Single neuron models: from Hodgkin-Huxley to AdEx

BMNN miniproject

Flore Barde, Jeannette Gommendy
flore.barde@epfl.ch, jeannette.gommendy@epfl.ch

June 5, 2022

Contents

Introduction

Neurons are cellular structures whose complex functioning is governed by many voltage-dependent ions channels. Multiple theoretical models were developed over the years in order to model single neurons, to better understand their intrinsic mechanisms. In this report, some models for the single neuron will be investigated. As the starting point, the Hodgkin Huxley function will be implemented in order to get a spiking model. Two functions will be implemented for regular spiking (I_{Na}, I_K currents) and adaptive spiking (I_{Na}, I_K, I_M currents). From these complex Hodgkin Huxley models, parameters will then be tuned out in order to develop a computationally less expensive model: the Adaptive Exponential Integrate and Fire (AdEx) neuron model.

1 Exploration of Hodgkin Huxley neurons

The Hodgkin-Huxley main equation is given by the formula (1) :

$$C\frac{dV}{dt} = -g_l(V - E_l) - I_{Na} - I_K - I_M + I_{ext} \tag{1}$$

with C = 1 uF, g_L = 0.1 mS, E_L = -70 mV and I_{ext} the external current injected. The three ion currents I_{Na}, I_K and I_M are associated with the equations (6), (7), (8) given in the appendix section 3.1.

1.1 Getting started

Two functions describing HH neuron models for regular spiking (I_{Na}, I_K currents) and adaptive spiking (I_{Na}, I_K, I_M currents) are implemented in the Python file `Implementation_HH.py`. First of all, to get the parameter initialization of the model's variables and avoid any transient at the beginning of the stimulation, the stable points of each variable are found, by putting the external current I_{ext} at 0 (see `find_stable_pt()` function in `Implementation_HH.py`). These values are reported in the table 1 in the appendix 3.2 for both regular and adaptative neurons. Now by initializing the variables to their resting values, and stimulating the model with a step current of 2.0 μA for 90 ms, both regular and adaptive models show spiking behaviors. This is shown by the figures 1a,2a (membrane voltage, input current and ion currents as functions of time). Moreover, the parameters of the model n, m, h (and p for adaptive) are plotted as a function of time (see figures 1b,2b). All figures are presented in the appendix 3.2.

1.2 Rebound spike ?

The regular spiking neuron implemented here does not have rebound spikes. To prove it, the stimulation protocol implemented in the Python file `Rebound_spike.py` consists in varying the negative injected step current amplitude from 0 to -20 μA (figure 3a) or changing the stimulation duration from 0 to 50 ms (figure 3b). To better understand the protocol, let's recall that an HH neuron can spike if it receives a sufficiently strong depolarizing input current, as seen in the getting started section, but also usually, after a hyperpolarizing current creating a rebound spike. That is what is happening in the **Exercise 5** mentioned in the instruction paper. In our case, the neuron does not spike. To better understand the reason, the channel gating variables x_∞ and τ_x are plotted

as a function of the membrane potential V_m ($x \in \{m, n, h\}$). This is done for our simulation with V_m going from -140 mV to -70 mV, presented by the figures 4a and 4b. To get a better insight about the general behavior of the variables with a larger range of membrane potential, analytical computations are performed (see function `gate_var_analytique` in `Rebound_spike.py`), presented by the figures 5a,5b. The analytical results are also plotted for the **Exercise 5**, presented by the figures 6a, 6b. In the **Exercise 5**, the stable membrane potential is at 0 mV. When the negative current is injected, the membrane potential approaches the reversal potential of the potassium channel (-12 mV), but gets further away for the sodium channel one. The moment the injected current stops the fast variable m (associated with sodium, with lower τ_m) starts to increase faster than other gating variables (as seen on the figure 6a(increases from 0 to 50 mV)) to let sodium ions come inside the neuron cell, and due to slower activation of h channel and m channel, a spike is induced.
However, in our case the m_∞ gate only increases when V_m reaches [-60,-50] mV and not for lower value and our stable state is at -70 mV, which means than even if a negative current is injected, the moment it stops the membrane potential will always return to its resting value (-70 mV) and not spike because the gating variables always stay at their stable value until the membrane potential reaches approximately [-60,-50] mV. Indeed the fast variable m_∞ has a value of 0 below -60 mV (see figure 5a). All figures are presented in the appendix 3.3.

1.3 Adaptation

Using the adaptive implementation, the neuron is stimulated for 1500 ms with a step current of 2.0 μA. The resulting voltage trace, ionic currents and gating variables are plotted, presented by the figures 7a, 7b. As observed in those figures, it seems like the time difference between two spikes increases with time for the adaptive neuron model until it reaches a stable value. Indeed this is due to the new channel I_M; this channel allows for positive ions to cross the membrane and leave the neuron taking a positive charge out, just like the I_K channel. Therefore, as the I_K, and I_M channels are open, they allow for more K^+ ions to go out (their goal is to reach -90 mV for the potential, because E_K = -90 mV) reducing the number of spikes in a certain amount of time. The additional I_M channel allows for a prolonged refractory period after the spike.
I_M takes more time to open than the two other ion channels and can provide a much more low current (fewer number of ions go through it) as observed on the figure 7a. For the gating variable p, it also shows the same behavior, observed on the figure 7b, as it can only go to the maximum value of 0.4 (the ion channel I_M cannot be fully opened), creating a low ion current I_M. Moreover, p does not oscillate as much as the other gating variables, it stays approximately at the same value during the stimulation time, meaning the I_M channel stays always partially opened during the stimulation. Those information indicate that the channel I_M and its gating variable p allow for the passage of positive ions from the neuron to the extracellular space, but at a relatively slow rate, allowing with the help of other gating variables (n) to increase the spike timing.
After the implementation of a function extracting the spike timing of the voltage trace (`spike_timings` in Python file `adaptation.py`), the adaptive figure is created, presented by the figure 8. As explained before, the adaptive neuron takes some time to reach a stable spike timing, therefore the figure chosen to exhibit this behavior represents Δ_t (time difference between two spikes) as a function of the stimulation time. With the adaptive model implemented, Δ_t increases with time until it reaches a stable value, as already observed on the adaptive figures.

Now, the parameters I_M, p_∞, τ_p are modified in order to do the following modifications :

Slow down the adaption rate :
(9a,9b) : Here, the goal is for the neuron to take more time to reach its stable firing rate. If the relaxation time constant τ_p is increased, then the ion channel I_M will take more time to react and open to its maximum, leading to a slower increasing rate of the time difference between two spikes Δ_t. In the end the stable firing rate remains the same.

Decrease the stable firing rate :
(10a,10b) : If the value of p_∞ is decreased, then the gate variable p would take a lower stable value, meaning the I_M channel will let fewer ions go through when it reaches its stable state, leading to a lower time difference between two spikes Δ_t (it is equivalent to change I_M : equation (8a) : if I_M decreases, then the ion current through the membrane decreases, less positive ions go from inside to outside the cell : lower Δ_t cause V_m increases faster)

Reverse the adaptation :
(11a,11b) : In order to reverse the adaptation, changing the sign of p_∞ or I_M leads to same results because of the formula (8a). If I_M changes its sign, then the positive ions will flow from outside to inside the cell (reverse situation) meaning the ion channel I_M will now act like the I_{Na} channel and help for the initiation of an action potential, leading to a lower Δ_t.
All figures are presented in the appendix 3.4.

