Investigation of Neuronal Dynamics

Resting, Spiking and Bursting behaviour

Received 3 November 2008

Abstract

A stability analysis of conductance based equations of neurons using rate equations is preformed to develop phase diagrams that outline boundaries between neuronal behaviour types of resting, spiking and bursting. This was done using eigenvalue analysis of Jacobian stability matrixes of four conductance variables of the I_{Na}, I_K, I_T and I_{AHP} ion currents to examine the effect of the after-hyperpolarizing conductance and depolarizing transient-inward conductance in the modulation of bursting behaviour. For bursting behaviour these conductances are shown to determine the duration of the burst when a minimum external current is present in a neuron. The benefits of basic behavioural analysis through phase diagrams circumvents the need for individual neuron simulation in large scale simulations of neuronal networks involving many cross connections between neurons rendering these large simulations computationally feasible.

I. Introduction

Neuronal behaviour is based on the electrical interaction of various ion currents that can give rise to complex and wide ranging behaviours. Approximating these systems into simpler models of capacitors that produce ion flows, one can investigate the underlying dynamical principals and biophysics.

In 1952 Hodgkin and Huxley began investigation on the biophysics of action potentials of the squid giant axon, specifically the activation and inactivation of sodium and potassium channels. Since then 12 currents have been shown to contribute to spiking and bursting of neocortical neurons (Wilson 1999). Ion channels are the drivers behind changing voltage levels in neurons, and their dynamics are used in models such as Wilson's capacitance based model,

$$C\frac{dV}{dt} = I_{ext} - I_{Na} - I_K - I_{leak}\ . \tag{1}$$

Models of neural networks can be formed on such individual neuron models, where the output signal from one neuron can propagate via axon terminals to be an input signal to many thousands of other adjacent neurons (Kandel et al.). Large scale simulations have been conducted to investigate network behaviours (Izhikevich 2004), however this soon becomes computationally prohibitive with the addition of neurons that are highly connected and thus highly interdependent.

A key reason for this is that the neuron models based on ion-channel dynamics themselves comprise of non-linear equations, Wilson's 4D model for example (Wilson 1999) used four main ion channels of Na+, K+ and Ca2+ to model spiking and bursting behaviour. As such, explicit answers cannot be solved for this model, the only method to solving the voltage signals over time is from numerical simulations.

Much of the behaviour of neural networks can largely be discovered from the spiking and bursting rates of neurons and the voltage levels in which they do so. All other details gained from modelling of neurons can be discarded, and hence the computational task of simulating neural networks is not so prohibitive. This is one of the practical implications behind this research, to correlate basic neuronal behaviour with respect to some specific ion-channel parameters using stability analysis of the ion-channel current levels. For definitive relations to be found that link model parameters (including an externally applied current) to behaviours such as resting, bursting spiking of individual neurons, and the rates in which they do so, then these computationally-heavy simulations of neurons can largely be avoided in large scale neural network simulations.

Neurons show a few typical behaviour types. Resting behaviour describes a neuron that is not in an excited state but stable with an unchanging voltage. The neocortical neuron's voltage here is called the resting potential, typically -70mV. Bursting is the rapid succession of voltage spiking in neurons, a bistable process which is followed by a quiescent period of relative inaction. This interesting behaviour is the result of 12 distinct ion channels that, as Wilson (1999) suggests, fall into 2 fundamental categories. Axonal channels, such as Na+ currents (I_{Na}) and K+ currents (I_K), are directly responsible for the rapid spiking response of a neuron, and other channels in the soma and dendrites which can effectively modulate this fast spiking activity. Bursting modulation is achieved with the building up of slow currents that eventually hyperpolarize the neuron and ceases the repetitive spiking activity, which is followed by a longer 'quiescent' period of relative inaction (see Fig. 1) as the built-up charge leaks away, depolarizing the cell. Wilson's 4D model pays particular attention to the depolarizing transient-inward Ca2+ current (I_T) and an after-hyperpolarizing Ca2+ current (I_{AHP}) Robinson et al. (2007).

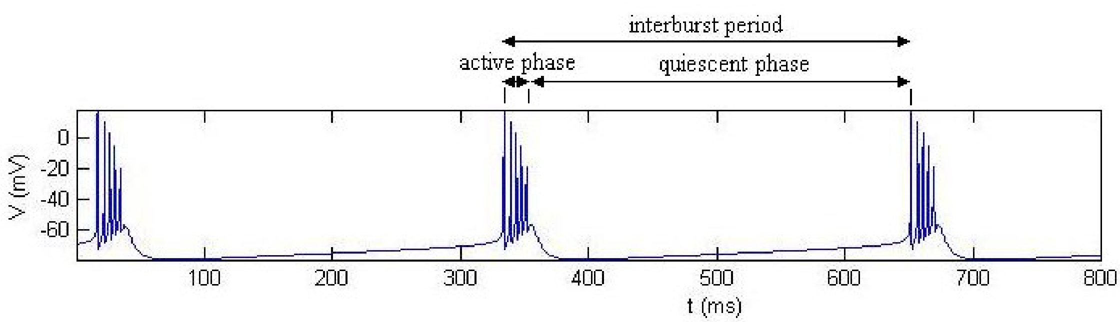

Figure 1 – Dynamics of Bursting shown over time for several bursts (using Wilson's 4D model where $\mathrm{g_H} = 130\mathrm{A/m^2V}, \mathrm{I_0} = 0.23\mathrm{nA}$)

<u>Significance of Bursting Phenomena</u>

Bursting plays a vital role in the communication and synchronisation between neurons. Bursting as a repeated spiking process is more reliable than single spikes in reducing the likelihood of synaptic transmission failure. If a synapse located at the axon terminal of one neuron repeatedly releases neurotransmitters in short but continuous bursts, directed by fast-acting ion-channels, the action potential of a postsynaptic target neuron is more likely to be triggered (Lisman 1997). This ultimately enables less-probabilistic signal propagation between neurons, and more deterministic outcomes essential for structured signal relay in the brain.

This behaviour also allows for selective communication. Postsynaptic targets often contain cells that behave differently depending how a potential voltage difference is placed over the membrane, i.e. the frequency in which a signal is applied to it. If a bursting signal, which has a frequency (determined by the inter-spike period, in turn determined by Na+ K+ channel biophysics), is close enough to a postsynaptic 'resonant' frequency, an action potential may occur that would not occur otherwise. Effectively many postsynaptic receptors are implemented with these band-pass filters, which reduces signal-to-noise ratios required for the transmission of a signal and is selective of what sort of bursting signal it receives (Izhikevich et al. 2003).

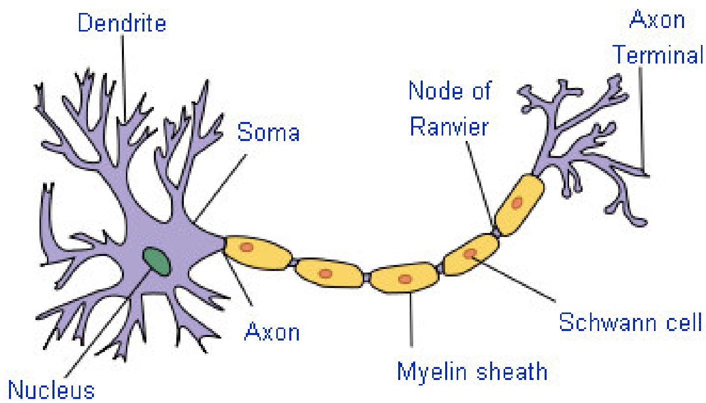

Figure 2 – Neocortical Neuron showing major features and areas of ion channels.
(http://en.wikipedia.org/wiki/Neurons)

II. Model

Wilson (1999) showed that a neocortical neuron can be modelled by just 4 separate currents, and still be able to reproduce all observed responses of neurons. This includes the fast acting I_{Na} and I_K currents and the modulating I_T and I_{AHP} currents discussed above which combine in the following relation for membrane potential:

$$C\frac{dV}{dt} = I_0 - I_{Na} - I_K - I_T - I_{AHP}\ . \tag{2}$$

Wilson's model assumes a neocortical neuron's lipid membrane as a capacitor 'C', of which each ion current can transfer charge from changing the internal soma voltage 'V'. Furthermore, each ionic current is represented as a modification of Ohm's law, where the conductance value is not linear but is comprised of a constant term 'g' and as nonlinear function of voltage called a *remembrance* value (Wilson 1999).

$$I_j = g_j(V - E_j). \tag{3}$$

Where g_j is conductance per unit area, E_j is equilibrium potential or *reversal* potential. Using this form from Hodgkin and Huxley (1952), the Wilson model breaks each ion current type as follows:

$$\begin{aligned} I_{Na} &= g_V(V)(V - V_1), & (4)\\ g_V(V) &= v_0 + v_1 V + v_2 V^2, & (5)\\ I_K &= g_R R(V - V_R), & (6)\\ I_T &= g_X X(V - V_X), & (7)\\ I_{AHP} &= g_H H(V - V_H). & (8) \end{aligned}$$

with $V_1 = 48\text{mV}$ the Na+ equilibrium potential, $V_R = -95\text{mV}$ the K+ equilibrium potential, $V_X = 140\text{mV}, V_H = -70\text{mV}$, $g_R = 260\text{Am}^{-2}$, $g_X = 20\text{Am}^{-2}$, $g_H = 130\text{Am}^{-2}$ and C = 0.010Fm^{-2} (Robinson et al. 2007). The Na+ conductance value $g_V(V)$ which changes with voltage is often referred by Wilson (1999) as the 'Na+ activation function'. Values v_0, v_1 and v_2 were that which best fitted Rinzel's (1985) approximation of Hodgkin & Huxley's (1952) model isoclines of $\frac{dV}{dt} = 0$, limited as a quadratic Taylor approximation such that the Wilson model is only third order polynomial and not more complex. Their respective values of $178.1\text{Am}^{-2}, 4758\text{Am}^{-2}$ and $3.38 \times 10^4\text{Am}^{-2}$ (Robinson et al. 2007) are such that $\mathrm{g_V}(V)$ is positive for all values V, and thus the polarity of $(V - V_1)$ always directs I_{Na} the same direction. However $\frac{dV}{dt} \propto -I_{Na} = -g_V(V)(V - V_1)$, where nominal spiking voltages are within the range $V_R < V < V_1$. Hence $\frac{dV}{dt} \propto= g_V(V)(V_1 - V)$ and the voltage would increase exponentially.

