Modelling and classification of neuronal dynamics through Generalised Linear Models

Auteur : Dardenne, Denis
Promoteur(s) : Sacré, Pierre; Drion, Guillaume
Faculté : Faculté des Sciences appliquées
Diplôme : Master en ingénieur civil biomédical, à finalité spécialisée
Année académique : 2023-2024
URI/URL : http://hdl.handle.net/2268.2/20447

University of Liège
School of Engineering and Computer Science

Modelling and classification of neuronal dynamics through Generalised Linear Models

Master's thesis completed in order to obtain the degree of
Master of Science in Biomedical Engineering
by Dardenne Denis

Promoter:
Professor Pierre Sacré

Jury members:
Professor Guillaume Drion
Professor Alessio Franci
Professor Gilles Vandewalle

Academic Year 2023-2024

Abstract

The generalised linear models, so-called GLM, are data-driven models able to capture a wide variety nonlinear behaviours which can be difficult to simulate in classical mechanistic models. Consequently, GLMs occasionally find applications in neuron modelling, providing a flexible solution to address the complexities of neuronal dynamics. Here, this work focus on the main behaviours studied in the neuroscience research field to design relevant GLMs.

Initially, the performance of the GLM is meticulously evaluated based on factors such as the length of the sequence being captured, the number of its basis functions and their designs. The parameters that remain untested are transparently highlighted. While key characteristics of those fitted filters are also discussed.

In a second time, deeper investigations are conducted into the feedback filter of the fitted GLM. Although the GLMs differ according the training sequences, it exists notable similitude between them when the training sequences belongs to the same family, *e.g.* spiking, bursting. The extraction of features is possible thanks to the elements of the GLM. Therefore, the classification of the original sequences according the features of the GLM is addressed at the end of this these, it contributes to a comprehensive understanding of the intricate dynamics underlying neuronal behaviour.

Through analysis and interpretation of GLM performance, this study offers valuable insights into the potential applications and limitations of these models in capturing and reproducing complex neuronal dynamics. By shedding light on the role of model parameters, training sequences, and extracted features, this thesis helps in the design and interpretation of GLM within the framework of neuronal representation.

Acknowledgements

This master thesis concludes my university journey, and I would like to express my gratitude to all those who have supported me in completing this final project.

I am grateful to Professor Pierre Sacré for proposing this topic, which bridges the worlds of engineering and neuroscience. His availability and guidance have been two precious elements to the completion of my thesis, providing support and direction throughout the process.

I want to thank Arthur Fyon, PhD student and teaching assistant, for taking time to explain his topic to me and for patiently answering my questions about it. I would also like to express my appreciation to Professor Jonathan Pillow at Princeton University for kindly addressing the questions that I had following the reading of one of his articles. I am thankful to Marc Frédéric who allows me to use the Julia language on the computers belonging at the school of engineering of ULiège.

I am appreciative to Maël Dengis for the engaging and stimulating discussions we had about our respective master's theses.

Lastly, I would like to express my heartfelt gratitude to my brother and grandparents, particularly to the latter, Willy and Anne-Marie, whose presence has made an indelible impact on my life, without which it would have been radically different.

Contents

Introduction

Motivations

The neuron is the basic element intervening in the information transfer within the nervous system, comprising its essential building blocks. This system is important in the animal kingdom, as its the one that ordains the action to perform. It could be voluntary action as walking, running, swimming, climbing, *etc.* or non-voluntary action as blinking, moving hand after pain *etc.*

Those orders are transmitted thanks to the succession of neuron in the form of ionic currents which are generated by an opening and closing game of channels constituting the neuron membrane. This game is dependant of many causes that could be classified into different categories: the external stimulus, the previous state of the neuron, and the stimuli coming from others neurons. Well understand the impacts of those mechanisms is a quite challenging tasks often investigated in the neuroscience field research.

Neurons represent one of the evolutionary fruit that began billions of years ago on Earth. Through this evolutionary journey, the actual neurons have evolved the remarkable capability to generate wide range of behaviours. They can engage in spiking or bursting activity, transition between bi-stable states, function as resonators, and so much more. This versatility highlights the complexity and adaptability of neural systems in processing and transmitting information. Therefore improve the knowledge of this information transfer is essential to elucidate the principles governing brain function, but also its dysfunction.

In the last decades, lots of models are born to reproduce the most faithfully as possible the observed behaviours. The main one is the conductance-based model, so-called the Hodgkin–Huxley model of the names of its authors, established in 1952 for the first time. This model provides physiological explanations for neuronal behaviours, representing a significant breakthrough in the understanding of how neurons function. Since then, improvements have been found the understand others behaviours. It offers an overview between stimulus, the neuron characteristics, and the induced response.

In recent years, computational modelling has revolutionised scientific research across various disciplines, and neuroscience is no exception. The ability to simulate complex systems and processes using computational algorithms has opened new avenues for understanding

and predicting phenomena that were previously inaccessible through traditional experimental methods alone. However, this progress has not been without its challenges.

One of the modelling approach addressed in the research to reproduce a specific sequence is the generalised linear model, GLM. Unlike traditional linear models, GLMs are capable of capturing nonlinear relationships inherent in neural data, making them well-suited for modelling complex biological systems as the neuronal activity. However, a such model can be very depending of its initial design then that is not so easy at all. Others advantages on GLM are the flexible and comprehensive frameworks provide for the analysis of response across the various of patterns, and the interpretation of its elements that is user-friendly in comparison to more sophisticated modelling techniques. Then the GLMs offer a promising solution to investigate on the neural responses.

For these reasons, this master's thesis explores the GLM through two approaches. Firstly, the thesis will delve into GLM design using some metrics, examining various aspects to determine an optimal design. Secondly, once a suitable GLM design is identified through the initial discussion, the thesis will investigate the deformation of GLM elements to replicate specific sequences. This dual approach aims to comprehensively analyse the effectiveness and flexibility of GLMs in capturing and reproducing complex neural dynamics.

Nowadays, understanding and interpreting neuronal behaviour is no longer solely the domain of researchers. It has become a real goal to companies. Indeed, some of them are primarily focused on advancing neural interface technology to enable control of computers and other devices using brain signals. Others are more oriented toward medical applications, working to develop neural interface systems for neuroprosthetics, or are focus in the treatment of neuronal disorder to restore communication, mobility for individuals affected. This shift highlights the diverse applications and commercial opportunities arising from advancements in neuroscience research.

Structure of this master thesis

This thesis is divided into three main chapters named "*Background*", "*The selection of Generalised Linear Model : approaches and considerations*", and "*Generalised Linear Model as classification tool*" respectively.

"*Background*" refers to the conceptual framework and tools used in this thesis, encompassing the physiology of neurons, both mechanistic and data-driven models, and computational languages utilised in the analysis.

"*The selection of Generalised Linear Model : approaches and considerations*" is the part associated to the investigations into the impacts of certain GLM parameters on the generated solutions.

"*Generalised Linear Model as classification tool*" tries to understand the impacts of firing sequences on the filter h (feedback filter) of the GLM. The deformations is used to extract features of the sequence, *e.g.* the behaviour, the inter spike interval, *etc.*

A last chapter wraps up this these by summarising the results. Further improvements and perspectives are also discussed to conclude this work.

Code availability

The codes used and the FIGURES generated through this work are available on the following GitHub page. https://github.com/DenisDardenne/TFE_neuron-GLM

Chapter 1

Background

This first chapter aims to clarify the theoretical concepts that will be used throughout the entirety of this work. By establishing a clear understanding of these fundamental ideas from the outset, it provides a solid foundation for the subsequent discussions and analysis presented in the following chapters.

The keys points that are treated include the neuronal physiology, its mathematical representation such as conductance-based models or hybrid model, and generalised linear model, commonly known as GLM. These concepts serve as the main foundation; however, additional ones are also used and detailed in their respective sections.

1.1 Physiology of a neuron

Neuron represents a fundamental and critical component within the nervous system of all animal species. Its aim is to transmit the information from a location to another, *e.g.* from an extremity to the brain and vice versa. This allows to animal to perceive the external environment and to execute an action in response to the received stimuli by sensors. It can be explained by the following example, if the animal's hand is above a heat source, like a flame, the hand's sensors transmit information about the temperature on the brain thanks to neurons. If the temperature is excessive, the brain will command to take off the hand away from the flame. This order will be carried out as soon as the movement information is transmitted through the neurons. The path taken by these signals is not constant across time, it can vary if a particular action is repeated multiple times, which is known as the neural plasticity. The network adapts itself to facilitate some tasks, such as learning, movement, and others vital functions. In other words, the neural network is the bases for communication overall the animal kingdom whatever the complexity of the organism. Roughly, whole neurons could be seen as collection of thoroughfares as highways, streets, and towpaths where the information transit.

1.1.1 Structure

The structure of neuron is not unique, it is very dependant of its type. The shape and size are the main elements which are affected by the neuron natures. However, a common structure is shared between them, and this general architecture is describe below.

This specialized cell in the information transport has an elongated shape allowing a better efficiency for its task. The heart of the neuron is called the soma which is the most important part of the cell as it contains the nucleus having a diameter ranging from 3 to 18μm. Other essential components as mitochondria, rough and smooth endoplasmic reticulum, ribosomes, *etc.* are located in the soma to maintain the neuron alive. The soma's size can achieved until 100μm but it is type-dependant as already mentioned. From the soma, numerous cellular extensions emerge, these are called dendrites and can branch into many sub-extensions. The dendrites are components of the cell that propagate information coming from the soma. They are often qualified of dendritic tree due to the sub-extensions that looking like a tree. Those large amount is a result of the role of dendrites which is to capture the stimulus originating from others neurons. As the neuron cannot predict the location where a stimulus will occur, neuron has developed this dentritic tree to increase the probability to catching it. After capturing the information, this is transmitted to soma for treating it before going further. The number of dendrites linked to the soma is function of multiples parameters; such as the species on which the neurons come from, its type, the level of utilisation and so much more. However, for the principal neuron for human, the multi-polar neuron, there are four dendrites linked to the soma on average, having a diameter size from 4 to 0.4μm, at the junction with the soma and the distal part, respectively. The two distances to kept in mind is the total length of the tree which is closed

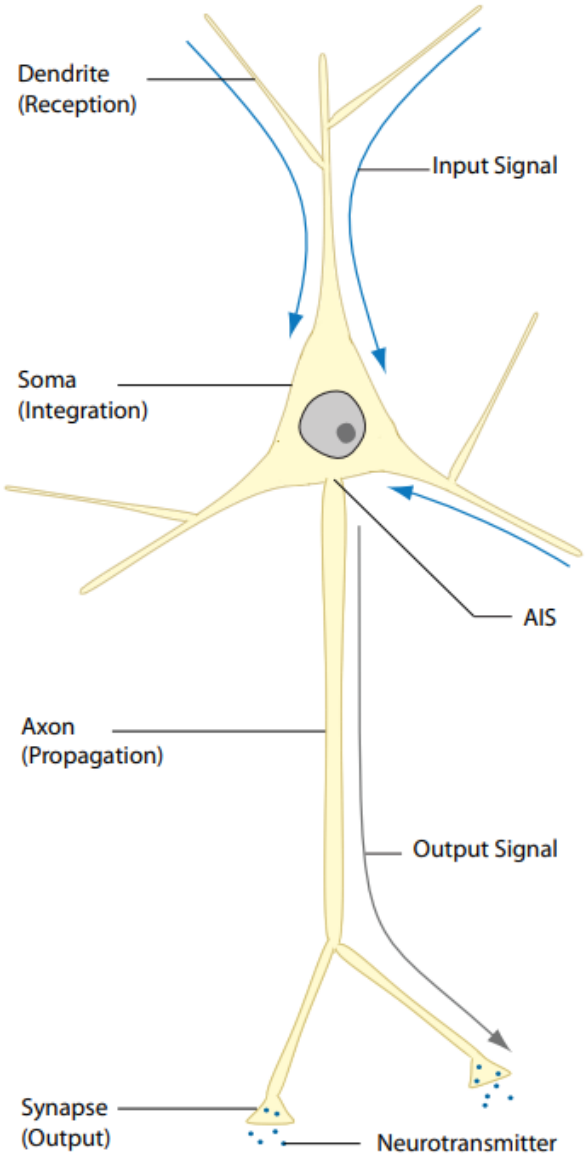


Figure 1.1: General structure of neuron. [1]

to 8,000μm for a single neuron, and the highest distance between two extremities of the dentritic tree which is 1,000μm. [2, 3]

The single axon, another component of the cell, transmits the treated information. Its function is opposite to dendrites, it achieves information to other cells to continue the propagation. The axon transmits information over long distances like a cable, it is the longest part of the neuron, with a diameter that can be quite large. The largest axon discovered so far belongs to the squid giant, measuring $1mm$ in diameter. However, in the most majority of cases it range around 1-25μm. In terms of length, lots of axons have some millimetres only but it can reach more than one meter such as those extending from the brain to the spinal cord. [4]

Consequently it is crucial to guaranty the integrity of the information as its transmission. These action are achieve thought to the myelin sheath that protects axon along some distances. When this shielding is compromised, it means that information is not well managed. Then the control centre of the body, the brain, may not receive stimulus to treat, or the action to perform may not reach its destination. This can result to degenerative disease as multiple sclerosis [5]. To increase the probability to successful delivering the information, the end of axon is branched, it is basically called axon terminal. At this location, synapses allow the information passing, where the level of branched is fewer for axons than dendrites. Then, the stimulus propagation is a spatial and temporal phenomenon. Due to the focus of this master's these, only the second aspect is discussed, the temporal part. So looking at a specific location on the neuron, *e.g.* on the axon, to study the propagation of the ionic current.

1.1.2 Action potential

The action potential is the mechanism allowing the transport of neuronal information through an ionic current, which is produced by the opening and closing of ion channels occurring successively. To go deeper in the phenomenon, a focus on a particular location is necessary, *e.g.* axon.

The neuron membrane is composed of a bilayer of glycerophospholipids containing ions gates authorizing the crossing of particular ions which modifies the amount of ions on each sides. The TABLE 1.1 gives typical values for ions K^+, Na^+, Ca^{2+}, and Cl^- at the resting state for each sides of the membrane.

	K^+	Na^+	Ca^{2+}	Cl^-
Outside	3 mM	**145 mM**	**1.2 mM**	**120 mM**
Inside	**135 mM**	18 mM	10^{-4} mM	7 mM

Table 1.1: Concentration values of main ions across membrane at resting state.

During the resting state, the membrane potential is approximately equal to $-70mV$, it

is called the steady state of the system as it remains unchanged if no stimulus occurs. From the macroscopic point of view, the potential stay around to $-70mV$ because of the closing of majority sodium and potassium channels. The channels have not one unique type; in reality, there are three main exiting types which are m, h, and n as explained below [6].

- Gate m: is closed at the steady state, and opens progressively as the potential becomes more positive,
- Gate h: is normally open, and progressively closed as the potential gets too positive,
- Gate n: is normally close, but slowly opens during the depolarization phase.

Those gates are represented in FIGURE 1.2 by their time constant and opening rate evolution according to the membrane potential.

The two voltage-gated related to sodium channel are the gates m and h, while the gate n is specific to the potassium channel. Depending on the state of a gate, it can be qualified as activated, desactivated, inactivated or desinactivated state, *e.g.* the n-gate is in the desactivated region if the membrane potential is close to $-70mV$ (steady state).

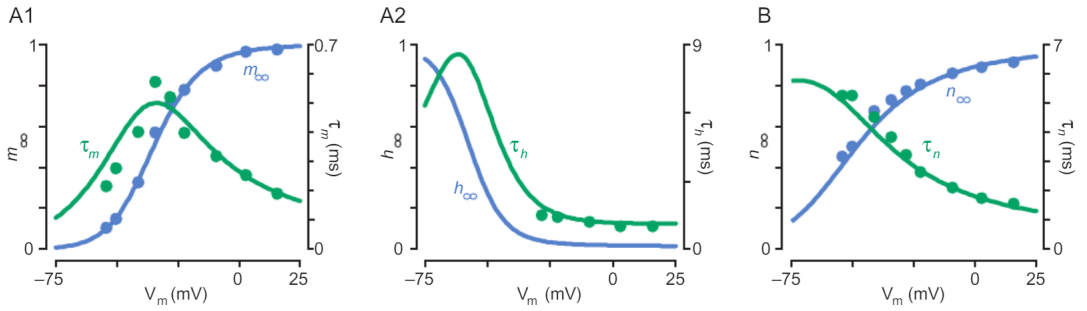


Figure 1.2: Voltage-dependences of gating variables and their respective time constants. [6]

The action potential is fast, and transient modifications occur in the state of the membrane. It is initialised by a signal, which is the ionic current, coming from other cells. The preceding neuron releases neurotransmitters near the dendrites of the neurons which are often Ca^{2+}. The ionic form of calcium is able to bind on ion channels to modify their status, *e.g.* a closed channel opens.

At this stage, during a certain among of time, the neurotransmitters interact with the sodium channel which have the effect to open those gates, and two scenarios can occur. First, not enough neurotransmitters have been delivered to overpass the threshold, around $-55mV$, then a game of opening-closing of gates takes place to return at the original resting state. Second enough neurotransmitters are present to have a sufficient quantity of Na^{+} going from outside to inside of the neuron. The direction of the flux can easily be established thanks to TABLE 1.1. As the membrane potential increases, the sodium channels, particularly the

m-gate, open massively making the potential higher then previously to reach until $30mV$, it is the depolarisation phase.

At this current stage, the percentage of opened n-gate potassium channels is maximum, since then massive flux of K^+ occurs from inside to outside causing a hyperpolarization. This means that the membrane potential is inferior to the one at the resting. As the channels are voltage-dependant, after these two phases, all channels have recovered their initial states. During hyperpolarization, it is essentially pumps, as the well-known the Na^+/K^+-ATPase pump, that work to restore the original state.

The hyperpolarization period can be compared to refractory periods where two types exist, the absolute and relative. During the first one, it is impossible to initiate another action potential. The h-gates remain closed, then it means that no sodium ion can pass, by consequence no action potential can occurs. In opposition, during the relative refractory period, although it is hard to initiate an action potential, it is not impossible. Here the h-gates are open again, allowing to the voltage potential to reach the threshold value thanks to sodium exchanges. However, the amount of Na^+ to depolarize the membrane must be higher then previously, as the membrane is still hyperpolarized.

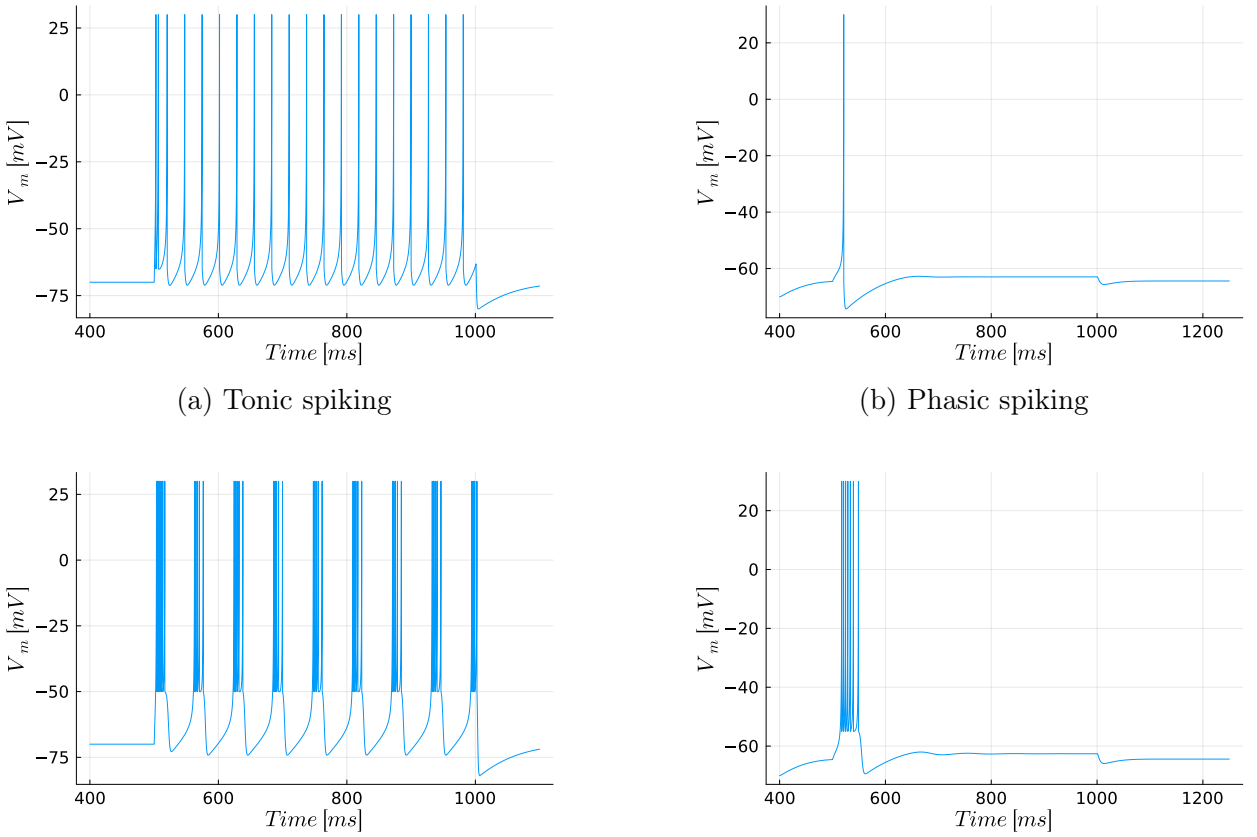
In simple terms, *spikes* are commonly used to refer to action potentials. When a neuron is subject to an action potential, it is commonly described as *firing*.

1.1.3 Behaviours

The behavioural response of a neuron to specific stimulation can vary according to multiple reasons, however the main cause is its intrinsic nature. Various neuron types can respond to various stimuli, therefore multiple firing patterns may be observed and studied. It exists more then twenty firing patterns that have been studied, the four main, available in FIGURE 1.3, are the tonic spiking, phasic spiking, tonic busting, and phasic bursting. Lots of others exist as: class 1, class 2, accommodation, frequency adaptation, spike latency, resonator, integrator, threshold variability, spiking induced by inhibition, busting induced by inhibition, *etc.* Many models attempt to establish a relationship between equations and firing patterns with some success. Two famous models of them are the Izhikevich model developed for the first time in 2003, and the conductance-based model developed by Hodgkin Huxley at mid-1900s. All models use in the master's thesis, including those two, are explained in their appropriate section.

1.1.4 Neuromodulation

The responses mentioned earlier can be altered in presence of chemical species, which can impact the time spiking interval, or the level of amount of spikes per train. These chemical species impacting the response are called neuromodulators. Various classes of neuromodulators exist, some can bind to ionic channels affecting their opening and closing rates, or



(a) Tonic spiking

(b) Phasic spiking

(c) Tonic bursting

(d) Phasic bursting

Figure 1.3: Firing patterns generated with a step-stimulation from $500ms$ to $1000ms$ according to the Izhikevich model, for four behaviours.

impacting the velocity of state transitions. Others trigger the behaviours thanks to complex signalling cascade, then the ions channels are impacted after a certain number of products initiated by the neuromodulators. A last type of neuromodulators require the presence of some chemical components to create new complex which will be capable of modifying the properties of channels.

From a mathematical perspective, these modifications can be translated in term of conductances in the model bearing the same name. These conductances are the heart of the conductance-based model and even a small modification can significantly impact the final response.

Studying the effect on neurotransmitters is a quite challenging task, as matching computational experiments and reality in not straightforward. Moreover, it can help to enhance the understanding of the underlying physiologic effects as such as "How to increase the amount of Ca^{2+} ?", "How to obtain spiking-like response instead of a bursting-like ?", *etc.* Providing

answers to similar questions could be a major advance in the treatment of some pathologies.

In the last part of the thesis, twos particular neurons are studied: the stomatogastric ganglion, and dopaminergic neurons, labelled as STG and DA, respectively. The firing traces will be generated using the *Julia* package *NmodController.jl* developed by A. Fyon, which is mainly composed of conductances based models. [7]

1.2 Mathematical representation

Developing robust and accurate models is a challenging task due to the complexity of behaviours. Some models succeed well to perform it with their own advantages and drawbacks. Two types are popular in the neuroscience field; the conductance-based model, and hybrid model. The conductance-based model tries to replicate the reality as closely as possible without considering the computational resources needed to solve it, and the second move away from biological reality in favour of ease of calculation.

However, it's important to remember that no model perfectly represents reality, they all provide an approximation. As written by George Box, a British statistician in 1976 *"All models are wrong but some are useful"*. [8]

1.2.1 Biologically-inspired models

Biologically-inspired models are developed by scientists to replicate observable behaviours in a way that closely mimics natural processes. These models aim to bridge the gap between theoretical predictions and real-world phenomena. Different approaches exist in the design of a such model.

One approach involves by making parallels with other scientific disciplines, allowing to researchers to compare the studied phenomena with those that are already well-understood. This interdisciplinary method can provide valuable insights and facilitate a deeper understanding of complex biological systems. For instance, scientists can compare neural networks in the brain to electrical circuits, leveraging well-established principles from fields to elucidate neural behaviour. This cross-disciplinary perspective can help identify underlying mechanisms governing the transition of information in a neuron.

Another approach focuses on the causes behind observable behaviours. Investigate into the biological and chemical processes are lead to specific neuronal responses, allowing to understand the fundamental principles managing these behaviours. This method involves understand the intricate interactions between different cellular components, such as ion channels and neurotransmitters.

Additionally, some models are designed to be easily adjustable, allowing the user to simulate various behaviours using empirical parameter values. These parameters, while not always having a direct physical interpretation, permit to researchers to fine-tune the model to match experimental data as close as the observed behaviours. This flexibility is particularly valuable when complex systems are challenging to capture.

1.2.1.1 Conductance-based model

In 1952, A.L. HODGKIN and A.F. HUXLEY were the first authors to publish a model about the neuronal behaviour explaining the underlying physiology. [9]
The proposed a groundbreaking concept by assuming that plasma membrane can be modelled as an electrical circuit. They have been able to made parallel with one other discipline which was already well-know at the time. This innovative approach has revolutionised the field of neuroscience which was only at its beginning, providing a powerful framework for studying the complex interactions within neurons. The membrane mainly composed of bilayer of glycerophospholipids, it exists channels allowing to ions to go inside and outside of the membrane. Each channel has a specific design permitting to one and only one type of ion to transit into it, where this flux of ions creates an ionic current. Hodgkin and Huxley propose their model by replacing the ion channels by variables conductances and voltage sources, and the membrane by a single capacitor. The FIGURE 1.4 illustrates the parallel draw between these two aspects. This electrical circuit can be expressed in mathematical terms in EQUATION 1.3, where the terms C_m, V_m, g_i, V_i, I_i, and I_{app} have a biological signification are defined in the following bullets points.

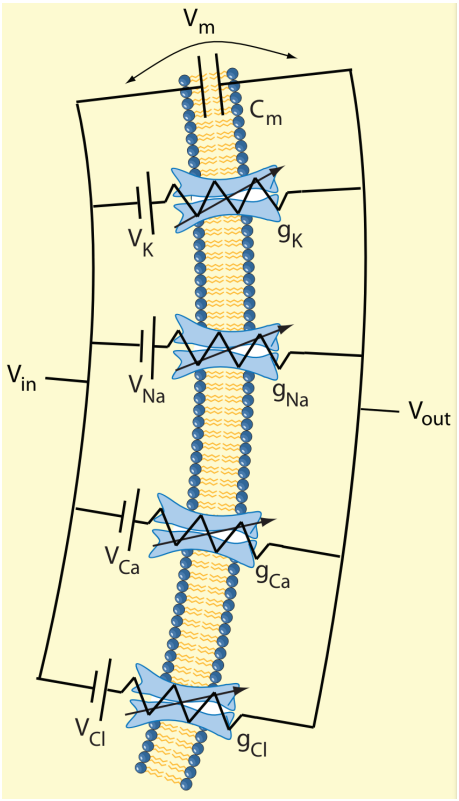


Figure 1.4: Conductance-based model representation. [1]

- C_m the membrane capacity is used per unit area. The capacitor effect arise from the composition of the membrane, indeed the membrane is mainly composed of a bilayer of glycerophospholipids which have polar heads and non-polar hydrophobic tails. Considering that the behaviour of the intra- and extra-cellular medium, composed of water as the matrix, is polar, it means the heads are turned into the exterior of the membrane and the tails to inside. Consequently, the heads are able to accumulate charges, and although diffusion effect exists it is generally uninvestigated. This particular effect is

often assumed as constant then C_m is represented by one constant value, chosen as 1 in the majority of cases.

- V_m membrane potential varies between $-90mV$ and $+40mV$ that corresponds roughly to V_{K^+} and V_{Na^+}. The membrane potential evolution is measurable effect of the opening and closing of gates which provoke concentration gradients thought the cellular membrane which is the difference of the potential between the inside to the outside $(V_{in} - V_{out})$. These potentials are the sum of all V_i related to in- and out-side, where each V_i evolution is time-dependant.