2 From HH to AdEx

In this part, the model HH adaptive neuron developed previously is considered as the ground truth of a biological neuron. Indeed, stimulation protocols are going to be performed on this neuron, as experimentalists would do, in order to extract the parameters necessary for the development of the AdEx neuron model. The AdEx model being described by the following two coupled equations:

$$C\frac{dV}{dt} = f(V) = -g_l(V - E_l) + g_l\Delta_T e^{\frac{V-\theta_{rh}}{\Delta_T}} - w + I_{ext} \tag{2}$$

$$\tau_w\frac{dw}{dt} = a(V - E_l) - w \tag{3}$$

where θ_{rh} is the rhéobase voltage. It is an exponential Leaky Integrate and Fire (eLIF) model, combined with an adaptation current w.

2.1 Passive properties

First of all, the interest was set on determining the passive membrane parameters. In order to do so, different stimulation protocols were performed on the HH adaptative neuron as shown in the Python file `Passive_properties.py`. All the parameters found are reported in table 2. To begin with, one could notice that due to the exponential component of the eLIF model visible in equation (2), the membrane potential of the neuron only diverges when it crosses a certain value θ_{rh} : the rhéobase voltage. Therefore, when the membrane potential is smaller then this value, equation (2) can simply be reduced to :

$$C\frac{dV}{dt} = -g_l(V - E_l) + I_{ext}. \tag{4}$$

Parameter E_l: In order to determine the parameter E_l corresponding to the potential of the leaky channel, the behavior of the neuron without any input current was observed. In this case, the expression 4 reduces to : $C\frac{dV}{dt} = -g_l(V - E_l)$. Therefore, when $V = E_l$, one gets $\frac{dV}{dt} = 0$, meaning that the membrane potential is constant. Thus, E_l corresponds to the value of the membrane potential when it is constant. By looking at figure 12a, the value $E_l = -70.6073$ mV was found.

Parameter R: The next parameter to find is the resistance of the membrane R. The stimulation protocol of the membrane was the following: a step current was injected (intensity small enough for the neuron to not spike). Thus the resistance of the membrane could be computed using the following equation: $R = \frac{V_{max} - V_{rest}}{I_{injected}}$. Where V_{max} is the maximal value of membrane potential reached and V_{rest} the potential of the neuron at rest, corresponding to the value of parameter E_l. The value $R = 9.4963$ kΩ was found.

Parameter g_l: Once the membrane resistance was determined, the parameter g_l was found throught the relation : $g_l = \frac{1}{R}$. The value $g_l = 105.3043\mu$S was found.

Parameter τ_m: The time constant of the membrane τ_m can be found by stimulating the neuron with a step current (still small enough for the neuron not to fire). It is defined as the time it takes for the membrane potential of the neuron to reach 63% of it's final value. For the model presented here, the membrane potential corresponding to 63% of the maximal value was $V_{63\%} = 64.6281$ mV and it took a time $\tau_m = 9.0$ ms to reach this value, as visible on figure 13.

Parameter C: In order to determine the value of the membrane capacity C, one can use the values of the previously found parameters of membrane resistance R and membrane time constant τ_m. Indeed, the three parameters are related through the following equation: $C = \frac{\tau_m}{R}$. The value $C = 0.9477\mu$F was found.

All figures of the stimulation protocols are presented in the appendix 3.5.

2.2 Exponential Integrate and Fire

In order to simulate an AdEx model of a neuron, it is now necessary to determine the parameters θ_{rh} and Δ_T involved in the exponential part of the equation (2) describing the model. This was done in the Python file `Exponential_Integrate_and_Fire.py`.

Parameter θ_{rh}: To determine the value of the rhéobase voltage θ_{rh}, the behavior of the membrane dynamics $f(V)$ was studied as a function of V without any adaptation current w. The membrane dynamics $C\frac{dV}{dt} = f(V)$ corresponds to the membrane current. A step current of intensity $I_{ext} = 1.5$ μA was applied to the HH adaptative neuron in order for its membrane dynamics minimum to be slightly above zero, as visible in figure 14a. In this case, the membrane dynamics doesn't have a stable point anymore (points where $f(V) = 0$) and thus, the adaptative neuron is able to fire. As visible on the figure 14a, the membrane dynamics firstly displays a linear behavior. However, after crossing a certain voltage value, an exponential behavior starts: it represents the action potential. By looking at equation (2), one can notice that the membrane voltage value corresponding to this change of behavior is the rhéobase voltage θ_{rh}. Therefore, θ_{rh} corresponds to the minimum of the curve presented in figure 14a. The value $\theta_{rh} = -55.7554$ mV was found.

One can notice that the simulation was stopped before the neuron began to spike. Indeed one is not interested in the reset mechanism of the neuron after the action potential, because it induces the hyperpolarization of the neuron, which would lead to negative values of $f(V)$.

Parameter V_S: This parameter represents the maximal voltage reached without making the neuron spike. It is found by stimulating the neuron with a very short current pulse, and increasing slowly the current amplitude until the threshold for firing of the neuron is reached. Injecting small pulse currents to the neuron allows to modify the initial conditions of the neuron: the initial membrane voltage to which the neuron is set. However, with pulse current, even if the initial membrane voltage is modified it doesn't shift the membrane dynamics curve (no modification of the stable points value), contrary to step current (which shifts the membrane dynamics curve upward). This allows to set the initial membrane potential to a value above the first stable point E_l, and to observe the moment it will go higher then the second one V_S. Thus, for a short pulse current of $I_{ext} = 10.6422\ \mu A$, the membrane potential increases but is right below the spiking threshold. The corresponding maximal voltage value reached was $V_S = -51.2638$ mV.

Parameter Δ_T: Now that V_S is found, Δ_T can be determined by solving $f(V_S) = 0$. The value $\Delta_T = 1.9633$ ms was obtained.

Overall, all the parameters found allowed to plot the membrane dynamics $f(V)$, as visible in figure 14b. All figures are presented in the appendix 3.6.

2.3 Subthreshold adaptation

In order to find the coefficient a from the equation (2), the voltage dynamics are considered slow and far from threshold, leading to the three conditions : $dV/dt \approx 0$, $dw/dt \approx 0$ and $\exp\left\{\frac{V-\theta_{rh}}{\Delta_T}\right\} \approx 0$. This leads to the dynamics presented by the equation (5) :

$$\begin{cases} -g_L(V - E_L) - w + I_{ext} = 0 \\ \\ w = a(V - E_L) \end{cases} \implies V = E_L + \frac{I_{ext}}{gl + a} \tag{5}$$

In order to find a, the slope $1/(g_L + a)$ from (5) needs to be found. This is done by plotting the $I - V$ curve of a neuron stimulated by a 10 s current ramp from 0.0 μA to 1.2 μA, and then extracting the slope s of the curve, with $a = 1/s - g_L$. See the function `plotI_V() in Subthreshold_adaptation.py` for more details of computation.
The $I - V$ curve is presented in figure 15 in appendix 3.7, and the value a equals to 10.6559 μS.

2.4 Remaining parameters

In order to find an AdEx model that fully replicates the HH adaptative neuron, one still needs to find the remaining parameters b, τ_w and V_{reset}. These parameters where determined by trying to match the spike trains of the voltage traces, as well as the adaptation behavior curves of both the HH adaptative neuron and AdEx model, as visible in figures 16a and 16b. These figures were produced using the file `Remaining_parameters.py`.
Each parameter had a different effect. Indeed, V_{reset} is the voltage to which the neuron is reset right after spiking. Therefore, it determines the firing rate at the beginning of the simulation, and shifts

the curve of adaptation behavior up or down. If V_{reset} is more negative, it will take more time for the membrane voltage of the neuron to reach again the rhéobase voltage (threshold for firing). The value of τ_w has an influence on the value of the stable firing rate, without changing the adaptation rate of the neuron. Finally, b changes the adaptation rate of the neuron. The higher is b, the more time it will take for the neuron to reach its stable firing rate. The parameter's values found for the AdEx model are all reported in table 3. All figures are presented in the appendix 3.8.

