



UNIVERSITY OF
CAMBRIDGE

Assignment 2

5568D

31th March 2020

1 Hopfield Network

1.1 Theoretical probability of error

According to the theory, recalling occurs once the network is in a stable state. The probability of recalling a bit incorrectly corresponds to the probability that in the next time-step a single bit will change its value. Therefore the probability of error can be defined as:

$$P_{\text{error}} = P(H_k < 0 | r_k^{(\mu)} = 1)P(r_k^{(\mu)} = 1) + P(H_k \geq 0 | r_k^{(\mu)} = 0)P(r_k^{(\mu)} = 0) \quad (1)$$

It is assumed that the bits are distributed according to Bernoulli distribution with probability 0.5. As discussed in the lecture notes, the random variable $H_k | r_k^{(\mu)}$ is a sum of many independent random terms which, according to central limit theorem, will be distributed according to a Gaussian distribution. As showed in the lecture notes the mean and variance of this distribution will be equal to:

$$\begin{aligned} \mu(H_k | r_k^{(\mu)} = 1) &= \frac{N-1}{8} \\ \mu(H_k | r_k^{(\mu)} = 0) &= -\frac{N-1}{8} \\ \sigma^2(H_k | r_k^{(\mu)} = 1) &= \sigma^2(H_k | r_k^{(\mu)} = 0) = \sqrt{\frac{(M-1)(N-1)}{2}} \end{aligned} \quad (2)$$

The error probability can be expressed using the cumulative probability function of a standard normal distribution $\Phi(z)$:

$$\begin{aligned} P_{\text{error}} &= \Phi\left(\frac{0 - \mu(H_k | r_k^{(\mu)} = 1)}{\sigma^2(H_k | r_k^{(\mu)} = 1)}\right) \frac{1}{2} + \left(1 - \Phi\left(\frac{0 - \mu(H_k | r_k^{(\mu)} = 0)}{\sigma^2(H_k | r_k^{(\mu)} = 0)}\right)\right) \frac{1}{2} \\ &\stackrel{(a)}{=} \frac{1}{2} \left(1 - \Phi\left(\frac{0 - \mu(H_k | r_k^{(\mu)} = 0)}{\sigma^2(H_k | r_k^{(\mu)} = 0)}\right) + 1 - \Phi\left(\frac{0 - \mu(H_k | r_k^{(\mu)} = 0)}{\sigma^2(H_k | r_k^{(\mu)} = 0)}\right)\right) \\ &= 1 - \Phi\left(\frac{0 - \mu(H_k | r_k^{(\mu)} = 0)}{\sigma^2(H_k | r_k^{(\mu)} = 0)}\right) \\ &= 1 - \Phi\left(\frac{(N-1)\sqrt{32}}{8\sqrt{(M-1)(N-1)}}\right) \\ &= 1 - \Phi\left(\sqrt{\frac{N-1}{2(M-1)}}\right) \end{aligned} \quad (3)$$

Where step (a) comes from the fact that $\Phi(-z) = 1 - \Phi(z)$.

Using the results found above the error probability is calculated using a different number of patterns for two different networks: with 100 neurons or with 1000 neurons. The results are displayed in the figure below:

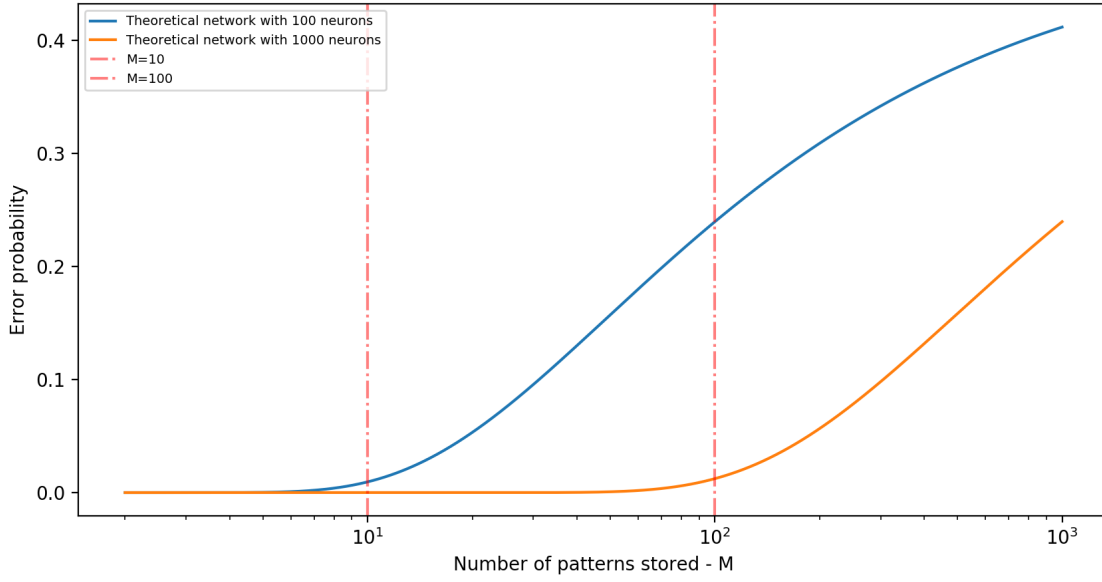


Figure 1: Error Probability vs Number of Patters Stored

As observed from the figure above the boundaries at $M = 10$ and $M = 100$ indicate that the error probability starts increasing at $M \approx 0.1N$ and therefore networks who have a large number of neurons can store more patterns error-free.

1.2 Simulated Network

Following the theory described in the lecture, first all the stored patters have to be learned i.e., the weights of the network are set according to *Hebbian Learning*:

$$W_{ij} = \sum_{m=1}^M \left(r_i^{(m)} - 0.5 \right) \left(r_j^{(m)} - 0.5 \right) \quad (4)$$

$$W_{ii} = 0$$

It is assumed that recollection happens once the network reaches a stable state. Therefore for this exercise the system is initialised in each of the stored patters. Each neuron is then updated asynchronously according to the following rule:

$$r_i(t + \Delta t) = F\left(\sum_{j=1}^N W_{ij} r_j(t)\right) \quad (5)$$

$$F(I) = \begin{cases} 1 & \text{if } I \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Following this calculations for each pattern, it is examined if a neuron stays at the initialised state or it flips and therefore causes an error. The practical error probability is found by dividing the number of flips by the number of updated neurons. As described in the task requirements, for a low number of patterns many networks are simulated and averaged over all of these. The practical

error probability is compared against the theoretical curve found in the first section. The result is displayed below:

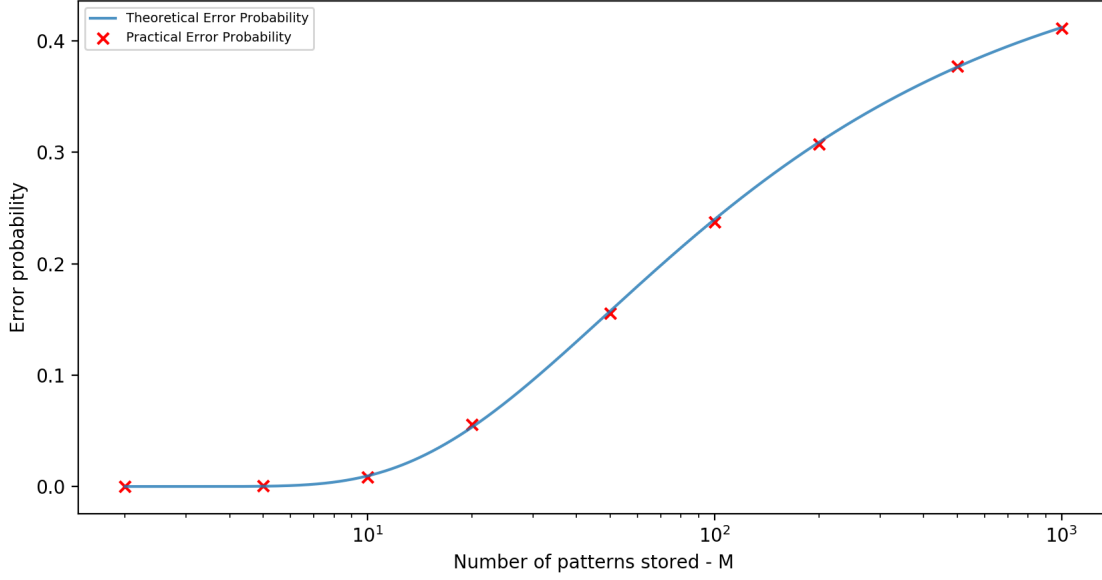


Figure 2: Error Probability vs Number of Patters Stored for a network with $N = 100$ neurons

As observed above the theoretical calculations performed in the previous section matches very well the practical simulation.