- V_i is the potential linked to the specific ion i. When an ion channel opens, the ions diffuse by this channel until the membrane potential reaches its specific Nernst potential, noted by V_{Nernst} defined as

$$\Delta V = V_{Nernst} = V_{ion} = \frac{RT}{zF} ln \frac{[ion]_{out}}{[ion]_{in}}. \tag{1.1}$$

  The main typical values for mammalian neurons are : $V_{Na} = +40mV$, $V_K = -90mV$, $V_{Ca} = +120mV$, $V_{Cl} = -70mV$. This EQUATION 1.1 comes from the Gibbs free energy ΔG being

$$\Delta G = -RT \cdot ln \frac{[ion]_{out}}{[ion]_{in}} + \Delta V zF, \tag{1.2}$$

  where R and F are the gas constant and the Faraday's constant respectively, T is the temperature in Kelvin, and z the valence of the considered ion (z_{Na^+}=z_{K^+}=1 ; $z_{Ca^{2+}}$=2 ; z_{Cl^-}=-1). To find the V_{Nernst}, the ΔG must be equals to zero.

- g_i the conductance of the i-channel which is not measurable directly, however it but can be found using the voltage clamp method. It is a technique developed by Hodgkin and Huxley in 1952 which allows to study the role of channels in an independent way. Voltage clamp is a iterative method comparing a measurable voltage with a fixed reference. This reference is tracked and adapted to match as close as possible with the measurable value. The measurable value is dependant of the reference what explains the notion of iteration in the technique. [10]

- I_i the ionic current of component carried by ion i. It is the product of the conductance i with the difference between V_m and V_i. As the conductance g_i, the dynamic of ionic current I_i can be study with the voltage clamp method using different extra-cellular solution.

- I_{app} the applied current which can be seen as the external stimulus to obtain a response of the neuron. The term *applied current* a misuse of language because, when neurons are studied, in majority of cases, it is a current which is injected to analysis its behaviour. However I_{app} could also be the stimulation generated by sensors at the extremity of fingers after having touched a needle.

This exhaustive list of elements takes place in the equation of conductance based model, as defined by

$$C_m \frac{dV_m}{dt} = -\overbrace{g_K(V_m - V_K)}^{I_K} - \overbrace{g_{Na}(V_m - V_{Na})}^{I_{Na}} - \overbrace{g_{Ca}(V_m - V_{Ca})}^{I_{Ca}} - \overbrace{g_{Cl}(V_m - V_{Cl})}^{I_{Cl}} + I_{app}. \quad (1.3)$$

This model is appreciated in the literature as it attempts to represent the phenomenon using only differential equation and no reset process. Although a reset process is a useful trick to obtain a rapid fall (which is instantaneous in this case), but in reality whether the fall is fast, it is not instantaneous. Therefore, the transfer of ions is a continuous tasks, then ODEs system must be able to model it accurately.

1.2.1.2 Izhikevich model

The Izhikevich model has published for the first time in 2003. This model, which is a inspired model of the intra-cellular membrane potential phenomena, reproduces the neuron behaviour thanks to a hybrid model. It contains two states variables (v, u) solving an ordinary differential equations system. The first variable v is membrane potential, and the second u a recovery variable. The system is the one solving the equations

$$\begin{cases} \dot{v} = 0.04v^2 + 5v + 140 - u + I(t), \\ \dot{u} = a(bv - u), \\ \qquad \text{if v} > 30\ [mV] \ \begin{cases} v = c, \\ u = u + d, \end{cases} \end{cases} \quad (1.4)$$

where all parameters are fixed as well as the time step between iterations for solving the equations system. In other words, the elapsed time between each value of the couple (v, u) is always the same. In another mathematical aspect, it can be transcribed as

$$(v, u)[t + 1] = (v, u)[t] + (\dot{v}, \dot{u}) \cdot dt, \quad (1.5)$$

where dt is a constant. [11]

Eugene Izhikevich has fitted his model for lots of neuronal behaviours, in the Table 1.2 a subset of the parameters value according to the response type. The parameters fitting has also be done for lots of others different behaviours; more that ten others parameters fitting are available in the Appendix on Table A1.

The Izhikevich model has some advantages but also few disadvantages, like any model. The main benefit is about the computational cost due to the system to solve (time step and reset equations). This system is also easier to understand than more complex one as the conductance based model, but this simplicity could not take into account the biological details that really occur. One other problem is about the parameters value which are found by a empirical

	a	b	c	d	dt
Tonic spiking	0.02	0.2	-65	6	0.1
Phasic spiking	0.02	0.25	-65	6	0.1
Tonic bursting	0.02	0.2	-50	2	0.1
Phasic bursting	0.02	0.25	-55	0.05	0.1

Table 1.2: Parameters values of the Izhikevich model for a subset of neuron behaviours. A larger set is available in the Appendix at Table A1.

adjustment, than it could be difficult to make links between the variation of results and the variation in terms of parameters value. Moreover a small variation can significantly impacts the response of the model, since then Izhikevich model can often qualified as norobust [12]. This no-robustness of the model can be supported using the Table 1.2 by analysing the parameters value of the two-first behaviours. Indeed, only the b value is different of 25%, and its difference creates a modification of the obtained behaviour.

Whether there are drawbacks, in the neuroscience field it is the Izhikevich model which is often used. Indeed, in this research field, the firing pattern are wanted to be used as reference, then the Izhikevich model is sufficient to generate a predetermined sequence.

1.2.2 Data-driven model, the Generalised Linear Model

The generalised linear model, often abbreviated as GLM in the rest of this master thesis, is a statistical tool developed by J. Nelder and R. Wedderburn in the 70s. The initial definition provides by the authors was : "*The technique of iterative weighted linear regression can be used to obtain maximum likelihood estimates of the parameters with observations distributed according to some exponential family and systematic effects that can be made linear by a suitable transformation.*" [13].

So, the aim of a GLM is to adapt its parameters value to be able to extract the specific pattern inside the response to a stimulus given in input. If the fitting is well executed, the stimulus can be given in entry to the fitted model which generated a sequence that could be similar to the initial pattern. The correctness of model will be evolve thanks to the score function using in the GLM.

Going further in the heart of GLM, it must contain three main elements : random component, systematic component, and link function. The majority of the following explanations come from the book "Generalized Linear Models" by P. McCullagh and J.A. Nelder. [14]

1.2.2.1 Components

This section refers to the general representation of a GLM in terms of black box where the three following components simplify the real computation process.

Random component The random component introduces the probability distribution relative to the response variable. Drawing the parallel with classical regression, it corresponds to a normal distribution. In the following representation named FIGURE 1.5, it is referred to the *stochastic process* box.

Systematic component It refers to the explanatory variables in the model x_1, x_2, ..., x_n. This component produces a combination of these variables which is called the *linear predictor* $\boldsymbol{\eta}$ which is given by

$$\boldsymbol{\eta} = \sum_{i=0}^{n} x_i \beta_i = \boldsymbol{X}\boldsymbol{\beta}. \tag{1.6}$$

The *linear predictor* term has always the same expression whatever the intrinsic architecture of the GLM. The term $\boldsymbol{\beta}$ merges all the unknown parameters β_i, it is useful to the general representation in term of input/output.

Link function The link function, often noted as $g(\cdot)$, makes the link between the two components previously defined. It is the bridge between the linear predictions and the nonlinear world of the responses. It indicated how closed is the linear predictor to the expected value of the response. Its mathematical representation is

$$g(\boldsymbol{\mu}) = \boldsymbol{\eta}, \tag{1.7}$$

where $\boldsymbol{\mu}$ is the mean value across simulation. The parallel with the random component in the case of classical regression can still be done, it will correspond to the mean of the normal distribution.

The link function differs according the problem, various expressions exist which can be qualified as problem-specific. A non-exhaustive list of those functions is shown in the TABLE 1.3 thus their application fields.

Therefore grouping the different mathematical representations of each component, the final response can be written as

$$\boldsymbol{Y_i} \approx E(\boldsymbol{Y}) = \boldsymbol{\mu} = g^{-1}(\boldsymbol{X}\boldsymbol{\beta}), \tag{1.8}$$

where $E(\boldsymbol{Y})$ is the expected value across trials.

Name of link function	**Expression of $g(\boldsymbol{\mu})$**	**Application**
Identity link	$\boldsymbol{\mu}$	For a simple linear regression
Log ling	$\log(\boldsymbol{\mu})$	The model uses a Poisson distribution for counting
Logit link	$\log(\frac{\boldsymbol{\mu}}{1-\boldsymbol{\mu}})$	Mostly for binary logistic regression
Complementary log-log link	$\log(-\log(1-\boldsymbol{\mu}))$	Useful when a survival/reproduction analysis is performed

Table 1.3: Link functions according to the application field. The base of $\log(\cdot)$ function is often the Euler's constant. [15]

1.2.2.2 GLM architecture

Year after year, lots of designs have emerged. The main technological advances in neural encoding modelling are feedback filter, and the coupling feedback filters. Three designs (simple, feedback filter and coupling feedback filter) are described below. To facilitate the differentiation between the feedback and coupling feedback filters, the following names have been chosen for the remainder of this work; feedback filters and coupling filters, respectively. Note that the added improvements are incremental, meaning that the GLM using coupling has also a classical feedback filter.

As this Master's thesis talked about neuron response, the following GLMs are specially related to this topic.

Simplest GLM

The simplest as possible GLM, available in FIGURE 1.5, is composed of 3 blocks : a first filter, called **feedforward filter** and noted as k-filter, applied on the stimulus, followed by a **nonlinear function** f. This function is often an exponential one, for this reason the label term "exponential" is sometimes used. At this stage, the stimulus underwent modifications to be the intermediate variable $\lambda(t)$ such that

$$\lambda(t) = f(k \cdot x(t) + C), \tag{1.9}$$

which is defined as the conditional intensity. However, it is also known as the spiking rate in agreement with this topic. The final response $y(t)$ is obtained by the **stochastic process** receiving the conditional intensity.

This implementation assumes that the output $y(t)$ only depends on the stimulus, which is not ideal to study the neuronal behaviour. Indeed the response provided by a neuron depends both stimulus and state of the neuron.

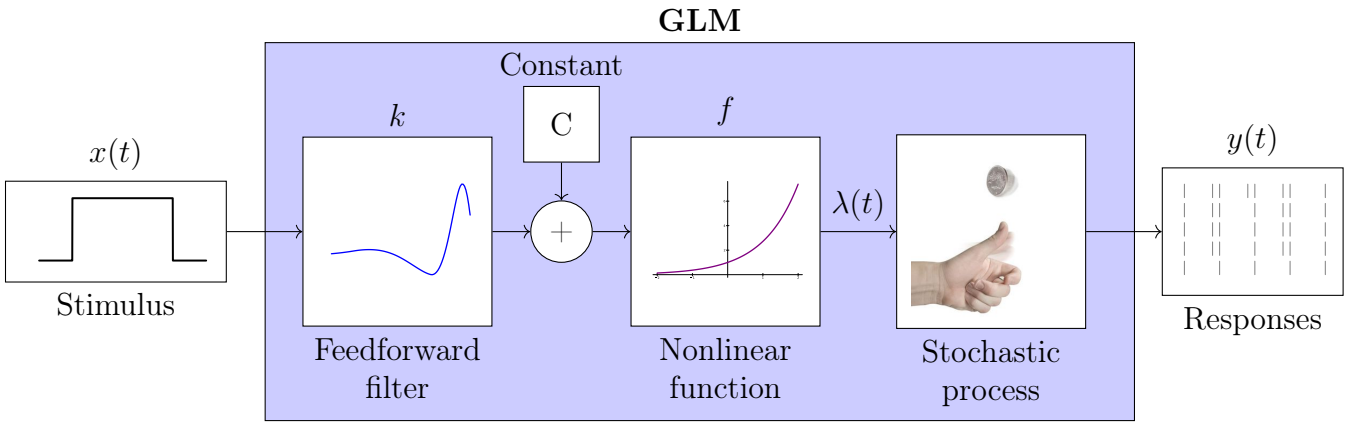


Figure 1.5: The architecture of a GLM only using the feedforward filter k.

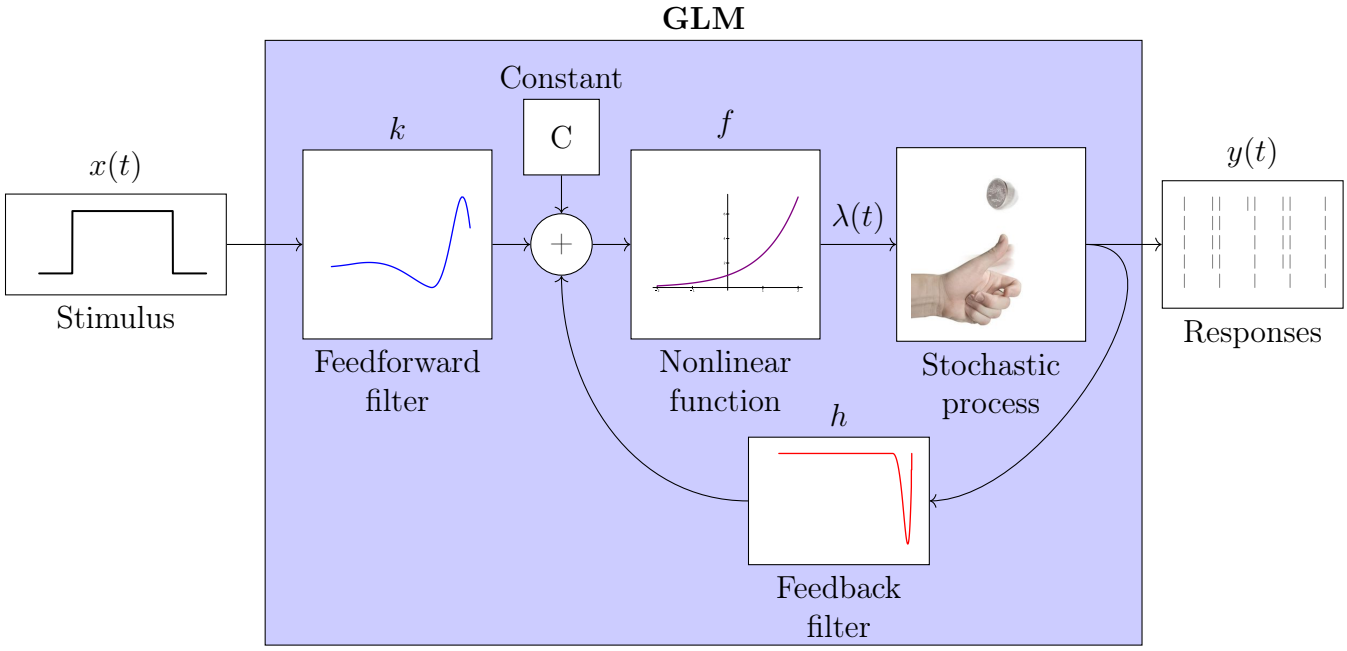


Figure 1.6: The architecture of a GLM using the feedforward filter k, and the feedback filter h.

GLM using feedback filter

The simplest GLM can be enhanced by adding the information on the responses previously found, this architecture is shown in FIGURE 1.6. The implementation is done by adding at $x(t) \cdot k$ the solution filtered by the **feedback filter** h. This modifications affects the conditional intensity that can be reformulated as

$$\lambda(t) = f(k \cdot x(t) + h \cdot y_{hist}(t) + C), \tag{1.10}$$

where $y_{hist}(t)$ is the responses that are already found by the GLM.

In comparison to a simplest GLM, this one takes into account the spike history, which is suitable in the study of behaviour of neuron. However, the architecture still has a drawback, it neglects the impact of others neurons on the behaviour of the one which is studied. The third architecture overcomes this problem, but increases the GLM's complexity.

GLM using coupling feedback filter

The architecture shows in FIGURE 1.7 is the one describing the neuron behaviour as faithfully as possible, it takes into account the effect of others neuron types. In the example available, only one other neuron is considered but this process can be extended to much more neuron types. For a such design, the conditional intensity becomes

$$\lambda_i(t) = f(k_i \cdot x(t) + C_i + \sum_{j=1}^{n} h_{ji} \cdot y_j(t)), \tag{1.11}$$

where i refers to the neuron studied and j to the others ones. The notation h_{ji} considers the **coupling filter** going from j to i. If $j = i$, it indicates that it is a self-coupling feedback filter, also named as feedback filter previously.

To this master's thesis, the second architecture is used for several reasons. First, the interactions between different neuron types are not considered as the behaviour is studied type per type, which rejects the third design. Second, the neuronal behaviour depends on the stimulus; its current state which itself depends on the previous. Then a mechanism of feedback is mandatory to match as close as possible with the observed behaviours. For those reasons, the second design (FIGURE 1.6) is the best for the considered problem.

Another advantage of this architecture is about its implementation, it is a good trade-off between computation time and the accuracy of results. Then the fitting GLM process will design only two filters for each neuron type. In a general way, the number of filters is not the main limiting factor in the fitting process, the number of bases to design it has much more significance impact.

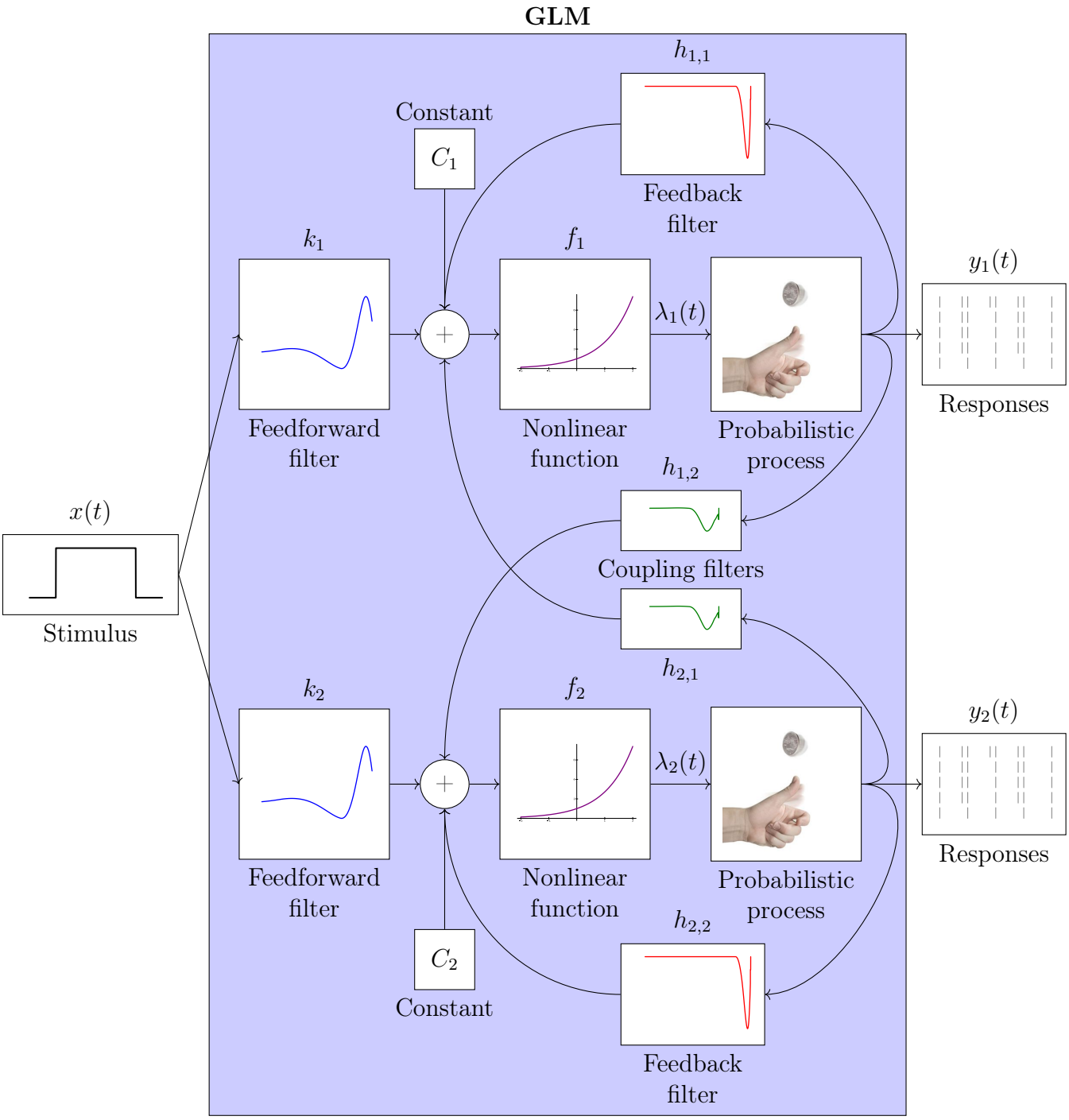


Figure 1.7: The architecture of a GLM using the feedforward filter k, and the coupling feedback filter h_{ij}; where $i = j$ refers to the self-feedback filter.

Different blocks in GLM

A built generalised linear model in composed of three types of processes : filter, nonlinear function, and stochastic event. The two last are established by the user during the design of the architecture; in other words, no modification occurs to these blocks during the fitting phase of the GLM.

The final aim of such model is to design all filters (feedforward k and feedback h) to obtain a prediction matching with the original version of output. To reach it, the GLM must have a set of bases to build the filters. These sets are functions which is used by the model to provide the final filter, which means that the final filter is a combination of all bases present in the sets. Then, having independent functions is suitable to provide high quality results. However, to build the set with a large number of functions is not a sustainable solution, indeed the higher the number of bases, the more challenging it is to find the solution. Then the computation time will increases since then the number of bases will increase as well. In others words, an optimal number of bases exists. It can be defined as the one providing filters which are similar to the found filters with one more bases, for instance.

The nonlinear function f, which is often an exponential function, makes the connection between the stimulus and the previous state of neuron, and with the final current state. The data generated by f represents the spiking rate. Then this rate must go still in the stochastic process to response to the question *"Does a spike occur at this time t ?"*.

The stochastic component induces variability thought predictions which is essential to generate a robust model. It is the last block which provides the final train of spikes, as represented in FIGURES 1.6, 1.6, and 1.7. Its implementation depends of the data type which is expected. It can provides significant differences inside the predictions according it. Then the choice of a selected distribution must not be neglected since each of them have their own operating fields. A first selection is quite easy this performed, below a quick list of distributions with their application areas.

- *Binomial distribution* (specific case of the Bernoulli distribution)
  This distribution is used when the output has only two different states as well as succes/failure, yes/no, *etc.* The probability mass function of such distribution can be written
  $$\binom{n}{k} p^k (1-p)^{n-k} \tag{1.12}$$
  in a general way, where n is the number of independent (Bernoulli) trials, k the number of wanted success, and p the probability of obtain a success at one unique trial.

- *Poisson distribution*
  The Poisson distribution allows to count the number of event occurrences per a certain

unit of time, which is fixed. The probability mass function

$$\frac{\lambda^k e^{-\lambda}}{k!} \tag{1.13}$$

depends on two parameters, λ and k that are the average rate and the wanted number of occurrences respectively. Here, λ, which is the expected value, is also the variance of the distribution.

- *Normal distribution*
  It is specially used for continuous problem. The Normal distribution is an important one often used in many fields when the distribution is not known. The mean μ and the variance σ^2 are the main parameters given the probability density function

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(\frac{x-\mu}{\sigma})^2}{2}\right). \tag{1.14}$$

Others distributions with different parameters and applicability fields exist, as well as Gamma, Exponential, Uniform, Beta, and Logistic distributions; but these one are not particularly relevant according this topic.

1.2.2.3 Optimisation phase

The previous sections explain only the mechanism of a GLM when it is fitted. But, before be able to used it properly, it could be convenient to have a good fitting.
This training phase is made possible thanks to a cost function allowing to the model to adapt its filters during this phase. This step is one of the most crucial in the design of such model because even if the set of bases for each filters, the link function, and the stochastic process have been well chosen, if this cost function is not, the GLM's predictions could be very bad.

In theory, it exists as much as score functions that GLM's users. But in practice, there are few sets of it that are appreciated. The two main are log-likelihood and the akaike information criterion, that will be noted $\mathcal{LL}$ and $\mathcal{AIC}$ respectively. The $\mathcal{LL}$ used the rule *"the higher the better"* which is opposite to the $\mathcal{AIC}$'s rule that is *"the smaller the better"*. Those quantifications are mathematically defined as

$$\mathcal{LL} = \sum_{t=1}^{n} y_t \cdot \log \lambda_t - \sum_{t=1}^{n} \lambda_t, \tag{1.15}$$

and

$$\mathcal{AIC} = 2 \cdot (k - \mathcal{LL}), \tag{1.16}$$

where y_t is the real output, λ_t the spiking rate of the prediction, and k represents the number of parameters in the model which is unrelated to the feedforward filter. Then the $\mathcal{AIC}$ make

a trade-off between the adjustment of the model and its complexity.

To facilitate interpretation, subsequent scores will be standardised by those of the simplest model, defined as the model with the fewest number of parameters in each figure, noted with a "ref" index. The new scores will respond to expressions

$$LL_i = \frac{\mathcal{LL}_i}{\mathcal{LL}_{ref}}, \tag{1.17}$$

and

$$AIC_i = \frac{\mathcal{AIC}_i}{\mathcal{AIC}_{ref}}, \tag{1.18}$$

Any alternative assumptions will be specified in the figure's title.

1.3 Programming language ∼ *Julia*

This work are realised with a teenager programming language, called *Julia*. It is a high-level dynamic programming language launched in 2012. In the official website, *Julia* is described in a nutshell as fast, dynamic, reproducible, composable, general, and open source.

Comparing different programming languages can be a tricky task, because the conclusion can vary a lot according to the type of problem and the computational resources available. Julia, Python, Matlab, C, *etc.* making the choice of a particular language can be difficult to manage, and most of time this choice is user-dependant.

I has selected three language that I managed well, the list is composed of Julia, Python, and Matlab. Then, I wanted that my all codes being available and runnable by everybody that will want it. Then Matlab would not be the best choice as it is not an open source language. All that remains is to watch about the computational performances. Some articles show that *Julia* is faster then Python (and Matlab) in the majority of cases, as [16] and [17]. Moreover, my supervisor Professor Pierre Sacré has encouraged me to select it. Since then, *Julia* has been selected to be the language of my master's thesis.

Chapter 2

The selection of Generalised Linear Model : approaches and considerations

"How do some GLM parameters affect the generated solution ?", this question is broad, so the article [18] is used as the main reference to construct a consistence response. All the selected parameters, and the approach to determined them, are the focus of this chapter. Therefore, using an appropriate set of parameters is mandatory to obtain responses having a high level of confidence. In addition, the model used to generate the firing pattern can also have impacts on the final model. Indeed, two sequences that are close could provide different fits for the GLM. However, in practice, this difference can be neglected, so no comparison is performed.

Another way to express the issue is the intrinsic parameters of the GLM that can significantly influence the results. Therefore this master's thesis focuses on the study on these parameters. In a first time, it examines the choice that are made in the design of the GLM, and the model used to generate the sequence. Afterwards, the impacts of the GLM parameters, such as the length of the sequence and the design of the basis functions, are studied to investigate their effects on the solution obtained.

2.1 The output sequence to capture

The selected model is the Izhikevich model, chosen for two considerations. Firstly, as discussed in Section 1.2.1.2 Izhikevich model, the parameters of this model have already been adjusted to generate lots of neuronal behaviours. Then it is very childish to generate predetermined patterns. Secondly, this model is hybrid, meaning that reset occurs at certain level. This reset allows to speed up the execution of code which can be advantageous when time execution is a constraint. Therefore, in the implementation, the model is used with a constant time step. This small adjustment in execution allows to be solved it without differential equation solver which can be quite time-consuming. While time consideration

may been relevant at first glance, the generation of patterns is negligible compared to the time required for GLM fitting. Then, this argue is not determining in the choice of the model.

As reminder, the Izhikevich model is defined by the following equations (reminder of EQUATIONS 1.4)

$$\begin{cases} \dot{v} = 0.04v^2 + 5v + 140 - u + I(t), \\ \dot{u} = a(bv - u), \end{cases}$$

$$\text{if v} > 30\ [mV]\ \begin{cases} v = c, \\ u = u + d, \end{cases}$$

where more details are available in its first appearance.

In the APPENDICES, a recapitulating table is available containing the parameters values according the observed firing pattern (see TABLE A1). However, only the four main ones are studied here, which are : phasic spiking, tonic spiking, phasic bursting, and tonic bursting.

2.2 General structure of GLM

The GLM developed by the authors follows the architecture previously referred to *"GLM using feedback filter"* in FIGURE 1.6. This decision is justified by the following assumptions.

- The neuron is studied independently of others, then no other type has influence on the behaviour of the studied neuron. Then it can be considered isolated from the rest of the world. It is a strong assumption that is done, but it is necessary because the model generating firing patterns is also designed to study the behaviour of one neuron independently of others.
  $\longrightarrow$ The architecture #3 labelled as *"GLM using coupling feedback filter"* is too complex; therefore, an architecture considering only one neuron must be used.

