CUED - Engineering Tripos Part IIB 2019-2020 **Module Coursework**

Module	4G3	Title of report	4G3 - Coursework One - The asynchronous and irregular state of cortical circuits

Date submitted: 25/2/20	Assessment for this module is ☐xxx 100% / ☐ 25% coursework of which this assignment forms _____ %

UNDERGRADUATE STUDENTS ONLY		**POST GRADUATE STUDENTS ONLY**			
Candidate number:	5584F	Name:		College:	

Feedback to the student

☐ **See also comments in the text**

		Very good	**Good**	Needs improvmt
CONTENT	**Completeness, quantity of content:** Has the report covered all aspects of the lab? Has the analysis been carried out thoroughly?			
	Correctness, quality of content Is the data correct? Is the analysis of the data correct? Are the conclusions correct?			
	Depth of understanding, quality of discussion Does the report show a good technical understanding? Have all the relevant conclusions been drawn?			
	Comments:			
PRESENTATION	**Attention to detail, typesetting and typographical errors** Is the report free of typographical errors? Are the figures/tables/references presented professionally?			
	Comments:			

Overall assessment (circle grade)	A*	**A**	B	C	D
Guideline standard	>75%	**65-75%**	55-65%	40-55%	<40%
Penalty for lateness:		*20% of marks per week or part week that the work is late.*			

Marker:

Date:

4G3 - Coursework One - The asynchronous and irregular state of cortical circuits

Candidate Number: 5584F

February 25, 2020

Abstract

Leaky integrate and fire neurons were considered in these experiments. Investigations consisted of gradually building up a network starting with a single excitatory input, working up to a randomly connected set of three populations with an external source X, an excitatory E and inhibitory I population. The spike variation of the populations was considered and compared against values within the cortex, the value obtained improved as model complexity grew. It was then shown that the final model had limitations and could be broken by generating a fully connected network.

I. Introduction

Mimicing the architecture of the cortex within a computer has a vast potential for creating complex systems that operate effectively and quickly. Within this report a leaky integrate and fire (LIF) neuron will be investigated and a network built up around it.

II. Poisson Spike Train

An external population 'X' is used to generate N independent spike train inputs to the internal network. These are generated through a Poisson process which can be approximated as a Bernoulli distribution in discrete time when δt is sufficiently small as to ensure a small probability of multiple spikes occurring in one time step. Figure 1 shows a raster plot for N=1000 over 2 seconds for firing rate $r_x = 10Hz$. The theoretic average spike count per neuron is Tr_x=20. An average rate of 20.06 was observed.

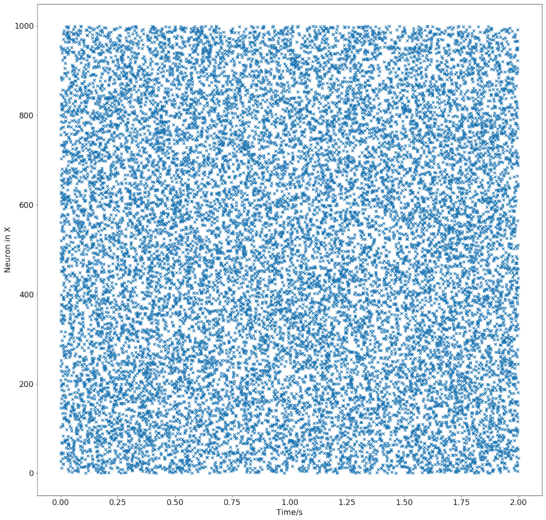


Figure 1: *Raster plot for X population* ✓

III. Single Input LIF Neuron

These spike trains are inputs to LIF neurons. The membrane potential of these follows equation 1 as the population X is excitatory.

$$V_k = V_{k-1} + \delta t[\frac{-V_{k-1}}{\tau} + \frac{w}{K}\sum_{j=1}^{K} S_j(k-1)] \quad (1)$$

When the neuron is subject to only one input with w=0.9 the membrane potential and output spike train in figure 2 is observed.