This is not the case however for neurons; Na+ ion currents that cause voltage changes will gate K+ channels. The K+ ion current's activation function is a constant conductance g_R multiplied by a dimensionless quantity R, the *remembrance* value, which has a time constant of τ_R as shown in:

$$\frac{dR}{dt} = -\frac{R - R_{inf}(V)}{\tau_R}, \tag{9}$$

$$R_{inf}(V) = r_0 + r_1 V + r_2 (V - V_2)^2. \tag{10}$$

with $V_2 = -38\text{mV}, r_0 = 0.79\text{Am}^{-2}, r_1 = 12.9\text{Am}^{-2}$ and $r_2 = 330\text{Am}^{-2}$ (Robinson et al. 2007). As I_{Na} causes the neuron voltage increases, the $R_{inf}(V)$ will increase due to its positive dependence on voltage, surpassing R and thus causing $\frac{dR}{dt}$ to take a positive value. Thus the remembrance R begins to increase which increases $[g_R R(V - V_R)]$ (NB: the g_R and $(V - V_R)$ values are both positive) which is equal to I_K. As I_K increases and surpasses the negative current I_{Na} which originally caused the voltage spike, the voltage rate $\frac{dV}{dt} \propto -I_{Na} - I_K$ is driven negative, and the spike subsides. This limit cycle between the Na+ and K+ ion currents is what gives rise to spiking behaviour.

As it is, using these two ion currents, neuron spiking can be modelled. However the extra terms I_T and I_{AHP} are needed to model bursting:

$$\frac{dX}{dt} = -\frac{X - X_{inf}(V)}{\tau_X}, \tag{11}$$

$$X_{inf}(V) = x_2 (V - V_3)(V - V_4), \tag{12}$$

$$\frac{dH}{dt} = -\frac{H - 3X}{\tau_H}. \tag{13}$$

with $x_2 = 900\mathrm{V}^{-2}$, $V_3 = -75.4\mathrm{mV}$ and $V_4 = -70\mathrm{mV}$ (Robinson et al. 2007). The time period of quiescence is much longer than the spiking duration in bursting neurons. As such the time constants responsible for depolarizing and after-hyperpolarizing the cell (which destroy the limit cycle behaviour) are much longer than the remembrance time constant:

$$\tau_R \ll \tau_X < \tau_H \tag{14}$$

with $\tau_R = 2.1\mathrm{ms}, \tau_X = 15\mathrm{ms}$ and $\tau_H = 56\mathrm{ms}$ (Robinson et al. 2007). The conductance relation X (the conductance of depolarizing current I_T) is analogous to R in that it models a conductance which decays exponentially over time. The evolution equation (Eq. 13) of conductance H of the after-hyperpolarizing current I_{AHP} however does not directly depend on the neuron voltage potential V but H and X only due to its biophysically dependent on Ca2+ ion concentration through the neuron membrane (Wilson 1999).

III. Stability Analysis

From Eqs (2) – (13), rate equations can be expressed as functions of conductance variables V, R, X and H,

$$\frac{dV}{dt} = f_1(V,R,X,H) = \frac{[I_0 - g_V(V)(V-V_1) - g_R R(V-V_R) - g_X X(V-V_X) - g_H H(V-V_H)]}{C}, \tag{15}$$

$$\frac{dR}{dt} = f_2(V,R,X,H) = -\frac{R - R_{inf}(V)}{\tau_R}, \tag{16}$$

$$\frac{dX}{dt} = f_3(V,R,X,H) = -\frac{X - X_{inf}(V)}{\tau_X}, \tag{17}$$

$$\frac{dH}{dt} = f_4(V,R,X,H) = -\frac{H - 3X}{\tau_H}. \tag{18}$$

Fixed points are obtained by setting the derivatives of the left hand side to zero, and Eqs. (16) - (18) become $\bar{R} = R_{inf}(V)$, $\bar{X} = X_{inf}(V)$ and $\bar{H} = 3X_{inf}(V)$. Thus, fixed points are roots of the following third order polynomial of V,

$$g_V(V)(V-V_1) + g_R\bar{R}(V-V_R) + g_X\bar{X}(V-V_X) + g_H\bar{H}(V-V_H) - I_0 = 0. \tag{19}$$

We now apply the linear stability analysis to determine the stability of the fixed point. For small deviations from the fixed point $(\bar{V},\bar{R},\bar{X},\bar{H})$,

$$\begin{aligned} V &= \bar{V} + v, \\ R &= \bar{R} + r, \\ X &= \bar{X} + x, \\ H &= \bar{H} + h. \end{aligned} \tag{20}$$

And we have a first order approximation of the deviation rate from the fixed point with

$$\begin{bmatrix} \dot{v} \\ \dot{r} \\ \dot{x} \\ \dot{h} \end{bmatrix} = J_{4\times4} \times \begin{bmatrix} v \\ r \\ x \\ h \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial V} & \frac{\partial f_1}{\partial R} & \frac{\partial f_1}{\partial X} & \frac{\partial f_1}{\partial H} \\ \frac{\partial f_2}{\partial V} & \frac{\partial f_2}{\partial R} & \frac{\partial f_2}{\partial X} & \frac{\partial f_2}{\partial H} \\ \frac{\partial f_3}{\partial V} & \frac{\partial f_3}{\partial R} & \frac{\partial f_3}{\partial X} & \frac{\partial f_3}{\partial H} \\ \frac{\partial f_4}{\partial V} & \frac{\partial f_4}{\partial R} & \frac{\partial f_4}{\partial X} & \frac{\partial f_4}{\partial H} \end{bmatrix} \times \begin{bmatrix} v \\ r \\ x \\ h \end{bmatrix}, \tag{21}$$

where $J_{4\times4}$ is the Jacobian matrix constructed of Wilson's conductance variables V, R, X and H. These 4 equations can be decoupled by using a change of coordinates to the eigenvectors of the Jacobian matrix, which (when solved for) have a magnitude term of $A_i e^{\lambda_i t}$, where A_i is a constant, λ_i is the respective eigenvalue and t is time. The properties of these eigenvalues are related to neuronal behaviour. Their sign and real/complex nature of the amplitude function $e^{\lambda_i t}$ determines how the variables v, r, x and h evolve over time. For example if all eigenvalues are negative, deviations v, r, x, h will approach zero as time progresses by the decreasing amplitude function $e^{\lambda_i t}$ of each eigenvector and this would be a stable system. More generally:

- (4) Negative Real λ_i : Stable System
- (1+) Positive Real λ_i : Unstable System (local max. or saddle node)
- (1+) Negative Imaginary λ_i : System has a rotational component between 2+ variables
- (1+) Positive Imaginary λ_i : System has a rotational component between 2+ variables in the other direction

(where the bracketed number indicates how many eigenvalues are required to be classified as such for that corresponding stability outcome to be true)

IV. Results

The bulk of the stability analysis for neuron voltage states was done using MATLAB. The eigenvalues are calculated about the fixed point for voltage as calculated as per Eq. (19). As Wilson limited this to a cubic equation, there can be up to 3 fixed points for the cell voltage, being the roots of that equation. However for most of the range of 'normal' values of g_H (around $130\mathrm{A/m^2V}$) and g_X (around $20\mathrm{A/m^2V}$) there is only one real root, the others being imaginary. Wilson's model was mapped for different values of g_H vs. I_0 and g_X vs. I_0 to yield phase diagrams (of which show where any of the 4 eigenvalues change state from positive to negative or real to complex) and surf-plots to indicate the amplitude of the real & imaginary eigenvalues. Parameters g_H and g_X were chosen for this project due to their significant and interesting effects on the after-hyperpolarizing current (I_{AHP}) and the depolarizing transient-inward current (I_T) which govern bursting behaviour. Other parameters related to these currents such as time constants τ_X and τ_H only changed the time periods of bursting and quiescence upon inspection and not pursued.