- The neuron's response depends on its previous states. In the neuron model, the current state is computed based on the previous state and others variables also. Thus, this implies that previous state must be taken into account in the design of the GLM.
  $\longrightarrow$ A feedback loop is mandatory, so the architecture #1 previously named *"Simplest GLM"* is oversimplified to be in agreement with this assumption.

$\Longrightarrow$ The second architecture *"GLM using feedback filter"*, available in FIGURE 1.6, is the one that most closely matches to this problem for the two reasons cited.

It is essential to have a coherent architecture for the following investigations, which has been established. The next steps are discussions about the implementation of each blocks. The four following sections focus on the filters, the nonlinear function, the stochastic process,

and the score fitting function, respectively.

Achieving responses to *"What is the best set of basis in terms of shape and number ?"*, *"Which nonlinear function is used to generate the conditional intensity ?"*, *"What type of stochastic process is aligned with the problem ?"*, and *"What is the most appropriated score fitting function to converge through a solution ?"* is crucial to obtain quality models.

2.2.1 Design of the feedforward k and feedback h filters

The basis vectors used by [18] have the form

$$b_j(t) = \frac{1}{2}\cos(a\log(t+c) - \phi_j) + \frac{1}{2} \tag{2.1}$$

which correspond to a raised cosine basis useful to *"reduce the dimensionality necessary to fit and ensure smoothness of the filters [...]. The parameter c determines the extent to which peaks of the basis vectors are linearly spaced, with larger values of c resulting in more linear spacing, and the parameter a is such that*

$$a\log(t+c) \in [\phi_j - \pi, \phi_j + \pi]$$

and 0 elsewhere". [18]. Note that the full set is added to a step function at the last millisecond to enhance causality between elements.

This set of functions is conserved across majority of simulations. Regarding the number of basis used, this one is also a parameter that is studied. The first assumption that could be made is that the accuracy of the result increases as the number of basis increases. However, this is not entirely true. This point will be discussed later.

Noted that the parameters a and c depend on the others parameters. These new parameters are designed specially to be more intuitive for the user, then in the code implementation the parameters $(a; c; \phi_j)$ are not directly requested. Instead, the parameters such as the number of basis, the elapsed time between the first and the last peaks of function are preferred due to the facility of representation and interpretation.

The raised-cosines basis are appreciated in the research world because of the advantages that it brings. It provides a set of functions that is easily adjustable using only few parameters as already explain. Furthermore, it composed of functions that can catch the information at different time scales; more in details, the meshing for short time scales is thinner than for large time scales, which is a significant advantage. Moreover this basis has been studied and used in many articles where authors conclude that it provides highly accurate results for nonlinear discrete-time problems. [19, 20]

2.2.2 The nonlinear function f

The nonlinear function, which provide the spiking rate $\lambda(t)$, can take various forms. Some of these have already be discussed in Section 1.2.2.1 Components. In the reference paper ($\rightarrow$[18]), two different designs are used; a classical exponential function, and a soft-rectifying nonlinearity function. These functions are labelled as $f_1(x)$ and $f_2(x)$ respectively to facilitate the comparison. The two exponential types are mathematically defined as

$$f_1(x) = \exp(x), \tag{2.2}$$

and

$$f_2(x) = \log(1 + \exp(x)). \tag{2.3}$$

The second function $f_2(x)$ exhibits a linear increasing as the argument value grows, that could generate a GLM log-likelihood which is provably concave. [21]
Functions, $f_1(x)$ and $f_2(x)$, have been tested on the tonic spiking, tonic bursting, phasic spiking, and phasic bursting behaviours, providing very similar results. However, the results obtained by $f_1(x)$ have higher temporal precision. Moreover, others studies have concluded than retinal ganglion cells and neocortical pyramidal neurons are well described using the classical exponential function, $f_1(x)$, as the nonlinear function in the GLM. [22, 23]

For these two reasons, the EQUATION 2.2, classical exponential function $f_1(x)$, is selected. This choice facilitates other operations during the fitting of the GLM, as well as the score evaluation and the computation of the conditional intensity $\lambda(x)$. For now, the function $f_1(x)$ will be denoted $f(x)$, as this usage is commonly accepted.

2.2.3 The stochastic process

The nonlinear function $f(x)$ provides the conditional intensity $\lambda(t)$ which is used in the stochastic process to determined the associated probability. The choice of a probability law instead of others could have a significant consequence on the results. According to the cited distributions in Section 1.2.2.2 Different blocks in GLM, two are serious candidates : $\mathcal{P}$oisson and $\mathcal{B}$inomial distributions.

The fundamental difference between the two is the nature of the counting. $\mathcal{B}$inomial distribution refers to binary data, which could be a relevant choice for the spikes identification. However, in practice, there is no rule prohibiting that multiple spikes occur during a same time step dt. For this reason, the $\mathcal{P}$oisson distribution is preferred as it permits a real counting of data. In a general way, the probability of obtaining a particular number of events is defined as

$$P(y(t)|\lambda(t)) = \frac{1}{y(t)!}(\lambda(t))^{y(t)} f(-\lambda(t)), \tag{2.4}$$

where $y(t)$ is the number of event at a particular time t.

The expression can be adapted to the specific case of neuronal application by assuming that only two states exist : 1 spike and 0 spike. This is the result of a Bernoulli approximation to the $\mathcal{P}$oisson process. It has the effect of disallowing multiple spikes in a single bin. Then probability to obtain a spike is given by

$$P(y(t) = 1|\lambda(t)) = 1 - P(y(t) = 0|\lambda(t)) = 1 - f(-\lambda(t)). \tag{2.5}$$

In practice, using $\mathcal{P}$oisson distribution with such an assumption means that the probabilities of finding more than one spike are redistributed to the probability of obtaining only one spike.

2.2.4 The score function evaluating the model

The fitting phase is an essential step in the design of an accurate GLM, then the selection of the scoring function must also be carefully argued. In the article [18], the model is fitted by maximising the log-likelihood

$$\mathcal{LL}(\theta) = \sum_{t=1}^{n} y_t \cdot \log \lambda(t) - \sum_{t=1}^{n} \lambda(t) = \sum_{t=spike} \log \lambda(t) - \sum_{t} \lambda(t), \tag{2.6}$$

where θ is the aggregation of all parameters. $\mathcal{LL}(\theta)$ is a measure commonly used in model design due to the advantages that its definition provides.

Firstly, the log-likelihood allows comparison between different models using a single value which is obtained without simulations of the model itself. This is a significant advantage as the model simulations are not required to obtain scores during the fitting. Moreover, $\mathcal{LL}(\theta)$ is not sensitive to scale transformation, it means that the time step will not have an impact on the generated score. Finally, the concept of log-likelihood has been introduced more than a hundred years ago. Consequently, it has been studied years after years to be well understood and largely used in many application fields.

However, the log-likelihood estimator has also drawbacks, the main one being that it does not take into account the number of parameters in the model. Then, the Akaike Information Criterion ($\mathcal{AIC}$) is also used to evaluate the difference in results according to the function used. The $\mathcal{AIC}$ has the following expression

$$\mathcal{AIC}(\theta) = 2k - 2\mathcal{LL}(\theta), \tag{2.7}$$

where $\mathcal{LL}(\theta)$ is the log-likelihood as defined in Equation 2.6, and k represents the number of parameters in the model. Typically

$$k = n_k + n_h + 1 \tag{2.8}$$

where n_i is the number of basis in the design of the i-filter, and the addition of 1 comes from the presence of offset variable, previously called "Constant".

2.3 The factors influencing the accuracy of the GLM

Some aspects that could impact the results are dealt in the following sections. Increasing understanding about the sensitivity parameters is essential to use GLM as properly as possible. So multiple parameters will be managed, such as the length of the periodic sequence, the choice of basis, and the design of the cost function during the fitting phase. A discussion of their impacts on the final results will be conducted analysis after analysis.
All the following GLM are designed using the function `optimize()` in *Julia*, where some parameters are provided to it, such as the solver and its options. All codes are available in the Github page, as already mentioned in the Section Introduction.

2.3.1 Length of data during the training

An important factor influencing the computation time of any model is the size of managed data. Thus, understand well the link between length of data and performance is essential to be able to make a trade-off between accuracy and computation time. For example, if the use of few periods in the set of training does not improve the model, it is more relevant to use only one period to speed up the code execution.

The term *accuracy* in model accuracy is a general word and it could be evaluated using various approaches. But here, the log-likelihood is predominantly used to refer it. Furthermore, since the computation time varies across the devices used, the notion of iterations is preferred to compare models. Fitting a GLM in a numerical way is an iterative process, so the iterations refer to the number of steps needed to obtain the final version of the model. However, the time needed to one iteration is not constant across the simulation; in reality, the first iterations take much more time than any others. Thenceforward, the time difference between a model taking 200 iterations versus one taking 400 iterations is much less than double.

The log-likelihood, as defined in Equation 2.6 is dependent on the length of the sequence, therefore to observe the evolution across different lengths of sequences, this measure is divided to its period number p. Each new measure, symbolised by $\mathcal{LL}_p(\theta)$, is compared to $\mathcal{LL}_1(\theta)$, noted $LL_p(\theta)$ $(= \frac{\mathcal{LL}_p(\theta)}{\mathcal{LL}_1(\theta)})$, allowing a simple comparison with the log-likelihood computed on a single time period. This evolution, $LL_p(\theta)$, is shown in Figure 2.1 for different behaviour types. It is essential to precise that each value is the mean score of the log-likelihood computed for the fitted GLM on n basis of filters k and h, where n varies from 3 to 14. Then, these evolution are a general observation to conclude about the best number of period to use. This conclusion does not take into account the best set of parameters that will be found later; in others words, this approach is general which means that the conclusion could differ to the best set of parameters. Here, assuming that the conclusion is correct for all sets of parameters, what is not verified by the following.

Thanks to Figure 2.1 (coming from Figure A1), a conclusion about the impact about the number of periods can be drawn. In a general way, increasing the number of periods

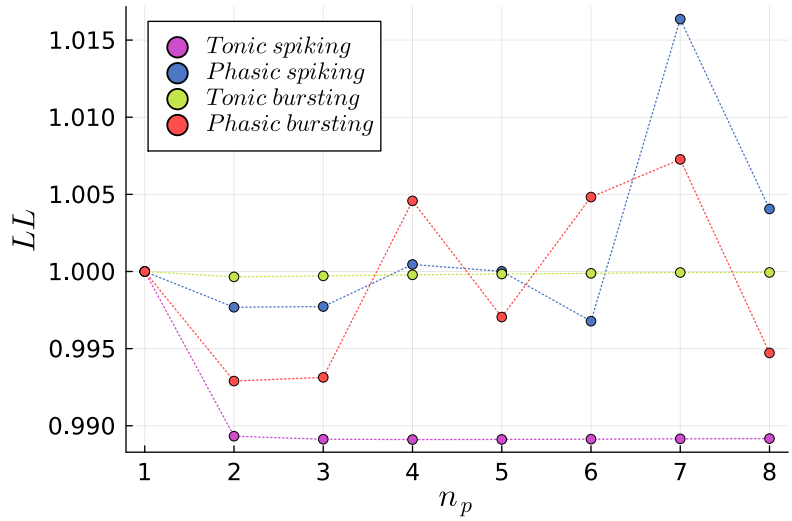


Figure 2.1: Mean LL ratio evolution. Each mean curve of LL is divided by its first value.

has no effect on the log-likelihood, less than 2% of difference. But before to conclude that the case 1-period is the best one, it is relevant to look at the number of iterations needed by the GLMs. This number of iterations is also a factor impacting the performance of a model; higher it is, the longer the fitting of model is. Thus, being able to understand the relation between this factor and the number of periods in the sequence fitting the model is just as essential. The following FIGURE 2.2 shows this evolution, where the values found follow the same succession of operations as the one explained for $LL_p(\theta)$.

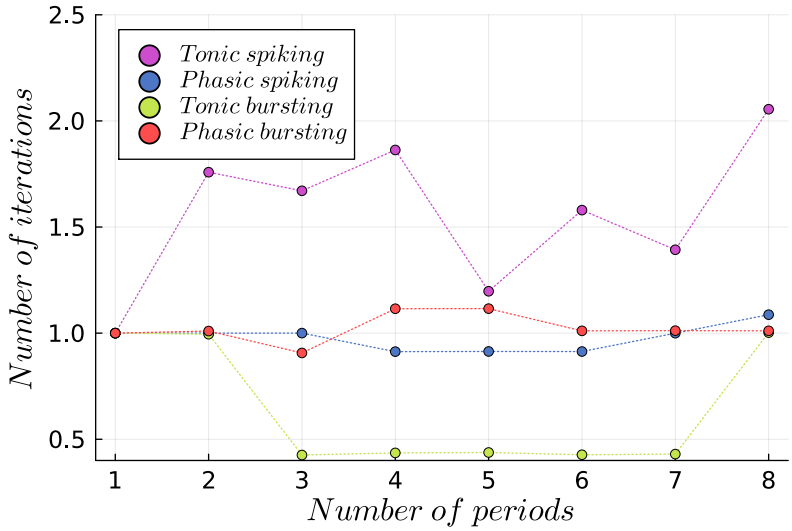


Figure 2.2: Mean iteration ratio evolution. Each mean curve of iteration is divided by its first value.

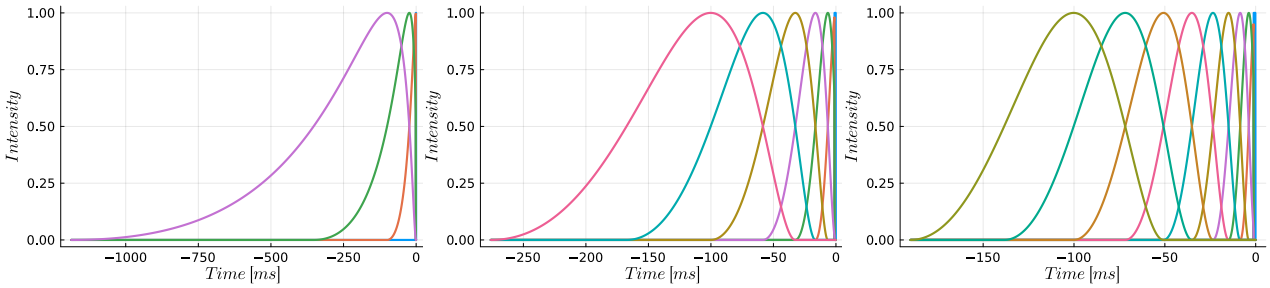
No conclusion can be extracted through FIGURE 2.2. It does not show a particular link between the number of periods and the number of iterations performed by the fitting process. However, one characteristic seems to be redundant : the insensitivity of phasic behaviour. Indeed, the number of iterations for phasic spiking and phasic bursting remains approximately the same across simulations. Therefore, increasing the number of periods p is not efficient in term of iterations.

To conclude on the link between time periods and performance, increase the number of time periods p does not improve neither the log-likelihood nor the amount of iterations needed to reach the final solution. Accordingly, the best case is $p = 1$, which is used for all the following results.

2.3.2 Number of bases in the set of functions designing the filters

Using the previous conclusion, which is that "fitting the GLM on one time period is the best solution in terms of accuracy and complexity", it is time to investigate about the impact of the set of basis on the final design of the GLM. This impact is evaluated using two different approaches : the accuracy of the model, and the filters deformation (k and h) thanks to log-likelihood (LL) and dynamic time warping evaluation, noted as dtw.

The functions constituting the basis for filters follow the EQUATION 2.1. As shown in FIGURE 2.3, adding one base to a set of basis modifies the entire set. The effect of such distortion makes the meshing thinner than previously.

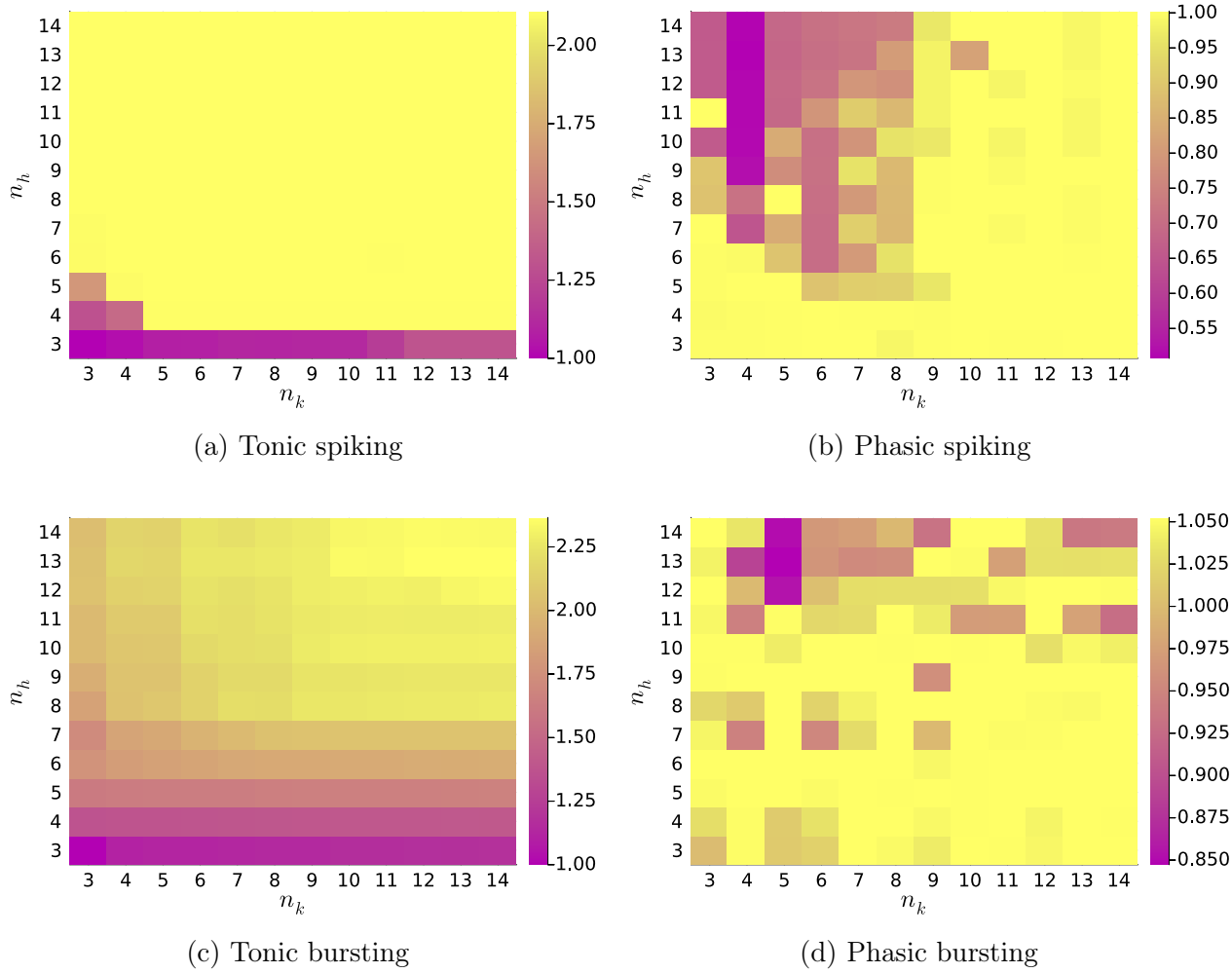


(a) Four bases functions (b) Seven bases functions (c) Ten bases functions

Figure 2.3: Different sets of bases to construct the filters.

The FIGURE 2.4 shows the evolution of the log-likelihood according to the number of basis on filters k and h. The observations that could be made are behaviour-dependant, but same conclusions can be reach for the *phasic* and *tonic* types.

Tonic type : The main limiting parameter is the number of basis in the design of h-filter. When it increases from three to four, the log-likelihood value obtained drastically approaches its quasi-best value. In opposition, the number of basis in k is not a predominant factor for improving results. Indeed, increasing the complexity of k-basis (*e.g.* from three to four for



(a) Tonic spiking

(b) Phasic spiking

(c) Tonic bursting

(d) Phasic bursting

Figure 2.4: Evolution of LL for the main behaviours according to the number of bases in the set of functions designing the filters.

spiking) only slightly improves the fitting according to the LL measurement. The effect of parameters is much more visible in the SUBFIGURE 2.4c. Indeed, increasing to the right side ($\nearrow n_k$) has no effect on the colour, indicating that the score remains the same. However, increasing in the upwards direction ($\nearrow n_h$) increases the score, which is wanted.

Phasic type : A small number of basis in each set is enough to reach a consistent value of log-likelihood. In fact using three to five basis in the set designing filter is sufficient to obtain a qualitative model, according the used measurement. Adding too much basis generate local minimum on the log-likelihood, which means that the model is poorly fitted in comparison to the reference. The definition of the reference model comes from Section 1.2.2.3 Optimisation phase, it is the model $(n_k; n_h) = (3; 3)$.

After having studied the modifications of the log-likelihood score in relation to the num-

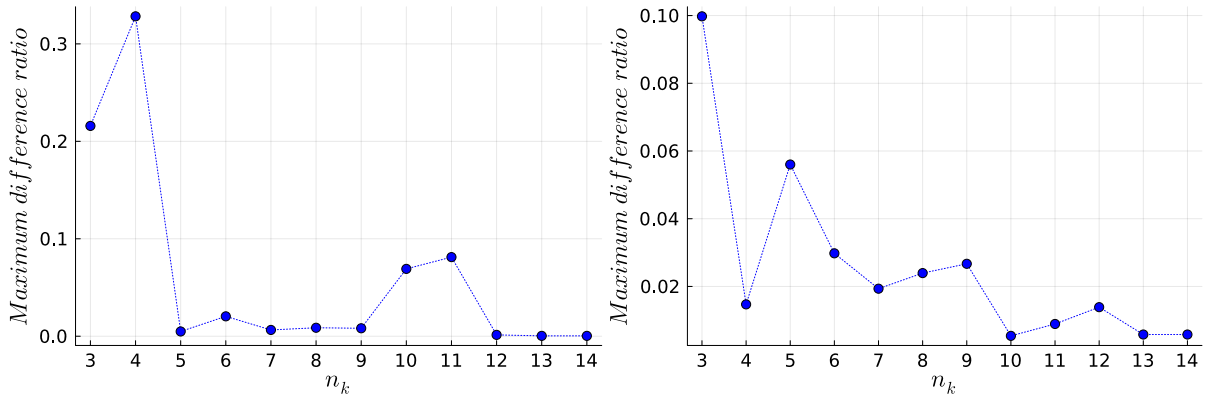
ber of basis in the design of the GLM model, the focus shifts to the modifications inside filters. For this focus, only the h-filter is studied for the two tonic behaviours, spiking and bursting. This spotlight is chosen for two main reasons; first, the design of the k-filter could dependant on the stimulation that is applied, making it difficult to extract relevant conclusions. Second, in the second part of this thesis, only the filter h will be used in the fitting of the GLM. Therefore, a deeper study on h is mandatory for the smooth running of the project.

For the selection of the number of basis used in k, it is fixed to 5 and 6 respectively. These numbers have been selected based on previous results. To find them, each LL-ratio has been divided by the right-value, ensuring a comparison with the same number of basis in h-filter. From here, only the maximum is extracted to each vertical raw (one value per number of basis in k-filter), those obtained values are available through FIGURE 2.5. Assuming that modification of 5% can be neglected, the best values are the found by browsing the curves from right to left, where 5 and 6 and the last values under the threshold fixed to 5% for tonic spiking and tonic bursting, respectively. Note that for the tonic spiking behaviour, the value that should be selected according to the previous path to follow is 12. However, the scores located at index 10 and 11 are due to the optimisation problem that does not find the right solution at index 11. For this reasons, in the search of the best number of basis in the design of k-filter, it can be wise to restrict it to area from 3 to 8 number of basis. The value used as a threshold, it is purely an arbitrary choice. The 5% threshold is selected in the aim of obtaining a relevant trade-off between accuracy of model and low number of basis as possible.

Noted that for SUBFIGURE 2.5b, the measurement at position 4 is also inferior to the threshold value which is due to the computation method followed. It is the percentage difference between two successive log-likelihoods. In other words, the results at index 4 and 5 are very similar, but they are bad ones. It is for this reason that the curve increases drastically and suddenly.

About the deformation of filters, considering that the size of h-filter depends on the number of basis (due to the way of code implementation), the analysis of filters is limited to the first $150ms$. This part to the filter part is the one containing the most relevant information for this particular problem. The relevant region depends mostly on the interspike interval. Here, the last $150ms$ corresponds to the non-null part of filter, which provides information about the spiking generation probabilities. If a previous spike is multiplied by a negative intensity (resp. positive), it has the effect of reducing (resp. increasing) the probability that a spike occurs at the current time. In other words, the feedback filter h can be seen as a representation of the refractory periods. Note that the effects of both the k-filter and the offset component are not taken into account, which can bias the interpretation in terms of refractory periods. This issue will be overcome in the next chapter.

In FIGURE 2.6, two graphs are available, each corresponding to the h-filter but associated with different behaviours : *tonic spiking* and *tonic bursting*. Several observations can be discussed from these figures.



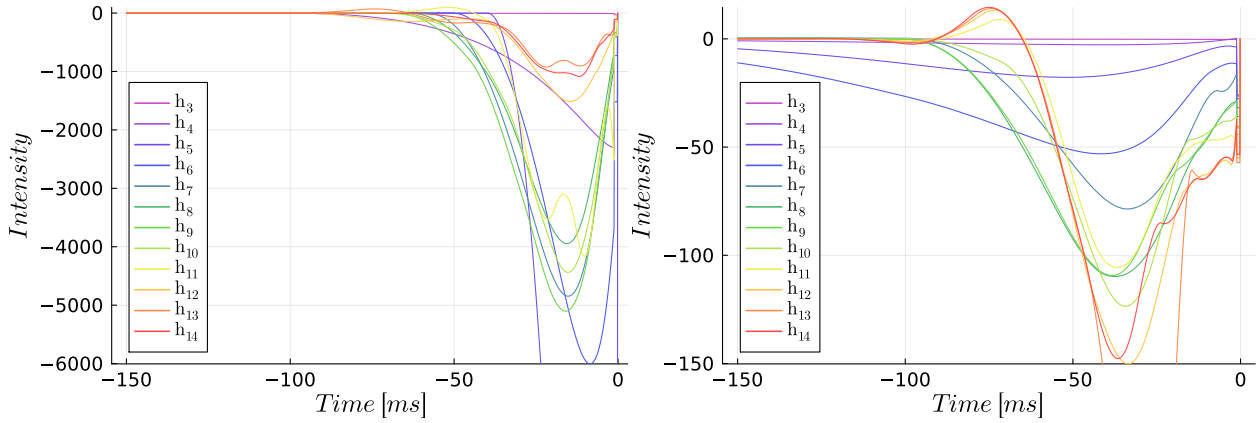
(a) Tonic spiking (b) Tonic bursting

Figure 2.5: Evolution of the LL-ratio difference for two neuronal behaviours. The ratio are found by comparing the LL at position i with the one at the position $i + 1$.

- For tonic spiking
  - Designing filters using too few basis can generate a post-spike filter being considered as outliers. In this topic, an outlier filter is one that has a shape significantly different from its neighbours.
  - As the number of basis increases, it generates some oscillations close to the extremity values, as seen in filters from h_{11} to h_{14}. However, this multiple bottom is not suitable for a such neuronal behaviour. It could perturb the neuronal dynamic and corrupt the ability of the model to reproduce the behaviour of type "tonic spiking". $\rightsquigarrow$ Even if these oscillations will have no impact due to their intensity.
  - Moreover, the majority of filters are composed of two main parts; a large negative, and a small positive in amplitude.
- For tonic bursting
  - The higher the number of basis in the design of the filter, the higher the amplitude. This observation is only valid for the tested design, and it does not ensure that it is effectively the case for a higher number of basis.
  - The region between $-50ms$ and $-100ms$ is approximately the same across the filters, which could mean that the majority of filters well-fitted in this area.
  - The higher the number of basis, the more abrupt the change at times closed to $-1ms$, $-20ms$, and $-45ms$.

How is the best filter determined ?

Before determining the best filter, it is important to establish its general characteristics. As previously explained, having a double bottom (close to the extremity value) in h_{11} for tonic



(a) Tonic spiking (b) Tonic bursting

Figure 2.6: Some feedback filters according to the number of basis for the two studied behaviours.

spiking is unexpected for practical reasons. In a biological standpoint, this double bottom means that to probability to obtain a spike increases slightly during two time periods highly no-spiking. According to the spike generation model to the model used to generate spike, Izhikevich model, the correlation doesn't exist as it doesn't use probabilistic approach.

