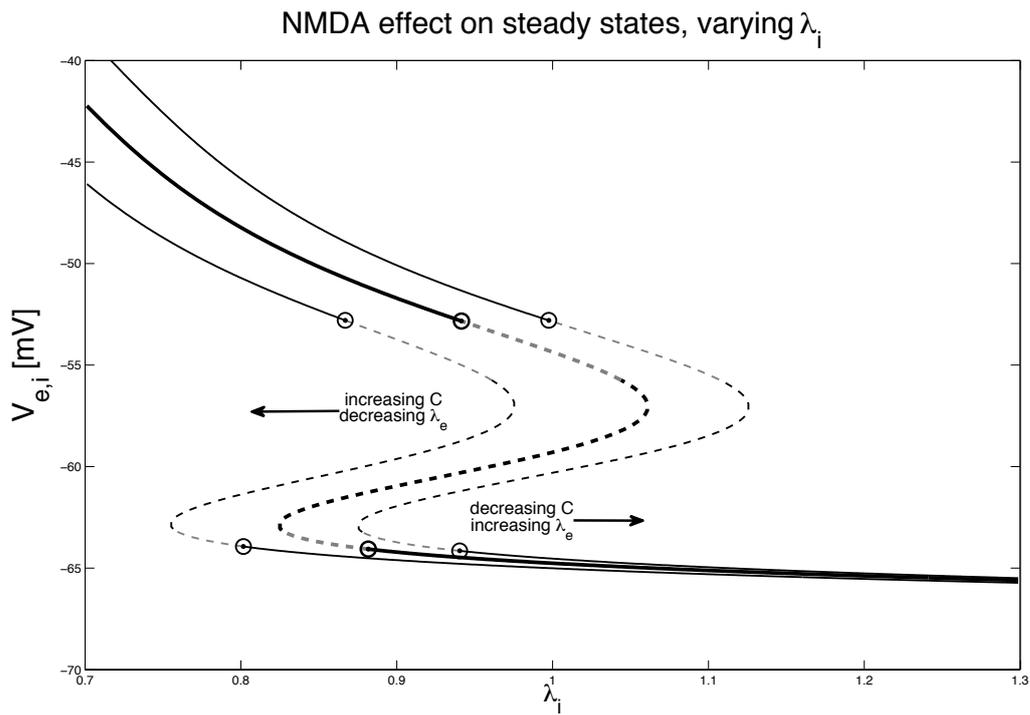


Figure 4.14: The NMDA effect on the macrocolumn stability (varying  $\lambda_i$ ): default values curve (bold),  $C$  increased to 0.85 mM (left),  $\lambda_e$  increased to 9.5 (right).

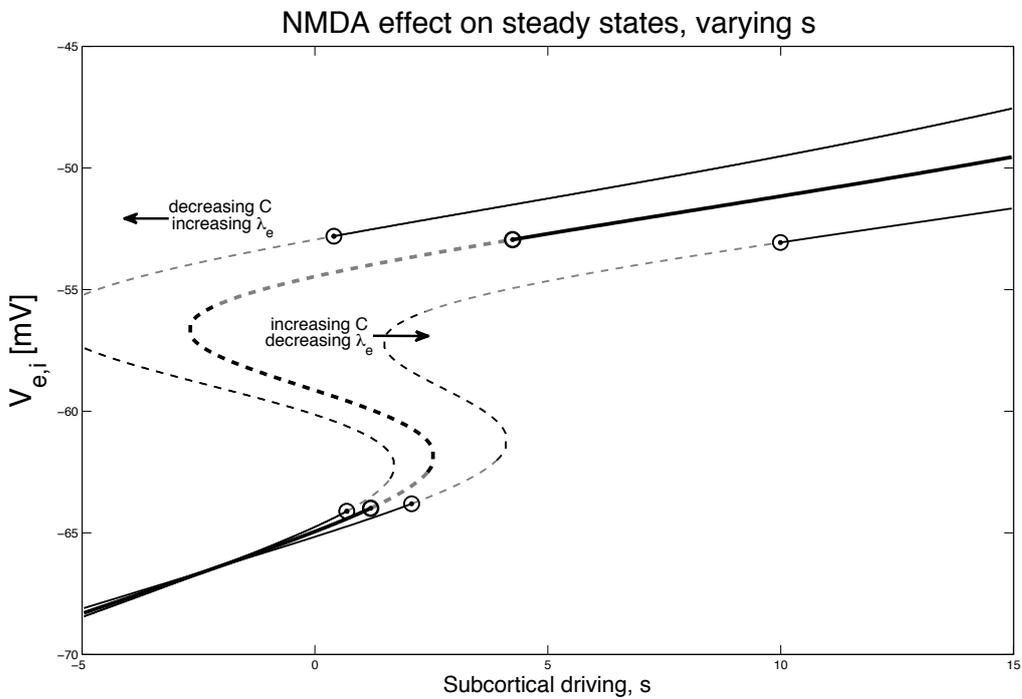


Figure 4.15: The NMDA effect on cortical stability (varying  $s$ ): the bold curve is a plot for the default values, on the left  $\lambda_e$  increased to 9.5 and on the right  $C$  increased to 0.85 mM.

## Chapter 5

# Numerical simulation

The aim of this chapter is to verify the stability results found in Chapter 4 using numerical simulation of the macrocolumn stochastic differential equations. This will be calculated by adding white noise to the system and then simulating it using a stochastic one-step Euler method.

### 5.1 Discrete approximation for white noise

The continuous-time stochastic differential equation Eq.(3.45) contains two independent zero-mean delta-correlated noise inputs  $\xi(t)$ . The aim is to transform the random term  $\xi(t)$  to a discrete-time numerical simulation  $\eta_n$  which can be achieved by following the method presented by Murthy (1983) in mapping for the random noise component. The zero-mean delta-correlated  $\xi(t)$  is modelled as a white noise as following:

$$\langle \xi(t) \rangle = 0, \quad \langle \xi(t)\xi(t') \rangle = A^2\delta(t - t') \quad (5.1)$$

where  $A$  = root-mean-square (rms) value of  $\xi(t)$ .

The white noise has a pathological feature of infinite variance and therefore cannot be precisely represented by any real signal. The continuous-time random time-series  $\xi(t)$  is simulated by an equivalent discrete random sequence  $\eta_n$  at time values  $t_n = n\Delta t$ , having 0 mean and variance =  $\sigma^2$ .

$$\langle \eta_n \rangle = 0, \quad \langle \eta_n \eta_m \rangle = \sigma^2 \delta_{nm}; \quad (5.2)$$

where  $\delta_{nm}$  is the unit impulse response.

Following Steyn-Ross (2002), the product  $\xi(t)\xi(t')$ , the expected value of the integral over all time is

$$\langle \lim_{T \rightarrow \infty} \int_{-T}^T \xi(t)\xi(t')dt \rangle = \int_{-\infty}^{\infty} \langle \xi(t)\xi(t')dt \rangle dt = 1; \quad (5.3)$$

and the expected value for the summation of the discrete products  $\eta_n \eta_m$ , is,

$$\langle \lim_{N \rightarrow \infty} \sum_{n=1}^N \eta_n \eta_m \Delta t \rangle = \lim_{N \rightarrow \infty} \sum_{n=1}^N \langle \eta_n \eta_m \rangle \Delta t = \sigma^2 \Delta t. \quad (5.4)$$

These two expected values must be equivalent, hence standard deviation for the discrete sequence is

$$\sigma = \frac{1}{\sqrt{\Delta t}} \quad (5.5)$$

The built in Matlab `randn` function is used to generate the Gaussian-distributed random numbers  $R_n$  of zero mean used in the stochastic simulations later.

$$\langle R_n \rangle = 0, \quad \langle R_n R_m \rangle = \delta_{n,m} \quad (5.6)$$

The stochastic sequence  $\eta_n$  required for the numerical simulation are calculated using the generated numbers  $R_n$

$$\eta_n = \frac{R_n}{\sqrt{\Delta t}} \quad (5.7)$$

The result of the previous step makes the mapping from continuous-time white noise  $\xi(t)$  to its discrete-time approximation  $\eta_n$

$$\xi(t) \rightarrow \eta_n = \frac{R_n}{\sqrt{\Delta t}}, \quad t = n\Delta t \quad (5.8)$$

The discrete sequence gives a better approximation of white noise as  $\eta_n \rightarrow \infty$  in the limit  $\xi(t) \rightarrow 0$ . Overall, the accuracy of the stochastic simulation is enhanced as the time-step is decreased.

## 5.2 Difference equations

As mentioned earlier, the aim of this chapter is to verify the predicted stability properties of the soma voltage  $V_{e,i}$  using a stochastic one-step Euler method. This is done by a simple iteration scheme at a time step  $n+1$  (at a time  $t = (n+1)\Delta t$ ) via linear extrapolation from the previous time step  $n$ :

$$x^{n+1} = x^n + \left[ \frac{dx}{dt} \right] \Delta t \quad (5.9)$$

where  $x^n$  is the value of  $x$  at  $t = n\Delta t$ . The random component of the derivative term is a function of the time step  $\Delta t$  used in the simulation as these system of differential equations are stochastic.

Now the simple iteration scheme of the Euler method will be applied in the full-equations (4.31)–(4.38).

$$V_e^{n+1} = V_e^n + \frac{1}{\tau_e} [V_e^{\text{rest}} - V_e^n + \rho_e^n \psi_{ee}^n \Phi_e^n + \rho_i \psi_{ie}^n \Phi_i^n] \Delta t \quad (5.10)$$

$$V_i^{n+1} = V_i^n + \frac{1}{\tau_i} [V_i^{\text{rest}} - V_i^n + \rho_e^n \psi_{ei}^n \Phi_e^n + \rho_i \psi_{ii}^n \Phi_i^n] \Delta t \quad (5.11)$$

$$\Phi_e^{n+1} = \Phi_e^n + [M_e]^n \Delta t \quad (5.12)$$

$$M_e^{n+1} = M_e^n + [-2\gamma_e M_e^n + [-\Phi_e^n + N_e^\alpha (\phi_e^\alpha)^n + N_e^\beta Q_e^n + s(\phi_e^{\text{sc}})^n] \gamma_e^2] \Delta t \quad (5.13)$$

$$\Phi_i^{n+1} = \Phi_i^n + [M_i]^n \Delta t \quad (5.14)$$

$$M_i^{n+1} = M_i^n + [-2\gamma_i M_i^n + [-\Phi_i^n + N_i^\alpha (\phi_i^\alpha)^n + N_i^\beta Q_i^n] \gamma_i^2] \Delta t \quad (5.15)$$

$$\phi_e^{n+1} = \phi_e^n + [\Omega_e]^n \Delta t \quad (5.16)$$

$$\Omega_e^{n+1} = \Omega_e^n + [-2\nu \Lambda_e \Omega_e^n + [v^2(-q)^2 - v^2 \Lambda_e^2] (\phi_e^\alpha)^n + v^2 \Lambda_e^2 Q_e^n] \Delta t \quad (5.17)$$

where

$$(\phi_e^{\text{sc}})^n = \langle \phi_e^{\text{sc}} \rangle + \sqrt{\phi_e^{\text{sc}}} \frac{R_n}{\sqrt{\Delta t}} \quad (5.18)$$

### 5.3 Verification of macrocolumn steady states

The focus of this section is to verify the theoretical distribution of macrocolumn steady states found in Chapter 4 by running the numerical simulation defined in Eq.(5.10)–(5.17). Due to the fact that the equations were strongly non-linear, the choice of time step had to be small in order for the Euler method to work (remains stable). It was found that the simulation's deterministic behaviour was unaffected when running the simulation with a range of time steps up to  $10^{-3}$ , therefore the choice of  $\Delta t = 10^{-3}$  s was decided to be safe enough for the simulation.

### 5.4 Validation of the stable branches

In order to demonstrate the stability of the steady states on the stable region of the cortical model, we choose some values on the upper and lower branches of the S-bend (Fig. 4.5). Several experiments were carried out to observe the behaviour of the steady states on the stable branches, and the finding was: the fluctuations can be seen about the stationary states, and the system never moves far from these stationary states. This is true for all steady states along the stable region (upper and lower branches). To confirm the above results, two values of stationary states

on the upper branch were selected, ( $\lambda_i = 0.8$ ) and ( $\lambda_i = 0.9$ ), where the first one is far from the Hopf point while the second one is much closer to it.

It was found that the displacement of the steady state from the Hopf point affects its stability. This is indicated in Fig.5.1 where the noise was turned off after 4 seconds. For  $\lambda_i = 0.8$  (far from the Hopf point) the system decays quickly to the steady state. But for  $\lambda_i = 0.9$  (close to the Hopf) it remains oscillating and decaying slowly to the stationary state. Similar results were found for the points of stable lower branch.