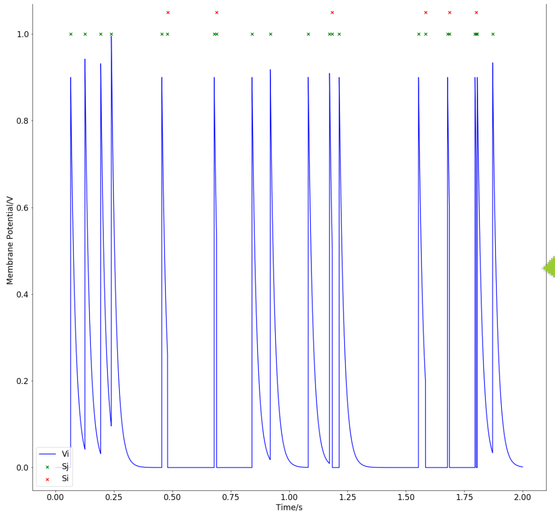


Figure 2: *Membrane potential, and output with only a single input*

IV. Multi Input LIF Neuron

i. a

A real network consists of each neuron having many inputs. Figure 3 shows the behaviour of the membrane potential for w=1 with K=100 inputs. The spike and reset mechanism has been disabled to allow the stationary distribution to be observed.

ii. b,c

After a burn-in time of approximately 100ms the membrane potential reaches a stationary distribution. The theoretical derivation for the mean and variance of V can be seen in the appendix section 1.

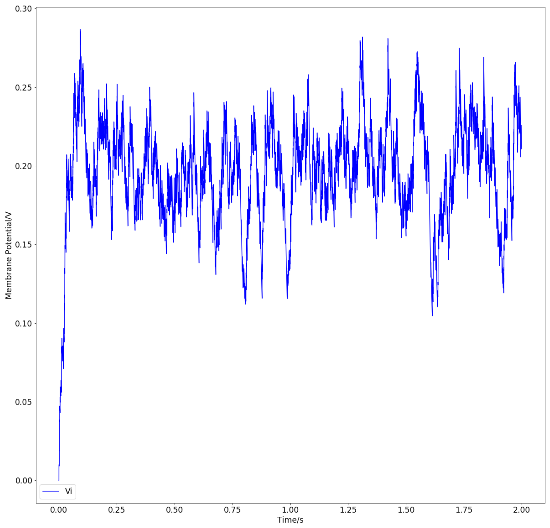


Figure 3: *Multi-Input with potential resetting and w=1*

Figure 4 shows the comparison of this theoretical result with values calculated from simulations at varying K values. Showing a good match between simulations and theory.

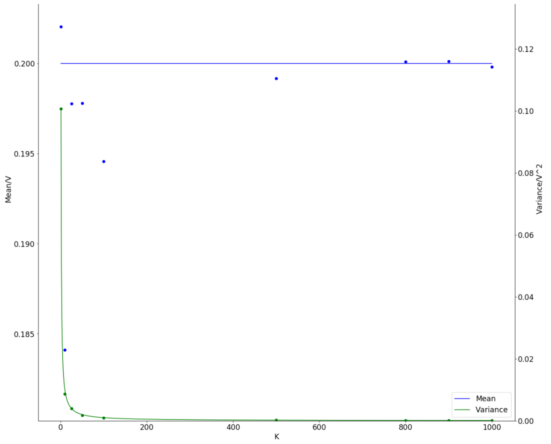


Figure 4: *Membrane potential vs inputs to neuron*

iii. d

As $\mu = wr_x\tau$, the mean can be set to the threshold voltage V_{th} easily by adjusting w. With V_{th}=1 a weight of 5 is required. Figure 5 shows a simulation of this neuron.