The amplitude of the filter is an important information as it modulates the probability of obtaining a spike. However, the lower the bottom, the lower the modification of the result. As a reminder, the nonlinear function used is the classical exponential, so

$$\exp(\frac{x_1}{2}) - \exp(x_1) > \exp(x_1) - \exp(2x_1) \quad \text{where} \quad x_1 < 0$$

which means that having a high amplitude, such as filters h_5 (tonic spiking) and h_{14} (tonic bursting), adds certainty in the prediction which can be neglected. Hence, the general shape of the curves are discussed in relation to the neuron behaviour in the following SECTION. Before to analyse it, it is relevant to determine the best number of basis in the filter for each behaviour.

To determine the optimal number of basis in filter h, the notion of Dynamic Time Warping, also known as dtw, is used. This technique facilitates the measurement of similarity between two sequences. Although it is a time-consuming approach in comparison to other methods as mean square error or absolute error, it provides more relevant information in this context. Indeed, the difference in terms of filters shape is much more significant than the differences in amplitude, given the kind of information that it delivers.

The dtw score is found by the following steps :

- Creation of the cost matrix where all elements are defined by

$$M(i,j) = dist(A(i), B(i)) + min(M(i-1,j), M(i,j-1), M(i-1,j-1)), \qquad (2.9)$$

where dist() is a particular metric.

- Finding the warping path score by selecting the minimum values from $M(end, end)$ to $M(1, 1)$.
- According to the reference, the score is established by the mean of the previous vector score, or by only its first component which is its maximal value. For the following discussion, the second score is used as it is the version that is implemented in the Julia function `dtw()`.

The TABLE 2.1 reflects the dtw score for each pair of neighbouring filters, h_- with the previous one, h_+ with the next one. Note that all h-filters have been normalised to facilitate the comparison between filter. The normalisation process allows to focus on the dwt analysis through the shape and not through the amplitude of filters, which is benefit in the particular case for one main reason. The particular shape of filter has much more impact on solution than its stretching, this notion has already be mentioned previously.

To choice the number of parameters, it is important to fix some criteria to follow. Firstly, to obtain a reasonable complexity, the research range is imitated from 5 to 10 numbers of basis. These numbers are a personal choice based of both the accuracy and the computation time during the GLM's fitting process. Indeed, the computation time increases drastically as more basis are added to the set. Secondly, a first h_i is selected by comparing the dtw scores, the one having the lowest score being selected.

Using this path to follow, the best h-filters are h_8 and h_9 for tonic spiking and bursting, respectively. The number of basis can still be reduced by considering the dwt of h_{+1}. This can be interpreted in two ways : first, h_i is reduced using a threshold value (*e.g.* the best set of basis in the small one having a score inferior to 2); second, the set is selected by analysing the magnitude between basis. The first interpretation is user bias as the threshold value is an arbitrary choice. Therefore, the best solution could differ according to the user's interpretation. The second interpretation is much more generic and follows rule that is not subject to interpretation. It means that the reduction of the basis' size in the design of filter h is allowed if and only if the dwt score related to the previous (at least one base less) is in the same magnitude order than the best solution found previously. Using this second approach, the best h-filter is the one containing 7 basis for the tonic spiking and bursting behaviours. For the following analysis, it is desirable to have a same number of basis whatever the behaviour.

To sum up the solutions found available in TABLE 2.1, the research ranges are inside the light gray boxs, the dark gray box is the lower value of sums in the range, **the bold value** referred to the best dtw of h_{+1} after reduction, and light green boxs denote the final best solution.

spiking	h_{-1}	h_{+1}	Sum	*bursting*	h_{-1}	h_{+1}	Sum
h_3	/	18.09	/	h_3	/	6.06	/
h_4	18.09	0.002	18.09	h_4	6.06	181	187
h_5	0.002	1.7	1.7	h_5	181	38	219
h_6	1.7	0.5	2.2	h_6	38	27	65
h_7	0.5	**0.002**	0.5	h_7	27	**0.15**	27.15
h_8	0.001	0.004	0.005	h_8	0.15	0.15	0.3
h_9	0.004	0.005	0.009	h_9	0.15	0.03	0.18
h_{10}	0.005	0.82	0.83	h_{10}	0.03	0.16	0.2
h_{11}	0.82	0.76	1.58	h_{11}	0.16	0.02	0.18
h_{12}	0.76	1.02	1.78	h_{12}	0.02	71	71
h_{13}	1.02	0.93	1.95	h_{13}	71	0.002	71
h_{14}	0.93	4.82	5.75	h_{14}	0.002	15	15
h_{15}	4.82	/	/	h_{15}	15	/	/

Table 2.1: Dynamic time warping analysis. Research ranges, best dwt-scores, **best dtw of h_{+1} after basis reduction**, and best h-filter.

2.3.3 The score function

The heat maps illustrating the evolution of LL and AIC for the two studied behaviours are in FIGURE 2.7. According to the particular behaviour, the conclusions are quite different.

Tonic spiking:

Comparing the SUBFIGURES 2.7a and 2.7c, the conclusion regarding the best set of parameters is differs significantly. For the LL, whatever the chosen set in the yellow area, the precision of the model would be the same. While, the score found thanks to AIC achieve different conclusion. In the yellow-orange area, it exists a score degradation as the number of basis increases. In other words, the LL increase is not sufficient to compensate the addition of parameters in the AIC score.

In this particular case, the AIC seems to be better to establish the best set of parameters as the LL converges rapidly. Therefore, by solely considering the akaike information criterion, the conclusions would have been different than previously. By a quick inspection of SUBFIGURE 2.7c, the best set of parameters is $(n_k; n_h)$ (4;5) according to the AIC approach. This set is different from the successive steps previously defined using the investigation about the dtw score.

Tonic bursting :

In the same way than tonic spiking, the SUBFIGURES showing LL and AIC scores are

2.7b and 2.7d, respectively. For this behaviour, the heat maps seem to be the same, then the conclusions provided by AIC are identical to the ones of LL. The only minor difference that could be noticed is the reduction in contrast as the number of basis increases, which indicates that increasing number of basis improves the result quality less then before. Therefore, if the figures hag shown many more results, the AIC should decrease. Consequently, the best combination of basis would be the couple (x, y) with the highest score. This conclusion is made by assuming that increasing the number of basis has no negative impact of the score.

To conclude about the best number of basis, different criteria must be used, to consider different approaches. For *tonic spiking*, some candidates could be directly identified using AIC which takes into account the complexity of the problem. In opposition, for the bursting behaviour, only used LL or AIC is insufficient to select potential candidates. For these reasons, other approaches have been employed, such as the aggregation of results to select the best number of basis in filter k, comparison of filter thanks to dynamic time warping, and LL/AIC score.

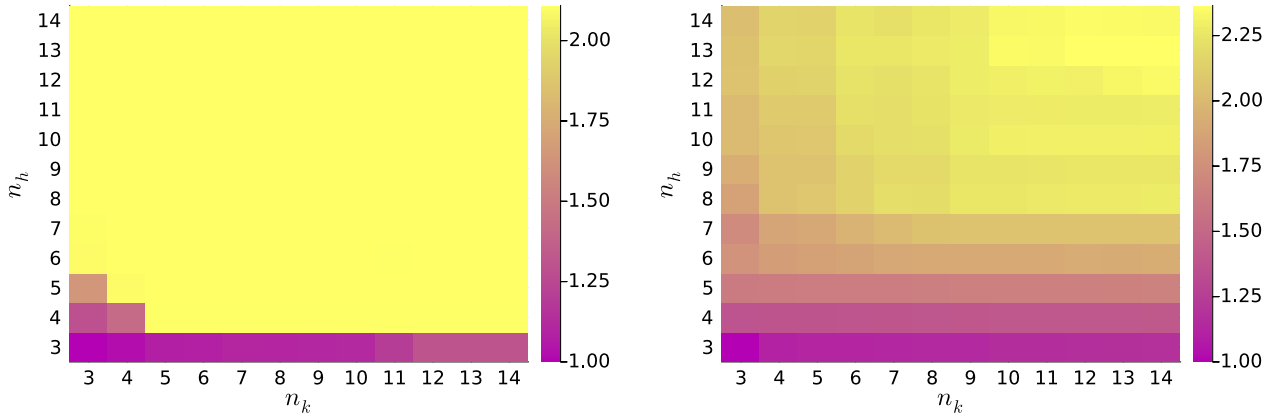
2.3.4 The design of the set functions designing the filters

As it has just been seen, the design of basis has a significant impact on the generation of filters. So far, the goodness of fit has been studied using the log-likelihood, which is a common measure in this context. However, analysing model is important, but ultimately, analysing the predictions is better.

To achieve this, a set of homemade basis function is introduced in the design of h-filter. It is specially designed to be as general as possible, and is composed of 7 elements. This choice comes from the previous conclusions. Additionally, a new scoring metric, called $TriangularScore$, symbolised by TrSc. This score is still a homemade score, which is computed as follows:

- *Template generation*, it generates template of a particular firing sequence. A triangle of height 1 and base $20ms$ is centred at each spike location. In the case where triangles are in superposition, the highest value is kept.
- *Application of template*, it applies the template generation to both the Izhikevich, and the prediction's GLM sequences.
- *Score calculation*, it computes the difference between these two templates to obtain TrSc.

This new method of computing the score of spikes sequences is applied to GLM predictions, such as the ones available thought the FIGURES A4 in the APPENDICES. The number of basis is fixed in this set of homemade basis, the evolution can be display across n_k. It is available in figure which compared both LL, and TrSc. These two measurements are metrics evaluating different aspects. The LL assesses the correctness of the model, instead of the



(a) Tonic spiking - LL

(b) Tonic bursting - LL

(c) Tonic spiking - AIC

(d) Tonic bursting - AIC

Figure 2.7: Evolution of the LL score and AIC for the tonic spiking and bursting behaviours according to the number of basis in k and h filters (n_k and n_h respectively).

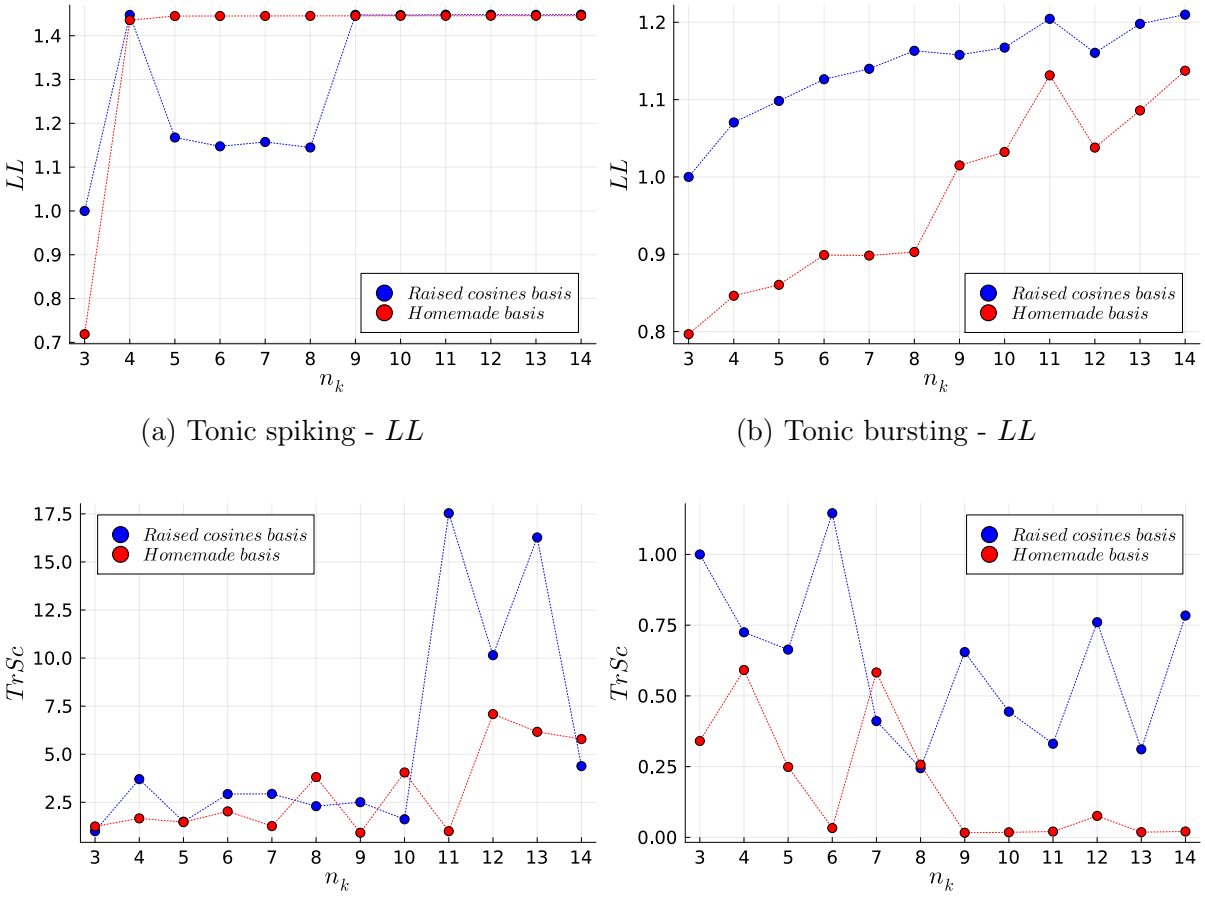
TriangularScore measures the similarity between two sequences. Note that all the GLM are still trained thanks to the LL measurement, then the provides values are the last value of the LL or the TrSc computed on the generated sequences.

Using FIGURES 2.8a and 2.8c, the conclusions are relatively the same. The models labelled "homemade basis" are similar or superior to the models using cosines raised basis functions. The only difference is in the areas which are localised from 5 to 8, and from 11 to 13 according the LL and the *TriangularScore*, respectively. The homemade model outperforms the other according to the considered metrics.

In another perspective, the FIGURES 2.8b and 2.8d can be used to conclude contrasting solutions. In the first graph, the curve corresponding to the cosine raised basis function is the highest, which mean that their models are assumed to be superior to the homemade models.

Otherwise, in the second figure, the models having homemade basis functions provide scores closer to zero compared to the models used so far. This indicated that the new models can be considered as better than others according to this particular metric.

By inspecting FIGURE A4 which illustrates GLM predictions using the specific case $(n_k; n_h) = (7; 7)$, a visual conclusion can be done. The bursting sequences generated by the GMLs appear to be closer to the Izhikevich sequence for the new version of basis. Consequently, it is essential to be aware that the LL measurement is just one way of model's evaluation as many others. It delivers only a general indication about the model accuracy, *e.g.* if two models have similar LL, nothing ensures that A-model is better than B-model, and *vice versa*.



(a) Tonic spiking - LL

(b) Tonic bursting - LL

(c) Tonic spiking - $TrSc$

(d) Tonic bursting - $TrSc$

Figure 2.8: GLM measurements for the two tonic behaviours.

2.4 Features of filters according to behaviours

Determining the behaviours of neuron without directly considering the sequence can be quite challenging task. However, identifying specific patterns related to specific behaviours could automatise the classification process. The paper used as reference [18] performs classification using the amplitude of filters k and h; but here, a other approach is used.
Starting by studying the deformations of stimulus filter and feedback filter; k_7 and h_7, respectively. These filters represent the best solutions according to the previous discussion in Section 2.3.2 Number of bases in the set of functions designing the filters, assuming that $n_k = n_h$.

The k and h filters are available in the APPENDICES on FIGURES A2 and A3, respectively. Delivering a rigorous interpretation is a hard task due to the variability of the input. Indeed, the intensity of stimulation differs for each behaviours in the Izhikevich model. Therefore, the discussion on filter k can only be led in terms of filter shape. Before starting the interpretation, it is important to understand that positive values on filter indicate that the probability to obtain spike increases, and conversely for negatives filter values.

Interpretation of k-filter
This analysis is particularly relevant for the first train of spikes, as in the area, modifications occur in the stimuli filtering, which will not be the case beyond 100ms. This time value refers to the length of k-filter, *e.g.* after 100ms, the value obtained by the product of k and stimulus will remain constant until the end of stimulation.

From phasic spiking to tonic spiking: The delay to obtain the first spike has decreased. For the phasic case, the duration to obtain a positive value after the beginning of the stimulation is about 20ms, while it is few ms (≈ 3) for the tonic spiking case. Another relevant difference is integral of these filters. The area under the curve is very interesting to investigate, as it informs if the model is able[1] to generate spikes after the first train. For the phasic spiking, its integral is around -27, versus 20 for the tonic spiking, which is expected in terms of positive and negative values.

From phasic bursting to tonic bursting: Similar observations can be made in terms of delay to obtain the first spike after the beginning of the stimulation. It is around $0.5ms$ and $15ms$ for tonic and phasic bursting, respectively. These time values do not mean that the probability is highest at this moment. The only meaning is that it is more probable to spike than the null case, which is represented by no stimulation at all. In reality, the spike should occur approximately $25ms$ and $8ms$ after the initiation of the stimulation (for phasic and tonic bursting, respectively). These number come from the highest increase of probabilities. About the integral point of view, the tonic bursting is still positive as asserted, and pha-

[1] in a probabilistic point of view

sic bursting is negative, with 2.8 and -56 as values. The additional difference which is also present in bursting cases, is the positioning of the double top. For phasic bursting, it exists a negative region between them although this is not the case for the tonic bursting behaviour.

From tonic spiking to tonic bursting: The only observation that can be manage is the stretching along the time axis, allowing to see that the filter stays longer in the positive area for bursting than for spiking. This duration does not permit to distinguish spiking versus bursting, but it could serve as an indicator.

Interpretation of h-filter
This study is much more relevant for the tonic behaviours as it provides insights into the refractory periods of neuron.

For tonic spiking: This filter allows to investigate of the refractory periods. Knowing the h-filter increases the probability of obtaining spike when a spike is expected; the location of this increase can be used as a reference in the extraction of the refractory period.

For tonic bursting: Similar to tonic spiking behaviour, the refractory period between trains of spikes could be visible in the h-filter. This is designed by the small peak after the well, which will be more clearly illustrated in the next section. Furthermore, when trains of spikes are close to $-100ms$, the probability to regenerate train increases, as the curve values are positive (not visible in the figure). This concept will be detailed in the following section, and figures will be more clear.

Summary of interpretations
Exploring the deformations on filters between different behaviours could be a solution in the neuron classification, but this task is still challenging due to variability of responses in a same set of behaviour. For example, the refractory period of a sequence impacts the shape of filters. Therefore, these shape are specific to the basis used, and other sets of basis could provide filters which are more or less different.

The Section 3 Generalised Linear Model as classification tool studies the filters deformations for a same set of behaviours having different characteristics, as well as neuron type classification according to the feedback filter h.

2.5 Degradation of solutions for phasic behaviours

Intuitively, adding more basis function increases the accuracy of the results, but it is not as simple as that. As a reminder, the set of basis corresponds to raised cosines basis defined as the Equation 2.1, which means that all basis undergo deformation when an additional one

is added. This deformation can be clearly seen in FIGURE 2.3; *e.g.* the third basis of each set is different of the others. Furthermore for instance, the top of the third base is reached at 40 and 5 $[ms]$ approximately for h_4 and h_{10}, respectively. Although a higher number of basis functions creates a finer mesh with a capacity of extract particular and precise behaviours, it still has some drawbacks. Increasing the set of basis functions means reducing the size of the filter, *e.g.* a set of 4 basis creates goes from $0ms$ to around $1200ms$, while 10 basis set goes only to $200ms$. This means that the size of filter is depends on the number of basis function. However, the duration separating the first and the last peak of the basis functions remains constant. Then increasing the number of basis would not decrease the quality of the model.

Therefore, to understand the cause of the decrease in quality in the model, other elements must be taken into account. The solver could be a reason for such differences between simulation, depending on the path followed to achieve the optimisation. It could deliver a solution located at a region plateau of the score, then the optimisation process doesn't achieved to improve the model in the number of fixed iteration. Nonetheless, this reason is ruled out as the possible cause in regards of FIGURE 2.9. If this assumption was the cause, the results obtained on different computers would be the same, which is not what is observed.

The other cause could be due to the computer device itself, which appears to be the true reason. As shown in FIGURE 2.9, the log-likelihood measurements differ according to the device. These differences come from when the data are managed in the processors. Depending on the architecture of a processor, various factors such as instruction set design, data pathways, cache management, and parallel processing capabilities can affect the performance and accuracy of computations. Moreover, variations in the manufacturing process can introduce variability between devices, even of the same model. This highlights the importance of understanding the hardware characteristics before analysing the computational results. However, the difference particularly affects the way of storing high numbers. The model must compute high exponential value, as $exp(700)$ and much more. These calculations can push the limits of numerical precision and stability, especially given the finite precision of floating-point representations in computer processors. The choice of floating-point precision (single, double, or even higher precision formats) can significantly impact the ability to manage large numbers. The double precision in floating-point format is often used in scientific computing because it offers a larger range and better precision compared to single precision. All the data shared in this thesis have been computed using this double floating-point representation, usually stored on 64 bits, also known as Float64 type.

Additionally, to confirm this cause more comparisons between the devices of FIGURES 2.9c and 2.9b have been done. First, the case $(n_k; n_h) = (4; 3)$ has been studied, the solutions provide by the GLM are identical, however the associated score is different, -73.5 versus -57.4 respectively. Second, investigating on the case where a 'good' solution is found on in one side contrarly to on a other computer, *e.g.* $(n_k; n_h) = (5; 13)$. Given in input the solution of the bad computer to the good one permit to approach of the optimal LL without reached it.

This problem is known as the *overflow error*, and occurs when the solution to an operation is incorrect. It is due to the limited range of value than a variable is able to take. For instance, the solution of the operation

$$10\hat{}10\hat{}10$$

is a very large number. However, the response of such operation is 0. Since then, as very large number in the GLM optimisation must be managed, the score provide by the informatic language can be badly evaluate due to this kind of errors. In regards to the score evolution, the *overflow errors* can effect the shape of the score evolution into two ways. It could create local minimum or truncate the log-likelihood. In the two case, the optimisation is not able to adapt the weights of the GLM parameters anymore which create a GLM badly designed.

Therefore, specialised hardware, such as tensor processing units (TPU) and graphics processing units (GPU) are designed to handle large-scale numerical computations more effectively than traditional central processing units (CPU). These devices often include specific optimisations for high precision arithmetic, which can be crucial for models requiring frequent computation of large exponential values, it is typically the case for the GLM. Then, it could be relevant to perform the simulation on such processing units to compare the results obtained. To sum up, finding a way to allow the high level of precision in each number could be a relevant alternative. Another approach would be to use a modified-log-likelihood needing less extreme values to be computed easily.

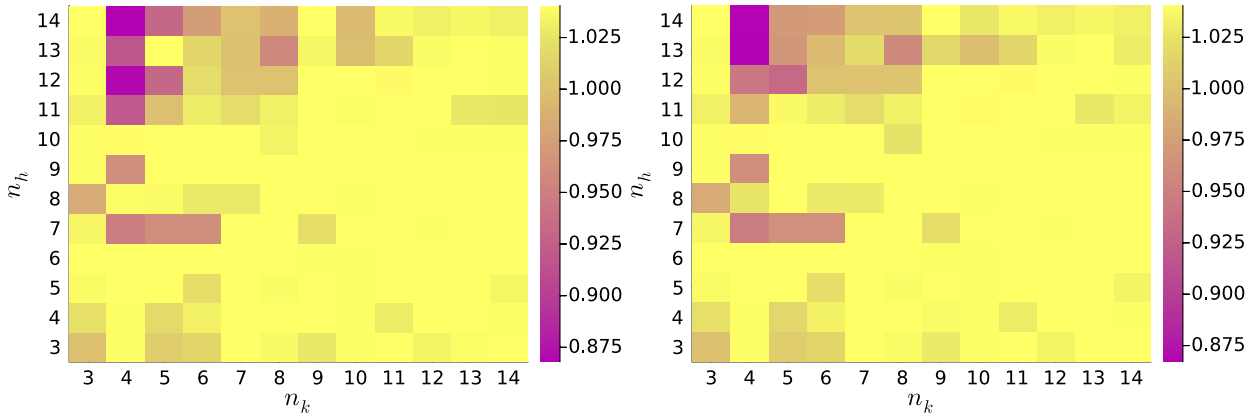
Note that the results are always the same when simulation are performed several times on same computer.

2.6 Initialisation of the GLM

The initiation phase of the weights of functions constituting the filters of the GLM, as so as the offset value is highly suggested to avoid converging to undesired solution due to *overflow errors* and/or numerical errors.

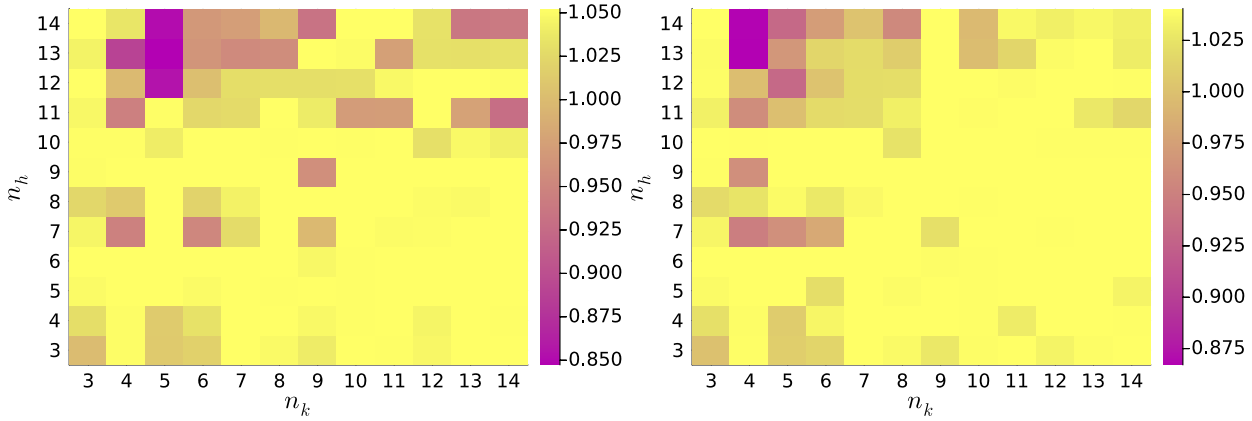
Before showing the presence of such solutions, called local solution in the following, it is important to remember that the score function used to measure the quality of the model is the log-likelihood, then higher is better. However the code implementation try to minimise the negative log-likelihood, noted nLL, then lower is better. In the following figures of this chapter, this nLL is used; and to stay in line with the previous heatmaps representation, the lower values of the nLL will ba localised in the purple area (resp. the higher value in the yellow area).

To show the presence of these local solutions, investigate about an easy way as the following :



(a) Windows 11 Familly 23H2 : Intel CORE i3-10110U CPU @ 2.10GHz - x64

(b) MacOS Big Sur 11.7.10 : Intel CORE i5-4260U CPU @ 1.4GHz - x86_64



(c) Linux 5.15.0-88-generic : Intel CORE i7-6700 CPU @ 3.40GHz - x86_64

(d) replit.com : AMD EPYC 7B13 @ 3.05GHz - x86_64

Figure 2.9: Evolution of the LL score for*phasic bursting* fitted on different computers.

- fitting a GLM in a traditional way, *e.g.* $n_k = 7$ and $n_h = 7$ (the found elements are noted with the * symbol),
- using the k-filter as reference,
- studying the evolution of nLL according to a linear stretching of h_7, meaning that the proportion between the weights of basis function are kept, and the offset value.

Therefore, if the presence of local solutions are demonstrated for a so easy case, it would exist many others solutions in the entire space, meaning the numerical errors must not be neglected. The FIGURE 2.10 shows this evolution. It exists 7 local solutions in the tested range of values where 3 of them have a positive value. It leads to two elements. First, the presence of local solution due to numerical errors must not be neglected. Second, initialising

all the weights to 0 is a judicious choice, allowing to avoid the majority of local solution. Indeed, using a such of initialisation generates a nLL of n (sequence length), which is updated due to time consideration making it equals to 1 for 10 seconds of recording. Then if the score function should converge to an unique solution, in practice it is not the case due to hardware design.

Then as nLL can easily be set between 0 and 1, it means that the majority of local solutions displayed in the associated figure can be avoid. It makes the initialisation to the weights at zero fully-sensed and justified.