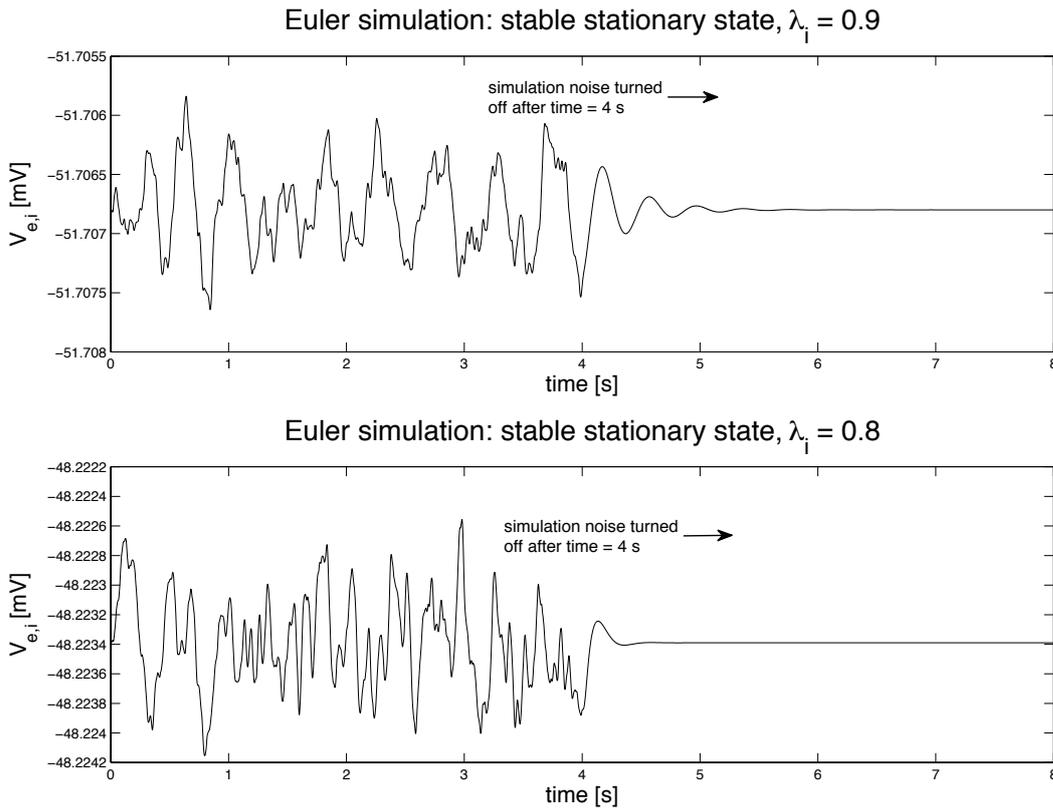


Figure 5.1: Simulation on the stable region, (varying  $\lambda_i$ , default values for the other inputs): after turning the noise off (4 s), the voltage remains oscillating and decaying slowly when  $\lambda_i = 0.9$  (close to the Hopf), while it decays quickly when  $\lambda_i = 0.8$  (far from the Hopf).  $[C, \lambda_e, s] = [0.78, 9.0, 0.25]$ .

## 5.5 Examination of the Hopf bifurcation

Starting the simulation at the Hopf points enable us to understand the macrocolumn stability there. Figure 5.2 shows the simulation of the two Hopf points: when  $\lambda_i = 0.9415$  on the upper branch and  $\lambda_i = 0.8817$  on the lower branch, turning off the noise after 10 seconds. The system oscillates with a frequency of approximately 2.40 Hz for the upper Hopf and 1.33 Hz for the lower Hopf. This is in agreement with predicted Hopf oscillation frequency of 2.417 Hz (H1) and 1.297 Hz (H2) respectively

discussed in Chapter 4. The start of the instability of the system corresponds to the Hopf bifurcation points. For longer simulations at the Hopf values the oscillations were seen slowly diverging from the steady state. This behaviour was exhibited by the system after the noise was turned off after some time in the simulation. Theoretically the oscillation magnitude should be constant at the Hopf point. The slow growth is likely to be due to inaccuracies in the numerical simulation arising from accumulation of round-off and Euler truncation errors.

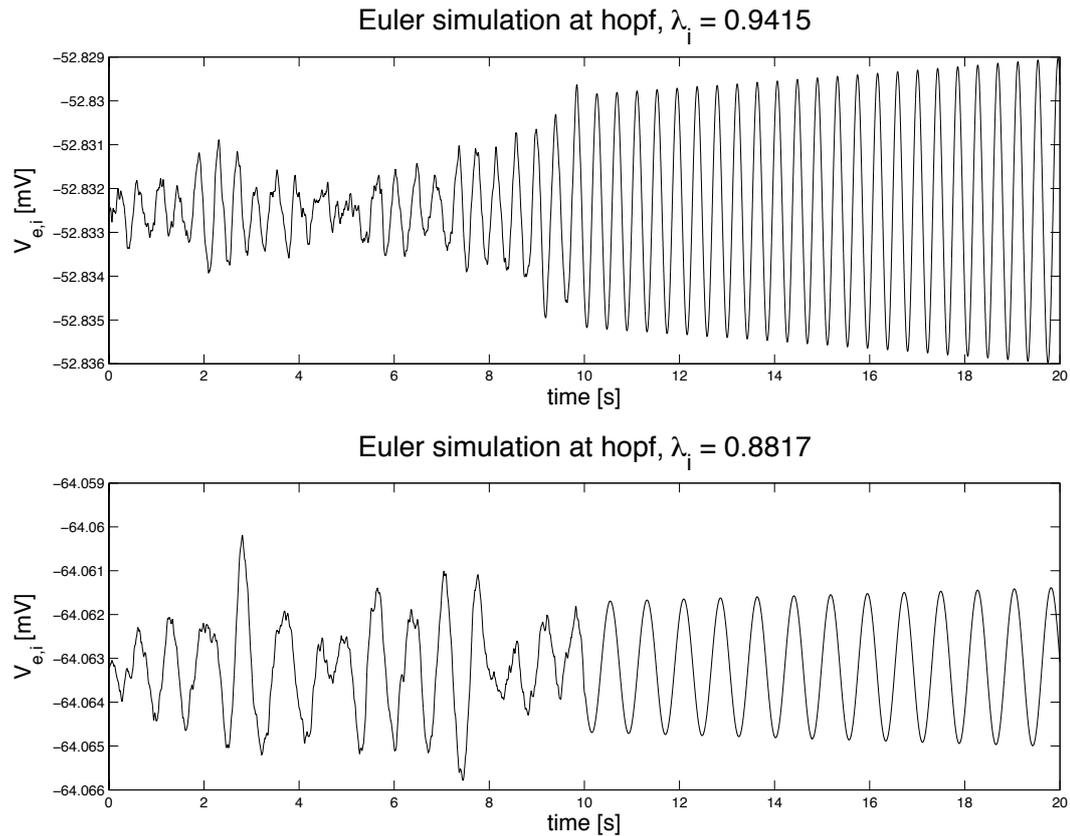


Figure 5.2: Simulating the Hopf points. When the noise was turned off (10 s) the oscillations were growing slowly for both Hopf points at  $\lambda_i = 0.9415$  (upper branch) and  $\lambda_i = 0.8817$  (lower branch).  $[C, \lambda_e, s] = [0.78, 9.0, 0.25]$ .

## 5.6 Simulation on the unstable region

There are two main behaviours seen in the simulation of the unstable region. These are either a transition between states, or emergence of limit cycle oscillations. The noise-stimulated system never remains at these unstable steady states, in agreement with stability prediction in Chapter 4. For the case of limit cycles, we see large amplitude oscillations that are insensitive to noise.

In the simulation the system diverges from an unstable steady state and attempts to make a transition to a stable steady state if one is available. If that destination

stable steady state is too close to the Hopf point it cannot attract the system strongly enough to capture it. Instead the system enters a limit cycle doing flybys past the steady states.

In contrast, if the stable destination is far enough from the Hopf the system make a transition and settles into the new equilibrium state with decreasing oscillations as shown in Fig 5.3. The system diverges where there is only unstable steady states until it goes into a limit cycle which agrees with a physical limitation preventing unbounded growth. This only happens for the case of subcortical drive  $s$  (see Fig 5.4).

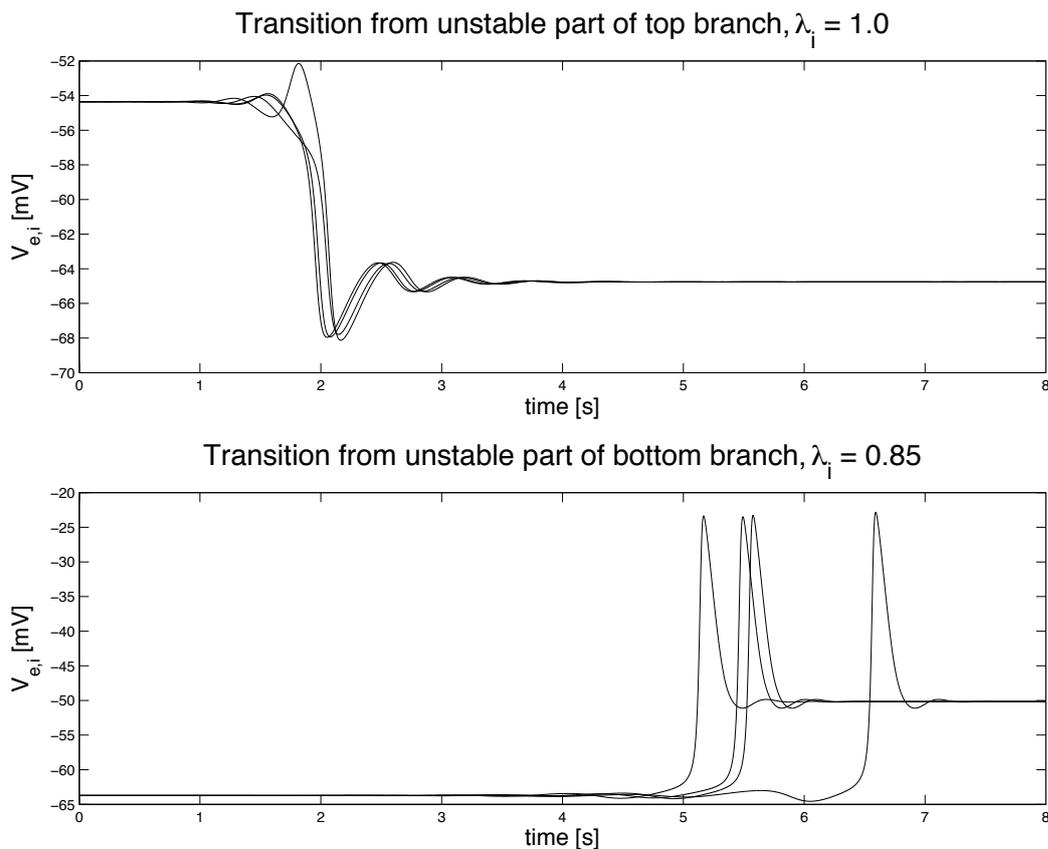


Figure 5.3: Transition between states. When  $\lambda_i=1.0$  a transition from unstable part of top branch to stable lower branch; when  $\lambda_i=0.85$  a transition from unstable part of bottom branch to stable upper branch.  $[C, \lambda_e, s] = [0.78, 9.0, 0.25]$ .

In the case where there are steady states above and below the point where the simulation is begun, the initial displacement from the unstable steady state may be important in determining the final evaluation of the system. Usually such starting points lie on the mid branch which are non oscillatory. When the system is displaced far enough from the stationary states by the noise addition, the exponential divergence suppresses the ability of the noise fluctuations to return the system to the neutral state. At this point the two possible paths are separated.

For certain starting points one path led to a transition while the other path

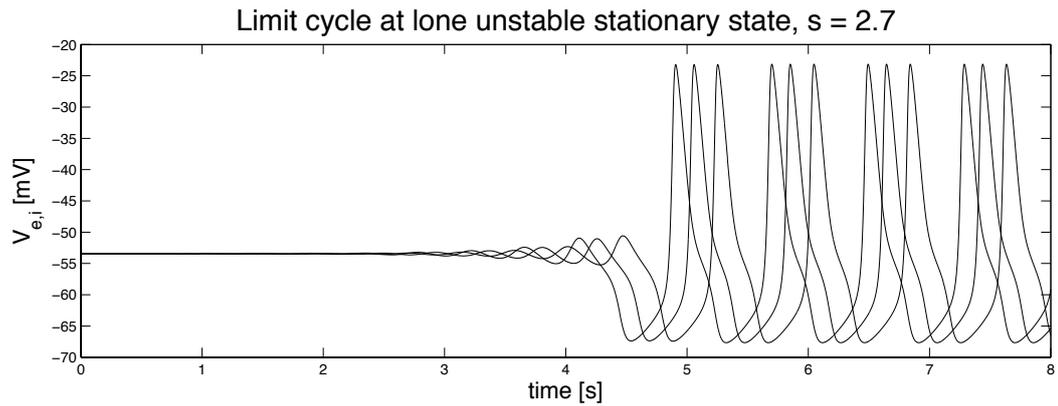


Figure 5.4: For the case of subcortical drive  $s$  only, a region of only unstable steady state is found ( $s = 2.7$ ) in which the system goes into a limit cycle. Here  $[C, \lambda_e, \lambda_i] = [0.78, 9.0, 1.0]$ .

resulted in limit cycles shown in Fig.5.5. Other starting points showed that the two directions of initial divergence both resulted in the same long term evolution of the system, a limit cycle shown in Fig.5.7 or a transition to another steady state as in Fig.5.6. The case of two separate transition states from one unstable point was not observed among the simulations.

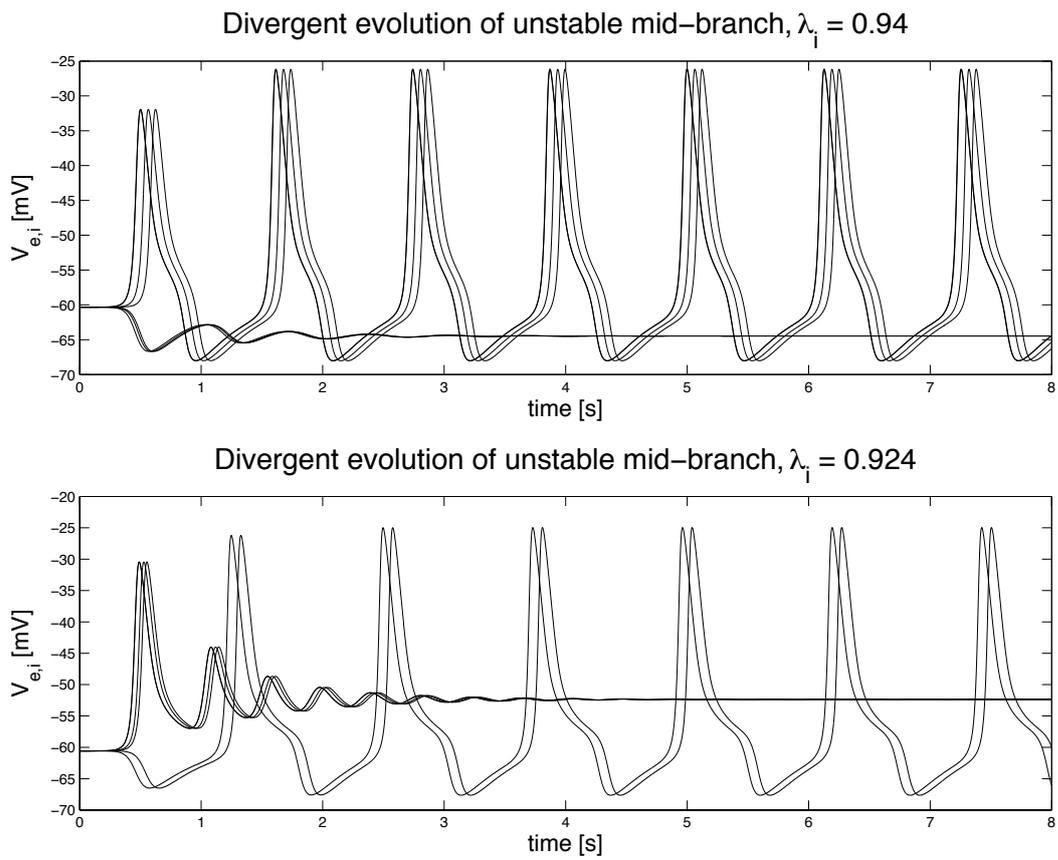


Figure 5.5: For certain points in the min-branch we get two distinct outcomes. One outcome leads to transition (upper to lower ( $\lambda_i=0.94$ )), lower to upper ( $\lambda_i=0.924$ ) and the other leads to limit cycling.  $[C, \lambda_e, s] = [0.78, 9.0, 0.25]$ .