Behaviour in the g_H - I_0 plane

Using the normal values of g_H and g_X supplied in Wilson's model, the real-components of eigenvalues are shown in Fig. 3 (forth eigenvalue not shown as very negative and off scale) for ranging external current I_0.

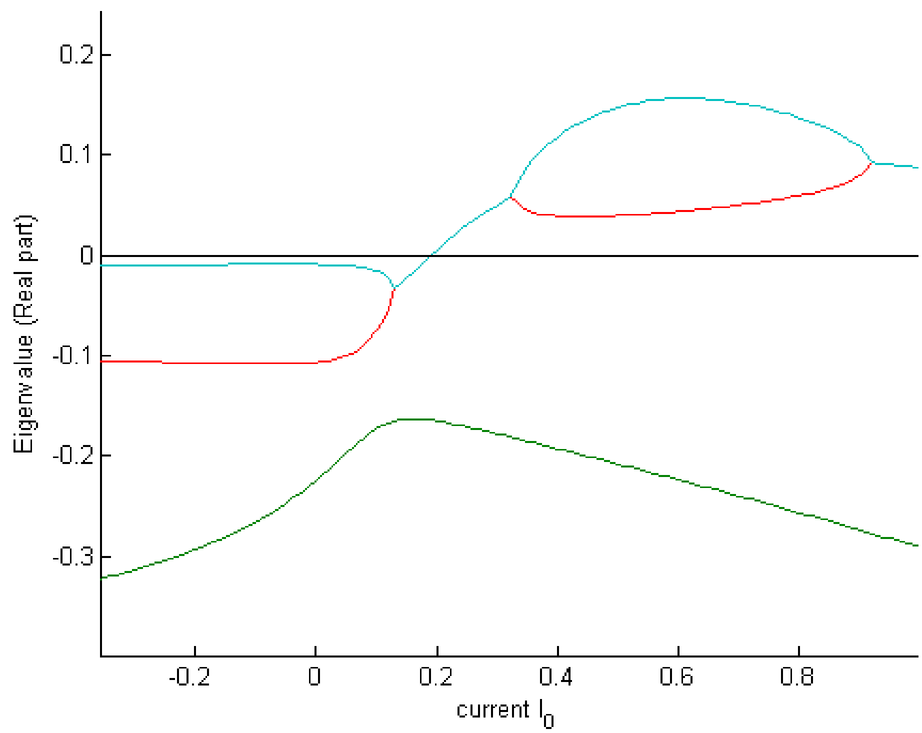

Figure 3 – Real-part Eigenvalues of Wilson's 4D model for $g_H = 130\mathrm{A/m^2V}$, $g_X = 20\mathrm{A/m^2V}$

The simulation results below show the effect the external input current I_0 has on the bursting behaviour. As I_0 increasing from below a threshold of $0.19nA$to above this threshold, the real-component's sign of the eigenvalues changes from being all negative, to 2 positive and 2 negative eigenvalues.

A simulation with $I_0 = 0.18\mathrm{nA}$ shown in Fig. 4 shows an oscillating yet stable soma voltage level in the neuron. However just a slight change of I_0 to $0.20\mathrm{nA}$ pushes the system into positive-eigenvalue territory, albeit very small positive eigenvalues, however as shown in Fig. 5, a sudden change in the neuron's behaviour occurs to well defined periodic bursting. This fine distinction between neuronal behaviours for very small eigenvalues shows the high dependence of their sign defining the stability of the four dimensional system.

Fig 6. Is a further simulation at $I_0 = 0.60\mathrm{nA}$, which where the complex conjugate pairs of eigenvalues that crossed into the positive zone at $I_0 = 0.19\mathrm{nA}$ and have separated into 2 real eigenvalues. It is clear from this simulation that it still agrees with eigenvalue signs of Fig. 3, and our stability criterion holds, yet there seems to be no big difference between Fig. 6 and Fig. 5, except for a change in the quiescent period between bursts. This shows that there is no immediately obvious change in behaviour of neurons if their eigenvalues contain an imaginary component or not (the addition of a 'rotational component' as discussed page 6).

Resting and Bursting simulations:

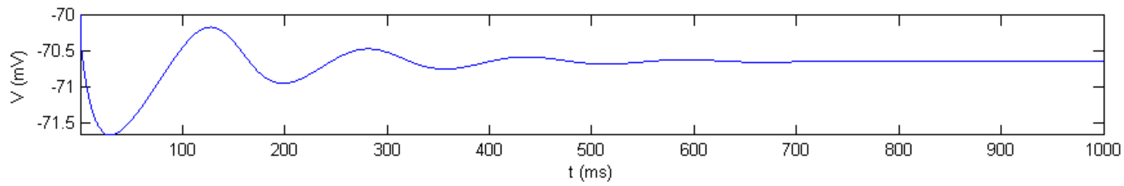

Figure 4 –Wilson's 4D model simulation of Resting where $g_H = 130\mathrm{A/m^2V}$, $I_0 = 0.18\mathrm{nA}$

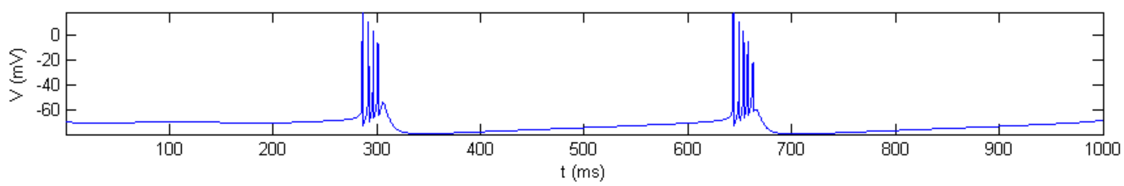

Figure 5 –Wilson's 4D model simulation of Bursting where $g_H = 130\mathrm{A/m^2V}$, $I_0 = 0.20\mathrm{nA}$

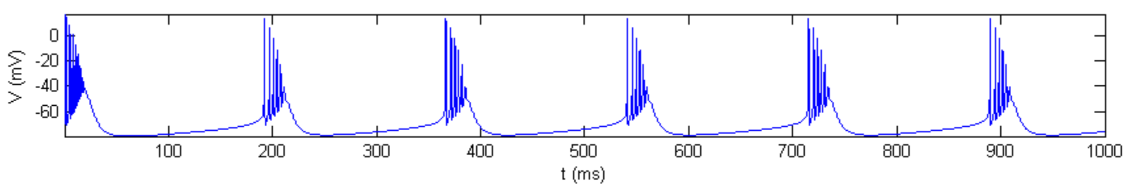

Figure 6 –Wilson's 4D model simulation of Bursting where $g_H = 130\mathrm{A/m^2V}$, $I_0 = 0.60\mathrm{nA}$

It is worth mentioning bifurcations of fixed points can appear for some values of g_H. Cross-sections of constant-g_H in Fig. 12, such as the $130\mathrm{A/m^2V}$ line has stable fixed points across the whole length of I_0. This is found using the sign of $\ddot{V}(\bar{V})$; if positive indicates an unstable fixed point, and stable if negative. By reducing g_H to $30\mathrm{A/m^2V}$, the contour shown in Fig. 12 is sliced into an S-shape with stable arms encompassing an unstable fixed-point arm. Note this does not lead to any limit cycles, the values $\mathrm{I_0}$ and $\mathrm{g_H}$ are assumed invariant in general neuron behaviour.

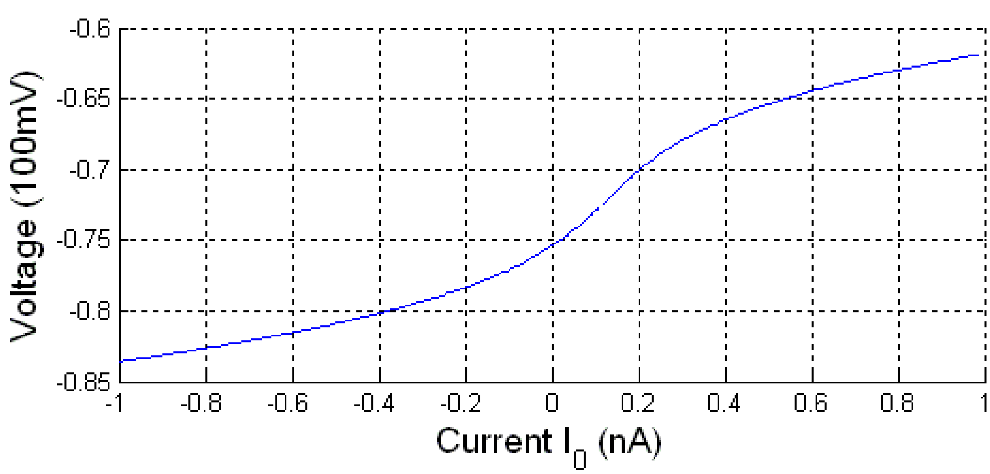

Figure 7 –Voltage Bifurcation of fixed point $\bar{V}$ vs. I_0 where $g_H = 130\mathrm{A/m^2V}$, $g_X = 20\mathrm{A/m^2V}$

As seen in the simulation of Fig. 8 below, normal bursting behaviour is observed at the $I_0 = 0.23\text{nA}$ point of the bifurcation diagram of Fig. 7.