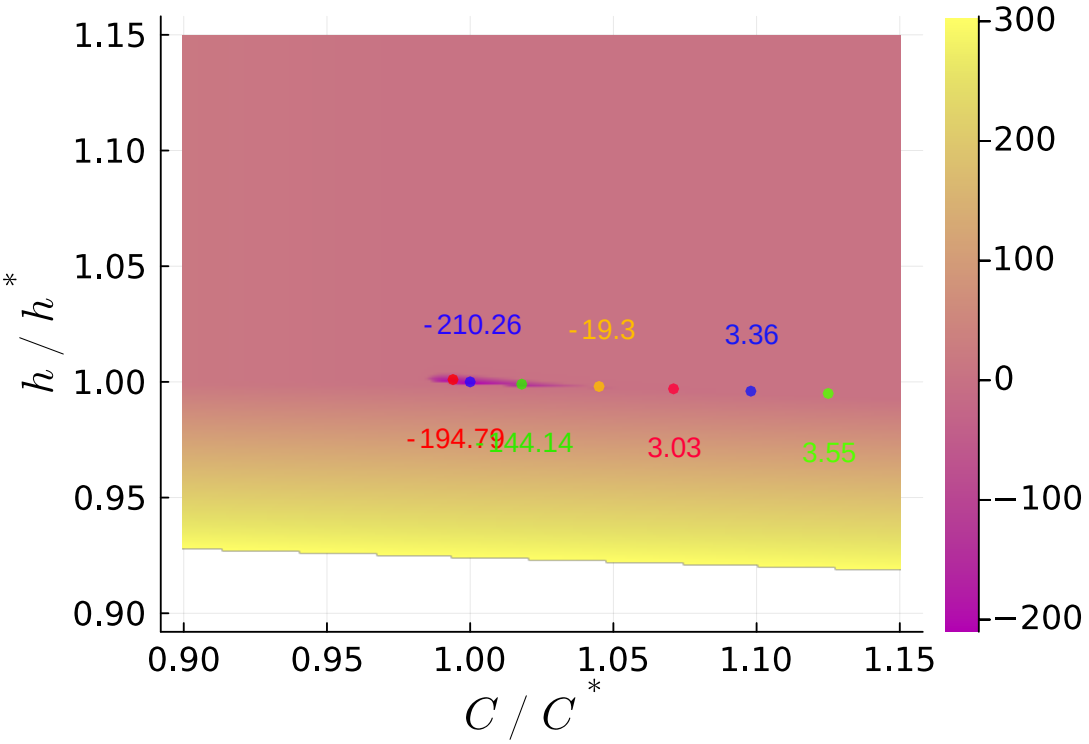


Figure 2.10: Evolution of the negative log-likelihood for the **tonic spiking behaviour** according to the linear stretching of h-filter and the offset value C. The nLL being too large the be compute are uncoloured. The dots are the local solutions associated with their own score, where the scores higher than 1 have been resize as $log(10, score)$ to facilitate the visual interpretation.

2.7 Conclusion

Designing a GLM for a specific task it not so simple as it my seems, *e.g.* generating neuronal patterns. It depends on lots of parameters that must be thoroughly studied beforehand. It goes from the GLM architecture to the impacts of its parameters, and even the score function

fitting it.

This chapter has delved into the consequences of some of these parameters, bringing a detailed analysis such as evaluating LL by varying the sequence length, the design of basis functions; and a dtw analysis to find to most suitable filter. The general architecture of a GLM is determined by the tasks that must be performed; *e.g.* in the case of reproducing neuronal behaviours, a feedback filter is mandatory to closely match to the true observed behaviours. For the nonlinear function, for instance, various sources have been taken into account as justification for such a choice. Oppositely, in determining a good combination of bases functions, which is a trade-off between accuracy and computation time, many simulations were executed on tonic behaviours to find it, resulting in the selection of the couple $(k_7;h_7)$.

However, the obtained model for some behaviours can be dependent on the computing device itself when high values must be stored. The limitation of the log-likelihood score function has been underlined, for phasic behaviours. In the one hand, problem occur in the evaluation of the logarithm function at zero; and the other hand, managing high numbers generate overflow errors being processor-dependant. Therefore, several approach must be take into account to be able to evaluate this kind of problem.. First, using devices that well handle large data is preferable. Second, training different GLM and comparing the evolution of results could ensure that the computational errors do not affect the optimisation process. Third, running the model on different CPU to increases trust in the results if those one are similar; of still better, using GPU at TPU. Fourth, finding another way to evaluate the accuracy of a model to avoid to manage large numbers.

In the following chapter, all the results come from a single device as no problems related to large numbers have been found for tonic behaviours.

Chapter 3

Generalised Linear Model as classification tool

The neuromodulation is defined as *"the alteration of nerve activity through targeted delivery of a stimulus, such as electrical stimulation or chemical agents, to specific neurological sites in the body."* according to the International Neuromodulation Society [24]. Whatever the way to modulate the neural activity, it results that the firing sequence is affected to those changes. In the neuron's level, the dynamics of channels are affected, which it translates through the expression of conductances the mathematical models.

About the sequence modifications in itself, two kinds of alterations are subject to occur. First, the interspike interval could increase or decrease. The concept of interspike interval is often called ISI is the literature, and is mostly used to nonbursting sequences. But the parallel can be done for bursting sequence, representing the duration between two trains of spikes. That leads to the second element, the modification of the intrinsic behaviour. In others words, the change of the number of spikes per train; one spike per train is spiking, two spikes per train is light bursting, *etc.* Those alterations can be difficult to identify without looking the sequence itself. It is here that GLM can be useful in the understanding of neuromodulation effects.

As previously seen, GLMs are able the capture the information about the firing patterns thought feedforward filter k and feedback filter h. The sequences to capture are the only cause of the modifications inside filters. From this point, studying the alterations of filters coming from neuromodulated sequences could be a approach paying off. Thanks to the large set of behaviours that a well designed GLM is able to catch, even small differences in the sequences could be detectable with a simple glance on filters.
e.g. Two similar sequences where the only difference is about the ISI. This modifications should be translated in the GLMs feedback filter by an increasing of the duration of the negative part of the h-filter. The second modification that could be occur is the value of the constant which is added before passing to the nonlinear function f.

Consequently, the effect of neuromodulation can be study using GLM. It allows to perform comparisons easily between sequences in an automatised way. However, this tool performs analysis only on the sequence itself, therefore it does not provide information in the cellular level like the opening rate of ions channels.

One other topic that could be explored using GLM and neuromodulation principles will be to take the GLM generated sequences as references to construct robust conductance-based models. Indeed GLM used a stochastic process, it generates different sequences across the trials. Therefore, matching those simulations with model resulting of ODE could be an approach allowing the increase the knowledge on this fascinating topic. Note that this last chapter focus only on the investigations about the filter deformations and the sequences classification according the behaviours.

3.1 Arthur Fyon's research

The rest of the master's these is based on the work of A. Fyon, a PhD student in the department of Electrical Engineering and Computer Science at the University of Liège. [7] In a first time, a quick overview of his work will be done to well understand the parallel that can be made with GLM.

3.1.1 Brief overview

To better understand how spiking activity results from the interplay of ion channel openings and closings, A. Fyon delves into investigating the influence of neuromodulators. These investigations link the cellular level (*e.g.* impacts on conductance) and the firing pattern level. This presents a challenge due to the diverse array of ion channel types. The abundance of different ion channels necessitates considerations about the numerous dynamics in constructing a realistic model. Consequently, there are as many distinct dynamics as there are types of ion channels. As a result, the firing pattern, as the observable aspect of neuronal behaviour, emerges from the combination of these numerous dynamics.

However, one particular firing pattern can emerge from different behaviours in the cellular level. In other words, different sets of ion channels dynamics can generate same firing behaviour. This phenomena is called *degeneracy*, as shown in FIGURE 3.1. Therefore, when neuromodulators are introduced in the system, the properties of channels are affected and can provoke modifications in the firing pattern. This intricate interplay between ions channel degeneracy and neuromodulators occurs in a complex way which is deeply investigated to design particular chemical species robust to the ion channel variability.

To model a large set of behaviours, A. Fyon uses a conductance-based model, so-called

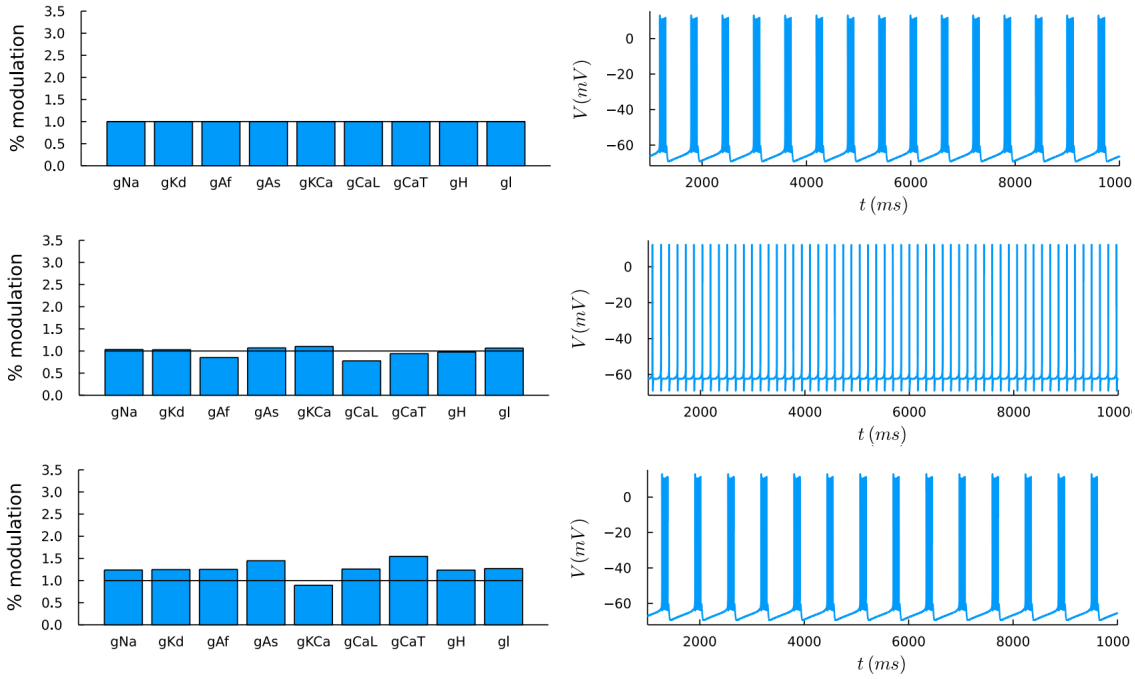


Figure 3.1: Visual example of the degeneracy phenomenon. [25]

CBM. The model being used incorporates multiple conductances expressions which are defined by two gating variable at most. Those gating variables express the opening or closing rate for each ions channels. The advantage of a such model is that it well represents the relations between each ions channels, then discussions can occur regarding the biological point of view. Otherwise, this kind of model are difficult to manage in an analytical resolution. Since then, this complex CBM can be reorganised as a feedback model.

The Figure 3.2 illustrates how a CBM can be translated into a feedback model. The entire set of branches modelling the effects of ion channel can be placed in parallel inside the controller element. As reminder, each ion channel has its own dynamics, but those dynamics can be quite similar. Therefore, the large number of parallel branches inside the controller can be aggregates in only three neuronal feedback gains. These three branches represent the dynamic input conductance (DIC) and differ only by their timescale. They model the fast, slow, and ultra-slow components with expressions being voltage-dependant. These three new aggregated conductances are noted $g_f(V)$, $g_s(V)$, and $g_u(V)$ respectively. [26]

Performing a reduction of a high dimensional problem to a smaller one has advantages but also drawbacks. This new neuron's representation is easier to manage in term of intuition to generate specific firing pattern. Moreover the effects of neuromodulators on the neuron response is also easier to interpret, *e.g.* a neuromodulator A impacts only the slow neuronal feedback gain. However, this reduction can lead to a loss of information. This quantity of

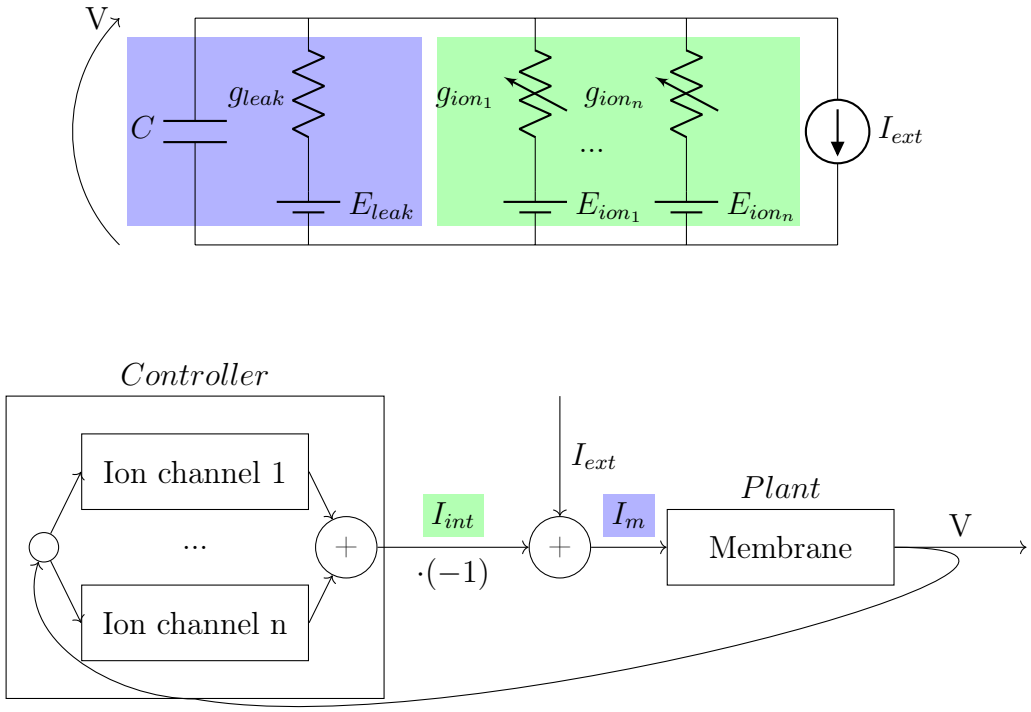


Figure 3.2: Conductance-based model and the related feedback model. [27]

loss depends on the chosen way to reduce the system. In this case, the DICs come from a linear combination of all the maximal conductances of the initial model, where this linear combination depends of the neural type. The way of combining information differs between a stomatogastric ganglion neuron and a dopaminergic neuron. Furthermore, the presence of neuromodulators affects the expression of dynamic input conductances, with these changes depending on the species involved and its concentration.

The simulations lead by A. Fyon have showed that degenerate population having different initial conditions of CBM delivering a same behaviours. The ion channel conductance are tuned by the neuromodulation effect which is neuron-dependant. The neuromodulator changes the behaviour of the neuron in an identical way among the population.

About the links of this reduced model and the GLM itself

The reduced model uses three different dynamics to represent a particular firing pattern : fast, slow, and ultra-slow. The GLM generates filters through a linear combination of a set of bases. This set is composed on different functions capturing information at different time scales. Therefore, the bases of GLM can be seen as elements representing different dynamics, like the reduced model discussed previously, *e.g.* the weight associated to the base A represents the fast dynamics, *etc.* Since then, being able to make the interpretation on GLM

filters can help to analyse sequences in more convenient way than the classical trace V.

To make this investigation, the models developed by A. Fyon have been used to generated the spikes sequences. Those models referred to neurons previously mentioned : stomatogastric ganglion neuron and dopaminergic neuron, noted STG and DA respectively.

3.1.2 Generation of firing sequences

The STG and DA models are able to produce different sets of behaviours as spiking, light bursting and strong bursting. However, the variability in a particular type of firing patterns that model is able to generate is specific to the neuron type.

As an example, let's investigate about the tonic spiking population. For the STG model, the model response generates patterns with interspike intervals, so-called ISI, that are quite variable. In contrast, the mathematical representation of the DA neuron generates firing patterns with ISI being more constant in term of relative difference. *e.g.* for a predetermined set of parameter values being neuron-specific, the interspike interval of the pattern obtained by STG model varies from $45ms$ to around $90ms$. It is a difference of 100%. While the model of DA neuron has only a relative difference of 20% (from $500ms$ to $600ms$). Therefore, a higher variability between filters should be observable for neuron of the STG type.

This variability comes from the model definition which tries to represent the most faithfully possible the true behaviours of the neuron. The dynamics describe by the neurons are presents in the Appendices by the Equations 3.7, and 3.8 for STG and DA models respectively, where g models the conductance, m and h the dynamics, and τ the time constant. Those systems come initially from papers [28], [29].

Regarding the generation of specific behaviours of DA and STG neurons, various methods could be employed, each with their own characteristics. The chosen way is to utilise some fixed neuronal feedback gains to achieve desired behaviours for the neuronal population. These values modulate the responses of neurons by influencing the overall dynamics of the system. The method is a relevant trade-off between the precision and computing time. Indeed, others methods are much more accurate but require lots of computational resources, *e.g.* the random search provides results highly reliable, but it can take more than one hour for the generation of 200 neurons. Furthermore, the potential occurrence of unexpected neurons can by relevant to study with GLM. The end of this chapter will discuss about that.

3.2 GLM design

To study and draw conclusions as rigorously as possible, some modifications in the GLM must be made.

The first modification being that filter k is neglected, which allows to focus deeper in the investigations of the feedback filter design. This adjustment is particularly relevant in this case, as the models do not require stimulation to produce firing patterns. Thenceforth bypassing the k-filter in the architecture previously used is trivial. It simply needs to deliver a null stimulation of the GLM.

The number of the bases functions used to design the filter h is established as 7. It has been determined being the best one in the previous research. The number remains the same regardless the behaviours that is reproduce, which offers two significant advantages. Firstly, having the same number of bases allows to facilitate the comparison of filters, as the only variable factor is the combination of bases functions. Secondly, it is about further investigation that will be address at the end of this thesis : the behaviour classification based on filters. For a meaningful classification, it is essential that the GLM design being common across all behaviours. In the opposite case, it would mean that the behaviours is known in advance, which has no sense in a classification problem. For these reasons, the number of bases inside the design of feedback filter is fixed.

However, only one modification is permitted : adjusting the time windowing of the filter, which is neuron type-dependant. This parameter is modelled using another parameter which is the gap between the first and the last functions of the raised-cosines bases. The choice on this value is user-dependant, and to be able to perform it properly the user must have knowledge about the studied neuron. Typically, users determine this parameter by examining the range of interspike intervals observed during tonic spiking behaviour. Its value is approximately twice the ISI value, *e.g.* the ranges of interspike intervals are from $40ms$ to $80ms$, and from $500ms$ to $600ms$, for STG and DA respectively. Since then the value of the parameter in the design of filter is fixed to 150 to STG neuron and 1000 for DA neuron. Therefore, the study of neuronal behaviour for neuron having a large variability in the ISI can be difficult by this kind of GLM. One solution would be to significantly increase the number of bases, or to add another filter h as feedback loop having differences in its windowing. However, those modifications will increase the complexity of the GLM which is one element that it is tried to be avoid.

The others elements of the GLM remain unchanged from the ones used previously. The nonlinear function f stays the Equation defined in 2.2. Additionally, the last GLM block determining if a spike occurs is still defined by the $\mathcal{P}$oisson distribution. As reminder, the probability rule adapted to this particular problem is expressed through Equation 2.5.

3.3 Filters deformation

In this section, deeper investigations are leads to understand the how variations of firing patterns affect the design of the h-filter. The trace used to generate sequences fitting the

previously defined GLM come from the models of STG and DA neurons. The behaviours generated are classified into three categories : spiking, light bursting, and strong bursting. The light bursting refers to sequence having two spikes per train of burst for the STG model, whereas it is considered as three for the DA neuron. Consequently, the strong bursting case is the one containing at least more spikes than the light bursting.

As already suggested, the fitted GLM on the DA model will be easier to interpret due to the weak variability between sequences of a same set of behaviours. Thus, the general shape of filter according to the behaviour is the first element which is studied before going further.

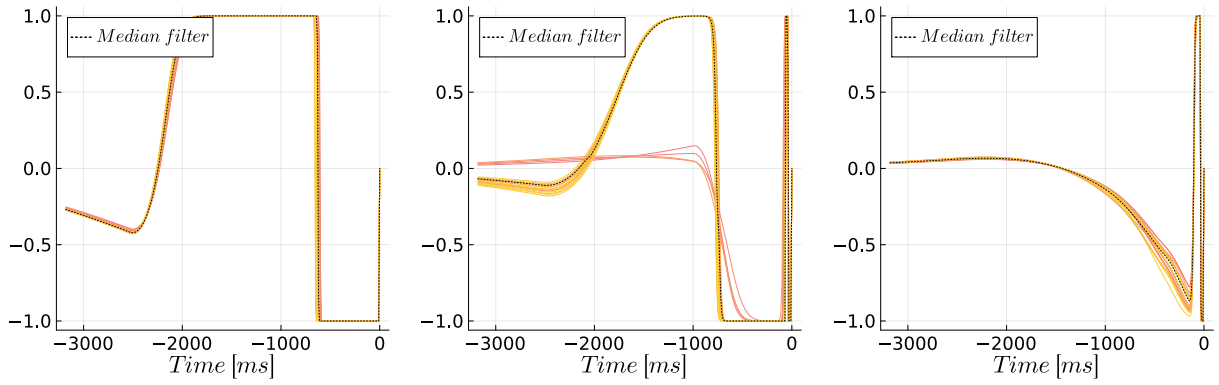
An important remark to make concerns the labelling of sequences, which is initially assumed as true but is not necessary the case. It is due the the chosen method of the generation of sequences (fixing neuronal feedback gains). All discussions will be accompanied by figures using, in the majority of cases, a colour code. This colour code refers to the among of spikes in a particular time window ranging from yellow to red as much as the previous defined quantity increases. However the colour scale varies across figures, therefore it does not allow a direct comparison between different figures.

Depending the part of sequence which has already be generated by the GLM, the feedback can be positive or negative. It means that the probability of generating spike at time t is high or low, respectively. The sign of feedback is determined by the shape of the filter h, if it delivers a positive outcome, the probability to obtain a spiking event would be high, unlike a negative outcome. However, an offset value is added before passing in the nonlinear function f. Due to its low value in comparison to the h-filter amplitude, this component is neglected most of the time. Details on this offset value according the behaviours are available in the Appendices in Figure A8.

To discussed in term of positive and negative feedback, all filters have been applied to a *hyperbolic tangent to facilitate the interpretation*. The filters related to spiking, light and strong bursting are shown in Figure 3.3 for the DA neuron. In a first time, the global shape of filter is not study to the STG neuron because of the filter shape are less homogeneous. Note that some filter have been resized in the time-axis to allow more comfortable visual analysis. The following interpretations assume that the full filters are the ones displayed across the figures.

To investigate the shape of the filter according the behaviours, the opposite approach can be performed by attempting to establish the sequence that should be generated.

Spiking The simulation begins by generating one spike thanks to the offset component. Immediately after having this first spike, the GLM does not produce any more spikes during several hundred of milliseconds. This no-spiking period is caused by the negative feedback



(a) Spiking (b) Light bursting (c) Strong bursting

Figure 3.3: GLM feedback filters for three different behaviours. Each curve represents the h-filter for an individual neuron.

generated by the filter h. After this duration, the previous spike reaches the positive area of FIGURE 3.3a. It means that feedback become positive, then the GLM will produce one spike again. After the generation of the second spike, two spikes have be generated in the last $3000ms$, which is the time window of this filter. Therefore, those two spikes must be take into account for the next generation by the GLM. The spikes, labelled as s_1 and s_2 respectively, have opposite effects on the feedback : s_1 is in the positive area while s_2 is in the negative one. However, the first negative area is must more negative that the other area is positive, it is the negative feedback that overcomes the feedback battle, then no event is generated at this time. This feedback battle continues by producing the same result as long as a spike is present in the first negative part. Therefore, the next spike occurs when the last generated spike leaves the negative feedback to enter in the positive one. Since then, the feedback battle restarts with the predominance of the negative area, and so on.

Light bursting The bursting are more complex to investigate, especially in the number of spike per train. At the beginning of the simulation, one spike is generated thanks to the offset element. To evaluate the sequence of further spikes, the h-filter must be consider in particular the filter labelled as median. The others filters which do not follow the median one are misclassified in reality. After the initial spike generation, it comes to the first and thin negative area where no spike will be generated. After a few milliseconds, the only generated spike enters to the thin and positive part, meaning that the probability for obtaining a spike is high, than the GLM produces one more. At that precise moment, two spikes are generated, one being in the positive and thin region of filter, and the last one in the first negative and thin feedback. A new battle of feedback ensues. In a first time, it is the negative part that wins as the intensity has much more impact than the feedback positive intensity. It is necessary to wait some supplementary milliseconds to obtain the second case of feedback battle. At this moment, the first spike is in the second negative feedback, while the second spike is in the first positive and thin feedback. The value of those region are similar, however

this new battle is won by the positive one. This is due to two elements, first the positive region has more impact that the negative one; second, event if the offset value is small, it is positive and play an important role in this battle. Afterwards, there are three spikes in the simulation, the two first being the the second negative region, and the third in the thin and negative part. Then no spiking occurs. A handful of millisecond latter, this last spike arrive in the thin an positive feedback region, provoking a new feedback battle. But this time, it is the negative side that will win. Indeed, two spikes are present in the negative side what is enough the overcomes the weight of the positive feedback generated by the offset value, and the filter. Since then, no spike occurs while the first train of three spikes stays in the second negative feedback. However, when the train arrives to the last positive past, one new spike can be generated thanks to the spiking probability that has increased. The process can restart as this last positive area has negligible intensity compared to the others.

Strong bursting The h-filter, as referenced in FIGURE 3.3c, has a global shape similar to the filter labelled as light bursting. However, the differences are twofold. First, the second negative and positive feedback areas are weaker than previously. This second modification has less consequences than the other. Indeed the second positive feedback impacts only the train generation which be generated thanks the number of spike into this particular regions being helped by the offset. Second, the second negative area referred to the number of spikes containing in a train. Due to this modification, more spikes are needed to counteract the positive feedback effect generated by the last spike. Then each train contains much more spikes then previously.

To be as comprehensive as possible, further investigations are conducted on both STG and DA neurons.

3.3.1 Spiking

As already discussed, the negative values of the h-filter reflect the period during which no spiking event occur. These value can be compared with the ISI extracted from the original sequence itself. The FIGURE 3.4 below illustrates the filter h for each neurons (DA and STG), and compares the obtained ISI. One coming from the spike sequence, and the other of the h-filter.

For the sequence modelling neurons as DA-type, the correlation between the real ISI and the ISI coming from h is quite high. Indeed, it is contained from 82% to 91% for low and high values of the ISI, respectively (where low and high refer to 500ms and 600ms, typically). For the h-filter generated thanks to STG model, the relation between those two values of the interspike intervals is less clear. It appears that certain sets of filters exhibit high correlation, particularly those where the extracted ISI values are below $70ms$ (both form h-filter and from the sequence). The difference in correlation between filters coming from the same model of neuron can derived from two cases. First, the sequence that is fitted by the GLM does not contain spiking behaviour, then the sequence is misclassified. Second, the sequence starts

with light bursting and degenerates to spiking. To highlight the true spiking sequence, a simple comparison can be performed on all the ISI of a sequence. The FIGURE A9 distinguishes this comparison, where the blue dots are linked to sequence having a standard deviation lower than $25ms$. This threshold value, which is user-dependant, permits to classify the sequences in two sets. Therefore, by only looking this new set composed of blue dots, the correlation between the ISI coming from the sequence itself and the ISI coming from the h-filter appears as being higher.

However, this correlation between those ISI could still be improve. In a first step, the TABLE 3.1 sums up the ratio value between the ISI according to the elements taken into account. Then the research of the ISI provides better results if the offset is considered, however only considering the filter h is more convenient for an user is the inspection is only done visually. Note that the offset value can be taken into account visually by shifting the h-filter by the offset value. This trick is not correct in a mathematical standpoint, but as changes in feedback are rapid, it is negligible.

	Considering h	Considering h and offset value
DA	from 82% to 91% → FIGURE 3.4c	from 85% to 93% → small differences
STG	from 90% to 110% → FIGURE A9	from 90% to 110% → negligible differences

Table 3.1: Differences of the ISI coming from the GLM and the ISI computed from the sequences. About the FIGURE A9, only the blue dots are considered.

Another explanation for the mismatch of these ISI is due to the threshold value chosen at which spike occurs. By default, it has been fixed to 0 to be able to discuss in terms of positive and negative feedback. However in reality, it is much more complex than that. In the EQUATION 2.5, the probability to obtain a spike is defined as :

$$P(spike) = 1 - f(-\lambda(t))$$

In terms of computational resources, it means that a spike occurs if a random number is lower that this probability. Knowing that the process is repeated every $0.1ms$, the following assumptions in made : "A spike will occur when the probability of obtaining one is superior than 0.5 during a time interval of 10ms.". Thus, it is equivalent to set the spiking probability of 0.007 for 0.1ms ($1 - 0.5^{1/100} = 0.0069 \approx 0.007$). By injecting this expression into the previous one, it successively gives

$$0.007 = 1 - f(-\lambda(t)) \tag{3.1}$$

$$1 - 0.007 = exp(-\lambda(t)) \tag{3.2}$$

$$1 - 0.007 = exp(-exp(y \cdot h + C)) \tag{3.3}$$

$$ln(-ln(1 - 0.007)) - C = y \cdot h \tag{3.4}$$

Using the EQUATION 3.4 as a reference to find the ISI through the GLM, a new relation can be found which is available in FIGURE A10. This improvement in significance for both DA and STG neurons. In reality, the correlation could still be increased by modifying the assumption to match as closely as possible with the real ISI.