2.5 Testing on random input

Now that all the parameters for building the AdEx neuron model have been found, it is time to evaluate the accuracy of this model when replicating the behavior of an HH adaptative neuron, with random input. In order to do so, both models were stimulated with a gaussian random current $I_{ext} \sim \mathcal{N}(\mu = 1\mu A, \sigma = 15\mu A)$ for 500 ms and 2500 ms as visble in the Python file `Random_input.py`. The voltage trace of HH adaptative neuron and AdEx neuron models, with such an input current are presented in figures 17 and 18. The spikes of both models are also marked on these figures. The threshold for counting a spike was set to -40 mV for both models.
In a general manner, one can observe that the AdEx model replicates quite well the voltage trace of the HH adaptative neuron. However, even if most of the spikes of the HH adaptative neuron are also replicated by the AdEx model, some of the spikes are missed. Indeed, for a stimulation of 500 ms, the HH adaptative neuron performed 40 spikes, while the AdEx neuron only had 44. It is the same for the stimulation of 2500 ms, where the HH adaptative neuron had 21 more spikes than the AdEx one (174 and 153 spikes respectively). Moreover, by looking closely at the figures, one can notice that sometimes the AdEx neuron produces two spikes, very close in time, when the HH adaptative neuron only produces one. Thus, even though the general accuracy of the AdEx model is good, and allows for a smaller computational cost, some errors persist. These differences with the original HH adaptative neuron could be due to some inaccuracies when determining the different parameters for the AdEx model.

Conclusion

During this report, the regular spiking Hodgkin Huxley neuron model is firstly implemented. Then, an additional channel I_M is brought to the model, leading to the adaptive Hodgkin Huxley neuron model. In order to understand the dynamics and differences between the two models, several stimulation protocols are conducted and some model's parameters modified. This reveals that the addition of a channel to a model can significantly change the spike timing behavior.
The adaptive HH neuron model is then considered as the ground truth of a biological neuron. Nevertheless, the model is complex and computationaly very expensive. To face this issue, stimulation protocol are performed on the HH neuron model in order to find the parameters of a less complex model : the AdEx neuron model. Finally, the adaptive HH and AdEx models are compared by looking at their spike timing behavior when exposed to a random input current (Gaussian random current). Despite a few errors, AdEx neuron model is able to reproduce most of the adaptative HH model behavior, with less computational cost.

3 Appendix

3.1 Equations for the ion currents

Equations for I_{Na} :

$$I_{Na} = \bar{g_{Na}} m^3 h (V - E_{Na}) \tag{6a}$$
$$dm/dt = \alpha_m (1 - m) - \beta_m m \tag{6b}$$
$$dh/dt = \alpha_h (1 - h) - \beta_h h \tag{6c}$$
$$\alpha_m = -\frac{0.32(V + 47)}{exp(-0.25(V + 47)) - 1} \tag{6d}$$
$$\beta_m = \frac{0.28(V + 20)}{exp(0.2(V + 20)) - 1} \tag{6e}$$
$$\alpha_h = 0.128 exp(-(V + 43)/18) \tag{6f}$$
$$\beta_h = \frac{4}{exp(-0.2(V + 20)) + 1)} \tag{6g}$$

with $\bar{g_{Na}} = 50$ mS, $E_{Na} = 50$ mV.

Equations for I_K :

$$I_K = \bar{g_K} n^4 (V - E_K) \tag{7a}$$
$$dn/dt = \alpha_n (1 - n) - \beta_n n \tag{7b}$$
$$\alpha_n = -\frac{0.032(V + 45)}{exp(-0.2(V + 45)) - 1} \tag{7c}$$
$$\beta_n = 0.5 exp(-(V + 50)/40) \tag{7d}$$

with $\bar{g_K} = 5$ mS, $E_K =$ -90 mV.

Equations for I_M :

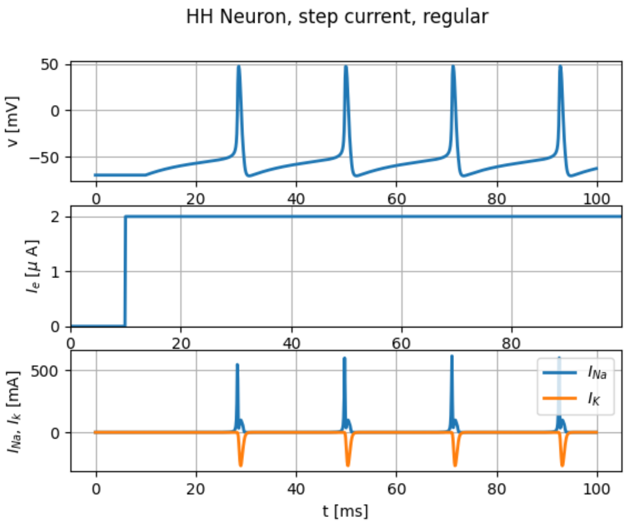
$$I_M = \bar{g_M} (V - E_K) \tag{8a}$$
$$dp/dt = \frac{p - p_\infty}{\tau_p} \tag{8b}$$
$$p_\infty = \frac{1}{exp(-0.1(V + 40)) + 1} \tag{8c}$$
$$\tau_p = \frac{2000}{3.3 exp((V + 20)/20) + exp(-(V + 20)/20)} \tag{8d}$$

with $\bar{g_M} = 0.07$ mS.

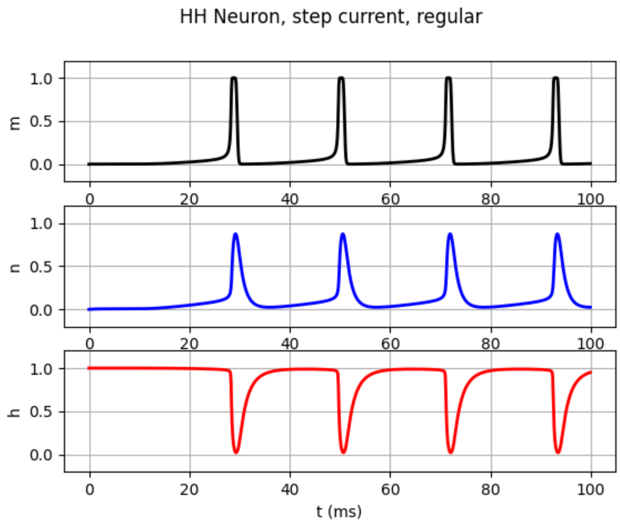
3.2 Getting started: Regular HH neuron

Variable stable points	V_m [mV]	m	h	n	p
Regular	-70	0	1	0	/
Adaptive	-70.60737	0	1	0	0.05

Table 1: Parameters initialization for the variables of the different neuron models, found with the external injected current at 0.