1.3 Simulated Network with noise

In this section, a Hopfield Network was simulated with $N = 100$ neurons. The difference from the previous task is that now, the system is initialised in a stored pattern which is corrupted with noise(i.e. some bits are flipped). The degree of this noise is measured by the probability p of a bit flipping. The purpose of this section is to observe if the network converges to the right stored pattenr or if it would converge to a different pattern. The system evolution was described in *Equation 5* in the previous section. The error probability for different levels of noise and a different number of stored patterns can be observed in the figure below:

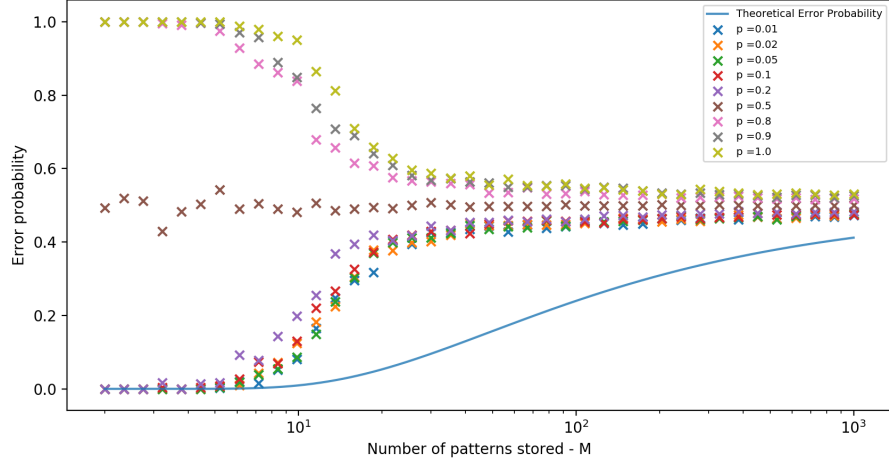


Figure 3: Error Probability as a function of the number of patterns stored with different noise levels

As expected the error probability is different from the theoretical simulation in the previous section. The reason is for this is the fact that the initial state is not a stored pattern but rather a noisy version of it. However, it can be observed that as the noise level increases ($p < 0.5$) the error probability increases as well. This is because the more the initial point differs from the true attractor, the more it is likely to converge to a different attractor.

The other source of errors are called *spurious states*. One of the spurious states appears when the number of pattern M increases. This introduces some local minima that are not correlated to any linear combination of stored patterns. Therefore the algorithm could converge to one of these local minima points. An example of these kinds of states can be observed in the figure below:

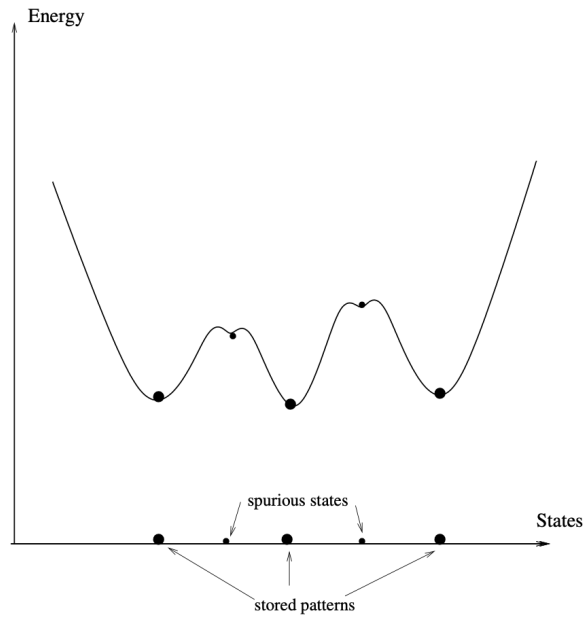


Figure 4: Graphical Visualisation of local minima points that affect the Hopfield Network [1]

As observed in the figure above where the energy (or Lyapunov) function is used to track the converges of the algorithm, these local minima can cause interference. A way to avoid these could be to introduce some form of a stochastic model of the Hopfield Network such as Evolution Strategies. The other form a *spurious states* is identified and described in the paper by Hopfield[2]. This consists of having the inverse of stable pattern becoming itself a stable pattern. For example, for the pattern stored 110001, its inverse 100011 becomes also a stable state. This is reason why the mirror effect can be observed once $p > 0.5$. All of the initial points will which have noise level $p > 0.5$ will try converge to this *spurious state*. Further evidence can be provided by looking at a mathematical explanation. Consider the probability of error P_{error} , which is measured by looking at how many bits are different compared to original stored patterns. This can rewritten in term of the error probability with respect to its inverse P_{error}^{-1} :

$$\begin{aligned} P_{\text{error}} &= \frac{n_{\text{incorrect}}}{\text{Total Number of Bits}} \\ &= \frac{\text{Total Number of Bits} - n_{\text{incorrect}}^{-1}}{\text{Total Number of Bits}} \\ &= 1 - P_{\text{error}}^{-1} \end{aligned} \quad (6)$$

When $p = 0.5$, where half of the bits from the initial pattern are flipped and therefore the point is at the equal distance between the original initial pattern and its inverse. This removes any information from the actual pattern and the network performance is equivalent to a coin toss.