To conclude about the investigations of the fitted GLM for spiking sequences, it exists a strong correlation between the characteristics of the filter and the interspike interval of the original sequence. The duration of the first negative feedback serves as an approximation of the ISI. This approximation can be enhanced using both the offset component and the EQUATION 3.4. The last approach is ones delivering qualitative results. However, if the approach is more quantitative, only using the time duration of the negative part of the filter is sufficient, *e.g.* to be able to distinguish which sequences have a lower or higher ISI.

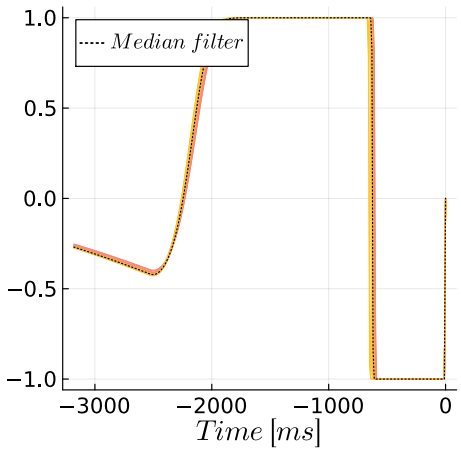
After the discussion about the spiking behaviour, let's move on to the strong bursting before addressing the light bursting. This order is useful to be able to analyse the intermediate state subsequently.

3.3.2 Strong bursting

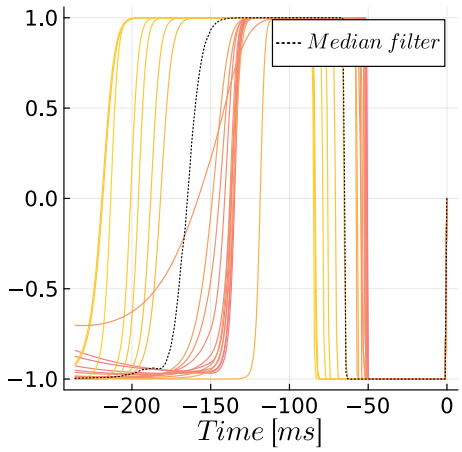
Contrary to the spiking behaviour, where only one component can be extracted (the interspike interval), for bursting behaviours, three components that can be extracted. There are the ISI of a train of spikes, the ISI between trains, and the number of spikes per train, denoted by T_s, T_b, and n_s, respectively. The h-filters of strong bursting, shown in FIGURE 3.5, are still passed through a hyperbolic tangent. The following inspection has been performed on strong bursting behaviours to extract the most stable patterns possible.

Using the same method previously defined for the investigation of spiking, the positions of the positive feedback determine the T_s and T_b. For this purpose, only the method of extracting the ISI thanks to the positive feedback part has been used without considering the offset value and the EQUATION 3.4. This approach is justified for two reasons. First, the offset component remains very similar across simulations unlike the spiking behaviour. Therefore, injected a value of offset to a filter h where the changes of sign in feedback are not abrupt could create incorrect conclusions. In other words, shifting the h-filter by the offset value directly means that this value is used n_w times, where n_w is the number of spikes considered in the time window of h. Second, using the equation is not relevant for the extraction of T_s because the increase in the filter to obtain this first positive feedback is so strong that the value of T_s would be approximately the same. In the other hand, to investigate about the value of T_b, it could be relevant to use it. However, it is not easy to implement it in a general way because of the number of information to consider, such as the number of spike and their ISI.

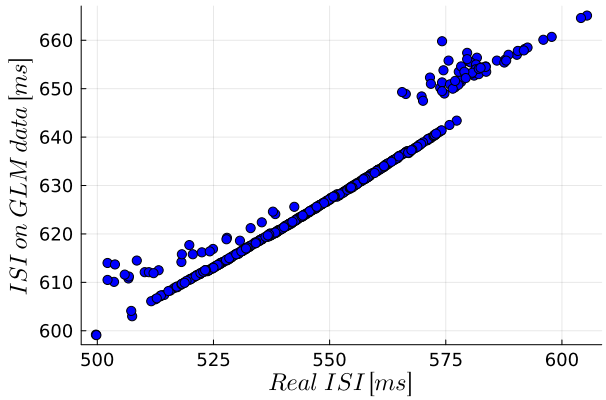
However, one exception is made for the T_s of the STG neuron. Adding the constant component is useful in this case due to the particularity of the filter. Looking precisely at



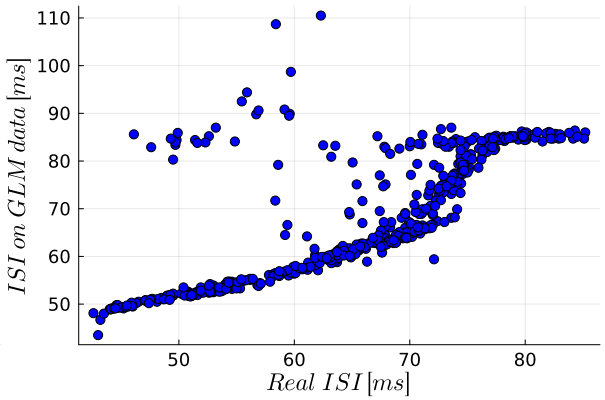
(a) DA - h-filter



(b) STG - h-filter



(c) DA - Comparison of ISI

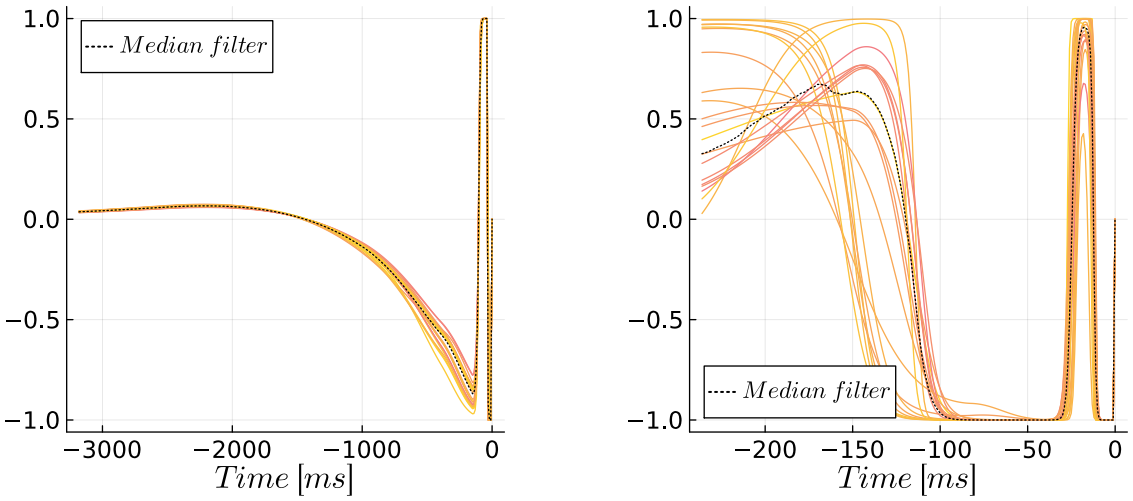


(d) STG - Comparison of ISI

Figure 3.4: Feedback filters and the relations between the ISI for DA and STG neurons. The ISI come from the sequences themselves and the features of the associated h-filter.

the bursting filter for STG neuron (FIGURE 3.5b), the first positive feedback shows different amplitudes across the simulations. Then, adding the offset component to the filter can significantly have an impact in the crossing of this positive part and the x-axis.

For the research of T_s and T_b per neuron type using the fitted GLM, the FIGURE 3.6 refers to it. The relation between T_s and the value extracted from h is highly correlated. Indeed this relation seems linear in the particular range of tested value, $[26; 38]ms$ for the DA neuron. Nevertheless, for the STG neuron, the correlation looks like strong from $6ms$ to $13ms$. But beyond this value, the found values are less well correlated. This degradation of result could be caused to several factors. First, the GLM has problem to model the associated sequence because it is a degenerated ones, meaning that the pattern evolves across the simulation (the first sort already performed is not enough). A second explanation could be that T_s varies



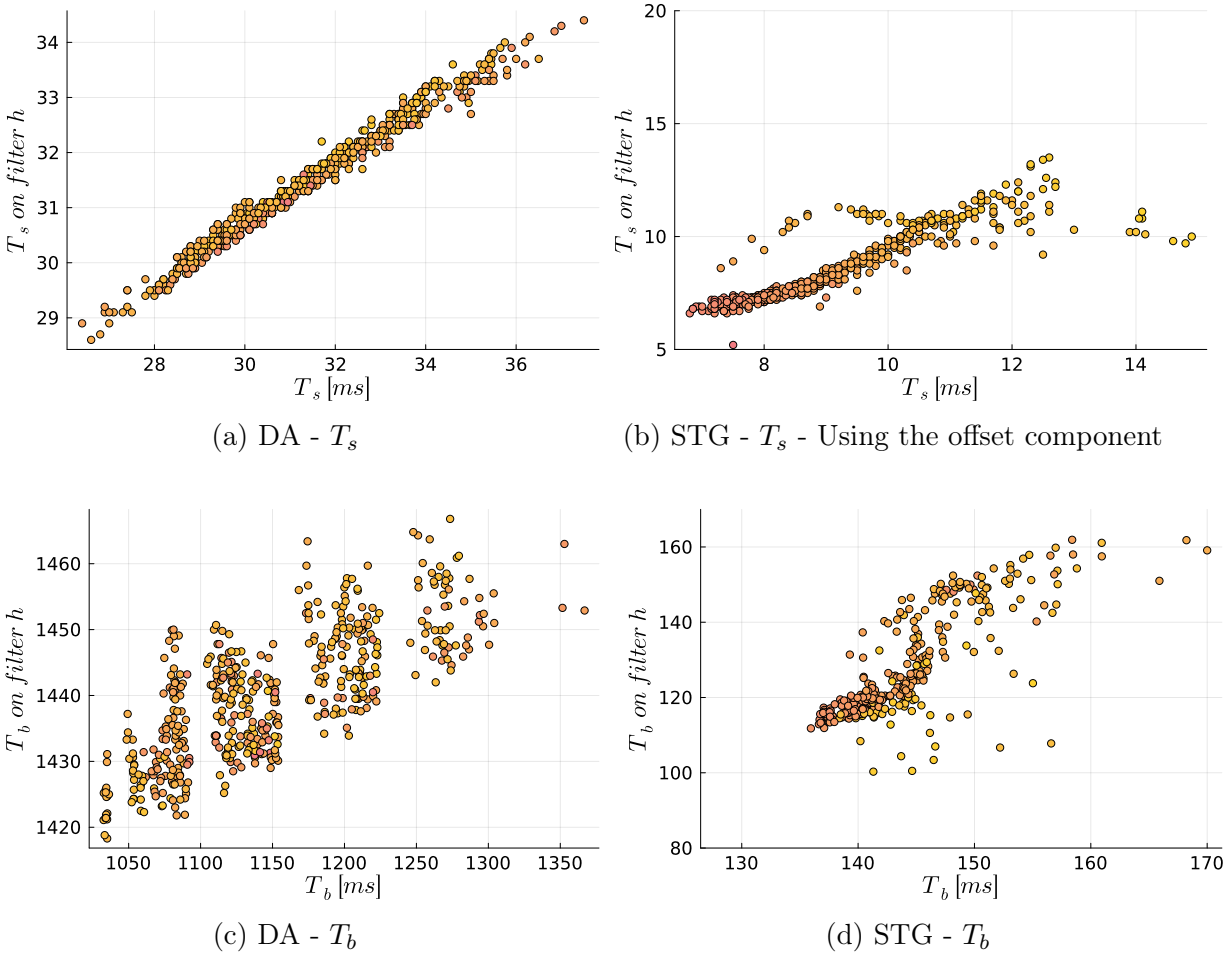
(a) DA (b) STG

Figure 3.5: Feedback filters h for strong bursting behaviour for two types of neuron.

across same trains of spikes which can be quite difficult to a GLM to catch it. The last reason is about the design of the GLM itself which is not able to capture the information of the sequence. This could be due to insufficient training of the GLM which is not enough to obtain good result, or to the number of bases. Indeed, in the fist investigation on GLM, the number bases for h has been fixed to 7, that is a trade-off between accuracy of prediction and complexity of the problem. Since then a higher number of bases will increase the accuracy of the results (see FIGURE 2.7), thereby it could improve the correlation between measured ISI from the GLM, and the true value coming from the original sequence.

Regarding the FIGURES 3.6c and 3.6d, which track the T_b evolution, the relation is less clear. As a reminder, T_b in the time delay between the initiation of two trains of spikes. In a general point of view, there is an increase in the extracted value of the GLM when the T_b of the sequence becomes larger. However it exists a significant variability in the set of extracted value, making it unreliable to draw definitive conclusions from these results. Once again, the variability in the results could be attributed to the GLM design and its training. Specifically, the set of bases function might be insufficient to obtain relevant solutions, or the iteration number is too low, preventing the GLM from converging properly. The first assumption seems more plausible in light of the FIGURE 2.7. Consequently, deeper investigations are necessary to obtain results having higher level of confidence.

At this stage, two investigations have been conducted; one on T_s, and the other on T_b. However, when neuron responds by generating a bursting-like behaviour, a third feature can be studied. It is the number of spikes per train, noted as n_s. To facilitate this study, the investigation will manage by comparing light and strong bursting. Looking at FIGURE 3.5a in



(a) DA - T_s

(b) STG - T_s - Using the offset component

(c) DA - T_b

(d) STG - T_b

Figure 3.6: Relations between the sequences and the h-filters of the GLM for the DA and STG neurons. Only information derived from h are used, excepted for the one mentioned. T_s refers to the elapsed time between spikes of a same train. T_b refers to the elapsed time between trains of spikes.

terms of colour, it can be observed the curves follow a colour gradient in the second negative feedback area. As a reminder, sequences with a low level of burstiness generate yellow curves, while those with a high burstiness level generate more red curves. Therefore the following question can be asked *"Does an increase in burstiness level correspond to a decrease in the amplitude of the second feedback area ?"*.

3.3.3 Light and strong bursting

The difference between these two behaviours lies in the number of spikes that is contained per train. To evaluate this number, in the GLM interpretation, one question can be ask *"How many spikes are needed in the second negative part of the h-filter to balance out the effect of the last spike being in the first positive feedback ?"*. The approach that is built here

is a comparison between the areas generated by the different parts of the filter. Practically, it consists of performing the ratio between these integrals. Noted that this analysis is performed on the true filters h, rather that on the ones which are passing in the hyperbolic tangent which are presented only to facilitate the visualisation. By the way, those modified filters are available in the APPENDICES in FIGURE A11.

In a first step, by examining the h-filter according to the type of neuron, it can easily be concluded that the approach described above does not work for the STG neuron. This is due to the absence of the first positive and thin feedback in h which can have three explanations being still mentioned. Firstly, the neurons associated to these kind of filters might badly classified. Secondly, the sequence degenerates and the GLM is unable to catch this modification of behaviour. And lastly, the offset value is necessary and should be considered in some case. However, the last one could cause problems in the majority of case, then for these reasons the investigation into the number of spikes is only performed on the DA neuron.

In a second step, looking the filters of light and strong bursting of the DA neuron reveals two kinds of curves in the light bursting filter. The majority of them are close to the median curve, while the minority are smoother that the others which look like closer then the ones constituting the h-filter for strong bursting. For facilitated the understanding, those two groups are labelled as A and B, respectively. By comparing those curves with the filters of the strong bursting (noted group C), it is obvious that the group B is closer to C than A. It suggests that

$$n_{s,A} < n_{s,B} < n_{s,C}, \tag{3.5}$$

where the second index refers to the group. Then, if this assumption is correct, the method that is built to distinguish the sequences according the number of spikes per burst would be create three clusters. The FIGURE 3.7 shows it. To be complete, the y-axis has be found with the expression

$$\beta = log(-\frac{\text{Integral of the considered positive feedback}}{\text{Integral of the considered negative feedback}}), \tag{3.6}$$

which allows the apparition of three different clusters. As expected, some points of the light bursting classification are out of the location of the majority. Those points could be refer to a higher number of spikes per burst, which also correspond to the number of filters belonging of the group B, approximately. This matching ensures the link between those features.

After examination of some sequences, the ones classified as light bursting contain three or four spikes per train. For strong bursting sequences, that number is greater then ten. Using this information, it means that the sequences having three spikes per train generate value close to 0.05 according the EQUATION 3.6. But if this value is closer to 0.1, it corresponds to bursting sequences having four spikes. On the other hand, if the generated value is between 0.4 and 0.7, a train of spikes will contain more than ten spikes. Note that these threshold values have been rounded to facilitate the discussions. It is nevertheless interesting to notice

two elements. First, assuming that the filters related to a particular number of spikes per train generated a particular value within a particular range; it means that the area between 0.1 and 0.4 is reserved for sequence having five to nine spikes. Second, but still using the first element, the higher the value of n_s is, the more difficult it is to distinguish the different ranges. The sequences having three or four spikes per train could be differentiable using this method. Therefore, for sequences having a n_b of ten or eleven, the clusters are not distinguished. This observation holds as a train of ten spikes is more similar to a train of eleven than a train of three compared to a train of four.

In the following, a similar investigation on the number of spikes per train is conducted, but a variant of this method is used to demonstrate that different approaches can yield similar insights.

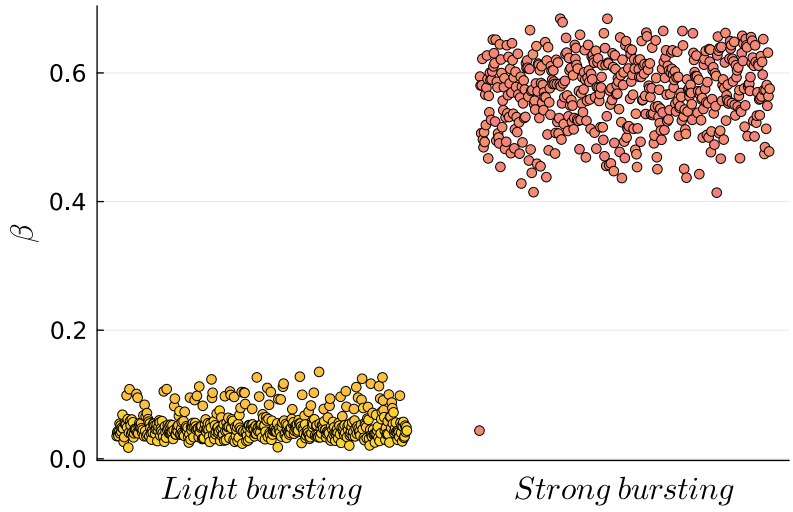


Figure 3.7: Classification of the intensity of the bursting using EQUATION 3.6, first approach.

This investigation concludes the series of the feature extractions that can be derived from the particular sequence thanks to the fitted GLM. However, some of them requires knowledge about the nature of the behaviours. Therefore, the next section will briefly deliver some classification ways to set up to reach this goal.

3.4 Classification of sequences using GLM

The investigation that have already been performed suggest that the information contained inside the fitted GLM can be used to extract some features of the sequences of spikes itself. Then it could be relevant to find a way to automatically classify the sequences based on

their filters h. However, the features depend of the h-filters shape, which depend on the neuronal sequences. Furthermore, a same sequence can be represented by different filters, which complicates the identification of those behaviours. This notion is addressed in the next section. For those reasons, this analysis focuses only on the shape filter obtained. Then, it is important to note that the conclusions drawn from this analysis may not hold true for different filter shapes.

To allow an easier investigation on the behaviour classification, the model of the DA neuron is used. In the previous discussion, it has been shown that the h-filter generated by the STG model is more variable across same behaviour, which is due to the variability of the neuronal response. Then, the GLM derived from the STG will only takes into account to weight the following classification process.

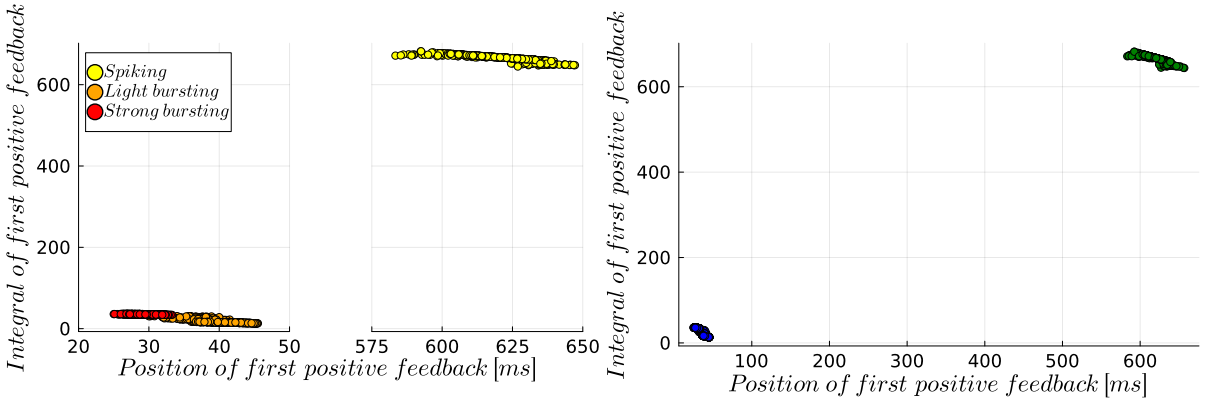
Some classification methods could be used to sort the neuronal response according the behaviour type. Choosing to first distinguish between spiking and bursting sequences using the GLM is a sensible approach for classification. Or directly performed a classification into three groups: spiking, light bursting, and strong bursting. To deliver a analysis as proper as possible, the first option has been selected. *e.g.* first distinguished the spiking and the bursting sequence thanks to the GLM, and therefore study the intensity of the bursting.

To separate spiking from bursting behaviours, various approaches can be used. One is to use the position of the first positive part in the filter, the integral of this filter region, or the extreme values of the filter. These features may be behaviour-dependants. For instance, the position of the filter positive part h delivers information on the interspike interval and T_s for spiking and bursting, respectively. In busting cases, its value will be lower than to spiking behaviour. Additionally, the integral of this region offers also information about the behaviour. In spiking, the integral tends to be higher than in bursting, primarily due to the longer duration of the positive feedback. The third aspect that has not been discussed yet is the intensity of filter. In fact, the filter intensity is also dependant of the behaviour. Intuitively, taking the complexity of the behaviour; the spiking is the simplest and bursting is the more complex one. Therefore, it is more challenging to fit a set of bases to a bursting behaviour, resulting in a lower amplitude compared to spiking for a given set of bases.

Thus, the used features to performed the first sort are extracted from the first positive feedback. The first one is the position of this part, and the second the ratio between its integral and its peak value. The combination of those values allows a better sorting through the entire dataset. For the DA neuron, the obtained results are in FIGURE 3.8, where 3.8a displays the location of data according the expected behaviours thanks to a colour code. The 3.8b shows the two clusters found by the function `kmeans()` implemented in a Julia library. These clusters effectively capture the desired information, one cluster is only composed of the spiking behaviours, and the second by the bursting (both light and strong).

To go deeper in the investigation of the intensity of the bursting behaviour, another

approach can be performed to distinguished this intensity. In the previous section, the EQUATION 3.6 was used to distinguish light to strong bursting, which as deliver the FIGURE 3.7. However, for this classification, another method is used. This choice is driven by the aim to demonstrate that different ways exist to conclude about a single neuron feature. The choice of the method employed may vary depending on how the user defines light and strong bursting; *e.g.* if strong bursting is considered as more that three spike per burst, this new approach is more convenient. The results are available in FIGURE 3.9, where the point close to 4.5 reflect filter h modelling bursting sequences containing three spikes. The points below 3.0 are linked to sequences having around ten spikes, while the other points (close to 3.25) represent filters from sequences containing four spikes. Consequently, the `kmeans()` function extracts clusters containing the green dots and the blues ones, respectively.



(a) Data representation according to the generation of sequences.

(b) Clusters representation using the extracted features.

Figure 3.8: Classification of the behaviours for the DA neurons.

This approach works pretty well for the DA neurons because of the sequences are approximately the same across the same behaviours, resulting in similar filters. However, for the STG neuron which is subject to a wide set of variability as already discussed, the distinction between behaviours are much more challenging. Therefore, the main assumption of the chapter that the filters are able to perfectly reproduce the behaviours of their training sequence is not valid. Indeed the filters are a combination of set of particular functions, then others set of functions could provided filters having more precision in the reproduction of sequences. Thus, neuronal classification could potentially offer a better distinction between behaviours using the GLM fitted using other set of bases functions.

All the classification results are available in the APPENDICES in FIGURES A12 and A13 for the DA and STG neurons, respectively. Each FIGURE is composed of three SUBFIGURES as the following:

- First, sequences classified as spiking according the classification (A12a and A13a)

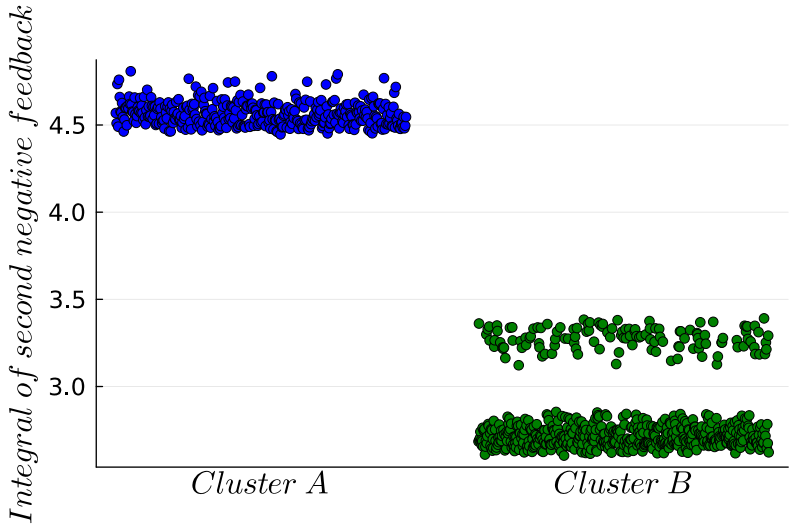


Figure 3.9: Classification of the intensity of the bursting, second approach.

- Second, sequences classified as light bursting according the classification (A12b and A13b)
- Third, sequences classified as strong bursting according the classification (A12c and A13c)

$\longrightarrow$ The sequences are selected randomly in the corresponding set.

Few words about the results, to the DA neuron, the classification is well performed; assuming that more than three spikes per burst can be classified as strong bursting. However, for the STG neuron, the classification is not perfect even if in the majority of cases the classification looks like correct.

Therefore, to use fitted GLM in the aim of the neuronal behaviours classification can be a real relevant method to automatise a procedure.

3.5 Different filters for the same firing pattern

The previous classification was performed under the assumption that only one filter h was able to generate a particular behaviour. However, it does not always hold true. As illustrated in the FIGURE 3.10, two different filters are able to generated bursting patterns. The blue one has the shape of bursting filter that has previously discussed, then none discussion will be done about it. No discussion has been conducted yet to confirm whether the red filter is capable of generating bursting sequences. Let's delve into an analysis of this red filter to verify its behaviour.

First, let's assume that the offset component generates the first spike. Having it, the first spike is placed within the first negative part of the filter (in the region 0 to -100ms), then none spiking event occurs. After 100ms, the spike goes in the first positive part, triggering the generation of a new spike. Now one is in a positive, and one in a negative part, it initialises a feedback battle. The negative part, with its higher amplitude, dominates the positive feedback, then no more spike is generated while the last remains in this first negative part. Then a new spike is generated by the GLM when the last goes out of the first negative part, and the process is generated as long the first spike is in this first positive part. When it goes to the second negative part, zero spike is generated thanks to the feedback battle. However, once all spikes exit this negative region, a new spike will be generated. Then, a new train of spikes starts to regenerate, and the process repeats because the fully generated train can be negligible due to the filter intensity.

This quick interpretation of the action of filter shows that bursting sequences can be generated using different shapes of filter. Therefore it is important to be aware that the classification method could differed according the filter shape, which is dependant of the choice of bases to model it.

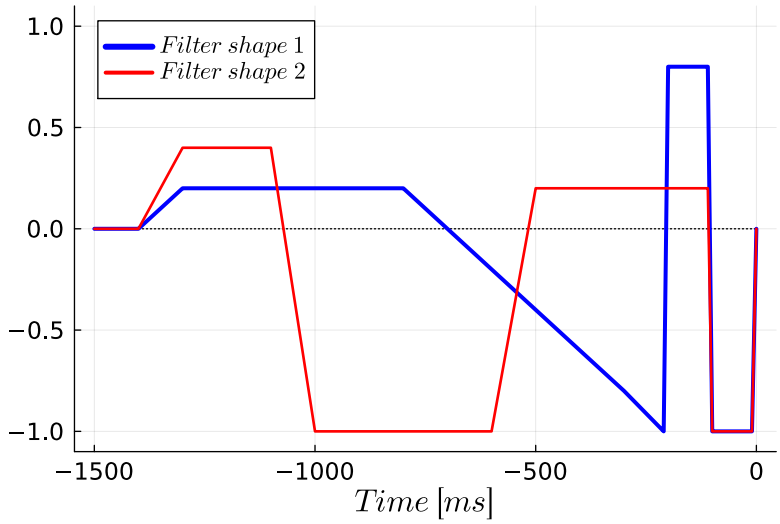


Figure 3.10: Different feedback filters h generating bursting sequences.