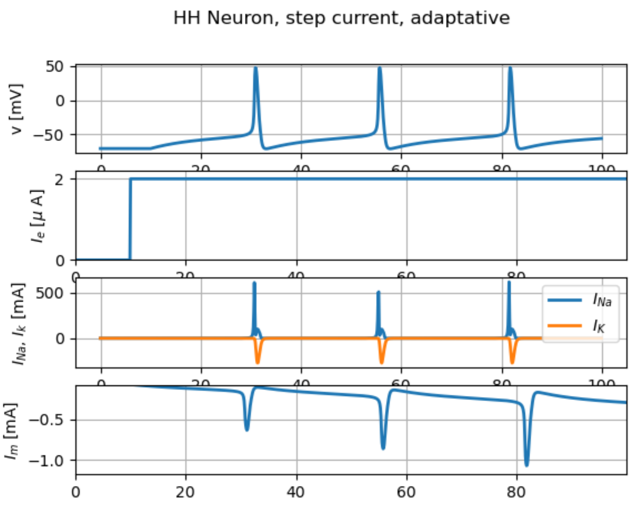


(a) Membrane voltage: V_m, Currents: I_{ext}, I_{Na}, I_K

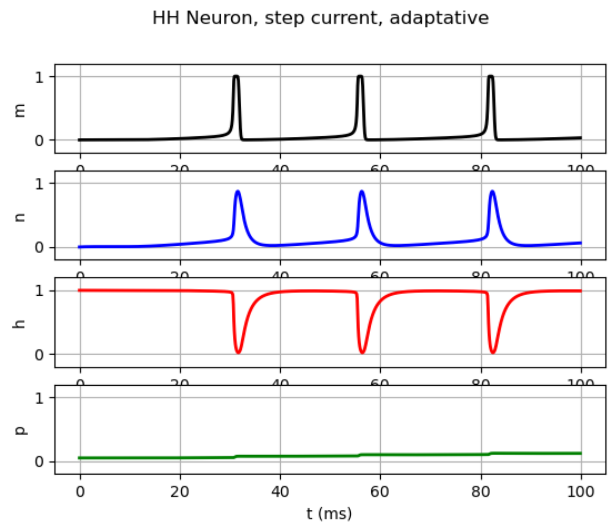


(b) Channel variables: m, n, h

Figure 1: Regular HH neuron model: stimulation protocol of 2.0 μA current from 10 ms to 100 ms.

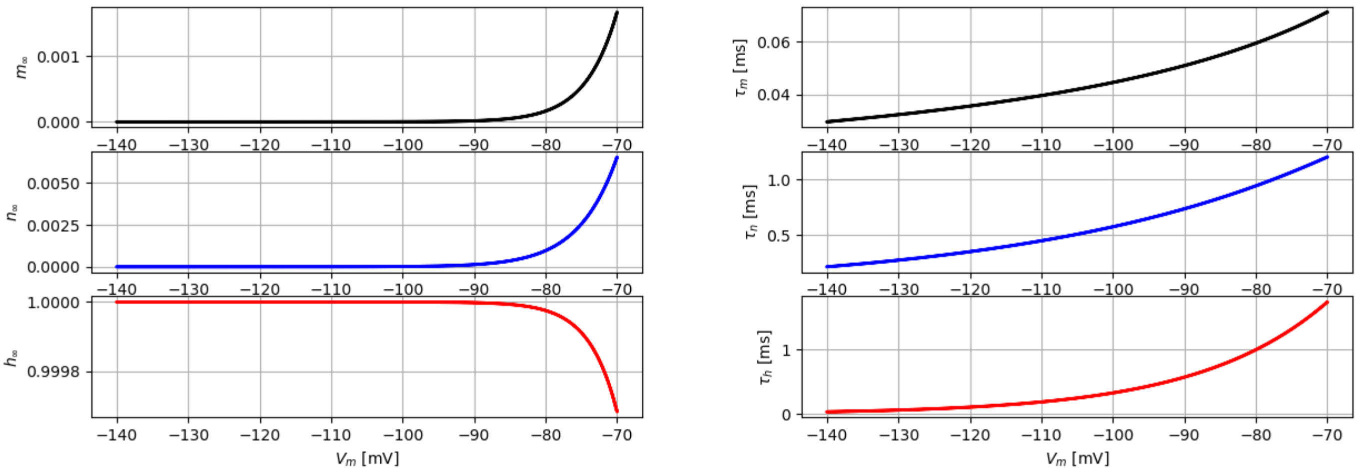


(a) Membrane voltage: V_m, Currents: I_{ext}, I_{Na}, I_K, I_m



(b) Channel variables: m, n, h, p

Figure 2: Adaptive HH neuron model: stimulation protocol of 2.0 μA current from 10 ms to 100 ms.