2 Exploration of a physiological model of a spiking neuron

The second part of the assignment consists of investigating **Hodgkin-Huxley** model of the action potential. This model is governed by the following differential equations:

$$\begin{aligned} \dot{v} &= -g_{\text{Na}}m^3h(v - e_{\text{Na}}) - g_{\text{K}}n^4(v - e_{\text{K}}) - g_{\text{L}}(v - e_{\text{L}}) + I_{\text{ext}} \\ \dot{m} &= \alpha_m(v)(1 - m) - \beta_m(v) \\ \dot{h} &= \alpha_h(v)(1 - h) - \beta_h(v) \\ \dot{n} &= \alpha_n(v)(1 - n) - \beta_n(v) \end{aligned} \quad (7)$$

Where the functions $\alpha_m, \beta_m, \alpha_h, \beta_h, \alpha_n$ and β_n are given in a MATLAB file and are displayed below:

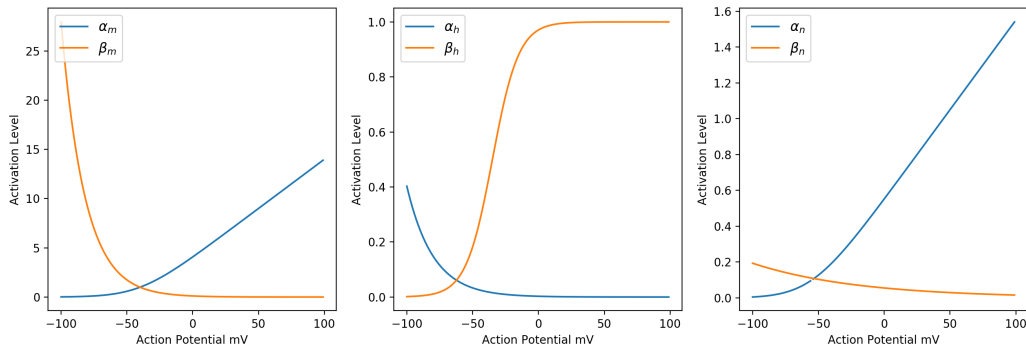


Figure 5: $\alpha_m, \beta_m, \alpha_h, \beta_h, \alpha_n$ and β_n Functions

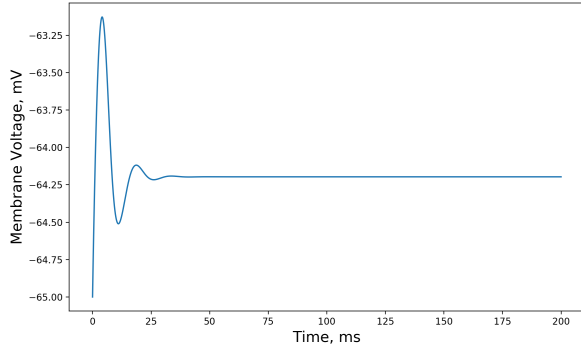
In order to simulate this model Euler method is used to predict the values for v, m, h and n in the next timestep $\Delta t = 0.001\text{ms}$:

$$\begin{aligned}
v[t + \Delta t] &= v[t] + \Delta t \times (-g_{\text{Na}}m[t]^3h[t](v[t] - e_{\text{Na}}) - g_{\text{K}}n[t]^4(v[t] - e_{\text{K}}) - g_{\text{L}}(v[t] - e_{\text{L}}) + I_{\text{ext}}) \\
m[t + \Delta t] &= m[t] + \Delta t \times (\alpha_m(v[t])(1 - m[t]) - \beta_m(v[t])) \\
h[t + \Delta t] &= h[t] + \Delta t \times (\alpha_h(v[t])(1 - h[t]) - \beta_h(v[t])) \\
n[t + \Delta t] &= n[t] + \Delta t \times (\alpha_n(v[t])(1 - n[t]) - \beta_n(v[t]))
\end{aligned} \tag{8}$$