3.6 Conclusion

In conclusion, managing investigations on GLM filters is a relevant approach to extract information about neuronal sequences. Many features can be easily extracted, such as the time periods and the number of spikes per train (for bursting sequences). Furthermore, some of then are clearly visible on a display of the filter, which is quite user-friendly. For some extracted features, there is a high correlation with the values extracted from the sequence

itself, allowing a high level of trust. Then, using those features to compare sequences is a smart approach to delve deeper into classification, *e.g.* classification of bursting sequence by examining the time between spikes within a same train. However the conclusions derived from GLM interpretations are subject to its initial design, which is user-dependant. A poor design can provide incorrect results. Furthermore, GLM requires input sequences that are no-degenerated (*e.g.* no sequences starting with light bursting to converge to spiking). Another type of sequence that is not easily captured is sequence where T_s and T_b are similar, because of the bases design.

To re-establish a connection with neuromodulation, the GLM can be trained on neuromodulated sequences to conveniently investigate the induced effects on behaviour. Regarding to the work of A. Fyon, the GLM can be used as a relevant tool to verify if the dimensional reduction has been well performed by comparing the feedback filters h. However, it is important to keep in mind that the accuracy of the results can be influenced by increasing the number of bases functions, though this comes at the cost of increased computation time.

Conclusion

The GLM is a famous tool to investigate about the neuron encoding provided that it is well design. The design depends on various factors which are user-dependant, as its architecture, its nonlinear function, its stochastic process, and the bases functions to design the filters, *e.g.* k and h. Others external parameters from the GLM have also been studied to investigations about theirs impacts on the obtained GLM. For instance the length of sequences is not a factors affecting the final results. However, the shape and the number of bases functions have significant impacts of the final fitted GLM, according to the score measurement. Therefore, well initially choice the components of the GLM is mandatory to obtain high quality results. Some characteristics can be extracted from the designed elements thanks to the training. Those characteristics are useful in a classification way of the sequences; spiking from bursting sequences are distinguishable, as well as light to strong bursting. Furthermore, it exists a high degree of correlation between the extracted value from the filters of the GLM with the characteristic of a given sequence. For instance, the ISI of the spiking sequence are highly correlated with the location of the first positive feedback of the h-filter.

Limitations and improvements

However fitting a GLM is not an easy task and let appear some drawbacks. First, it is a time-consuming task and trade-off with accuracy must be consider. Adding bases in the set of bases function, which is the most impacting component, increasing the quality of results, but it also increase the complexity of the problem to optimise. Then compromise must be done in regards to these elements. Second, it can exist different relevant solution to reproduce a particular firing pattern, since then it complexities the extractions of characteristics. Therefore, the existence of several solutions means that lots of local solutions could also exist. Then performed a relevant initialisation phase is mandatory to allow to the optimisation to find a real solution instead of a local one. Third, the optimisation phase must manage very high value, *e.g.* $exp(700)$, which is quite challenging to traditional CPU, especially for phasic behaviours. The computed results can vary according the used devices as previously explained.

Perspectives

The GLM could be used at various end. The first possibility is in the research field; indeed, be able to classified sequences according to the intrinsic behaviour of the sequences could help to automatise the process. Studying the modifications of the behaviours on a single sequence could also be a relevant approach. The GLM does not require lots of data to be fitted, therefore, the modification of the sequence behaviours could also be investigated.

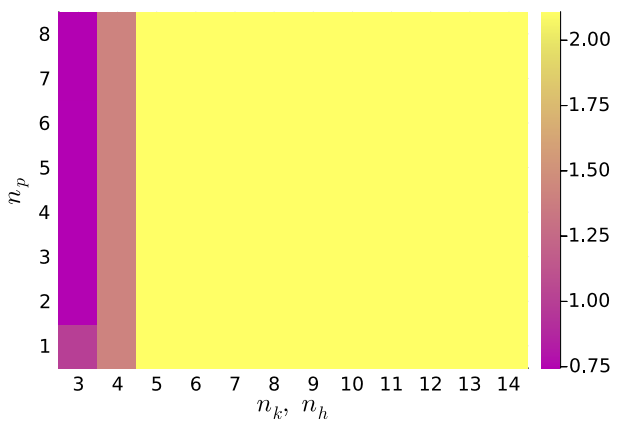
A second uses case would be to consider the GLM in its entirety by taking into account the sequences that it generated. Those sequences can be used as data base to others applications requiring many data, *e.g.* deep learning applications. The generated sequences are generated using a stochastic process which introduce variability between generations, It could be a smart way to increase the among of sequence easily.

A third possibility is to investigate deeply the relationship between different neuron. To study this kind of influence, others GLM architecture is mandatory as the most complex one describe in the Section 1 Background for instance. Investigating the perception of external stimulus could also be a relevant tasks. As already mentioned in the last chapter, the effect of neuromodulators can also be studied by analysing the filters generated by neuromodulated sequences. Therefore, investigations on others diseases as neurodegenerative diseases could also be studied and better yet predicted.

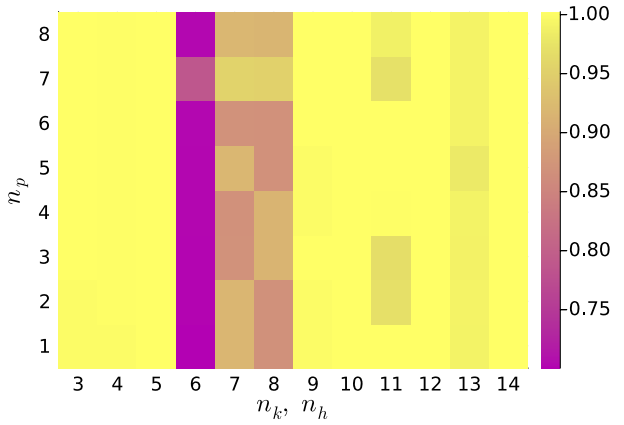
Another possibility, in a more advanced time would be to used GLM as a decoder to infer action to mechanical prosthesis. Assuming that someone have an artificial hand to due an accident, being able to interpreted the signal coming from the nerve to transcript it as mechanical action to the prosthesis could a famous application increasing the life quality of concerned people. The mains obstacles to this quite of solutions remains the computation time of the GLM which is not negligible in those kind of applications.

Appendices

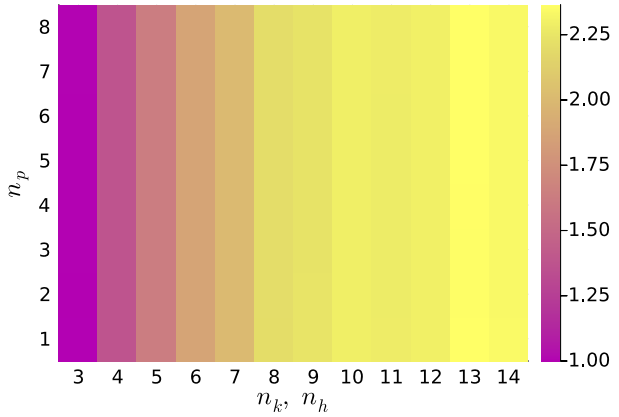
Figures



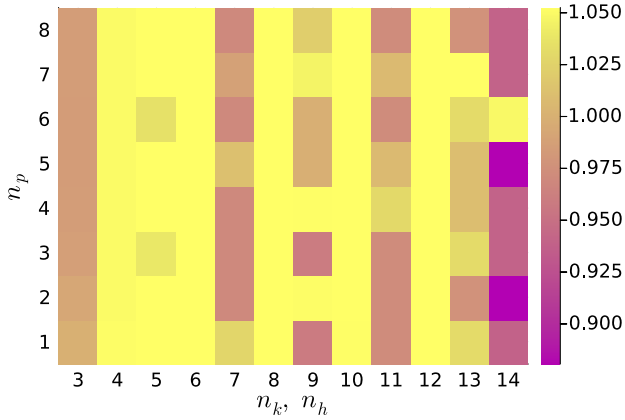
(a) Tonic spiking



(b) Phasic spiking

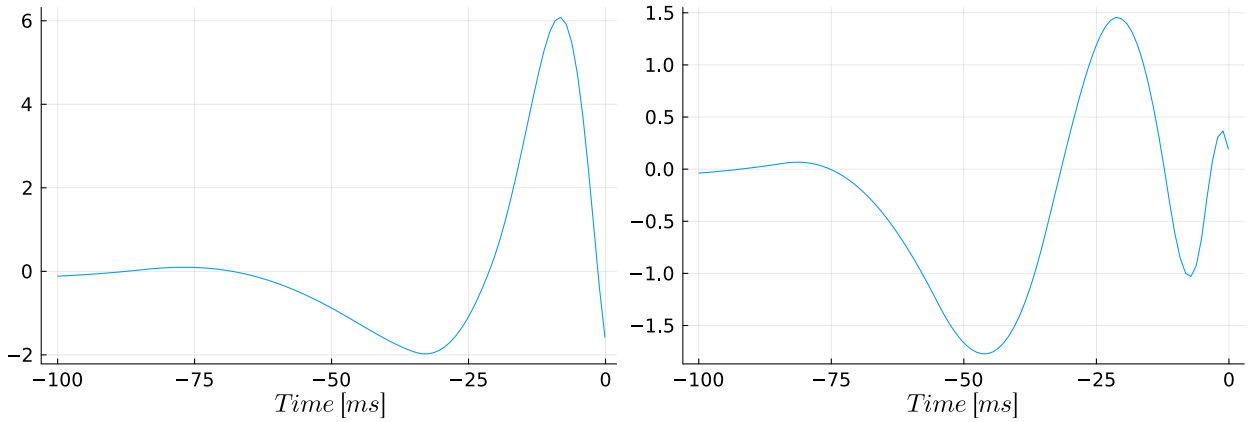


(c) Tonic bursting



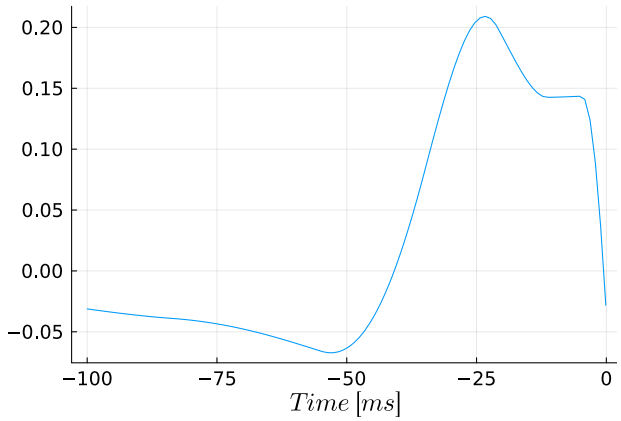
(d) Phasic bursting

Figure A1: LL scores according to the number of bases functions and the number of periods for the four studied behaviours.

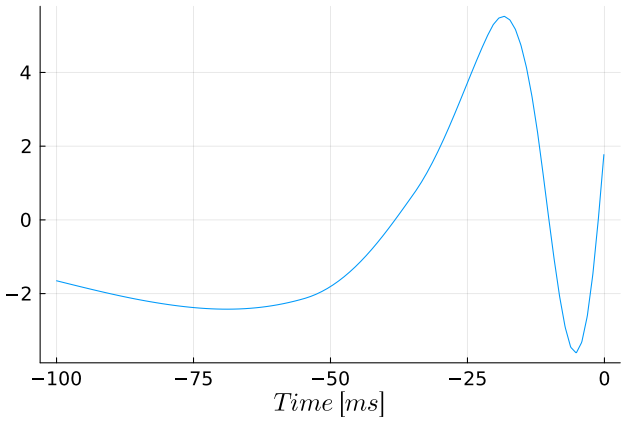


(a) Tonic spiking

(b) Phasic spiking

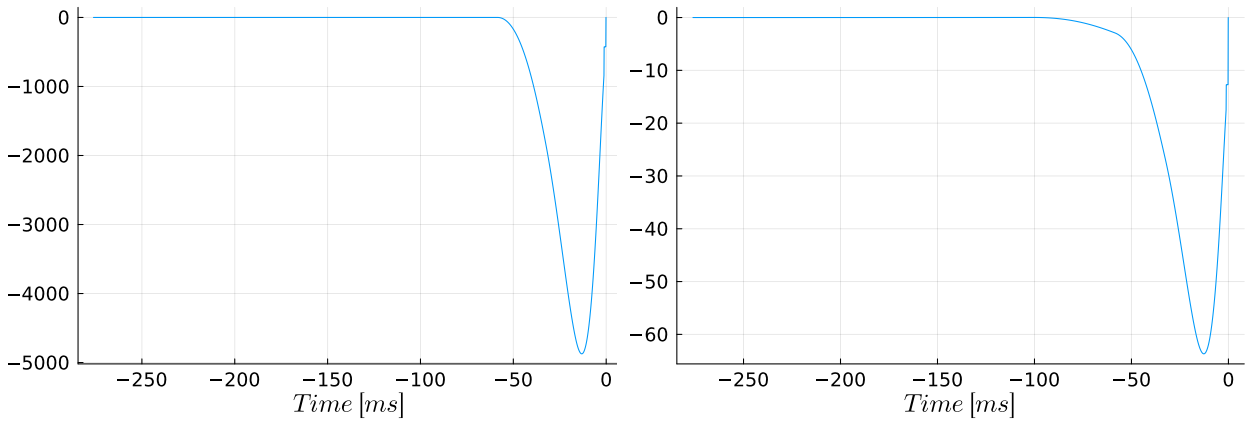


(c) Tonic bursting



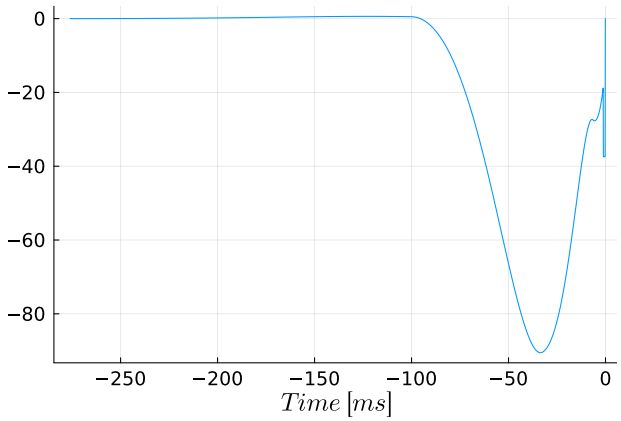
(d) Phasic bursting

Figure A2: Feedforward filter k of GLM according to different behaviours.

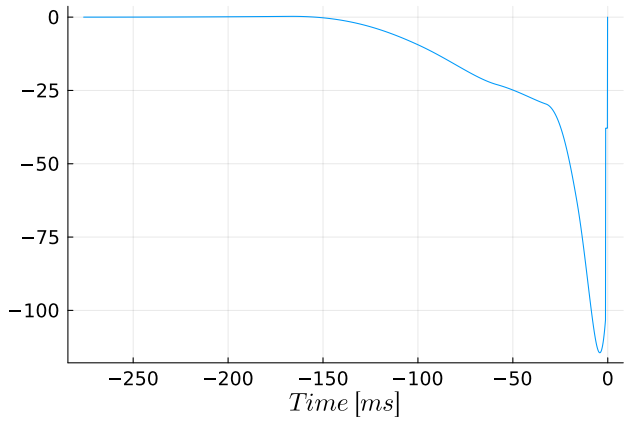


(a) Tonic spiking

(b) Phasic spiking

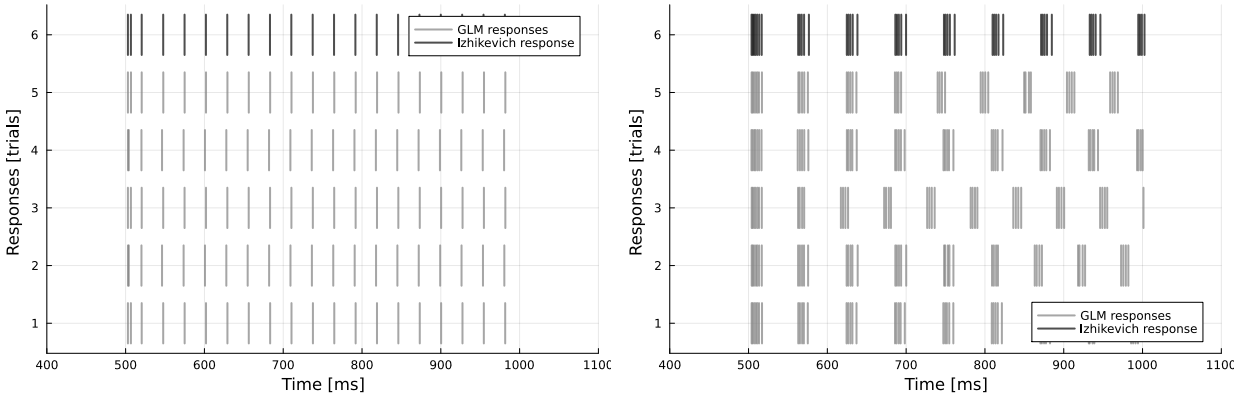


(c) Tonic bursting



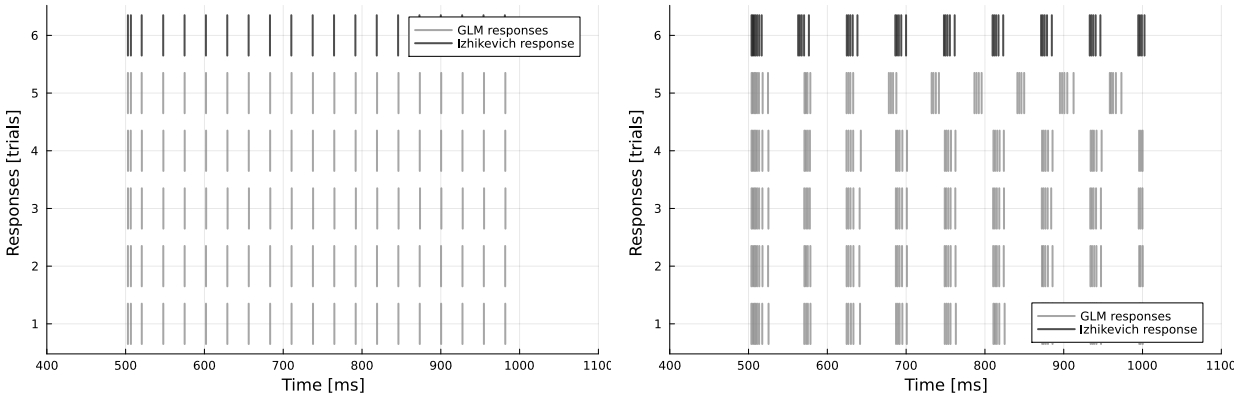
(d) Phasic bursting

Figure A3: Feedback filter h of GLM according to different behaviours.



(a) Tonic spiking - Raised-cosines bases

(b) Tonic bursting - Raised-cosines bases



(c) Tonic spiking - Home-made bases

(d) Tonic bursting - Home-made bases

Figure A4: Generated sequences by the GLM for the tonic behaviours according to different designs of bases functions. The number of bases in the set designing both feedforward and feedback filters are equal to 7.

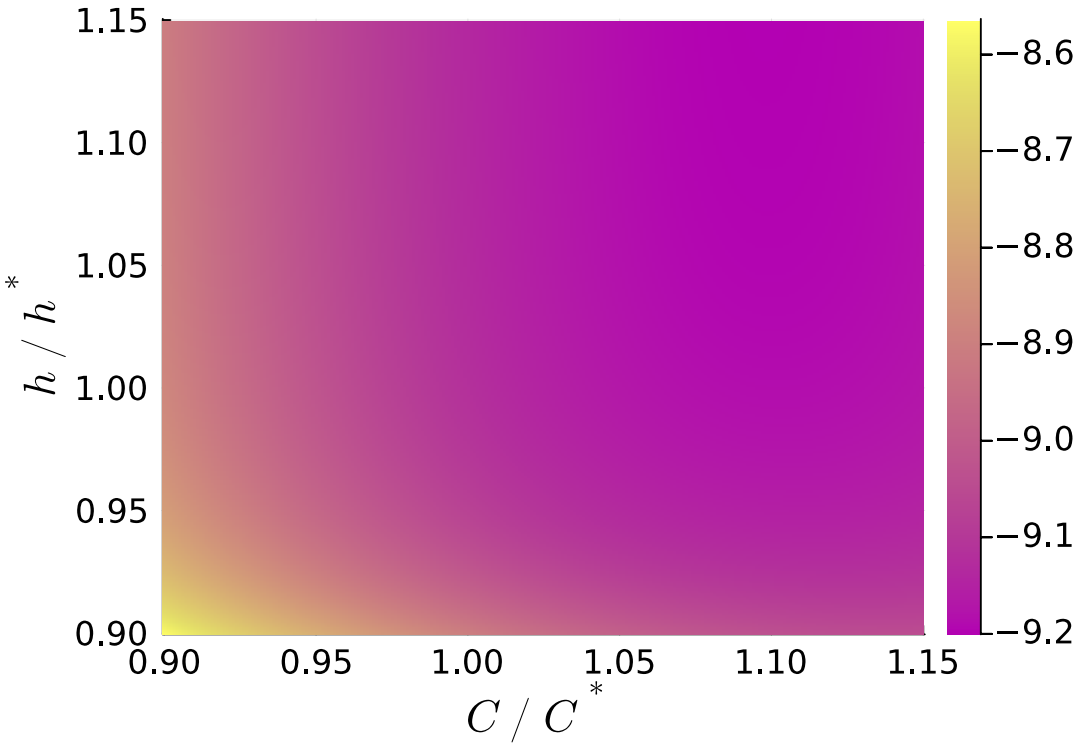


Figure A5: Evolution of the negative log-likelihood for the **phasic spiking behaviour** according to the linear stretching of h-filter and the offset value C. The nLL being too large the be compute are uncoloured. The dots are the local solutions associated with their own score, where the scores higher than 1 have been resize as $log(10, score)$ to facilitate the visual interpretation.

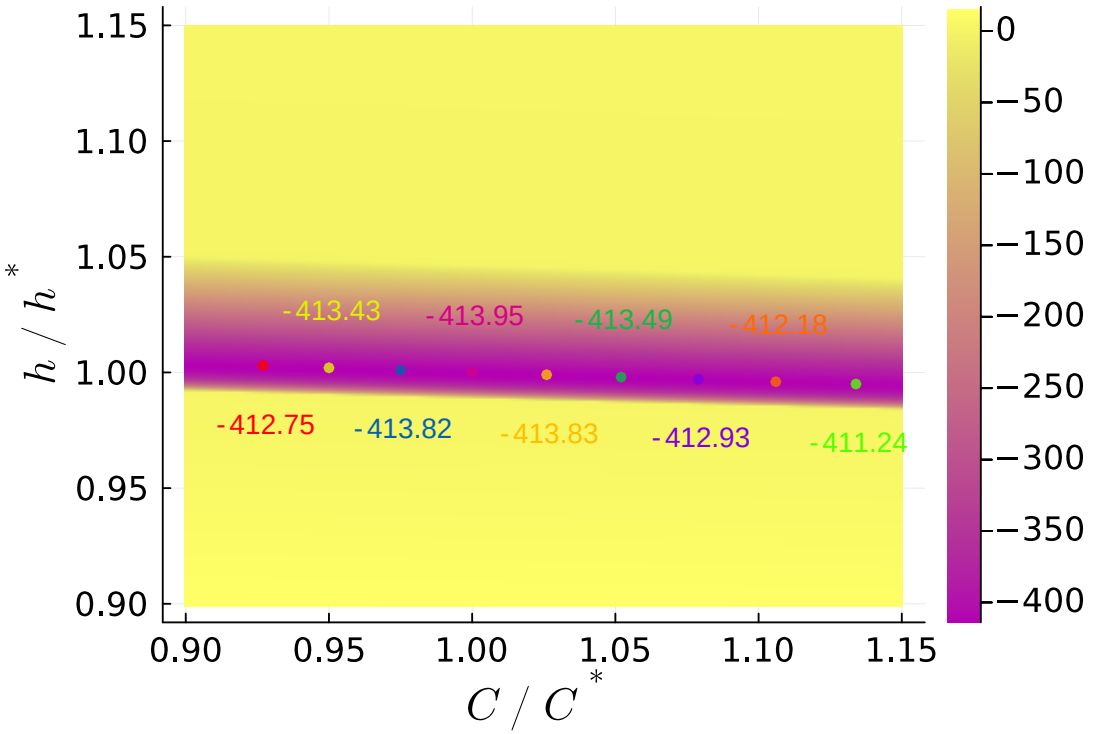


Figure A6: Evolution of the negative log-likelihood for the **tonic bursting behaviour** according to the linear stretching of h-filter and the offset value C. The nLL being too large the be compute are uncoloured. The dots are the local solutions associated with their own score, where the scores higher than 1 have been resize as $log(10, score)$ to facilitate the visual interpretation.

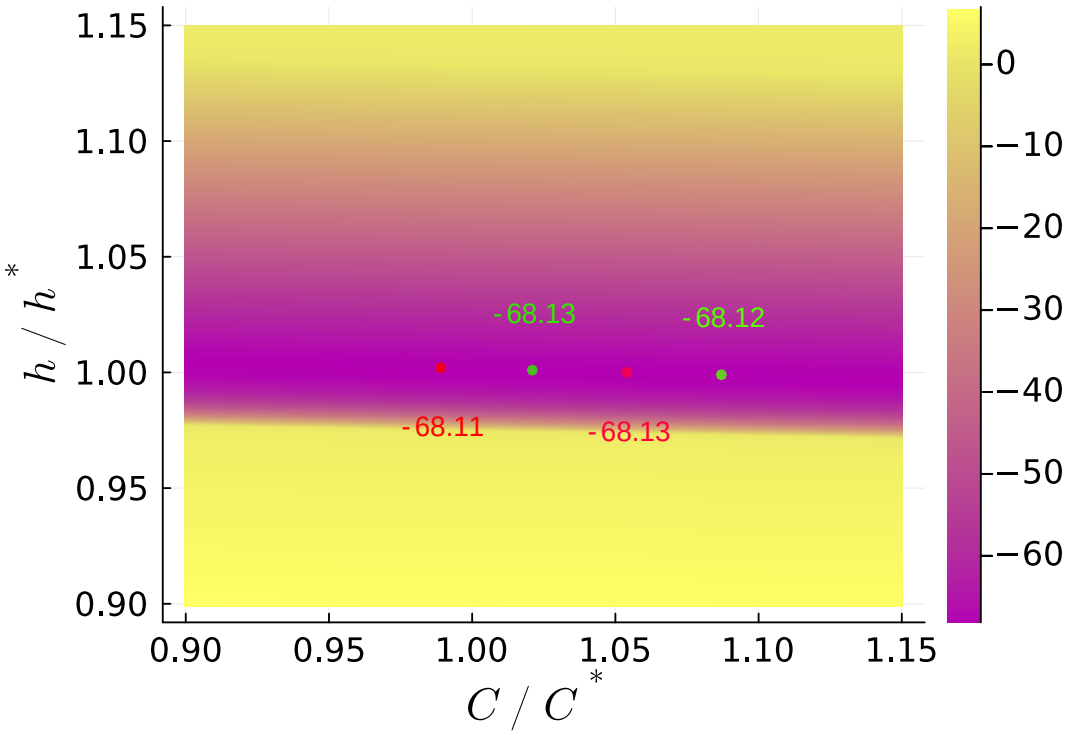


Figure A7: Evolution of the negative log-likelihood for the **phasic bursting behaviour** according to the linear stretching of h-filter and the offset value C. The nLL being too large the be compute are uncoloured. The dots are the local solutions associated with their own score, where the scores higher than 1 have been resize as $log(10, score)$ to facilitate the visual interpretation.