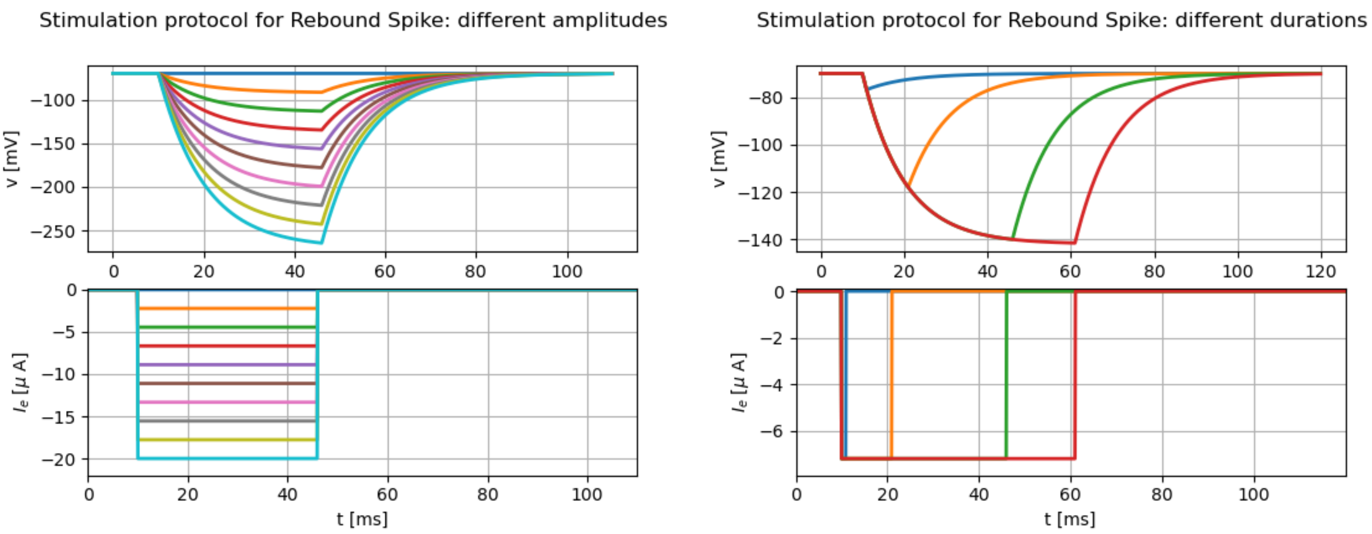


(a) Variable at infinity as a function of the membrane potential V_m

(b) Time constant for the variables as a function of the membrane potential V_m

Figure 4: Our simulation

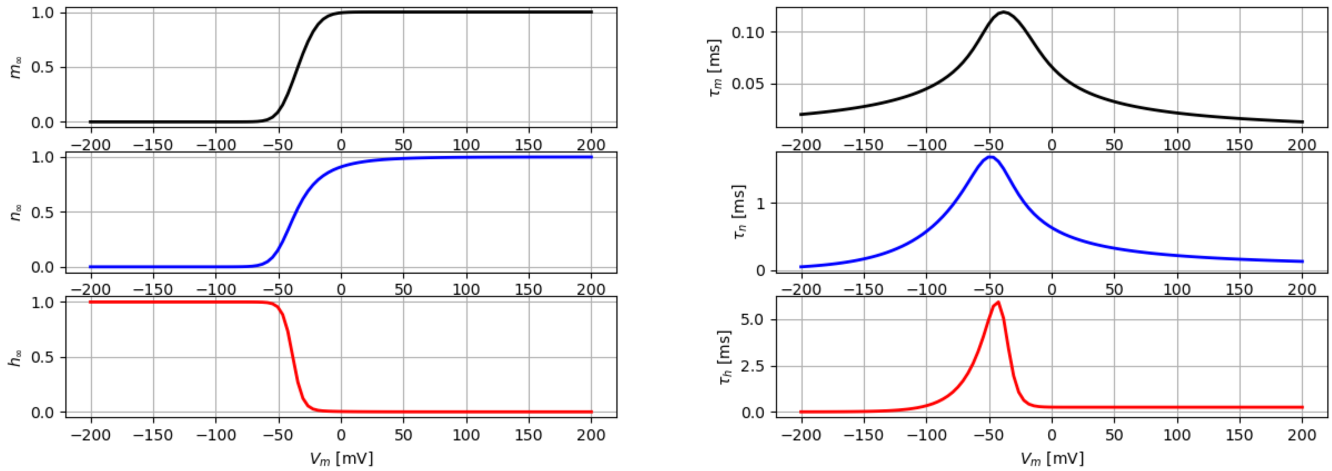
3.3 Rebound spike



(a) different amplitudes of current input I_{ext}

(b) different stimulation time for $I_{ext} = -7$ uA

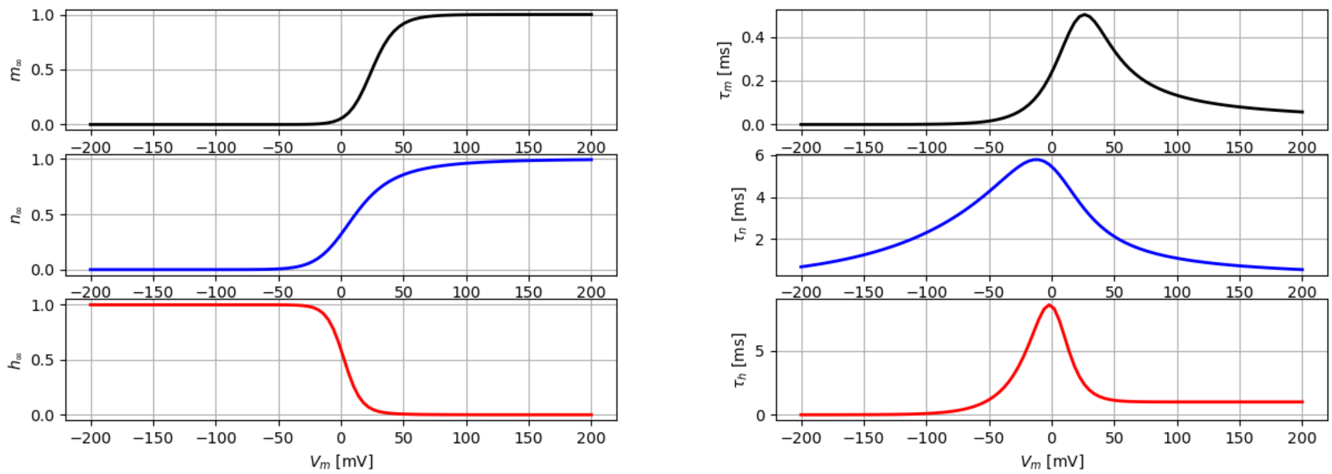
Figure 3: Stimulation protocol to study rebound spike in the regular neuron model



(a) Variable at infinity as a function of the membrane potential V_m

(b) Time constant for the variables as a function of the membrane potential V_m

Figure 5: Analytical computation

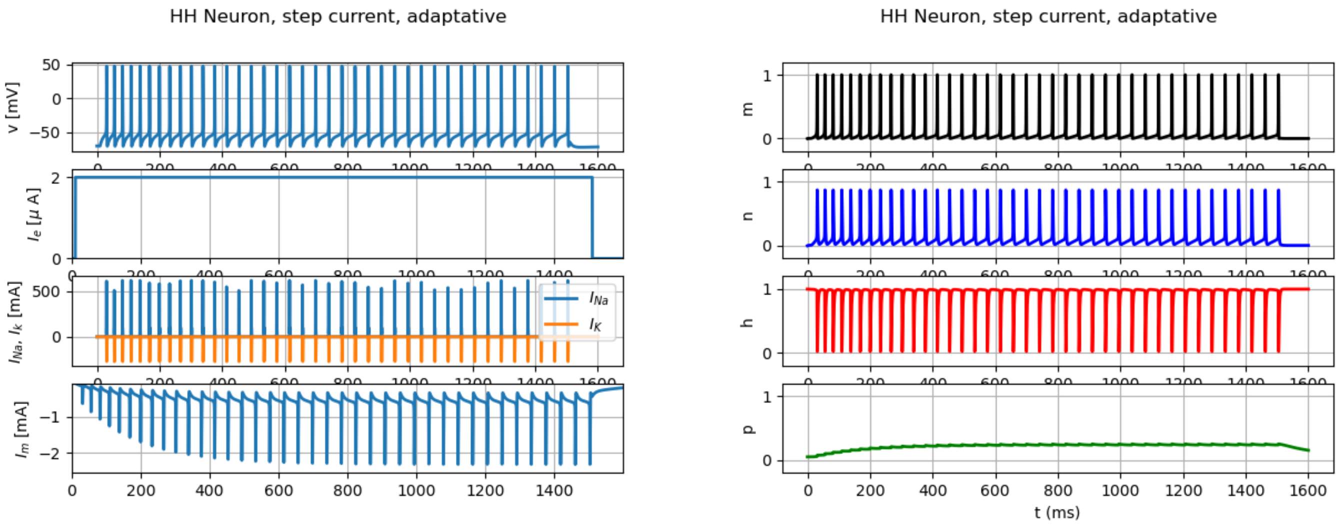


(a) Variable at infinity as a function of the membrane potential V_m

(b) Time constant for the variables as a function of the membrane potential V_m

Figure 6: Analytical computation for the exercise 5

3.4 Adaptation



(a) Membrane voltage: V_m, Currents: I_{ext}, I_{Na}, I_K, I_m

(b) Channel variables: m, n, h,p

Figure 7: Adaptative HH neuron model: stimulation protocol of 2.0 μA current from 0 ms to 1500 ms.