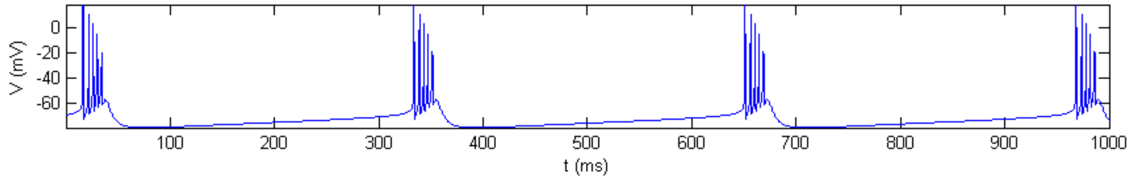

Figure 8 –Wilson's 4D model simulation of Bursting where $g_H = 130\text{A/m}^2\text{V}$, $I_0 = 0.23\text{nA}$

Fig. 9 shows an 'S-shape' bifurcation when g_H is low enough.

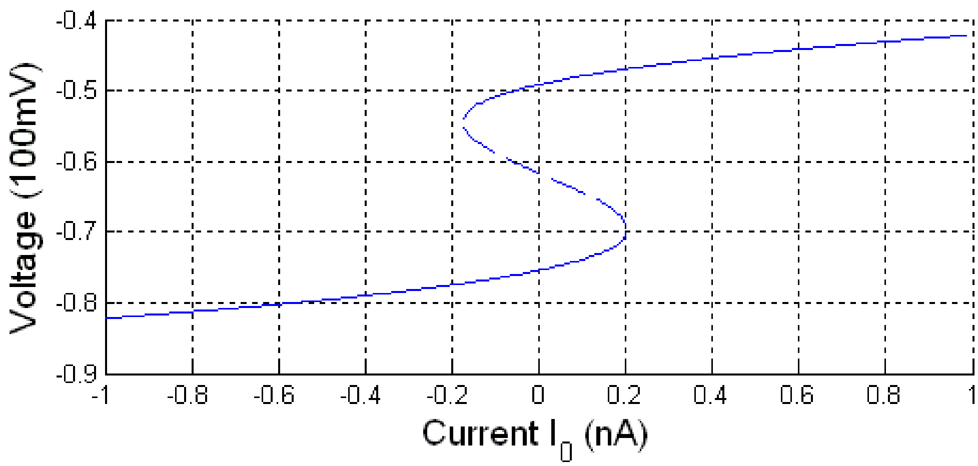

Figure 9 –Saddle Node Bifurcation of fixed point $\bar{V}$ vs. I_0 where $g_H = 30\text{A/m}^2\text{V}$, $g_X = 20\text{A/m}^2\text{V}$

Fig. 10 shows a simulation again at $I_0 = 0.23\text{nA}$, but for a different g_H value of $30\text{A/m}^2\text{V}$. Note however rehular bursting is observed as the 0.23nA vertical line intersects only one of the fixed points in Fig. 9.

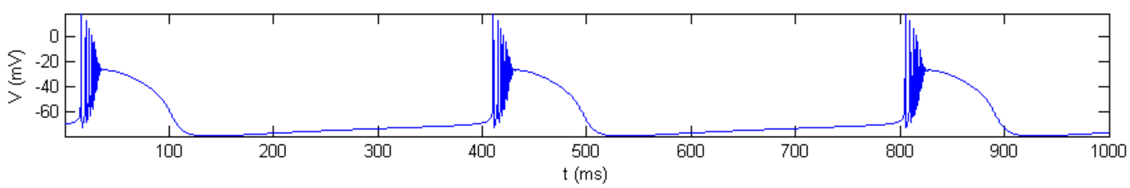

Figure 10 –Wilson's 4D model simulation of Bursting where $g_H = 30\text{A/m}^2\text{V}$, $I_0 = 0.23\text{nA}$

However in Fig. 11, this simulation again commenced at the resting potential -70mV, like the simulation of Fig. 10 did, but straight away moves away from the unstable S-branch (located at -62mV when $I_0 = 0.00nA$) of Fig. 9 and permanently sticks to the lower stable branch at roughly -75mV. This is shown later in Fig. 12 that areas of three fixed points within an S-shape bifurcation do not exhibit bursting behaviour, yet they can exhibit sporadic spiking.

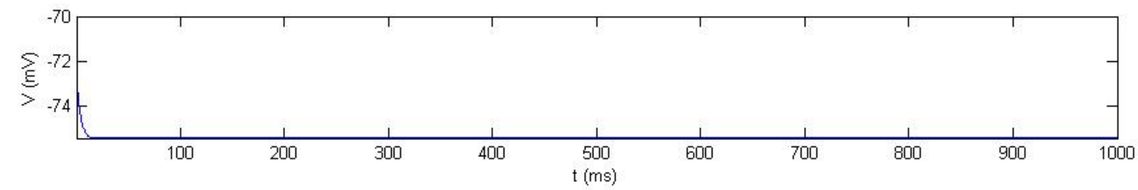

Figure 11 –Wilson's 4D model simulation of Bursting where $g_H = 30A/m^2V,\ \ I_0 = 0.00nA$

As discussed, the after-hyperpolarizing conductance g_H represents Ca2+ channels which serve to regulate bursting behaviour. The phase plot of eigenvalues over a range of g_H can be seen in Fig. 12. The eigenvalue colour scheme is listed below the diagram. Superimposed on the Figure is a series of dots indicating the response-type of Wilson's neuron simulated with the g_H - I_0 values used at that location on the phase plot.

Additionally, the thick black contour shows the amount of real roots at each g_H - I_0 location. Most of this area just has the one root, but as can be seen at the top and bottom centre of Fig. 3, three roots exist in some locations as well (note there is only ever 1 real root or 3 real roots because if any one root goes imaginary it much be accompanied by another conjugate pair-root)

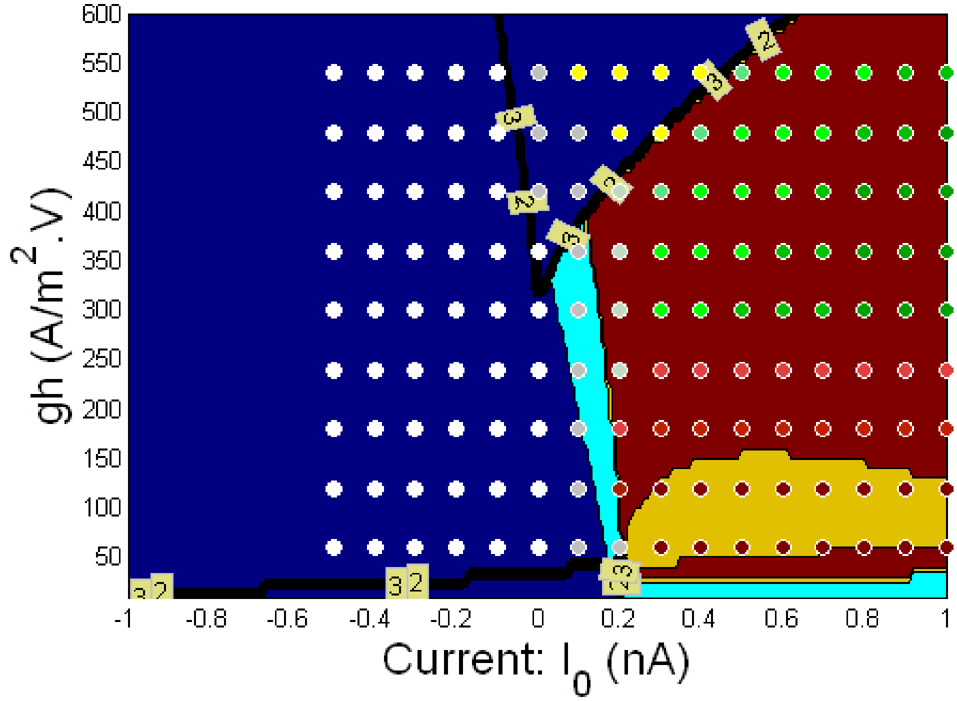

Figure 12 – g_H - I_0 phase plot of Eigenvalue-types where $g_X = 20\mathrm{A/m^2V}$

Eigenvalue-phase colour code

Dark Blue:		-Real		
Cyan:		-Real	+Imag	-Imag
Orange	+Real	-Real		
Brown:	+Real	-Real	+Imag	-Imag

They response-type dots colour code is:

White:	Resting
Grey:	Oscillating
Yellow - Dark Green:	Spiking (slow frequency (lighter) – high frequency (darker), 1 – 8 spikes within 500ms)
Light Red – Dark Red	Bursting (darkness by number of spikes per burst, 2 – 8 spikes)

As Fig. 12 shows, a I_0 current of roughly 0.19nA is needed for any spiking or bursting behaviour to occur at all, across all the values of g_H. The after-hyperpolarizing current's (I_{AHP}) modulating effect is apparent here, as its conductance $g_H$is increased, the amount of spiking per burst decreases and at a point ($> 250\mathrm{A/m^2V}$) the bursting ceases. From here only spiking occurs which happens less frequently as g_H is increased further still. It is worth noting that the transition from bursting to spiking is a smooth process, from bursts with 7 spikes in the first row of $g_H = 60\mathrm{A/m^2V}$, to 6, to 3, to 2 spikes per burst in the forth for of $g_H = 240\mathrm{A/m^2V}$ and then by the fifth row of $g_H = 300\mathrm{A/m^2V}$, the period has not changed a great deal but the number of spikes per burst again decreases, to 1 this time, and suddenly this is not classified as a *burst* but a *spike*. This clean transition between bursting and spiking behaviours makes it very hard to predict by using phase plots of eigenvalues or maximum-real component maps such as in Fig. 13. This transition is definitely not as clear cut in stability analysis as from resting to bursting as discussing in light of Fig. 3.