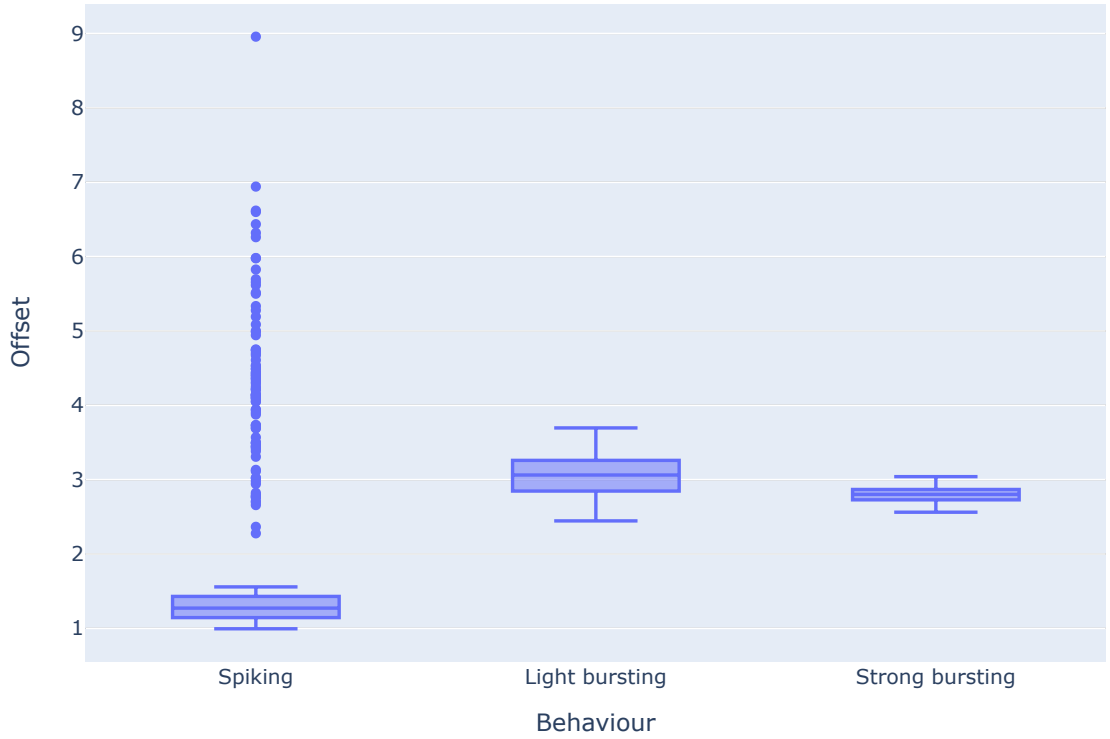


Figure A8: Value of the offset component for the DA neuron according to the behaviour

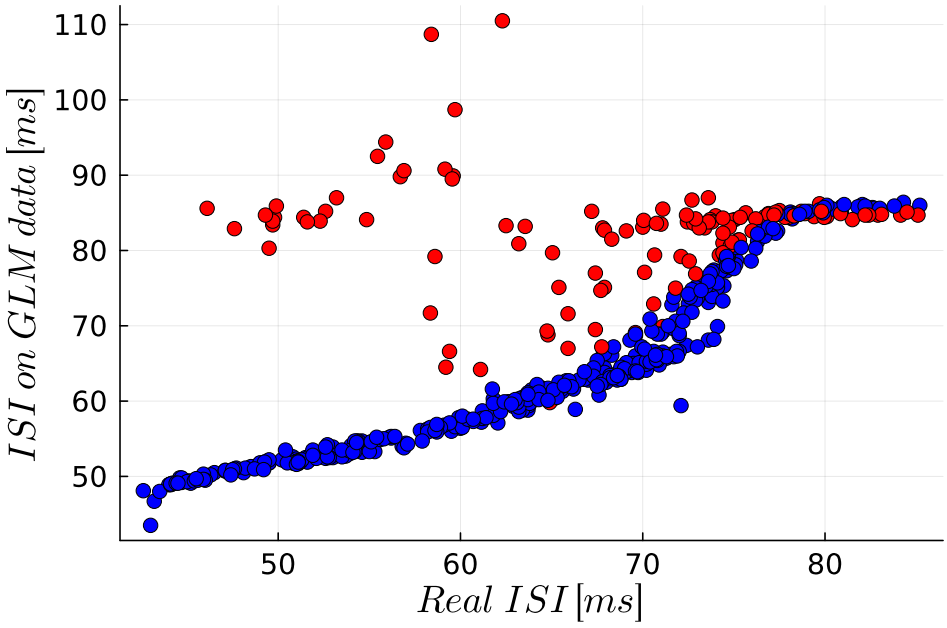
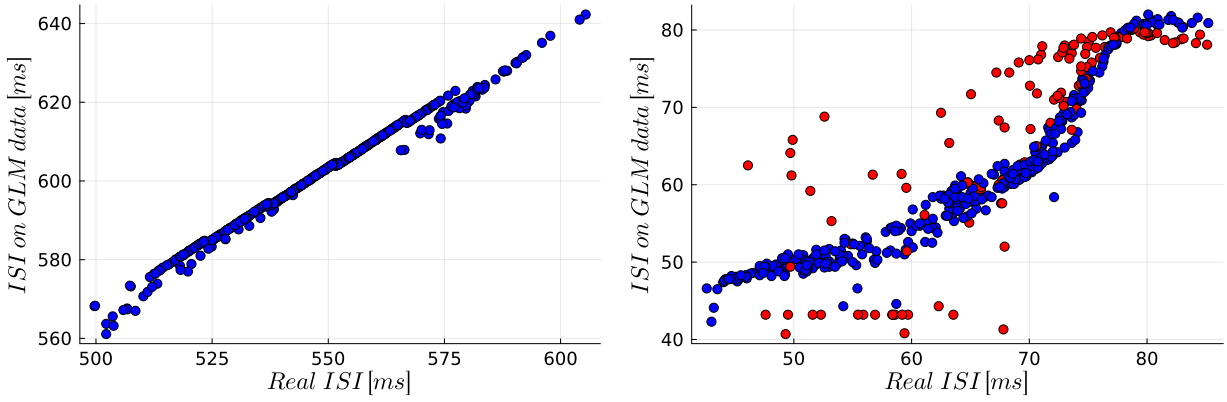


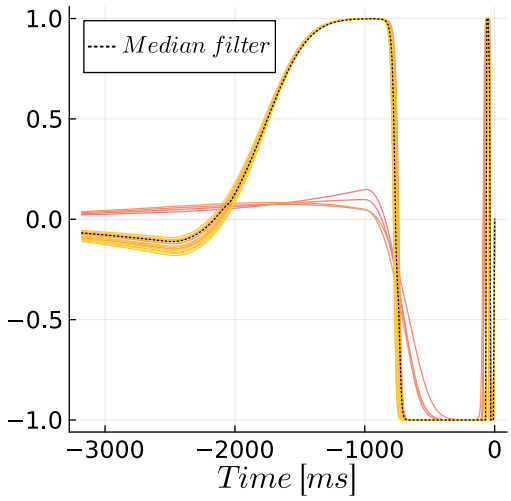
Figure A9: Comparison between the real interspike interval and the interspike interval coming from the GLM. The red dots corresponds to sequences having a standard deviation of the interspike interval higher then $25ms$.



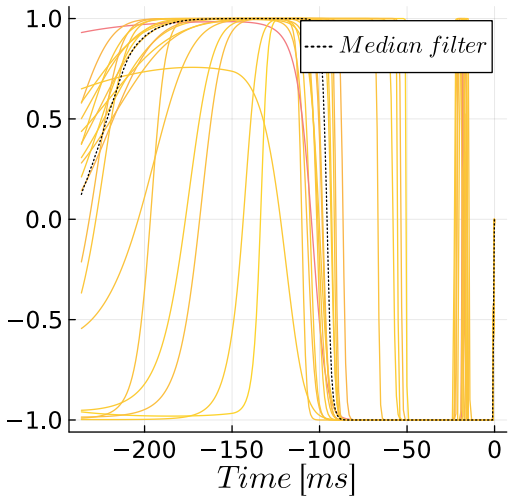
(a) DA

(b) STG

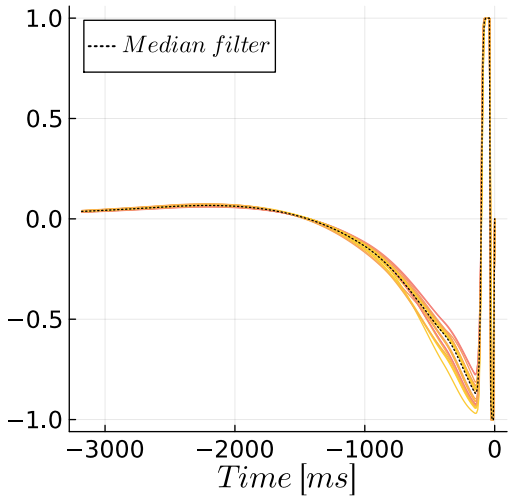
Figure A10: Comparison between the real interspike interval and the interspike interval coming from the GLM using the EQUATION 3.4 for the two studied neuron types. The red dots corresponds to sequences having a standard deviation of the interspike interval higher then $25ms$.



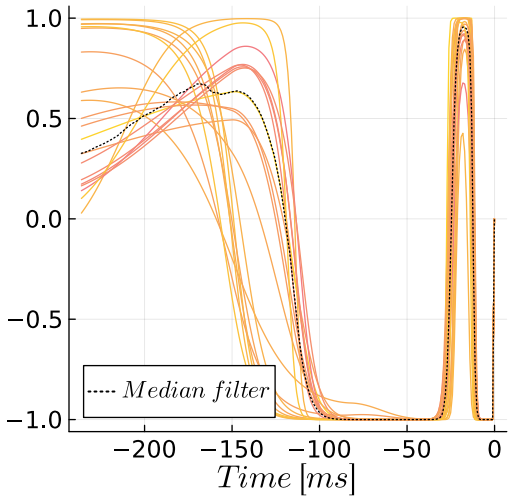
(a) DA - Light bursting



(b) STG - Light bursting

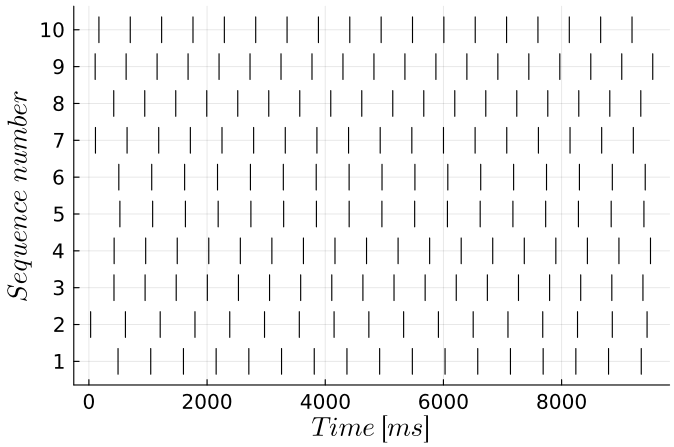


(c) DA - Strong bursting

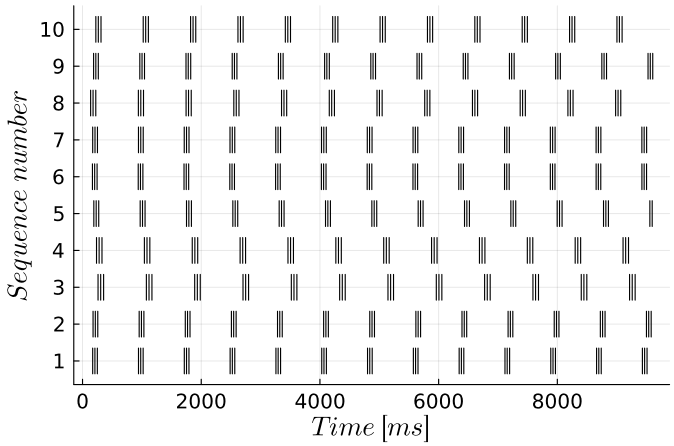


(d) STG - Strong bursting

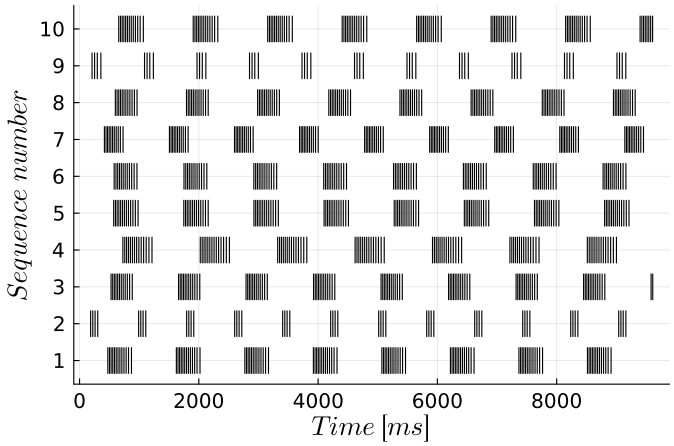
Figure A11: Feedback filters h for bursting generated sequences for the two studied neuron types.



(a) Spiking classification

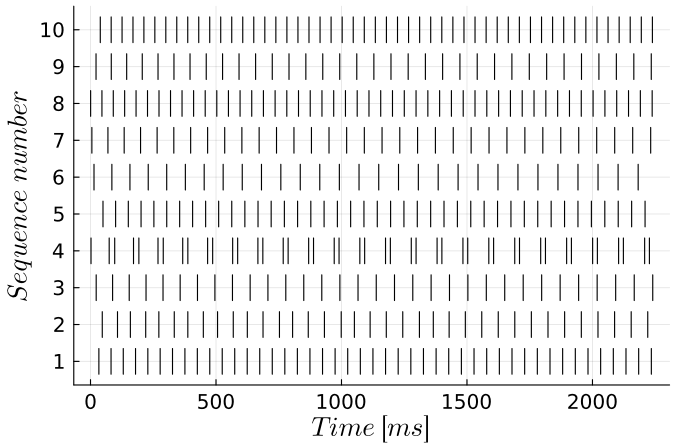


(b) Light bursting classification

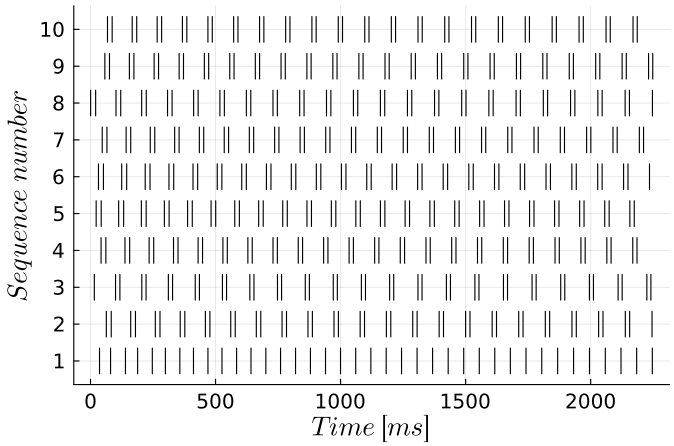


(c) Strong bursting classification

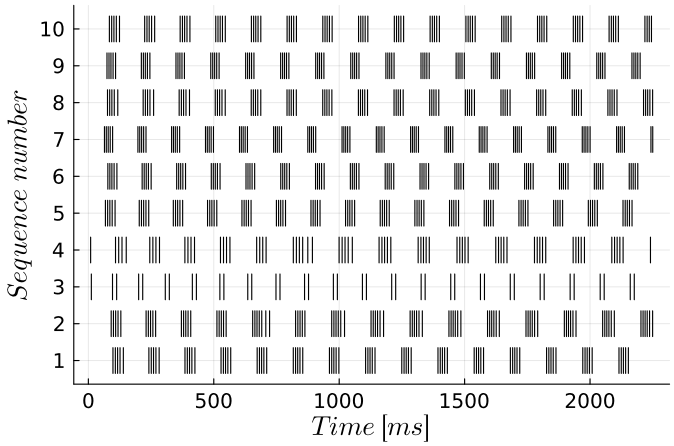
Figure A12: Some results of the patterns classification using the features of the fitted GLM for the DA neuron.



(a) Spiking classification



(b) Light bursting classification



(c) Strong bursting classification

Figure A13: Some results of the patterns classification using the features of the fitted GLM for the STG neuron.

M

Tables

	a	b	c	d	dt
Tonic spiking	0.02	0.2	-65	6	0.1
Phasic spiking	0.02	0.25	-65	6	0.1
Tonic bursting	0.02	0.2	-50	2	0.1
Phasic bursting	0.02	0.25	-55	0.05	0.1
Spike frequency adaptation	0.01	-0.1	-55	5	0.1
Class 1	0.02	-0.1	-55	6	0.1
Class 2	0.2	0.26	-65	0	0.1
Spike latency	0.02	0.2	-65	3.49	0.1
Resonator	0.1	0.26	-60	-1	0.5
Integrator	0.02	-0.1	-55	6	0.5
Rebound spike	0.03	0.25	-60	4	0.1
Rebound burst	0.03	0.25	-52	0	0.1
Bistability 1	1	1.5	-60	0	0.1
Accommodation	0.02	1	-55	4	0.1

Table A1: Parameters value of the Izhikevich model for some neuronal behaviours. [18]

Equations

$$
\text{STG dynamics} \begin{cases}
\dfrac{dV}{dt} = \dfrac{1}{C}\big(- g_{\mathrm{Na}} \cdot m_{\mathrm{Na}}^3 \cdot h_{\mathrm{Na}} \cdot (V - V_{\mathrm{Na}}) - g_{\mathrm{CaT}} \cdot m_{\mathrm{CaT}}^3 \cdot h_{\mathrm{CaT}} \cdot (V - V_{\mathrm{Ca}}) \\
\qquad - g_{\mathrm{CaS}} \cdot m_{\mathrm{CaS}}^3 \cdot h_{\mathrm{CaS}} \cdot (V - V_{\mathrm{Ca}}) - g_{\mathrm{A}} \cdot m_{\mathrm{A}}^3 \cdot h_{\mathrm{A}} \cdot (V - V_{\mathrm{K}}) \\
\qquad - g_{\mathrm{KCa}} \cdot m_{\mathrm{KCa}}^4 \cdot (V - V_{\mathrm{K}}) - g_{\mathrm{Kd}} \cdot m_{\mathrm{Kd}}^4 \cdot (V - V_{\mathrm{K}}) - g_{\mathrm{H}} \cdot m_{\mathrm{H}} \cdot (V - V_{\mathrm{H}}) \\
\qquad - g_{\mathrm{leak}} \cdot (V - V_{\mathrm{leak}}) + I_{\mathrm{app}}\big) \\
\dfrac{dm_{\mathrm{Na}}}{dt} = \dfrac{1}{\tau_{m\mathrm{Na}}(V)} \cdot (m_{\mathrm{Na,inf}}(V) - m_{\mathrm{Na}}) \\
\dfrac{dh_{\mathrm{Na}}}{dt} = \dfrac{1}{\tau_{h\mathrm{Na}}(V)} \cdot (h_{\mathrm{Na,inf}}(V) - h_{\mathrm{Na}}) \\
\dfrac{dm_{\mathrm{CaT}}}{dt} = \dfrac{1}{\tau_{m\mathrm{CaT}}(V)} \cdot (m_{\mathrm{CaT,inf}}(V) - m_{\mathrm{CaT}}) \\
\dfrac{dh_{\mathrm{CaT}}}{dt} = \dfrac{1}{\tau_{h\mathrm{CaT}}(V)} \cdot (h_{\mathrm{CaT,inf}}(V) - h_{\mathrm{CaT}}) \\
\dfrac{dm_{\mathrm{CaS}}}{dt} = \dfrac{1}{\tau_{m\mathrm{CaS}}(V)} \cdot (m_{\mathrm{CaS,inf}}(V) - m_{\mathrm{CaS}}) \\
\dfrac{dh_{\mathrm{CaS}}}{dt} = \dfrac{1}{\tau_{h\mathrm{CaS}}(V)} \cdot (h_{\mathrm{CaS,inf}}(V) - h_{\mathrm{CaS}}) \\
\dfrac{dm_{\mathrm{A}}}{dt} = \dfrac{1}{\tau_{m\mathrm{A}}(V)} \cdot (m_{\mathrm{A,inf}}(V) - m_{\mathrm{A}}) \\
\dfrac{dh_{\mathrm{A}}}{dt} = \dfrac{1}{\tau_{h\mathrm{A}}(V)} \cdot (h_{\mathrm{A,inf}}(V) - h_{\mathrm{A}}) \\
\dfrac{dm_{\mathrm{KCa}}}{dt} = \dfrac{1}{\tau_{m\mathrm{KCa}}(V)} \cdot (m_{\mathrm{KCa,inf}}(V, \mathrm{Ca}) - m_{\mathrm{KCa}}) \\
\dfrac{dm_{\mathrm{Kd}}}{dt} = \dfrac{1}{\tau_{m\mathrm{Kd}}(V)} \cdot (m_{\mathrm{Kd,inf}}(V) - m_{\mathrm{Kd}}) \\
\dfrac{dm_{\mathrm{H}}}{dt} = \dfrac{1}{\tau_{m\mathrm{H}}(V)} \cdot (m_{\mathrm{H,inf}}(V) - m_{\mathrm{H}}) \\
\dfrac{dCa}{dt} = \dfrac{1}{20}\big(- 0.94 \cdot (g_{\mathrm{CaT}} \cdot m_{\mathrm{CaT}}^3 \cdot h_{\mathrm{CaT}} \cdot (V - V_{\mathrm{Ca}}) \\
\qquad + g_{\mathrm{CaS}} \cdot m_{\mathrm{CaS}}^3 \cdot h_{\mathrm{CaS}} \cdot (V - V_{\mathrm{Ca}})) - Ca + 0.05\big)
\end{cases}
\tag{3.7}
$$

$$
\text{DA dynamics}\begin{cases}
\dfrac{dV}{dt} = \dfrac{1}{C}\big(- g_{\mathrm{Na}} \cdot m^3 \cdot h(V - V_{\mathrm{Na}}) - g_{\mathrm{Kd}} \cdot n^3 \cdot (V - V_{\mathrm{K}}) \\
\qquad - g_{\mathrm{CaL}} \cdot m_{\mathrm{CaL}}^2 \cdot (V - V_{Ca}) - g_{\mathrm{CaN}} \cdot m_{\mathrm{CaN}} \cdot (V - V_{Ca}) \\
\qquad - g_{\mathrm{ERG}} \cdot o_{\mathrm{ERG}} \cdot (V - V_K) - g_{\mathrm{leak}} \cdot (V - V_{\mathrm{leak}}) \\
\qquad - \dfrac{g_{\mathrm{NMDA}} \cdot (V - V_{\mathrm{NMDA}})}{1 + M_g \exp(-0.08V)/10} + I_{\mathrm{app}}\big) \\
\dfrac{dm}{dt} = \dfrac{1}{\tau_m(V)} \cdot (m_{\mathrm{inf}}(V) - m) \\
\dfrac{dh}{dt} = \dfrac{1}{\tau_h(V)} \cdot (h_{\mathrm{inf}}(V) - h) \\
\dfrac{dn}{dt} = \dfrac{1}{\tau_n(V)} \cdot (n_{\mathrm{inf}}(V) - n) \\
\dfrac{dm_{\mathrm{CaL}}}{dt} = \dfrac{1}{\tau_{m\mathrm{CaL}}(V)} \cdot (m_{\mathrm{CaL,inf}}(V) - m_{\mathrm{CaL}}) \\
\dfrac{dm_{\mathrm{CaN}}}{dt} = \dfrac{1}{\tau_{m\mathrm{CaN}}(V)} \cdot (m_{\mathrm{CaN,inf}}(V) - m_{\mathrm{CaN}}) \\
\dfrac{do_{\mathrm{ERG}}}{dt} = a_{0\mathrm{ERG}}(V) \cdot (1 - o_{\mathrm{ERG}} - i_{\mathrm{ERG}}) + b_{\mathrm{iERG}}(V) \cdot i_{\mathrm{ERG}} \\
\qquad - o_{\mathrm{ERG}} \cdot (a_{\mathrm{iERG}}(V) + b_{0\mathrm{ERG}}(V)) \\
\dfrac{di_{\mathrm{ERG}}}{dt} = a_{\mathrm{iERG}}(V) \cdot o_{\mathrm{ERG}} - b_{\mathrm{iERG}}(V) \cdot i_{\mathrm{ERG}}
\end{cases} \tag{3.8}
$$

Bibliography

[1] Guillaume Drion. "Regulation of Excitability, Pacemaking, and Bursting: Insights from Dopamine Neuron Electrophysiology". In: (2013).

[2] Kristen Harris (principal investigator). *SynapseWeb*. Consulted the 25^{th} November 2023. 1999. URL: `https://synapseweb.clm.utexas.edu/dimensions-dendrites`.

[3] Charles F Stevens. "The neuron". In: *Scientific American* 241.3 (1979), pp. 54–65.

[4] Maria Rosaria Muzio and Marco Cascella. "Histology, axon". In: (2020).

[5] WI McDonald. "The mystery of the origin of multiple sclerosis." In: *Journal of Neurology, Neurosurgery & Psychiatry* 49.2 (1986), pp. 113–123.

[6] John H Byrne, Ruth Heidelberger, and M Neal Waxham. *From molecules to networks: an introduction to cellular and molecular neuroscience*. Academic Press, 2014.

[7] Arthur Fyon. *NmodController*. Consulted the 25^{th} April 2024. 2023. URL: `https://github.com/arthur-fyon/NmodController.jl`.

[8] "Robustness in the strategy of scientific model building". In: *Robustness in statistics*. Elsevier, 1979, pp. 201–236.

[9] AL Hodgkin and AF Huxley. "Current and Its Application to Conduction". In: *J. Physiol* 117 (1952), pp. 500–544.

[10] Alan L Hodgkin, Andrew F Huxley, and Bernard Katz. "Measurement of current-voltage relations in the membrane of the giant axon of Loligo". In: *The Journal of physiology* 116.4 (1952), p. 424.

[11] Eugene M Izhikevich. "Simple model of spiking neurons". In: *IEEE Transactions on neural networks* 14.6 (2003), pp. 1569–1572.

[12] Kathleen Coutisse. "Sensitivity and robustness analysis of thalamic neuron models at the cellular and network levels". PhD thesis. Master's thesis, University of Liège, 2018.

[13] John Ashworth Nelder and Robert WM Wedderburn. "Generalized linear models". In: *Journal of the Royal Statistical Society Series A: Statistics in Society* 135.3 (1972), pp. 370–384.

[14] Peter McCullagh. *Generalized linear models*. Routledge, 2019.

[15] Child Age Average. *Logistic Regression*. 2019.

[16] Chase Coleman et al. "Matlab, Python, Julia: What to Choose in Economics?" In: *Computational Economics* 58 (2021), pp. 1263–1288.

[17] Ioana Dogaru and Radu Dogaru. "Using Python and Julia for efficient implementation of natural computing and complexity related algorithms". In: *2015 20th International Conference on Control Systems and Computer Science*. IEEE. 2015, pp. 599–604.

[18] Alison I Weber and Jonathan W Pillow. "Capturing the dynamical repertoire of single neurons with generalized linear models". In: *Neural computation* 29.12 (2017), pp. 3260–3289.

[19] Robert J Schilling, James J Carroll, and Ahmad F Al-Ajlouni. "Approximation of nonlinear systems with radial basis function neural networks". In: *IEEE Transactions on neural networks* 12.1 (2001), pp. 1–15.

[20] Ahmad F Al-Ajlouni, Robert J Schilling, and SL Harris. "Identification of nonlinear discrete-time systems using raised-cosine radial basis function networks". In: *International Journal of Systems Science* 35.4 (2004), pp. 211–221.

[21] Liam Paninski. "Maximum likelihood estimation of cascade point-process neural encoding models". In: *Network: Computation in Neural Systems* 15.4 (2004), p. 243.

[22] Jonathan W Pillow et al. "Spatio-temporal correlations and visual signalling in a complete neuronal population". In: *Nature* 454.7207 (2008), pp. 995–999.

[23] Renaud Jolivet et al. "Predicting spike timing of neocortical pyramidal neurons by simple threshold models". In: *Journal of computational neuroscience* 21 (2006), pp. 35–49.

[24] International Neuromodulation Society. *Neuromodulation, or Neuromodulatory Effect*. Consulted the 10^{th} April 2024. 2013. URL: `https://www.neuromodulation.com/neuromodulation-defined`.

[25] Guillaume Drion et al. *Principles of Neuroengineering*. Slides. Course given at the ULiège, School of Engineering and Computer Science. 2023-2024.

[26] Guillaume Drion et al. "Dynamic input conductances shape neuronal spiking". In: *eneuro* 2.1 (2015).

[27] Arthur Fyon and Guillaume Drion. "An adaptive controller of reliable neuromodulation on mixed feedback systems". Anglais. In: F.R.S.-FNRS - Fonds de la Recherche Scientifique [BE]. Blankenberge, Belgium, 2024. URL: `https://www.beneluxmeeting.nl/2024/`.

[28] Zheng Liu et al. "A model neuron with activity-dependent conductances regulated by multiple calcium sensors". In: *Journal of Neuroscience* 18.7 (1998), pp. 2309–2320.

[29] Guillaume Drion et al. "How modeling can reconcile apparently discrepant experimental results: the case of pacemaking in dopaminergic neurons". In: *PLoS Computational Biology* 7.5 (2011), e1002050.