Neural field models with a dendritic dimension

Thesis submitted for the degree of Doctor of Philosophy

Viviana Culmone

School of Psychology and Clinical Language Sciences

August 2019

Supervisors: Prof. Ingo Bojak and Dr. Etienne Roesch

Declaration

I, Viviana Culmone, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Abstract

Neural field models (NFMs) describe the spatio-temporal evolution of neuronal populations as a continuous excitable medium. The resulting tissue-level description can be employed to fit data from macroscopic recordings of electrocortical brain activity like the electroencephalogram (EEG) and local field potentials (LFPs). The standard neural field approach models the cortex as a two-dimensional sheet, neglecting the actual cortical depth. Although a small number of studies have considered the anatomical cortical layers to model different connectivity patterns, their mathematical description does not commonly use the cortical depth to determine the model dynamics. Therefore, within the framework of neural field theory, the impact of dendrites on brain activity remains far from being exhaustively explored.

In the present work, we extend the geometry of a two-dimensional (2D) NFM to incorporate a dendritic dimension for the excitatory neural populations, representing the cortical depth. Dendritic trees are modelled as linear cables, spatially discretized in multiple subsections (compartments). Spatio-temporal patterns of the new cortical model are studied for systems consisting of either a single or multiple microcolumns. A powerful approximation, extended from the one for the 2D NFM, is introduced to predict the power spectral density of the mean membrane potential from the Jacobian matrix of the linearized system evaluated at a singular point. Our numerical analysis reveals a variety of dynamics, ranging from those characterized by "flat" power spectra without alpha rhythmicity due to signal loss over the tree, up to sharp alpha resonances corresponding to proximity to a Hopf bifurcation. The research focuses on the identification of plausible EEG dynamics, e.g., those exhibiting a dominant alpha activity, conceived as the central rhythm of spontaneous EEG.

Crucial to this endeavour has been the careful tuning of key dendritic parameters introduced with the three-dimensional (3D) geometry, such as the "synaptic factor" (i.e. synaptic conductance) and the membrane length constant, and wider parameter sweeps using the Particle Swarm Optimization (PSO) technique. The dynamics are mainly studied for a single microcolum systems with different dendritic configurations (e.g. varying conductance and length constant) during synchronous and asynchronous synaptic activation in either a single or multiple dendritic domains.

Our results explain the impact of key dendritic parameters on the 3D NFM dynamics. Heuristics characterizing these effects can be regarded as representative of the well-known phenomenon of "dendritic democracy", classically indicating the normalisation of post-synaptic somatic potentials compensating for dendritic filtering activity. While several experimental studies have investigated the genesis of this compensation, to date this phenomenon has not been explored concerning a potential interplay with the alpha rhythm. Our findings suggest that physiological conditions enhancing the onset of action potentials in active models also promote alphoid dynamics in our passive neural field models including the dendritic dimension. In particular, synaptic strength has to increase with distance from the soma. We found several parameter configurations giving rise to alpha rhythmicity in the 3D geometry. Dynamical analysis highlights the impact of key dendritic parameters at different cortical depths on the genesis of alpha rhythm, providing a clearer insight into dendritic mechanisms and cortical dynamics. Indeed, the model can be used as a valid starting point for NF studies aiming to encompass further dendritic properties, implement more detailed connectivity schemes and incorporate data from depth electrode recordings.

Ai miei amati genitori, grazie ai quali
tutto nella mia vita è stato possibile

Acknowledgements

Undertaking this PhD has been a truly life experience for me and it would not have been achievable without the support and guidance I received from many people.

Firstly, I would like to express my sincere gratitude to my supervisor Prof. Ingo Bojak for his precious support of my PhD study, for his patience, enthusiasm and inspiring knowledge. Ingo, thank you for encouraging my research in all the time, especially during research challenges and difficulties, and thank you for providing advice on both research as well as on my career. Your invaluable guidance allowed me to grow as a research scientist and as a person.

My sincere thanks also go to my co-supervisor Etienne Roesch, who provided help to use the CINN's cluster and IT facilities of University of Reading. Without this precious support it would not have been possible to conduct this research.

I would especially like to thank my PhD examiners Ying Zheng and Steven Coombes for serving as committee members and for letting my defense be an enjoyable moment. Your brilliant comments and suggestions improved my knowledge widening my research from various perspectives. Ying, I would like to thank you for the kind advice on both my research and my future career provided during our enjoyable meetings. Your guidance has been a precious support during the correction process.

I thank all the researchers I have met during this fantastic journey, at scientific conferences, seminars and meetings. Thanks to my colleagues and labmates Mohamad Alquoatli, Antonis Markakis, Jingjing Luo, Michael Haylett, Zhivko Stoyanov, Gabriella Cefalù, Sungmin Kang, Asad Malik, Luis Patino, Catriona Scrivener, Timothée Thewetone, David Laszuk for the stimulating scientific discussions and for

the great time we spent together both as colleagues and as friends. Words can not express how grateful I am to Mohamad Alquoatli for listening, understanding and finding "magic" words to support and encourage me as a scientist and as a person, over the past years and today. A heartfelt thanks goes to Michael Haylett who has been my best flatmate, becoming a special and precious friend. Mike, thank you for sharing your passions, your intelligence and your ideas, brilliant and funny. Thank you for lightening my thoughts and making me feeling relieved, optimistic an confident in life by simply being yourself. You make my life a much better and smiling place!

Special thanks go to my students and colleagues of University College of London and King's College of London, where I had a fantastic experience as a Teaching Assistant. In particular, I thank Andrew Whiter, Senior Teaching Fellow at University College of London, and my student Raja Khan and his family for all the experiences shared and for reading some chapters of my thesis to provide help with the language. Andrew, working together has made me a better scientist and a better person. Raja, your genuine authenticity makes me optimistic about human beings. I think we shared the funniest classes ever... Well, I am actually not sure about who is teaching and who is learning! Yes, I am sure. Thank you!

I would like to thank all my friends who make my stay in Reading and in London a fantastic journey: Ania Smagała, Cornel and Cristina Vida, Kevin Lelu, Nacho Mapache, Tomoko, Ezlyn Barends, Saqib Javed, Dhanya Roji, Bruno Pacifico, Santo Neglia, Nunzio Puntillo, Andrea Moiola, Giorgio Mangano, Elena Di Fatta, Silvia Baldanza, Pietro Locatelli, Andrea Favale, Fabio e Gloria Serenelli, Sabrina Morciano, Sonia Roccella, Giacomo Bialetti, Andrea Peruffo, Sofia Filippi e Matteo Cattaneo.

Il più importante e sentito ringraziamento va ai miei genitori, Luigi Culmone e Giuseppa Marchese, per il loro costante supporto e per avermi sostenuta sempre, moralmente ed economicamente, con il loro amore e i loro sacrifici. Papà, grazie per avermi sempre spronata a concludere questo percorso, per avermi amata sempre e per essere un costante esempio di forza, tenacia e carattere. Mamma, grazie per

avermi compresa e aiutata sempre, grazie per la tua dolcezza e immenso amore e grazie per la forza e il coraggio che dimostri ogni giorno. Insieme, abbiamo realizzato un sogno, grazie all'amore che ci rende una squadra!

Grazie ai miei nonni, Giuseppa Simone e Vincenzo Marchese, eterno e indimenticabile esempio di amore e benevolenza. Una guida che illumina il mio cammino. Nel mio cuore e nel mio essere in ogni istante della mia vita.

Un grazie ultra-speciale va a mia sorella Milena Culmone, la più preziosa compagna di giochi e di vita. Milena, con la tua genialità mi aiuti, mi scuoti, mi sblocchi e mi accompagni infondendomi gioia, amore, coraggio, speranza e tenacia. Lo hai fatto durante il dottorato, quando credevo che non ce l'avrei fatta e invece ce l'ho fatta! Lo hai fatto quando mi hai incitata ad ultimare le correzioni a fine luglio a Londra. Lo fai sempre, in ogni cosa, in ogni dove e quando. Grazie ai miei nipoti Elisa e Michele Fonte. Siete una inestimabile fonte di gioia e d'ispirazione! Grazie per le avventure che abbiamo condiviso in questi anni a Londra, a Parigi e in Sicilia. Grazie per volermi bene, io ve ne voglio tantissimo e mi diverto tanto con voi. Grazie al mio amorevole e prezioso cognato Giuseppe Fonte, per la pazienza e l'affetto dimostrati da quando ci conosciamo.

Sono grata a tutta la mia famiglia, ai miei zii e cugini che sono tantissimi. Silvana e Tommaso Marchese, grazie per il vostro amore e il vostro costante esempio. La vostra più che preziosa presenza ha reso la mia crescita migliore e mi accompagna ogni giorno. Un grazie particolare va a mia zia Nina Culmone, esempio di saggezza e amorevolezza. Grazie zia Nina, per pensarmi sempre con amore sincero desiderando il meglio per me. Zia Silvana e zio Salvatore Culmone, grazie per il vostro costante affetto che ha arricchito la mia infanzia e la mia vita. Anna e Gaspare Simone, il mio soggiorno a Guildford è iniziato con voi e non poteva essere più bello! Che divertimento! Grazie per l'amore e per le risate. Ancora, ringrazio i miei cugini Miriana e Vincenzo Marchese che hanno cambiato la mia vita con il loro arrivo. Siete pezzi di cuore! Grazie Daniela e Rosario De Gaetano, per l'affetto sincero che avete sempre dimostrato e per i vostri consigli e la vostra preziosa presenza nella mia vita. Grazie Fabio Culmone, per avere giocato con me bambina.

Con molto calore ringrazio le mie zie francesi, Enzina e Tita Simone, due raris-

sime perle e bellissime rose. Grazie a voi ho conosciuto una gioia e un amore nuovi, che hanno cambiato il mio essere. Enzina, grazie per avermi fatto amare Parigi. Grazie per l'arte, la bellezza, la famiglia, Scarlette, Più più, i dolci, i menù e i profumi. Grazie per le storie, le passeggiate, le risate e la vita che ci doniamo. Non bastano le parole a descrivere quello che abbiamo nel cuore. Grazie, grazie, grazie!

Un grazie speciale va a Fabio D'Asaro che è stato presente in tutte le fasi di questa ricerca, per avermi sempre supportata e consigliata. Grazie per avere condiviso le gioie e i dolori di questo percorso. Grazie per la magia di questi anni, insieme, indescrivibili. Particolari e sentiti ringraziamenti vanno a Michele Balsamo, per il supporto offerto durante la stesura di questa tesi e di tutte le tesi precedenti, per le meravigliose discussioni scientifiche e il sostegno morale. Michele, grazie per credere in me e grazie per rendere il mondo un posto migliore. Sinceri ringraziamenti sono rivolti a Nunzio Puntillo, per avere scommesso su di me, per aver creduto che ce l'avrei fatta anche nei momenti più bui. Grazie per tutto il supporto fornito, per aver portato allegria, benessere e spensieratezza laddove mancavano. Grazie, per sapere mitigare i miei schemi con il tuo saggio sorriso. E grazie per sapere vivere aldilà delle regole. Grazie, Gabriele Berlich per l'aiuto e il supporto degli ultimi anni. Grazie per esserci stato e per esserci con tutta la tua dolcezza, per condividere con me la tua visione della vita, che mi regala un sorriso e mi infonde coraggio. Quello che abbiamo condiviso mi ha arricchita e resa migliore. Grazie a Claudia Insinga, per il supporto quotidiano e l'enorme forza e affetto trasmessi negli ultimi anni, pur senza vederci e grazie ai nostri messaggi vocali. Un grazie particolare è dedicato a Paola Campini Sorce, per volermi bene, avermi supportata ed essere diventata un'amica del tutto speciale! Grazie ai carissimi Roberto Signorello e Francesco Dolce per la nostra intramontabile cara amicizia. Grazie a Daniele Gerbino, per essere mio amico da quando sono nata. Grazie ai miei meravigliosi amici Eleonora Avila, Alexandro Bock, Fabio Cusumano, Alexandra Pierri, Giuseppe Colletta, Loredana Bonanno, Francesco Di Bartolo, Alessandra Simone (anche cugina), Gianluca Accardo, Alessio Nobile, Gemma Torre, Enrico Donadello, Stefania Morselli e Ilenia Ferrara, che con il loro affetto hanno indirettamente contribuito a questo lavoro. Un grazie particolare è dedicato alla mia psicologa Dottoressa Marialuisa Novara. Grazie per avere lavorato

insieme a me in questi anni. Credo sinceramente che la mia vita sarebbe andata diversamente se non ti avessi incontrata. Non sarei quella che sono. Grazie.

Infine, vorrei ringraziare me stessa, per la mia instancabile dedizione al lavoro, per aver dato prova di grande forza, volontà e resistenza. Viviana, grazie perché mantieni vivido l'entusiasmo e l'amore per la conoscenza. Grazie perché ti brillano gli occhi mentre lavori. Grazie per avere accettato, conosciuto e superato la solitudine e la tristezza di questi anni. Grazie per amare e gioire oggi.

Last but not least, I would like to thank myself, for my strong commitment to work, firmness and love for science. Viviana, thanks for keeping a fervid enthusiasm and love for knowledge. Thanks for feeling happy while working. Thanks for having known, accepted and overcome loneliness and sadness over these years. Thanks for loving and feeling joyful here and now.

Contents

Chapter 1

Introduction

Studying the structural properties and the architecture of neural tissues and cortex has been an important goal, partly achieved by biologists and neuroscientists in the last century.

The neural cortex can be mathematically and computationally modelled at different levels of description, each level relating to a particular set of neuroscientific data [1–3]. Single model neurons capture important features of the complex neurophysiological mechanisms of single cells, such as the role that ionic channels and synaptic properties play in the cellular behaviour [4, 5]. To understand the importance of these models, we need to highlight that the single cell level is the level at which the exchange of information between the constitutional elements of the brain occurs [6]. Obviously, data recorder from single neurons are well reproduced and sometimes predicted by such models [7]. Mesoscopic models [8–11] instead aim to reproduce observations emerging from neuronal action and communication at the level of microcolumns, cortical columns and macrocolumns. Results from these models can be compared to data from the electroencephalogram (EEG), the magnetoencephalogram (MEG) and the local fields potentials (LFPs) [11]. Finally, macroscopic models investigate the interactions between brain regions and large-scale neural systems. Dynamic Causal Modelling [12, 13] is a method for the interpretation of functional neuroimaging data at this level of description.

At any spatial scale, the resulting models are complex systems. The analysis of

neural system dynamics is important for understanding the neurobiological phenomena typically occurring in the cortex, both in normal and in pathological conditions. The typical dynamic states occurring in the cortex underlie many physiological functions, such as perception, learning and cognition [14–16] as well as pathological behaviours of cortical tissues damaged by neurological disorders [17, 18].

Although researchers have been investigating neuronal and network mechanisms for several decades, various questions remain far from resolved. For example, thanks to both experimental studies [19–21] and digital reconstructions [22] much has been learned about neuronal morphology. However, the molecular origin and functional implications of neuronal shape remain enigmatic. From the perspective of neurodegenerative disorders, greater attention to understanding the mechanisms underlying the cellular metabolism of neuronal and glial cells should enable progress on the genesis and evolution of neurologic diseases. Despite the important studies on the role of mithocondria [23, 24] and transport of metabolites, the energy and metabolic intracellular mechanisms are poorly understood. One of the major unsolved questions in contemporary neuroscience concerns the mechanism underlying the synapse formation. Despite several molecules have been identified as possible candidate for synapse formation [25–27], many essential questions about the signals involved are still unanswered [28]. Another well-known mechanism is the dendritic and synaptic integration [29]: in most neurons, thousands of synaptic inputs converge to generate a postsynaptic responses, but the importance of the spatial organization, the interaction and cooperation of the signals is still largely unexplained [28].

Dendrites act as filters for the incoming inputs [30], attenuating the somatic postsynaptic responses to synapses located in distal dendritic regions. Several recent studies provided evidence that this phenomenon is compensated by a variety of mechanisms normalizing the somatic responses, making them insensitive to the dendritic origin of the stimulus [31]. This compensation, often referred as "dendritic democracy", has been studied by many researchers [31–34] and, among all the possible underlying mechanisms, the synaptic strength scaling with distance seems to have a predominant role. However, the intrinsic cellular mechanisms essential for the genesis of the synaptic scaling are still largely unknown [31]. Moreover, incorpo-

rating such dendritic properties in mesoscopic neural models would lead to a better insight into the role of dendrites in cortical dynamics.

The core of the present work is the formulation of a computational cortical model. It is a three-dimensional (3D) Neural Field Model (NFM) of the cortex, including a dendritic dimension. So far, this issue has been addressed insufficiently in the field, although the role of the dendrite has been demonstrated to be fundamental both in synaptic integration and in neuronal transmission [30]. This work aims to bridge the gap between the well known two-dimensional NFMs [35] and other works pointing out the indispensability of considering the dendritic trees to explain the behaviour of neural systems [30].

Our starting point is the two-dimensional (2D) NFM developed by Bojak and Liley in 2005 [36]. They modelled the entire cortex as a two-dimensional sheet and reproduced the cortical activity both at rest and during anesthesia. They described cortical (both local and long-range) and extra-cortical connections between cortical microcolumns. Each microcolumn consists of a population of about 10^4 to 10^5 neurons and includes one inhibitory and one excitatory sub-populations. We extended the geometry of this model to include a dendritic dimension for the excitatory populations (while the inhibitory populations continue to be modelled as points in this work at least) and studied the cortical activity at rest of a small square piece of the cortex. The dimension of the cortical portion observed varies from 1 mm^2 to 16 mm^2, corresponding respectively to the activity of a single microcolumn extending its dendritic tree through the 4 mm depth of the cortex, and of 16 microcolumns distributed on the square sheet at a mutual distance of 1 mm, with their dendritic trees extending through the 4 mm depth of the cortex.

In the second chapter we motivate the reasons leading to this kind of modelling from literature. We firstly focus our attention on the scientific literature underlying neural field theories. We illustrate the scientific reasons behind the NFM formulation, briefly considering its scientific historical background. Several applications are then mentioned, together with some important achievements of this theory. We also outline the shortcomings and weaknesses of NFMs. Secondly, after having addressed

the fundamental morphologic properties of the cortex and of pyramidal neurons, we discuss the role that the dendrites can have in neural dynamics. The importance of the dendritic tree in neural activity provides the motivation to include the dendritic dimension in NFMs. The resulting models will exhibit a different behaviour, compared to the dynamics of models neglecting the presence of the dendrite.

In Chapter 3, we illustrate both the mathematical formulation and the numerical implementation (in Matlab), of the 2D and 3D cortical NFMs. We start providing a full description (both the mathematical and numerical) of [36]. We then explain all the physiological assumptions made to derive the mathematical equations of the three-dimensional cortical NFM, and illustrate the numerical approximation used for the implementation of the model as a system of ordinary differential equations (ODEs). This includes an elucidation of the compartmental cable equation [37], used to model the dynamics of the third dendritic dimension. Finally, we illustrate the methods used to analyse the resulting dynamics of the model. Since we are interested in reproducing the alpha rhythm typical for human cortical activity at rest, the methods used to calculate the power spectral density (PSD) of the mean membrane potential, both from the time series (non-linear spectrum) and from the Jacobian matrix of the linearised system evaluated at a singular point (linear spectrum), will be carefully explained.

In the fourth chapter we study the spatiotemporal activity of both 2D and 3D geometries at 4 different cortical depth, for systems consisting of 256 microcolumns distributed on a square 16x16 grid. Time series, non-linear and linear power spectra are studied for systems with either a single or multiple microcolumns, pointing out the differences in between the dynamics of the two geometries. A 3D model naively extending the 2D configuration,with all the parameters of the original model [36] unchanged, results in a loss of alpha activity due the spread on signal along the dendrite. Parameter modifications (within physiological bounds) are studied to restore the alpha rhythmicity for single microcolumn systems, with a particular focus on key dendritic parameters introduced in the 3D geometry, namely the electrotonic

length constant and the synaptic strengthening factor. However, due to the high number of parameters influencing the alpha rhythm of the system, we decided to use a more sophisticated optimisation algorithm, to find optimal parameter sets giving rise to alphoid dynamics.

In Chapter 5, we apply the Particle Swarm Optimisation (PSO) technique to detect PSDs similar in shape to the ones encountered in humans. After reviewing the PSO literature, we explain how we implemented the algorithm from scratch. Two search-spaces and two fitness functions have been considered. Several different sets of parameters leading to satisfactory PSDs have been found, for various locations of the synaptic input along the dendrite.

In the sixth chapter, we study the behaviour of single microcolumn systems, exploring the impact of the key dendritic parameters on the general dynamics and on the occurrance of alpha rhythm. Firstly, we study in great detail the role of the synaptic factor on the dendritic dynamics. The factor strengthening the synaptic input is the main novelty introduced in the three-dimensional NF model, and we dedicate particular attention to this topic. The dendritic behaviour is studied during simulations where a single pulse input is provided to the systems and all the cortical (local and long-range) and extra-cortical connections are switched off. This clarifies the intrinsic dendritic dynamics and motivates the introduction of the synaptic factor in the 3D-NFM. The impact of the dendritic length constant on the dendritic dynamics is also studied during pulse simulations, confirming the physiological role of the length constant illustrated in Section 3.2.3. Secondly, the relationship between the nature of the steady state of the system (equilibrium points or limit cycles) and the linear spectra is illustrated. Finally, we systematically study the PSDs depending on the location of the synaptic inputs along the dendrite, on the synaptic factor and on the dendritic length constant. Interesting configurations, with multiple simultaneous synaptic inputs in different dendritic locations, are briefly investigated, together with a new search space explored with the PSO technique.

In the final chapter we summarise our main contributions to the study of the dynamics of 3D-NFMs, pointing out some interesting future investigations that could be carried out on this topic.

The main functions and procedures implemented in MatLab to study the nonlinear and linear dynamics of the studied NFMs are reported, explained and commented in the Appendix, for either single or multiple microcolumn 3D NFM systems. The Appendix also illustrates the procedures implementing the Particle Swarm Algorithm, together with the main fitness functions that have been considered in the present work.

Chapter 2

Literature Review

In this chapter we present the NFM formulation, considering some of the most important scientific studies in the field. The central ideas of both neural mass and neural field theories are briefly illustrated, together with some of their main applications. Secondly, after a brief discussion about the morphologic properties of the cortex and a description of pyramidal neurons[1], we motivate the importance of introducing a dendritic dimension in NFMs. Several dendritic mechanisms are considered together with their impact on the cortical dynamics. In the present work, we have inlcuded a dendritic dimension in a NF model. Although not all the mechanisms illustrated in this chapter have been included in our model, the properties reviewed here could be considered in any future work aiming to incorporate a more detailed dendritic structure in NF models.

2.1 Mesoscopic and Neural Field Models.

Although single neuron models are fundamental to explaining the electric mechanism of communication between single neuronal cells, there are several factors leading to the necessity to investigate the neural circuitry at a mesoscopic level of organization [35, 38]. Firstly, several non-neuronal elements have been observed

[1]In the present work, we modelled both inhibitory and excitatory cortical populations. As excitatory populations, we mainly modelled the activity of pyramidal neurons.

to be crucial in determining the behaviour of neural systems. For example glial cells, with their synaptic interactions with neurons, have a significant role in determining the firing rate of the neurons [39, 40], and the neuronal interconnections. Secondly, non-invasive measurement methods as electroencephalogram (EEG), magnetoencephalogram (MEG) and functional magnetic resonance imaging (fMRI) observe only field signals, emerging from neural masses instead of from single neurons. For these reasons, networks of cortical neurons, with their connectivity and microcircuitry, explain the behaviour of neural systems without including single neuron detailed properties, to avoid an explosion of the model complexity.

2.1.1 Bulk models, ensemble density approach, mass models and neural field models

The basic idea of any type of mesoscopic description is to observe and study neural systems as bulk models, describing the interactions between neural masses or populations of neurons [10]. The idea is based on a couple of neurobiological properties of brain communication: firstly most of neuronal cells use only one kind of chemical neurtransmitter; secondly the cortex is spatially organised in localised domains of population interacting with each other through synaptic transmission. This homogeneity of characteristics makes it possible to adopt a model of description in which neuronal populations can be regarded as excitatory or inhibitory masses, leading to substitute the neural microcircuits of neural networks with the cortical mesocircuit of a mesoscopic level of description.

The most natural descriptive approach, in this regard, is the ensemble theory, used in statistical physics to deal with any complex system characterised by a massive number of elements. In our case, the approach of the classical mechanics fails to account for the explosion of complexity of the systems, if one wants to describe each neuron with all its physiological properties. On the contrary, it has been demonstrated that, describing many particles systems in terms of their probabilistic behaviour, leads to an efficient and tractable characterization. In the case of neural systems, the common approach provides a statistical description of the quantities that can be empirically measurable [41]. They typically are the *mean soma mem-*

brane potential [42], or the *mean neuronal firing rate* [43, 44], and can be averaged with respect to space or time, depending on which phenomenon is being described.

This averaged neuronal ensemble description can lead to two different types of models: *"neural mass models"* and *"neural field models"*. The first description is a discrete approach concerned with bulk models of spatially localized populations. On the contrary, the second approach is continuous with respect to space and time [41]. Neural populations in the cortex establish local and long-range connections, influencing many other regions of the brain. In the perspective of the *neural field models*, each neuronal ensemble defines its connectivity scheme, like a *"field"* propagating its influence on to the other ensembles. Depending on the averaged variable (firing rate of the neurons or somatic membrane potential), they can be respectively *activity-based models* or *conductance-based models*. The voltage based models are more suitable to represent data from EEG. In the present work we use this type of formulation.

NFM studies, originally based on the works of Amari [45, 46], Wilson and Cowan [43, 44], have been addressed by many scientists using slightly different approaches, but all converging on similar formulations of the model, describing the mesoscopic dynamics through a system of nonlinear integrodifferential equations, with an associated kernel representing the spatial distribution of synaptic connectivity. Here, we show the general formulation of neural field model equations, briefly mentioning the goals that the scientists have achieved in the last decades. In fact, many biological properties have been progressively included in these models to take into account more and more relevant aspects. However, it has to be noted that in order to make the integral equations mathematically and computationally more tractable, many groups of scientists have managed to provide an equivalent formulation in terms of partial differential equations (PDE).

One of the most important NFM formulation is due to Amari [45, 46]. It is a voltage-based continuum model, considering m different neuronal populations. The field equations have the form

$$\tau_i \frac{\partial u_i}{\partial t} = s_i(\mathbf{x}, t) - u_i(\mathbf{x}, t) + \sum_{j=1}^{m} \int d\mathbf{x}' w_{ij}(\mathbf{x}, \mathbf{x}', t) f_j \left[u_i(\mathbf{x}', t)\right] \tag{2.1}$$

where

- $i = 1, ...m$;
- $u_i(\mathbf{x}, t)$ is the difference between the mean somatic and the resting membrane potential of neurons of type i, at position $\mathbf{x}$ and time t;
- s_i is the extracortical input;
- $w_{ji}(\mathbf{x}, \mathbf{x}, t)$ is the synaptic connectivity kernel, representing the synaptic connection that neurons of type j, at position $\mathbf{x}'$, form to neurons of type i, at position $\mathbf{x}$;
- f_j is the averaged firing rate depending on the membrane potential u_j.

This equation expresses that the neuronal mean membrane potential, in a specific position and time, depends on the "field" of other populations (located in other positions) through the non-linear firing rate f (that is a function of the membrane potential) and the synaptic connectivity w_{ji}. The contribution of the m neuronal population are considered in the equation, and integrated over time and space. In this formulation the time delay has not been considered. Some other hypothesis are commonly assumed for the synaptic connectivity such as the omission of the explicit dependence on time t and the dependence of w_{ji} on only the distance between the point $\mathbf{x}$ and $\mathbf{x}'$, so that

$$w_{ij}(\mathbf{x}, \mathbf{x}) = w_{ij}(|\mathbf{x} - \mathbf{x}'|) \equiv w_{ij}(r). \tag{2.2}$$

Moreover, the *inverted Mexican hat* is the shape that is usually adopted for the connectivity: that is to say $w(r)$ is considered to be positive (excitatory connectivity) in the neighborhood of $\mathbf{x}$, when $r < r_0$ and zero (inhibitory connectivity) elsewhere. We decided to mention this particular formulation of Neural Field Models because it is similar to the formulation that we will introduce in the next chapter to present our model.

Previous continuum neural field models, such as the works of Beurle [47] and Griffith [48, 49], only include excitatory populations. Wilson and Cowan introduced in their activity-based model the inhibitory populations [43, 44]. Their work is

also important because they used a sigmoidally shapedfiring rate, and a mesocircuit contemplating all the possible inhibitory and excitatory connections between the populations. Freeman formulated a mass model to describe the olfactory bulb [10]. He elaborated a hierarchy of connections between neural populations, yielding to a sets of third order differential equations. Other scientists investigated the role that postsynaptic potential (PSP) delays have on the cortical neural field model dynamics ([36, 50–53]).

Griffith [48] was the first who elaborated an equivalent formulation of NF integral equations in terms of partial differential equations (PDE). However, it was with Jirsa and Haken [54] that this approach became popular.

2.1.2 Applications and alpha-rhythm prediction

NFMs have been used to model electrocortical activity and large-scale neuronal dynamics, providing an interesting description of some cognitive states. Instabilities and spatio-temporal patterns have been modelled [55, 56] using PDE formulation of NFMs. Many scientists used NFM theory to study the anaesthesia induced changes in cortical activity [36, 57–59]. Other phenomena, explored using NFM descriptions, are the human sleep cycles [60, 61], the consolidation of memory during sleep [62, 63] and the neuronal activity during learning process [64–68]. Moreover, several studied have provided a continuum neural-field description of some dynamical states, as non-equilibrium behaviour (*metastability*) [69–73], epileptic [36, 74–81] and gamma activity [82–85]. Finally, NF formulations have been used to describe fMRI BOLD recordings and simultaneous EEG/fMRI [12, 72, 86–103].

The alpha activity was first discovered by Berger in the 1920s [104], and later confirmed by the studies of Andrian and Mathews in the 1930s [105]. While it is classically defined as oscillatory activity recovered over occipital regions (which are reactive to eyes opening and closing), with frequencies between 8 and 13 Hz, activity in the same range of frequency can be recorded from different cortical regions. Indeed, alpha rhythm is the most common rhythm observed in scalp-recorded electroencephalogram and it plays a fundamental role in explaining brain cognitive and

behavioural functions. Two different approaches have emerged to explain the genesis of the alpha rhythm. The first approach assumes that the alpha rhythm arises from cortical neurons driven at alpha frequencies. This phenomenon can emerge from the intrinsic oscillatory properties of cortical neurons [106, 107] or from the oscillatory activity of a feed-forward subcortical structure, as the thalamus [108, 109]. According to the second approach the alpha activity emerges from the interactions between cortical neuronal populations synaptically connected, and from the interactions between the cortex and the thalamus [110]. The second approach has been supported by different theorethical and computational studies [42, 110–114, 114].

Here we want to highlight that, although discrete neural networks have been successfully used to predict alpha activity [112] , the continuum or mean-field approach is more suitable to describe alpha rhythm [42, 54, 114].

2.1.3 Advantages and drawbacks of neural field approach

Representing neural tissues through bulk models has several advantages: firstly it allows to model some properties that cannot be addressed with single neurons models, such as the influence that glial cells have on synapses and neuronal firing rate; secondly behavioural patterns emerge from the interaction of populations of neurons; finally, the spatial scale of description of NFMs is close to the spatial resolution of neuroimaging data from EEG, MEG and fMRI, yielding the possibility to easily compare NFM results with neuroscientific data.

However, there are some aspects that cannot be included in a NFM description, such as the effects of fluctuations and correlations in single neuron activity [115], and the synergetic effects that microscopic, mesoscopic and macroscopic activities have on each other, when considered simultaneously [116]. Nevertheless, neural field model theory can be considered as a fundamental starting point to develop a more holistic approach, including multiscale levels of description.

2.2 Cortical structure

The neocortex of human brain is the most developed of the brain tissues and it is considered to be the structure where cognitive tasks take place. It is between 1 and 5 mm thick, and it has a surface of about $\sim 0.19 \mathrm{m}^2$ [117]. About $\sim 2 \cdot 10^{10}$ neurons [118], each having on average 6,900 synaptic connections with other neurons [119], are distributed in the neocortex, also containing the same number of non-neuronal glial cells and microglia. The mammalian neocortex presents a well organized structure at different spatial scale of observation and can be divided into six horizontal layers, characterized by different cellular densities, types, morphology and by different patterns of cellular connectivity with cortical and subcortical elements. Although several differences have been observed in the cortex of different mammals [120], a number of common features can be identified, in terms of cellular elements, spatial organization. All these features allows the formulation of tractable models describing highly complex cortical tissues.

It has to be noted that the neural cortex is composed of two types of ontogenetic structures: the *homogenetic* cortex, that is the neocortex; and the *heterogenetic* cortex, that includes structure like the olfactory bulb, the amygdala and the hippocampus, whose organization is different from the six layered neocortex.

Columnar organization of neocortex

It is widely accepted that the cortex can be divided into vertically oriented columns. The presence of vertically stacked cellular laminae has been observed in the cerebral cortex in many studies. Examples of vertical structures are the clusters of apical dendrites of layer V pyramidal neurons [121], bundles of myelinated axons of pyramidal cells (known as *radiations of Meynert*), columns of pyramidal cell bodies and clusters of interneurons axonal collaterals. In the neocortex it is possible to distinguish vertically well connected populations of neurons, whose horizontal connectivity is defined by the lateral axonal ramifications, expanding over different spatial scales [122]. The concept of columns had been hypothesised for the first time by Mountcastle [123]. He formulated the idea of macrocolumns as aggregation

of minicolumns bound together by short range excitatory and inhibitory horizontal connectivity [124]. Cortico-cortical or neocortical columns, instead, can be identified as cylindrical arborization volumes exhibiting long-range connections from a single cortico-cortical fibre [125–128].

2.3 Pyramidal neurons

There are several kind of neurons in the brain, that differ for size, location, morphology and functions [129]. In [130], Markram classifies dozens of neuronal subtypes, all belonging to two fundamental classes of cellular elements: pyramidal neurons, and non-pyramidal neurons. The former are the most numerous neuronal class (between 65 and 80% of cortical neurons [122, 131]) and are excitatory. In the neocortex they occur in all layers. On the contrary, the majority of non-pyramidal neurons are inhibitory, although some excitatory neurons of this type have been found. In the neocortex they occur in layers II to V. Since EEG signals are mostly due to the activity of pyramidal neurons of layer V [132, 133], in the computational model implemented here (see chapter 3) we consider this type of neurons. Moreover, we also modelled inhibitory neurons.

Pyramidal neurons are characterized by two distinct domains: the basal and the apical dendrites descending respectively from the base and the apex of the soma, that has a pyramidal shape [134]. The basal dendrites are usually short. Typically one main apical dendrite derives from the apex soma, bifurcates many times (giving rise to oblique dendrites, each of them eventually bifurcating again) and terminates with a tuft of dendrites in the distal region. Although they can widely vary between different layers, brain regions and species [135, 136], the presence of the common features in the dendritic tree shape, mentioned above, suggests that the distinct dendritic domains can be specialized to take on different functions. As explained in the next section, each domain receives distinct synaptic inputs originating from distinct brain regions and has a different excitability, integration and plasticity properties, confirming the idea of specialized domains performing various functions. Moreover variations in the structure and in the molecular composition of pyramidal neurons

also suggests that different types of pyramidal neurons are specialized for different functions.

2.4 Why include the dendrite in NFMs

Remarkably, the dendritic tree constitutes about 90% of the cellular surface. Including all the synapses and the spines, the tree is the most important volumetric component of the cortex, consuming 60% of the energy of the brain [137]. The dendritic tree can be regarded as a nonlinear information processor [137], that influences, with its complexity, not only the neuronal intrinsic microcircuits, but also neural activity at level of local circuits and networks [3, 138].

There are fundamental reasons providing a strong motivation to include dendrites in NFMs: first of all, it has been demonstrated that the presence of the dendrite can considerably change the network dynamics. But for the studies of Bressloff et al [3, 138], who explored the effects of the dendritic structure on single neurons and neural populations, and for the work of Coombes [139], this issues has not been targeted so far. However, Bressloff models including the dendritic tree are not based on the NFM formulation. In [138] the authors used one dimensional array of neurons with the dendritic tree modelled as a uniform cable, whereas in [3] they studied some similarities between model of the dendritic tree and physical systems that are theoretically well known to all physicists. The dynamics of one and two-dimensional continuum neural fields were also studied [140], without including the dendritic dimension. Conversely, the work of Coombes at al [139], which is providing an overview of dynamical system tools used to study patterns of neural activity, discusses a particular NFM including both synaptic and dendritic processing properties. Following the work of Bressloff [3], the scientists used the Rall compartmental theory [141] to describe a cortical NFM with axo-dendritic patterns of synaptic connectivity. The model studies the activity of one single population, without considering spatial extensions in the direction transversal to the dendritic tree. With the exception of the work of Coombes, in NFM theory the models so far developed have neglected the presence of the dendritic tree.