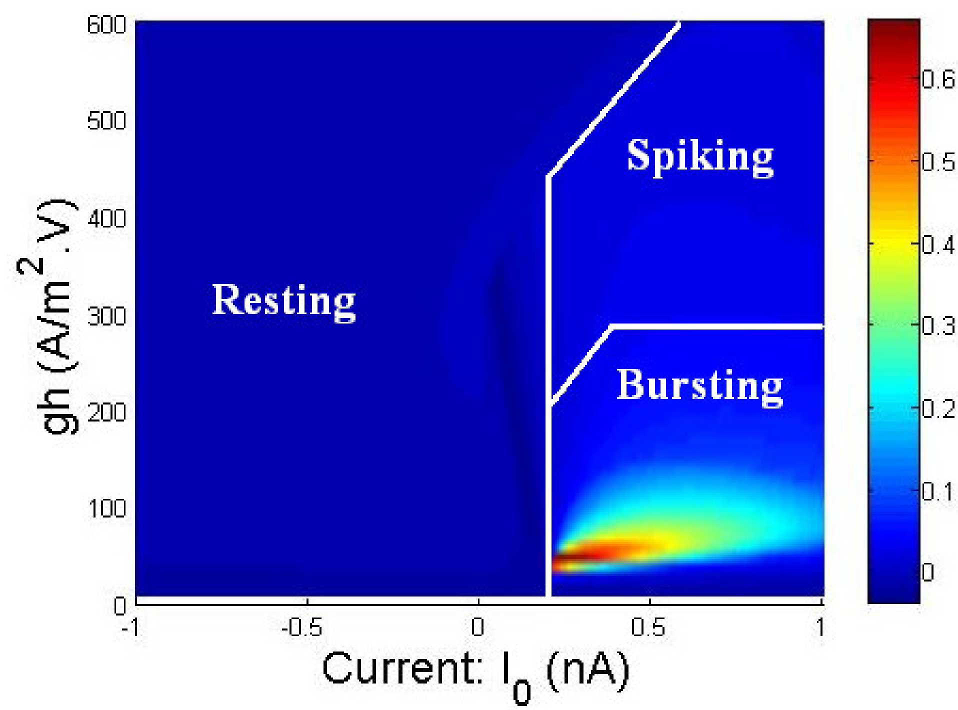

Figure 13 –Max. (Real-part) of 4 Eigenvalues map over g_H - I_0 plane where $g_X = 20\mathrm{A/m^2V}$

The imaginary component of eigenvalues initially appears to have some correlation with behaviour type. Within the rough boundaries defined by the simulation points shown in Fig. 12, resting behaviour has no imaginary parts as all. Points within the spiking zone do have a small imaginary component (but as post discussion of Fig. 3, the magnitude of an eigenvalue is not necessarily important). The bursting zero does have a region of large imaginary components for $g_H < 30\text{A/m}^2\text{V}$ above the I_0 threshold 0.19nA but between values of $g_H = 70\text{A/m}^2\text{V}$ and $g_H = 120\text{A/m}^2\text{V}$ the eigenvalues are completely real, and yet simulations show (Fig. 12) that bursting still occurs there, and so the bursting behaviour must not be dependent on the imaginary components of those eigenvalues.

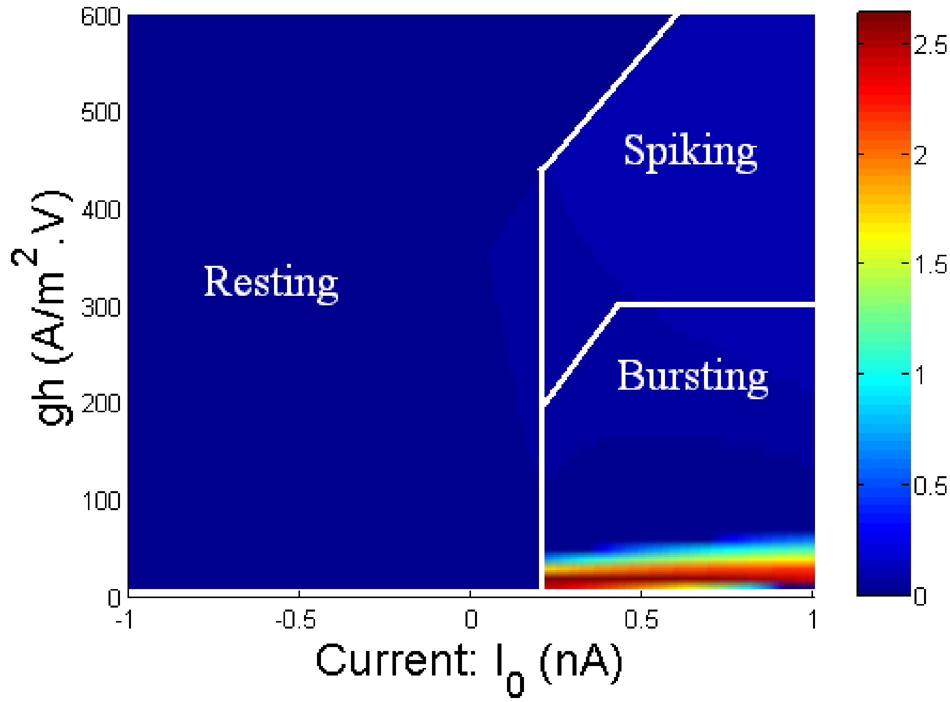

Figure 14 –Max. (Imaginary-part) of 4 Eigenvalues map over g_H - I_0 plane where $g_X = 20\text{A/m}^2\text{V}$

Behaviour in the g_X - I_0 plane

The depolarizing transient conductance g_X acts analogously to g_H. The phase diagram of eigenvalues (Fig. 15) appears different but the bursting dependence of this parameter is very much the same. As before no spiking or bursting occurs below I_0 threshold of 0.19nA and the spiking / bursting divide is defined by a horizontal threshold of the g value. This threshold of g_X in this case is approximately $12\text{A/m}^2\text{V}$. The reason bursting occurs for values of g_X greater than this threshold (rather than 'less than' as for g_H) is due to the equations:

$$I_T = g_X X(V - V_X), \quad (22)$$

$$I_{AHP} = g_H H(V - V_H), \quad (23)$$

$$I_{eff} = I_0 - I_T - I_{AHP}. \quad (24)$$

where $V_X = 140\text{mV}$, $V_H = -95\text{mV}$ and resting potential $\bar{V}$ for the neuron is approximately -70mV. Hence $(\bar{V} - V_X)$ is normally negative, $(\bar{V} - V_H)$ is normally positive, and therefore g_X and g_H need to increase and decrease respectively in order to increase I_{eff} which drives neocortical neuron behaviour from bursting from spiking.

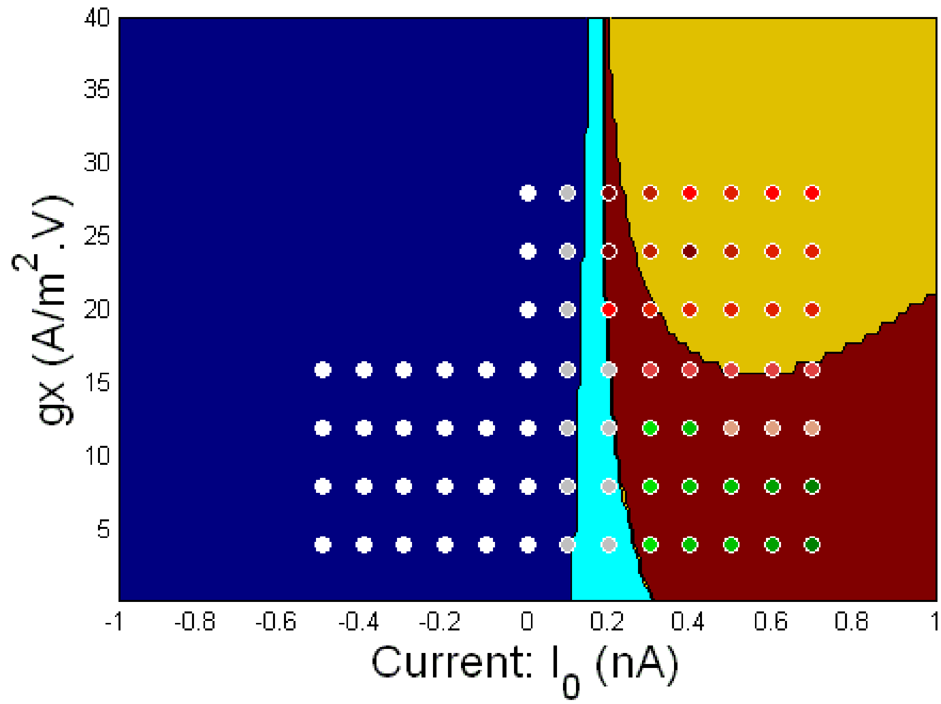

Figure 15 – g_X - I_0 phase plot of Eigenvalue-types where $g_H = 130\mathrm{A/m^2V}$
(Note: Colour codes of Fig. 15 as per those listed under Fig. 12)

As in Fig. 12, the conductance coefficient g determines whether a neuron is able to spike or burst as long as the external threshold current of 0.19nm is met. It controls the 'duty cycle' of active periods of bursting compared to the period between bursts. The external current I_0 however has a direct control over the period between bursts, and reduces this period for an increase in current. Two simulations below, Fig. 16 and Fig. 17, show this. At this low g_x value where spiking occurs, there are 4 spikes within a 500ms period when $I_0 = 0.30\mathrm{nA}$, and when the current I_0 is raised to $0.70\mathrm{nA}$, there is 8 spikes within a 500ms period.