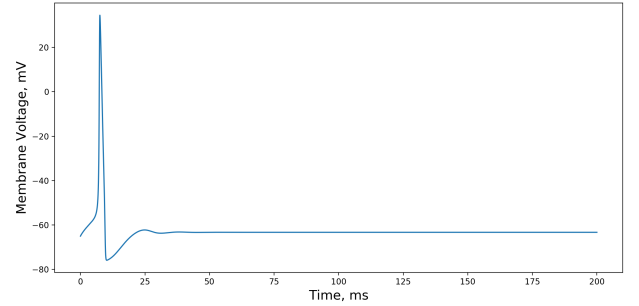
Therefore using I_{ext} as input the **Hodgkin-Huxley** model of the action potential can be simulated

2.1 Input from a single square pulse

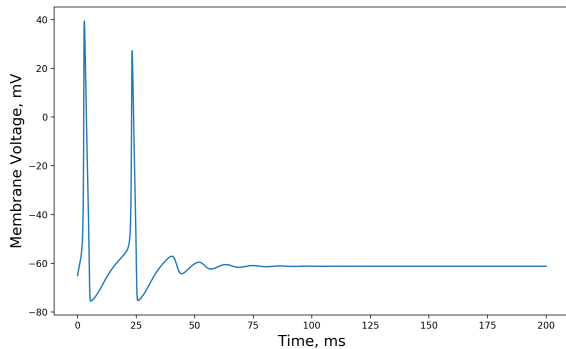
In the beginning, it will be examined how the model responds to a relatively small amplitude of $I_{\text{ext}} = 1\text{mA/nF}$. The results can be observed in *Figure 6a* below. It can be observed that at such a low level of current the membrane potential oscillates with an initial transient before it settles at the equilibrium point after $T \approx 30\text{ms}$.



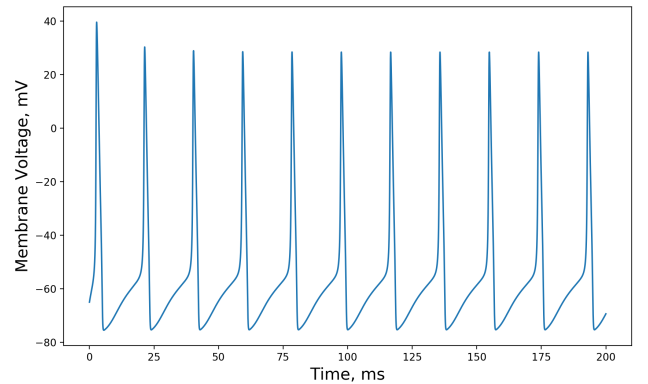
(a) Neuron Membrane Potential Evolution
 $I_{\text{ext}} = 1.0 \text{ mA/nF}$



(b) Neuron Membrane Potential Evolution
 $I_{\text{ext}} = 2.3 \text{ mA/nF}$



(c) Neuron Membrane Potential Evolution
 $I_{\text{ext}} = 6.0 \text{ mA/nF}$



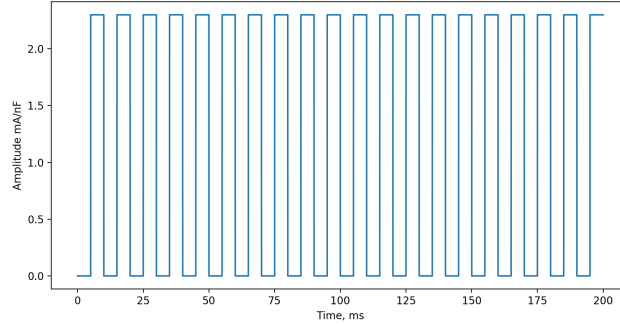
(d) Neuron Membrane Potential Evolution
 $I_{\text{ext}} = 6.3 \text{ mA/nF}$

Figure 6: **Hodgkin-Huxley** model simulation results

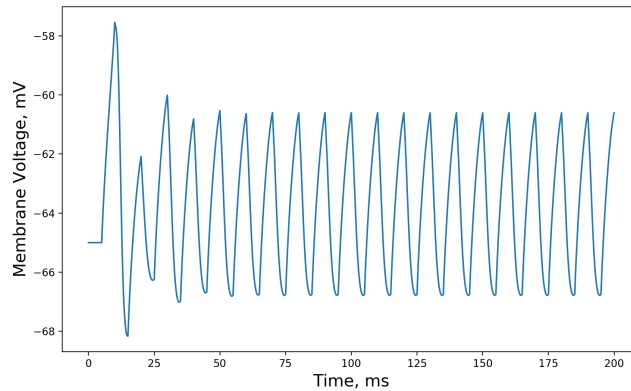
By increasing the current to a level equal to $I_{\text{ext}} = 2.3\text{mA/nF}$, it is high enough such that the membrane potential outputs a spike. This can be observed in *Figure 6b* above. As the amplitude is taken to a higher level, $I_{\text{ext}} = 6.0\text{mA/nF}$, the number of spikes emitted by the membrane potential is increased, see *Figure 6c* above. However it is observed that even at this amplitude the membrane potential will settle around the equilibrium point after the spikes are emitted. It is observed that a current level of $I_{\text{ext}} = 6.3\text{mA/nF}$, see *Figure 6d* will cause the membrane to continuously emit spikes without the membrane potential returning to the equilibrium region. As current level is increased beyond this point the frequency in which spikes are emitted will increase. As a result it can be said that using this type of input the **Hodgkin-Huxley** operates in three regimes: no action potential regime ($0 < I_{\text{ext}} < 2.3\text{mA/nF}$), single spike regime ($2.3 < I_{\text{ext}} < 6\text{mA/nF}$) and periodic spiking regime ($I_{\text{ext}} > 6\text{mA/nF}$)[3].