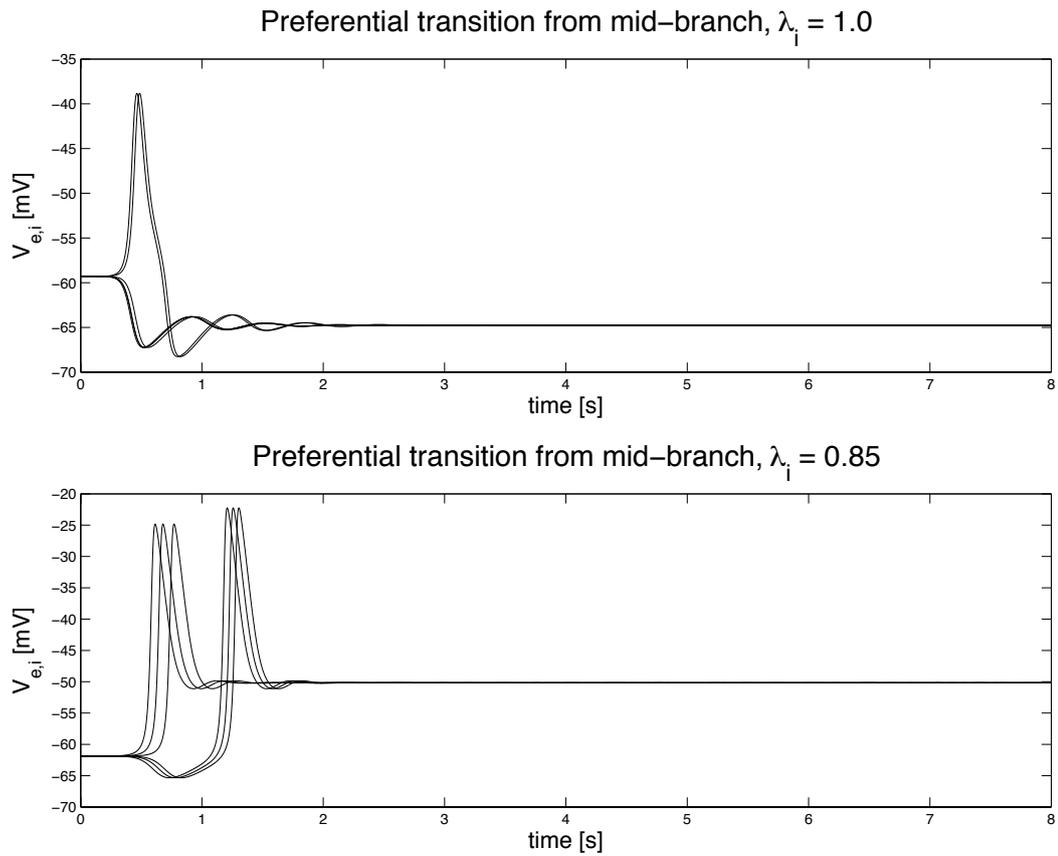


Figure 5.6: In some cases in the mid-branch, the initial divergence all resulted in a transition to another steady state, upper to lower when  $\lambda_i = 1.0$ , and lower to upper when  $\lambda_i = 0.85$ . Here  $[C, \lambda_e, s] = [0.78, 9.0, 0.25]$

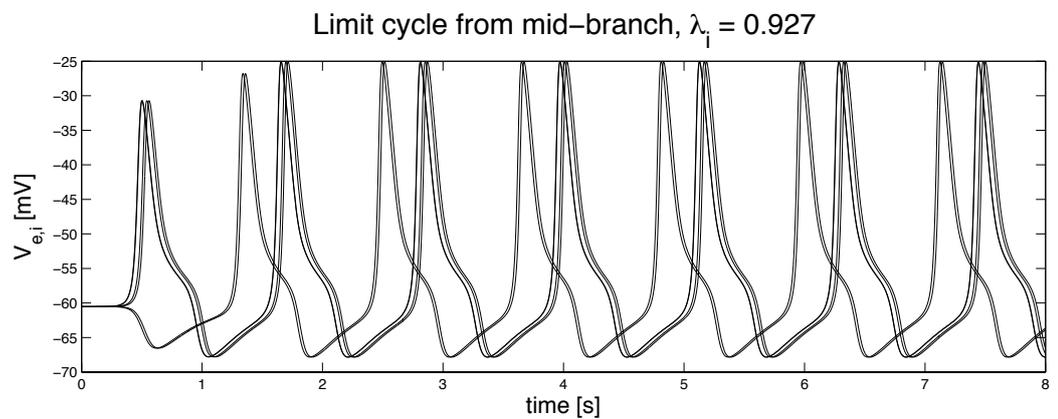


Figure 5.7: For some points in the mid-branch, the initial divergence all resulted in a limit cycle ( $\lambda_i = 0.927$ ).  $[C, \lambda_e, s] = [0.78, 9.0, 0.25]$ .



## Chapter 6

# Conclusions and further work

### 6.1 Overview

The main goal of this thesis was to investigate the behaviour of an electrodynamic model of the cortex, including the effect NMDA effects (an excitatory neurotransmitter). This was achieved by adopting the mean-field model developed by Steyn-Ross *et al.* (2005) which models the average behaviour of populations of neurons (neural mass). The effects of NMDA were introduced in our model by following Jahr and Stevens (1990). The main four input parameters used to define the neurotransmitter effect are: inhibitory neurotransmitter  $\lambda_i$ , subcortical drive  $s$ , NMDA excitatory neurotransmitters ( $\lambda_e$  and magnesium concentration  $C$ ). The theoretical stability predictions (Chapter 4) showed an agreement with the numerical simulations of the macrocolumn stochastic differential equations (Chapter 5).

#### 6.1.1 Equilibrium solutions

Our method relies on the investigation of the behaviour of small perturbations around the stationary states of the cortex. The stationary states were found by setting all time and space derivatives of the eight ODEs (3.48) to (3.54) equal to zero ( $d/dt = d/dx = 0$ ) and also the noise = 0. Since it was impossible to analytically solve for  $V_e$  and  $V_i$ , a numerical method was adopted. Two methods were used to determine the stationary states of the cortex: isocline method and root finding method (following Steyn-Ross (2002)).

It was found that the system can either have three or one stationary states depending on the NMDA concentration where the default values for  $[C, \lambda_e, \lambda_i, s]$  are  $[0.78, 9.0, 1.0, 0.25]$ . By varying each input parameter in turn, the distribution of the steady states was plotted resulting in S-bend shapes. Each S-bend has three distinct regions as shown in Fig.4.9: region I has a single low-firing state (“quiescent”); region II with multiple steady states; region III where a strong firing rate state (“highly activated”) can be found.

### 6.1.2 Theoretical stability predictions of the steady states

We computed the Jacobians to derive the stability of the system and found that the upper and lower branches are mostly stable. However, unlike the adiabatic anaesthesia modelling reported by Steyn-Ross (2002), the middle unstable region extends into the upper and lower branches, terminated by a pair distinct Hopf bifurcation points (where the real part of the dominant eigenvalue is equal to zero) because of the NMDA effect. Since the dominant eigenvalue (most positive or least negative) determines the stability of the steady states, the distribution of dominant eigenvalues was plotted for each input parameter (Figs.4.10, 4.11, 4.12 and 4.13). The previous stability classification was confirmed by the real-part dominant eigenvalue graphs where the stable branches (upper and lower) always have a negative dominant eigenvalue while the unstable branch has a dominant eigenvalue that is positive. The Hopf bifurcation (has no real part) is the point where the transition between the stable and unstable regions occur. The graph of the imaginary-part dominant eigenvalue(s) represents the oscillation component. Only the mid-branch of the unstable region has no oscillatory behaviour. The frequency of the upper branch is generally higher than that of the lower branch. For example, the Hopf point (H1) belonging to the upper branch has approximate frequency of 2.4 Hz which is higher than the frequency of the lower branch Hopf (H2) that was approximately 1.3 Hz.

The NMDA effect ( $C$  and  $\lambda_e$ ) on the macrocolumn stability was studied by comparing different values of  $C$  and  $\lambda_e$  in the S-bend plots of both  $\lambda_i$  and  $s$  (Figs.(4.14) and (4.15)). For the plot of  $\lambda_i$ , increasing the value of  $\lambda_e$  has the effect of shifting the graph to the right (excites the system), while increasing  $C$  result in shifting the graph to the left (aiding the inhibition). Increased  $\lambda_e$  will require less  $s$  to reach a high firing state and increased  $C$  will cancel out some of the effects of  $s$  and therefore needing more  $s$  to reach a high state. Overall, it can be seen that  $\lambda_e$  and  $C$  have opposite effect on steady-state stability.

### 6.1.3 Numerical simulation of the steady states

The system was simulated using a stochastic one-step Euler method with some noise added to the system in order to verify the stability results found by the theoretical stability predictions. Since the equations were strongly non-linear, the choice of time step had to be small in order for the Euler method to work ( $\Delta t = 10^{-3}$ ).

Many numerical experiments were carried out to study the behaviour of the steady states in stable and unstable regions, and the following was found:

- for the stable region on the upper and lower branches, the system fluctuates around the steady states and never moves far away. However, the closer the steady state to the Hopf point, the longer it takes for the system to decay back to the stationary state when the noise is turned off;
- at the Hopf points, when the noise was turned off, the oscillations were seen slowly diverging from the steady state, where theoretically the oscillation magnitude should

be constant;

- for the unstable region, the system either makes a transition between states or exhibits a limit cycle behaviour. If the stable destination is far enough from the Hopf the system make a transition and settles into the new equilibrium state. For certain starting points a divergence behaviour was found where in some cases one path led to a transition while the other path resulted in limit cycles. In other cases, the two direction of initial divergence both resulted in the same long term evolution of the system either to a limit cycle or to a transition between states. A final case occurs at the critical value of subcortical drive  $s$  where a region of only unstable steady state is found, in which the system goes into a limit cycle.

The listed results of the numerical simulation agree with the results of the stability predictions. The oscillations that are seen in our results might be caused by the instability in the stationary states. Traditionally the cortical oscillations have been interpreted by neuroscientists as being caused by either delay reverberation in subcortical-cortical circuits, or at the level of individual neurons, oscillations in various membrane currents. These situations may be sufficient, but are not necessary, for the production of the cortical rhythms.

#### 6.1.4 Further work

This thesis assumed a homogenous cortex in the stability analysis and simulations (i.e.,  $q = 0$ ). Hence it would be worthwhile demonstrating the results of adding space to the system.

The case of subcortical drive ( $s$ ) is slightly different than the other three parameters ( $\lambda_i$ ,  $\lambda_e$ ,  $C$ ). Therefore, it might be beneficial if it would be studied in more detail. Finally, the different oscillatory behaviours of the steady states need to be physiologically explained.