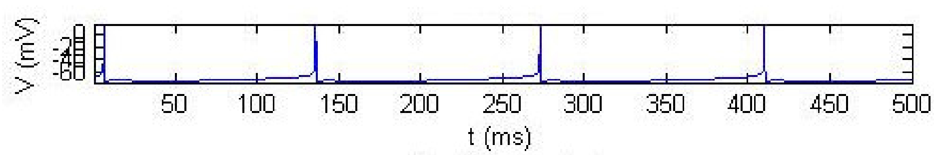

Figure 16 –Wilson's 4D model simulation of Bursting where $g_x = 4\mathrm{A/m^2V}$, $I_0 = 0.30\mathrm{nA}$

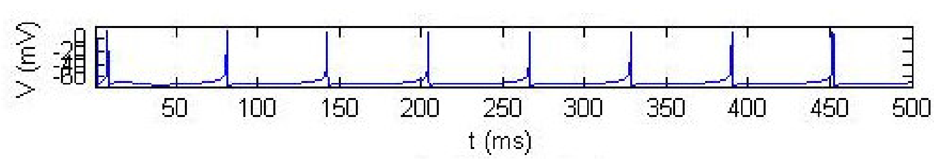

Figure 17 –Wilson's 4D model simulation of Bursting where $g_X = 4\mathrm{A/m^2V}$, $I_0 = 0.70\mathrm{nA}$

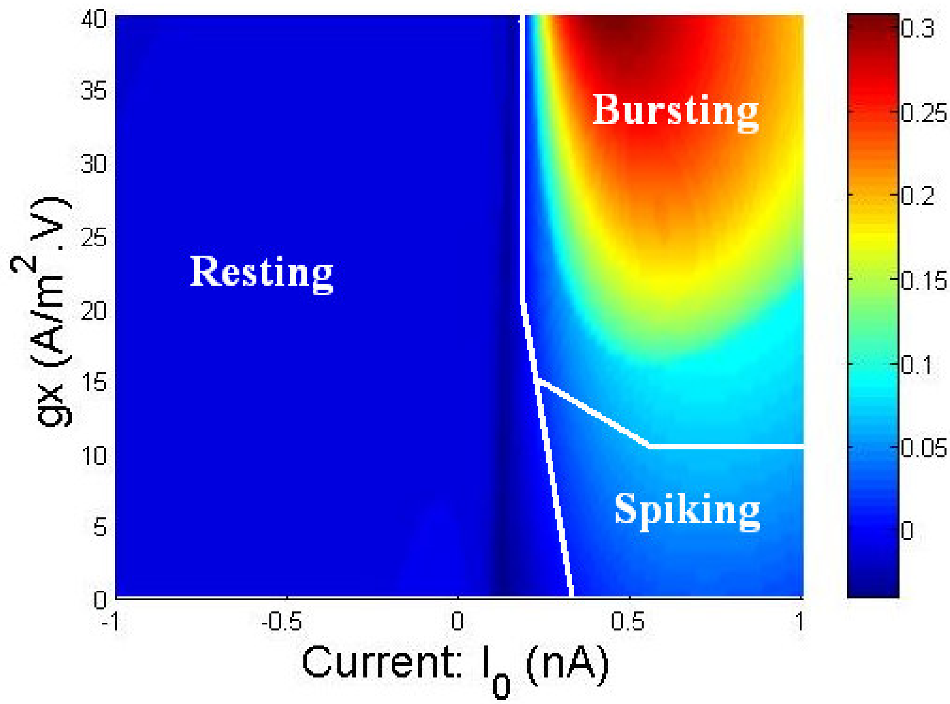

Figure 18 –Max. (Real-part) of 4 Eigenvalues map over g_X - I_0 plane where $g_H = 130\mathrm{A/m^2V}$

As in the equivalent length figures Fig. 13 and Fig. 14 for parameter g_H, the hypothesis hold here as well that real component eigenvalues determine the behaviours of how a neuron behaves, or at least boundaries between resting and spiking / bursting. The imaginary component t shown in Fig. 19 does not appear to correlate with behaviour boundaries.

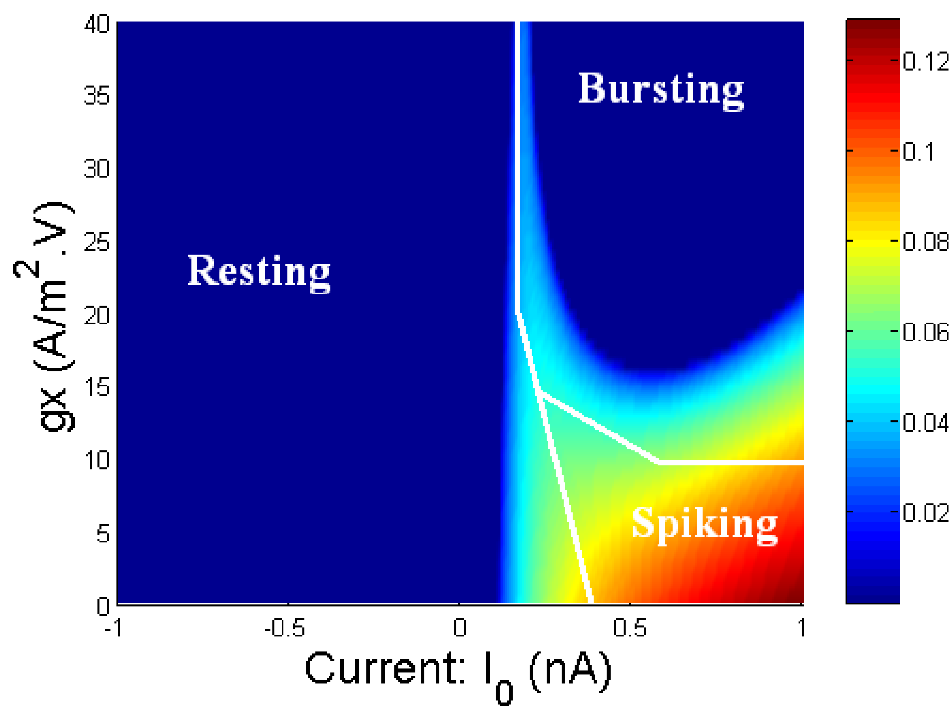

Figure 19 –Max. (Imaginary-part) of 4 Eigenvalues map over g_X - I_0 plane where $g_H = 130\mathrm{A/m^2V}$

V. Conclusions

In conclusion the after-hyperpolarizing conductance g_H and the depolarizing transient-inward conductance g_X have shown to provide defined boundaries between spiking and bursting (as seen Fig. 12 and Fig. 15 respectively). It is not however clear as to where these boundaries are if not for many simulations of the neuron for different parameter values of g_H, g_X and I_0. There is however a clear distinction between resting and spiking / bursting behaviour. It is found from linear algebra theory and confirmed with simulations (Fig. 12 and Fig. 15) that if all Jacobian eigenvalues are negative, the system is stable and limit cycling cannot occur, however if at least one has a positive real component, the system will exhibit some form of spiking or bursting. No relationship was found to link the imaginary part of Jacobian eigenvalues to the behaviour of a neuron though, but that may be a task for future work.

For planes of $g_H - I_0$ and $g_X - I_0$, a threshold value of $I_0 = 0.19\text{nA}$ was required for any spiking or bursting behaviour to occur, and if above this threshold, the period between spikes/bursts is reduced with an increase in external current I_0, speeding up of the system dynamics.

Acknowledgements

Many thanks to Dr. Jong-Won Kim and Prof. Peter A. Robinson of the Complex Systems group of the Physics Department for their efforts in guiding me in this project and offering their help whenever I needed it.