Another important reason why we think it is essential to include the dendrites in our NFM is the strong connection between dendrite structure and synaptic mechanisms. Synaptic spatial and temporal integration, synaptic filter role of the dendrite, Hebbian learning, spikes generation, neuronal firing rate, synaptic plasticity, the synaptic background noise are all example of mechanisms having a fundamental role in determining many of the most important brain functions, and they are strictly related to dendritic organization. Other dendritic properties that can be relevant in determining the behaviour of the networks are the distribution of the ionic channels on the membrane surface and the role of the dendritic branches and subunits. In the following sections we want to explore in more details the role that dendrites can have on the dynamics of the networks and on neuronal and synaptic mechanisms.

2.4.1 Dendrite and network dynamics

Although most of the models neglect the presence of the dendritic tree, it has been demonstrated that it can change the dynamics of the network. Using linear stability analysis, bifurcation theory and numerical simulations, it has been demonstrated that the effect of the diffusion along the dendritic tree can induce the formation of dynamic spatial pattern [138]. In [3] the effects of dendritic structure on single neurons and on neuronal populations are addressed. Analogies between neuronal processes and some physics systems are explored, to highlight both the organization of neuronal synaptic activity and the pattern formation and synchronization phenomena in neuronal populations.

2.4.2 Dendrite and neuronal mechanisms

Some of the mechanisms mentioned in this section can concern many types of neurons. Here, however, we are specifically referring to the properties of pyramidal neurons.

Dendrite morphology and synaptic integration

As previously explained, all pyramidal neurons share a common shape, in which it is possible to distinguish the axon, the pyramidal shaped soma, and two dendritic regions (basal and apical). Pyramidal neurons receive synaptic inputs in all the region of the membrane, but inputs in different neuronal locations originate from different brain regions. Usually proximal dendrites receive excitatory inputs from local sources, while distant tuft dendrites receive inputs from distant cortical or thalamic regions. For this reason, several studies distinguished in layer V pyramidal neurons six distinct domains (somatic domain, three apical domains and two basal domains). This distinction suggest that inputs from different domains may be integrated differently [142, 143], as confirmed by the studies of Gasparini and Losonczy on main apical and oblique dendrites inputs [144–146]. Some studies suggest that there is also a connection between neuronal firing patterns and the geometry and the organization of dendritic branches [147, 148].

Distance-dependent synaptic integration

The integration of synaptic inputs strongly depend on the location of the synapses on the dendritic tree. Usually synapses at different distances from the soma contributes differently to the initiation of action potentials in the axon. Distant synapses have less strength in this mechanism because of the dispersion of charge along the dendrite, from the eliciting site to the soma and axon [149, 150]. However, compensation mechanisms, consisting in the increase of distal synaptic conductance to balance the loss of charge, have been observed in CA1 pyramidal neurons [151–153], but not in layer V pyramidal neurons [154]. Other mechanisms involved in the initiation of action potential, through their interactions with distal synapses are the voltage-dependent dendritic channels, the high impedance of small-diameter dendrites, the dendritic spikes and depolarization. These and other interaction mechanisms suggest that synaptic integration is a complex process that may take advantage of the different location and efficacy of the synapses to perform more specialized functions [29].

Time-dependent synaptic integration

Distance-dependent synaptic efficacy is strictly correlated with time-dependent synaptic integration. Usually, to determine the axonal initiation of an action potential a coincident activation and sum of several inputs is necessary. Coincident distal synaptic inputs, spikes from multiple dendritic branches or domains need to be summed to reach the action-potential threshold. The relative timing of activation of the different mechanisms is fundamental for action-potential generation both during asynchronous and synchronous activation. In the first case distal synaptic inputs are integrated with proximal synaptic inputs. In the case of synchronous activation dendritic spike can occur, triggered by distal synaptic activation [145, 155, 156]. In layer V pyramidal neurons strong synaptic activation can lead to depolarizing spikes in distal dendrites, due to Na^+ and Ca^{2+} ionic currents [157, 158]. Ca^{2+} spikes in basal or apical dendrites can also be triggered by high-frequency action potential firing [159, 160], or by backpropagating action potentials paired with distal excitatory post-synaptic potentials (EPSPs) [161, 162].

Dendrite as spatio-temporal filter of synaptic inputs

Because of its role in the selection of incoming inputs, the dendritic tree is a spatio-temporal filter of synaptic inputs. The response of the soma to synaptic inputs, in fact, strongly depends on the time it takes for signals to spread along the dendritic tree. Consequently, the geometrical extension of the dendrite, its passive electric properties, the location of spines and synapses, the relative time of activation of synapses at different locations receiving inputs from different brain regions are all mechanisms that determine the filtering role of the dendrite [3]. Moreover, mechanisms like the neuro-modulation and the background synaptic activity can modify some electric properties of all or part of the dendritic tree. For example, the activation of ionic channels in the spines can influence the synaptic response. This dynamic behaviour of the dendrite with developmental changes can be involved in learning and memory.

The passive spread of the signal along the dendrite can be efficiently described

with a linear partial differential equation of second order in space and first order in time [163]. Initially developed for a single branch nervous cell [164], it can be modified to describe the geometry of a branched tree [163]. Rall was the first who developed the idea of compartmental models: the dendritic cable can be segmented in subsections or *compartments* that are sufficiently small, such that within the subsection the membrane potential can be considered constant [37]. The finite difference approximation can be used to calculate the membrane potential in the compartments, leading to the reduction of the partial differential equations of cable theory to a first order system of ordinary differential equations. This approach allows one to include and model in each compartments many biological details, such as different properties of synaptic inputs, dendritic properties depending on the voltage and chemical gradients. Each dendritic branch can be modelled as a linear cable and, knowing the synaptic distribution on the dendritic tree one can calculate the synaptic current entering the soma.

Dendrite and synaptic plasticity

One of the most studied form of synaptic plasticity is the spike-timing-dependent-plasticity (STDP). Long term potentiation (LTP) and long term depression (LTD) of synapses occur when the excitatory postsynaptic potential (EPSP) is paired with the backpropagating action potential from the initiation site in the axon back to the synapse. The relative timing of the two mechanisms (eliciting EPSP and backpropagation of action potential) is responsible of the induction of plasticity mechanisms [165–169]. Although many mechanisms remain unclear, especially the synaptic and firing conditions in which LTP and LTD occur *in vivo*, many studies suggest that dendritic spikes initiation have a significant role in plasticity induction in many physiological conditions (strong synaptic stimulation [170, 171], postsynaptic [167, 172] or presynaptic bursts [170]). In cortical pyramidal neurons of layer V, the amplification of the backpropagating action potential by dendritic depolarization is necessary to potentiate small EPSPs of distal synapses [173]. Moreover, depending on time

and location of the two associative signals, Hebbian LTP[2] can occur [174]. It has to be noted that also the organization distribution and plasticity of the spines play a fundamental role in determining neuronal excitability.

Dendritic intrinsic properties and neuronal firing

The distribution of voltage-gated ionic channels on the dendrites has an important role in determining the neuronal firing. They in fact can modify some of the intrinsic properties of the neuron, such as synaptic integration [175, 176], action-potential threshold, afterhyperpolarization and afterdepolarization. Moreover, as mentioned above, voltage-gated currents can compensate for the low efficiency of distal synapses, eliciting local dendritic spikes. Other mechanisms that can influence the neuronal firing rate are the homeostatic plasticity [177], the synaptic background noise [3] and modulation of synaptic integration due to neurotransmitters.

[2]To explain the synaptic plasticity, Hebb's rule claims that an increase in synaptic efficacy arises from a presynaptic cell's repeated and persistent stimulation of a postsynaptic cell.

Chapter 3

Methods

In this chapter, the mathematical formulation of a three-dimensional (3D) cortical NF model, extending the two-dimensional (2D) NF model [36], is illustrated. Details are provided about the numerical implementation and methods used to analyse the dynamics.

3.1 Two-dimensional Liley and Bojak model

Our starting point is the model of Liley *et al.* [42, 59] used by Liley and Bojak [36] to model the effects of anesthesia on the electroencephalogram. It is a two-dimensional NF model describing the activity of cortical macrocolumns, where each microcolumn consists of $10^4 - 10^5$ neurons. The description is made in terms of spatial averages over all kinds of relevant parameters and variables, including the *mean membrane potential* and the *mean firing rate*, while no averaging is considered with respect to time. A schematic illustration of the model is given in Fig. (3.1). Each macrocolumn contains one excitatory (E) and one inhibitory (I) subpopulation, locally interacting with each other through their mean firing rate S_k. These interactions also depend on the number of *intracortical* connections N_{lk}^{β}, where $l, k = e$ (excitatory), i (inhibitory). The subscritps lk indicates that a population of type l is acting on a population of type k. The macrocolumns also interact with each other through the *long-range cortical* connections Φ_{lk}. Only the excitatory subpop-

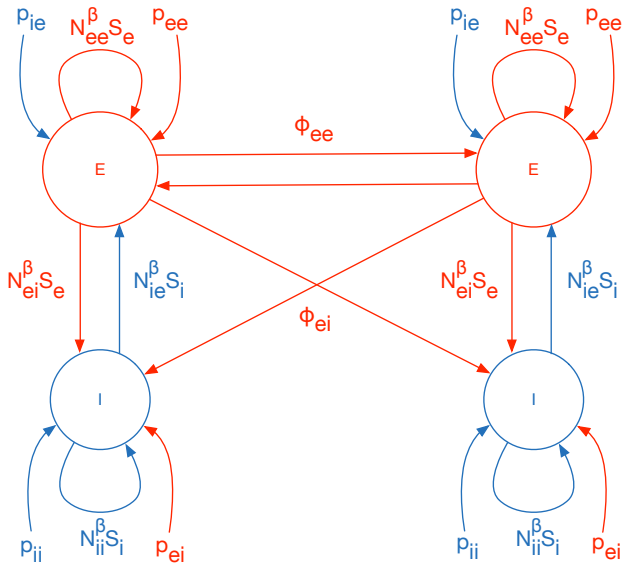

Figure 3.1: Schematic illustration of the Liley and Bojak two-dimensional model [36]. Two cortical macrocolumns with their connections are represented. Each macrocolumn includes one excitatory and one inhibitory sub-population.

ulations are considered to form long-range connections Φ_{ek} to other macrocolumns, both excitatory and inhibitory. Conversely, the inhibitory cortical connections are set to zero $\Phi_{ik} = 0$. Each subpopulation also receives excitatory and inhibitory inputs p_{lk} from *extracortical* regions (for example from the thalamus).

The following equations provide a mathematical formulation of this model:

$$\tau_k \frac{\partial h_k}{\partial t}(\mathbf{x}, t) = -[h_k(\mathbf{x}, t) - h_k^r] + \sum_l \psi_{lk}(h_k) I_{lk}(\mathbf{x}, t) \tag{3.1}$$

$$\left(\frac{\partial}{\partial t} + \gamma_{lk}\right)^2 I_{lk}(\mathbf{x}, t) = e\Gamma_{lk}\gamma_{lk}\left[N_{lk}^{\beta} S_l(h_l) + \Phi_{lk}(\mathbf{x}, t) + p_{lk}(\mathbf{x}, t)\right] \tag{3.2}$$

$$\left[\left(\frac{\partial}{\partial t} + v\Lambda_{ek}\right)^2 - \frac{3}{2}v^2\nabla^2\right]\Phi_{ek}(\mathbf{x}, t) = v^2\Lambda_{ek}^2 N_{lk}^{\alpha} S_e(h_e), \tag{3.3}$$

$$\Phi_{ik} = 0 \tag{3.4}$$

The first equation of the Liley model describes the evolution of the mean somatic membrane potential h_k in response to synaptic inputs I_{kl}; $\mathbf{x}$ is the position on the cortical sheet, h_k^r is the *mean resting membrane potential* (mV), and the function ψ_{lk} takes into account the *mean Nernst reversal potential* h_{lk}^{eq} (mV) through the equation[1]:

$$\psi_{lk}(h_k) = \frac{h_{lk}^{eq} - h_k(\mathbf{x}, t)}{\left|h_{lk}^{eq} - h_k^r\right|}. \tag{3.5}$$

The second equation describes all the contributes (local S_k, cortical Φ_{lk} and extra-cortical p_{lk} connections, respectively) to the synaptic input I_{lk}. The extra-cortical input p_{lk} is implemented as white noise. Finally, the third and the forth equations describe the evolution of the cortical inputs Φ_{lk}. The firing rate S_k is described by a sigmoidal function of the local mean membrane potential h_k:

$$S_k(h_k) = S_k^{max} \Big/ \left\{1 + (1 - r_{abs} S_k^{max}) \times exp\left[-\sqrt{2}\frac{h_k(\mathbf{x}, t) - \mu_k}{\sigma_k}\right]\right\}. \tag{3.6}$$

The derivation of Eq. (3.3) from the NFM theory will be discussed in Section 3.2.1. The meaning of all parameters appearing in Eqs. (3.1)-(3.3), with the corresponding units and physiological range of variability, are reported in Tables 3.1 and 3.2.

Bojak and Liley [36] implemented the model in MPI FORTRAN, to use parallel computation. The entire cortex has been simulated, considering a square cortical sheet of 51.2 x 51.2 cm^2, implemented as a discretized grid of 512 x 512 points. Each point represents a macrocolumn population, whose activity is described by the system of Eqs. (3.1)-(3.3). As a result, the entire system describing the cortex consists of 512 x 512 copies of the sub-system (3.1)-(3.3) and the spatial mean is calculated over 1 mm^2. The authors reduced Eqs. (3.1)-(3.3) to ordinary differential equations (ODEs) of first order, solved using *forward Euler* method [178]. Equation

[1]A more detailed description of function ψ_{ee} is provided in Section 4.5.

Parameter		Min.	Max.
Mean resting membrane potential	h_e^r	-80 mV	-60 mV
	h_i^r	-80 mV	-60 mV
Passive membrane decay time constant	τ_e	5 ms	150 ms
	τ_i	5 ms	150 ms
Postsynaptic potential amplitude	Γ_{ee}	0.1 mV	2 mV
	Γ_{ei}	0.1 mV	2 mV
	Γ_{ie}	0.1 mV	2 mV
	Γ_{ii}	0.1 mV	2 mV
Total number of intracortical connections	N_{ee}^{β}	2000	5000
	N_{ei}^{β}	2000	5000
	N_{ie}^{β}	100	1000
	N_{ii}^{β}	100	1000
Maximum mean firing rate	S_e^{max}	50 s^{-1}	500 s^{-1}
	S_i^{max}	50 s^{-1}	500 s^{-1}
Firing thresholds	$\overline{\mu_e}$	-55 mV	-40 mV
	$\overline{\mu_i}$	-55 mV	-40 mV
Std. deviation of firing thresholds	$\hat{\sigma_e}$	2 mV	7 mV
	$\hat{\sigma_i}$	2 mV	7 mV

Table 3.1: Physiological ranges of parameters of Liley and Bojak [36] model. Note that the range of variability of other parameters of the model are reported in Tab. 3.2. This table has been reported from Liley and Bojak model [36], TABLE I.

(3.3) is iterated in second order, using three time points

$$\frac{\partial^2 \Phi_{ek}(t)}{\partial t^2} = \frac{\Phi_{ek}(t+\Delta t) - 2\Phi_{ek}(t) + \Phi_{ek}(t-\Delta t)}{\Delta t^2}. \tag{3.7}$$

The Laplacian of Eq. (3.3) is numerically computed using the 5-points stencil approximation, i.e. considering the four nearest neighbours in two dimensions

$$\nabla^2 \Phi_{ek}(\mathbf{x}) = \left[\Phi_{ek}(\mathbf{x_u}) + \Phi_{ek}(\mathbf{x_r}) + \Phi_{ek}(\mathbf{x_d}) + \Phi_{ek}(\mathbf{x_l}) - 4\Phi_{ek}(\mathbf{x})\right] / \Delta \mathbf{x}^2 \tag{3.8}$$

where $\mathbf{x_u}$, $\mathbf{x_r}$, $\mathbf{x_d}$ and $\mathbf{x_l}$ are, respectively, the *upper, right, down* and *left* neighbours of the point $\mathbf{x}$ in the two-dimensional spatial grid. This approximation makes the sub-systems (3.1)-(3.3) spatially coupled. Since the connectivity in the cortex is supposed to be isotropic, the points at the boundaries of the square grid are geometrically connected to form a toroid.

Bojak and Liley found several sets of parameters yielding power spectra of frequencies for h_e reproducing the main features of resting EEG activity of human subjects, characterised by the presence of strong peaks in δ and α regions. Furthermore, they found that a linear approximation of the system is able to predict the non-linear system dynamics, providing enormous advantages with regards to numerical computations.

3.1.1 Implementation of Liley and Bojak model

For the purpose of clarity, we illustrate in this section the set of mathematical equations that have been implemented in MatLab to reproduce the 2D model [36], using the assumptions and numerical approximations described above in this Section. The set of parameters used in the simulations is shown in Table 3.3 and is one of the fundamental parameter sets used in [36]. Setting

$$\begin{aligned} \frac{\partial I_{lk}}{\partial t} &= \widetilde{I}_{lk} \\ \frac{\partial \Phi_{ek}}{\partial t} &= \widetilde{\Phi}_{ek}, \end{aligned} \tag{3.9}$$

and considering the 5-points stencil approximation for the Laplacian of the cortical inputs, the original PDE system (3.1)-(3.3) is transformed into an ordinary

Parameter		Min.	Max.
Mean Nernst	h_{ee}^{eq}	-20 mV	10 mV
reversal potential	h_{ei}^{eq}	-20 mV	10 mV
	h_{ie}^{eq}	-90 mV	$h_e^r - 5$ mV
	h_{ii}^{eq}	-90 mV	$h_i^r - 5$ mV
Postsynaptic potential	γ_{ee}	100 s^{-1}	1000 s^{-1}
rate constant	γ_{ei}	100 s^{-1}	1000 s^{-1}
	γ_{ie}	10 s^{-1}	500 s^{-1}
	γ_{ii}	10 s^{-1}	500 s^{-1}
Total number	N_{ee}^{α}	2000	5000
of cortico-cortical connections	N_{ei}^{α}	1000	3000
Cortico-cortical decay scale and	$\Lambda_{(ee=ei)}$	0.1 cm $^{-1}$	1 cm $^{-1}$
conduction velocity	v	100 cm/s	1000 cm/s
Rate of extracortical	$\overline{p_{ee}}(\delta p_{ee}/\overline{p_{ee}})$	0 s^{-1} (0.1)	10000 s^{-1} (0.25)
(noise) input	p_{ei}	0 s^{-1}	10000 s^{-1}
	p_{ie}	0 s^{-1} (fixed)	
	p_{ii}	0 s^{-1} (fixed)	
Mean synaptic delay	ξ	0 ms (fixed)	
Absolute refractary period	r_{abs}	0 ms (fixed)	

Table 3.2: Physiological ranges of parameters of Liley and Bojak [36] model. Note that the range of variability of other parameters of the model are reported in Tab. 3.1. This table has been reported from Liley and Bojak model [36], TABLE I.

differential equations (ODEs) system:

$$\begin{aligned}
\frac{\partial h_k}{\partial t} &= \frac{1}{\tau_k}\left\{-[h_e(t)-h_e^r]+\sum_l \psi_{lk}(h_k)I_{lk}(t)\right\}\\
\frac{\partial I_{lk}}{\partial t} &= \widetilde{I}_{lk}\\
\frac{\partial}{\partial t}\widetilde{I}_{lk}(t) &= -2\gamma_{lk}\widetilde{I}_{lk}-\gamma_{lk}^2 I_{lk}+N_{lk}^{\beta}S_l(h_l)+\Phi_{lk}+p_{lk}(t)\\
\frac{\partial \Phi_{ek}}{\partial t} &= \widetilde{\Phi}_{ek}\\
\frac{\partial}{\partial t}\widetilde{\Phi}_{ek}(\mathbf{x},t) &= -v\Lambda_{ek}\left\{2\widetilde{\Phi}_{ek}+v\Lambda_{ek}\Phi_{ek}(t)-v\Lambda_{ek}N_{ek}^{\alpha}S_e(h_e)\right\}+\\
&\quad+\frac{3}{2}\frac{v^2}{\Delta x^2}\left[\Phi_{ek}(\mathbf{x_u},t)+\Phi_{ek}(\mathbf{x_r},\mathbf{t})+\Phi_{ek}(\mathbf{x_d},t)+\Phi_{ek}(\mathbf{x_l},t)-4\Phi_{ek}(\mathbf{x},t)\right]
\end{aligned} \tag{3.10}$$

where $\Delta x = 1$ mm is the distance between two microcolums. The system above is describing the temporal evolution of the 14-dimensional vector

$$\mathbf{S} = (h_e, h_i, I_{ee}, \widetilde{I}_{ee}, I_{ei}, \widetilde{I}_{ei}, I_{ie}, \widetilde{I}_{ie}, I_{ii}, \widetilde{I}_{ii}, \Phi_{ee}, \widetilde{\Phi}_{ee}, \Phi_{ei}, \widetilde{\Phi}_{ei})^T \tag{3.11}$$

corresponding to the activity of a single microcolumn. Multiple microcolumn systems are described considering several copies of the system (3.10).

3.2 Three-dimensional model

As discussed in Chapters 1 and 2, the dendritic properties play a fundamental role in synaptic integration and generation of EEG signals. Therefore, including a dendritic description in the 2D-NFM of Liley and Bojak can provide a deeper insight in understanding synaptic connectivity and EEG activity.

NFMs describe both inhibitory and excitatory neuronal cortical activity. Pyramidal excitatory neurons of V layer are the most important in the generation of EEG signals [133]. Therefore, in our 3D-NFM we included the description of the dendritic trees of pyramidal cells, extending through the depth of the cortex. Inhibitory neuronal populations continue to be modelled as point-shaped [179].

h_e^r	-62.226 mV	h_i^r	-65.666 mV
τ_e	132.55 ms	τ_i	135.91 ms
h_{ee}^{eq}	-18.038 mV	h_{ei}^{eq}	-16.554 mV
h_{ie}^{eq}	-81.976 mV	h_{ii}^{eq}	-78.995mV
Γ_{ee}	0.10631 mV	Γ_{ei}	0.64105 mV
Γ_{ie}	46477 mV	Γ_{ii}	0.28663 mV
N_{ee}^{β}	2185.8	N_{ei}^{β}	3749.8
N_{ie}^{β}	466.30	N_{ii}^{β}	160.69
N_{ee}^{α}	4611.6	N_{ei}^{α}	1372.4
$\Lambda_{(ee=ei)}$	0.92809 cm^{-1}	v	684.24 cm/s
S_e^{max}	196.08^{-1}	S_i^{max}	454.40 s^{-1}
$\overline{\mu_e}$	-45.104 mV	$\overline{\mu_i}$	-43.910
$\hat{\sigma}_e$	3.8420 mV	$\hat{\sigma}_i$	4.5793 mV
$\overline{p_{ee}}$	6603.4 s^{-1}	δp_{ee}	660.34^{-1}
p_{ei}	2625.7 s^{-1}	$p_{ie,ii}$	0 s^{-1}
ξ	0 ms	r_{abs}	0 ms

Table 3.3: Parameter set of the implemented 2D model [36]. This table has been reported from Liley and Bojak model [36], TABLE III.

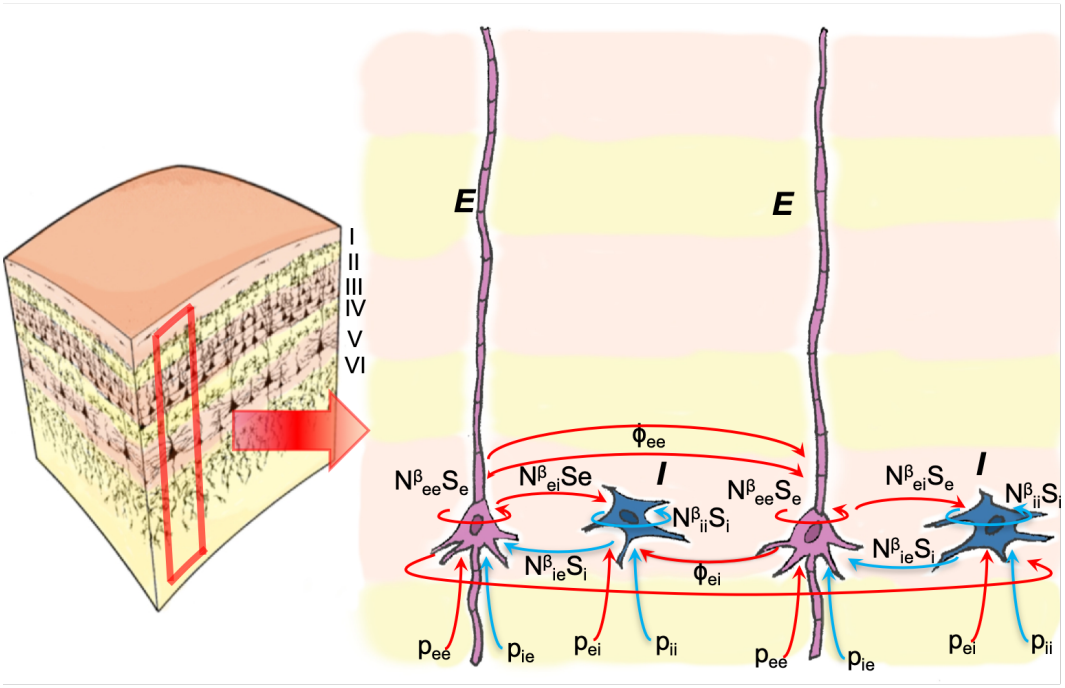

Figure 3.2: Schematic illustration of the 3D-NFM geometry and connections of 2 macro-columns. Each macro-column consists of one excitatory (E) and one inhibitory (I) sub-populations. Excitatory populations extend their dendritic tree through the 6 layers of the cortex, while inhibitory sub-populations are considered point-like. Short N^{β}_{lk}, long range (Φ_{ek}) cortical connections as well as extracortical (p_{lk}) connections are shown. Arrows in red represent connections from excitatory (E) subpopulations. Arrows in blue represent connections from inhibitory (I) subpopulations.

A geometrical illustration of the 3D Neural Field Model, with its connections is represented in Fig. 3.2. Two distinct macro-columns are illustrated in the figure. As in [36], each macro-column contains one neuronal population ($10^4 - 10^5$ neurons). Each population consists of one excitatory (E) and one inhibitory (I) subpopulation. Short N_{lk}^{β}, long range (Φ_{ek}) cortical connections as well as extracortical (p_{lk}) connections are shown. Arrows in red represent connections from excitatory (E) subpopulations, while arrows in blue represent connections from inhibitory (I) subpopulations.

To describe the dynamics of the 3D geometry illustrated in Fig. 3.2 the equations of the 2D model have been opportunely modified. The introduction of a third spatial dimension, representing the distance z along a dendritic tree, is necessary to model the depth of the cortex. In the present work, we introduced a spatial dependence of the mean membrane potential h_e from the z coordinate. Other variables are described through same equations used in [36]. However, as explained in the next paragraph, while the equation for the synaptic currents I_{lk} does not explicitly depend on z, it is possible to include in the model specific locations for the synapses along the dendrite.

3.2.1 Cortical connections

In this section, the derivation of the differential equation for the cortical input Φ_{lk} in the three-dimensional model is illustrated. We first show how to derive equation (3.3) of the 2D NFM from neural field theory. Secondly, assuming particular hypothesis, an analogous equation is derived for the three-dimensional model.

Cortical connections in the 2D model

Let $\Phi_{lk}(x,t)$ be the mean rate of presynaptic inputs arriving in x to population of type k from neurons of type l, located in a generic position $\mathbf{x}'$, at time t:

$$\Phi_{lk}(\mathbf{x},t) = \int_C d\mathbf{x}' \int_0^\infty dv f_{lk}(v|\mathbf{x},\mathbf{x}') w_{lk}(\mathbf{x},\mathbf{x}') S_l \left(\mathbf{x}', t - \frac{|\mathbf{x}-\mathbf{x}'|}{v}\right) \tag{3.12}$$

where f_{lk} is the velocity distribution function [2] and w_{lk} is the synaptic footprint. Consider the *Green's function*

$$G_{lk}(\mathbf{x}, \mathbf{x}', t) = \int_0^\infty dv f_{lk}(v|\mathbf{x}, \mathbf{x}') w_{lk}(\mathbf{x}, \mathbf{x}') \delta\left(t - \frac{|\mathbf{x} - \mathbf{x}'|}{v}\right). \tag{3.13}$$

Then Eq. (3.12) can be written as

$$\Phi_{lk} = \int_{-\infty}^{\infty} dt' \int_C G_{lk}(\mathbf{x}, \mathbf{x}', |t - t'|\ )\ S_l(\mathbf{x}', t')\ d\mathbf{x}' \tag{3.14}$$

where in the last passage the property of δ-function

$$f(x) = \int_{-\infty}^{+\infty} \delta(x - \alpha) f(\alpha) d\alpha \tag{3.15}$$

has been used.
In fact we have

$$\begin{aligned}
\Phi_{lk} &= \int_{-\infty}^{\infty} dt' \int_C G_{lk}(\mathbf{x}, \mathbf{x}', t, t')\ S_l(\mathbf{x}', t')\ d\mathbf{x}' \\
&= \int_{-\infty}^{\infty} dt' \int_C d\mathbf{x}' \int_0^\infty dv\ f_{lk}(v|\mathbf{x}, \mathbf{x}')\ w_{lk}(\mathbf{x}, \mathbf{x}')\ \delta\left(t - \frac{|\mathbf{x} - \mathbf{x}'|}{v} - t'\right) S_l(\mathbf{x}', t') \\
&= \int_C d\mathbf{x}' \int_0^\infty dv\ f_{lk}(v|\mathbf{x}, \mathbf{x}')\ w_{lk}(\mathbf{x}, \mathbf{x}')\ S_l\left(\mathbf{x}', t - \frac{|\mathbf{x} - \mathbf{x}'|}{v}\right)
\end{aligned}$$

As a common assumption, the synaptic connectivity is considered to be homogeneous and isotropic in space[3], so that

$$\begin{aligned}
w_{lk}(\mathbf{x}, \mathbf{x}') &= w_{lk}(|\mathbf{x} - \mathbf{x}'|) \\
f_{lk}(v|\mathbf{x}, \mathbf{x}') &= f_{lk}(v||\mathbf{x} - \mathbf{x}'|) \\
G_{lk}(\mathbf{x}, \mathbf{x}', t - t') &= G_{lk}(|\mathbf{x} - \mathbf{x}'|, t - t')
\end{aligned} \tag{3.16}$$

Then Eq. (3.13) holds for every $\mathbf{x}, \mathbf{x}'$ and can be considered as a function of $r = |\mathbf{x} - \mathbf{x}'|$:

$$G_{lk}(\mathbf{x}, t) = G_{lk}(r, t) \tag{3.17}$$

[2] the condition $\int_0^\infty dv f_{lk}(v|\mathbf{x}, \mathbf{x}') = 1$ holds.

[3] This means that all populations at positions $\mathbf{x}'$ having fixed distance r from $\mathbf{x}$, have the same effect of the population located at $\mathbf{x}$.

It is important to note that, under the particular conditions (3.16), the integral (3.14) has a convolution structure. Thus, in the Fourier space, one has:

$$\Phi_{lk}(k,\omega) = G_{lk}(k,\omega)S_l(k,\omega). \tag{3.18}$$

Furthermore, we considered a connectivity exponentially decaying with the distance $r = |\mathbf{x} - \mathbf{x}'|$, taking as *Green's function*

$$G_{lk}(r,t) = \frac{N_{lk}}{2\pi\sigma_{lk}^2}\exp(-r/\sigma_{lk})\delta\left(t - \frac{r}{v}\right) \tag{3.19}$$

where

- N_{lk} is the number of connections originating from neurons of type l and terminating on neurons of type k;
- v is the conduction velocity;
- $1/\sigma_{lk} = \Lambda_{lk}$ is the connectivity decay scale;
- the Dirac δ function expresses the distribution of delays.

Each pulse disturbance $\delta(\tau - r/v)$ travels with the same velocity v, and is reduced by the exponentially decaying synaptic connectivity. Applying to G_{lk} the Fourier transform in time one has

$$\begin{aligned} G(\mathbf{x},\omega) &= \int dt \exp\Big\{-i\omega t\Big\} G(\mathbf{x},t) = \\ &= \frac{w}{2\pi\sigma^2}\int dt \exp\left\{-i\omega t - \frac{r}{\sigma}\right\}\delta\left(t - \frac{r}{v}\right) = \\ &= \frac{w}{2\pi\sigma^2}\exp\left\{-r\left(\frac{1}{\sigma} + \frac{i\omega}{v}\right)\right\} \end{aligned} \tag{3.20}$$

where in the last passage the properties of the δ-function have been used and, for simplicity, the subscripts lk for σ_{lk} have been omitted. Applying the Fourier trans-

form with respect to space one obtains

$$\begin{aligned} G(k,\omega) &= \int\int_{R^2} d\mathbf{x^2} e^{ik*\mathbf{x}} G(\mathbf{x},\omega) = \\ &= \frac{w}{2\pi\sigma^2}\int_0^\infty \int_0^{2\pi} dr d\theta e^{ikr\cos\theta} G(\mathbf{x},\omega) = \\ &= \frac{w}{2\pi\sigma^2}\int_0^\infty dr e^{-r\left(\frac{1}{\sigma}+\frac{i\omega}{v}\right)} r \int_0^{2\pi} d\theta e^{ikr\cos\theta} = \\ &= \frac{w}{\sigma^2}\int_0^\infty dr e^{-\alpha r} r J_0(kr) = \frac{w}{\sigma^2}\frac{\alpha}{(\alpha^2+k^2)^{3/2}} \end{aligned} \tag{3.21}$$

where $\alpha = 1/\sigma + i\omega/v$. In the second-last step it has been considered that

$$\int_0^{2\pi} d\theta e^{ikr\cos\theta} = 2\pi J_0(kr) \tag{3.22}$$

where J_0 is the Bessel function of the first type. Finally, one can write

$$\begin{aligned} G(k,\omega) &= \frac{N_{lk}^2 v^2}{\sigma^2} \frac{\frac{v}{\sigma}+i\omega}{\left[\left(\frac{v}{\sigma}+i\omega\right)^2 \left(1+\frac{v^2k^2}{\left(\frac{v}{\sigma}+i\omega\right)}\right)\right]^{3/2}} = \\ &= \frac{N_{lk}^2 v^2}{\sigma^2} \frac{1}{\left(\frac{v}{\sigma}+i\omega\right)^2 \left[1+\frac{v^2k^2}{\left(\frac{v}{\sigma}+i\omega\right)^2}\right]^{3/2}} \end{aligned}$$

Using Taylor expansion, one has

$$G(k,\omega) = \frac{N_{lk}^2 v^2}{\sigma^2} \frac{1}{\left(\frac{v}{\sigma}+i\omega\right)^2 + \frac{3}{2}v^2k^2}. \tag{3.23}$$

Formula (3.23) is based on the common "long wavelength" assumption[4] used in [180]. Including this result in Eq. (3.18), substituting $\sigma = \Lambda_{lk}$ and operating an inverse Fourier transform with respect to both space and time[5], one obtains Eq. (3.3):

$$\left[\left(\frac{\partial}{\partial t} + v\Lambda_{lk}\right)^2 - \frac{3}{2}v^2\nabla^2\right]\Phi_{lk}(\mathbf{x},t) = v^2\Lambda_{lk}^2 N_{lk}^\alpha S_l(h_l). \tag{3.24}$$

where $1/\sigma_{lk} = \Lambda_{lk}$.

[4]This assumption is considering small values for the wave number k.

[5]The Fourier replacements $k^2 \to \nabla^2$ and $i\omega \to \partial/\partial t$ have been used here.