# Appendix A

## Matlab code

### A.1 Sigmoid functions and their derivatives

```
% sigmoid.m
% Drawing the graphs of Sigmoid and its derivative

dV = 0.01;
V = [-100: dV: -20];          % values of V [mV]
ge = .30; gi = .18;
the = -60; thi = -60;

%define smooth Sigmoid curve and its gradient
Qe = 100./(1 + exp(-ge.*(V - the)));
Qi = 100./(1 + exp(-gi.*(V - thi)));

dqe = 100*ge.*exp(-ge.*(V - the))./(1 + exp(-ge.*(V - the))).^2;
dqi = 100*gi.*exp(-gi.*(V - thi))./(1 + exp(-gi.*(V - thi))).^2;

figure(1); clf;

subplot (121);
    plot(V, Qe, 'k');
    hold on;
    plot(V, Qi, 'k', 'linewidth', 2);
    hold on;
    xlabel ('Soma Potential [mV]', 'fontsize', 18);
    title ( '(a) Firing rate vs Voltage', 'fontsize', 20);
    legend ('\it{Q_e} ', '\it{Q_i}', 'Location', 'NW')
    hold on; grid on; zoom on;
```

```

subplot (122);
plot(V, dqe, 'k');
hold on;
plot(V,dqi, 'k','linewidth', 2);
hold on;
xlabel ('Soma Potential [mV]', 'fontsize', 18);
title ( '(b) First-derivative', 'fontsize', 20);
legend ('d{\itQ}_{e} /d{\itV}_{e} ', 'd{\itQ}_{i} /d{\itV}_{i} ',...
        'Location', 'NW');
hold on; grid on; zoom on;

% Using the numerical differentiation to verify that the analytic
% derivative curves are correct.

deltaV = 0.001;
% we choose this size of deltaV because when the step is small we get
% better result but we try not to make it very small because it will
% take long time to run.
V = [-100: deltaV: -20];

% define sampled Sigmoid and its gradient
Se = 100./(1 + exp(-ge.*(V - the)));
Si = 100./(1 + exp(-gi.*(V - thi)));

dse = 100*ge.*exp(-ge*(V - the))./(1 + exp(-ge*(V - the))).^2;
dsi = 100*gi.*exp(-gi*(V - thi))./(1 + exp(-gi*(V - thi))).^2;

% we choose the centred-derivative because it gives better estimation for
% the gradient.
% estimate the gradient via r centred-derivative

cd_Se = (Se(3:end) - Se(1:end-2))/(2*deltaV);
cd_Se = [NaN cd_Se NaN];
cd_Si = (Si(3:end) - Si(1:end-2))/(2*deltaV);
cd_Si = [NaN cd_Si NaN];

% compute discrepancy between numerical and true derivatives
err_e = cd_Se - dse;
err_i = cd_Si - dsi;

```

```

figure (2); clf;
hold on;
plot (V,err_e,'k','linewidth', 2);
plot (V,err_i,'k');
xlabel('Increment,V');    ylabel(' Error');
title('Numerical Error vs V');
legend('SeError', 'SiError');
hold on; grid on; zoom on;

```

## A.2 Normalised conductance for NMDA

```

% function  g_NMDA = JS_NMDA(V,C)
% Plotting the normalized conductance for NMDA receptor using ...
% Jahr & Stevens (1990).
% g_NMDA = 1 ./ (1 + k*C*exp(-a*V)).
% Inputs: V = excitatory voltage (mV), C = Mg concentration (mMol)

clf;
[k a] = deal(1/3.57, 0.062); % Experimental constants [1/mM, 1/mV]

for c = 1:10;          % [mM]
V = [-70: 0.05: 100]; % [mv]
g_NMDA = (1 + k*c*exp(-a*V)).^-1;

figure(555);
plot (V,g_NMDA, 'k', 'linewidth', 1); hold on;

end
title('Conductance of NMDA-receptor (g) vs Excitatory Voltage (V_e)',...
      'fontsize', 18);
xlabel('V_e [mV]', 'fontsize', 18);
ylabel('g [\Omega^{-1}]' , 'fontsize', 18);

axis([-75 100 0 1.1]);

grid on;

```

## A.3 Global initial constants of the cortical model

```
function W = nmda_init_globs
```

```

% Version of init_conformal_globs() customized for
% Waleed's NMDA thesis project
% We initialize a global structure named 'W'
% (previously 'H' for Hasselmo)
% ASR: 24-Aug-2007

% membrane rate-constants ("Sleep Cycle" J.Biol.Px [2005] values)
[W.tau_e, W.tau_i] = deal(50/1000, 50/1000); % (sec)

% sigmoid characteristics
[W.Qe_max, W.Qi_max] = deal(30, 60); % sigmoid maximum (s^-1)
[W.theta_e, W.theta_i] = deal(-58.5, -58.5); % sigmoid threshold (mV)

[W.sigma_e, W.sigma_i] = deal(4.0, 6.0);

% gain per synapse at resting voltage (millivolt.sec)
[W.rho_e, W.rho_i] = deal(1.00e-3, -1.05e-3);

% voltage limits
[W.Ve_rev, W.Vi_rev] = deal(0, -70); % reversal potential (mV)
[W.Ve_rest, W.Vi_rest] = deal(-64, -64); % resting potential (mV)

% connectivities: j2k convention (dimensionless)
% cortico-cortical
[W.Nee_a, W.Nei_a] = deal(3710, 3710);
% intracortical
[W.Nee_b, W.Nei_b, W.Nie_b, W.Nii_b] = deal(410, 410, 800, 800);
% subcortical
[W.Nee_sc, W.Nei_sc, W.Nie_sc, W.Nii_sc] = deal(50, 50, 0, 0);

% Nee and Nie totals for cortico-cortical plus intracortical
W.Nee_ab = W.Nee_a + W.Nee_b;
W.Nei_ab = W.Nei_a + W.Nei_b;

% default subcortical fluxes
W.phi_ee_sc = W.Nee_sc * W.Qe_max;
W.phi_ei_sc = W.Nei_sc * W.Qe_max;
W.phi_ie_sc = 0; % no inhibitory subcortical flux!
W.phi_ii_sc = 0;

% d/dV derivatives of psi_ij weighting functions
W.d_psi_ee = -1/(W.Ve_rev - W.Ve_rest);

```

```

W.d_psi_ei = -1/(W.Ve_rev - W.Vi_rest);
W.d_psi_ie = -1/(W.Vi_rev - W.Ve_rest);
W.d_psi_ii = -1/(W.Vi_rev - W.Vi_rest);

% derived AMPA/GABA rate-constants (sec^-1) for Wright model (j2k)
% [gamma_ee, gamma_ei, gamma_ie, gamma_ii] = deal(57.4, 82.5, 51.3, 65.8);
% simplified rate constants for MSR's model
W.gamma_e = 70.0; % mean([57.4, 82.5])
%W.gamma_i = 58.6; % mean([51.3, 65.8])
W.gamma_i = 15;

% velocity and space-constant
% W.Lambda = 40; % inverse-length scale for connectivity (metre^-1)
% W.v = 9; % axonal conduction velocity (m/s)
W.Lambda = 0.4; % inverse-length scale for connectivity (/cm)
W.v = 900; % axonal conduction velocity (cm/s)

return

%-----

```

## A.4 NMDA steady states finder

```

function [Qe_root, Qi_root, Ve_root, Vi_root] = nmda_ss_finder(...
Mg_conc, lambda_e, lambda_i, sc, graphFlag, Nsearch)
% Find steady states for Waleed's NMDA project.
% Inputs:
%     Mg_conc (in mM) determines NMDA scaling of rho_e (EPSP area)
%     lambda_e gives independent scaling of rho_e (EPSP area)
%     lambda_i scales rho_i (IPSP area)
%     sc scales ee and ei subcortical flux
%
% This code based on conformal_ss_finder().
% This finder assumes full 'conformal' symmetry,
% thus at steady state, Ve = Vi (but Qe is not equal to Qi).

% 25-Aug-2007: (ASR) First version

if nargin == 0
    % These are the default values of the four parameters controlling
    % the cortical model.
    [Mg_conc, lambda_e, lambda_i, sc] = deal(0.78, 9.0, 1.0, 0.25);

```

```

    graphFlag = 1;
Nsearch = 8000;
end

global W
W = nmda_init_globs;

n_roots = 0;
Nsearch_max = 8*128000;

% if no roots found, keep increasing search density until
% Nsearch_max reached
% NB: We treat zero roots or 2 roots as error condition
% (expect odd number of roots)
while((n_roots == 0 | n_roots == 2) & Nsearch <= Nsearch_max)

% zeros of F1 give the steady states
%(expect F2 zeros to be identical)

Qe = linspace(0, W.Qe_max, Nsearch)';
[Ve, Vi] = deal(invQsige(Qe));

g_NMDA = JS_NMDA(Ve, Mg_conc);

term1 = W.Ve_rest - Ve;
term2 = lambda_e*g_NMDA*W.rho_e .* Psi_ee(Ve) .* ...
(W.Nee_ab * Qsige(Ve) + sc*W.phi_ee_sc);
term3 = lambda_i*W.rho_i * Psi_ie(Ve) .* W.Nie_b .* Qsige(Vi);

F1 = (term1 + term2 + term3)/W.tau_e;

% plot F2 residuals (should be identical to F1!)
% term1_i = W.Vi_rest - Vi;
% term2_i = lambda_e*(1 + g_NMDA)*W.rho_e .* Psi_ei(Vi) .* ...
% (W.Nei_ab * Qsige(Ve) + sc*W.phi_ei_sc);
% term3_i = lambda_i*W.rho_i * Psi_ii(Vi) .* W.Nii_b .* Qsige(Vi);
%
% F2 = (term1_i + term2_i + term3_i)/W.tau_i;

% Count the number of F1 roots (look for sign-changes)
chs_F1 = F1(1:end-1) .* F1(2:end);

```

```

chs_F1_index = find(chs_F1 < 0);

n_roots = length(chs_F1_index);
if (n_roots == 0 | n_roots == 2)
Nsearch = 2*Nsearch;
%disp(['Doubling Nsearch to ' num2str(Nsearch)]);
end
end

if n_roots == 0
    disp(['...failed to find a root at [Mg_conc lambda_e lambda_i sc]=['...
num2str([Mg_conc, lambda_e, lambda_i, sc]) '] ---skipped! '])
end

% chs_F1_index

% Form a bracketting interval for each root
if n_roots >= 1
brack_Ve = [Ve(chs_F1_index) Ve(chs_F1_index + 1)];

% Check for NaN at either end of bracket
if any(isnan(brack_Ve(:)))
error('Detected NaN in a bracket!');
end
end

% Polish each Ve root
Qe_root = NaN*ones(n_roots, 1);
Qi_root = NaN*ones(n_roots, 1);
Ve_root = NaN*ones(n_roots, 1);
Vi_root = NaN*ones(n_roots, 1);
for i = 1: n_roots
Ve_root(i) = fzero('F1_resid', brack_Ve(i,:), [], [], ...
Mg_conc, lambda_e, lambda_i, sc);
Qe_root(i) = Qsige(Ve_root(i));
Vi_root(i) = Ve_root(i);
Qi_root(i) = Qsigi(Vi_root(i));
end

if nargin == 0
disp([n_roots*ones(size(Qe_root)) Qe_root Qi_root Ve_root Vi_root]);
disp([Mg_conc lambda_e lambda_i sc]);

```

```

end

%-----
if ~graphFlag
return
end

% plot graphs of steady-state error residuals
figure(2); clf;
subplot(121);
plot(Qsize(Ve), F1, 'color',[0.7 0.7 0.7], 'linewidth', 3); hold on;
    plot(Qsigi(Ve), F1, 'K-', 'linewidth', 3);
    legend('Q_{e}', 'Q_{i}', 'fontsize', 18);
    plot(Qe_root(:,1), 0*Qe_root(:,1), 'ko', 'linewidth', 1,...
        'markersize', 10);
plot(Qi_root(:,1), 0*Qi_root(:,1), 'ks', 'linewidth', 1,...
    'markersize', 10);
    ax = axis;
    plot(ax([1 2]), [0 0], 'k-'); hold off;
    title('F1 vs Q_{e,i}', 'fontsize', 26);
ylabel('F1 [mV/s]', 'fontsize', 22);
xlabel('Q_{e,i} [s^{-1}]', 'fontsize', 22);
    zoom on;
subplot(122);
plot(Ve, F1, 'K-', 'linewidth', 3); zoom on; hold on;
plot(Ve_root(:,1), 0*Ve_root(:,1), 'ko', 'linewidth', 1,...
    'markersize', 10); zoom on;
%plot(Vi_root(:,1), 0*Vi_root(:,1), 'kx', 'linewidth', 1,...
    %    'markersize', 15); zoom on; grid on;
    ax = axis;
    plot(ax([1 2]), [0 0], 'k-'); hold off;
    hold off;
title('F1 vs V_{e,i}', 'fontsize', 26);
ylabel('F1 [mV/s]', 'fontsize', 22);
xlabel('V_{e,i} [s^{-1}]', 'fontsize', 22);

if graphFlag, drawnow, end

return

%-----

```

```

%-----
function err_F1 = F1_resid(Ve, Mg_conc, lambda_e, lambda_i, sc)
% Compute F1 = dVe/dt. Assuming conformal symmetry, the zeros of F1 will
% be identical to the zeros of F2 = dVi/dt. This vastly simplifies the
% search for steady states.

global W

Vi = Ve;
g_NMDA = JS_NMDA(Ve, Mg_conc);

term1 = W.Ve_rest - Ve;
term2 = lambda_e*g_NMDA*W.rho_e .* Psi_ee(Ve) .* ...
(W.Nee_ab * Qsige(Ve) + sc*W.phi_ee_sc);
term3 = lambda_i*W.rho_i * Psi_ie(Ve) .* W.Nie_b .* Qsige(Vi);

err_F1 = (term1 + term2 + term3)/W.tau_e;

return

%-----
function [x_clean, y_clean] = filterNaN(x, y)
% Given vectors x and y, return the subset vectors than contain no NaN
% elements. We treat the (x,y) as ordered pairs, and require that neither
% element of a pair contain an NaN.

valids = isfinite(x .* y);
[x_clean y_clean] = deal(x(valids), y(valids));

return

%-----
function g_NMDA = JS_NMDA(V,C)
% Compute normalized conductance for NMDA receptor using
% Jahr & Stevens (1990)
% Inputs: V = excitatory voltage (mV), C = Mg concentration (mMol)

[k a] = deal(1/3.57, 0.062); % 1/mM, 1/mV

g_NMDA = 1 ./ (1 + k*C*exp(-a*V));
return

```

```

%-----
function QC_plot_JS_NMDA
% Plot the Jahr & Stevens (1990) NMDA conductance
% curves for stepped Mg conc

V = linspace(-100, 100, 500); % mV
conc_set = logspace(0, 2, 5); % mM concentration for extracellular Mg

figure(1); clf;
for i = 1: length(conc_set);
C = conc_set(i);
g = JS_NMDA(V, C);
plot(V, g); hold on; zoom on; grid on;
end
return

%-----
function Qe = Qsige(V)
% Excitatory sigmoid function for Wright model; input V in millivolts
% Previous Liley defn:
% sig = Se_max./(1 + exp(-ge*(h - theta_e)) );

global W

gain = pi/(sqrt(3) * W.sigma_e);
temp = 1./(1 + exp(-gain*(V - W.theta_e)));
Qe = W.Qe_max * temp;
return

%-----
function Qi = Qsigi(V)
% Inhibitory sigmoid function for Wright model
% Previous Liley defn:
% sig = Si_max./(1 + exp(-gi*(h - theta_i)) );

global W

gain = pi/(sqrt(3) * W.sigma_i);
temp = 1./(1 + exp(-gain*(V - W.theta_i)));
Qi = W.Qi_max * temp;
return