References

Hodgkin, A. L. & Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. J. Physiol. 117, 500-544

Izhikevich, E.M., 2004. Which model to use for cortical spiking neurons? IEEE Trans. Neural Networks 15, 1063-1070

Robinson, P. A., Wu H., Kim, J.W. (2007). Neural rate equations for bursting dynamics derived from conductance-based equations. J. Theor. Biol. 250, 663-672

Wilson, H. R. (1999). Simplified dynamics of human and mammalian neocortical neurons. J. Theor. Biol. 200, 375-388

Kandel, ER, JH Schwartz and TM Jessell (2000) Principles of Neural Science. New York: McGraw-Hill

Appendix

MATLAB CODE "bifurcation.m" (used for bifurcations & eigenvalue-map figures)

```
clear all
close all
clc

%% Parameters: code taken from;  function ts=wilson4(dt,Tmax)
 para(1)=1.0;       C = para(1);     %C   %Capacitance   %%
1.0uF/(cm^2)....10mF/(m^2)
 para(2)=26.0;      gr= para(2);     %gr  %Conductance 'R'  (eqn10)
 para(3)=2.0;       gx= para(3);     %gx  %........... 'X'  (eqn11)
!RowanChanged!!
 para(4)=13;      gh= para(4);     %gh  %........... 'H'  (eqn11)
!RowanChanged!!  %%13.0
 para(5)=2.1;       tr= para(5) ;    %tr  %time const R
 para(6)=15.0;      tx= para(6);     %tx  %.......... X
 para(7)=56.0;      th= para(7);     %th  %.......... H
 para(8)=17.81;     v0= para(8);     %v0  %Conductivity_Base (='g' when
V=0)  (eqn5)
 para(9)=47.58;     v1= para(9);     %v1  %............_Linear
(eqn5)                     #  48 mV
 para(10)=33.8;     v2= para(10);    %v2  %............_Quad
(eqn5)                     ###  3.38*10^4 A.m^-2.V^-2
 para(11)=-0.95;    vr= para(11);    %vr  %K+  reversal potential (eqn3)  =
-95mV       NB = 'vh'        ###  -95 mV
 para(12)=1.4;      vx= para(12);    %vx
 para(13)=-0.95;    vh= para(13);    %vh  %K+  reversal potential (eqn3)  =
-95mV       NB = 'vr'        ###  -95 mV
 para(14)=0.48;     V1= para(14);    %V1  %Na+ reversal potential (eqn3)  =
+48mV
 para(15)=-0.38;    V2= para(15);    %V2    %used in Rinf
###  -38   mV
 para(16)=-0.754;   V3= para(16);    %V3    %used in Xinf
###  -75.4 mV (?)
 para(17)=-0.7;     V4= para(17);    %V4    %used in Xinf
###  -70   mV
 para(18)=0.79;     r0= para(18);    %r0
 para(19)=1.29;     r1= para(19);    %r1
 para(20)=3.3;      r2= para(20);    %r2
 para(21)=9.0;      x2= para(21);    %x2
 %para(22)=0.23;     I0= para(22);    %I0


%% Find Fixed Points:
 %--- Changables ----%
 I0 = -1:0.01:1;
 V  = -2:0.04:2;
 %--------------%
 length_I0 = length(I0);
 YES = 1;       %enum
 NO = 0;        %enum
 STABLE   = 5; %enum
 UNSTABLE = 6; %enum
 ONE = 1;   %used for heuristic contour
 TWO = 2;   %used for heuristic contour
 THREE = 3; % no. of roots
```

```
FOUR  = 4; % side length jacobian (4 elements V,R,X,H)

VdotM = zeros(4,4); %allocation

R_inf = [0 r1 r0] + r2*conv([1 -V2],[1 -V2]);
X_inf = x2*conv([1 -V3],[1 -V4]);
gv = [v2 v1 v0];

R_fix = R_inf;
X_fix = X_inf;
H_fix = 3*X_fix;

VdotM(1,:) =    conv(gv,    [1 -V1]);
VdotM(2,:) = gr*conv(R_fix,[1 -vr]);
VdotM(3,:) = gx*conv(X_fix,[1 -vx]);
VdotM(4,:) = gh*conv(H_fix,[1 -vh]);
VdotTmp = -VdotM(1,:)-VdotM(2,:)-VdotM(3,:)-VdotM(4,:);

dVdotDV = polyder(VdotTmp)/C;  %note the I0 added on later would get
cancelled out in differentiation so ok to disregard here.

% %Display Vdot function when I0 = zero;
figure(1)
plot(V,polyval(VdotTmp,V)/C);
title('Plot of Vdot(I0=0)')
ylabel('Vdot (Volts/10)')
xlabel('Voltage (Volts/10)')
grid on

% %Find the Voltage roots for each I0 value
roots_Vdot = zeros(length_I0,THREE); %allocation
for i = 1:length_I0
    Vdot = ([0,0,0,I0(i)]+VdotTmp)/C;
    roots_Vdot(i,:) = roots(Vdot);
end


% %Find the Real roots & compile
realRoots_Vdot      = NO * ones(length_I0,THREE); %init & allocation
realRoots_Vdot_stab = NO * ones(length_I0,THREE); %init & allocation
for i=1:length_I0
    for j = 1:THREE
        if isreal(roots_Vdot(i,j));
            if polyval(dVdotDV,roots_Vdot(i,j)) <= 0
                %test = polyval(dVdotDV_fix,I0(i));
                stability = STABLE;
            else
                stability = UNSTABLE;
            end
            realRoots_Vdot(i,j)            = roots_Vdot(i,j);
            realRoots_Vdot_stab(i,j)       = stability;
        end
    end
end

figure(2)
hold on
for j = 1:THREE  %numReals = [0,3]
prev_i = 1;         %init
prevStability = NO; %init
    for i = 1:length_I0
```

```
        stability = realRoots_Vdot_stab(i,j);
        if ((stability ~= prevStability) && (i > 1)) || i==length_I0
            if prevStability == STABLE
               plot(I0(prev_i:i-1),realRoots_Vdot(prev_i:i-1,j))
            elseif prevStability == UNSTABLE
               plot(I0(prev_i:i-1),realRoots_Vdot(prev_i:i-1,j),'--')
            end
            prev_i = i;
        end
        prevStability = stability;
    end
end
hold off
title(['Voltage Bifurcation:  gh = ',num2str(gh),' (10.A/m^2V), gx =
',num2str(gx),' (10.A/m^2V)',]) %"C*dV/dt = I_0 - I(leak) - I_R - I_X -
I_H" (Wilson 92)')
ylabel('Voltage (100mV)')
xlabel('Current I_0 (nA)')
set(get(gca,'XLabel'),'FontSize',14);
set(get(gca,'YLabel'),'FontSize',14);
set(get(gca,'title'),'FontSize',16);
grid on


%% Jacobian stuff 25th September
%NB INIT VALUES:
%-------------
%para(6)=15.0;  tx
%para(7)=56.0;  th
%para(3)=2.0;   gx
%para(4)=13.0;  gh
%---changeables:
lengthJ = 60;
txJ = linspace(tx,tx,lengthJ);
thJ = linspace(th,th,lengthJ);
gxJ = linspace(gx,gx,lengthJ);
%ghJ = linspace(gh,gh,lengthJ);

ghJ = linspace(1,60,lengthJ);
%gxJ = linspace(0.025,4,lengthJ);
%txJ = linspace(2,120,lengthJ);
%thJ = linspace(2,300,lengthJ);
paraJ = ghJ;       %!this is the change variable !!!
stringJ = 'gh';    %!this is the change variable !!!

%%% NOTE THIS ONLY WORKS WITH THE FIRST ROOT

j=0; %inint
signEigenJacM = zeros(length_I0,lengthJ,FOUR);   %allocation
eig_Jacobian = zeros(length_I0,lengthJ,FOUR);    %allocation
record_realRoot1 = zeros(length_I0,lengthJ);     %allocation
test_roots_Vdot = zeros(length_I0,lengthJ,3);    %allocation
totalRealRoots = zeros(length_I0,lengthJ);       %allocation
biggestRealEigenJacM = zeros(length_I0,lengthJ); %allocation
biggestImagEigenJacM = zeros(length_I0,lengthJ); %allocation
realEigenvalues = zeros(length_I0,lengthJ,FOUR); %allocation
for j = 1:lengthJ
    tx = txJ(j);
    th = thJ(j);
    gx = gxJ(j);
    gh = ghJ(j);
```

```
    %% copied in from above:
    %% %-----------------------------------------------------------------
    R_inf = [0 r1 r0] + r2*conv([1 -V2],[1 -V2]);
    X_inf = x2*conv([1 -V3],[1 -V4]);
    gv = [v2 v1 v0];
    R_fix = R_inf;
    X_fix = X_inf;
    H_fix = 3*X_fix;
    VdotM(1,:) = conv(gv,[1 -V1]);
    VdotM(2,:) = gr*conv(R_fix,[1 -vr]);
    VdotM(3,:) = gx*conv(X_fix,[1 -vx]);
    VdotM(4,:) = gh*conv(H_fix,[1 -vh]);
    VdotTmp = -VdotM(1,:)-VdotM(2,:)-VdotM(3,:)-VdotM(4,:);
    dVdotDV = polyder(VdotTmp)/C;  %note the I0 added on later would get
cancelled out in differentiation so ok to disregard here.

    %%% (not cpoied in, an interum)
    DVdotDV=-polyder(conv(gv,[1 -V1])) - gr*R_fix - gx*X_fix - gh*H_fix;