Cortical connections in the 3D model

In the three-dimensional case one has:

$$\begin{aligned}\Phi_{lk}(\mathbf{x},\mathbf{x}',z,t) = \\ &= \int_{\mathbb{R}^3} dx'\,dz' \int_0^\infty dv\, f\left(v \mid \mathbf{x},\mathbf{x}',z,z'\right) w(\mathbf{x},\mathbf{x}',z,z') S\left(\mathbf{x}, t - \frac{|\mathbf{x}-\mathbf{x}'|}{v}\right) \\ &= \int_{-\infty}^{\infty} dt \int_{\mathbb{R}^3} d\mathbf{x}' dz'\, G(\mathbf{x},\mathbf{x}',z,z',t-t') S(\mathbf{x}',t)\end{aligned}$$

where z is the dendritic coordinate and G is the Green's function

$$G(\mathbf{x},\mathbf{z},t) = \int_0^\infty dv\, f\left(v \mid \mathbf{x},\mathbf{x}',z,z'\right) w(\mathbf{x},\mathbf{x}',z,z') \delta\left(t - \frac{|\mathbf{x}-\mathbf{x}'|}{v}\right) \tag{3.25}$$

A variety of studies have revealed that the location of a synapse along the dendrite is often correlated with the relative distance for the interacting cells in the network. In particular, as the separation between neurons increases, the synapse tends to be located further away from the soma [181–183]. Although more realistic synaptic connectivity schemes [138, 140, 184] could be adopted to take into account this phenomenon, in the present work we adopted a simple connectivity scheme

$$w(\mathbf{x},\mathbf{x}',z,z') = w(z,z')\, w(\mathbf{x},\mathbf{x}') \tag{3.26}$$

where the factor $w(z,z')$ is 1 if the population at location $\mathbf{x}'$ and depth $\mathbf{z}'$ is creating a connection with the population at $\mathbf{x}$ and depth $\mathbf{z}$, and 0 otherwise[6]. The same connection scheme has been considered for all neuronal populations. This particular choice significantly simplifies the model allowing to adopt for the 3D long-range cortical connections Φ_{ek} a differential equation that is analogous to the 2D equation, while more realistic and diversified connectivity schemes could be considered in the future. For ease, it has also been assumed that the velocity distribution function f does not depend on the dendritic coordinate z:

$$f\left(v \mid \mathbf{x},\mathbf{x}',z,z'\right) = f\left(v \mid \mathbf{x},\mathbf{x}'\right) \tag{3.27}$$

[6]As indicated in the next chapters, this corresponds to considering synaptic inputs located at different compartments, i.e. at different depths $\mathbf{z}$.

Consequently, the Green's function (3.25) can be written as

$$G(\mathbf{x},\mathbf{x}',\mathbf{z},t) = w(z,z') \int_0^\infty dv f_{lk}(v|\mathbf{x},\mathbf{x}') w_{lk}(\mathbf{x},\mathbf{x}') \delta\left(t - \frac{|\mathbf{x}-\mathbf{x}'|}{v}\right) \tag{3.28}$$

Applying the Fourier transform, with respect to both time and space, one has:

$$G(k,\omega) = w(z,z') \ \frac{N_{lk}^2 v^2}{\sigma^2} \ \frac{1}{\left[\left(\frac{v}{\sigma} + i\omega\right)^2 + v^2 k^2\right]^{3/2}} \tag{3.29}$$

and

$$\left[\left(\frac{\partial}{\partial t} + v\Lambda_{lk}\right)^2 - \frac{3}{2}v^2\nabla^2\right] \Phi_{lk}(\mathbf{x},t) = w(z,z') \ \ v^2\Lambda_{lk}^2 N_{lk}^\alpha \ S_l(h_l). \tag{3.30}$$

3.2.2 Dendritic dimension

To describe the mean membrane potential h_e along the dendrite, a dependence on the dimension z has to be introduced in equation (3.1). An illustration of the three-dimensional model with the implemented connections is given in Fig. 3.2. To describe the diffusion of the signal along the dendrite, we used the Rall compartmental model [37, 185, 186], schematically illustrated in Figs. 3.3-3.5.

During the 1960s Wilfrid Rall proved that, given a certain set of assumptions [187], a symmetrically branched dendritic tree can be reduced to an *equivalent cylinder* (Fig. 3.3) [185, 188]. Taking into account the complex dendritic morphology, one should consider *each* dendritic branch as a cylinder of uniform passive membrane. Each cylinder in turn is characterized through a number of parallel *equivalent membrane circuits* (see Figs. 3.5 and 3.4), where the number of circuits depends on the length of the cable and each local circuit represents a portion of iso-potential membrane, corresponding to a "dendritic compartment". This electric model of the nerve membrane was known from a variety of previous studies [189–193] and can be used to represent both passive (Fig. 3.4) and active membrane[7].

[7]Differently from the passive circuits represented in Fig. 3.4, the active equivalent membrane circuits also include active properties as the conductance and batteries for the ionic currents flowing through the cell membrane [185].

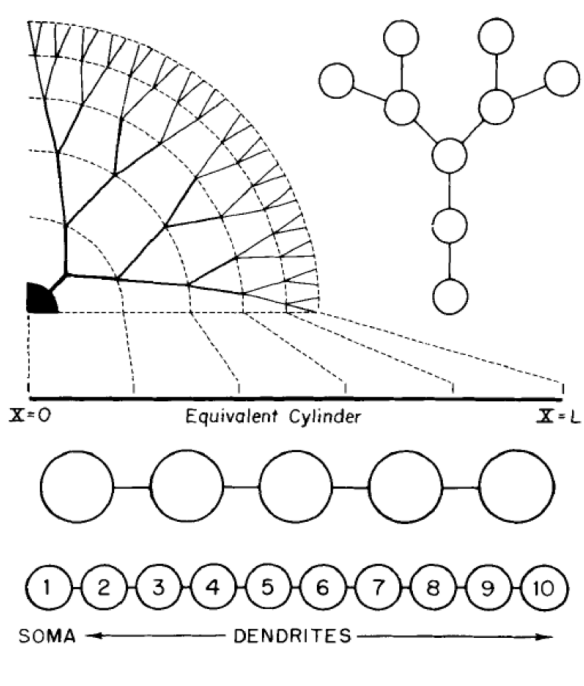

Figure 3.3: Diagram of a symmetrically branched dendritic tree, with corresponding mapping to an equivalent cylinder and to a chain of equal compartments. This figure has been adapted from wikipedia: https://commons.wikimedia.org/wiki/File:Rall_model_-_Equivalent_cylinder.png

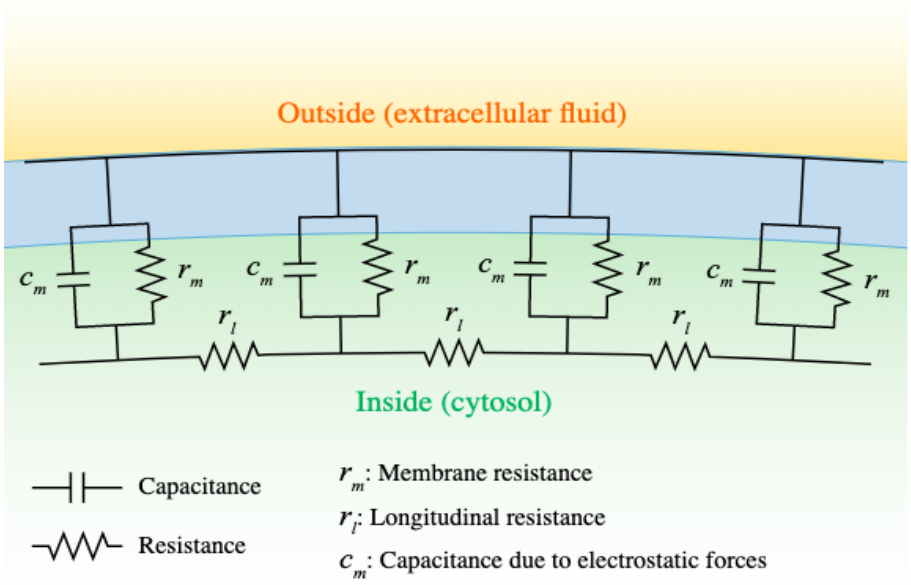

Figure 3.4: Schematic illustration of the Rall model theory, as a chain of parallel local equivalent membrane circuits. This figure has been reported from wikipedia: https://en.wikipedia.org/wiki/Cable_theory

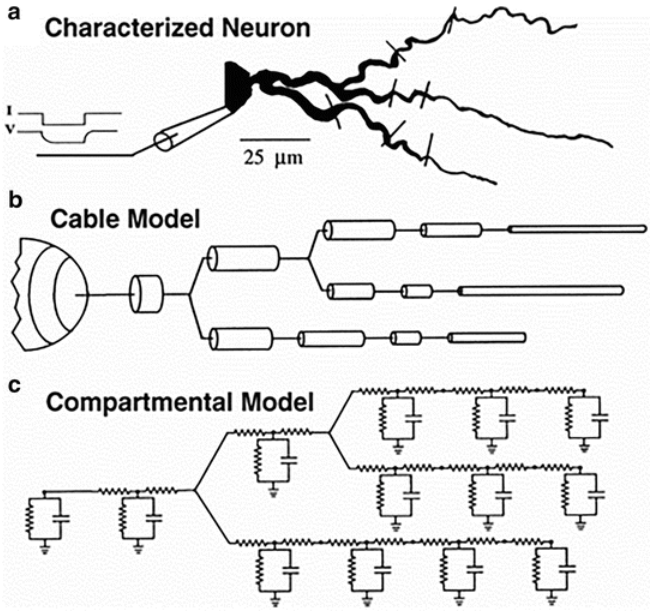

Figure 3.5: Schematic illustration of cable and compartmental model. This figure has been reported from [194], Fig.1, https://doi.org/10.1007/978-1-4939-3411-9_15.

The partial differential equation used to describe the distribution of the membrane potential along the cylinder [163, 185, 195, 196] is:

$$r_m c_m \frac{\partial V}{\partial t} = \lambda^2 \frac{\partial^2 V}{\partial z^2} - V + r_m I \tag{3.31}$$

where:

- z is the distance along the dendrite (centimeters);
- t is the time (seconds);
- $V = V(z,t)$ is the membrane *depolarization* (Volts), defined as the difference between *membrane potential* $V_m(z,t)$ and *resting membrane potential* V_r; $V = V(z,t) = V_m(z,t) - V_r$;
- r_m is the membrane resistance (Ohms centimeters);
- c_m is the membrane capacitance (Farads per centimeter);
- λ is the *membrane space constant* (cm);
- $I = I(z,t)$ is the applied current density (Amperes per centimeter).

If one wants to consider the complex dendritic morphology, analysis would require to solve the cable equation for each dendritic segment, given boundary conditions at the end of the compartments and initial conditions. Even considering a small number of dendritic sections gives rise to unwieldy mathematical solutions [197]. By proving that the *entire* complex morphology can be modelled as an equivalent cylinder [185, 188], Rall introduced a massive simplification, allowing tractable mathematical analyses of neurite models and providing a valuable insight into dendritic function.

Performing a spatial discretization for the spatial second derivative of the membrane potential (as shown in Section 3.2.4), one obtains the n equations describing the potential of the n local circuits corresponding to the n dendritic compartments. Conversely, the equations for single compartments (of length Δz) can be obtained by applying to each local equivalent circuit the *Ohm's law* and the *Kirchhoff's current law* [163, 198]. Thus the cable equation can be obtained by taking the limit $\Delta z \to 0$.

Most of the studies analysing the dendritic dynamics are based on the Rall model compartmental theory. The softwares
NEURON (http://www.neuron.yale.edu/) and
GENESIS (http://www.genesis-sim.org/GENESIS/), designed to model neuronal systems, use the principia of Rall theory to model the electric properties of nerve membrane.

In the present work, the dendritic tree of each excitatory population is modelled as a linear cable using Eq. (3.31). A schematic illustration of the compartmentalized model is illustrated in Fig. 3.6, where the *black* points schematically represent the equivalent membrane local circuits, i.e. the iso-potential dendritic segments (compartments).

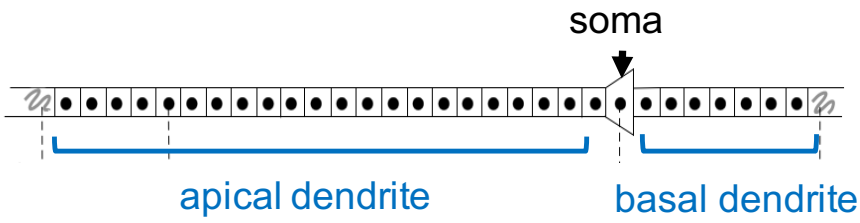

Figure 3.6: Schematic illustration of the linear cable representing the dendritic tree of one excitatory population of neurons.

Defining the *membrane time constant* τ_m (seconds) as

$$\tau_m = r_m c_m \tag{3.32}$$

and rewriting the Eq. (3.31) for the resting membrane potential $V_m = V + V_r$, one has

$$\tau_m \frac{\partial V_m}{\partial t} = \lambda^2 \frac{\partial^2 V_m}{\partial z^2} - [V_m - V_r] + r_m I. \tag{3.33}$$

Eq. (3.1) and Eq. (3.33) represent the same equation if one considers that:
- h_e is the *membrane potential* V_m;
- the sum $\sum_l \psi_{le}(h_e)\ I_{le}(\mathbf{x}, t)$ in Eq. (3.1) represents the applied current term $r_m I$ in Eq. (3.33);

- if a point-shaped neuron is considered instead of a neuron with extended dendritic tree, then the term $\lambda^2 \frac{\partial^2 V_m}{\partial z^2}$ in Eq. (3.33) can be set to zero (since $\lambda = 0$).

Therefore, with the inclusion of the second derivative of the excitatory membrane potential h_e, Eq. (3.1):

$$\tau_e \frac{\partial h_e}{\partial t}(\mathbf{x}, t) = -[h_e(\mathbf{x}, t) - h_e^r] + \sum_l \psi_{le}(h_e) I_{le}(\mathbf{x}, t)$$

is transformed into a cable equation:

$$\begin{aligned} \tau_e \frac{\partial h_e}{\partial t}(\mathbf{x}, z, t) = -[h_e(\mathbf{x}, z, t) - h_e^r] + \lambda^2 \frac{\partial^2 h_e}{\partial z^2}(\mathbf{x}, z, t) + \\ + \sum_l \psi_{le}(h_e) I_{le}(\mathbf{x}, z, t) \end{aligned} \tag{3.34}$$

where z is the distance along the dendrite and $\mathbf{x}$ is the position on the two-dimensional cortical sheet (transversal to the dendritic direction). A geometrical illustration of $\mathbf{x}$ and z directions is provided in Fig. 3.7.

In our model, we used *sealed end* boundary conditions [163], meaning that no current is flowing through both ends of the linear cable. Detailed equations for h_e in the boundary compartments are discussed in Section 3.2.6.

3.2.3 Electrotonic length constant

The electrotonic dendritic length constant λ is used to quantify the distance that a potential spreads along the dendrite via passive conduction [163]. A current injection results in a local depolarization at the injection site. This depolarization longitudinally spreads along the neural cable, depending on the electric properties of the membrane. At the injection site, the depolarization V is following the *Ohm's law*

$$V = I * r_m \tag{3.35}$$

where I is the injected current and r_m is the *membrane resistance* (the force impeding the flow of electric ionic current across the membrane). A higher membrane

resistance results in a larger difference of potential across the membrane, at the injection site. The extent of depolarization falls off with distance from the initial depolarization and can be described by the following equation:

$$V(x) = V_{max} e^{-\frac{|x|}{\lambda}} \tag{3.36}$$

where x is the spatial position ($x = 0$ at the injection point) and λ is the *length constant*. Setting for $x = \lambda$ in Eq. (3.36), one can see that λ is the distance at which $V(x)$ has dropped to 37% of its original value V_{max}. The greater the value of the length constant, the farther the potential will travel.

Considering the intrinsic resistance properties of the cable, the length constant can be defined [163] as:

$$\lambda = \sqrt{\frac{r_m}{r_a}} \tag{3.37}$$

where r_m is the *membrane resistance* and r_a is the *axial resistance* (the force impeding the current flow travelling along the dendrite). The membrane resistance r_m enhances the spread of voltage along the neural cable since it prevents losses across the membrane, while the axial resistance impedes the current flow along the cable.

Given the variety of functions performed by neurons, there is a wide diversity in their morphology, size and electrical mechanisms [6, 129]. The dendritic length constant is one of the properties that highly depends on the geometry of the neuron; it increases with both neuron size and dendritic branch radius [163, 199]. As previously explained, in the present work we are mainly interested in the activity of pyramidal neurons. The diameter of pyramidal dendrites varies from a few microns for the main apical trunk to less than half micron for the terminal branches, while the linear distance from the basal to the apical tuft ranges from 0.2 to over 1 mm [30, 134]. Typical estimates for the length constant range from 0.1 to 1 mm [200–204]. The description of electrotonic structure of a neuronal tree [205], approximated as an equivalent cylinder, usually refers to the *electrotonic length* L of the neurite [206], defined as its physical length l (mm) divided by its electrotonic space constant, $L = l/\lambda$. In [207, 208], Rall provided many formulas to theoretically estimate the electrotonic length of a neurite, finding that it ranges approximately

from 1 to 2. Over the years, Rall's formulas have been applied in several studies and in most cases the estimate for L is about 1 [205, 209–213].

In the present work we have considered excitatory populations extending their "dendritic trees" through the entire cortex. Although pyramidal neurons have dendritic trees of about 1 mm [30, 134], the modelling scheme adopted here incorporates entire microcolumns of neurons with their cell bodies in different layers [6] and trees extending over the entire cortical depth (4 mm). The dendritic trees of *one population* are modelled as *one linear cable* and the spatio-temporal evolution of the mean membrane potential of the entire microcolumn is described. For our analysis, we mainly considered electrotonic distances in the physiological range, varying from 0.1 to 1 mm. Ideally, if one wants to enable a stimulus to spread over the entire "dendritic length", larger electrotonic constants (up to 4 mm, corresponding to electrotonic length $L = 1$) could be considered. To take this into account, we extended the explored range for the electrotonic length constant λ up to 2 mm.

3.2.4 Model discretization

To implement the 3D system in MatLab, a discretization on both $\mathbf{x}$ and z directions is necessary. We maintained the same scheme for the spatial discretization along the cortical sheet implemented in Section 3.1.1, transversal with respect to the dendritic extension. This means, that the neuronal populations are distributed on a square spatial grid, at a relative distance $\Delta x = 1$ mm. The excitatory subpopulations extend their dendritic trees through the depth of the cortex, along the z direction, with soma located in layer V in correspondence of the two-dimensional grid nodes, while inhibitory populations are modelled as points. A geometrical illustration of the implemeted 3D-model, extending the 2D geometry is provided in Fig. 3.7. As in [36], the boundaries of the grid are then connected to form a toroid.

The cable Eq. (3.34) for a **specific** point $\mathbf{x}$ on the two-dimensional can be written as[8]:

[8]The synaptic factor f_{syn} will be introduced in the next section

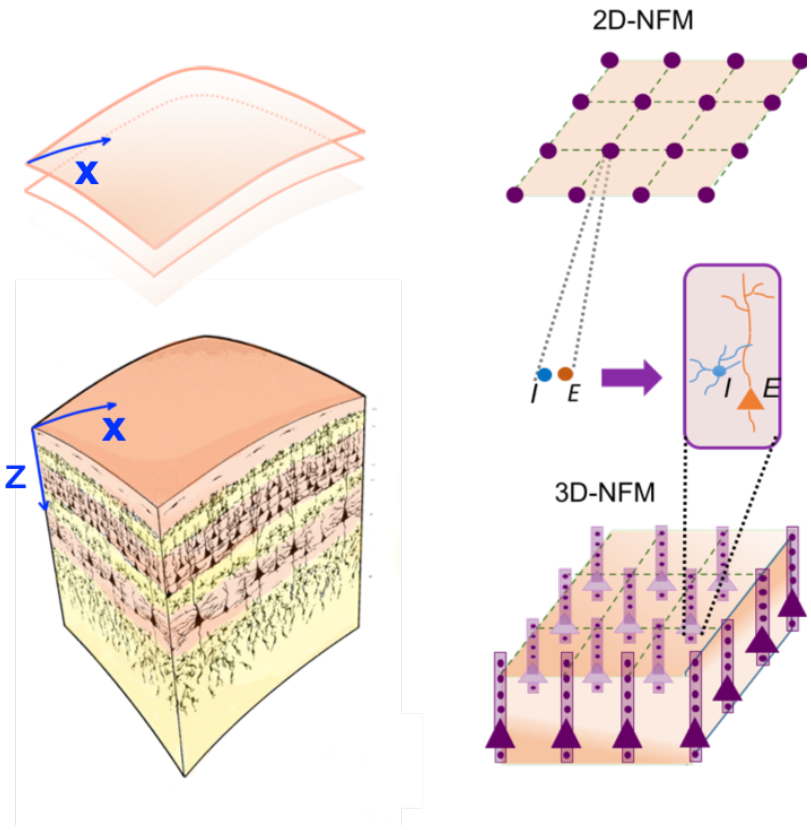

Figure 3.7: Transforming NFMs from 2D to 3D scheme. **Left**. Morphology of the biological systems represented. **Right**. Illustration of the computational models implementing the morphology in A. The purple points and their corresponding 3D extension represent one neuronal macrocolumn with its excitatory (E) and inhibitory (I) subpopulations.

$$\tau\frac{\partial h_e}{\partial t}(\mathbf{x},z,t) = -[h_e(x,t) - h_e^r] + \lambda^2\frac{\partial^2 h_e}{\partial z^2}(\mathbf{x},z,t) + \quad \sum_l \psi_{le}(h_e)I_{le}(z,t) \tag{3.38}$$

Dividing the dendrite into n subsections of length $\Delta z = l/n$, where l is the length of the cable, and denoting with h_j, $(j = 1, ..., n)$ the *mean membrane potential* in the subsection j, the *mean membrane potential* along the dendrite can be described by the n-dimensional vector $\mathbf{h} = [h_1, ...h_n]$. Using the second order *finite difference approximation*, the diffusion term of Eq. (3.38), for $j = 1, ...n$, can be written as

$$\frac{\partial^2 h_j}{\partial z^2} \simeq \frac{h_{j-1} - 2h_j + h_{j+1}}{\Delta z^2}. \tag{3.39}$$

Thus, Eq. (3.38), for the particular subsection j, becomes:

$$\tau\frac{\partial h_j}{\partial t} = -[h_j - h^r] + \lambda^2\frac{h_{j-1} - 2h_j + h_{j+1}}{\Delta z^2} + \sum_l \psi_l(h_j)I_l(z,t) \tag{3.40}$$

where for simplicity $h_{e,j}(z,t)$ has been denoted with h_j. Let

$$J(h_j,z) = \begin{cases} \sum_l \psi_l(h_j)I_l(z,t) & \text{in dendritic subsections with synaptic inputs} \\ 0 & \text{elsewhere} \end{cases} \tag{3.41}$$

then Eq. (3.40) becomes:

$$\begin{aligned} \frac{\partial h_j}{\partial t} &= -\left(\frac{1}{\tau} + \frac{2\lambda^2}{\tau dz^2}\right)h_j + \frac{\lambda^2}{\tau dz^2}h_{j-1} + \frac{\lambda^2}{\tau dz^2}h_{j+1} + \frac{1}{\tau}J(h_j,z) + \frac{h^r}{\tau} \\ &= K\,h_{j-1} - B\,h_j + K\,h_{j+1} + \frac{1}{\tau}J(h_j,z) + c_e \end{aligned} \tag{3.42}$$

where

$$K = \frac{\lambda^2}{\tau\Delta z^2}, \quad B = \left(\frac{1}{\tau} + 2K\right) \quad \text{and} \quad c_e = \frac{h^r}{\tau} \tag{3.43}$$

As mentioned above, through Eq. (3.41), it is possible to specify the position of the synaptic input along the dendrite.

3.2.5 Synaptic factor in the discretised dendrite

As it will be discussed in Chapter 4, naively extending the 2D model to a 3D geometry leads to a loss of signal along the dendrite. As a result, the somatic response to single pulse input (discussed in Section 6.2) is lower in the 3D geometry than in the 2D system. Besides this, such systems are characterized by a loss of alpha rhythmicity. We refer to these systems as "less excitable". Here, the notion of "excitability" is used to indicate the system sensitivity to pulse and synaptic inputs and the corresponding suitability to produce alpha rhythms (discussed in Section 6.2).

To restore the excitability, a synaptic factor strengthening the effect of the synaptic currents on the mean excitatory membrane potential has been introduced:

$$I_{syn} = \frac{1}{\tau_e} f_{syn} \sum_l \psi_{le}(h_j) I_{le}(z,t) \tag{3.44}$$

As seen in the previous section, the synaptic input (in its original formulation without the synaptic factor)

$$I_{syn} = \frac{1}{\tau_e} \sum_l \psi_{le}(h_j) I_{le}(z,t) \tag{3.45}$$

can be applied in one or more subsections of the compartmentalized linear cable.

While in the 2D model the synaptic input is applied to the entire point-shaped neuronal population (without dendritic arborization), in the 3D geometry the input is located in one specific region and then spreads along the z dendritic coordinate. As a result, the h_e response elicited in the 3D system is expected to be lower than in the 2D geometry since the current is distributed over the cable. A schematic illustration is provided in Fig. 3.8 A and B. This behaviour will be confirmed in Section 6.2, where the response of both the 2D and 3D system to a single pulse synaptic input will be analysed, for systems with different number of dendritic compartments.

Another important consideration concerns the relationship between the factor f_{syn}, the location of the synaptic factor along the dendrite, and the extension of the dendritic region where the current is applied. This is particularly important when one compares the dynamics of systems with different number of compartments. As

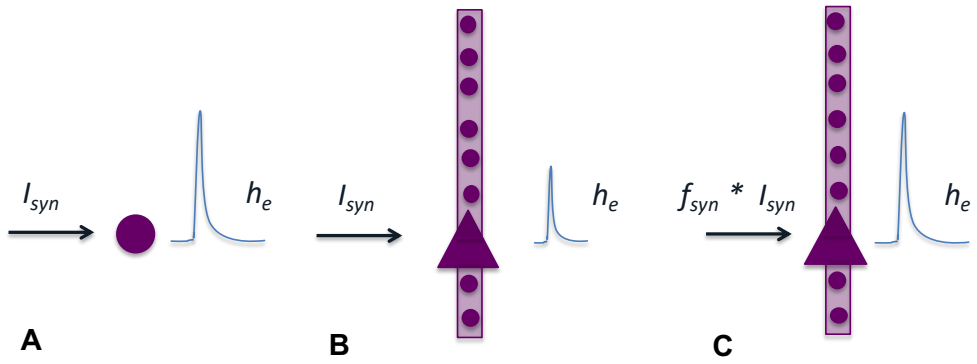

Figure 3.8: Schematic illustration of the h_e response to a single pulse synaptic input in the 2D (**A**) and 3D (**B**) geometries. A factor f_{syn} strengthening the synaptic current is introduced in the 3D system to elicit a similar response at the injection site (**C**).

an illustrative example, we consider a case where the synaptic input is applied in the distal region (apical dendrite) at a distance z_s from the soma, in a system with n compartments. A schematic illustration of the local and somatic responses h^e_j and h^e_{soma} to an input in the j-th compartment is provided in Fig. 3.9 A. Considering a system with $3n$ subsections (Fig. 3.9 B), and locating the synaptic input in the $3j$-th compartment at exactly the same distance z_s from the soma, one can observe a reduced h_e response both at the injection site and at the soma. Moving from configuration A to B, the size of the compartment where the current is injected is reduced by a factor of 3. This means that in the same time interval Δt, a different amount of charge is actually flowing into the systems respectively through the compartments j and $3j$. More in detail, in the first case a larger charge Q will accumulate in the compartment j having length l/n and will propagate along the cable in both directions. In the second case, compartment $3j$ having length $l/3n$, will host a charge $q < Q$, that will spread in both directions evoking at the soma a lower response. To elicit in B, the same response of A, the current I_{syn} needs to be applied in 3 compartments ($3j-1$,$3j$,$3j+1$) as shown in figure C, *or* a factor $f_{syn} = 3$ needs to be introduced to strengthen the synaptic input applied in compartment j (D). All these qualitative considerations will be numerically confirmed in Section

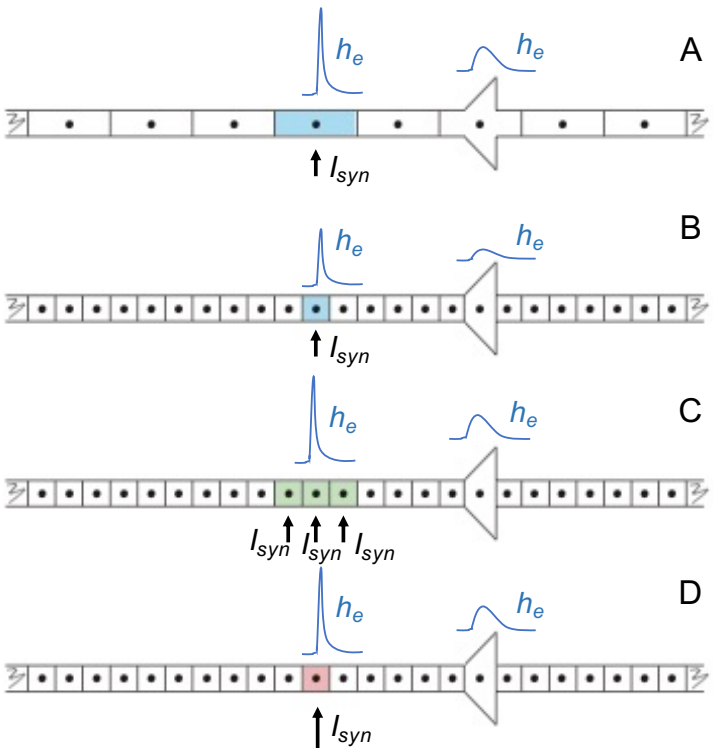

Figure 3.9: Schematic illustration of the h_e local and somatic responses to a single pulse synaptic input in 3D (**B**) geometries, with different numbers of compartments (A vs B); for synaptic inputs located in one or more compartments (A vs C); with and without synaptic factor f_{syn} (A vs D).

6.2.

3.2.6 Boundary conditions

Sealed end boundary conditions have been used for the dendritic cable. This means that no current flows out of either ends of the cable. Thus, for the compartments at both the ends of the cable, we have:

$$\begin{aligned} \frac{\partial h_{1,l}}{\partial z} &= 0 \\ \frac{\partial h_{n,r}}{\partial z} &= 0 \end{aligned} \tag{3.46}$$

where the subscripts l and r indicate that there is no current flowing through to the left side of compartment 1 and through the right side of compartment n (see illustration 3.6). Equation (3.42) for h_e in the j-th dendritic subsection, is computed through the potential of the adjacent subsections $j-1$ and $j+1$. Let h_1 and h_n be the potentials in the first and last compartment, respectively. Since there is no current flowing (i.e. no change of the potential h along z), one has

$$\begin{aligned} h_1 &= h_2 \\ h_n &= h_{n-1} \end{aligned} \tag{3.47}$$

Equations for the boundary compartments then become [9]:

$$\begin{aligned} \frac{\partial h_1^e}{\partial t} &= (K_e + B_e)h_1^e + K_e h_2^e + c_e \\ \frac{\partial h_n^e}{\partial t} &= K_e h_{n-1}^e + (B_e + K_e)h_n^e + c_e \end{aligned} \tag{3.48}$$

3.2.7 Dendritic system in matrix form

Equations (3.38), for the entire dendrite, can be written as

$$\frac{\partial \mathbf{h}}{\partial t} = \mathbf{A} * \widehat{\mathbf{h}} + \mathbf{c} + \mathbf{J} \tag{3.49}$$

where $\widehat{\mathbf{h}}$ is the $n+2$-dimensional vector $\widehat{\mathbf{h}} = [h_1, h_1, h_2, ...h_{n-1}, h_n, h_n]^T$, $\mathbf{c}$ and $\mathbf{J}$ are the n-dimensional vectors

$$\mathbf{c} = \left[\frac{h^r}{\tau_e}, ..., \frac{h^r}{\tau_e}\right],$$

$$\mathbf{J} = \left[\frac{1}{\tau_e}J(h_1, z), ..., \frac{1}{\tau_e}J(h_n, z)\right], \tag{3.50}$$

[9] Here, the equations for the boundaries are written considering that synaptic inputs are not applied at the ends (i.e. $J(h, z) = 0$ at both the ends).

and $\mathbf{A}$ is the $(n-2)$xn matrix

$$\mathbf{A} = \begin{bmatrix} K & B & K & 0 & \dots & \dots & 0 \\ 0 & K & B & K & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & K & B & K & 0 \\ 0 & \dots & \dots & \dots 0 & K & B & K \end{bmatrix}.$$

3.2.8 Complete 3D system

In this section, the equations describing the full 3D model are reported, taking into account the considerations and assumptions illustrated in previous sections.

The mathematical formulation of three-dimensional model is:

$$\begin{aligned} \tau_e \frac{\partial h_e}{\partial t}(\mathbf{x}, z, t) &= -[h_k(\mathbf{x}, z, t) - h_e^r] + \lambda^2 \frac{\partial^2 h_e}{\partial z^2}(\mathbf{x}, z, t) + \\ &+ f_{syn} \sum_l \psi_{le}(h_e) I_{le}(\mathbf{x}, z, t) \\ \tau_i \frac{\partial h_i}{\partial t}(\mathbf{x}, t) &= -[h_i(\mathbf{x}, t) - h_i^r] + \sum_l \psi_{li}(h_i) I_{li}(\mathbf{x}, t) \\ \frac{\partial I_{lk}}{\partial t} &= \widetilde{I}_{lk} \\ \frac{\partial}{\partial t}\widetilde{I}_{lk}(t) &= -2\gamma_{lk}\widetilde{I}_{lk} - \gamma_{lk}^2 I_{lk} + N_{lk}^{\beta} S_l(h_l) + \Phi_{lk} + p_{lk}(t) \\ \frac{\partial \Phi_{ek}}{\partial t} &= \widetilde{\Phi}_{ek} \\ \frac{\partial}{\partial t}\widetilde{\Phi}_{ek}(\mathbf{x}, t) &= -v\Lambda_{ek}\left\{ 2\widetilde{\Phi}_{ek} + v\Lambda_{ek}\Phi_{ek}(t) - v\Lambda_{ek} N_{ek}^{\alpha} S_e(h_e) \right\} + \\ &+ \frac{3}{2} v^2 \nabla^2 \Phi_{ek}(\mathbf{x}, t) \end{aligned} \tag{3.51}$$

Performing a spatial discretization and using the 5-point stencil approximation for the cortical connections, the spatio-temporal evolution of the (n+13)-dimensional state variable

$$\mathbf{S} = (\mathbf{h_e}, h_i, I_{ee}, \widetilde{I}_{ee}, I_{ei}, \widetilde{I}_{ei}, I_{ie}, \widetilde{I}_{ie}, I_{ii}, \widetilde{I}_{ii}, \Phi_{ee}, \widetilde{\Phi}_{ee}, \Phi_{ei}, \widetilde{\Phi}_{ei})^T \tag{3.52}$$

representing the activity of a single microcolumn, is described by the following system of discretized equations:

$$
\begin{aligned}
\frac{\partial h_1^e}{\partial t} &= (K_e + B_e)h_1^e + K_e h_2^e + c_e \\
\frac{\partial h_2^e}{\partial t} &= K_e h_1^e + B_e h_2^e + K_e h_3^e + c_e \\
\frac{\partial h_3^e}{\partial t} &= K_e h_2^e + B_e h_3^e + K_e h_4^e + c_e \\
\ldots &\qquad \ldots \\
\frac{\partial h_j^e}{\partial t} &= K_e h_{j-1}^e + B_e h_j^e + K_e h_{j+1}^e + c_e + \frac{1}{\tau_e}\, f_{syn}\ \left[\psi_{ee}(h_j^e) I_{ee} + \psi_{ie}(h_j^e) I_{ie}\right] \\
\ldots &\qquad \ldots \\
\frac{\partial h_{n-1}^e}{\partial t} &= K_e h_{n-2}^e + B_e h_{n-1}^e + K_e h_n^e + c_e \\
\frac{\partial h_n^e}{\partial t} &= K_e h_{n-1}^e + (B_e + K_e) h_n^e + c_e
\end{aligned}
\tag{3.53}
$$

for the membrane potential of the excitatory dendrite, with synaptic input located at compartment j;

$$
\tau_k \frac{\partial h_k}{\partial t}(\mathbf{x}, t) = -[h_k(\mathbf{x}, t) - h_k^r] + \sum_l \psi_{lk}(h_k) I_{lk}(\mathbf{x}, t) \tag{3.54}
$$

for the membrane potential of the inhibitory population;

$$
\begin{aligned}
\frac{\partial}{\partial t} I_{ee} &= \widetilde{I}_{ee}(t) \\
\frac{\partial}{\partial t} \widetilde{I}_{ee}(t) &= -2\gamma_{ee}\widetilde{I}_{ee} - \gamma_{ee}^2 I_{ee} + e\gamma_{ee}\Gamma_{ee}\left(N_{ee}^{\beta} S_e(h_e) + \Phi_{ee} + p_{ee}(t)\right) \\
\frac{\partial}{\partial t} I_{ei} &= \widetilde{I}_{ei}(t) \\
\frac{\partial}{\partial t} \widetilde{I}_{ei}(t) &= -2\gamma_{ei}\widetilde{I}_{ei} - \gamma_{ei}^2 I_{ei} + e\gamma_{ei}\Gamma_{ei}\left(N_{ei}^{\beta} S_e(h_e) + \Phi_{ei} + p_{ei}(t)\right) \\
\frac{\partial}{\partial t} I_{ie} &= \widetilde{I}_{ie}(t) \\
\frac{\partial}{\partial t} \widetilde{I}_{ie}(t) &= -2\gamma_{ie}\widetilde{I}_{ie} - \gamma_{ie}^2 I_{ie} + e\gamma_{ie}\Gamma_{ie}\left(N_{ie}^{\beta} S_i(h_i) + \Phi_{ie}\right) \\
\frac{\partial}{\partial t} I_{ii} &= \widetilde{I}_{ii}(t) \\
\frac{\partial}{\partial t} \widetilde{I}_{ii}(t) &= -2\gamma_{ii}\widetilde{I}_{ii} - \gamma_{ii}^2 I_{ii} + e\gamma_{ii}\Gamma_{ii}\left(N_{ii}^{\beta} S_i(h_i) + \Phi_{ii}\right)
\end{aligned}
\tag{3.55}
$$

for the synaptic inputs,

$$
\begin{aligned}
\frac{\partial}{\partial t} \Phi_{ee} &= \widetilde{\Phi}_{ee} \\
\frac{\partial}{\partial t} \widetilde{\Phi}_{ee}(\mathbf{x}, t) &= -v\Lambda_{ee}\left\{2\widetilde{\Phi}_{ee} + v\Lambda_{ee}\Phi_{ee}(t) - v\Lambda_{ee}N_{ee}^{\alpha} S_e(h_e)\right\} + \\
&\quad + \frac{3}{2}\frac{v^2}{\Delta x^2}\left[\Phi_{ee}^{u}(\mathbf{x_u}, t) + \Phi_{ee}^{r}(\mathbf{x_r}, \mathbf{t}) + \Phi_{ee}^{d}(\mathbf{x_d}, t) + \Phi_{ee}^{l}(\mathbf{x_l}, t) - 4\Phi_{ee}(\mathbf{x}, t)\right] \\
\frac{\partial}{\partial t} \Phi_{ei} &= \widetilde{\Phi}_{ei} \\
\frac{\partial}{\partial t} \widetilde{\Phi}_{ei}(\mathbf{x}, t) &= -v\Lambda_{ei}\left\{2\widetilde{\Phi}_{ei} + v\Lambda_{ei}\Phi_{ei}(t) - v\Lambda_{ek}N_{ei}^{\alpha} S_e(h_e)\right\} + \\
&\quad + \frac{3}{2}\frac{v^2}{\Delta x^2}\left[\Phi_{ei}^{u}(\mathbf{x_u}, t) + \Phi_{ei}^{r}(\mathbf{x_r}, \mathbf{t}) + \Phi_{ei}^{d}(\mathbf{x_d}, t) + \Phi_{ei}^{l}(\mathbf{x_l}, t) - 4\Phi_{ei}(\mathbf{x}, t)\right]
\end{aligned}
\tag{3.56}
$$

for the cortical connections.