```

```

%-----
function d_Qe = d_Qsige(V)
% d/dV derivative of excitatory sigmoid function for Wright model;
% input V in millivolts

global W

gain = pi/(sqrt(3) * W.sigma_e);
E = exp(-gain*(V - W.theta_e));
d_Qe = gain * W.Qe_max * E ./ (1 + E).^2;
return

%-----
function d_Qi = d_Qsigi(V)
% d/dV derivative of inhibitory sigmoid function for Wright model

global W

gain = pi/(sqrt(3) * W.sigma_i);
E = exp(-gain*(V - W.theta_i));
d_Qi = gain * W.Qi_max * E ./ (1 + E).^2;
return

%-----
function invsig = invQsige(val)
% Inverse of excitatory sigmoid function; output in millivolts

global W
gain = pi/(sqrt(3) * W.sigma_e);
invsig = NaN*ones(size(val));
ok = find(val > 0 & val < W.Qe_max);

invsig(ok) = W.theta_e - (1/gain)*log(W.Qe_max./val(ok) - 1);
return

%-----
function invsig = invQsigi(val)
% Inverse of inhibitory sigmoid function

global W
gain = pi/(sqrt(3) * W.sigma_i);
invsig = NaN*ones(size(val));

```

```

ok = find(val > 0 & val < W.Qi_max);

invsig(ok) = W.theta_i - (1/gain)*log(W.Qi_max./val(ok) - 1);
return

%-----
function weight = Psi_ee(V)
% Wright form for weighting function, but note use of i2j notation

global W
weight = (W.Ve_rev - V)/(W.Ve_rev - W.Ve_rest);
return

%-----
function weight = Psi_ei(V)

global W
weight = (W.Ve_rev - V)/(W.Ve_rev - W.Vi_rest);
return

%-----
function weight = Psi_ie(V)

global W
weight = (W.Vi_rev - V)/(W.Vi_rev - W.Ve_rest);
return

%-----
function weight = Psi_ii(V)

global W
weight = (W.Vi_rev - V)/(W.Vi_rev - W.Vi_rest);
return

%-----
function QC_plot_sigmoids
% Quality control check: Plot sigmoids

global W
W = nmda_init_globs;

num_pts = 1e3;

```

```

Ve = linspace(invQsige(1e-10), invQsige(W.Qe_max - 1e-10), num_pts)';
Qe = Qsige(Ve);

```

```

Vi = linspace(invQsigi(1e-10), invQsigi(W.Qi_max - 1e-10), num_pts)';
Qi = Qsigi(Vi);

```

```

figure(1);
subplot(111);
plot(Ve, Qe, '-r', Vi, Qi, '-b'); zoom on; grid on;
title('Sigmoids: Firing rate vs Soma voltage (mV)', 'fontsize', 16);
legend('Qe', 'Qi');
return

```

```

%-----
function QC_deriv_sigmoids
% Quality control check for d_Qsige() and d_Qsigi()

global W

W = nmda_init_globs;
num_pts = 1e3;

Ve = linspace(invQsige(1e-10), invQsige(W.Qe_max - 1e-10), num_pts)';
Qe = Qsige(Ve);
dQe = d_Qsige(Ve); % analytic derivative
dQe_2 = cdiff(Qe)/(Ve(2) - Ve(1)); % numerical derivative

Vi = linspace(invQsigi(1e-10), invQsigi(W.Qi_max - 1e-10), num_pts)';
Qi = Qsigi(Vi);
dQi = d_Qsigi(Vi); % analytic derivative
dQi_2 = cdiff(Qi)/(Vi(2) - Vi(1)); % numerical derivative

% compare derivative curves with Gaussian approximation
scale = max(dQe);
z = (Ve - W.theta_e)/W.sigma_e;
gauss_e = 1/(W.sigma_e*sqrt(2*pi))*exp(-z.^2/2);

figure(2); clf;
% plot(Ve, gauss_e, Ve, dQe/W.Qe_max); zoom on; grid on;
plot(Ve, cdiff(gauss_e), Ve, cdiff(dQe/W.Qe_max)); zoom on; grid on;

```

```

figure(1);
subplot(221);
plot(Ve, dQe, '-r', Ve, dQe_2, '-b'); zoom on; grid on;
title('Q_e deriv: analytic (red) vs numerical (blue)', 'fontsize', 16);
subplot(222);
plot(Ve, dQe - dQe_2, '-r'); zoom on; grid on;
title('Q_e deriv: Error residual', 'fontsize', 16);
subplot(223);
plot(Vi, dQi, '-r', Vi, dQi_2, '-b'); zoom on; grid on;
title('Q_i deriv: analytic (red) vs numerical (blue)', 'fontsize', 16);
subplot(224);
plot(Vi, dQi - dQi_2, '-b'); zoom on; grid on;
title('Q_i deriv: Error residual', 'fontsize', 16);

return

%-----
function b = fzero(FunFcn,x,tol,trace,varargin)
%FZERO Find zero of function of one variable.
% FZERO(F,X) tries to find a zero of F. F is a string containing
% the name of a real-valued function of a single real variable.
% The value returned is near a point where F changes sign, or NaN
% if the search fails.
%
% FZERO(F,X), where X is a vector of length 2, assumes X is an
% interval where the sign of F(X(1)) differs from the sign of F(X(2)).
% An error occurs if this is not true. Calling FZERO with an interval
% guarantees FZERO will return a value near a point where F changes
% sign.
%
% FZERO(F,X), where X is a scalar value, uses X as a starting guess.
% FZERO looks for an interval containing a sign change for F and
% containing X. If no such interval is found, NaN is returned.
% In this case, the search terminates when the search interval
% is expanded until an Inf, NaN, or complex value is found.
%
% FZERO(F,X,TOL) sets the relative tolerance for the convergence test.
% FZERO(F,X,TOL,TRACE) displays information at each iteration when
% TRACE is nonzero.
% FZERO(F,X,TOL,TRACE,P1,P2,...) allows for additional arguments
% which are passed to the function, F(X,P1,P2,...). Pass an empty
% matrix for TOL or TRACE to use the default value.

```

```
%
% Examples
%     fzero('sin', 3) returns pi. Note the quotes around sin.
%     Ordinarily, functions are defined in M-files.
%     fzero('abs(x)+1', 1) returns NaN since this function does
%     not change sign anywhere on the real axis (and does not have
%     a zero as well).
%     fzero('sin', 3, [], 1) returns pi, uses the default tolerance,
%     and displays iteration information.
%
% See also ROOTS.

% Copyright (c) 1984-96 by The MathWorks, Inc.
% $Revision: 5.10 $ $Date: 1996/10/28 22:13:28 $

% This algorithm was originated by T. Dekker. An Algol 60 version,
% with some improvements, is given by Richard Brent in "Algorithms for
% Minimization Without Derivatives", Prentice-Hall, 1973. A Fortran
% version is in Forsythe, Malcolm and Moler, "Computer Methods
% for Mathematical Computations", Prentice-Hall, 1976.

% Initialization

if nargin < 3 | isempty(tol), tol = eps; end
if nargin < 4 | isempty(trace), trace = 0; end
if trace
    header = ' Func evals      x              f(x)              Procedure';
    step=' ';
    count = 0;
end
if (~isfinite(x))
    error('Second argument must be finite.')
end

% Convert to inline function as needed.
FunFcn = fcnchk(FunFcn,length(varargin));

% Interval input
if (length(x) == 2)
    a = x(1);
    b = x(2);
    fa = feval(FunFcn,a,varargin{:});
```

```

fb = feval(FunFcn,b,varargin{:});
if any(~isfinite([fa fb])) | any(~isreal([fa fb]))
    error('Function values at interval endpoints must be finite and real.')
```

---

```

end
if trace
    disp(header)
    data = [a fa]; step='          initial';
    count = count + 1;
    disp([sprintf('%5.0f   %13.6g %13.6g ',count, data), step])
    data = [b fb]; step = '          initial';
    count = count + 1;
    disp([sprintf('%5.0f   %13.6g %13.6g ',count, data), step])
end
if (fa > 0) == (fb > 0)
    error(['The function values at the' ...
          'interval endpoints must differ in sign.'])
end

% Starting guess scalar input
elseif (length(x) == 1)
    fx = feval(FunFcn,x,varargin{:});
    if fx == 0
        b = x;
        return
    elseif ~isfinite(fx) | ~isreal(fx)
        error('Function value at starting guess must be finite and real.');
```

---

```

    end
    if trace
        disp(header)
        data = [x fx]; step='          initial';
        count = count + 1;
        disp([sprintf('%5.0f   %13.6g %13.6g ',count, data), step])
    end
    if x ~= 0,
        dx = x/50;
    else,
        dx = 1/50;
    end

% Find change of sign.
twosqrt = sqrt(2);
a = x; fa = fx; b = x; fb = fx;
```

```

while (fa > 0) == (fb > 0)
    dx = twosqrt*dx;
    a = x - dx; fa = feval(FunFcn,a,varargin{:});
    if ~isfinite(fa) | ~isreal(fa)
        disperr(a,fa);
        b = NaN;
        return
    end
    if trace
        data = [a fa]; step='          search';
        count = count + 1;
        disp([sprintf('%5.0f   %13.6g %13.6g ',count, data), step])
    end
    if (fa > 0) ~= (fb > 0)
        break
    end
    b = x + dx; fb = feval(FunFcn,b,varargin{:});
    if ~isfinite(fb) | ~isreal(fb)
        disperr(b,fb);
        b = NaN;
        return
    end
    if trace
        data = [b fb]; step='          search';
        count = count + 1;
        disp([sprintf('%5.0f   %13.6g %13.6g ',count, data), step])
    end
end % while
if trace
    disp(' ')
    disp([' Looking for a zero in the interval [', ...
        num2str(a) , ', ', num2str(b), ']'']);
    disp(' ')
end
else
    error('Second argument must be of length 1 or 2.');
```

```

end % if (length(x) == 2

fc = fb;
% Main loop, exit from middle of the loop
while fb ~= 0
```

```

% Insure that b is the best result so far, a is the previous
% value of b, and c is on the opposite of the zero from b.
if (fb > 0) == (fc > 0)
    c = a; fc = fa;
    d = b - a; e = d;
end
if abs(fc) < abs(fb)
    a = b; b = c; c = a;
    fa = fb; fb = fc; fc = fa;
end

% Convergence test and possible exit
m = 0.5*(c - b);
toler = 2.0*tol*max(abs(b),1.0);
if (abs(m) <= toler) + (fb == 0.0), break, end

% Choose bisection or interpolation
if (abs(e) < toler) + (abs(fa) <= abs(fb))
% Bisection
    d = m; e = m;
    step=' bisection';
else
% Interpolation
    s = fb/fa;
    if (a == c)
% Linear interpolation
        p = 2.0*m*s;
        q = 1.0 - s;
    else
% Inverse quadratic interpolation
        q = fa/fc;
        r = fb/fc;
        p = s*(2.0*m*q*(q - r) - (b - a)*(r - 1.0));
        q = (q - 1.0)*(r - 1.0)*(s - 1.0);
    end;
    if p > 0, q = -q; else p = -p; end;
% Is interpolated point acceptable
    if (2.0*p < 3.0*m*q - abs(toler*q)) * (p < abs(0.5*e*q))
        e = d; d = p/q;
        step=' interpolation';
    else
        d = m; e = m;
    end
end

```

```

        step='        bisection';
    end;
end % Interpolation

% Next point
a = b;
fa = fb;
if abs(d) > toler, b = b + d;
else if b > c, b = b - toler;
    else b = b + toler;
    end
end
fb = feval(FunFcn,b,varargin{:});
if trace
    data = [b fb];
    count = count + 1;
    disp([sprintf('%5.0f    %13.6g %13.6g ',count, data), step])
end
end % Main loop
return

%-----
function [f,msg] = fcnchk(fun,varargin)
%FCNCHK Check FUNFUN function argument.
% FCNCHK(FUN,...) returns either a function name string or an
% inline object depending on whether the string contains an expression
% involving parentheses, variables, and math operators.
%
% FCNCHK is a helper function for FMIN, FMINS, FZERO, etc. so they
% can compute with string expressions in addition to m-file functions.
%
% FCNCHK(FUN,...,'vectorized') processes the string (e.g., replacing
% '*' with '.*') to produce a vectorized function.
%
% When FUN contains an expression then FCNCHK(FUN,...) is the same as
% INLINE(FUN,...) except that the optional trailing argument 'vectorized'
% can be used to produce a vectorized function.
%
% [F,MSG] = FCNCHK(...) returns an empty string in MSG if successful
% or an error message string if not.
%
% See also INLINE.

```

```

% Copyright (c) 1984-98 by The MathWorks, Inc.
% $Revision: 1.8 $ $Date: 1997/11/21 23:30:42 $

msg = '';

nin = nargin;
if (nin>1) & strcmp(varargin{end},'vectorized')
    vectorizing = 1;
    nin = nin-1;
else
    vectorizing = 0;
end

if isstr(fun)
    fun = strtrim(fun);
    % Check for non-alphanumeric characters that must be part of an
    % expression.
    if isempty(fun),
        f = inline('[]');
    elseif ~vectorizing & isidentifier(fun)
        f = fun; % Must be a function name only
    else
        if vectorizing
            f = inline(vectorize(fun),varargin{1:nin-1});
            var = argnames(f);
            f = inline([formula(f) '.*ones(size(' var{1} '))'],'',var{1:end});
        else
            f = inline(fun,varargin{1:nin-1});
        end
    end
end
elseif isa(fun,'inline')
    f = fun;
else
    f = '';
    msg = 'FUN must be a function name or inline function object.';
end
return

%-----
function F = vectorize(F)
%VECTORIZE Vectorize a symbolic expression.