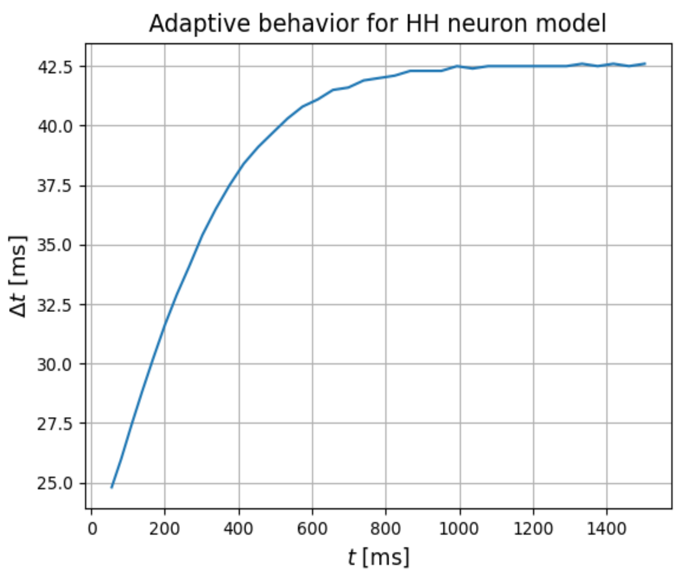
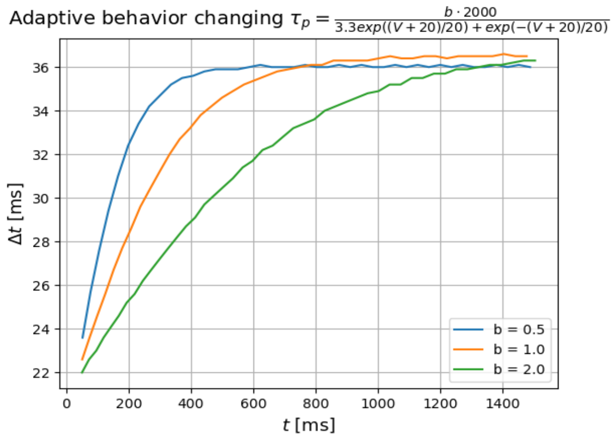
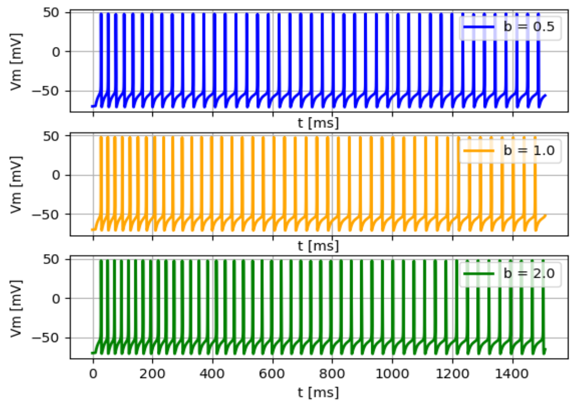


Figure 8: time difference between two spikes as a function of time for the HH neuron model.

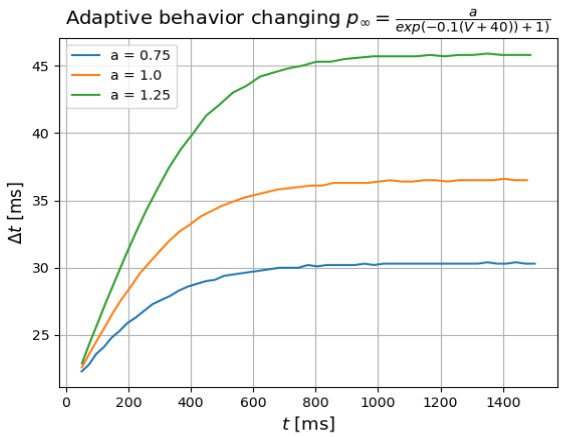


(a) Adaption figure

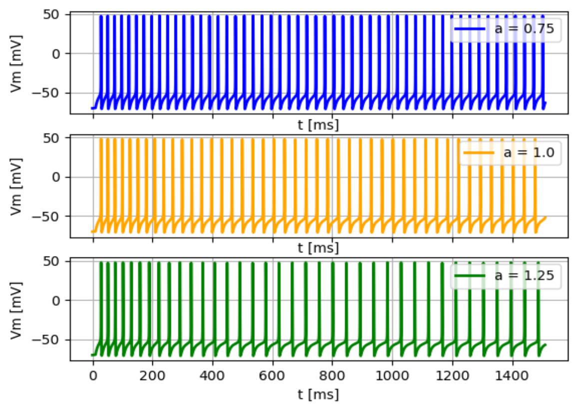


(b) Voltage trace

Figure 9: Change in τ_p to slow down the adaption rate.

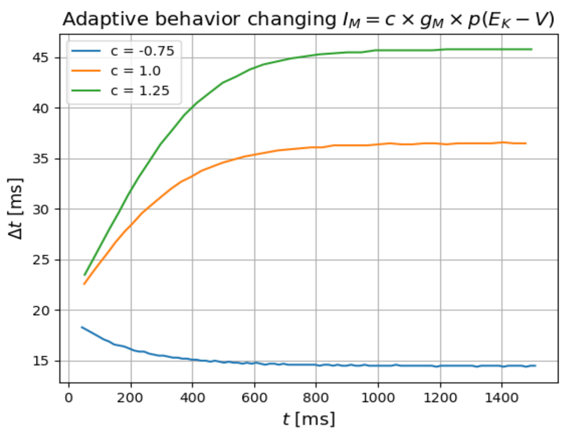


(a) Adaption figure

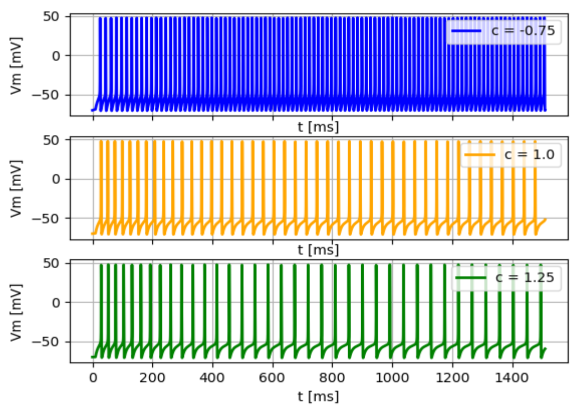


(b) Voltage trace

Figure 10: Change in p_∞ to decrease the stable firing rate.



(a) Adaption figure



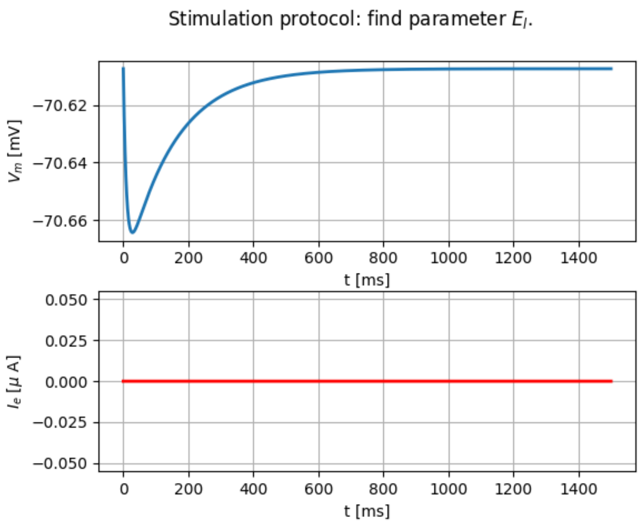
(b) Voltage trace

Figure 11: Change in I_M to reverse the adaptation.