The cortical input $p_{ee}(t)$ has been modelled as white noise. The functions S and Ψ appearing in the 3D system above have been described in Section 3.1, and are

repeated here for completeness:

$$\psi_{lk}(h_k) = \frac{h_{lk}^{eq} - h_k(\mathbf{x}, t)}{\left|h_{lk}^{eq} - h_k^r\right|} \tag{3.57}$$

$$S_k(h_k) = S_k^{max} \Big/ \left\{1 + (1 - r_{abs} S_k^{max}) \times exp\left[-\sqrt{2}\frac{h_k(\mathbf{x}, t) - \mu_k}{\sigma_k}\right]\right\} \tag{3.58}$$

Note that the term in square brackets (divided by Δx^2), in Eq. (3.56), is the 5 points stencil approximation to compute the Laplacian $\nabla^2 \Phi_{ek}$. This approximation makes the points of the grid coupled. The (n+13) equations of the system (3.53)-(3.56), for one particular point of the grid, describing the dynamics of the state variables (3.52), contains 8 more variables for the cortical connections Φ_{ek}^s of the nearest neighbours points, where $s = u$ (*upper*), r (*right*), d (*down*), l (*left*).

Using a matrix description, the system (3.53)-(3.56) can be written as:

$$\frac{\partial \mathbf{S}}{\partial t} = \mathbf{F}(\mathbf{S}) + \mathbf{P} \tag{3.59}$$

where $\mathbf{P}$ is the vector for the extra-cortical inputs p_{lk}.
The description of the entire two-dimensional grid is provided by $N = p \cdot p$ systems, analogous to the system (3.53)-(3.56), containing therefore $C = N(n+13)$ equations. Denoting with $\mathbf{S_c}$ the C-dimensional vector representing the state variable for the entire system, and with $\mathbf{F_s}$ the corresponding ODEs, we obtain the full system of C equations:

$$\frac{\partial \mathbf{S_c}}{\partial t} = \mathbf{F_c}(\mathbf{S_c}) + \mathbf{P_c} \tag{3.60}$$

where the vector $\mathbf{P_c}$ is the vector for all the extra-cortical inputs p_{lk}. Note that the non zero components of $\mathbf{P}$ are:

$$\left\{\mathbf{P_c}_{,n+3}, \mathbf{P_c}_{,(n+13)+(n+3)}, ..., \mathbf{P_c}_{,(N-1)(n+13)+(n+3)}\right\} \tag{3.61}$$

The functions implementing the 2D and 3D systems with a single or multiple microcolumns are reported in the Appendix.

3.2.9 Stable points

In order to numerically detect the steady state solutions for both 2D and 3D systems, we run a test simulation (without noise) until all variables reach their steady state. At a first stage, we run the test simulation for a fixed amount of time. However, this was not successfully leading to correct results, since the transient time needed for the convergence highly depends on the parameters of the simulation. For this reason, the duration has been set automatically during the test simulation, depending on the convergence rate of the membrane potential to its steady value. Initial values for excitatory and inhibitory membrane potentials are set between -55 mV and -65 mV. Other variables are initially set to zero. After running the system for 2 seconds, we consider the values the variables are converging to and set these as new initial conditions. This procedure is repeated until the convergence rate is less than 0.01%. More details are illustrated in the code reported in the Appendix.

3.2.10 Non-Linear Power Spectra Densities (PSDs)

The system (3.53)-(3.56) has been numerically solved using *forward Euler* method [178]. Simulations are run for 20 to 400 s. From the h_e time series, the power spectrum of frequencies has been calculated using the Welch's estimator [214]. For this purpose, the Matlab function *pwelch* (with window size = 1000, overlap of 50% of window length, number of sampling = 1000, sampling frequency = 500, 'onesided' option) has been employed. An equivalent function to calculate the spectrum has been also implemented from scratch (details are provided in the Appendix, script *win_spec_full_grid*). The frequency range is 0-250 Hz. For systems modelling more than one microcolumn, the non-linear spectrum is calculated considering the signals of the N microcolumns included in the model, i.e. the time series of the somatic membrane potential of all the N populations (shown in Appendix A, script *win_spec_full_grid*).

3.2.11 Linear PSD prediction

Liley and Bojak [36] found several parameter sets yielding power spectra similar in shape to the ones encountered in humans. In all the analysed cases, they found that a linearisation of the system was possible. The enormous advantage of linearisation consists in the simplification of mathematical analysis and in a massive reduction of the computational time. A method has been elaborated by the two scientists to predict the spectrum of frequencies from the Jacobian matrix of the linearised two-dimensional system evaluated at a singular point. In the present work, this approach has been extended to the three-dimensional system (3.60).

Defining the perturbation $\mathbf{s}(\mathbf{x}, t) = \mathbf{S}(\mathbf{x}, t) - \mathbf{S}^*$, where $\mathbf{S}^*$ is the singular point, and expanding $\mathbf{F}$ around $\mathbf{S}$ to first order in $\mathbf{s}$, one obtains[10]:

$$\begin{aligned} \frac{\partial \mathbf{S}}{\partial t} &= \frac{\partial(\mathbf{S}^* + \mathbf{s})}{\partial t} = \frac{\partial \mathbf{s}}{\partial t} \\ &= \mathbf{F}(\mathbf{S}^*) + \frac{\partial \mathbf{F}}{\partial \mathbf{S}}(\mathbf{S} - \mathbf{S}^*) + O[(\mathbf{S} - \mathbf{S}^*)^2] + \mathbf{P} \\ &= 0 + \mathbf{J}\,\mathbf{s} + O(\mathbf{s^2}) + \mathbf{P} \\ &\simeq \mathbf{J}\,s + \mathbf{P} \end{aligned} \tag{3.62}$$

where $\mathbf{J}$ is the Jacobian matrix of the function $\mathbf{F}(\mathbf{S})$, evaluated at $\mathbf{S}^*$:

$$\mathbf{J} \equiv \left.\frac{\partial \mathbf{F}(\mathbf{S})}{\partial \mathbf{S}}\right|_{S=S^*} \tag{3.63}$$

In the last step we considered that $\mathbf{F}(\mathbf{S}^*) = 0$, since S^* is a singular point.

The Jacobian matrix $\mathbf{J}$, having dimension $[N(n+13)] \times [N(n+13)]$, is almost a block matrix in the individual populations. Each block has dimension $(n+13) \times (n+13)$. Other 8 non-zero elements occur for each macrocolumn, corresponding to the derivatives of Eqs. (3.56) for $\widetilde{\Phi}_{ek}$ with respect to the cortical connections Φ_{ek} of

[10] Please, note that the subscript c is omitted, but the full system (3.60) is considered here.

the nearest neighbour points

$$\begin{aligned}
\frac{\partial}{\partial \Phi_{ek}^{l}} \frac{\partial \Phi_{ek}}{\partial t} &= \frac{3}{2} \frac{v^2}{\Delta x^2} \\
\frac{\partial}{\partial \Phi_{ek}^{d}} \frac{\partial \Phi_{ek}}{\partial t} &= \frac{3}{2} \frac{v^2}{\Delta x^2} \\
\frac{\partial}{\partial \Phi_{ek}^{r}} \frac{\partial \Phi_{ek}}{\partial t} &= \frac{3}{2} \frac{v^2}{\Delta x^2} \\
\frac{\partial}{\partial \Phi_{ek}^{u}} \frac{\partial \Phi_{ek}}{\partial t} &= \frac{3}{2} \frac{v^2}{\Delta x^2}
\end{aligned} \tag{3.64}$$

for $k \in \{e, i\}$.

In order for Eq. (3.62) to be a good approximation of Eq. (3.60), it should be stable. In other words, if we start our simulation from a point that is close to the equilibrium point $\mathbf{S}^*$, i.e. if we consider a small disturbance $\mathbf{s}$, the system should quickly return to the singular point $\mathbf{S}^*$. To guarantee the stability we also need the noise term $\mathbf{P}$ to be small. To ensure the stability of our approximation we need that all the eigenvalues λ_n of the system have real part Re $\lambda_n < 0$. The Jacobian matrix is not symmetric, so we need to decompose it with left $\mathbf{L}$ and right $\mathbf{R}$ eigenvectors matrices:

$$\begin{aligned}
\mathbf{LR} &= \mathbf{R}\ \mathrm{diag}(\lambda_n) \\
\mathbf{LJ} &= \mathrm{diag}(\lambda_n)\ \mathbf{L} \\
\mathbf{LR} &= \mathrm{diag}(e_n)
\end{aligned} \tag{3.65}$$

Applying the Fourier transform with respect to space to Eq. (3.62), one obtains

$$\frac{\partial \mathbf{s}(\mathbf{k}, t)}{\partial t} \simeq \mathcal{J}\, \mathbf{s}(\mathbf{k}, t) + \mathbf{P}(\mathbf{k}, t) \tag{3.66}$$

where $\mathcal{J}$ is the Jacobian matrix of the function $\mathbf{F}(\mathbf{S})$, evaluated at $\mathbf{S}^*$, with the formal replacement $\nabla^2 \to -\mathbf{k^2}$

$$\mathcal{J} \equiv \left. \frac{\partial \mathbf{F}(\mathbf{S})}{\partial \mathbf{S}} \right|_{S=S^*}^{\nabla^2 \to -\mathbf{k^2}} \tag{3.67}$$

Performing this Fourier transform, the coupling terms in square brackets of Eqs. (3.56) are substituted with the term $-\mathbf{k^2}\Phi_{ek}$, and Eqs. (3.64) become:

$$\begin{aligned}
\frac{\partial}{\partial \Phi^l_{ek}}\frac{\partial \Phi_{ek}}{\partial t} &= 0 \\
\frac{\partial}{\partial \Phi^d_{ek}}\frac{\partial \Phi_{ek}}{\partial t} &= 0 \\
\frac{\partial}{\partial \Phi^r_{ek}}\frac{\partial \Phi_{ek}}{\partial t} &= 0 \\
\frac{\partial}{\partial \Phi^u_{ek}}\frac{\partial \Phi_{ek}}{\partial t} &= 0
\end{aligned} \tag{3.68}$$

Each macrocolumn is then formally decoupled from other macrocolumns and the Jacobian matrix $\mathcal{J}$ is a pure block matrix in the single populations. The decomposition in right and left eigenvectors matrices for the matrix $\mathcal{J}$ is analogous to Eqs. (3.65):

$$\begin{aligned}
\mathcal{L}\mathcal{R} &= \mathcal{R}\ \mathrm{diag}(\lambda_n) \\
\mathcal{L}\mathcal{J} &= \mathrm{diag}(\lambda_n)\ \mathcal{L} \\
\mathcal{L}\mathcal{R} &= \mathrm{diag}(e_n)
\end{aligned} \tag{3.69}$$

The sign of the eigenvalues λ_n depends on the parameter set and on the value of $\mathbf{k^2}$ in Eq. (3.67).

Applying the temporal Fourier transform to Eq. (3.62), one obtains

$$\begin{aligned}
i\omega\mathbf{s} &= \mathbf{J} * \mathbf{s} + \mathbf{P} \\
\mathrm{diag}(i\omega - \lambda_n) * \mathcal{L} * \mathbf{s} &= \mathcal{L} * \mathbf{P} \\
\mathbf{s} &= \mathcal{R}\,\mathrm{diag}\left[\frac{1}{e_n(i\omega - \lambda_n)}\right] * \mathcal{L} * \mathbf{P} \equiv \mathcal{P} * \mathbf{P}
\end{aligned} \tag{3.70}$$

Let $h_{e,s}$ be the somatic mean membrane potential of the population with index s of the three-dimensional grid. In order to compute the power spectrum of frequencies we need to compute

$$\sum_{s=1}^{N} |h_{e,s}(\mathbf{x},\omega)|^2 = \sum_{s=1}^{N} \left| \sum_{j=1}^{N} [\mathcal{P}_{s,n+3}(\mathbf{x},\omega)]_j \, [P_{n+3}(\mathbf{x},\omega)]_j \right|^2 \tag{3.71}$$

where the $[P_{n+3}]_j$ corresponds to the non zero component $(j-1)*(n+13)+(n+3)$ of the vector $\mathbf{P_c}$ (3.61), and likewise the index j in $[\mathcal{P}_{s,n+3}(\mathbf{x},\omega)]_j$ addresses the corresponding column elements of the row s.

If one considers small grids of 4×4 points (i.e. 4×4 mm), it can be assumed the "same" cortical noise input $p_{ee}(t)$ (i.e. white noise with the same standard deviation) for all the points of the grid, since it represents the same cortico-cortical fiber connection. Furthermore, Fourier transforming, the noise terms become the same constant. Moreover, since all the points are the same, instead of using Eq. (3.71), the PSD can be calculated from one single macrocolumn j:

$$\sum_{s=1}^{N} |h_{e,s}(\mathbf{x},\omega)|^2 = N \cdot \left| \sum_{j=1}^{N} [\mathcal{P}_{s,n+3}(\mathbf{x},\omega)]_j \, [P_{n+3}(\mathbf{x},\omega)]_j \right|^2 \tag{3.72}$$

that, in the case of a one point grid simply becomes:

$$|h_{e,s}(\mathbf{x},\omega)|^2 = |\mathcal{P}_{s,n+3}(\mathbf{x},\omega) \, P_{n+3}(\mathbf{x},\omega)|^2 \tag{3.73}$$

Operating also a spatial Fourier transform, the PSD can be calculated using the formula:

$$\sum_{s=1}^{N} |h_{e,s}(\mathbf{k},\omega)|^2 = \sum_{s=1}^{N} \left| \sum_{j=1}^{N} [\mathcal{P}_{s,n+3}(\mathbf{k},\omega)]_j \, [P_{n+3}(\mathbf{k},\omega)]_j \right|^2 \tag{3.74}$$

or, since all the points of the grid are equivalent

$$\sum_{s=1}^{N} |h_{e,s}(\mathbf{k},\omega)|^2 = N \cdot \left| \sum_{j=1}^{N} [\mathcal{P}_{s,n+3}(\mathbf{k},\omega)]_j \, [P_{n+3}(\mathbf{k},\omega)]_j \right|^2 \tag{3.75}$$

which, in the case of a single microcolumn system, simply becomes:

$$|h_{e,s}(\mathbf{k},\omega)|^2 = |\mathcal{P}_{s,n+3}(\mathbf{k},\omega) \, \mathbf{P}_{n+3}(\mathbf{k},\omega)|^2. \tag{3.76}$$

Chapter 4

3D model spatio-temporal activity

4.1 Introduction

In this Chapter, the activity of two-dimensional (2D) [36] and three-dimensional (3D) NFM with the dendritic dimension is studied in terms of power spectral densities (PSDs) of the excitatory membrane potential and spatio-temporal patterns arising during rest activity. The portion of cortical sheet observed varies from 1 mm^2 to 256 mm^2 and includes one or multiple microcolumns distributed on a square grid (1 to 16x16 points). Neuronal excitatory populations extend their dendrites through the 4 mm cortical depth, while inhibitory populations are modeled as points. In this study, the dendrites are spatially discretized in 30 dendritic subsection.

While alpha-band activity appears in single and multiple microcolumns 2D systems, the model including the extended dendrite dramatically loses this rhythmicity if one keeps all the original parameters of the 2D geometry unchanged. To deal with this, the impact of a range of parameters on the alphoid dynamics is evaluated for single macrocolumn systems, providing evidence for the possible occurrence of alpha rhythm in 3D NF models.

Details about the main procedures implemented to numerically solve and analyse the ODE system modelling the different geometries are provided both in this Chapter and in the Appendix.

4.2 2D NFM spectra

The ODE system (3.10) formulated in [36] has been implemented in MatLab and solved using the Euler method[1].

Figure 4.1 shows 6s time series of the excitatory membrane potential h_e, for a system including a single microcolumn. This corresponds to 14 ODEs of the form (3.10) where the cortical inputs Φ_{ek} of the neighbour populations are set to zero. The corresponding non-linear (*blue*) and linear (*green*) PSDs are shown in Fig. 4.3. The non-linear spectrum is computed from 200s time series, while the linear one is calculated using Eq. (3.73). Figures 4.2 and 4.4 show 6s time series and the corresponding PSD for a system of 16 microcolumns, distributed on a square grid. This configuration is mathematically represented by a 4x4 ODE system

$$\frac{\partial \mathbf{S}}{\partial t} = \mathbf{F}(\mathbf{S}) + \mathbf{P}$$

where the state variable S is obtained by a concatenation of 16 14-dimensional state variables corresponding to single macrocolumns and P is the vector for the extra-cortical input. The neural populations are spatially coupled through the Laplacian of the cortical connections, calculated through the 5-points stencil approximation, as shown in Eqs. (3.10). The time series represented in Fig. 4.2 corresponds to one single microcolumn located in a central region of the grid, while the non-linear spectrum is computed taking into account the time series of all the microcolumns[2]. Furthermore, the linear spectrum is calculated using Eq. (3.71). The procedure used to calculate the linear spectrum for both one and multiple microcolumn systems use the MatLab function *jacobian*. Analogous procedures are defined for the 3D system and reported in the Appendix.

[1]The MatLab procedures used for the 2D system are analogous to the ones used for the 3D system, which are reported in the Appendix.

[2]The procedure applied to calculate the non-linear spectrum for the entire grid is analogous to the procedure *non_linear_spectrum_full_grid* used for the 3D system and illustrated in the Appendix.

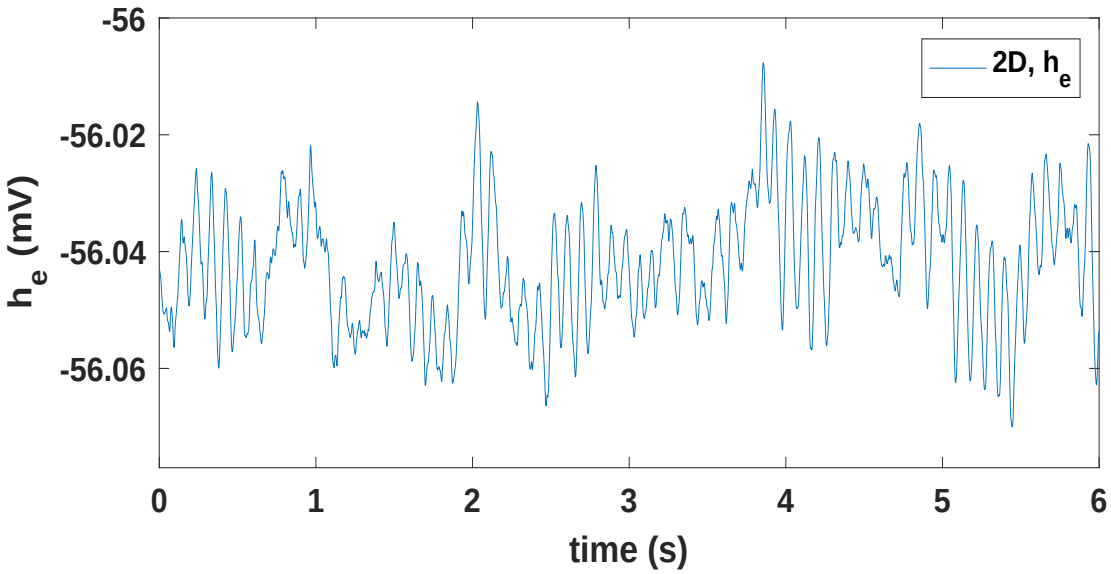

Figure 4.1: h_e time series for the 2D NF model [36] with a single microcolumn.

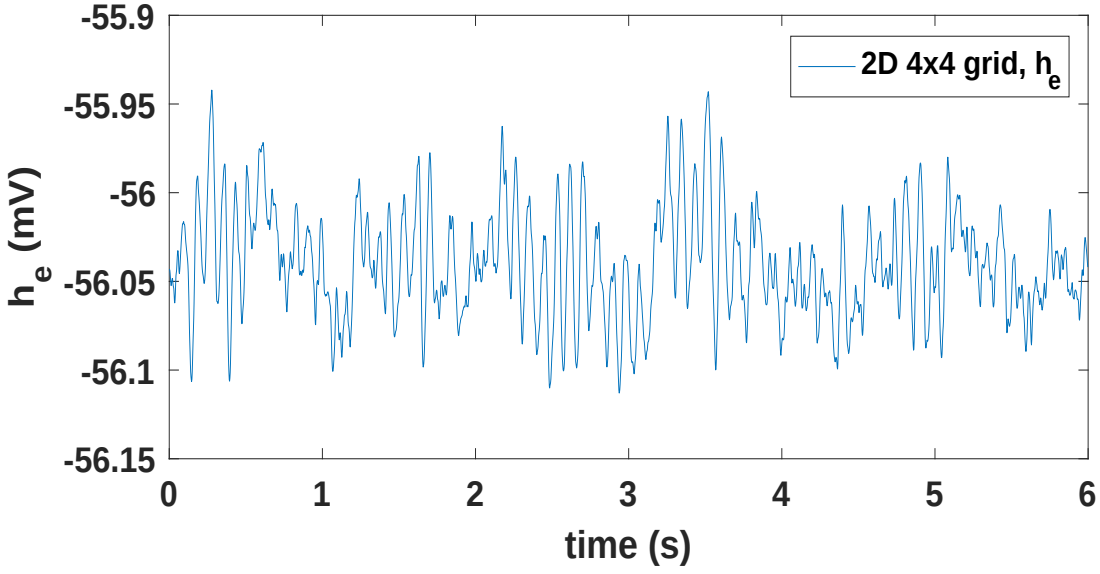

Figure 4.2: h_e time series for the 2D NF model [36] with 4x4 microcolumns. The signal is from a single microcolumn located in the central region of the square grid.

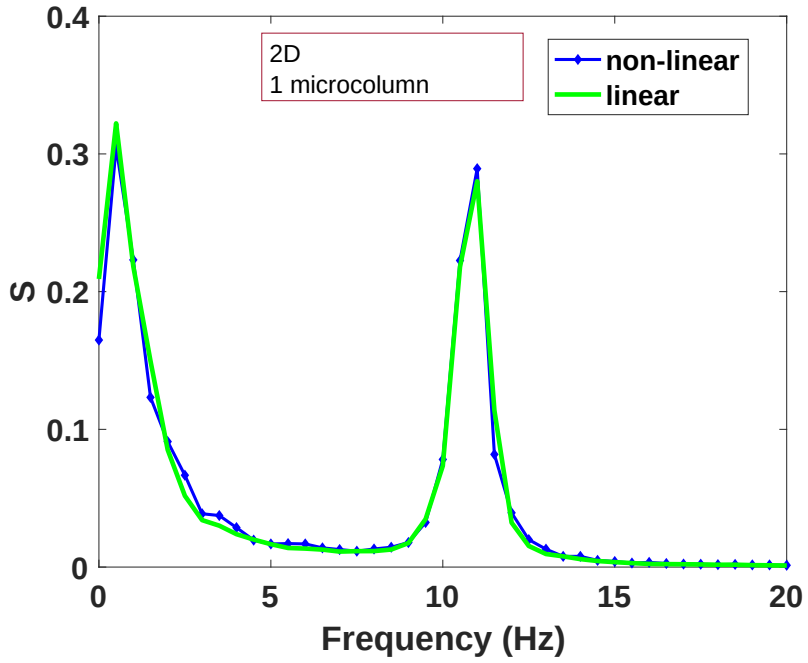

Figure 4.3: Linear and non-linear spectra for one macrocolumn 2D system.

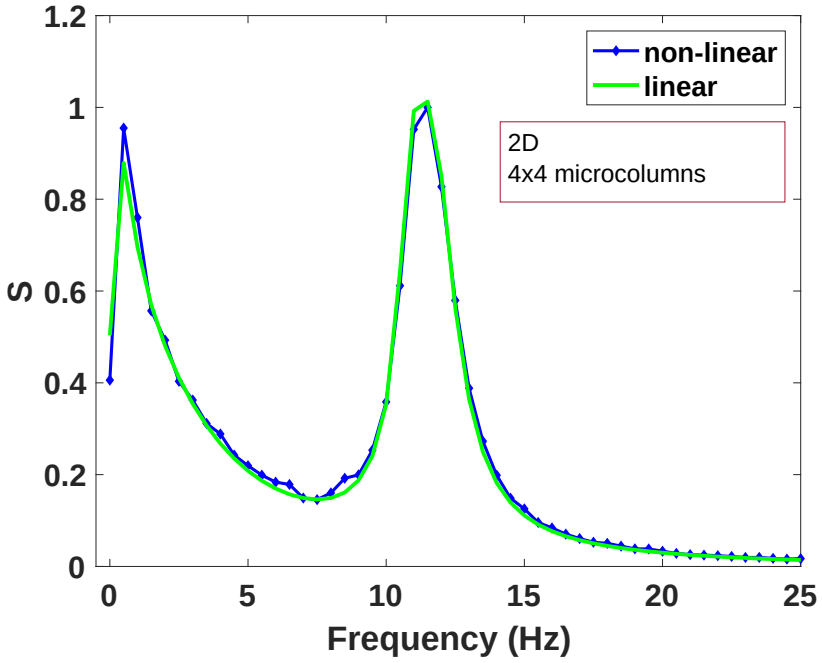

Figure 4.4: Linear and non-linear spectra for a 4x4 macrocolumns 2D system.

4.3 3D NFM spectra

The ODE system (3.53)-(3.56), including 30 dendritic compartments, has been implemented in MatLab and solved using the Euler method. The main procedures and functions used to calculate the time series and the linear and non-linear PSD are reported and explained in the Appendix, for systems with a single or multiple microcolumns.

4.3.1 Dynamics and computational time

Figure 4.5 shows 6s time series of the excitatory membrane potential h_e, for a system including a single microcolumn. This corresponds to 43 ODEs of the form (3.53)-(3.56) where the cortical inputs Φ_{ek} of the neighbour populations are set to zero. The corresponding non-linear (*blue*) and linear (*green*) PSDs are shown in Fig. 4.7. The non-linear spectrum is computed from 200s time series, while the linear one is calculated using Eq. (3.73). Figures 4.6 and 4.8 show 6s time series and the corresponding PSD for a system of 16 microcolumns, distributed on a 4x4 square grid. This configuration is mathematically represented by a 4x4 ODE system

$$\frac{\partial \mathbf{S}}{\partial t} = \mathbf{F}(\mathbf{S}) + \mathbf{P}$$

where the state variable S is obtained by concatenation of 16 43-dimensional state variables corresponding to single macrocolumns and P is the extra-cortical input vector. The neural populations are spatially coupled via the Laplacian of the cortical connections, calculated through the 5-points stencil approximation, as shown in Eqs. (3.56). The system is run for 50 seconds and the time series represented in Fig. 4.6 corresponds to a single microcolumn located in a central region of the grid, while the non-linear spectrum is computed taking into account the time series of all the microcolumns[3]. Furthermore, the linear spectrum is calculated using Eq. (3.71). In the procedure used to calculate the linear PSD of both one and multiple microcolumn models, the state variable of the system is defined as a symbolic vector and the Jacobian matrix is computed using the MatLab function *jacobian*.

[3]See procedure *non_linear_spectrum_full_grid* illustrated in the Appendix.

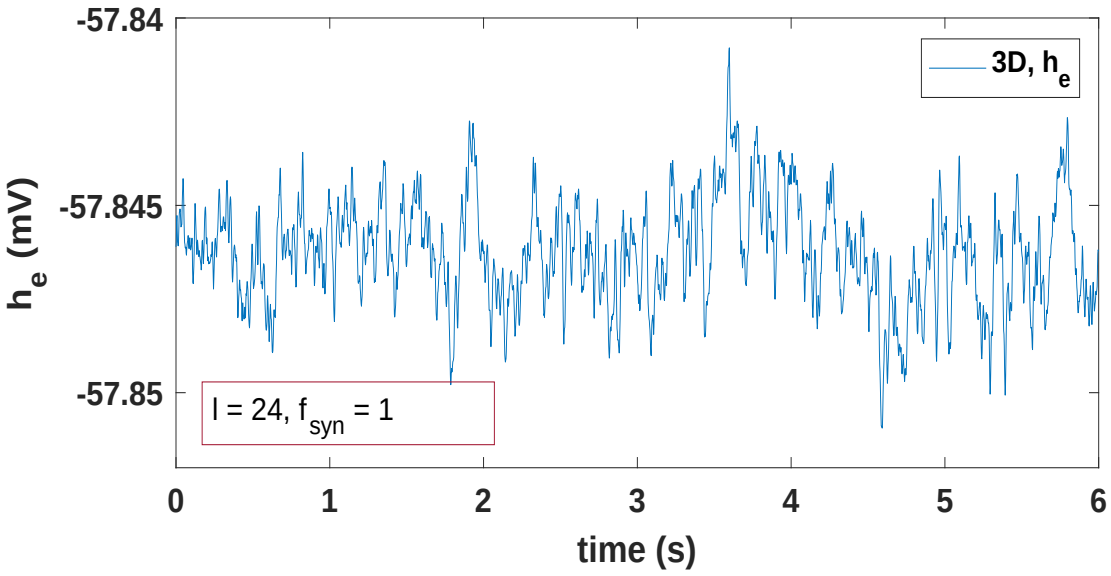

Figure 4.5: Somatic h_e time series for the 3D implemented NF model (3.51).

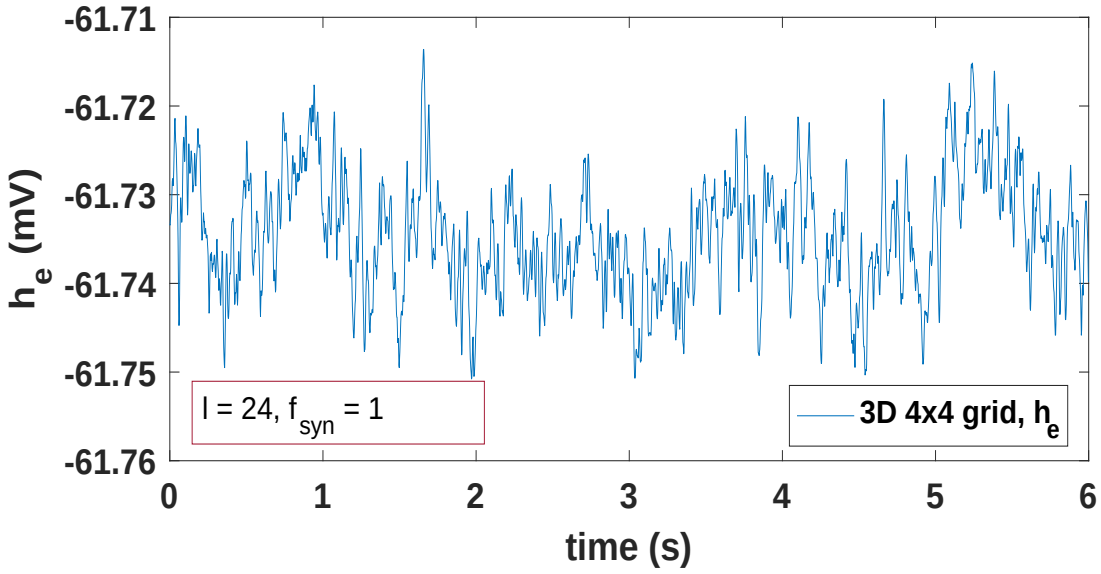

Figure 4.6: Somatic h_e time series from one central microcolumn on a 4x4 square grid representing a 3D NF model (3.51) with 4x4 populations.

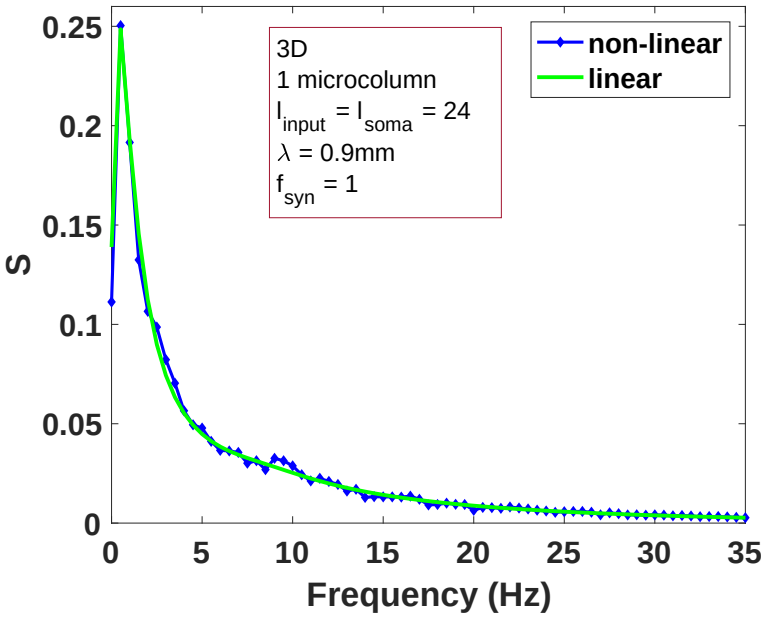

Figure 4.7: Linear and non-linear spectra for one macrocolumn 3D system, with 30 dendritic compartments.