```

```

% VECTORIZE(F) inserts a '.' before any '^', '*' and '/' in F.

l = length(F);
for k = fliplr(find((F=='^') | (F=='*') | (F=='/'))))
    F = [F(1:k-1) '.' F(k:l)];
    l = l+1;
end
F(findstr(F, '..')) = []; % Remove any possible ..*, ../, etc.
return

%-----
function s1 = strtrim(s)
%STRTRIM Trim spaces from string.

if ~isempty(s) & ~isstr(s)
    warning('Input must be a string.')
end

if isempty(s)
    s1 = s;
else
    % remove leading and trailing blanks (including nulls)
    c = find(s ~= ' ' & s ~= 0);
    s1 = s(min(c):max(c));
end
return

%-----
function tf = isidentifier(str)

tf = 0;

if ~isempty(str)
    first = str(1);
    if (isletter(first))
        letters = isletter(str);
        numerals = (48 <= str) & (str <= 57);
        underscore = (95 == str);
        if (all(letters | numerals | underscore))
            tf = 1;
        end
    end
end
end

```

```

end

tf = logical(tf);
return

%-----
function disperr(y, fy)
%DISPERR Display an appropriate error message when FY is Inf,
% NaN, or complex. Assumes Y is the value and FY is the function
% value at Y. If FY is neither Inf, NaN, or complex, it generates
% an error message.

if ~isfinite(fy) % NaN or Inf detected
    disp('NaN or Inf function value encountered during ');
    disp(' search for an interval containing a sign change. ');
    disp(['Function value at ', num2str(y), ' is ', num2str(fy)]);
    disp('Aborting since no such interval was found. ');
    disp('Check function or try again with a different starting value. ');
elseif ~isreal(fy) % Complex value detected
    disp('Complex function value encountered during ');
    disp(' search for an interval containing a sign change. ');
    disp(['Function value at ', num2str(y), ' is ', num2str(fy)]);
    disp('Aborting since no such interval was found. ');
    disp('Check function or try again with a different starting value. ');
else
    error('Disperr called with invalid argument. ')
end

%-----

```

## A.5 Locus of equilibrium states

```

function [V_ss, Var_ss, Q_e_ss, Q_i_ss] = variable_ss_finder...
    (varname, params, npoints, startval, endval, graphflag, Nsearch)
%[V_ss, Var_ss, Q_e_ss, Q_i_ss] = variable_ss_finder...
% (varname, params, npoints, startval, endval, graphflag, Nsearch)
%INPUTS
% varname - the varying parameter
% params = [mg_conc, lambda_e, lambda_i, sc]
% npoints - number of points
% startval, endval of parameter being varied
% graphflag - to graph or not, 0/1

```

```

%OUTPUTS
% V_ss - steadystate voltages  $V_e = V_i = V_{ss}$ 
% Var_ss - parameter being varied corresponding to  $V_{ss}$ , contains multiple
% values if multiple roots exist at that point.
% Q_e_ss, Q_i_ss - steadystate  $Q_e, Q_i$  values.
%
%Plots the locus of steady states for the cortical model controlled by the
% parameter [varname].

if nargin >=2    % if input is given then use them
    [mg_conc, lambda_e, lambda_i, sc] = ...
        deal(params(1),params(2),params(3),params(4));
else    % else if input parameters are not defined then use default values
    [mg_conc, lambda_e, lambda_i, sc] = deal(0.78, 9.0, 1.0, 0.25);
end

if nargin < 7
    Nsearch = 8000;
    if nargin < 5
        graphflag = 0;
        if nargin <= 2
            npoints = 2000;
            if nargin < 1
                varname = 'mg_conc';
                graphflag = 1;
            end
        end
    end
end

end

% if endpoints are not given then set to default range
endpoints_on = (nargin <= 3);

if strcmp(varname,'mg_conc')
    variablename = 'Magnesium Concentration,C [mM]';
    if endpoints_on
        startval = 0.6;
        endval    = 1.00;
    end
elseif strcmp(varname,'lambda_e')
    variablename = '\{\lambda\}_{e}';

```

```
    if endpoints_on
        startval = 8.00;
        endval   = 12.00;
    end
elseif strcmp(varname, 'lambda_i')
    variablename = '{\lambda}_{i}';
    if endpoints_on
        startval = 0.65;
        endval   = 1.10;
    end
elseif strcmp(varname, 'sc')
    variablename = 'Subcortical drive, s';
    if endpoints_on
        startval = -5.00;
        endval   = 5.00;
    end
end
end

vari = linspace(startval,endval,npoints);

% Approximately over-allocate and then shorten length later
V_ss = zeros(3*npoints,1);
Var_ss = zeros(3*npoints,1);
Q_e_ss = zeros(3*npoints,1);
Q_i_ss = zeros(3*npoints,1);

% exact preallocation
if graphflag
    v_max = zeros(size(vari));
    v_min = zeros(size(vari));

    q_e_max = zeros(size(vari));
    q_e_min = zeros(size(vari));
    q_i_max = zeros(size(vari));
    q_i_min = zeros(size(vari));
end

lengthcounter = 0;
```

```
for i = 1:length(vari)
    switch varname
        case 'mg_conc'
            mg_conc = vari(i);
        case 'lambda_e'
            lambda_e = vari(i);
        case 'lambda_i'
            lambda_i = vari(i);
        case 'sc'
            sc = vari(i);
        otherwise
            error('Unknown variable')
    end

    [Qe_root, Qi_root, Ve_root] = nmda_ss_finder...
        (mg_conc, lambda_e, lambda_i, sc, 0, Nsearch);

    lgthadd = length(Qe_root);

    if lgthadd == 0;
        Ve_root = NaN;
        Qe_root = NaN;
        Qi_root = NaN;
        lgthadd = 1;
    end

    % concatenates Ve_root(:) onto v vertically
    V_ss (lengthcounter+(1:lgthadd)) = Ve_root(:);
    Q_e_ss(lengthcounter+(1:lgthadd)) = Qe_root(:);
    Q_i_ss(lengthcounter+(1:lgthadd)) = Qi_root(:);
    Var_ss(lengthcounter+(1:lgthadd)) = vari(i);

    if graphflag

        v_max(i) = max(Ve_root);
        v_min(i) = min(Ve_root);
        q_e_max(i) = max(Qe_root);
        q_e_min(i) = min(Qe_root);
        q_i_max(i) = max(Qi_root);
        q_i_min(i) = min(Qi_root);

    end
end
```

```

    lengthcounter = lengthcounter + lgthadd;

end

lengthcounter = lengthcounter + 1;
% remove overallocated space
V_ss (lengthcounter:end) = [];
Q_e_ss(lengthcounter:end) = [];
Q_i_ss(lengthcounter:end) = [];
Var_ss(lengthcounter:end) = [];

% sort into V order
[V_ss, ind] = sort(V_ss);
Var_ss = Var_ss(ind);
Q_e_ss = sort(Q_e_ss);
Q_i_ss = sort(Q_i_ss);

%-----
if graphflag

figure(2003)
subplot(121)
    plot(Var_ss, V_ss,'k', 'linewidth', 3);hold on;
    hold off; grid on; zoom on;
    xlabel(variablename, 'fontsize', 18);
    ylabel('V_{e,i} [mV]', 'fontsize', 22);
    title(['(a) Stationary States (V_{e,i} vs ' variablename ')'],...
        'fontsize', 22);
    ax = axis; axis([startval, endval,ax([3 4])])
subplot(122)
    plot(Var_ss, Q_e_ss,'color',[0.7 0.7 0.7], 'linewidth', 3);hold on;
    plot(Var_ss, Q_i_ss,'k', 'linewidth', 3);
    xlabel(variablename, 'fontsize', 18);
    ylabel('Q_{e},Q_{i} [s^{-1}]', 'fontsize', 22);
    title(['(b) Stationary States (Q_{e,i} vs ' variablename ')'],...
        'fontsize', 22);
    legend('Q_{e}','Q_{i}');
    ax = axis; axis([startval, endval,ax([3 4])])
    hold off; grid on; zoom on;

```

```
set(gcf,'Color',[1,1,1]);
end
%-----
```

## A.6 Eigenvalues of the cortical model

```
function [Lm, Var_ss] = eigenvalues...
    (params, varname, qwavenum, n, startval, endval, graphflag2)
%function [Lm, Var_ss] = eigenvalues...
% (params, varname, qwavenum, n, startval, endval, graphflag2)
%
% Computes the eigenvalues of steadystates specified by the PARAMS input.
% The steadystates are found for a range of values of the VARNAME variable
% VARNAME is one of ['mg_conc','lambda_e','lambda_i','sc','qwavenum'].
% N - number of eigenvalues to compute within the range STARTVAL to ENDVAL.
% GRAPHFLAG2 - 0 or 1 to graph or not. Passed to variable_ss_finder
% function.
%
% Outputs
% Lm - an array of eigenvalues [8 x n]
% Var_ss - corresponding series of VARNAME variable.
%
% Eigenvalues can be used to determine the stability of the steady states.
% Computed from jacobian matrix partial derivative elements
%
% Created: Sept 2007

if nargin < 4
    n = 2001;
    if nargin < 3
        qwavenum = 0;
        if nargin < 2
            varname = 'lambda_i';
            if nargin < 1
                params = [0.78, 9.00, 1.0, 0.25];
            end
        end
    end
end
end
```

```

[mg_conc, lambda_e, lambda_i, sc] = ...
    deal(params(1),params(2),params(3),params(4));

W = nmda_init_globs;

%-----
if nargin < 7
    graphflag2 = 0;
    if nargin <= 5;
        if strcmp(varname,'mg_conc')
            startval = 0.00;
            endval   = 1.00;
        elseif strcmp(varname,'lambda_e')
            startval = 8.00;
            endval   = 11.00;
        elseif strcmp(varname, 'lambda_i')
            startval = 0.65;
            endval   = 1.20;
        elseif strcmp(varname, 'sc')
            startval = 0.00;
            endval   = .500;
        end
    end
end
end
%-----

switch varname
    case 'mg_conc'
        [v, mg_conc] = variable_ss_finder(varname, params, n,...
            startval, endval, graphflag2);
        n = length(mg_conc); variablename = 'Magnesium Concentration';
    case 'lambda_e'
        [v, lambda_e] = variable_ss_finder(varname, params, n,...
            startval, endval, graphflag2);
        n = length(lambda_e); variablename = '{\lambda} _{e}';
    case 'lambda_i'
        [v, lambda_i] = variable_ss_finder(varname, params, n,...
            startval, endval, graphflag2);
        n = length(lambda_i); variablename = '{\lambda} _{i}';
    case 'sc'
        [v, sc] = variable_ss_finder(varname, params, n,...
            startval, endval, graphflag2);

```

```

        n = length(sc);          variablename = 'Subcortical drive';
    case 'qwavenum'
        [Qe_root, Qi_root, v] = nmda_ss_finder(mg_conc, lambda_e,...
            lambda_i, sc, 0, 8000);
        qwavenum = linspace(0,0.2,n)*2*pi;
    end

%-----
% define some quantities to be used in calculating the Jacobian entries J##

v = v(:);

[k a] = deal(1/3.57, 0.062);

g_NMDA = JS_NMDA(v,mg_conc);

Qe = Qsige(v);
Qi = Qsigi(v);

weight_ee = Psi_ee(v);
weight_ei = Psi_ei(v);
weight_ie = Psi_ie(v);
weight_ii = Psi_ii(v);

Phi_e = W.Nee_ab .* Qe + sc .* W.phi_ee_sc;
Phi_i = W.Nie_b  .* Qi;

%-----
% List of entries in the 8x8 Jacobian begins here

J11 = (1/W.tau_e).* ...
        (-1 + (lambda_e*W.rho_e) .* (g_NMDA)      .* W.d_psi_ee  .* Phi_e ...
        + (lambda_i*W.rho_i)                      .* W.d_psi_ie  .* Phi_i ...
        + (lambda_e*W.rho_e) .* (g_NMDA).^2      .* Phi_e ...
        .* a .* k .* mg_conc .* exp(-a*v) .* weight_ee);

% J12 = 0;
J13 = (1/W.tau_e) .* (lambda_e.*W.rho_e) .* g_NMDA .* weight_ee;
% J14 = 0;
J15 = (1/W.tau_e) .* (lambda_i.*W.rho_i)          .* weight_ie;
% J16 = 0;

```

```

% J17 = 0;
% J18 = 0;

J21 = (1/W.tau_i) .* ...
      ((lambda_e*W.rho_e) .* (g_NMDA).^2 .* Phi_e ...
       .* a .* k .* mg_conc .* exp(-a*v) .* weight_ei );
J22 = (1/W.tau_i) .* ...
      (-1 + (lambda_e.*W.rho_e) .* g_NMDA .* Phi_e .* W.d_psi_ei ...
       + (lambda_i.*W.rho_i) .* Phi_i .* W.d_psi_ii);
J23 = (1/W.tau_i) .* (lambda_e.*W.rho_e) .* g_NMDA .* weight_ei;
% J24 = 0;
J25 = (1/W.tau_i) .* (lambda_i.*W.rho_i) .* weight_ii;
% J26 = 0;
% J27 = 0;
% J28 = 0;

% J31 = 0;
% J32 = 0;
% J33 = 0;
J34 = 1;
% J35 = 0;
% J36 = 0;
% J37 = 0;
% J38 = 0;

J41 = (W.gamma_e)^2 .* W.Nee_b .* d_Qsige(v);
% J42 = 0;
J43 = - (W.gamma_e)^2;
J44 = -2*W.gamma_e;
% J45 = 0;
% J46 = 0;
J47 = (W.gamma_e)^2 .* W.Nee_a;
% J48 = 0;

% J51 = 0;
% J52 = 0;
% J53 = 0;
% J54 = 0;
% J55 = 0;
J56 = 1;
% J57 = 0;
% J58 = 0;

```

```

% J61 = 0;
J62 = (W.gamma_i)^2 .* W.Nii_b .* d_Qsigi(v);
% J63 = 0;
% J64 = 0;
J65 = -(W.gamma_i)^2;
J66 = -2*W.gamma_i;
% J67 = 0;
% J68 = 0;

% J71 = 0;
% J72 = 0;
% J73 = 0;
% J74 = 0;
% J75 = 0;
% J76 = 0;
% J77 = 0;
J78 = 1;

J81 = W.v^2 .* W.Lambda^2 .*d_Qsige(v);
% J82 = 0;
% J83 = 0;
% J84 = 0;
% J85 = 0;
% J86 = 0;
J87 = W.v^2 .*(-qwavenum.^2 - W.Lambda^2);
J88 = -2 .* W.v .* W.Lambda;

% J = [J11 J12 J13 J14 J15 J16 J17 J18;...
%      J21 J22 J23 J24 J25 J26 J27 J28;...
%      J31 J32 J33 J34 J35 J36 J37 J38;...
%      J41 J42 J43 J44 J45 J46 J47 J48;...
%      J51 J52 J53 J54 J55 J56 J57 J58;...
%      J61 J62 J63 J64 J65 J66 J67 J68;...
%      J71 J72 J73 J74 J75 J76 J77 J78;...
%      J81 J82 J83 J84 J85 J86 J87 J88];

% end of Jacobian entries list
%-----

% assign values from a specific steadystate to the Jacobian J and calculate