DVdotDR=-gr*[1 -vr]/C;   DVdotDX=-gx*[1 -vx]/C;    DVdotDH=-gh*[1 -vh]/C;
    DRdotDV=polyder(R_inf)/tr;     DRdotDR = -1/tr;         DRdotDX = 0;
DRdotDH = 0;
    DXdotDV = polyder(X_inf)/tx;   DXdotDR = 0;             DXdotDX = -
1/tx;          DXdotDH = 0;
    DHdotDV = 0;                   DHdotDR = 0;             DHdotDX =
3/th;           DHdotDH = -1/th;

    roots_Vdot = zeros(length_I0,THREE); %allocation
    for i = 1:length_I0
        Vdot = ([0,0,0,I0(i)]+VdotTmp)/C;
        roots_Vdot(i,:) = roots(Vdot);
        for k = 1:THREE
           totalRealRoots(i,j) = totalRealRoots(i,j) +
isreal(roots_Vdot(i,k));  %count up #fixed-pts at locations
        end
    end
    % %Find the Real roots & compile
    numReals=zeros(1,THREE);      %allocation
    countI0withReals = 0;         %init
    realRoots_Vdot = NO * ones(length_I0,THREE,3); %init & allocation
    for i=1:length_I0
        numReals = 0;             %init  %num reals in the one i'th value
of I0
        flag_atLeastOneRoot = NO; %init
        for k = 1:THREE
            if isreal(roots_Vdot(i,k));
                if flag_atLeastOneRoot == NO
                    flag_atLeastOneRoot = YES;
                    countI0withReals=countI0withReals+1;
                end
                if polyval(dVdotDV,roots_Vdot(i,k)) <= 0
                    stability = STABLE;
                else
                    stability = UNSTABLE;
                end
                numReals=numReals+1; % count real roots every i'th I0
                realRoots_Vdot(countI0withReals,numReals,:) =
[I0(i),roots_Vdot(i,k),stability];
            end
        end
    end
```

```
     size1_realRoots_Vdot = size(realRoots_Vdot,1);  %%size of:
countI0withReals
     size2_realRoots_Vdot = size(realRoots_Vdot,2);  %%size of: numReals

     %%%%
     %end code copied in from above
     %% %----------------------------------------------------------------------

     for i = 1:length_I0
     realRoot1   =realRoots_Vdot(i,1,2);
%%realRoots_Vdot(countI0withReals,numReals,:) =
[I0(i),roots_Vdot(i,j),stability];
     record_realRoot1(i,j) = realRoot1;

     Jacobian = [polyval(DVdotDV,realRoot1), polyval(DVdotDR,realRoot1),
polyval(DVdotDX,realRoot1), polyval(DVdotDH,realRoot1);...
                 polyval(DRdotDV,realRoot1), polyval(DRdotDR,realRoot1),
polyval(DRdotDX,realRoot1), polyval(DRdotDH,realRoot1);...
                 polyval(DXdotDV,realRoot1), polyval(DXdotDR,realRoot1),
polyval(DXdotDX,realRoot1), polyval(DXdotDH,realRoot1);...
                 polyval(DHdotDV,realRoot1), polyval(DHdotDR,realRoot1),
polyval(DHdotDX,realRoot1), polyval(DHdotDH,realRoot1)];

     eig_Jacobian(i,j,:) = eig(Jacobian);
     signEigenJacM(i,j,:) = sort(real(sign(eig_Jacobian(i,j,:))));  %NB
'Sign': negReal = -1, posReal = 1, zero = 0, 0<imagPosReal<1,-
1<imagNegReal<0
     biggestRealEigenJacM(i,j) = max(real(eig_Jacobian(i,j,:)));
     biggestImagEigenJacM(i,j) = max(imag(eig_Jacobian(i,j,:)));
     realEigenvalues(i,j,:) = sort(real(eig_Jacobian(i,j,:)));
     end
 end



if YES
    figure(3)
    hold on
    plotData = zeros(length_I0,FOUR); %allocation
    for i=1:length_I0
       for k = 1:FOUR
           plotData(i,k) = realEigenvalues(i,13,k);  %13 if gh is going 1-
60, will be it's default value.
       end
    end
    plot(I0,plotData)
    title('Jacobian Eigenvales of Fixed Points V,R,X,H')
    ylabel('Eigenvalue')
    xlabel('current I_0')
    plot(I0,0*I0,'k')  % construct a reference line along zero
    hold off
end



 testheur_signEigenJacM = NO*ones(1,15); %allocation
 Total_flag = YES*ones(1,FOUR);
 heur_signEigenJacM = zeros(size1_realRoots_Vdot,lengthJ); %allocation
 ONE_CLOSE = 0.999999; % close to one to avoid float/integer errors
 ZERO_CLOSE = 0.0000001;
```

```
 for i = 1:size1_realRoots_Vdot
    for j = 1:lengthJ
       flags = [0,0,0,0];   %ENCODE flags:  1)NegImag  2)PosReal  3)PosImag
4)BothImag
       for k=1:FOUR
            %%%eig_Jacobian
           test66 = eig_Jacobian(i,j,k);
           if imag(eig_Jacobian(i,j,k)) < -ZERO_CLOSE
% 1)NegImag
               flags(1) = YES;
           else
               Total_flag(1) = NO;
           end
           if imag(eig_Jacobian(i,j,k)) > ZERO_CLOSE
% 2)PosImag
               flags(2) = YES;
           else
               Total_flag(2) = NO;
           end
           if real(eig_Jacobian(i,j,k)) < -ZERO_CLOSE
% 3)NegReal
               flags(3) = YES;
           else
               Total_flag(3) = NO;
           end
           if real(eig_Jacobian(i,j,k)) > ZERO_CLOSE
% 4)PosReal
               flags(4) = YES;
           else
               Total_flag(4) = NO;
           end
       end

       heur_signEigenJacM(i,j) = 1*flags(1) +2*flags(2) +4*flags(3)
+8*flags(4);
       testheur_signEigenJacM(heur_signEigenJacM(i,j)) = YES;
       %% only the heuristic-values of 4,7,12,15 occurred, so change to
1,2,3,4 so not a heaps of grouped contours to look at:
       if heur_signEigenJacM(i,j) == 4     %flags(3) = NegReal eigens
           heur_signEigenJacM(i,j) = 1;
       elseif heur_signEigenJacM(i,j) ==7  %flags(1) & flags(2) & flags(3)
= NegImag & PosImag & NegReal eigens
           heur_signEigenJacM(i,j) = 2;
       elseif heur_signEigenJacM(i,j) ==12 %flags(3) & flags(4) = NegReal &
PosReal eigens
           heur_signEigenJacM(i,j) = 3;
       elseif heur_signEigenJacM(i,j) ==15 %flags(1) & flags(2) & flags(3)
& flags(4) = NegImag & PosImag & NegReal & PosReal eigens
           heur_signEigenJacM(i,j) = 4;
       else
           disp('heur_signEigenJacM(i,j) error')
       end
    end
 end


 disp(Total_flag)
 figure(5)
 hold on
 [C,h] = contourf(I0,paraJ,heur_signEigenJacM','LevelStep',1);
```

```
 set(gca, 'CLim', [1, 4]);
 title(['Eigenvalue-Type map:  ',stringJ,' vs. I_0'])
 ylabel([stringJ,' (10.A/m^2.V)'])
 xlabel('Current: I_0 (nA)')
 [CC,hh] =
contour(I0,paraJ,totalRealRoots','LevelStep',1,'LineWidth',3,'LineColor','k
');
 text_handle = clabel(CC,hh);
 set(text_handle,'BackgroundColor',[1 1 .6],'Edgecolor',[.7 .7 .7]);
 spots();
 set(get(gca,'XLabel'),'FontSize',18);
 set(get(gca,'YLabel'),'FontSize',18);
 set(get(gca,'title'),'FontSize',20);
 hold off



 figure(8)
 H_surf1 = surf(I0,paraJ,biggestRealEigenJacM');
 %colormap(winter);
 set(H_surf1, 'linestyle', 'none');
 colorbar
 shading interp
 hold on
 spots();
 hold off
 title(['Eigenvalue map, Max. Real-part of 4 Eigenvalues:  ',stringJ,' vs.
I_0'])
 ylabel([stringJ,' (10.A/m^2.V)'])
 xlabel('Current: I_0 (nA)')
 set(get(gca,'XLabel'),'FontSize',18);
 set(get(gca,'YLabel'),'FontSize',18);
 set(get(gca,'title'),'FontSize',20);
 grid off

 figure(9)
 H_surf2 = surf(I0,paraJ,biggestImagEigenJacM');
 set(H_surf2, 'linestyle', 'none');
 colorbar
 shading interp
 hold on
 spots();
 hold off
 title(['Eigenvalue map, Max. Imag-part of 4 Eigenvalues:  ',stringJ,' vs.
I_0'])
 ylabel([stringJ,' (10.A/m^2.V)'])
 xlabel('Current: I_0 (nA)')
 set(get(gca,'XLabel'),'FontSize',18);
 set(get(gca,'YLabel'),'FontSize',18);
 set(get(gca,'title'),'FontSize',20);
```