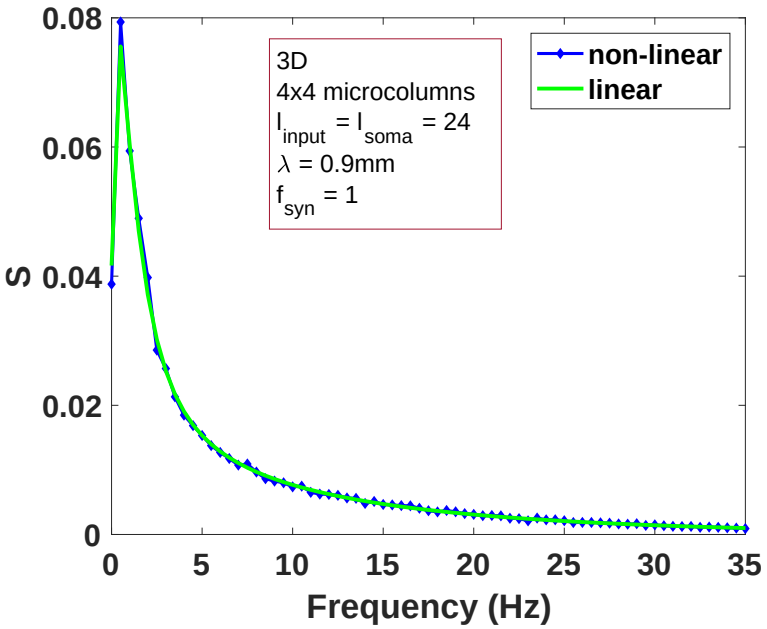

Figure 4.8: Linear and non-linear spectra for a 4x4 macrocolumns 3D system, with 30 dendritic compartments.

The mean computational time to simulate the non-linear time series together with the PSD are 3.2567×10^4 s (corresponding to 200 s cortical activity) and 3.0841×10^3 s (corresponding to 200 s cortical activity) for 2D system with single or multiple microcolumns (4x4 grid), while the corresponding simulations for a 3D system with 30 dendritic compartments take about 3.3072×10^4 s and 4.0900×10^3 s. The linear prediction based on the MatLab function *jacobian* takes 3.1998 s and 541.7867 s for the 2D cases; 4.4261 s and 1.6026×10^4 s for the 3D simulation. To improve the linear computational time, a procedure to calculate the linear PSD has been implemented from scratch for single microcolumn systems. The procedure calculates the Jacobian matrix of the linearized system by computing all the derivatives

$$\frac{\partial}{\partial \mathbf{S}} \frac{\partial \mathbf{S}}{\partial t}$$

and does not employ the use of symbolic variables. The time needed to compute the linear PSD is 0.0821 s and 3.3072 s respectively for the 2D and 3D cases. The procedure is reported in the Appendix , for the 3D configurations.

Although single microcolumn systems only reproduce the activity of $10^4 - 10^5$ neurons, they are able to capture the qualitative features of multiple microcolumn systems (Figs. 4.1-4.8). Both in Chapters 5 and 6 we mainly analyse the dynamics of a single microcolumn 3D systems with 30 dendritic compartments. As future work, this investigation could be extended to multiple microcolumns 3D systems.

4.4 Loss of alpha rhythm

As it can be noted from Figs. 4.3-4.4 and 4.7-4.8, the PSDs of the 3D systems do not exhibit a peak in the alpha band typical of spectra encountered in humans. This phenomenon is due to the spread of the signal along the dendrites. As expected, the PSD dynamics of a 3D single microcolumn system where the length constant has been set to zero is identical to the one of the 2D geometry (Fig. 4.9). A progressively increase in λ makes the alpha peak smaller and shifted towards higher grequencies (see Fig. 4.9), until, for $\lambda = 0.9$ mm, the alpha rhythmicity is lost (see Figs. 4.7 and 4.9). However, we found that this behaviour can be modified by either strengthening

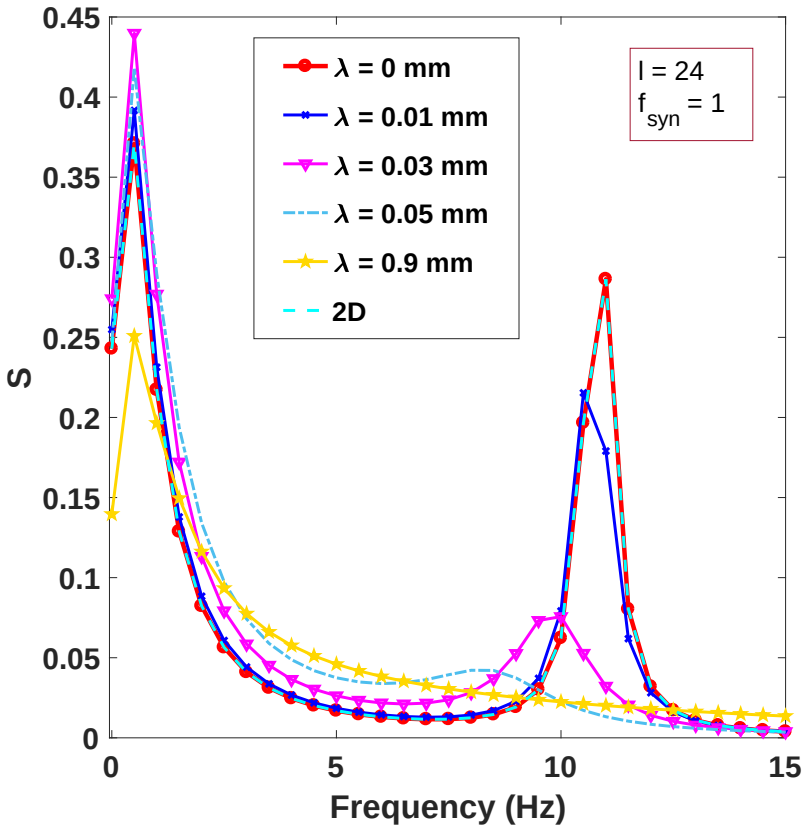

Figure 4.9: Linear spectra for one macrocolumn 3D system, with 30 dendritic compartments and different values of λ.

the synaptic input and/or changing some of the parameters of the system. For a 3D system with 30 dendritic compartments and input located at the soma ($l = 24$), an increase in synaptic strength is sufficient to restore the alpha rhythm (see *black line* in Fig. 4.13). Conversely, for increasing input distance a simultaneous physiological modification of other parameters of the model is needed to produce alphoid spectrograms (Chapters 5 and 6). In the next Section the impact of some of the parameters of the system on the alpha rhythmicity is explored, while in the next Chapter an optimization technique is used in order to find "suitable" parameters, e.g. giving rise to aplhoid dynamics.

4.5 Parameters to restore the alpha rhythm

In this section the role of some of the parameters of the model on the system dynamics is illustrated, with a particular focus on the effect on the alpha rhythm. This study is useful to detect suitable search spaces to identify physiologically admissible parameter sets giving rise to alpha activity (Chapters 5 and 6).

Mean resting membrane potential h_e^r and passive membrane decay time constant τ_e

Figure 4.10 shows the equilibrium membrane potential along the dendrite for systems with an without synaptic inputs. The equilibrium value is constant and equal to the mean resting membrane potential h_e^r for the passive cable, while it presents a peak in correspondence with the input location when synaptic currents are active. If no inputs are applied to the system (3.51), the membrane potential

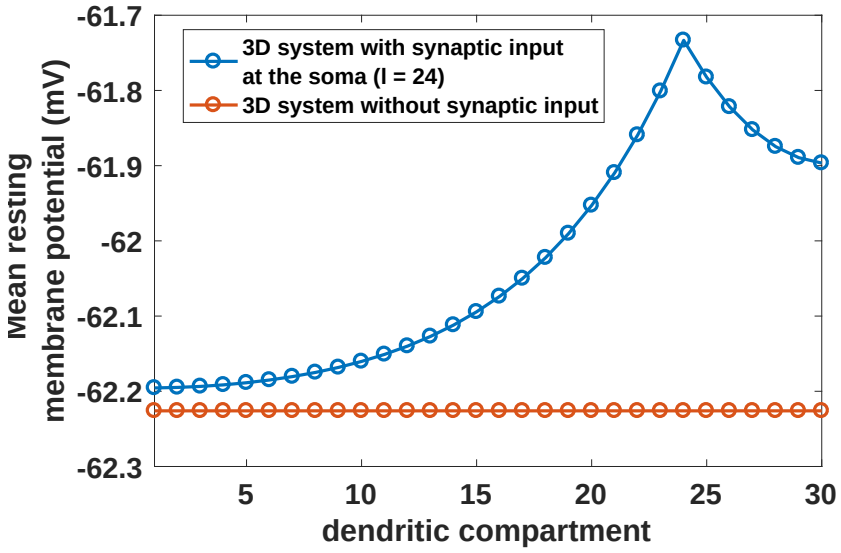

Figure 4.10: Equilibrium membrane potential h_e over the dendrite for 3D systems with (*blue line*) and without synaptic input (*orange line*) respectively; $\lambda = 0.9$ mm.

converges to its resting value h_e^r, with a rate that is regulated by the decay time constant τ_e (Fig. 4.11).

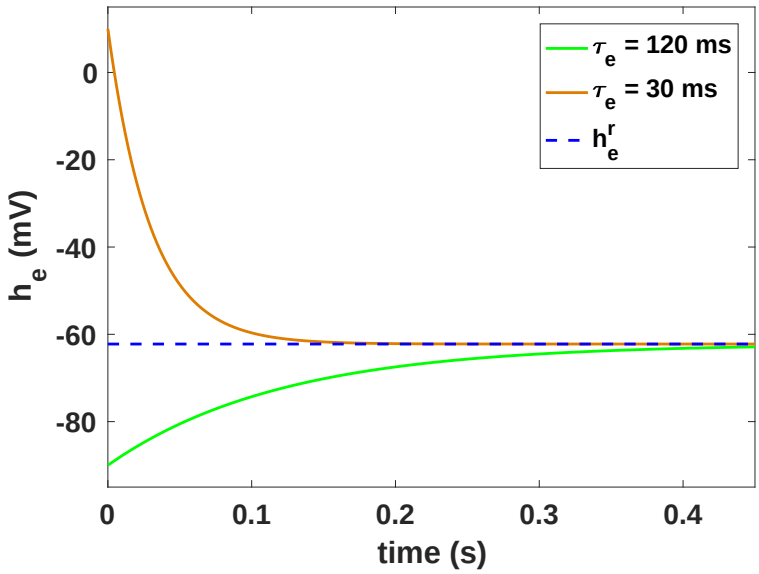

Figure 4.11: Membrane potential h_e converging to the mean resting membrane potential h_e^r, in 3D systems without synaptic input and with different decay time constants τ_e; $\lambda = 0.9$ mm.

In Fig. 4.12 we see that larger h_e^r imply higher frequency and less damping of the alpha peak, since the system becomes more excitable with rest closer to firing threshold.

In Fig. 4.13 we see that large τ_e mean higher damping of the alpha peak and less damping of the delta peak. The system becomes less responsive to rapid changes but shows prolonged reactions, cf. Fig. **??**.

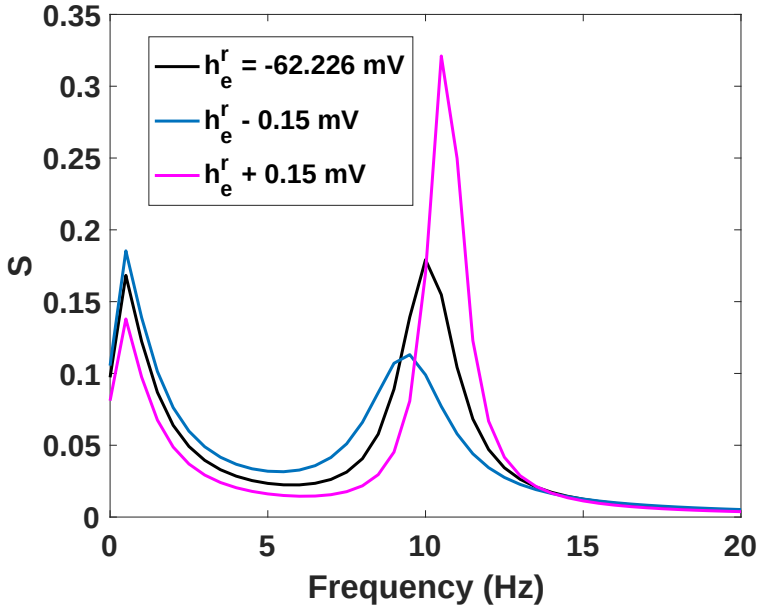

Figure 4.12: Alpha-rhythm for 3D systems with different values of h_e^r. The input is located at the soma with $f_{syn} = 9$ and $\lambda = 0.9$ mm.

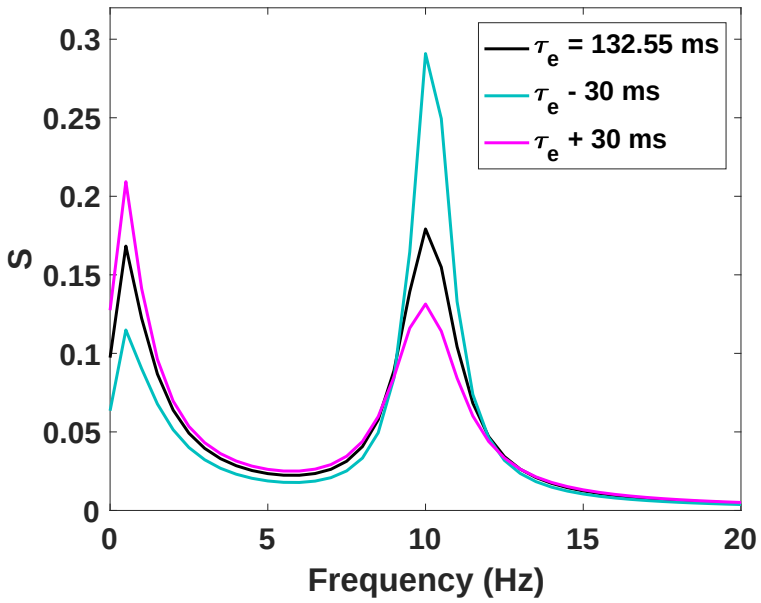

Figure 4.13: Alpha-rhythm for 3D systems with different values of τ_e. The input is located at the soma with $f_{syn} = 9$ and $\lambda = 0.9$ mm.

Mean Nernst reversal potential h_{ee}^{eq}

The synaptic currents I_{lk} are modulated by the functions ψ_{lk}, defined in Eq. (3.57) as

$$\psi_{lk}(h_k) = \frac{h_{lk}^{eq} - h_k(\mathbf{x}, t)}{\left|h_{lk}^{eq} - h_k^r\right|}.$$

Function ψ_{ee} is shown in Fig. 4.14. The post-synaptic potential (PSP) is normed at resting potentials, i.e. $\psi_{ee}(h_e) = 1$ if $h_e(t) = h_e^r$, and scaled by the function $\psi(h_e)$ at other potentials. If $h_e(t) < h_{ee}^{eq}$ then the modulating function drives the membrane potential towards positive values, while for potentials higher than the Nernst reversal potential, h_e is pushed towards depolarization, since $\psi_{ee}(h_e)$ becomes negative. Figure 4.15 shows that higher h_{ee}^{eq} induce less damping of the alpha peak.

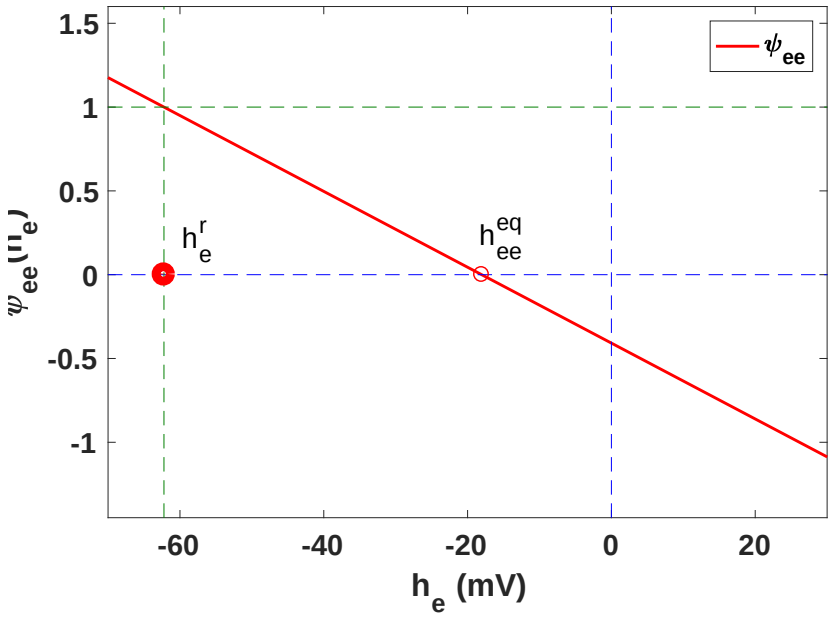

Figure 4.14: Function Ψ_{ee} modulating the synaptic current I_{ee} in Eqs. (3.53).

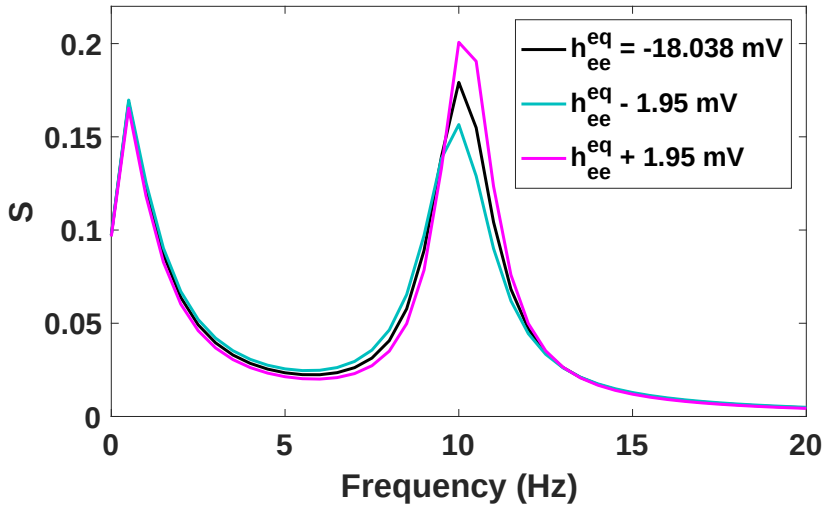

Figure 4.15: Alpha-rhythm for 3D systems with different values of h_{ee}^{eq}. The input is located at the soma with $f_{syn} = 9$ and $\lambda = 0.9$ mm.

Post-synaptic potential rate constants γ_{lk}

The post-synaptic potential rate constants γ_{lk} regulate the post-synaptic membrane potential response. Figure 4.16 shows the responses to single pulse input of the form (6.2) for different γ_{ee}. The input is applied at the soma[4] in a 3D system with $f_{syn} = 9$ and $\lambda = 0.9$ mm. A significant change in γ_{ee} (of about 100 s^{-1}) is needed to induce small variations (of about 0.6 mV) in the membrane potential response. Figures 4.17-4.20 show the impact of varying γ_{ee}, γ_{ei}, γ_{ie}, and γ_{ii}, respectively, on the alpha rhythm of 3D systems with the input located at the soma, $f_{syn} = 9$ and $\lambda = 0.9$ mm. Larger γ_{ee} mean lower frequency and more damping of the alpha peak. The same effect is obtained for larger γii. Larger γ_{ei} implicates less damping of the peak, while larger γ_{ie} implicates higher frequency and less damping.

[4]Responses to pulse input will be analysed in Chapter 6.

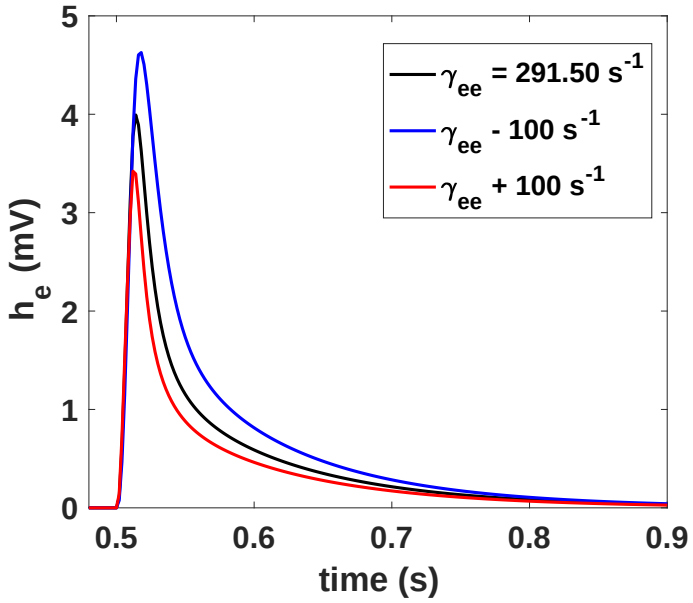

Figure 4.16: h_e responses to pulse input for 3D systems with different values of γ_{ee}. The input is located at the soma with $f_{syn} = 9$ and $\lambda = 0.9$ mm.

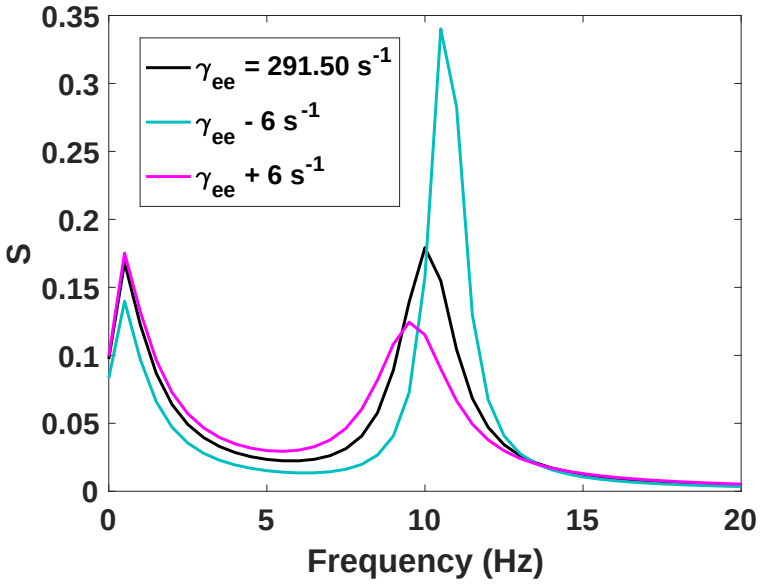

Figure 4.17: Alpha-rhythm for 3D systems with different values of γ_{ee}. The input is located at the soma with $f_{syn} = 9$ and $\lambda = 0.9$ mm.

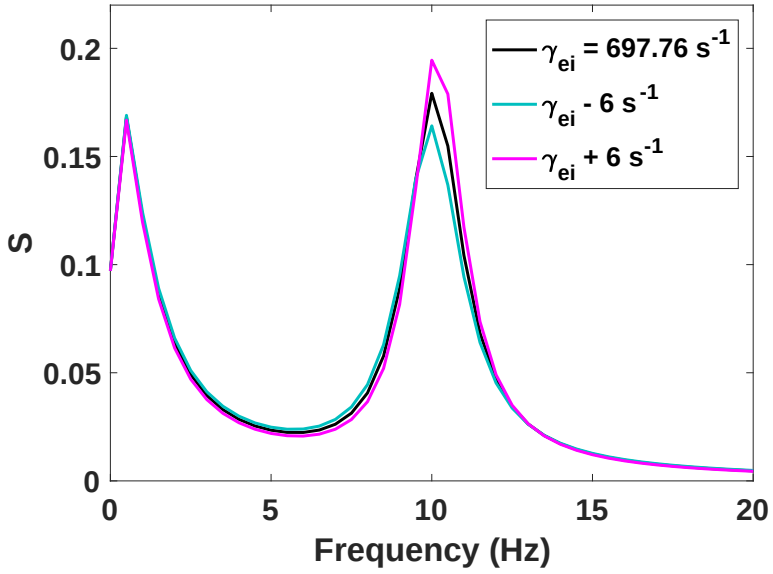

Figure 4.18: Alpha-rhythm for 3D systems with different values of γ_{ei}. The input is located at the soma with $f_{syn} = 9$ and $\lambda = 0.9$ mm.

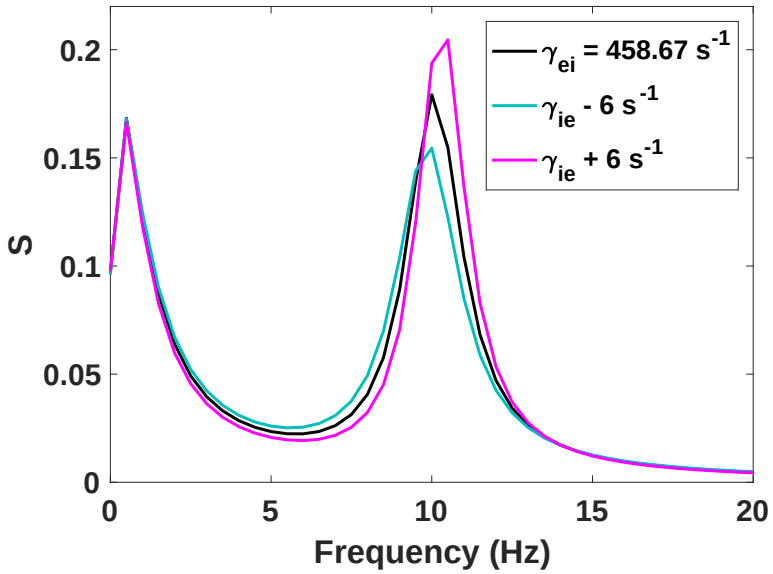

Figure 4.19: Alpha-rhythm for 3D systems with different values of γ_{ie}. The input is located at the soma with $f_{syn} = 9$ and $\lambda = 0.9$ mm.

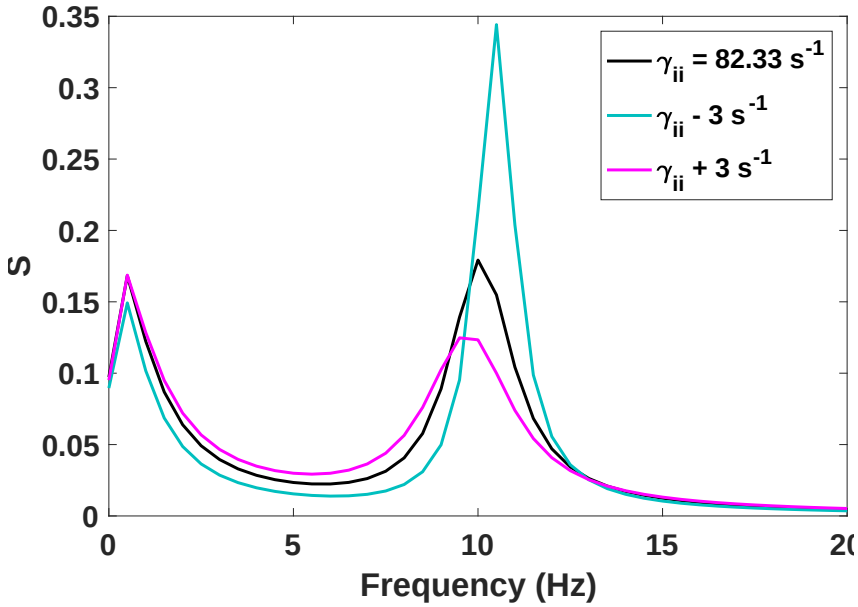

Figure 4.20: Alpha-rhythm for 3D systems with different values of γ_{ii}. The input is located at the soma with $f_{syn} = 9$ and $\lambda = 0.9$ mm.

4.6 Spatio-temporal patterns

The spatio-temporal dynamics of 2D and 3D systems consisting of 256 micro-columns (distributed on a 16x16 square grid) is illustrated in this section. The membrane potential of the 3D system has been observed at 4 different cortical depth: at the soma (compartment 24), in a proximal region (compartment 22, distance $d = 0.2667$ mm from the soma), in a central and distal apical domains (corresponding to compartments 15 and 6, distances $d = 1.2$ mm and $d = 2.4$ mm, respectively). For each configuration, the spatio-temporal patterns for the entire grid are represented, while the 6s time-series correspond to the activity of one of the macrocolumns of the system. A factor $f_{syn} = 9$ has been introduced for the synaptc input located at the soma. Results are shown in the following figures (4.21-4.30) and indicate that as the distance from the soma increases, the amplitude and frequency of the membrane potential oscillations is reduced.

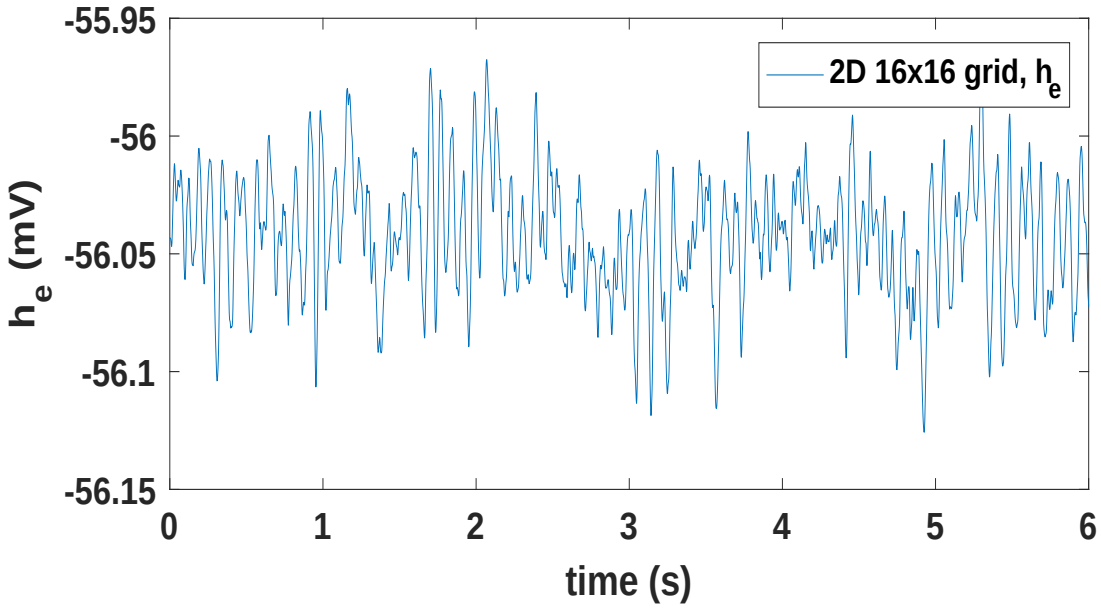

Figure 4.21: h_e time series for the 2D NF model [36] with 16x16 microcolumns. The signal is from a single microcolumn located in the central region of the square grid.

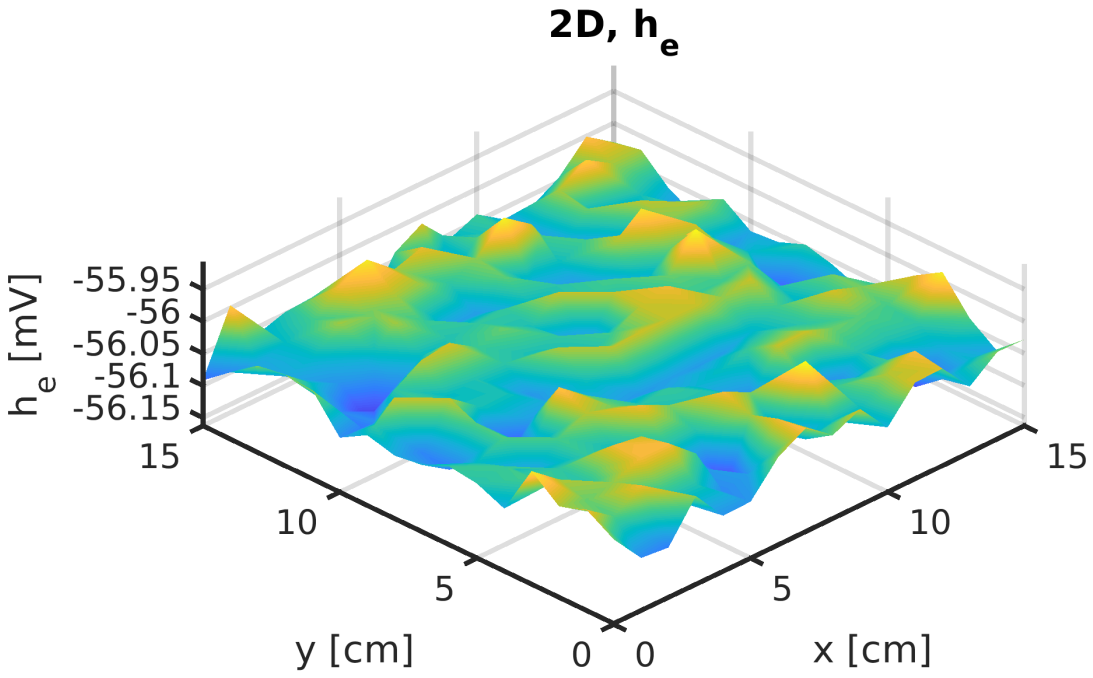

Figure 4.22: h_e cortical activity for the 2D NF model [36] with 16x16 microcolumns.

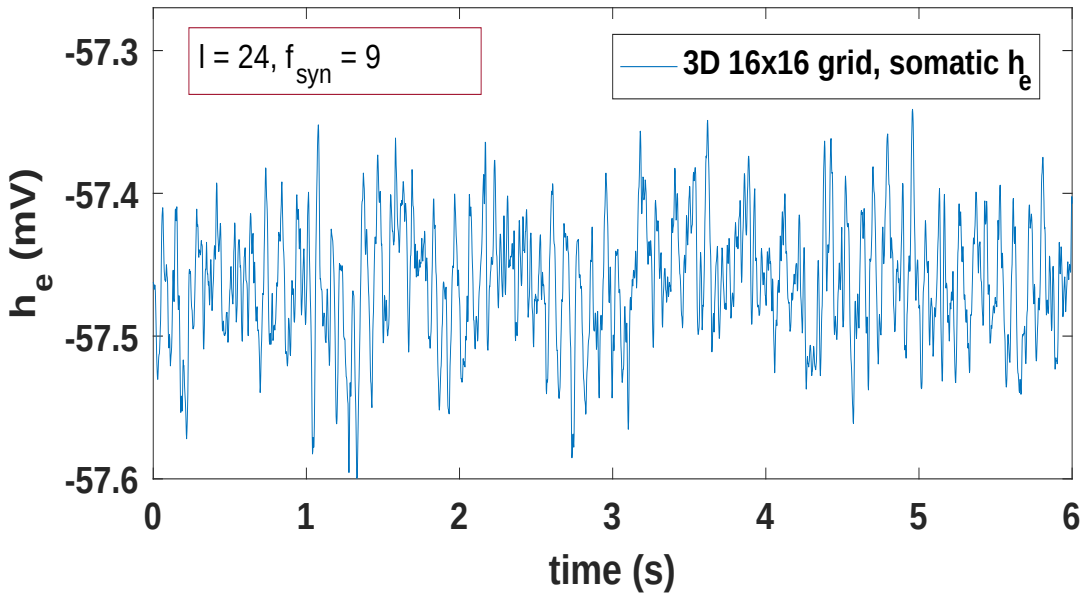

Figure 4.23: Somatic (compartment 24) h_e time series from a single microcolumn of the 3D NF model (3.51) with 16x16 microcolumns.