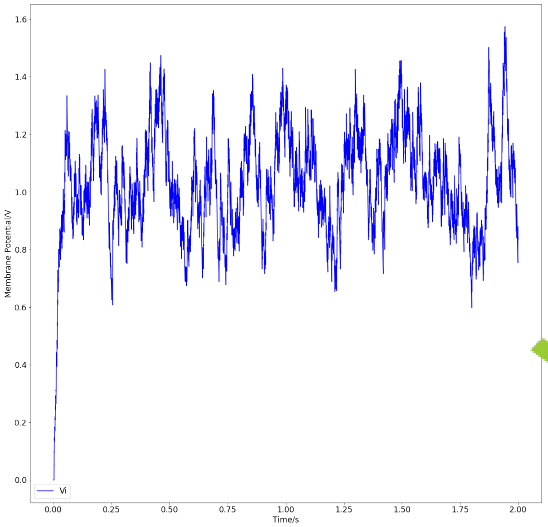


Figure 5: *Mean membrane potential = V_{th}=1*

iv. e

The input weight 'w' has a direct effect on neuron firing rate. Table 1 shows approximate values of w required to achieve a rate of 10Hz. This value varies a lot as it's highly dependant on the random input spike trains from population X. The Fano factor of the output spikes can be computed using the variance and mean spike count within a 100ms window, then dividing the variance by the mean. These factors are also shown in table 1. The Fano factor gives information about the spike variability in a network. For our cortex this is greater then 1, which is not what is observed for this simulated network which shows lesser variability.

w	r/Hz	Fano factor
4.2	10	0.446
4.22	9	0.54
4.23	10.5	0.5
4.26	10.5	0.487

Table 1: *Effect of w on firing rate and Fano factor for a multi input excitatory neuron*

V. Multi E and I Input LIF Neuron

To Increase variability the model can be extended to take both excitatory inputs which increase potential and inhibitory ones which decrease potential. Equation 1 needs to be modified to account for this change. Equation 2 shows the modification required. In these examples the population weights (J) are equal in magnitude (w) but of opposite sign.

$$V_k = V_{k-1} + \delta t\Big[\frac{-V_{k-1}}{\tau} + \sum_{\beta\in[E,I,X]} \frac{J_{\alpha\beta}}{\sqrt{K}} \sum_{j=1}^{K} S_j(k-1)\Big] \tag{2}$$

i. a

The theoretic mean and variance of V can once again be calculated. This follows the same method as in appendix section 1. The variance and mean of h(k) is $2w^2 r_x(\frac{1}{\delta t} - r_x)$ and 0 respectively. These can then be used as before once again ignoring terms of $\mathbb{O}(\delta t^2)$. The mean of V is then calculated to be 0, and the variance is constant with K and is $\tau w^2 r_x$.

ii. b

The effect of w on firing rate and spike variability can be estimated with the new model. Table 2 shows the values of w that achieve a rate of approximately 10Hz. The Fano factor observed is closer to or above 1 and therefore closer to the variability of spikes in the cortex and shows the model has improved with the addition on I inputs. However, as the Fano factor sometimes is below 1 there are other improvements that may make the model more realistic.

w	r/Hz	Fano factor
1.44	10	1.06
1.46	10.5	1.34
1.48	9.5	0.63

Table 2: *Effect of w on firing rate and Fano factor for a multi E/I input neuron*

VI. Full Network

The next step in making a full network is to set up each neuron in E and I so that it takes K random inputs from each of the E,I,X populations to form the network shown in figure 6. The connectivity weights will take the values $J_{EE} = 1$, $J_{IE} = 1$, $J_{EI} = -2$, $J_{II} = -1.8$, $J_{EX} = 1$, $J_{IX} = 0.8$.

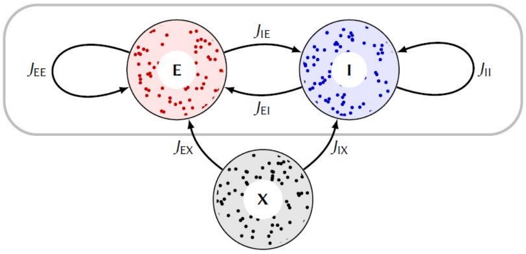


Figure 6: *Full Network*

i. a

The mean firing rate of population E and I can be calculated as follows:

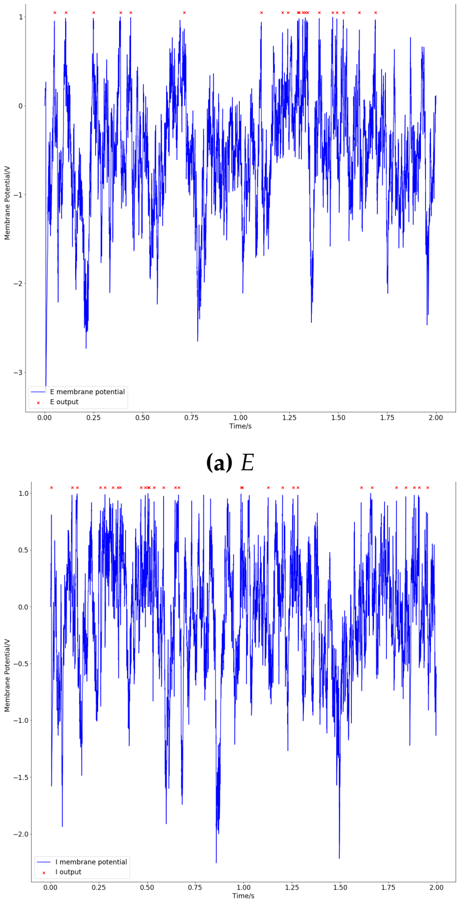
$$\begin{bmatrix} J_{EE} & J_{EI} \\ J_{IE} & J_{II} \end{bmatrix} \begin{bmatrix} r_E \\ r_I \end{bmatrix} = - \begin{bmatrix} J_{EX} \\ J_{IX} \end{bmatrix} r_x + \mathbb{O}(\frac{1}{\sqrt{K}})$$

With the values previously stated and the constant taken to be $\frac{1}{\sqrt{K}} = 0.1$ results in r_E, r_I = $1.1r_x, r_x$ respectively after solving the linear equations. With sufficiently large K $r_x = r_E = r_I$.

✓

ii. b,c

With N=1000, and K=100 the network was simulated for 2 seconds for varies values of r_x. Table 3 shows the simulated rates for population E and I. Both rates are always higher then r_x with the rate of E neurons being the highest. The ordering agrees with the previous predictions. However, the simulated results show a relationship of approximately r_E, r_I = $1.25r_x, 1.13r_x$. This difference is likely due to the network (K,N) being small and creating loops within the network that increase the firing rate. In the cortex (K,N) is large enough that this is less likely.



(a) *E*

(b) *I*

Figure 7: *Full network simulation for one neuron in E and I*

Figure 7 shows an example neuron in E and I behaving as expected.

iii. d

Reducing the network to N=K=100 causes the average rates to be r_E=40, r_I=20. Figure 8 shows the membrane potential of every E and I neuron. As each neuron is connected to every other neuron each neuron will have the same connections causing the behaviour of a population to be identical. This is shown in the figure as the lines all overlap, meaning a

r_x	$< r_E >$	$< r_I >$
5	6.95	5.83
10	12.97	11.64
15	18.29	16.88
20	24.20	22.51

Table 3: *Average firing rate of E/I neurons given an input of rate r_x*

population will behave identically as inputs, weights and connections are the same. There are differences between the two populations as the connectivity weights differ slightly, causing the difference in behaviour at the spikes. The E population fires twice as often as the I population. Upon closer examination this occurs because E neurons fire first, the next timestep they'll fire again in addition to the I population firing. This happens because all E neurons are inputs back into their population so upon all firing the potential of the E population can be raised above the threshold again within one timestep, ($\frac{K}{\delta t}\frac{J_{EE}}{\sqrt{K}} > V_{th}$). This double fire doesn't occur when the I population fires as the inhibitory connection lowers the potentials. If J_{EX} and J_{IX} are reversed then the firing rates become roughly 0 and 17 for the E,I populations. This is because with a greater weight to I from X the I population will always fire first, causing the E population lose potential. This is shown in figure9.

Several assumptions which may have been broken include. Random network connectivity broken as all connections identical. K=N which may cause chaotic behaviour as an eigenvalue value could equal 1. Loops within the network have additional side effects that weren't considered.

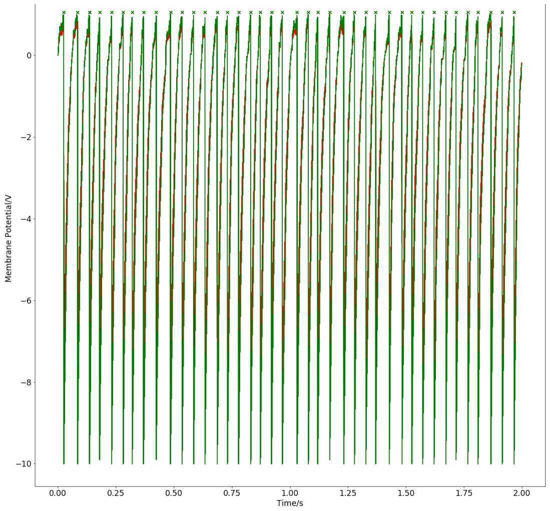


Figure 8: *Full Network when K=N=100, showing all neuron activity with I in red and E in green*