```

```
% the eigenvalues at this state. Repeat for all values in the range.
```

```
J = zeros(8,8); % the Jacobian, most of it is zero
```

```
% nonzero constants - these values in the Jacobian do not change
```

```
J(3,4) = J34;
```

```
J(4,3) = J43;
```

```
J(4,4) = J44;
```

```
J(4,7) = J47;
```

```
J(5,6) = J56;
```

```
J(6,5) = J65;
```

```
J(6,6) = J66;
```

```
J(7,8) = J78;
```

```
J(8,8) = J88;
```

```
switch varname
```

```
%-----
```

```
case 'qwavenum'
```

```
    for branch = 1:length(v)
```

```
        Lm = zeros(8,n);
```

```
        J(1,1) = J11(branch);
```

```
        J(1,3) = J13(branch);
```

```
        J(1,5) = J15(branch);
```

```
        J(2,1) = J21(branch);
```

```
        J(2,2) = J22(branch);
```

```
        J(2,3) = J23(branch);
```

```
        J(2,5) = J25(branch);
```

```
        J(4,1) = J41(branch);
```

```
        J(6,2) = J62(branch);
```

```
        J(8,1) = J81(branch);
```

```
    for p = 1:n % the entry containing qwavenum will change.
```

```
        J(8,7) = J87(p);
```

```
%-----
```

```
% if there are infinities or Non-numbers in the Jacobian
```

```
% then don't calculate eigenvalues
```

```
        infnan = sum(J(:));
```

```

        if isnan(infnan)||isinf(infnan)
            Lm(:,p) = NaN;
        else
            Lm(:,p) = eig(J);
        end
    %-----

end
%-----
% dominant (real) eigenvalue
LM_real_dominant = max(real(Lm));

figure(branch+300)

Var_ss = qwavenum/(2*pi);
if branch == 1, br = 'Bottom Branch';
elseif branch == 2, br = 'Middle Branch';
elseif branch == 3, br = 'Top Branch';
end

plot(Var_ss,LM_real_dominant,'b-', 'linewidth',2); %hold on;
%plot(qwavenum/(2*pi),real(Lm)); hold off
xlabel('q/2{\pi}', 'fontsize',20)
ylabel('dominant (real part of) eigenvalue', 'fontsize',20)
title(['real eigs vs qwavenumber, ' br'],'fontsize',20)
grid on
zoom on
legend('dominant eigenvalue')

% figure
% plot(qwavenum/(2*pi),imag(Lm));
% title('imag eigs')
%-----

end

otherwise % ie mg_conc or lambda_e/i or sc

Lm = zeros(8,n);

```

```

J(8,7) = J87; % Jacobian entry with qwavenum constant in this case

for p = 1:n % the entries depending parameter being changed

    J(1,1) = J11(p);
    J(1,3) = J13(p);
    J(1,5) = J15(p);
    J(2,1) = J21(p);
    J(2,2) = J22(p);
    J(2,3) = J23(p);
    J(2,5) = J25(p);
    J(4,1) = J41(p);
    J(6,2) = J62(p);
    J(8,1) = J81(p);

    %-----
    % if there are infinities or Non-numbers in the Jacobian
    % then don't calculate eigenvalues
    infnan = sum(J(:));

    if isnan(infnan)||isinf(infnan)
        Lm(:,p) = NaN;
    else
        Lm(:,p) = eig(J);
    end
    %-----

end

%-----
% rename the variable being altered into the generic name Var_ss
if strcmp(varname,'mg_conc')
    Var_ss = mg_conc;
elseif strcmp(varname,'lambda_e')
    Var_ss = lambda_e;
elseif strcmp(varname,'lambda_i')
    Var_ss = lambda_i;
elseif strcmp(varname,'sc')
    Var_ss = sc;
end

%-----

```

```

color = [1 0 0; 0 1 0; 0 0 1; 0 1 1; 1 0 1; 0 0.5 0.5; 1 0.5 0; 0 0 0];

%-----
figure(202);
subplot(211)
plot(Var_ss, max(real(Lm)), 'b-', 'linewidth', 2); hold on
legend('Dominant eigenvalue','location','nw')
for i = 1:8
    plot(Var_ss, real(Lm(i,:).'), 'color', color(i,:));
end
grid on; zoom on
ax1 = axis; axis tight; ax2 = axis; axis([ax2([1 2]) ax1([3 4])])
xlabel(variablename, 'fontsize',20);
ylabel('Eigenvalue real part', 'fontsize',20)
title(['(a) Real Part Eigenvalues vs ' variablename], 'fontsize',20)
hold off

subplot(212)
plot(Var_ss, max(imag(Lm)), 'b-', 'linewidth', 2); hold on
legend('Dominant eigenvalue','location','nw')
for i = 1:8
    plot(Var_ss, imag(Lm(i,:).'), 'color', color(i,:));
end
hold off; grid on;
ax1 = axis; axis tight; ax2 = axis; axis([ax2([1 2]) ax1([3 4])])
xlabel(variablename, 'fontsize',20);
ylabel('Eigenvalue imaginary part', 'fontsize',20)
title(['(b) Imag Part Eigenvalues vs ' variablename], 'fontsize',20)

%-----

end

%-----
%-----

function g_NMDA = JS_NMDA(V,C)
% Compute normalized conductance for NMDA receptor
% using Jahr & Stevens (1990)
% Inputs: V = excitatory voltage (mV), C = Mg concentration (mMol)

```

```

[k a] = deal(1/3.57, 0.062); % 1/mM, 1/mV

g_NMDA = 1 ./ (1 + k.*C.*exp(-a.*V));

return

%-----
function Qe = Qsige(V)
% Excitatory sigmoid function for Wright model; input V in millivolts
% Previous Liley defn:
% sig = Se_max./(1 + exp(-ge*(h - theta_e)) );

global W

gain = pi/(sqrt(3) * W.sigma_e);
temp = 1./(1 + exp(-gain*(V - W.theta_e)));
Qe = W.Qe_max * temp;
return

%-----
function Qi = Qsigi(V)
% Inhibitory sigmoid function for Wright model
% Previous Liley defn:
% sig = Si_max./(1 + exp(-gi*(h - theta_i)) );

global W

gain = pi/(sqrt(3) * W.sigma_i);
temp = 1./(1 + exp(-gain*(V - W.theta_i)));
Qi = W.Qi_max * temp;
return

%-----
function d_Qe = d_Qsige(V)
% d/dV derivative of excitatory sigmoid function for Wright model;
% input V in millivolts

global W

gain = pi/(sqrt(3) * W.sigma_e);

```

```

E = exp(-gain*(V - W.theta_e));
d_Qe = gain * W.Qe_max * E ./ (1 + E).^2;
return

%-----
function d_Qi = d_Qsigi(V)
% d/dV derivative of inhibitory sigmoid function for Wright model

global W

gain = pi/(sqrt(3) * W.sigma_i);
E = exp(-gain*(V - W.theta_i));
d_Qi = gain * W.Qi_max * E ./ (1 + E).^2;
return

%-----
function weight = Psi_ee(V)
% Wright form for weighting function, but note use of i2j notation

global W
weight = (W.Ve_rev - V)/(W.Ve_rev - W.Ve_rest);
return

%-----
function weight = Psi_ei(V)

global W
weight = (W.Ve_rev - V)/(W.Ve_rev - W.Vi_rest);
return

%-----
function weight = Psi_ie(V)

global W
weight = (W.Vi_rev - V)/(W.Vi_rev - W.Ve_rest);
return

%-----
function weight = Psi_ii(V)

```

```

global W
weight = (W.Vi_rev - V)/(W.Vi_rev - W.Vi_rest);
return

```

```

%-----

```

## A.7 System stability and Hopf

```

%HOPF.m
% Checks the eigenvalues returned by EIGENVALUES.m for hopf bifurcation
% points and plots a graph of the dominant eigenvalue showing equilibrium
% state stability.
% Also requires the VARIABLE_SS_FINDER function to plot the equilibrium
% voltage S-bend.
% Created: 24-Jan-2008

% VARNAME: choose one of ['mg_conc', 'lambda_e', 'lambda_i', 'sc']
varname = 'lambda_i';

% PARAMS: fix values for the other variables
% value assigned to the VARNAME variable will be redundant
% params = ['mg_conc', 'lambda_e', 'lambda_i', 'sc']
params = [0.78, 9.0, 1.0, 0.25];

% STARTVAL,ENDVAL: initial and end value of the VARNAME variable
switch varname
    case 'mg_conc', startval = 0.6; endval = 1.0; % for mg_conc
    case 'lambda_e',startval = 8.0; endval = 12; % for lambda_e
    case 'lambda_i',startval = 0.7; endval = 1.3; % for lambda_i
    case 'sc',      startval = -5; endval = 5; % for sc
end

% Make changes above
%-----
qwavenum = 0;
npoints = 3000;
graphflag = 0; % passed to variable_ss_finder

%% computation of equilibrium voltages and eigenvalues
[Vss] = variable_ss_finder...
    (varname, params, npoints,startval,endval,0,8000);

```

```

[Lm,Var_ss] = eigenvalues...
    (params, varname, qwavenum, npoints, startval, endval, graphflag);

%% Code to Classify system stability

[a b] = max(real(Lm));
DomEig = Lm(b+(1:length(Lm))*8-8);
rDomEig = real(DomEig);
iDomEig = abs(imag(DomEig)); % abs because +ve/-ve imag parts
                                % occur in conjugate pairs

stability = zeros(1,length(Var_ss));
stability((rDomEig > 0)&(iDomEig == 0)) = 1; % Unstable non-oscillatory.
stability((rDomEig > 0)&(iDomEig ~= 0)) = 2; % Unstable oscillatory.
stability((rDomEig < 0)&(iDomEig ~= 0)) = 3; % Stable oscillatory.
stability((rDomEig < 0)&(iDomEig == 0)) = 4; % Stable non-oscillatory

%% Approximate hopf values by linear interpolation
x = Var_ss(:);
V = Vss(:);
yreal = rDomEig(:);
yimag = iDomEig(:);

idx = stability == 2;
x0 = find((diff(stability) == -1)&(idx(2:end))); % index
x1 = find((diff(stability) == 1)&(idx(1:end-1))); % index

Vhopf = V([x0;x1]) + (V([x0;x1]+1)-V([x0;x1])).*...
    (xhopf - x([x0;x1]))./(x([x0;x1]+1)-x([x0;x1]));

xhopf = x([x0;x1]) - yreal([x0;x1]).*...
    (x([x0;x1]+1)-x([x0;x1]))./...
    (yreal([x0;x1]+1)-yreal([x0;x1]));

yhopfimag = yimag([x0;x1]) + (yimag([x0;x1]+1)-yimag([x0;x1])).*...
    (xhopf - x([x0;x1]))./(x([x0;x1]+1)-x([x0;x1]));

lg = [false(1,4), any(xhopf)];
%% Plotting

```

```

figure(67)
% run once for each of the four stability types: unstable non-oscillatory,
% unstable oscillatory, stable oscillatory, stable non-oscillatory.
% The stable non-oscillatory is not found in the examined cases.
for k = 1:4
    x = Var_ss(:);
    yreal = rDomEig(:);
    yimag = iDomEig(:);
    V = Vss(:);
    x((x == startval)|(x == endval)) = NaN;
    % NaN to break spurious connections at endvalues
    idx = (stability == k);

    if k == 1, % Unstable non osc.
        idx = idx + [0, idx(1:end-1)] + [idx(2:end), 0];
        color = [0 0 0]; linestyle = '--';
    elseif k == 2, % Unstable osc.
        idx([x0; x1+1]) = 1;
        x([x0; x1+1]) = xhopf; V([x0; x1+1]) = Vhopf;
        yreal([x0; x1+1]) = 0; yimag([x0; x1+1]) = yhopfimag;
        color = 0.7*[1 1 1]; linestyle = '--';
    elseif k == 3, % Stable osc.
        idx([x0+1; x1]) = 1;
        x([x0+1; x1]) = xhopf; V([x0+1; x1]) = Vhopf;
        yreal([x0+1; x1]) = 0; yimag([x0+1; x1]) = yhopfimag;
        color = [0 0 0]; linestyle = '-';
    else % Stable non osc.
        color = 0.7*[1 1 1]; linestyle = '-';
    end

    if any(idx)
        yreal(idx == 0) = NaN;
        yimag(idx == 0) = NaN;
        V(idx == 0) = NaN;

        subplot(311)
        plot(x,V,'color',color,...
            'linestyle',linestyle,'linewidth',2); hold on;

        subplot(312)
        plot(x,yreal,'color',color,...
            'linestyle',linestyle,'linewidth',2); hold on;

```

```

        subplot(313)
        plot(x,yimag*[-1 1], 'color', color, ...
            'linestyle', linestyle, 'linewidth', 2); hold on;

        lg(k) = true; % for graph legend
    end

end

if any(xhopf),
    subplot(311), hold on
    plot(xhopf,Vhopf, 'ok', 'markersize', 12, 'linewidth', 1.5)
    plot(xhopf,Vhopf, '.k', 'markersize', 12, 'linewidth', 1)

    subplot(312), hold on
    plot(xhopf,zeros(size(xhopf)), 'ok', 'markersize', 12, 'linewidth', 1.5)
    plot(xhopf,zeros(size(xhopf)), '.k', 'markersize', 12, 'linewidth', 1)

    subplot(313), hold on
    plot(xhopf,yhopfimag*[-1 1], 'ok', 'markersize', 12, 'linewidth', 1.5)
    plot(xhopf,yhopfimag*[-1 1], '.k', 'markersize', 12, 'linewidth', 1)
end
hold off

%% Graph Labels

switch varname
    case 'lambda_i', titlevar = '\lambda_{i}';
    case 'lambda_e', titlevar = '\lambda_{e}';
    case 'mg_conc', titlevar = 'Magnesium Concentration, C [mM]';
    case 'sc', titlevar = 'Subcortical driving, s';
end

subplot(311)
ax = axis; axis([startval, endval, ax([3 4])])
lgnd = {'Unstable non oscillatory'; 'Unstable oscillatory'; ...
        'Stable oscillatory'; 'Stable non oscillatory'; ...
        'Hopf bifurcation'};
legend(lgnd(lg)); hold off
ylabel('V_{e,i} [mV]', 'fontsize', 16);
title(['(a) Stationary States (V_{e,i} vs ' titlevar ')'], ...