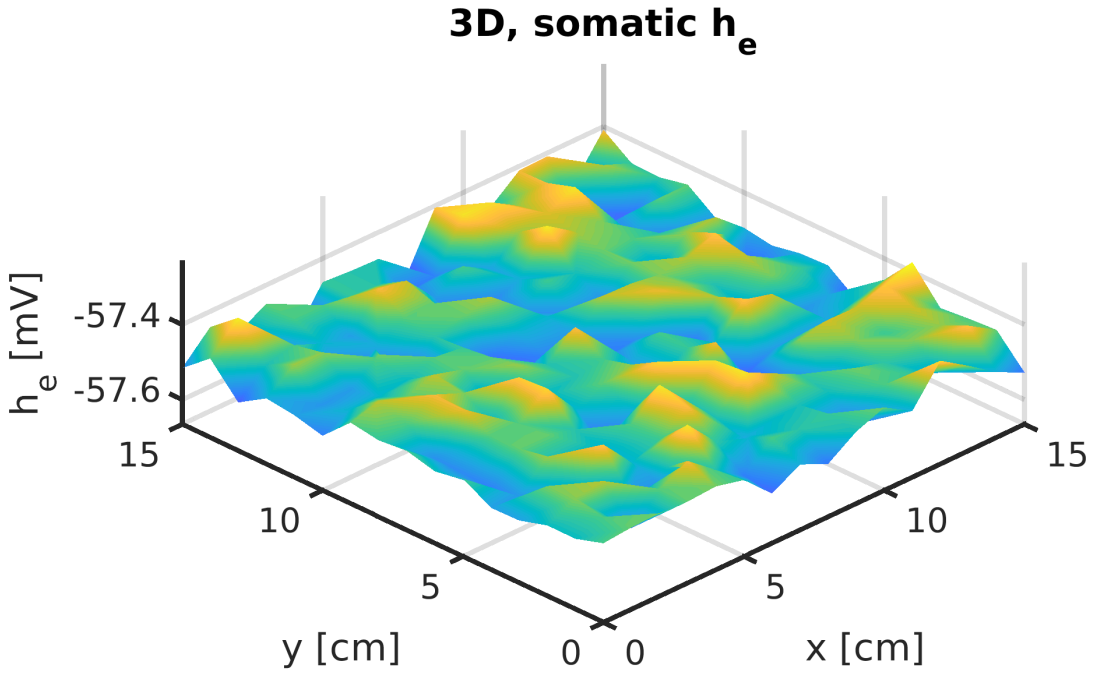

Figure 4.24: Somatic ($l = 24$) h_e cortical activity for the 3D NF model (3.51) with 16x16 microcolumns.

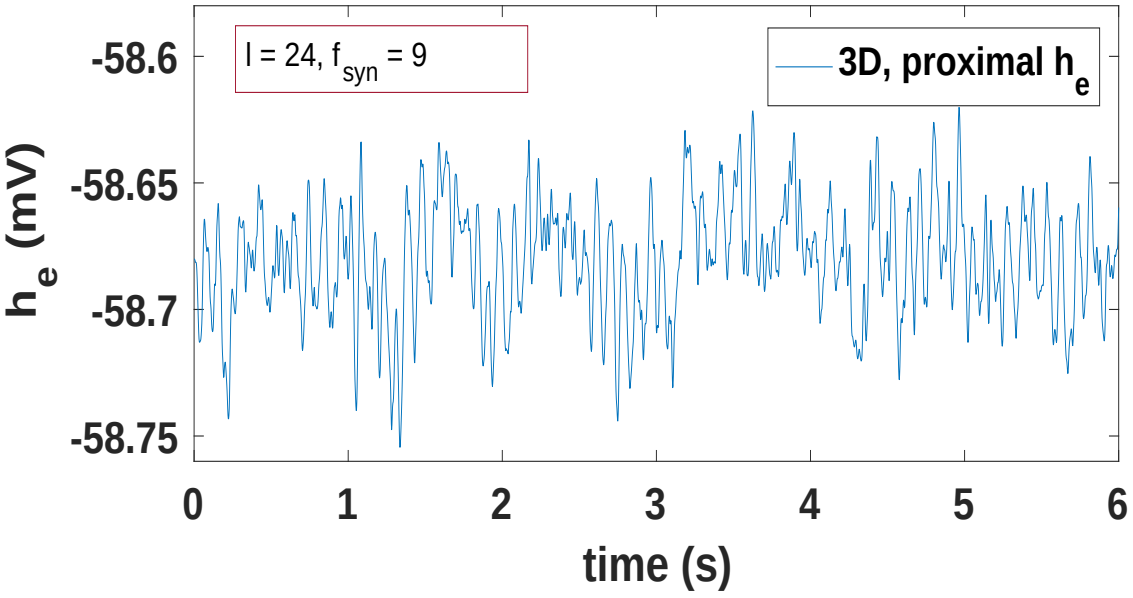

Figure 4.25: Proximal (compartment 21) h_e time series from a single microcolumn of the 3D NF model (3.51) with 16x16 microcolumns.

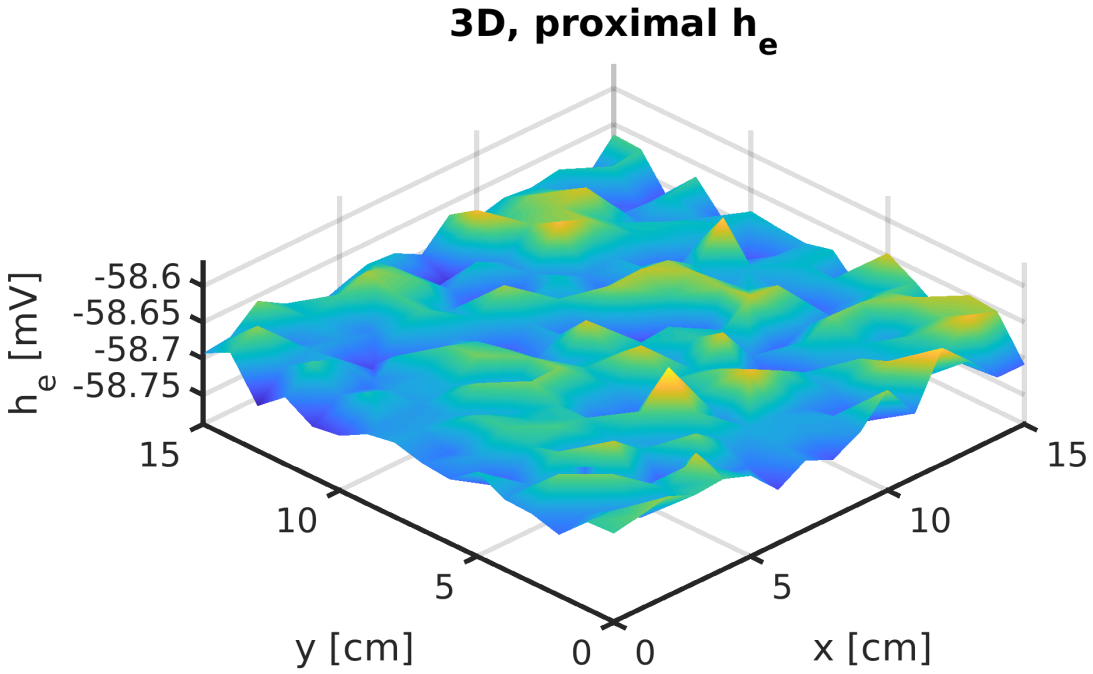

Figure 4.26: Proximal h_e ($l = 22$) cortical activity for the 3D NF model (3.51) with 16x16 microcolumns.

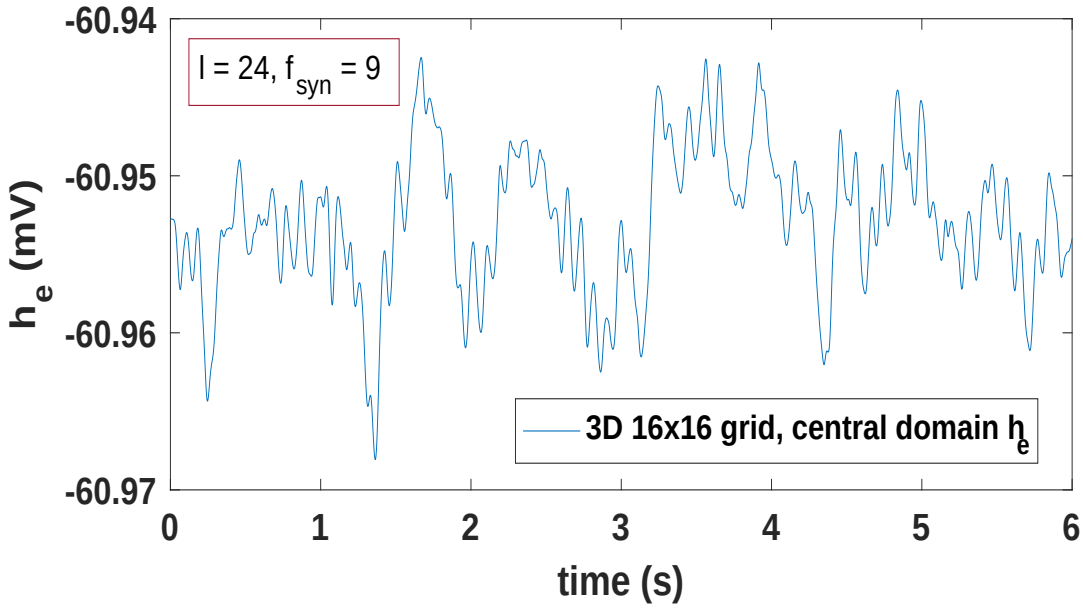

Figure 4.27: Central apical (compartment 15) h_e time series from a single microcolumn of the 3D NF model (3.51) with 16x16 microcolumns.

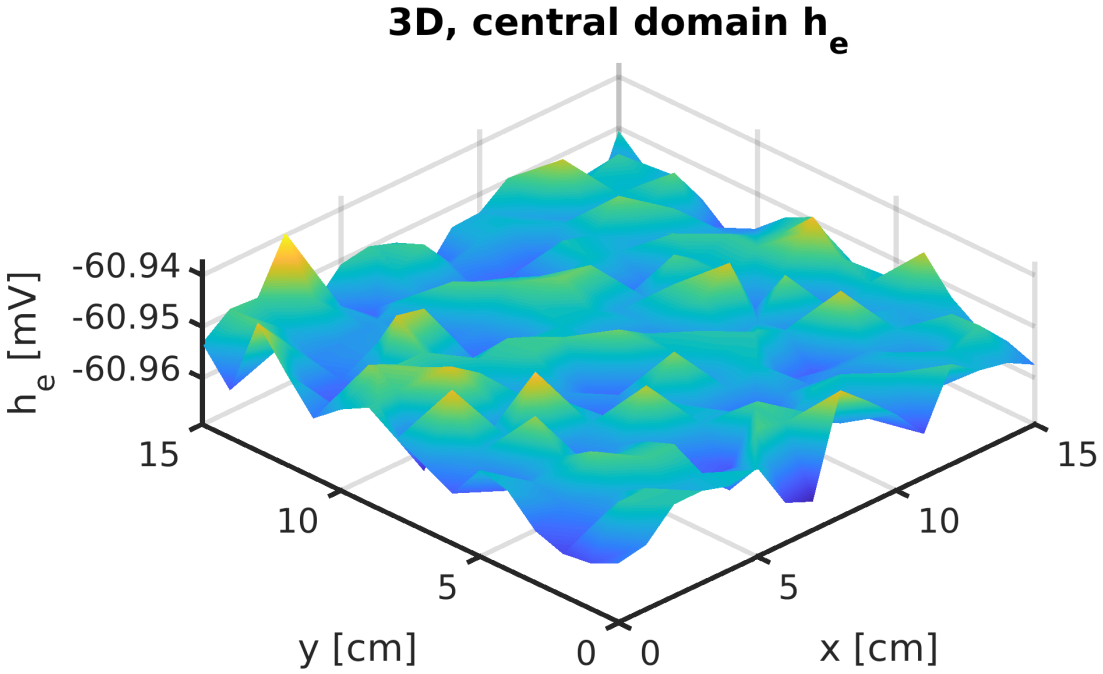

Figure 4.28: Central domain h_e ($l = 15$) cortical activity for the 3D NF model (3.51) with 16x16 microcolumns.

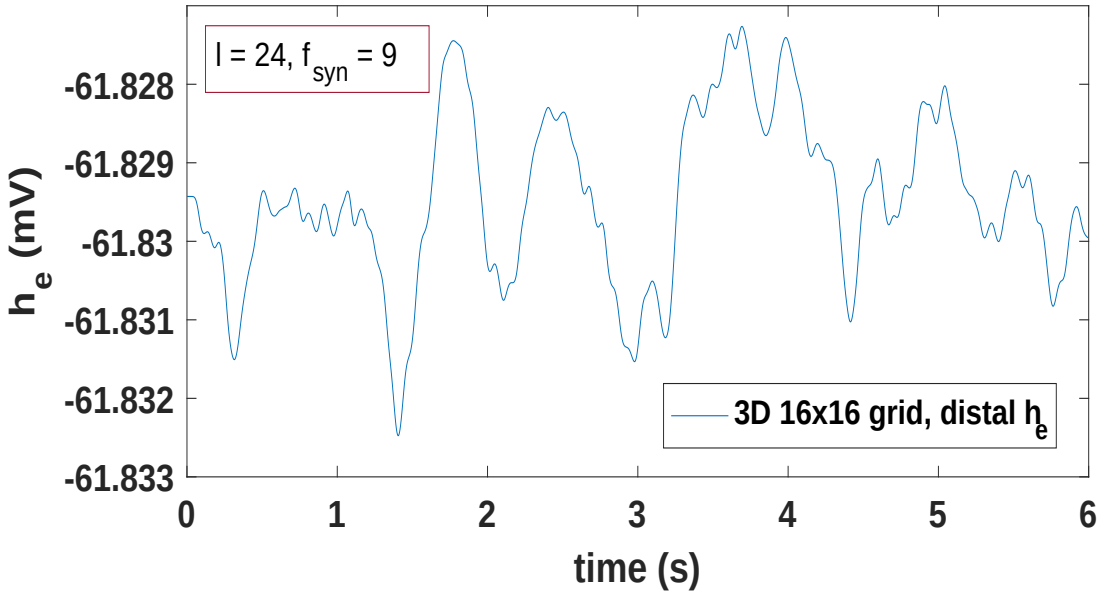

Figure 4.29: Distal apical (compartment 15) h_e time series from a single microcolumn of the 3D NF model (3.51) with 16x16 microcolumns.

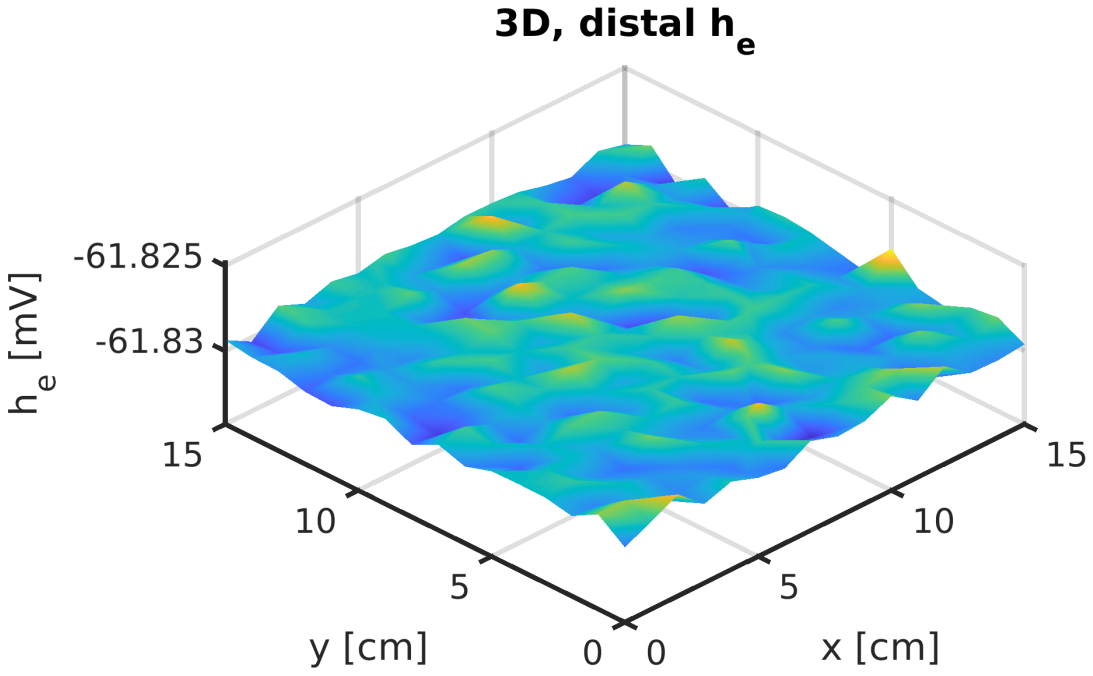

Figure 4.30: Distal h_e ($l = 6$) cortical activity for the 3D NF model (3.51) with 16x16 microcolumns.

4.7 Discussion

The 3-dimensional cortical NFM including the dendritic dimension has been developed starting from the 2-dimensional Liley and Bojak model [36]. Firstly, the model [36] has been implemented and the alpha-rhythmicity typical of human cortical activity at rest has been reproduced. Secondly, the dynamics of the 3D model has been explored.

The technique to predict the power spectra density (PSD) from the Jacobian matrix of the linearised system evaluated at a singular point was formulated by Liley and Bojak in [36]. Having performed a Fourier transform of the linear system with respect to both space and time, the authors calculated the PSD using equation (3.74) considering all possible wave numbers k, while equations (3.56) are formally decoupled[5]. In the present work, only the temporal Fourier transform is applied and the PSD is computed taking into account the contribution of all microcolumns (using Eq. (3.71)), spatially coupled through the cortical input[6]. Results show that, in close proximity to a stable fixed point, the formula accurately predicts the PSD for both 2D and 3D systems of 16 microcolumns distributed in a 4x4 grid.

Equally, close to the stable fixed point, the linear prediction matches the non-linear spectrogram for systems consisting of one microcolumn. Moreover, under the given condition of isotropic and homogeneous connectivity, our results show that a single microcolumn exhibits spectra tipically similar to the ones of multiple microcolumns.

Alpha-rhythmicity in the 3D model

Naively extending the 2D NF model to include a dendritic dimension keeping all the parameters of the 2D model [36] unchanged, results in a 3D model that loses the alphoid dynamics, as a consequence of the loss of signal along the dendrite.

[5]The Laplacian of the cortical input Φ_{ek} of Eqs. (3.51) is formally replaced with the square of the wave number, $\nabla^2 \to -\mathbf{k^2}$. As a consequence, the cortical input Φ_{ek} of each microcolumn is not coupled with the cortical variable of neighbour populations.

[6]The Laplacian of the cortical input Φ_{ek} of Eqs. (3.51) is computed using the cortical input of neighbour populations through the 5-points stencil approximation (see Eqs. (3.56)).

Our findings provide evidence that the alpha-rhythm can be restored by introducing a factor f_{syn} strengthening the synaptic inputs, and/or modifying some of the parameters of the model.

Spatio-temporal patterns

The spatio-temporal activity of the 2D NFM has been compared with the one of the 3D NFM, for 4 different cortical depths: at the soma, in a proximal domain, in central and distal apical regions. The two models (2D and 3D) consist of 256 microcolumns distributed in a 16x16 square grid. In the specific configuration considered for the 3D model, a synaptic factor $f_{syn} = 9$ has been introduced for the synaptic input located at the soma. Results show that, over the entire network, the membrane potential of the proximal regions is slightly reduced both in amplitude and oscillation frequency, while the activity is significantly reduced in the apical domains. This behaviour can be regulated by modifying the dendritic length constant and other parameters of the system.

Chapter 5

Use of Particle Swarm Optimization to find optimal PSDs

5.1 Introduction

In this chapter the use of the Particle Swarm Optimization (PSO) technique, applied to detect PSDs similar in shape to the ones observed in humans, is illustrated. The approach is based on the optimization of some parameters of the model with respect to an *objective function* measuring this similarity. Although this algorithm have been applied in [36] with the same purpose, it is the first time it is used to optimize the dendritic parameters of a 3D Neural Field Model in order to detect "good" spectra.

The analysis of the results that followed the literature's study and the implementation of the algorithm from scratch, has been particularly rich in terms of new directions arose from my research. Different systems (in terms of distribution of synaptic input along the dendrite) have been analyzed and optimal solutions have been found for some specific configurations. The investigation on the cases where the solutions were not optimal as expected required a deeper analysis and determined a substantial improvement of my research. A deeper understanding of the dynamics of the system has been acquired with respect to the functions and parameters of the system. Different techniques have been developed to improve the performance of

the algorithm, involving both the choice of the PSO parameters and a progressive improvement of the algorithm's *objective function*.

5.2 Particle Swarm Optimization (PSO) literature

The Particle Swarm Optimization (PSO) is a computational method used to find best solutions to nonlinear global maximization and minimization numeric problems. The PSO was first introduced in 1995 by James Kenney and Russell C. Eberhart [215], to describe the social behaviour of bird flocking or fish schooling. It is based on the idea that all the individuals (*particles*) work in the same way. Therefore the global best solution, is provided by adjusting, at each step, the trajectory of each individual toward its own best location and toward the best location among all the particles of the entire swarm. Since its first appearance, PSO techniques have gained increasing attention both in science and technology. The fields of applications include electric power systems [216], image clustering and pattern recognition [217, 218], artificial intelligence based neural networks [219–222], nonlinear regression [223], system control [224], heating systems [225], geotechnical engineering [226], Cloud workflow scheduling [227]. An extensive survey of PSO applications is provided in [228] and [229]. Due to the wide range of possible applications, the PSO technique evolved in several developments and variants [230, 231]. A reviews on both theoretical- experimental studies and recent developments can be found respectively in [232] and [233].

The nature of PSO, its successful applications to solve nonlinear problems together with its suitability to optimize neural networks and find optimal solutions in multidimensional parameter spaces, makes it a promising approach to my problem. The nonlinear cortical system I developed in my thesis is particularly sensitive to the wide variety of parameters included the model. As seen in previous chapters the problem of finding specific values for the parameters leading to PSDs shapes encountered in humans, is particularly interesting for the purpose of the present work, and can be approached only with sophisticated optimization techniques as the PSO.

5.3 How the PSO algorithm works

The PSO technique is a computational method that iteratively improves the candidate solutions with regard to a given *fitness function* (also called *cost function* or *objective function*). The *fitness function* is the function that need to be optimised (minimised or maximised). The positions of the *particles* in the search-space are the candidate solutions. How "good" is the position of a certain particle is determined by the evaluation of the *fitness function* in that position. At each iteration, the particles adjust their positions and velocities in the search-space according to simple rules: their movement is influenced by their local known best position and also by the global best known positions in the entire search-space. This process is expected to make the swarm eventually converge towards its best solution.

The basic version of the PSO algorithm is illustrated in this section. Formally, let

$$f\colon \mathbb{R}^m \to \mathbb{R}$$

be the *fitness function* to be minimized (or maximized). Let

$$S \subset \mathbb{R}^m$$

be the *m-dimensional* search-space where the M particles are located, each having a position $\mathbf{x_i} \in S$ and a velocity $\mathbf{v_i} \in \mathbb{R}^m$, for $i \in \{1, ..., M\}$. The goal is to find a solution $\mathbf{x_{best}} \in S$ such that

$$f(\mathbf{x_{best}}) < f(\mathbf{y}) \quad \forall \mathbf{y} \in S.$$

We want to find the best solution within the search-space S. Let $\mathbf{b_{lo}} \in \mathbb{R}^m$ and $\mathbf{b_{up}} \in \mathbb{R}^m$ be respectively the lower and upper bounds defining the legal region S for the positions $\mathbf{x_i}$. Let $\mathbf{p_i}$ be the best known position of particle i among all the positions visited by i, and let $\mathbf{g}$ be the best known global position of the entire swarm (i.e. the position giving the smallest fitness evaluation among all the positions visited by all the particles of the swarm).

The particles are initialized with uniformly distributed initial positions

$$\mathbf{x_{i,0}} \sim U([\mathbf{b_{lo}}, \mathbf{b_{up}}]) \quad \forall i \in \{1, ..., M\} \tag{5.1}$$

Each particle's best position is initialized to its current initial positions

$$\mathbf{p_{i,0}} = \mathbf{x_{i,0}} \quad \forall i \in \{1, ..., M\}, \tag{5.2}$$

while the global best $\mathbf{g}$ is initially defined as

$$\mathbf{g_0} = \mathbf{p_j} \text{ where } f(\mathbf{p_j}) = min\left\{f(\mathbf{p_{i,0}}) \ : \ i \in \{1, ..., M\}\right\}. \tag{5.3}$$

Finally the velocities can be initialized to zero or have a uniform random distribution:

$$\mathbf{v_{i,0}} \sim U([- \mid \mathbf{b_{up}} - \mathbf{b_{lo}} \mid, \mid \mathbf{b_{up}} - \mathbf{b_{lo}} \mid]) \quad \forall i \in \{1, ..., M\}$$

At each step $t + 1$:

- The particles are moving influenced by the best local position $\mathbf{x_i}$ reached so far and by the current best global position $\mathbf{g}$. The formulas to update the velocities $\mathbf{v_i}(t + 1)$ are discussed below, while the positions are updated to:

$$\mathbf{x_i}(t + 1) = \mathbf{x_i}(t) + \mathbf{v_i}(t + 1) \tag{5.4}$$

- The fitness function is evaluated in all the positions $\mathbf{x_i}$ and the best local position $\mathbf{p_i}$ (for each particle i) is updated if a better local evaluation of the the fitness has been found:

$$\mathbf{p_i}(t + 1) = \mathbf{x_i}(t + 1) \quad \text{if} \quad f(\mathbf{x_i}(t + 1)) < f(\mathbf{p_i}(t)), \ \ \forall \ i \in \{1, ..., M\}$$

- The global best $\mathbf{g}$ is updated if among all the best local solutions a global better evaluation has been found:

$$\mathbf{g}(t + 1) = \mathbf{p_i}(t + 1) \quad \text{if} \quad f(\mathbf{p_i}(t + 1)) < f(\mathbf{g}(t)) \ \ \forall \ i \in \{1, ..., M\}$$

  that is equivalent to

$$\mathbf{g}(t + 1) = \mathbf{p_j}(t + 1) \text{ where } f(\mathbf{p_j}(t + 1)) = min\left\{f(\mathbf{p_i}(t + 1)) \ : \ i \in \{1, ..., M\}\right\}.$$

Different formulas have bee used in literature to update the values of the particles velocities. The original formula introduced by [215] is:

$$\mathbf{v_i}(t + 1) = W_0 \ \mathbf{v_i}(t) + w_{soc} \ rand \ (\mathbf{p_i}(t) - \mathbf{x_i}(t)) + w_{cog} \ rand \ (\mathbf{g}(t) - \mathbf{x_i}(t)) \tag{5.5}$$

where *rand* is a random number in between 0 and 1; the parameter W_0 (*inertia weight*) has been proved to have a better control over the velocity of the swarm particles [234]; finally, the factors w_{soc} and w_{cog}, modulating respectively the *social* and *cognitive term* in Eq. (5.5), control the behaviour and the efficacy of the PSO method [234]. In the last two decades, a variety of studies have investigated on the parameters of the swarm velocities, to improve the overall performance of the algorithm. Clerc and Kennedy introduced in 2002 a new variant of PSO [235], based on a constriction factor α ensuring a better convergence of the search procedure. The velocity formula they introduced is:

$$\mathbf{v_i}(t+1) = \alpha\big(\ \mathbf{v_i}(t) + w_{soc}\ rand\ (\mathbf{p_i}(t) - \mathbf{x_i}(t)) + w_{cog}\ rand\ (\mathbf{g}(t) - \mathbf{x_i}(t)\big) \quad (5.6)$$

where

$$\alpha = \frac{2}{2 - w - \mid \sqrt{(w-4)w} \mid} \quad \text{and} \quad w = w_{soc} + w_{cog}. \quad (5.7)$$

As other optimisation algorithms, the PSO is easy to fall into local optima in high-dimensional spaces [236]. To deal with this problem, a great number of investigations have been carried out in the last decades, based on new "topologies" for the PSO. The topology of the swarm defines the subset of particles with which each particle can exchange information. At each iteration, the particles update their positions and velocities according to the evaluation of the fitness function of the "neighbours". As previously explained, they tend to converge towards the particle located in the best position. The canonical version of the algorithm is based on a global topology, meaning that all particles share the same *global best* position and converge towards this unique best evaluation. Alternative *local topologies* have been proposed, where the particles only share information with a subset of particles [237] and therefore tend to converge towards local best positions. These subsets, defining the new neighbourhood of the particles, can be geometrical [238] or social [239]. Algorithms with local neighbourhood schemes take inspiration from realistic swarms and generally provide more accuracy [240]. The reason behind the better performance is the reduced probability to fall into local minima [234, 239, 241, 242].

In fact, whenever a particle approaches a local minimum it only influences its neighbourhood, while other particles explore the rest of the search space.

5.4 Applying PSO to neural fields

We used the PSO algorithm to detect possible "good solutions" for the 3D NFM, in terms of PSD shapes encountered in humans. As seen in previous chapters, the flat spectra obtained naively extending the 2D model [36] into a 3D configuration, is mainly the consequence of the loss of signal along the dendritic tree that makes the system overall "less excitable". We remind the reader that, in the present work, the notion of excitability, explained in Section 3.2.5, is highly related to the capability of the system to generate alphoid spectra. In fact, as it will be explained in Section 6.2, systems that are more responsive to synaptic inputs are more likely able to give rise to alpha rhythm.

As seen in Section 4.5, several parameters have an impact on the alpha rhythmicity of the system. A manual search of possible combinations leading to "good" spectra would be intractable, due to the non-linear nature of the system. For this reason, we decided to use the PSO technique to address this problem. We have mainly studied two different parameter spaces, taking into account some of the parameters discussed in Section 4.5. The exploration of search spaces including other parameters would be an interesting target for future work.

The boundaries of the *search-spaces* are defined according to the physiological range of variability of the parameters. The particles *positions* represent particular states of the system, defined by fixed values of the *search-parameters*. The *particles* update their movement, in the parameter space, according to the evaluation of a *fitness function* that measures the similarity of the resulting PSD to typical PSDs observed in humans.

The procedure *particle_swarm* implementing the PSO Algorithm with global neighbourhood has been implemented from scratch and is reported and commented in the Appendix.

5.4.1 Assumptions

The analysis has been conducted on single microcolumn systems[1], with standard configuration:

- Inhibitory neuronal populations have point-shaped bodies.
- Excitatory neuronal populations have dendrites extending through the 4mm cortical depth. Dendrites are modelled as linear cables divided in 30 compartments and soma located in compartment 24.

The location of the synaptic input varies along the dendrite from the somatic compartment up to distal compartments.

In cases where the swarm global best solution does not improve (i.e. the variation in the fitness evaluation is less than a certain *tolerance* μ) for more than k consecutive iterations, the PSO search is interrupted and the MatLab function "fminsearch", that is more efficient into finding local minima, is used.

Settings

The investigated **search spaces** involve some of the parameters influencing the alpha rhythm discussed in Section 4.5. *For each different location l of the synaptic input and for every different value of the dendritic length constant* λ, we run the PSO algorithm in the search-space $\mathbf{S_1}$ defined by:

$$S_1 = \left\{ h_{ee}^{eq}, \gamma_{ee}, f_{syn} \right\} \tag{5.8}$$

The parameter γ_{ee} modulates the response of the excitatory membrane potential to a single excitatory pulse input: smaller values of γ_{ee} corresponds to larger responses of both the membrane potential h_e and the synaptic excitatory current I_{ee}. The parameter h_{ee}^{eq} is the mean Nernst reversal potential[2]. Results for this search space are illustrated in this Chapter, for different input locations and electrotonic length constants.

[1] This means that the velocity term in the equations for cortical connections (3.53) is zero.

[2] See Section 4.5 for more details about the parameters and their impact on the alpha rhythm.

In addition, a number of PSO simulations have been run in the search space $\mathbf{S_2}$:

$$S_2 = \left\{\tau_e, h_e^r, f_{syn}, \lambda\right\} \tag{5.9}$$

The time constant τ_e affects the time decay of the signal along the dendrite and the mean resting membrane potential h_e^r influences the steady states and the excitability of the system[3]. Results are shown in Section 6.8, for different dendritic configurations.

Lower and upper bounds for the parameters are illustrated in the table 5.1. The parameters τ_e, γ_{ee}, h_{ee}^{eq} and h_e^r vary in their physiological ranges defined in Tables 3.1 and 3.2. The range of variability considered for the dendritic length constant λ is $[0.1,\ 2]$ mm (discussed in Section 3.2.3). We considered $f_{syn} \in [8,\ 1000]$, where both the bounds have been set heuristically based on observations of the results.

		$\mathbf{b_{lo}}$	$\mathbf{b_{up}}$
Synaptic dendritic factor	f_{syn}	8	1000
Postsynaptic excitatory potential rate constant	γ_{ee}	$100s^{-1}$	$1000s^{-1}$
Mean Nernst membrane potential	h_{ee}^{eq}	-20 mV	10 mV
Passive membrane decay time constant	τ_e	5 ms	150 ms
Dendritic length constant	λ	0.1 mm	2 mm
Mean resting membrane potential	h_e^r	-80 mV	-60 mV

Table 5.1: Range of variability of the parameters of the *search-spaces* for the PSO.

The swarm has a total of $M = 40$ **particles** and operates $N = 500$ **iterations**. Thus, legal positions for the particles in an *m-dimensional* parameter space, are *m-dimensional* vectors of the form

$$\mathbf{x} = (x_1, ...x_m)\,, \text{where } x_j \in [b_{lo,j}, b_{up,j}]\,, \forall j \in \{1, ..., m\}.$$

[3]See Section 4.5 for more details about the parameters and their impact on the alpha rhythm.

The **initial condition** for the position of the particles is a matrix $\mathbf{X_{init}}$ having dimension M x m. The i-th row of the matrix contains the initial spatial coordinates of the i-th particles in the search-space S. They are uniformly randomly distributed in $[\mathbf{b_{lo}}, \mathbf{b_{up}}]$.

$$\mathbf{X_{init}} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1j} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2j} & \dots & x_{2m} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ x_{i1} & x_{i2} & \dots & x_{ij} & \dots & x_{im} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ x_{M1} & x_{M2} & \dots & x_{Mj} & \dots & x_{Mm} \end{bmatrix}. \tag{5.10}$$

The **velocities** are initialized to zero, while the local and global best are initialized according to (5.2) and (5.3). We used the the classic version of the PSO Algorithm, with a communication structure based on a **global topology**. **Positions and velocities** are updates according to formulas (5.4) and (5.6) with a constriction factor (5.7) to ensure convergence.

5.4.2 Fitness functions

We implemented two *fitness functions* to evaluate the similarity of the analysed PSDs with PSDs encountered in humans:

1. The function f_1 consists of different tests for the Jacobian prediction. Firstly, the singular point solution should lead to excitatory and inhibitory firing rates S between 0.1 and 20 s^{-1}. The other tests evaluate the shape of the linear eigenspectrum, based on the power in the δ (0-4 Hz), θ (4-7.5 Hz), α (7.5-13 Hz), and β (13-30 Hz) bands, and on maximum and minimum frequencies, f_{max} and f_{min}, in each range. These criteria require that:
   - 15-50% of total power is in δ band
   - 10-25% of total power is in θ band
   - 15-40% of total power is in α band
   - 15-40% of total power is in β band

- power ratio $\theta/\delta < 0.6$
- power ratio $\theta/\alpha < 0.7$
- $12\ Hz < \mathrm{SEF}_{90} < 21\ Hz$, where SEF_{90} is defined as the frequency below which 90% of power resides
- $1/3\ f^{\delta}_{max} < f^{\alpha}_{min} < 5\ f^{\delta}_{max}$
- $f^{\theta}_{min} < 1/2\ f^{\alpha}_{max}$ or $f^{\theta}_{min} < 1/2\ f^{\theta}_{max}$
- $f^{\theta}_{min} < 0.9\ f^{\delta}_{min}$

A *cost* is opportunely associated to each of these criteria. The cost is zero if the criterium is satisfied, or positive if the spectrum does not meet the specific criterium. More in detail, for each criterium an opportune interval $[A_1, A_2]$ is defined including the allowed regions of zero cost $[B_1, B_2]$. For example, for the first criterium $[B_1, B_2] = [15\%, 40\%] \subset [A_1, A_2] = [0\%, 100\%]$. In between the extrema $[A_1, B_1]$ and $[B_2, A_2]$ the cost increases linearly, with a steepness varying depending on heuristic observation of the results. The fitness evaluation is the sum of all the costs, and the swarm algorithm looks for minimum evaluations of f_1.

2. The fitness function f_2 is a variation of the f_1. We found that, in some observed cases (but not in all cases), it performs better than the fitness f_1. The test for the firing rates S is the same as in f_1. The last 4 criteria of f_1 give an accurate evaluation of the spectrum shape when the maximum peak of the spectrum (after δ region) occurs at a frequency f_{max} in the α band. We then introduced another criterium requiring f_{max} in the α band. Some requirements for the total power in the bands have been also modified. All together, the criteria of f_2 are:

   - 15-50% of total power is in δ band
   - 8-25% of total power is in θ band
   - 30-50% of total power is in α band
   - 5-25% of total power is in β band
   - power ratio $\theta/\delta < 0.6$

- power ratio $\theta/\alpha < 0.7$
- 12 Hz < SEF_{90} < 21 Hz, where SEF_{90} is defined as the frequency below which 90% of power resides
- If $f_{max} \in \alpha$ band
  - 1/3 f^{δ}_{max} < f^{α}_{min} < 5 f^{δ}_{max}
  - f^{θ}_{min} < 1/2 f^{α}_{max} or f^{θ}_{min} < 1/2 f^{θ}_{max}
  - f^{θ}_{min} < 0.9 f^{δ}_{min}
- If $f_{max} \in \theta$ band or If $f_{max} \in \beta$ band, we require
  - 7.5 Hz = $\theta_{max} < f_{max} < \beta_{min}$ = 30 Hz

In this chapter we mainly used the fitness function f_1 (except for results shown in Fig. 5.5), while further PSO simulations in search space S_2 using the fitness function f_2 will be presented in Section 6.8. The implemented fitness functions are reported in the Appendix.