2.2 Input from a periodic square pulse

In this section, the **Hodgkin-Huxley** model is further investigated but this time the applied current (I_{ext}) is a periodic square pulse. The pulse can be described by three variables: pulse width p , amplitude I and period T . All of these variables can be varied. It is first looked at how the model varies with T . To start with, a pulse with $p = 5\text{ms}$, $I = 2.3\text{mA/nF}$ and $T = 10\text{ms}$ is simulated and applied to the **Hodgkin-Huxley** model. The results can be observed in the figure below:



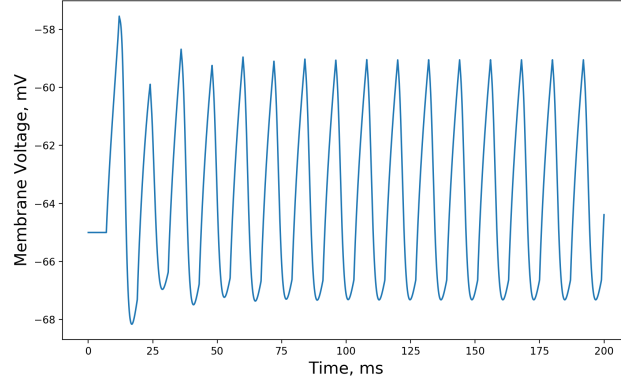
(a) Input Periodic Square Pulse with $p = 5\text{ms}$, $I = 2.3\text{mA/nF}$ and $T = 10\text{ms}$



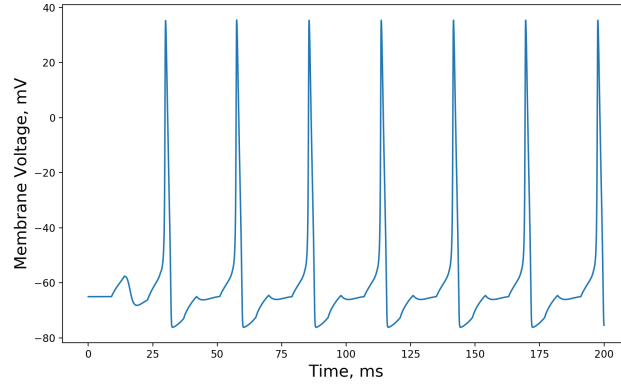
(b) Neuron Membrane Potential Evolution when a periodic pulse is used as input

Figure 7: **Hodgkin-Huxley** model simulation results with a periodic pulse

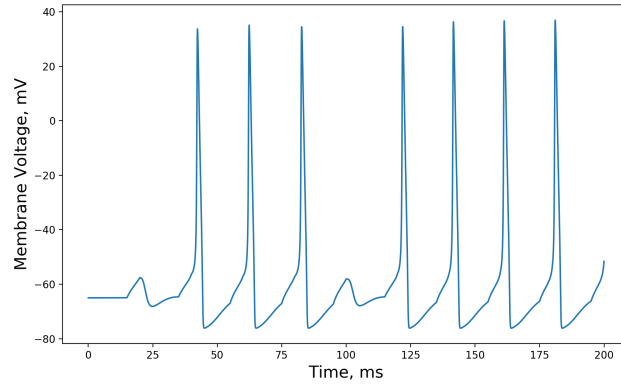
It is observed that even if the pulse amplitude I is above the threshold found in the previous section $I = 2.3\text{mA/nF}$, the model fails to elicit a spike. Therefore it can be interpreted that the model is sensitive to the period of the input pulse. To test this theory the period is varied from $T = 10\text{ms}$ to $T = 20\text{ms}$. In the figures below some of the relevant results of the membrane potential are displayed:



(a) Neuron Membrane Potential Evolution, $T = 12\text{ms}$



(b) Neuron Membrane Potential Evolution, $T = 14\text{ms}$

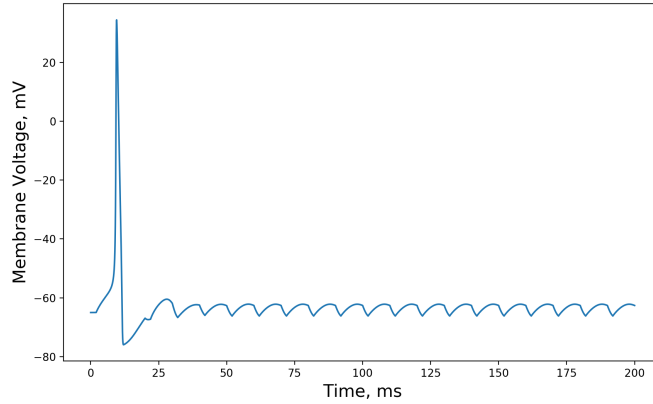


(c) Neuron Membrane Potential Evolution, $T = 20\text{ms}$

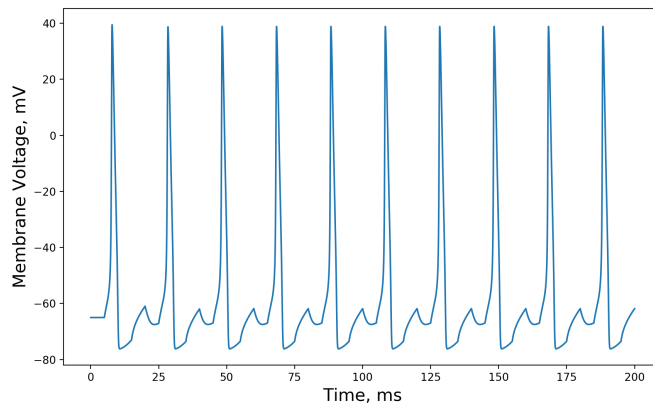
Figure 8: **Hodgkin-Huxley** model simulation results with a periodic pulse

In the previous section, it was observed how the **Hodgkin-Huxley** has an oscillatory behaviour when a step input has been applied. Therefore, looking at the figures above it can be observed how the period of input pulse affects the membrane potential. This all due to constructive and destructive interference. In *Figure 8b* it has been observed that spikes are emitted. This due to the fact that when $T = 14\text{ms}$ the square input pulse from I_{ext} appear at the exact moment when the membrane potential is in the positive cycle from the previous pulse and as a result, it will interfere constructively. However, it is observed that in *Figure 8a* the exact opposite effect is happening. The pulse from I_{ext} appears at the moment when the membrane potential is in the negative cycle and causing destructing interference. By further increasing the period T in can observed that the frequency of the spikes in the membrane potential is increasing, *Figure 8c*. This is due to the fact that T gets closer to the resonance frequency.

Now, that this temporal dependency has been studied it is time to see how the model is responding to the periodic pulse when the other variables are varied. A non-resonant period of oscillation is selected, $T = 10\text{ ms}$, and then the other variables are increased one at a time. The results are displayed in the figures below:



(a) Neuron Membrane Potential Evolution, $p = 8\text{ms}$



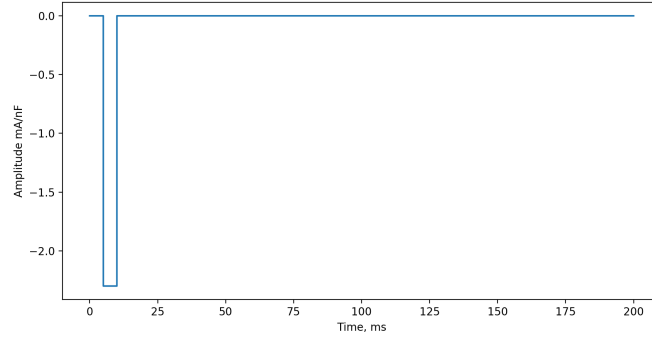
(b) Neuron Membrane Potential Evolution, $I = 6\text{mA/nF}$