```

```

    'fontsize', 18);

subplot(312)
ax = axis; axis([startval, endval,ax([3 4])])
title('(b) Dominant eigenvalue', 'fontsize',18);
ylabel('Real part','fontsize',16); hold off

subplot(313)
ax = axis; axis([startval, endval,ax([3 4])])
ylabel('Imag part','fontsize',16)
xlabel(titlevar,'fontsize',18); hold off

%% End of code

```

## A.8 Euler numerical simulation of the cortical model

```

function Euler_simulation
% Euler Simulation

% The basic principle:
% we are given a differential equation governing the system  $Y(t) = dy/dt$ 
% we wish to solve for  $y(t)$  using the euler method
% knowing a given starting point  $y_0 = y(t = t_0)$ , the state of the system at
%  $t = t_0 + dt$  where  $dt$  is a finite time step is given by
%  $y(t_0 + dt) \sim y(t_0) + Y(t_0)*dt$ 
% This is done recursively further using the approximations obtained in the
% earlier step as new starting points.

global W

for i = 1:3 % Run several times because the simulation is stochastic.

W = nmda_init_globs;

% Mg_conc, lambda_e & i, sc and q
% Arbitrary values for starting point - but based
% on clinical experimental values
Mg_conc = 0.78; % adjust these four values
lambda_e = 9.0; % they will decide the ss V value(s)

% chosen values of lambda_i for the mid-branch [V(2)]:

```

```

lambda_i = 1;
%lambda_i = 0.9416;    % hopf
%lambda_i = 0.8817;

sc = 4;
q = 0;
which_root = 1; % set to one of 1,2 or 3
                % where there are 3 steady state branches, to pick the
                % right starting point: 1-bottom, 2-mid and 3-top
                % otherwise which_root doesn't matter.

% Using the above values, find the steady state (S.S.) voltages, the ...
% function may return up to 3 steady states, only one of these is chosen.
[Qe_root, Qi_root, V_e, V_i] = nmda_ss_finder...
                                (Mg_conc, lambda_e, lambda_i, sc, 0, 8000);

% Continue with determining the initial state of the other system variables
% using the clinical values and the chosen S.S. voltage set by them.
% Expressions for the initial values are derived from S.S. eqns dY/dt = 0.
% S.S. implies time independence. Therefore initial values are determined
% solely by mg, lambda,... , and thereby V

% initial values
V_e    = V_e(rem(which_root-1,length(V_e))+1);
V_i    = V_i(rem(which_root-1,length(V_i))+1);
Me     = 0;
Phi_e  = W.Nee_ab*Qsige(V_e) + sc*W.phi_ee_sc;
Mi     = 0;
Phi_i  = W.Nii_b*Qsigi(V_i);
omega_e = 0;
phi_e_a = Qsige(V_e);

y_0 = [V_e; V_i; Phi_e; Me; Phi_i; Mi; phi_e_a; omega_e];

nend = 8000;           % number of simulation steps
y = zeros(8, nend+1); % (8 x length of number of steps of simulation)
y(:,1) = y_0;         % initial values evaluated earlier
dt = 0.001;          % step size, [s]

Noise_amplitude = 0.01 ;

for n = 1:nend

```

```

%% This is for when noise is turned off some way into the simulation
%   time = n*dt;
%   if time > 3
%       Noise_amplitude = 0;
%   end

g_NMDA = JS_NMDA(V_e, Mg_conc);

[V_e, V_i, Phi_e, Me, Phi_i, Mi, phi_e_a, omega_e] = ...
    deal(y_0(1),y_0(2),y_0(3),y_0(4),y_0(5),y_0(6),y_0(7),y_0(8) );

dVedt = (...
    W.Ve_rest - V_e ...
    + lambda_e*g_NMDA * W.rho_e * Psi_ee(V_e) * Phi_e ...
    + lambda_i * W.rho_i * Psi_ie(V_e) * Phi_i) / W.tau_e;

dVidt = (...
    W.Vi_rest - V_i ...
    + lambda_e * g_NMDA * W.rho_e * Psi_ei(V_i) * Phi_e ...
    + lambda_i * W.rho_i * Psi_ii(V_i) * Phi_i) / W.tau_i;

dPhi_edt = Me;

dMedt = -2*W.gamma_e * Me + (W.gamma_e)^2 * ...
    (- Phi_e + W.Nee_a * phi_e_a + W.Nee_b * Qsige(V_e) ...
    + sc*W.phi_ee_sc ...
    + sqrt(sc*W.phi_ee_sc)*Noise_amplitude*randn/sqrt(dt));

dPhi_idt = Mi;

dMidt = -2*W.gamma_i * Mi + (W.gamma_i)^2 * ...
    (-Phi_i + W.Nii_b.*Qsigi(V_i));

dphi_e_adt = omega_e;

domega_edt = W.v*W.Lambda * ...
    (-2*omega_e + W.v*W.Lambda * (Qsige(V_e)- phi_e_a));

dydt = [dVedt; dVidt; dPhi_edt; dMedt; dPhi_idt; dMidt; dphi_e_adt;...
    domega_edt] ;

```

```

    % Euler step
    y_0 = y_0 + dydt*dt; % <<== This is the euler step!!
    % set the Euler simulated (n+1)th values in the appropriate position
    % in the preallocated time series
    y(:, n+1) = y_0;
end

figure(64); % plot of Ve-vs-t
time = (0:nend)*dt; % Simulation time from 0 to n_end*dt, [s]
Vsim = y(1,:);      % The 1st and 2nd rows contain the voltage values
plot(time(1:2:end),Vsim(1:2:end),'k-', 'linewidth', 2); hold on;
title('Euler Simulation of V_{e} vs time')
xlabel('time [s]')
ylabel('V_{e} [mV]')

% % phase plot of Phi_i(t)-vs-Phi_e(t)
% figure(65);
% plot(y(5,1:2:end),y(3,1:2:end), 'k-' )
% title('Phase plot of Euler Simulation ')
% xlabel('\Phi_e(t) [s^{-1}]')
% ylabel('\Phi_i(t) [s^{-1}]')

end

%-----
function g_NMDA = JS_NMDA(V,C)
% Compute normalized conductance for NMDA receptor
% using Jahr & Stevens (1990)
% Inputs: V = excitatory voltage (mV), C = Mg concentration (mMol)

[k a] = deal(1/3.57, 0.062); % 1/mM, 1/mV

g_NMDA = 1 ./ (1 + k*C*exp(-a*V));
return

%-----
function Qe = Qsige(V)
% Excitatory sigmoid function for Wright model; input V in millivolts
% Previous Liley defn:
%      sig = Se_max./(1 + exp(-ge*(h - theta_e)) );

global W

```

```

gain = pi/(sqrt(3) * W.sigma_e);
temp = 1./(1 + exp(-gain*(V - W.theta_e)));
Qe    = W.Qe_max * temp;
return

%-----
function Qi = Qsigi(V)
% Inhibitory sigmoid function for Wright model
% Previous Liley defn:
%     sig = Si_max./(1 + exp(-gi*(h - theta_i)) );

global W

gain = pi/(sqrt(3) * W.sigma_i);
temp = 1./(1 + exp(-gain*(V - W.theta_i)));
Qi    = W.Qi_max * temp;
return

%-----
function weight = Psi_ee(V)
% Wright form for weighting function, but note use of i2j notation

global W
weight = (W.Ve_rev - V)/(W.Ve_rev - W.Ve_rest);
return

%-----
function weight = Psi_ei(V)

global W
weight = (W.Ve_rev - V)/(W.Ve_rev - W.Vi_rest);
return

%-----
function weight = Psi_ie(V)

global W
weight = (W.Vi_rev - V)/(W.Vi_rev - W.Ve_rest);
return

%-----

```

```
function weight = Psi_ii(V)
```

```
global W
```

```
weight = (W.Vi_rev - V)/(W.Vi_rev - W.Vi_rest);
```

```
return
```

```
%-----
```



# References

- Beurle, R. L. Properties of a mass of cells capable of regenerating pulses. *Phil. Transactions of the Royal Society London*, **240**, pp. 55–94 (1956).
- Boyce, W. E. and R. C. DiPrima. *Elementary Differential Equation and Boundary Value Problems*. John Wiley and Sons Inc, sixth edition (1997).
- Braitenberg, V. and A. Schüz. *Anatomy of a Cortex: Statistics and Geometry*. Springer-Verlag (1991).
- Flohr, H., U. Glade, and D. Motzko. The neural correlate of consciousness and the mechanisms of general anaesthesia. *Pabst Science Publishers*, pp. 12–24 (2000).
- Franks, N. P., R. Dickenson, S. L. M. de Sousa, A. C. Hall, and W. R. Lieb. How does xenon produce anaesthesia? *Nature*, **396**, p. 324 (1998).
- Franks, N. P. and W. R. Lieb. Anaesthetics set their sites on ion channels. *Nature*, **389**, p. 334335 (1997).
- Freeman, W. J. *Mass Action in the Nervous System*. Academic Press, New York (1975).
- Guedel, A. E. *Inhalational Anesthesia: A Fundamental Guide*. Macmillan (1937).
- Halsey, M. J. and E. B. Smith. Effects of anaesthetics on luminous bacteria. *Nature*, **227**, p. 13631365 (1970).
- Hodgkin, A. L. and A. F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology*, **117**, p. 500544 (1952).
- Jahr, C. E. and C. F. Stevens. Voltage dependence of nmda-activated macroscopic conductances predicted by sigle-channel kinetics. *The Journal of Neuroscience*, **10**, pp. 3178–3182 (1990).
- Jevtovic-Todorovic, V., S. M. Todorovic, S. Menneric, S. Powell, K. Dikranian, N. Benshoff, C. F. Zorumski, and J. W. Olney. Nitrous oxide (laughing gas) is an nmda antagonist, neuroprotectant and neurotoxin. *Nature Medicine*, **4**, p. 460464 (1998).

- Jirsa, V. K. and H. Haken. A field theory of electromagnetic brain activity. *Physical Review Letters*, **77**, p. 960963 (1996).
- Jirsa, V. K. and H. Haken. A derivation of a macroscopic field theory of the brain from the quasi-microscopic neural dynamics. *Physica D*, **99**, p. 503526 (1997).
- Kuizenga, K. *Pharmacokinetic pharmacodynamic modelling of EEG effects in hypnotic drugs used in anaesthesia*. Ph.D. thesis, Groningen University Hospital, The Netherlands (2001).
- Liley, D. T. J., P. J. Cadusch, and M. P. Dafilis. A spatially continuous mean field theory of electro-cortical activity. *Network: Computation in Neural Systems*, **13**, pp. 67–113 (2002).
- Liley, D. T. J., P. J. Cadusch, and J. J. Wright. A continuum theory of electro-cortical activity. *Neurocomputing*, **2627**, p. 795800 (1999).
- Nunez, P. L. The brain wave function: A model for the eeg. *Mathematical Biosciences*, **21**, p. 279297 (1974).
- Nunez, P. L. *Electric Fields of the Brain: The Neurophysics of EEG*. Oxford University Press, New York (1981).
- Reichl, L. E. *A Modern Course in Statistical Physics*. John Wiley and Sons Inc (1998).
- Robinson, P. A., C. J. Rennie, and J. J. Wright. Propagation and stability of waves of electrical activity in the cerebral cortex. *Physical Review E*, **56**, p. 826840 (1997).
- Robinson, P. A., C. J. Rennie, J. J. Wright, and P. D. Bourke. Steady states and global dynamics of electrical activity in the cerebral cortex. *Physical Review E*, **58**, p. 35573571 (1998).
- Steyn-Ross, D. A. *Modelling the Anaestheto-Dynamic Phase Transition of the Cerebral Cortex*. Ph.D. thesis, The University of Waikato (2002).
- Steyn-Ross, D. A., M. L. Steyn-Ross, J. W. Sleight, M. T. Wilson, I. P. Gillies, and J. J. Wright. The sleep cycle modelled as a cortical phase transition. *Journal of Biological Physics*, **31**, pp. 547–569 (2005).
- Steyn-Ross, M. L., D. A. Steyn-Ross, J. W. Sleight, and D. T. J. Liley. Theoretical electroencephalogram stationary spectrum for a white-noise-driven cortex: Evidence for a general-anaesthetic-induced phase transition. *Physical Review E*, **60**, pp. 7299–7311 (1999).
- Steyn-Ross, M. L., D. A. Steyn-Ross, M. T. Wilson, and J. W. Sleight. Gap junctions mediate large-scale turing structures in a mean-field cortex driven by subcortical noise. *Physical Review E*, **76**, p. 011916 (2007).

- Tortora, G. J. and S. R. Grabowski. *Principles of Anatomy and Physiology*. John Wiley and Sons Inc, ninth edition (2000).
- Tuckwell, H. C. *Introduction to Theoretical Neurobiology: Linear Cable Theory and Dendritic Structure*, volume 1. Cambridge University Press, Cambridge (1988).
- Waters, K. A. and R. Machaalani. Nmda receptors in the development brain and effects of noxious insults. *Neuro-Signals*, **13(4)**, pp. 162–174 (2004).
- Wilson, H. R. and J. D. Cowan. Excitatory and inhibitory interactions in localized populations of model neurons. *Biophysical Journal*, **12**, p. 124 (1972).
- Wilson, H. R. and J. D. Cowan. A mathematical theory of the functional dynamics of cortical and thalamic nervous tissue. *Kybernetik*, **13**, p. 5580 (1973).