5.4.3 Results

After testing formulas (5.5) and (5.6) for the velocity update, we found the latter performs better with the 3D system, ensuring the existence of best solutions in most proximal locations for the synaptic input. We also tried different social and cognitive parameters (as $w_{soc} = 1.5$ and $w_{cog} = 1.5$), finding that the values $w_{soc} = 1.3$ and $w_{cog} = 2.8$ used in [36] work better in all the analysed cases.

Best solutions for 3D system with $\lambda = 0.9$ mm

Figures 5.1 and 5.2 show the best spectra found running a set of PSO simulations in the search space $\mathbf{S_1}$ for 3D configurations with electrotonic lentgh constant $\lambda = 0.9$ mm. The corresponding best positions and fitness evaluations are illustrated in Tables 5.2 and 5.3.

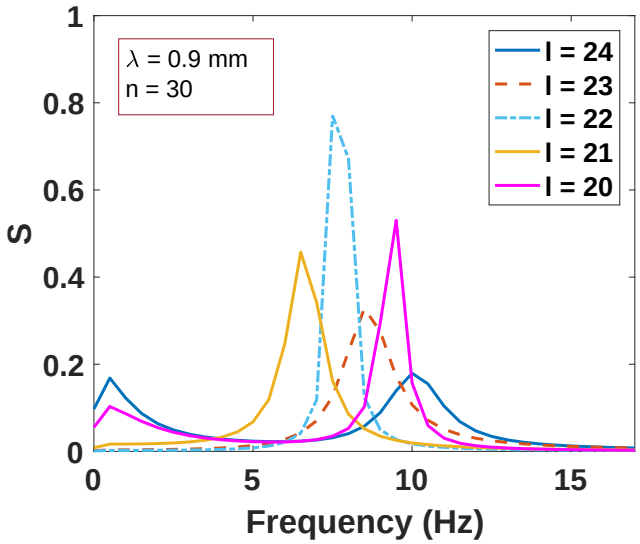

Figure 5.1: PSDs from PSO solutions (simulation A) in the search-space S_1 with fitness function f_1, for different synaptic locations l.

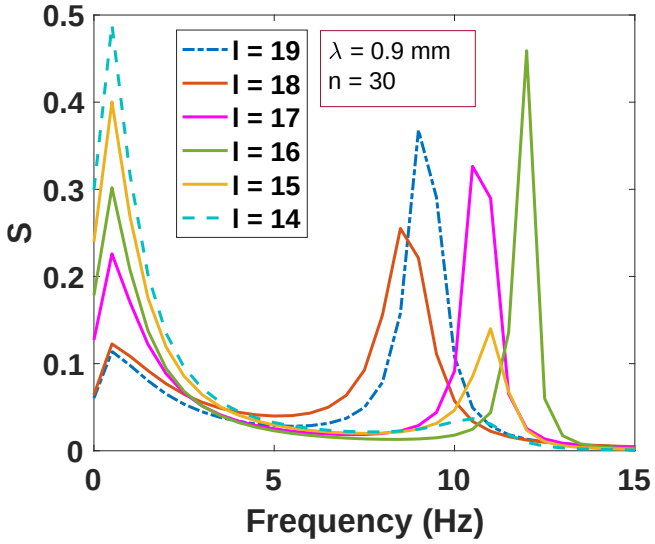

Figure 5.2: PSDs from PSO solutions (simulation B) in the search-space S_1 with fitness function f_1, for different synaptic locations l.

Table 5.2: Best positions from PSO simulation A.

Input location l	best x h_{ee}^{eq}	best x γ_{ee}	best x f_{syn}	fitness evaluation
24	−18.038	291.5	9	4.034
23	−20	856.22	871.573	42.91
22	−20	725.243	1,000	49.061
21	−12.065	620.807	300	48.451
20	−18.038	291.5	48	11.155

Table 5.3: Best positions from PSO simulation B.

Input location l	best x h_{ee}^{eq}	best x γ_{ee}	best x f_{syn}	fitness evaluation
19	−19.975	281.041	100	8.588
18	−19.975	281.041	680.34	6.753
17	−10	200	219.912	8.969
16	−7.982	150.631	753.682	8.955
15	9.967	157.471	917.758	9.927
14	10	127.591	1,000	22.829

As it can be noted from the graphs, optimal spectra are obtained for proximal and distal region up to compartment 15 (corresponding to a distance from the soma of 1.2 mm). Compared to proximal input configurations of Fig. 5.1, distal inputs

induce higher frequency of the alpha peak (Fig. 5.2). We found that for dendritic distances larger than 1.2 mm (compartments 14 to 1) the PSO algorithm does not find any parameter set generating alphoid dynamics.

Best solutions for 3D system with $\lambda = 1.6$ mm

Our second set of PSO simulations has been carried out for configurations with electrotonic length constant $\lambda = 1.6$ mm. Best spectra are shown in Figs. 5.3 and 5.4 and the corresponding best positions and fitness evaluation are reported in Tables 5.4 and 5.5. As for the first set of simulations, the frequency of the alpha peak is larger for more distal input locations. Good spectra are obtained up to compartment 9 (2 mm from the soma). Despite the low fitness evaluation for input locations $l = 23$ and $l = 24$ (see Table 5.5), the shape of the corresponding PSDs (Fig. 5.4) is not similar to typical PSDs encountered in humans [36]. Moreover, although configuration $l = 13$ has a "better" PSD, the corresponding fitness evaluation is larger than those of configurations $l = 23$ and $l = 24$. To deal with this problem, we implemented the fitness function f_2 (Section 5.4.2). Results for these 3 configurations are shown in Fig. 5.5 and Table 5.6. As mentioned before, however, the fitness function f_2 does not perform better than f_1 for all configurations.

Computational time

The simulations have been carried out on the CINN's cluster of the Department of Psychology at University of Reading. In a 3-dimensional search space, each PSO simulation for a specific input configuration takes on average 5.6×10^4 s.

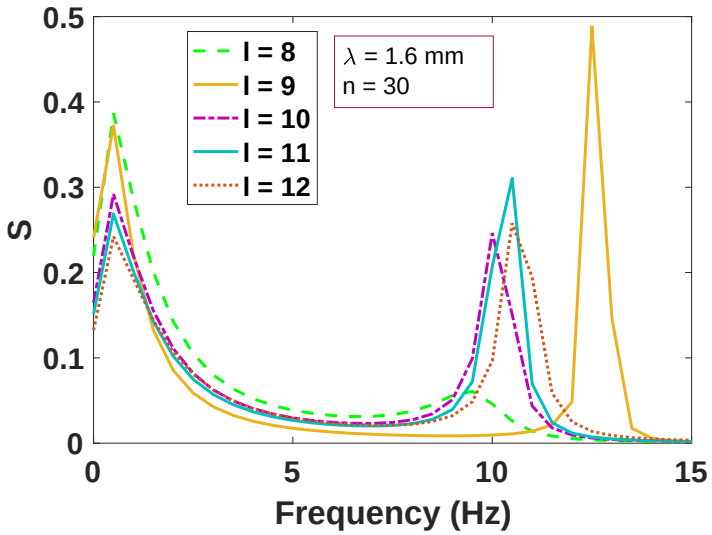

Figure 5.3: PSDs from PSO solutions (simulation C) in the search-space S_1 with fitness function f_1, for different synaptic locations l.

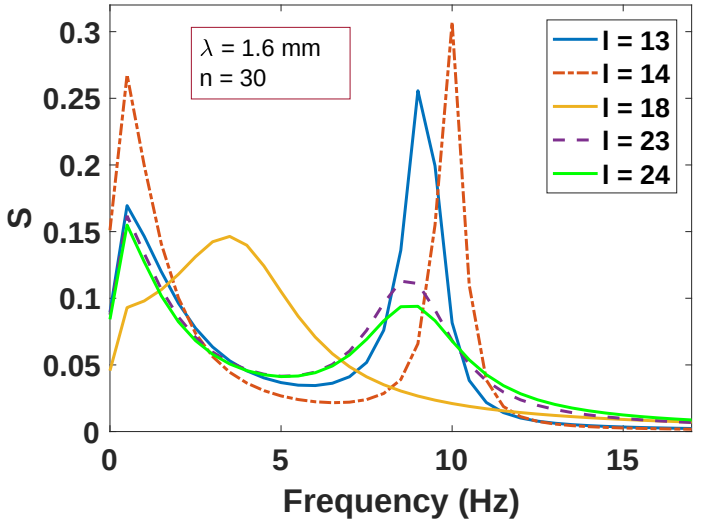

Figure 5.4: PSDs from PSO solutions (simulation D) in the search-space S_1 with fitness function f_1, for different synaptic locations l.

Table 5.4: Best positions from PSO simulation C.

Input location l	best x h_{ee}^{eq}	γ_{ee}	f_{syn}	fitness evaluation
8	9.824	203.41	995.636	16.457
9	7.943	170.895	594.307	10.657
10	10	230.514	697.388	8.322
11	2.149	233.416	323.43	8.525
12	-20	214.895	951.747	7.137

Table 5.5: Best positions from PSO simulation D.

Input location l	best x h_{ee}^{eq}	γ_{ee}	f_{syn}	fitness evaluation
13	-16.442	275.309	468.525	7.182
14	-2.737	265.026	85.386	8.373
18	-19.139	1,000	489.761	20.621
23	-18.103	334.298	20.544	3.261
24	-20	331.454	17.012	2.005

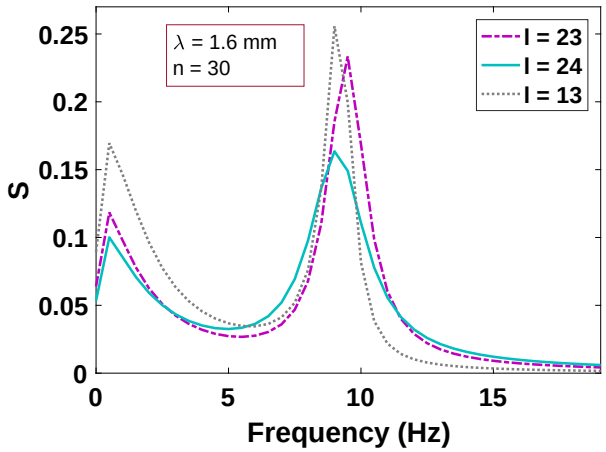

Figure 5.5: PSDs from PSO solutions (simulation E) in the search-space S_1 with fitness function f_2, for different synaptic locations l.

Table 5.6: Best positions from PSO simulation E.

Input location l	best x h_{ee}^{eq}	γ_{ee}	f_{syn}	fitness evaluation
23	-19.65	331.096	22.478	0
24	-18.188	352.587	20.771	$3.23 \cdot 10^{-2}$
13	-16.442	275.309	468.525	2.6

5.5 Discussion

Although the PSO technique has been already used in [36] for 2D NFM to find optimal parameters reproducing the alpha rhythm, in the present work it is applied

for the first time to a 3D NF model. Dendritic parameters, namely the synaptic factor and the dendritic length constant, not considered in [36], have been included in the search space. Results suggest that 3D NF models can be used to explore the role of the dendrite as filter for synaptic inputs.

Dendritic "democracy" and synaptic factor

Several morphological and physiological properties are diversified across the dendritic tree in order to maximise the current transfer to the soma [243, 244] and generally optimise neuronal processing. Examples are the progressive tapering of the dendrites and protrusions and structural specializations frequently located at the end of the dendrites [30], referred to as *synaptic specializations*. Other physiological properties involve the distribution of ionic channels and currents along the membrane [245–247], the filtering of synaptic inputs and the variation of the dendritic length constant [163].

An important mechanism depending on the dendritic distance from the cell body is the normalisation of the efficacy of synaptic inputs at the soma. This is often referred to as "dendritic democracy" [31]. Due to the filtering behaviour of the dendrite, somatic responses to inputs generated in the distal domains are much more attenuated than those generated by more proximal synapses [150]. Recent studies indicate that the democracy can be restored by a variety of mechanisms compensating for the filtering in the synaptic input [31]. One of the possibilities is the scaling of the synaptic strength (i.e. the synaptic conductance) with the distance from the soma. In the present work, this corresponds to increasing the synaptic factor f_{syn} of equation (3.34). Our results suggest that a synaptic factor is necessary to reproduce the alpha rhythmicity encountered in human electroencephalogram recordings. According to the basic idea of "democracy", this factor tends to be higher when the input arrives from synapses located in distal dendritic domains. Another solution for the synaptic filtering is the synchrony with other distal inputs or with input in different dendritic domains. The "democratic" role of both the synaptic factor and the simultaneous synaptic activation will be more extensively discussed in the next chapter.

Our results suggest that higher values of the length constant are more likely to give rise to alpha rhythmicity when synaptic inputs are located in more distal regions (up to 2 mm from the soma corresponding to compartment 9; Figs. 5.3 and 5.4, Tables 5.4 and 5.5), while lower values of the length constant enhance the alphoid dynamics in systems with synaptic inputs active in the proximal regions (up to 1.2 mm from the soma corresponding to compartment 15; Figs. 5.1 and 5.2, Tables 5.2 and 5.3)). This phenomenon and its implication on dendritic democracy will be discussed in the next chapter.

Chapter 6

Analysis of human spectra in a 3D Neural Field Model

6.1 Introduction

The extended 3D Neural Field Model including the dendritic dimension presents new and interesting scientific challenges, involving the dynamics analysis and the use of complex methods to explore biologically relevant results. Facing and solving a wide variety of numerical problems has led to a constant upgrade of the computational model. Simultaneously, many simulations have been carried out to examine the role of biological parameters in the dynamics. A constant reflection on both biological and numerical matters has oriented the direction of the research and established new targets to be explored, also resulting in a progressive improvement of the model.

The main novelty successfully introduced in the 3D extended model is the synaptic strengthening factor f_{syn}, indispensable to compensate the loss of signal along the dendrite and reproduce Power Spectra of Density (PSD) typical of humans, i.e. exhibiting a peak in the *alpha* band of frequencies. With the usage of sophisticated algorithms (such as the PSO algorithm illustrated in Chapter 5), some regions in the parameter space have been detected where the PSD exhibits a peak in the *alpha* band of frequencies. The algorithm searches for optimal PSDs, predicted from the

Jacobian of linearised system evaluated at a stable point.

However, the regions for "good" spectra in the parameter space are not common and their occurrence seems to be predominant for high values of the synaptic strengthening factor.

In this chapter, the most important properties of the single microcolumn 3D system are illustrated. Firstly, the role of the synaptic factor f_{syn} on the model dynamics is explored when a single pulse input is applied to systems with different number of compartments, confirming the behaviour qualitatively illustrated in Section 3.2.5. The response of the system to single pulse input is also studied for different values of dendritic length constant λ (in systems with a fixed number of compartments). Results are consistent with the physiological description of the dendritic length constant (provided in Section 3.2.3). Secondly, the relationship occurring between the stable points (calculated with the procedure illustrated in Section 3.2.9) and the linear spectra is illustrated, for some particular configurations (i.e. for some specific values of λ and f_{syn}). The results will be confirmed in the study conducted in Section 6.6, where the dynamics is systematically studied for all possible values of λ and f_{syn}, for each possible location l of the synaptic input. Finally, the findings illustrated in Section 6.6 will be further enhanced using multiple inputs configurations and optimal variation of some parameters, through the PSO technique with fitness function f_2.

6.2 Role of the synaptic factor in single pulse responses

Most of my results confirm that, in all the 3D systems observed, a factor strengthening the synaptic input is required to reproduce an alpha rhythm in the PSD. The strengthening factor is the most important novelty introduced with the 3D extension of the pre-existent 2D Neural Field Model [36], and its crucial role on the dynamics is here studied by looking at the responses of systems, with different number of compartments, with or without this factor. As qualitatively described in Section

3.2.5, the synaptic factor is strictly related to the extension of membrane surface where the synaptic input is located. There are two important issues that need to be considered here:

1. The synaptic input that in the 2D configuration was applied to the entire "neuron", in the 3D geometry it is located in one (or more) specific dendritic site, and then spreads in both directions in the entire cable. It is then reasonable to expect a lower response at the injection site of the 3D system compared to the 2D response. If we want to elicit the same response in the 2D system and *at the injection site* of the 3D system, we can then consider to strengthen the synaptic input with a factor f_{syn}.

2. If we increase the number of dendritic compartments (from n to m, with $m > n$), since the length of the cable is fixed ($L = 4$ mm), the dimension of the actual compartment is reduced. As qualitatively explained in Section 3.2.5, less current will flow in the system if the compartments are smaller. We will now prove that we can correct this effect by simply multiplying the synaptic input by a factor $f_{syn} = m/n$.

Point 1. will be explored (section 6.3.1) by analysing in detail the dynamics of the membrane potential in the two geometries 2D and 3D with number of compartments $n = 17$ and $n = 31$, when a single pulse synaptic input is applied. We will prove that 3D response is reduced with respect to the 2D response, and that the different 3D responses with $n = 17$ and $n = 31$ are due to the different size of the compartment (i.e. different n). The linear increase of the factor with the number of compartments (Point 2.) will be clearly proved in Sections 6.3.2 and 6.3.3.

It must be noted that the problem of finding the "right" factor when moving from the 2D to the 3D geometry (with a specific number of compartments) is not trivial. First of all, we want to clarify that we considered, as "right" synaptic factor, the one that is leading to the presence of an α peak in the PSD. Secondly, we point out that, rather than comparing the 2D response with the 3D response at the injection site, we should compare the 2D response with the entire dynamics of the dendrite, with particular attention to the soma (since we calculate the PSD for the somatic

h_e). Finally and as a consequence, the factor would depend on the location of the input and its distance from the soma. Even in the simplest case where the input is located at the soma, we found that the right factor is not always the one that makes the amplitude of the local 3D response similar to the amplitude in the 2D response.

In Section 6.3.4 it will be shown that once we know (from the swarm simulation) the right factor f_{syn}^A for a system A with n_A compartments leading to a PSD with an α peak, the same spectrum can be obtained in a k n compartment configuration B when considering a factor $f_{syn}^B = k\ f_{syn}^A$. The same effect is achieved by considering k synaptic inputs with factors $f_{syn}^B = f_{syn}^A$.

6.3 Single pulse

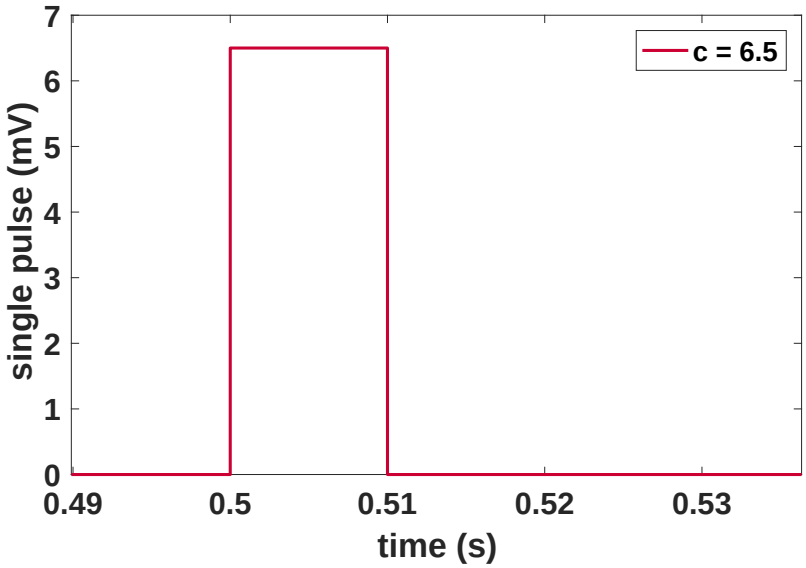

Figure 6.1: Single pulse input

We studied the dynamics of the membrane potential when a single pulse input is provided to the system, and all the connections (*local, long-range* and *extra-cortical*) are switched off. This means that the firing rate function (3.58) describing the *local*

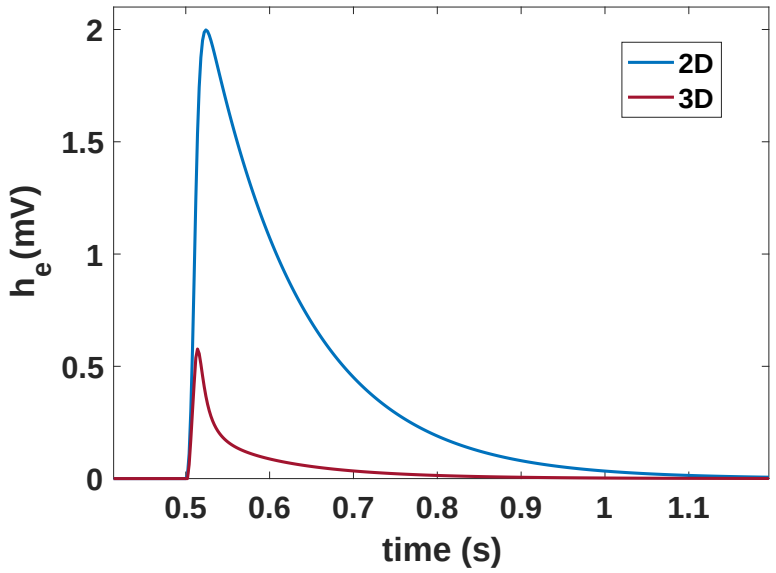

Figure 6.2: Somatic h_e responses to a single pulse input (6.2) applied at the soma, in 2D and 3D geometries. The excitatory dendritic cable, in the 3D system, has 30 compartments. The dendritic length constant is $\lambda = 0.9$ mm.

activity in the Eqs. for the synaptic currents (3.53-3.56), is

$$S_e = 0, \tag{6.1}$$

leading to *long-range* connections[1] $\Phi_{ek} = 0$. The white noise component of p_{ee} describing the *extra-cortical* activity is zero. The single pulse input is implemented through the "extra-cortical" connection p_{ee}, here representing an external input, modelled as a step function:

$$p_{ee} = c * \bar{p}_{ee} \; for \; 0.5s \leq t \leq 0.51s \tag{6.2}$$

$$p_{ee} = c * \bar{p}_{ee} \; otherwise \tag{6.3}$$

where, c is a constant. A typical pulse input is illustrated in Fig. 6.1, for $c = 6.5$.

[1]Eq. 3.56 with $S_e = 0$ means that Φ_{ek} has no spatial and temporal variation. As a consequence $\Phi_{ek}(t) = 0$, since initial conditions are $\Phi_{ek}(0) = 0$. Note that, the same result is obtained when $v = 0$ (i.e. when there is no spatial coupling in the system).

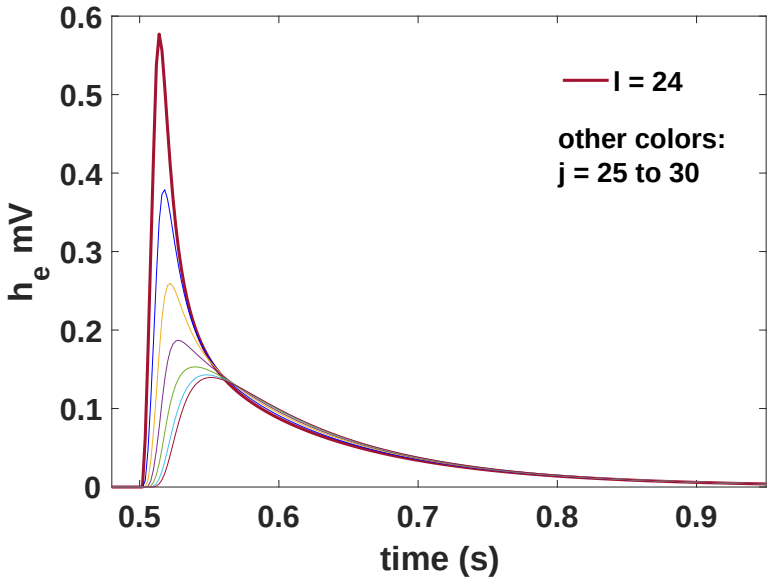

Figure 6.3: h_e dendritic basal (compartments 25 to 30) responses to single pulse input (6.2) applied at the soma (compartment 24). The excitatory dendritic cable has 30 compartments. The dendritic length constant is $\lambda = 0.9$ mm.

Equations for the synaptic currents:

$$
\begin{aligned}
\frac{\partial}{\partial t} I_{ee} &= \widetilde{I}_{ee}(t) \\
\frac{\partial}{\partial t} \widetilde{I}_{ee}(t) &= -2\gamma_{ee}\widetilde{I}_{ee} - \gamma_{ee}^2 I_{ee} + e\gamma_{ee}\Gamma_{ee}\left(N_{ee}^{\beta} S_e(h_e) + \Phi_{ee} + p_{ee}(t)\right) \\
\frac{\partial}{\partial t} I_{ei} &= \widetilde{I}_{ei}(t) \\
\frac{\partial}{\partial t} \widetilde{I}_{ei}(t) &= -2\gamma_{ei}\widetilde{I}_{ei} - \gamma_{ei}^2 I_{ee} + e\gamma_{ei}\Gamma_{ei}\left(N_{ei}^{\beta} S_e(h_e) + \Phi_{ei} + p_{ei}(t)\right) \\
\frac{\partial}{\partial t} I_{ie} &= \widetilde{I}_{ie}(t) \\
\frac{\partial}{\partial t} \widetilde{I}_{ie}(t) &= -2\gamma_{ie}\widetilde{I}_{ie} - \gamma_{ie}^2 I_{ie} + e\gamma_{ie}\Gamma_{ie}\ N_{ie}^{\beta} S_i(h_i) \\
\frac{\partial}{\partial t} I_{ii} &= \widetilde{I}_{ii}(t) \\
\frac{\partial}{\partial t} \widetilde{I}_{ii}(t) &= -2\gamma_{ii}\widetilde{I}_{ii} - \gamma_{ii}^2 I_{ii} + e\gamma_{ii}\Gamma_{ii}\ N_{ii}^{\beta} S_i(h_i)
\end{aligned}
\tag{6.4}
$$

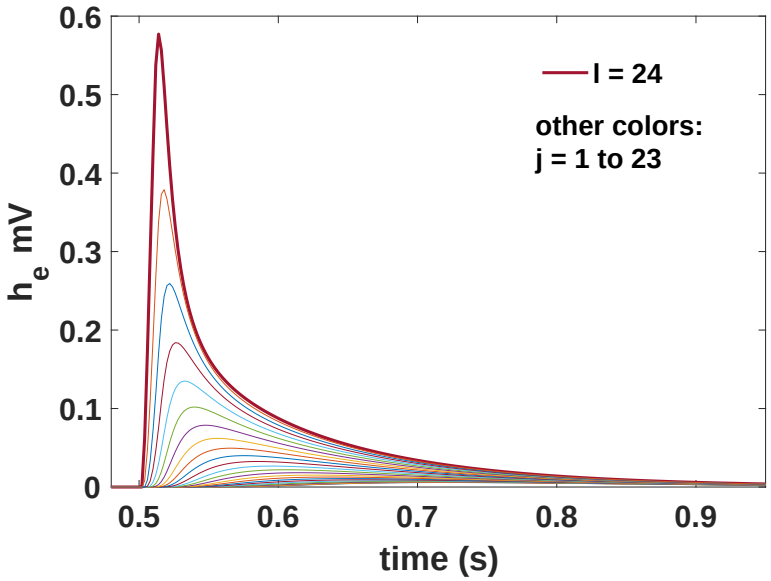

Figure 6.4: h_e dendritic apical (compartments 1 to 23) responses to single pulse input (6.2) applied at the soma (compartment 24). The excitatory dendritic cable has 30 compartments. The dendritic length constant is $\lambda = 0.9$ mm.

then become

$$
\begin{aligned}
\frac{\partial}{\partial t} I_{ee} &= \widetilde{I}_{ee}(t) \\
\frac{\partial}{\partial t} \widetilde{I}_{ee}(t) &= -2\gamma_{ee}\widetilde{I}_{ee} - \gamma_{ee}^2 I_{ee} + e\gamma_{ee}\Gamma_{ee} N_{ee}^{\beta}\ p_{ee}(t) \\
\frac{\partial}{\partial t} I_{ei} &= \widetilde{I}_{ei}(t) \\
\frac{\partial}{\partial t} \widetilde{I}_{ei}(t) &= -2\gamma_{ei}\widetilde{I}_{ei} - \gamma_{ei}^2 I_{ee} + e\gamma_{ei}\Gamma_{ei} N_{ei}^{\beta}\ p_{ei}(t) \\
\frac{\partial}{\partial t} I_{ie} &= \widetilde{I}_{ie}(t) \\
\frac{\partial}{\partial t} \widetilde{I}_{ie}(t) &= -2\gamma_{ie}\widetilde{I}_{ie} - \gamma_{ie}^2 I_{ie} \\
\frac{\partial}{\partial t} I_{ii} &= \widetilde{I}_{ii}(t) \\
\frac{\partial}{\partial t} \widetilde{I}_{ii}(t) &= -2\gamma_{ii}\widetilde{I}_{ii} - \gamma_{ii}^2 I_{ii}
\end{aligned}
\tag{6.5}
$$

Initial conditions for all the variables, except the membrane potentials, are set to zero. As a consequence, Eqs. (6.5) can be written as:

$$\begin{aligned}
\frac{\partial}{\partial t} I_{ee} &= \widetilde{I}_{ee}(t) \\
\frac{\partial}{\partial t} \widetilde{I}_{ee}(t) &= -2\gamma_{ee}\widetilde{I}_{ee} - \gamma_{ee}^2 I_{ee} + e\gamma_{ee}\Gamma_{ee} N_{ee}^{\beta}\ p_{ee}(t) \\
\frac{\partial}{\partial t} I_{ei} &= \widetilde{I}_{ei}(t) \\
\frac{\partial}{\partial t} \widetilde{I}_{ei}(t) &= -2\gamma_{ei}\widetilde{I}_{ei} - \gamma_{ei}^2 I_{ee} + e\gamma_{ei}\Gamma_{ei} N_{ei}^{\beta}\ p_{ei}(t) \\
\frac{\partial}{\partial t} I_{ie} &= 0 \\
\frac{\partial}{\partial t} \widetilde{I}_{ie}(t) &= 0 \\
\frac{\partial}{\partial t} I_{ii} &= 0 \\
\frac{\partial}{\partial t} \widetilde{I}_{ii}(t) &= 0
\end{aligned} \tag{6.6}$$

It is useful here to note how p_{ee} contributes to the "synaptic input" I_{ee} (it is the only contribution in the equation for $\widetilde{I}_{ee}$ in (6.5)-(6.4), since $\Phi_{ee} = 0$ and $S_e = 0$) and, therefore, to the membrane h_e at the injection site:

$$\frac{\partial h_e}{\partial t}(t) = \frac{1}{\tau_e}\left\{ -[h_e(t) - h_e^r] + \lambda^2 \frac{\partial^2 h_e}{\partial z^2}(t) + \psi_{ee}(h_e) I_{ee}(t) \right\} \tag{6.7}$$

Figure 6.2 shows the somatic response to a single pulse input of the form (6.2) in the 2D and standard 3D systems (with 30 dendritic compartments and soma located at compartment 24), with the input applied at the soma (*control case*). The depolarizations h_j in other compartments (25 to 30, and 1 to 23) are shown in Figs. 6.3 and 6.4.

6.3.1 Dynamics with different number of compartments

In Section (6.2) we have seen that 2D systems and 3D systems exhibit different responses to single pulse inputs of the form (6.2). Here, illustrate in detail the

dynamics of the systems, to explicitly show how the 3D geometry and the number n of dendritic subsections modify the equations for h_e to produce lower responses in the 3D geometry. Two 3D geometries will be considered, with 17 and 31 compartments.

For the purpose of clarity, the equations for synaptic currents and for the excitatory membrane potential at the injection site at reported here, for both the 2D and 3D geometries. As explained in previous section, the pulse input (6.2) consists of a step current injected through the excitatory synaptic current I_{ee}, while the inhibitory current I_{ie} is set to zero, as well as the the *local* and *long-range* connections ($S_{lk} = 0$ and $\Phi_{ek} = 0$). Equations (3.55) for the excitatory synaptic current

$$\begin{aligned} \frac{\partial}{\partial t} I_{ee} &= \widetilde{I}_{ee}(t) \\ \frac{\partial}{\partial t} \widetilde{I}_{ee}(t) &= -2\gamma_{ee}\widetilde{I}_{ee} - \gamma_{ee}^2 I_{ee} + e\gamma_{ee}\Gamma_{ee}\left(N_{ee}^{\beta} S_e(h_e) + \Phi_{ee} + p_{ee}(t)\right) \end{aligned} \tag{6.8}$$

then become

$$\begin{aligned} \frac{\partial}{\partial t} I_{ee} &= \widetilde{I}_{ee}(t) \\ \frac{\partial}{\partial t} \widetilde{I}_{ee}(t) &= -2\gamma_{ee}\widetilde{I}_{ee} - \gamma_{ee}^2 I_{ee} + e\gamma_{ee}\Gamma_{ee} * p(t) \end{aligned} \tag{6.9}$$

where $p(t)$ is the pulse injection of the form (6.2).
The 2D and 3D Eqs. (3.1) and (3.38)

$$\begin{aligned} \tau_e \frac{\partial h_e}{\partial t}(\mathbf{x}, t) &= -[h_e(\mathbf{x}, t) - h_e^r] + \sum_l \psi_{le}(h_e) I_{le}(\mathbf{x}, t) \\ \tau_e \frac{\partial h_e}{\partial t}(\mathbf{x}, t) &= -[h_e(x, t) - h_e^r] + \lambda^2 \frac{\partial^2 h_e}{\partial z^2}(z, t) + \sum_l \psi_l(h_e) I_l(z, t) \end{aligned} \tag{6.10}$$

simply become[2]:

$$\frac{\partial h_e}{\partial t}(t) = \frac{1}{\tau_e}\left\{ -[h_e(t) - h_e^r] + \psi_{ee}(h_e) I_{ee}(t) \right\} \tag{6.11}$$

[2]Please note that, in Eq. (3.1), the dependency on the position of the neural population $\mathbf{x}$ on the grid has been removed, since only one population (i.e. one point of the 2 dimensional grid) is considered here; also, Eq. (3.38) does not depend on the dendritic coordinate z, given that the equation describes the potential of a single compartment (where z is fixed).

$$\frac{\partial h}{\partial t}(t) = \frac{1}{\tau_e}\left\{-[h_e(t) - h_e^r] + \lambda^2 \frac{\partial^2 h_e}{\partial z^2}(t) + \psi_{ee}(h_e) I_{ee}(t)\right\} \tag{6.12}$$

Rearranging these equations, using the *finite difference approximation*

$$\frac{\partial^2 h_j}{\partial z^2} \simeq \frac{h_{j-1} - 2h_j + h_{j+1}}{dz^2}. \tag{6.13}$$

and defining the excitatory synaptic current as

$$I_{syn}^{ee}(t) = \frac{1}{\tau_e}\psi_{ee}(h_e) I_{ee}(t) \tag{6.14}$$

one has:

$$\frac{\partial h_e}{\partial t}(t) = -\frac{1}{\tau_e} h_e(t) + I_{syn}^{ee}(t) + c_e \tag{6.15}$$

$$\frac{\partial h_e}{\partial t}(t) = K_e\, h_{j-1}(t) - B_e\, h_j(t) + K_e\, h_{j+1}(t) + I_{syn}^{ee}(t) + c_e \tag{6.16}$$

where

$$c_e = \frac{h_e^r}{\tau_e}$$

$$K_e = \frac{\lambda^2}{\tau_e dz^2} \tag{6.17}$$

$$B_e = \left(\frac{1}{\tau_e} + \frac{2\lambda^2}{\tau_e dz^2}\right) = \left(\frac{1}{\tau_e} + 2K_e\right) \tag{6.18}$$

and j indicates the compartment of the injection site ($j-1$ and $j+1$ are adjacent compartments).