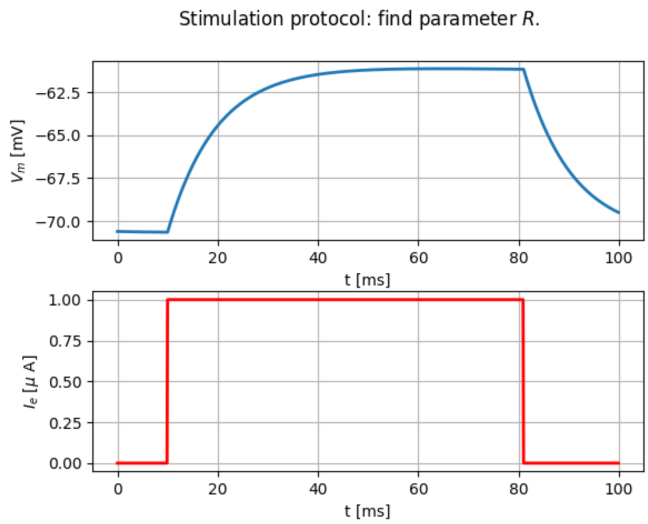
3.5 Passive properties

Parameters	Value
E_l	-70.6073 [mV]
R	9.4963 [k Ω]
g_l	105.3043 [μS]
$V_{63\%}$	-64.6281 [mV]
τ_m	9.0 [ms]
C	0.9477 [μ F]

Table 2: Passive membrane parameters of the HH adaptative neuron model.



(a) Parameter E_l.



(b) Parameter g_l.

Figure 12: Stimulation protocol of the adaptative HH neuron for finding parameters E_l and R.

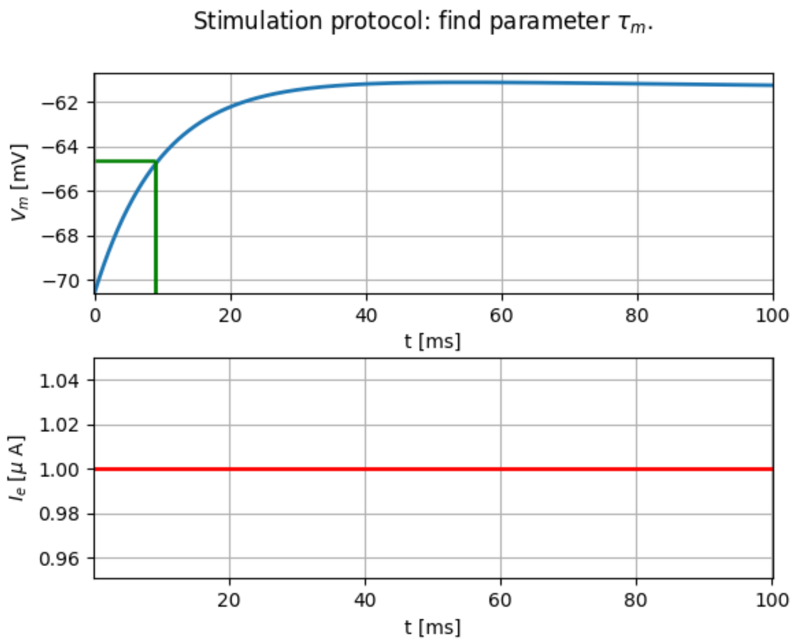
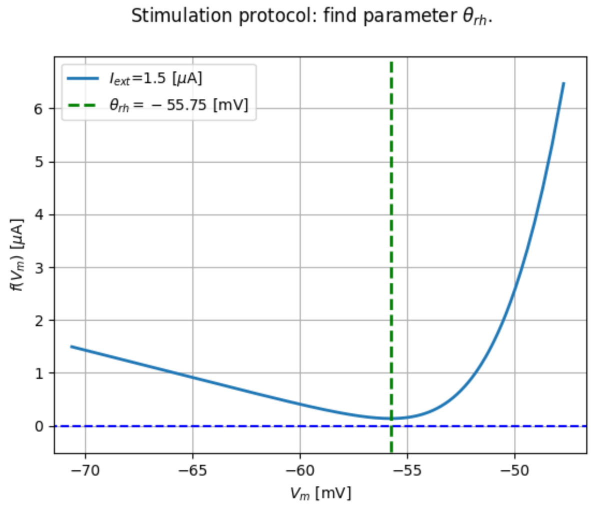
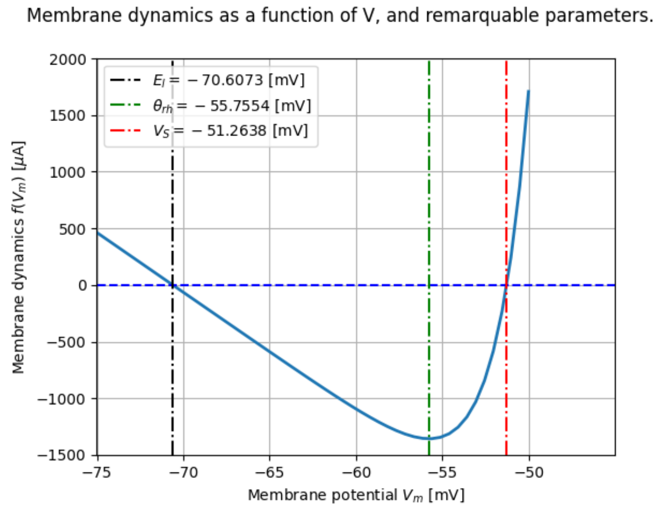


Figure 13: Stimulation protocol of the adaptative HH neuron for finding parameters τ_m.

3.6 Exponential Integrate and Fire



(a) Step current of intensity $I_{ext} = 1.5$ μA applied to an HH adaptative neuron.



(b) Membrane dynamics for the parameters E_l, θ_{rh}, V_S.

Figure 14: Membrane dynamics $f(V)$ as a function of V_m

3.7 Subthreshold adaptation

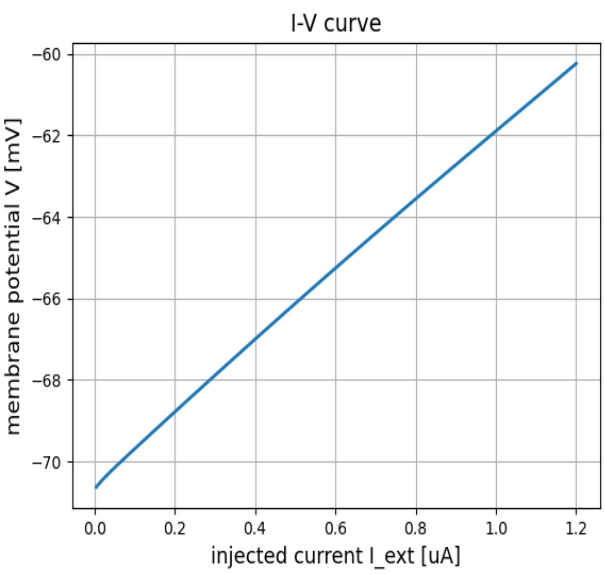
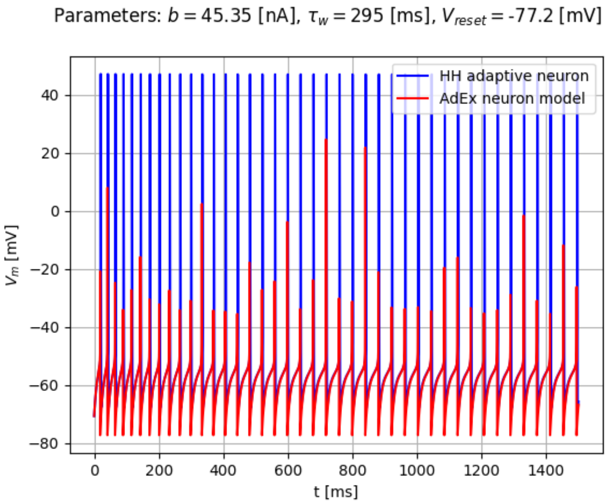


Figure 15: membrane potential V as a function of the injected current I for subthreshold adaptation.