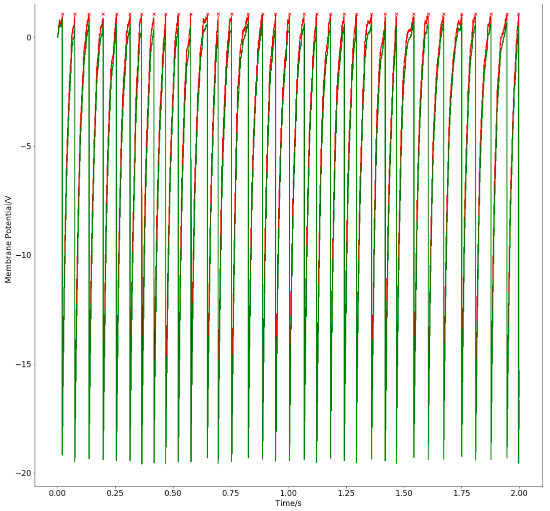


Figure 9: *Full Network when K=N=100, showing all neuron activity with I in red and E in green, with J_{EX} and J_{IX} reversed*

VII. Appendix

i. 1. Derivation of mean and variance of V(t)

First consider the spike train S_j:

$$\begin{gathered} S_j(k) \sim \mathbb{B}(r_x \delta t) \\ < S_j(k) >= r_x \\ << S_j(k) >>= \mathbb{E}(S_j(k)^2) - \mu^2 \\ = \frac{r_x \delta t}{\delta t^2} + 0 - r_x^2 = r_x(\frac{1}{\delta t} - r_x) \end{gathered} \tag{3}$$

Taking note that each input neuron j is independent and uncorrelated allows the following manipulation to occur for calculating the moments of h(k). This is because $Cov[S_i(k)S_j(k)]$=0 for $i \neq j$, allowing the variance of a sum to equal the sum of variances.

$$\begin{gathered} h(k) = \frac{w}{K}\sum_{j=1}^{K} S_j(k) \\ < h(k) >= \frac{w}{K}\sum_{j=1}^{K} \mathbb{E}[S_j(k)] = \frac{w}{K} K r_x = w r_x \\ << h(k) >>= \frac{w^2}{K^2}\sum_{j=1}^{K} << S_j(k) >> \\ = \frac{w^2}{K} K r_x(\frac{1}{\delta t} - r_x) = \frac{w^2 r_x}{K}(\frac{1}{\delta t} - r_x) \end{gathered} \tag{4}$$

Using equation 1 and taking advantage that when stationary the moments of V and h are equal at times k and k-1, the following simplifications can occur:

$$\begin{gathered} < V_k >=< V_{k-1} > +\delta t[\frac{- < V_{k-1} >}{\tau} + < h_{k-1} >] \\ < V_\infty >=< V_\infty > +\delta t[\frac{- < V_\infty >}{\tau} + < h_\infty >] \\ < V_\infty >= \tau < h_\infty >= \tau w r_x \\ \\ << V_k >>= (1 - \frac{\delta t}{\tau})^2 << V_{k-1} >> +\delta t^2 << h_{k-1} >> \; ✓ \\ << V_\infty >>= (1 - \frac{\delta t}{\tau})^2 << V_\infty >> +\delta t^2 << h_\infty >> \\ << V_\infty >> (1 - (1 - \frac{\delta t}{\tau})^2) = \delta t^2 << h_\infty >>= \delta t^2 (\frac{w^2 r_x}{K}(\frac{1}{\delta t} - r_x)) \\ << V_\infty >> (\frac{2\delta t}{\tau} - \frac{\delta t^2}{\tau^2}) = \delta t \frac{w^2 r_x}{K} + \delta t^2 \frac{w^2 r_x^2}{K} \end{gathered} \tag{5}$$

Eliminating everything of $\mathbb{O}(\delta t^2)$ results in the following:

$$\begin{gathered} << V_\infty >> \frac{2\delta t}{\tau} = \delta t \frac{w^2 r_x}{K} \\ << V_\infty >> = \frac{w^2 r_x \tau}{2K} \end{gathered} \tag{6}$$

✓