Figure 9: **Hodgkin-Huxley** model simulation results with a periodic pulse

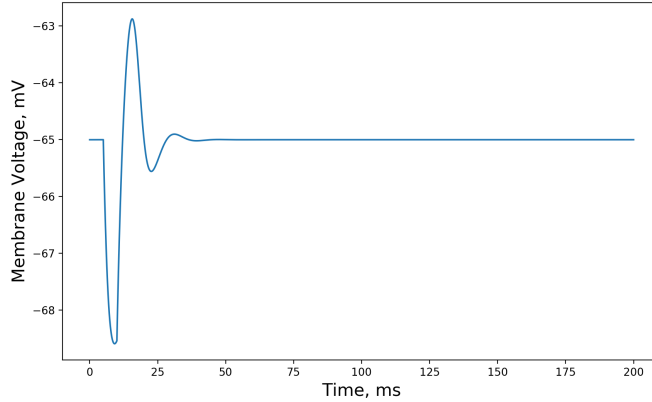
From *Figure 9a* it can be observed that when p increases the result becomes similar to the one found in the previous section, *Figure 6b*. This is what it was expected because, as p increases the

input becomes more and more similar to a step input. If instead, the amplitude I is increased *Figure 9b* it can be observed that a spike is emitted at each and single input pulse. This is because at each pulse the membrane potential is above the threshold and as a result, a spike is emitted. It can be said that by varying the other variables, the temporal dependency of the model is removed.

After observing how varying the parameters of the input pulse affects **Hodgkin-Huxley** model it now time to see how it responds to an inhibitory stimulation. A single negative square pulse is simulated at $I = -2.3\text{mA/nF}$ and applied to the model. The results are observed below:



(a) Single Inhibitory stimulation



(b) Neuron Membrane Potential Evolution from a single Inhibitory pulse stimulation

Figure 10: **Hodgkin-Huxley** model simulation results with a periodic pulse

As observed in *Figure 10b*, the membrane potential has the same oscillating behaviour as when an excitatory input was applied. Therefore it can deduced that if the right period is found to match the resonant frequency a spike could be emitted. However, because the positive cycle amplitude is too low compared to the negative cycle it is required to increase the amplitude of the input pulse in order to elicit a spike. A sub-threshold value of $I = -4\text{mA/nF}$ is chosen. The result is displayed in the figure below:

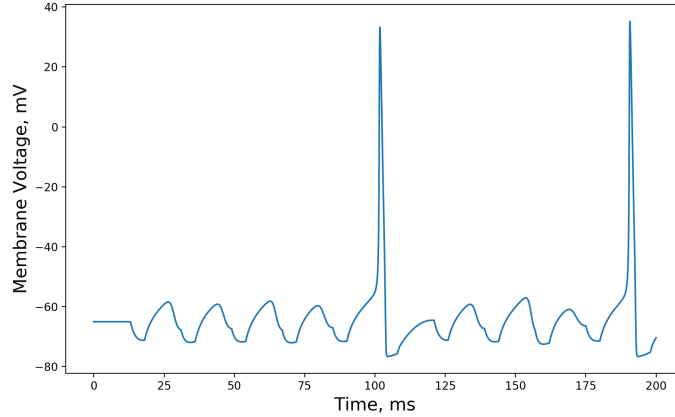


Figure 11: Neuron Membrane Potential Evolution with $T = 18\text{ms}$ and $I = -4\text{mA/nF}$

From *Figure 11* it is observed that if the period T is brought closer to the resonant frequency a spike will occur even from an inhibitory input. As a result, it can be said that the existence of post-inhibitory spikes is a salient property of **Hodgkin-Huxley** model, and it is closely related to the rebound or anodal break excitation[4]. Compared to the other model **leaky integrate-and-fire(LIF)** this seems to be a key difference. As it is known using a **LIF** model, a spike cannot be emitted from an inhibitory input. On the other hand, using a **Hodgkin-Huxley** model it was observed that it is possible to elicit a spike from an inhibitory input. The second difference and the key difference between a **rate** model and **Hodgkin-Huxley** model is the fact that both **LIF** and **rate** model prefer high frequency of the input while **Hodgkin-Huxley** model prefers input which has an input frequency closer to the resonant frequency. Regarding the efficiency of each model, it is noted how computationally inefficient **Hodgkin-Huxley** model is and as such, it might pose a problem when simulating a large network. An alternative method to counter these inefficiencies is called **resonate-and-fire** model and it is described in [4]. It shows how a **LIF** model which is computationally efficient can be modified to incorporate the oscillating behaviour found in **Hodgkin-Huxley** model.

References

- [1] Imperial College. Hopfield network. <https://www.doc.ic.ac.uk/~ae/papers/Hopfield-networks-15.pdf>. [Online; accessed 25-March-2020].
- [2] J J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 1982.
- [3] Wulfram Gerstner and Werner M. Kistler. *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge University Press, 2002.
- [4] Eugene M. Izhikevich. Resonate-and-fire neurons. *Neural Networks*, 14(6):883 – 894, 2001.