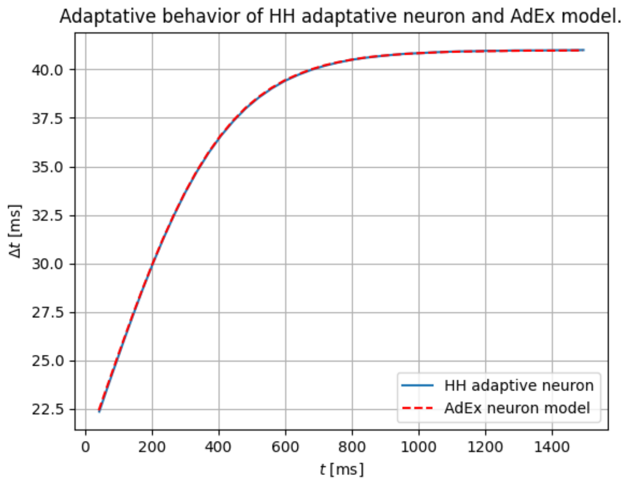
3.8 Remaining parameters

Parameters	Value
V_{reset}	77.2 [mV]
τ_w	295 [ms]
b	45.35 [nA]

Table 3: Parameters of the AdEx neuron model.



(a) Voltage trace of HH adaptative neuron and AdEx model.



(b) Adaptative behavior of HH adaptative neuron and AdEx model.

Figure 16: Comparision of HH adaptative neuron and AdEx model.

3.9 Testing on random input

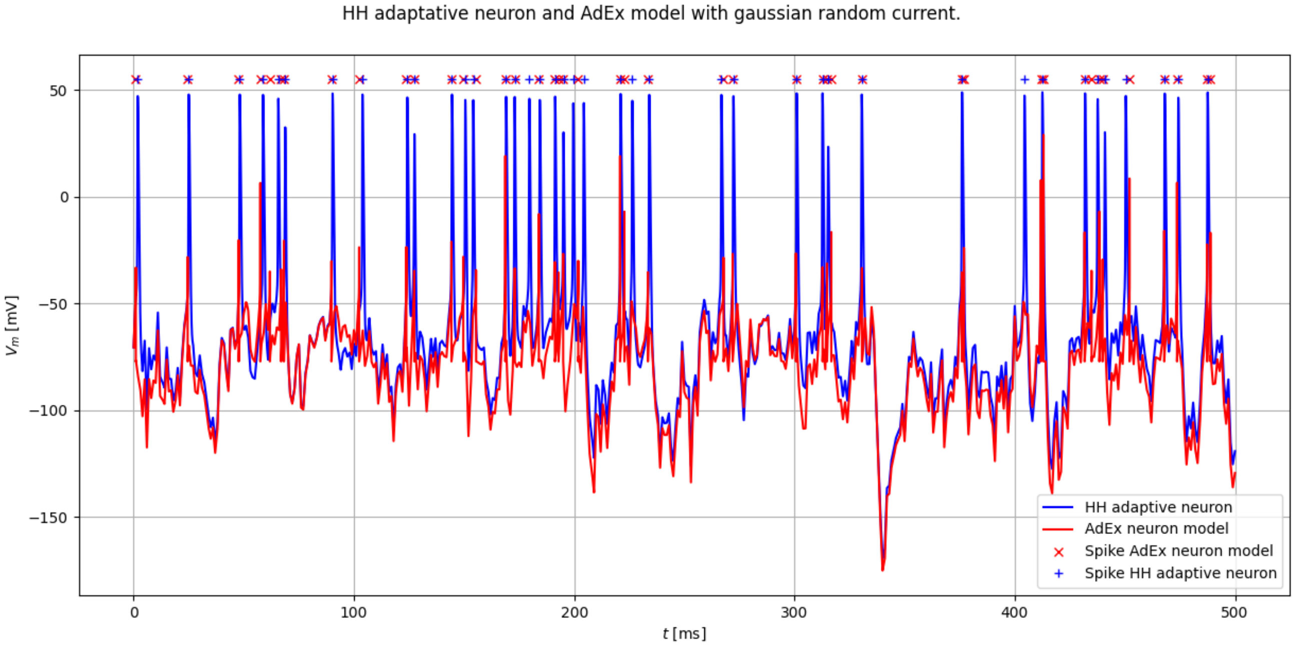


Figure 17: Voltage trace of HH adaptative neuron and AdEx neuron models, stimulated with a gaussian random current of duration 500 ms. The spikes of both models are marked.

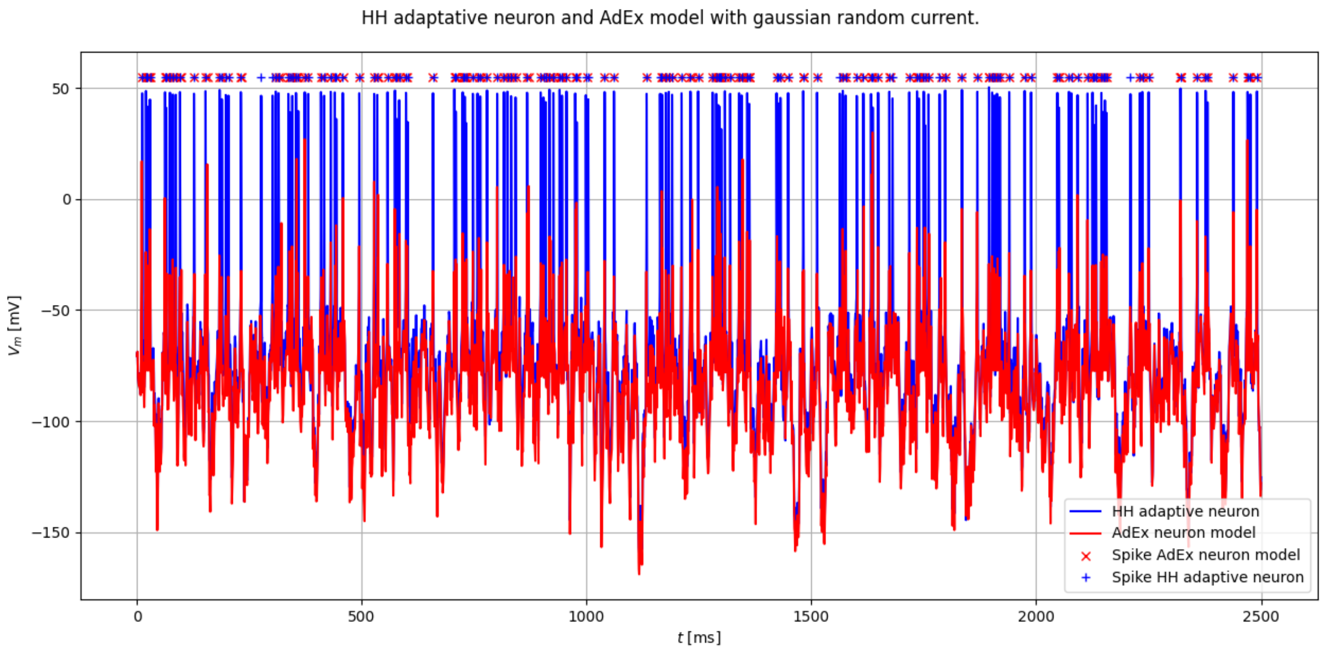


Figure 18: Voltage trace of HH adaptative neuron and AdEx neuron models, stimulated with a gaussian random current of duration 2500 ms. The spikes of both models are marked.