We want to remind the reader that Δz is defined as

$$\Delta z = \frac{l}{n} \tag{6.19}$$

where l is the dendritic length (4 mm) and n is the number of compartments used in the discretization of the neural cable. It is useful to note the quadratic dependence of both B_e and K_e on the number of compartments:

$$K_e = \frac{\lambda^2\ n^2}{\tau_e\ l^2} \tag{6.20}$$

$$B_e = \left(\frac{1}{\tau_e} + \frac{2\lambda^2\ n^2}{\tau_e\ l^2}\right) \tag{6.21}$$

The dynamic of both Eqs. (6.15) and (6.16) are illustrated below, varying the number n of compartments[3] and evaluating the impact of each term contained in the ODEs on the depolarization or hyperpolarization of the membrane. The values of n considered in this study are $n = 1$ (2D), $n = 17$ and $n = 31$. The injection site is always the central compartment, respectively $j = 9$ and $j = 16$. The local responses in the three cases are illustrated in Fig. 6.5.

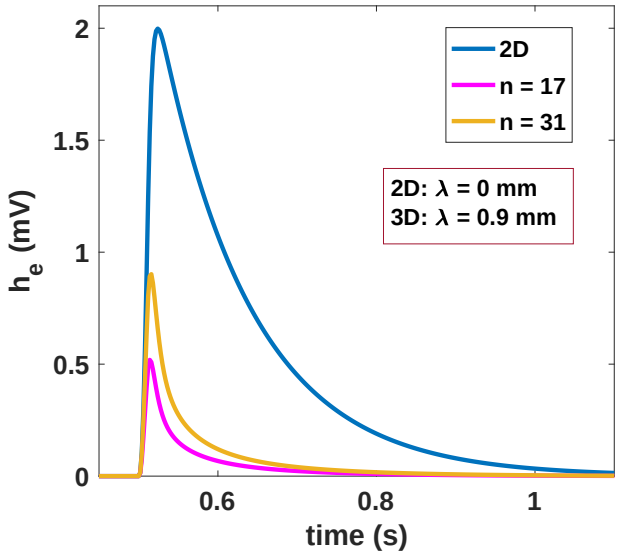

Figure 6.5: Local responses to single pulse stimulus, for different number of compartments n.

When analysing the differential Eq. (6.15) one should observe that the synaptic current is the force that is driving the depolarization of h_e, while the negative term $-\frac{h_e(t)}{\tau_e}$ is pushing the potential back towards negative values (resting membrane potential is typically around -60 mV). The evolution of these driving forces over time

[3] As it is easy to note, Eqs. (6.15) and (6.16) are identical when $\lambda = 0$.

is plot in Figs. 6.6 and 6.7, where, for clarity, the variables are plotted subtracting their resting values. The resulting force (i.e. the time evolution of Eq. (6.15),

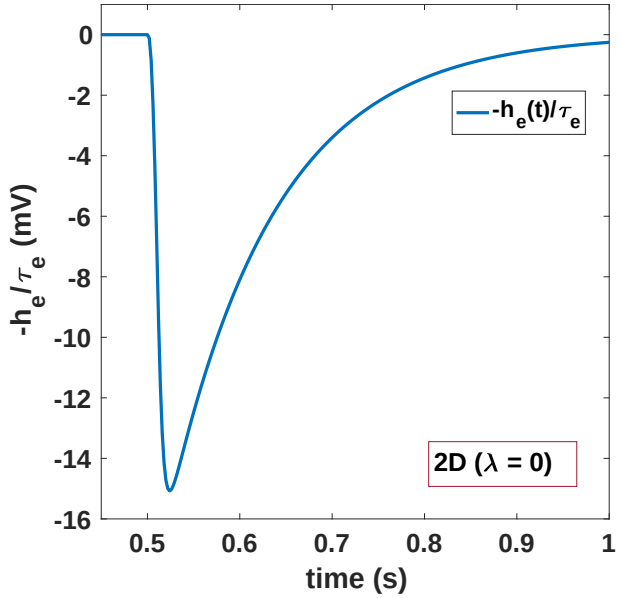

Figure 6.6: Time evolution of the hyperpolarising term appearing in (6.15) (the ODE for h_e in the 2D system).

including the constant term c_e), is plotted in Fig. 6.8. As it can be noted from the figure, the depolarising term corresponding to the synaptic excitatory current is predominant on the hyperpolarization, given the difference in amplitude of the two forces represented in Figs. 6.6 and 6.7. Figure 6.9 shows the same driving forces of Fig. 6.8 for the 3D cases corresponding to $n = 17$ and $n = 31$. The time derivatives of h_e have a lower amplitude when increasing the number of compartments n. This explains the difference in amplitude of the solutions of Fig. 6.5, while their sharpness and the extension over time are explained by the progressive shift of the derivatives towards the left in Fig. 6.9. The reason for this dynamic lies in the depolarizing and hyperpolarizing terms of Eq. (6.16) and in their dependence on the number of compartments n. While the synaptic current has a similar amplitude in the 2D and 3D cases (see Fig. 6.10 for comparison between 2D and 3D with 17 compartments), the hyperpolarising terms in (6.16) have an massive impact on the new dynamics. To be more precise, the difference in between the depolarising "K_e terms" and the

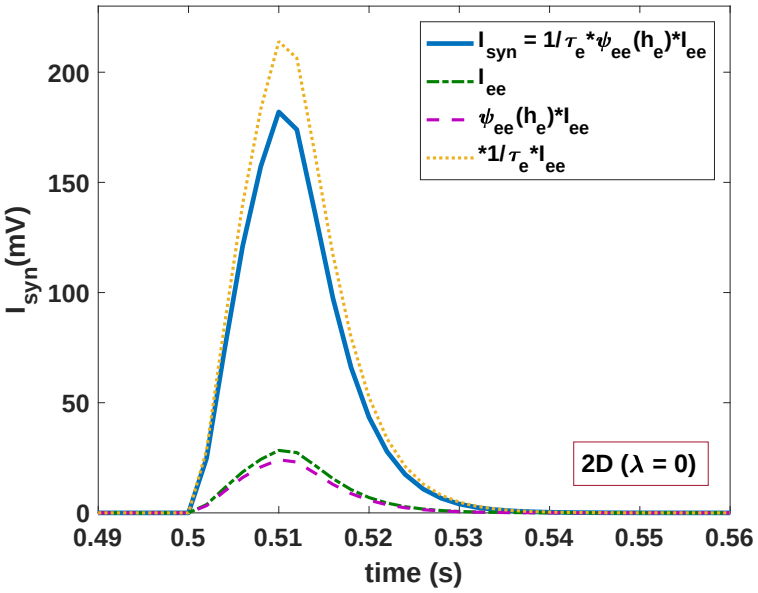

Figure 6.7: Time evolution of the depolarising term appearing in (6.15) (the ode for h_e in the 2D system).

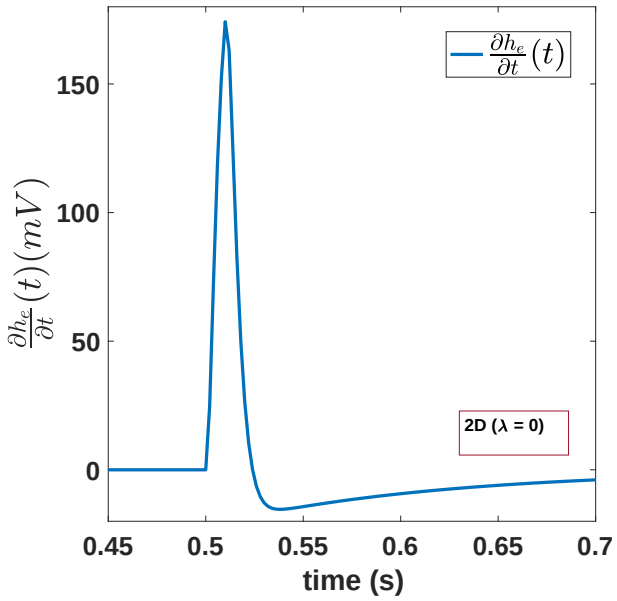

Figure 6.8: Time evolution of Eq. (6.15) (ode for h_e in the 2D system).

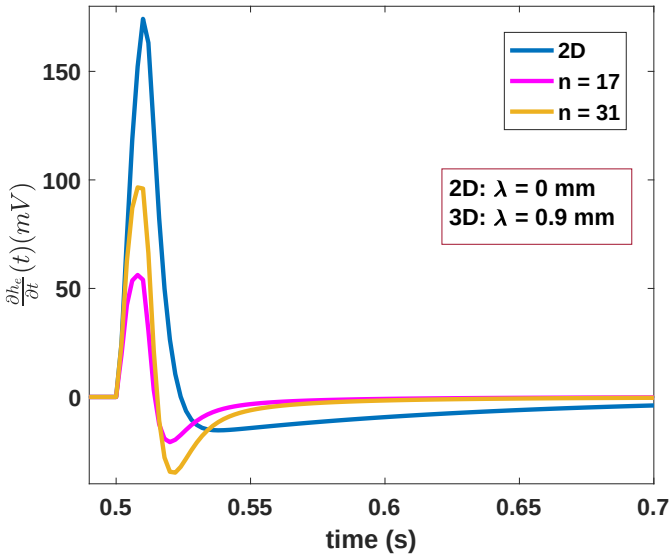

Figure 6.9: Time evolution of ODEs Eqs. (6.15) and (6.16) for different number of compartments n.

hyperpolarising "B_e term" result in a total hyperpolarization

$$K_e h_{j-1}(t) - B_e h_j(t) + K_e h_{j+1}(t) = n^2 \frac{\lambda^2}{\tau_e L^2}[h_{j-1}(t) - 2h_j(t) + h_{j+1}(t)] + \frac{1}{\tau_e} h_j(t) \quad (6.22)$$

that is higher than the 2D hyperpolarization. Results are shown in Fig. 6.11. For the 3D cases, the total hyperpolarization is comparable in size to the depolarization due to the synaptic current. The values for B_e and K_e in the three different systems are illustrated in table 6.1.

Overall, this explains that the reduced 3D response, compared to the 2D response, is due to an increased hyperpolarization term (6.22) in the equation for the excitatory membrane potential. This term quadratically depends on the number of compartments. As a consequence, systems with increasing number of compartments will exhibit lower response to single pulse inputs.

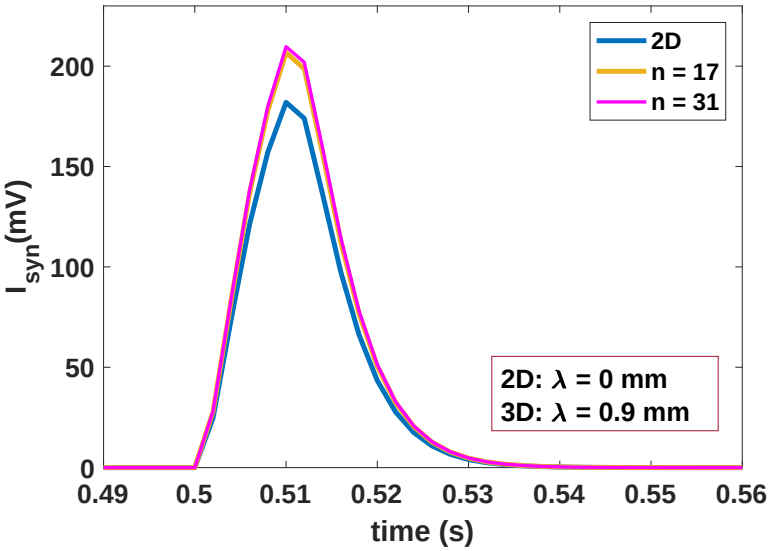

Figure 6.10: Time evolution of the depolarising terms (i.e. synaptic currents) appearing in (6.15) and (6.16) for $n = 17$ and $n = 31$.

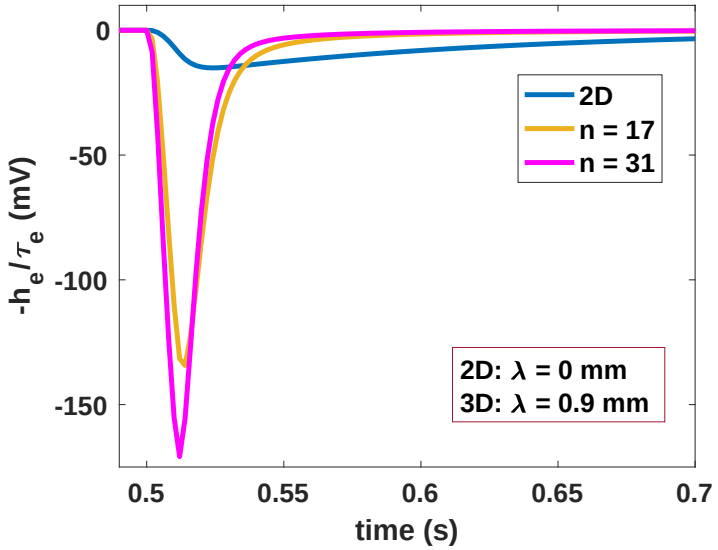

Figure 6.11: Time evolution of the hyperpolarising terms appearing in (6.15) and (6.16) for $n = 17$ and $n = 31$.

	2D	3D n = 17	3D n = 31
λ (mm)	0	0.9	0.9
n	1	17	31
n^2	1	289	961
Δz (mm)	0	0.0235	0.0129
K_e	0	110.378	367.036
B_e	-7.544	-228.301	-741.616
$1/\tau_e$ (ms^{-1})	7.544	7.544	7.544

Table 6.1: Cable equation constants of Eqs. (6.15) and (6.16) for $n = 17$ and $n = 31$.

6.3.2 Compensating the loss of signal

In this section, the dendritic response to single pulse input is explained, for systems with different number of dendritic segments and with different synaptic factors f_{syn}. For this analysis, the following configurations have been considered:

- System **A** has $n_A = 27$ compartments;
- System **B** has $n_B = 3\ n_A = 81$ compartments;
- The synaptic factor f_{syn} is 1 in both systems;
- The input is located in the central compartment, at a position $p = 0.2$ mm along the dendritic coordinate z, corresponding to compartments $j_P^A = 14$ and $j_P^B = 41$ in the two systems;
- In both systems the somatic compartment is located at a position $s_2 = 0.3185$ mm along z. (This corresponds to $j_{soma}^A = 22$ and $j_{soma}^B = 65$);
- The h_e responses to the single pulse input (6.2) are checked in three different positions: at the injection site p, at the soma s_2 and at an intermediate position between p and s_2, $s_1 = 0.2741$ mm, (with $j_{s_1}^A = 19$ and $j_{s_1}^B = 56$)

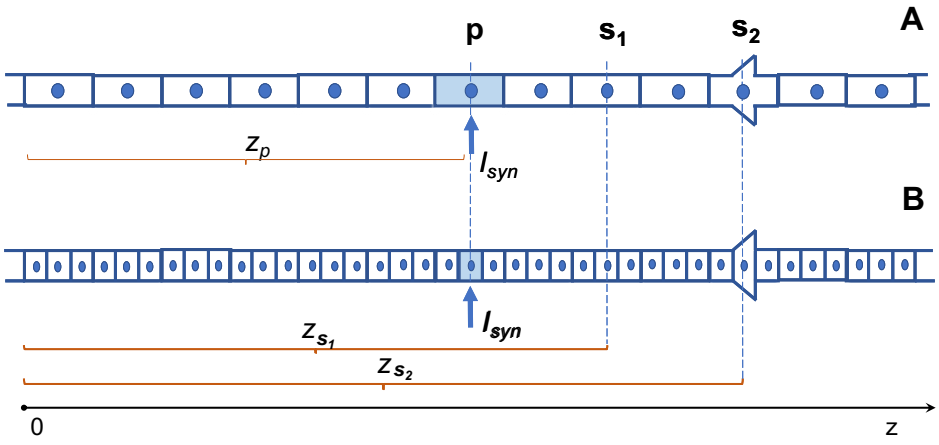

Figure 6.12: Schematic representation of two dendritic systems **A** and **B**, having n and $3n$ compartments, synaptic input (6.2) located at position p, and soma at position s_2.

A schematic representation of the two systems is provided in Fig. 6.12. Please, note that figures are merely illustrative and the actual number of compartments represented is reduced. The h_e responses of systems **A** and **B**, at the three locations, are shown, respectively, in figures 6.13 and 6.14. The maximum peaks at p,s_1 and s_2 are $[0.5978,\ 0.0958,\ 0.0451]$ mV in **A** and $[0.2032,\ 0.0325,\ 0.0153]$ mV in **B** (the ratio in between the maximum peaks in the two cases is $[2.9416,\ 2.9445,\ 2.9533]$ mV). We maintain the same number of compartments of **B** and add a synaptic factor $f_{syn} = 3$, compensating for the reduction of the maximum peaks observed when moving from system **A** to **B**. This new configuration **C** is illustrated in Fig. 6.15. The responses of systems **A** and **C** are compared in Fig. 6.16. Peaks in **C** are $[0.6032,\ 0.0965,\ 0.0453]$ mV. A similar result is obtained in configuration **D**, where 3 inputs (6.2) with synaptic factor $f_{syn} = 1$ are located in 3 different compartments (illustration in Fig. 6.17, results in Fig. 6.18).

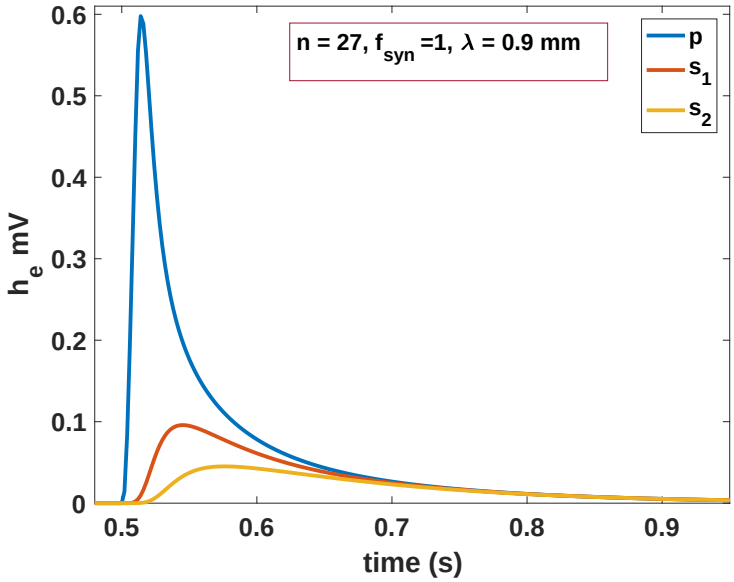

Figure 6.13: h_e responses (at positions p, s_1 and s_2) to a single pulse input (6.2) applied p, in system $\mathbf{A}$, i.e. $n_A = 27$ compartments with synaptic factor $f_{syn} = 1$.

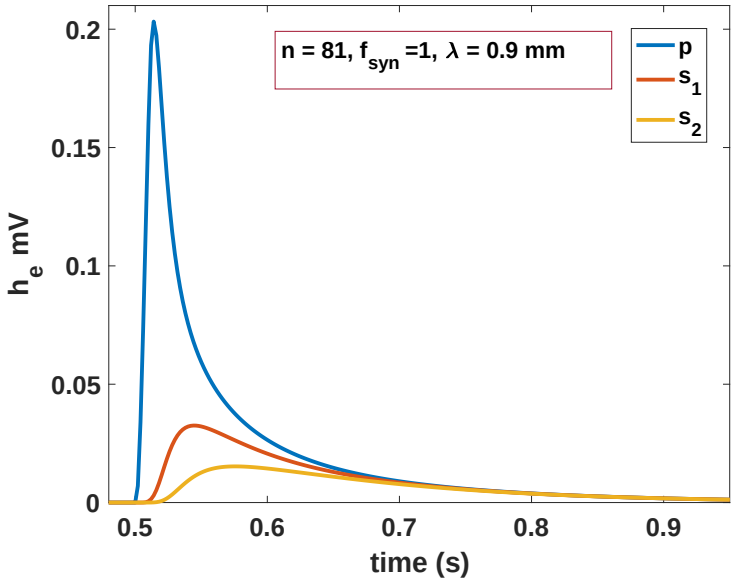

Figure 6.14: h_e responses (at positions p, s_1 and s_2) to a single pulse input (6.2) applied p, in system $\mathbf{B}$, i.e. $n_B = 81$ compartments with synaptic factor $f_{syn} = 1$.

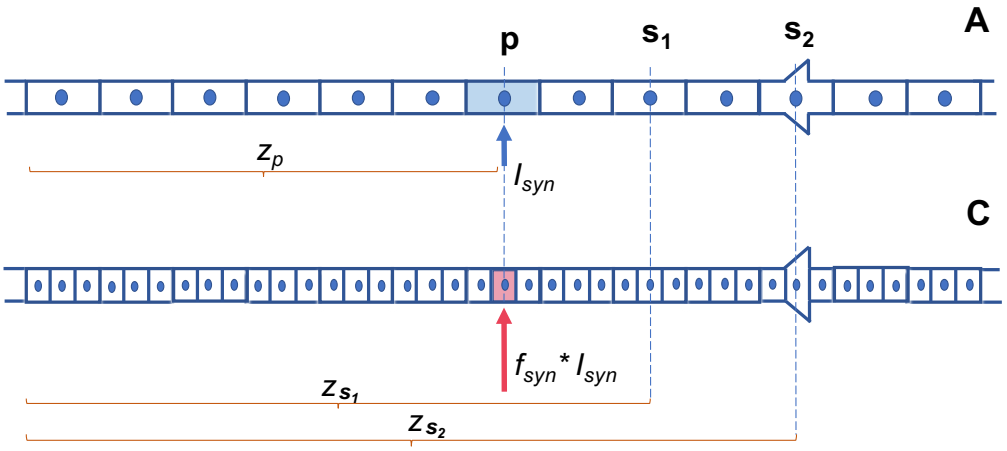

Figure 6.15: Schematic representation of two dendritic systems **A** and **C**, having n and $3n$ compartments, soma at position s_2, synaptic input (6.2) located at position p with different synaptic factors, $f_{syn} = 1$ (**A**) and $f_{syn} = 3$ (**C**).

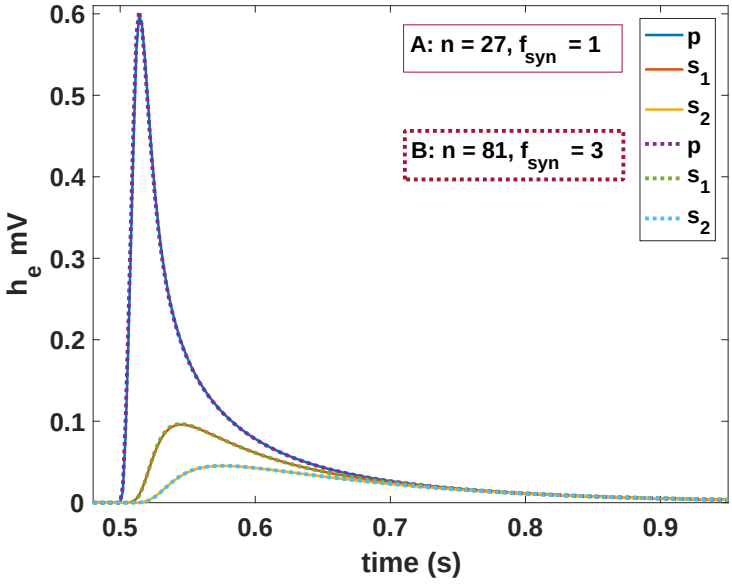

Figure 6.16: h_e responses (at positions p, s_1 and s_2) to a single pulse input (6.2) applied p, in system **A** (n compartments) without factor ($f_{syn} = 1$) and in system **C** ($3n$ compartments) with synaptic factor $f_{syn} = 3$.

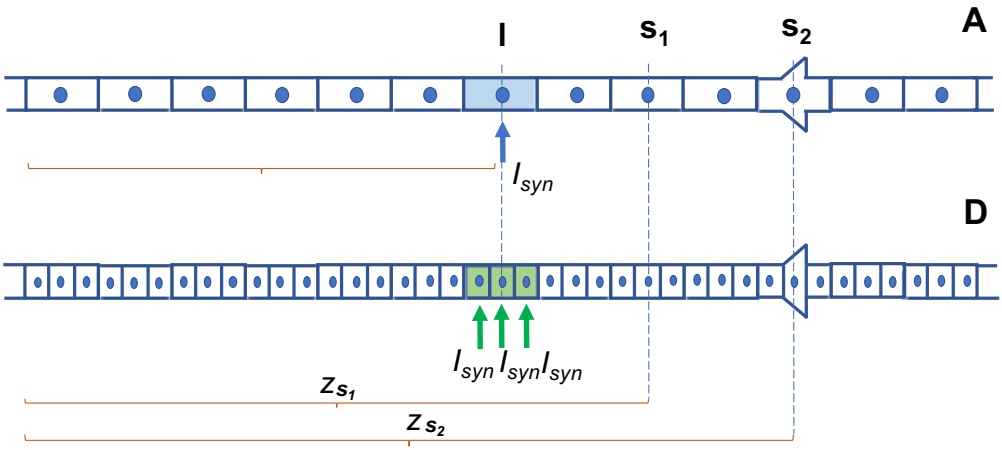

Figure 6.17: Schematic representation of two dendritic systems **A** and **D**, having n and $3n$ compartments, soma at position s_2, synaptic input (6.2) located at position p in **A** and at positions $p-1$, p, $p+1$ in **D**, synaptic factors, $f_{syn} = 1$ in both cases.

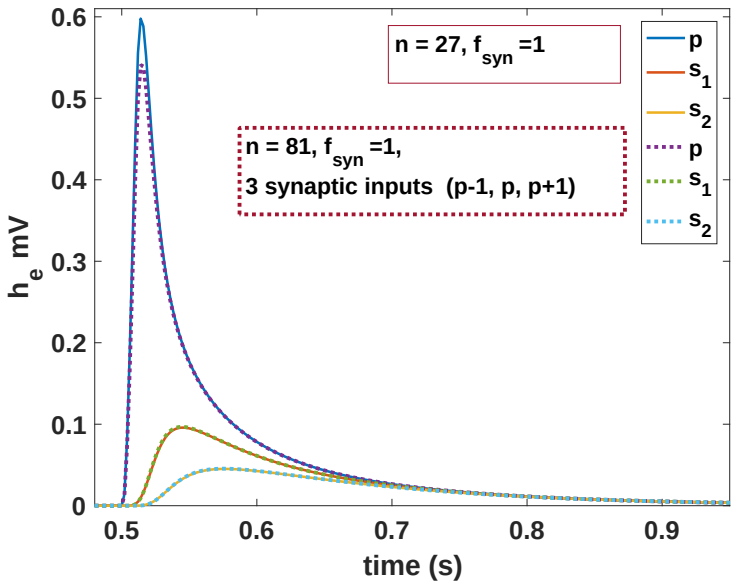

Figure 6.18: h_e responses (at positions p, s_1 and s_2) to a single pulse input (6.2) applied p, located at position p in **A** and at positions $p-1$, p, $p+1$ **D**, synaptic factors, $f_{syn} = 1$ in both cases.

6.3.3 General dependence of the synaptic factor on the number of dendritic compartments

The maximum depolarization at the injection site (during a pulse simulation) is depending both on the length constant λ and on the number of compartments n. We will explain in the next section (6.4) why the depolarization peak, at the injection site, decreases with increasing λ. Typical values for the maximum depolarization reached along the dendrite during a pulse simulation for systems with a different number of compartments are illustrated in Fig. 6.19. In these simulations, the same pulse input (6.2) is located in the central subsection and the length constant is fixed at $\lambda = 0.9$ mm.

The maximum depolarisations (*blu line*) at the injection site, as functions of the number of dendritic compartments n, are illustrated in Fig. 6.20. The figure shows that the depolarisation decreases as $1/n$ (the *orange line* represents the funtion $f(n) = c/n$, where $c = h_{101}^{max} * 101$ and h_{101}^{max} is the maximum depolarization at the injection site for the system with $n = 101$ dendritic subsections). As a consequence, the factor to be considered when moving from n to m compartments is $f_{syn} = m/n$. Some examples are illustrated in Fig. 6.21.

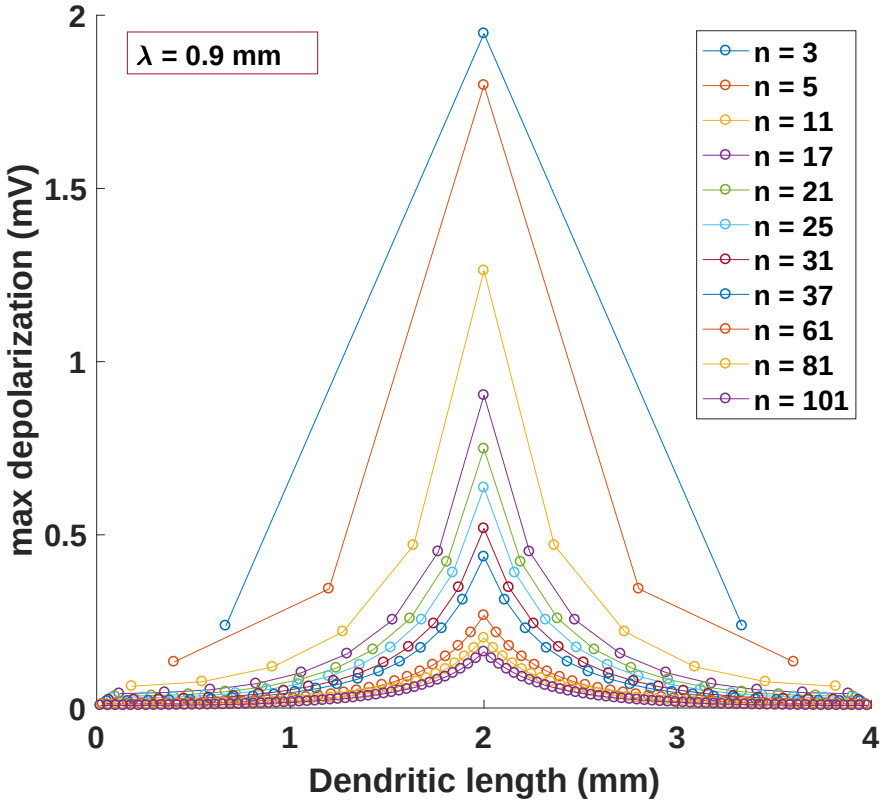

Figure 6.19: Max depolarization along the dendrite during a pulse simulation (6.2), for systems with different number of compartments n. The dendritic length constant is fixed at $\lambda = 0.9$ mm.

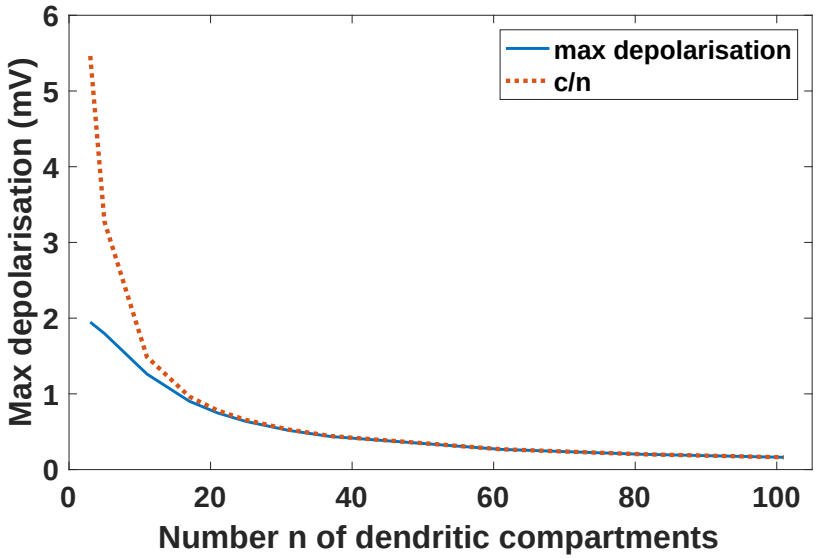

Figure 6.20: The max depolarization at the injection site, during a pulse simulation (6.2), is inversely proportional to the number of compartments n; $c = h_{101}^{max} * 101$ and h_{101}^{max} is the maximum depolarization at the injection site for the system with $n = 101$ dendritic subsections.

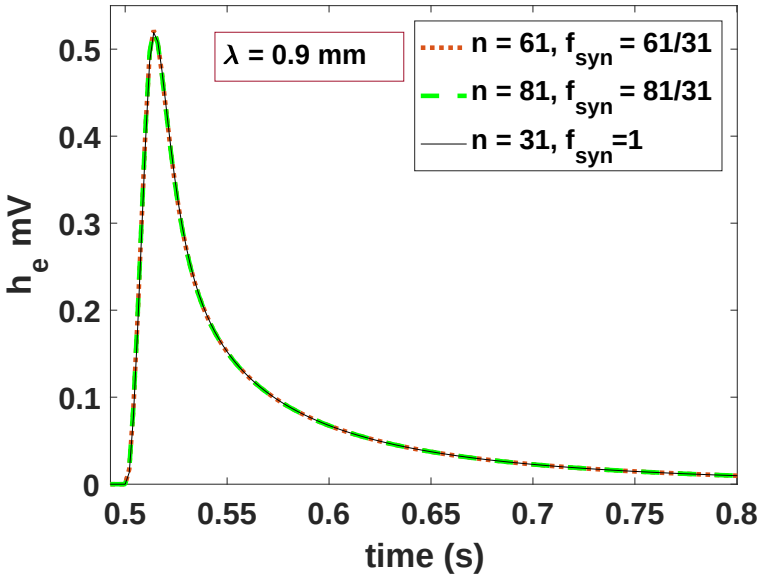

Figure 6.21: h_e response to a single pulse (6.2) for systems with $n = 31$ compartments is the same as the responses of systems with $m = 61$ and $m = 81$ compartments and factors $f_{syn} = m/n$.

6.3.4 Effect of synaptic factor and multiple inputs on linear spectra

In the previous sections, it has been shown that:
When moving from a system $\mathbf{A}$ *with* n *compartments to a system* $\mathbf{B}$ *with* $m = k \cdot n$ *compartments* ($k \in \mathbb{R}$), *the* h_e *responses along the dendrite are preserved by simply considering the linear relationships:*

$$f_{syn}^{B} = k \cdot f_{syn}^{A}, \quad or \quad f_{syn}^{B} = f_{syn}^{A} \quad for \quad k \quad different\ inputs \tag{6.23}$$

where f_{syn}^{X} *is the factor that multiplies the synaptic input in the system* $\mathbf{X}$, $\mathbf{X} \in \{\mathbf{A}, \mathbf{B}\}$. *Importantly, when using the relationships* (6.23), *also the PSD shape is preserved (shown below).*
An example is illustrated in Fig. 6.22, for a standard configuration (30 dendritic compartments, input and soma located at compartment 24).

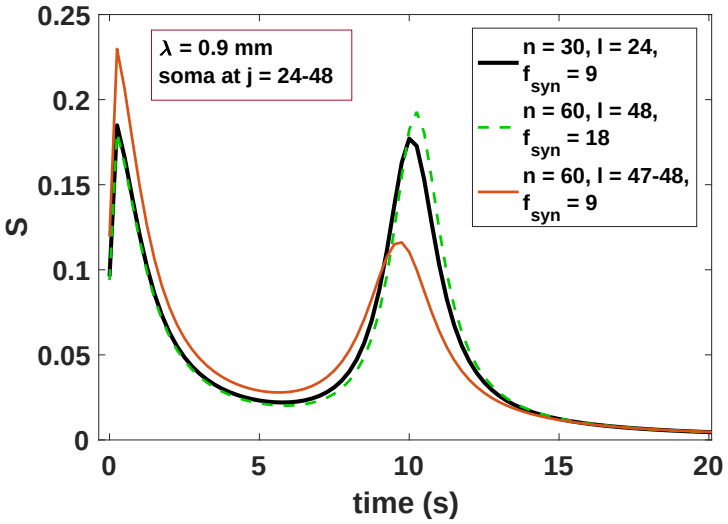

Figure 6.22: Using the linear relationships (6.23), for the synaptic factor f_{syn}, the PSD shape is preserved when moving from systems with $n = 30$ to $m = k \cdot n = 2 \cdot 30$ compartments.

6.4 Role of λ on the dendritic dynamics

In this section, the influence of λ on the dendritic dynamics is analysed in configurations where a single pulse input (6.2) is applied to the system. We consider here the standard system with $n = 30$ dendritic compartments, soma and input located at compartment 24. Both local ($l = 24$) and distal ($j = 15$) responses are plotted respectively in the Figs. 6.23 and 6.24. As can be noted in the figures, lower values of λ enhance the local response at the injection site (Fig. 6.23), while larger values of λ improve the propagation of the signal, eliciting larger distal responses (Fig. 6.24). This is confirming what explained in Section 3.2.3.

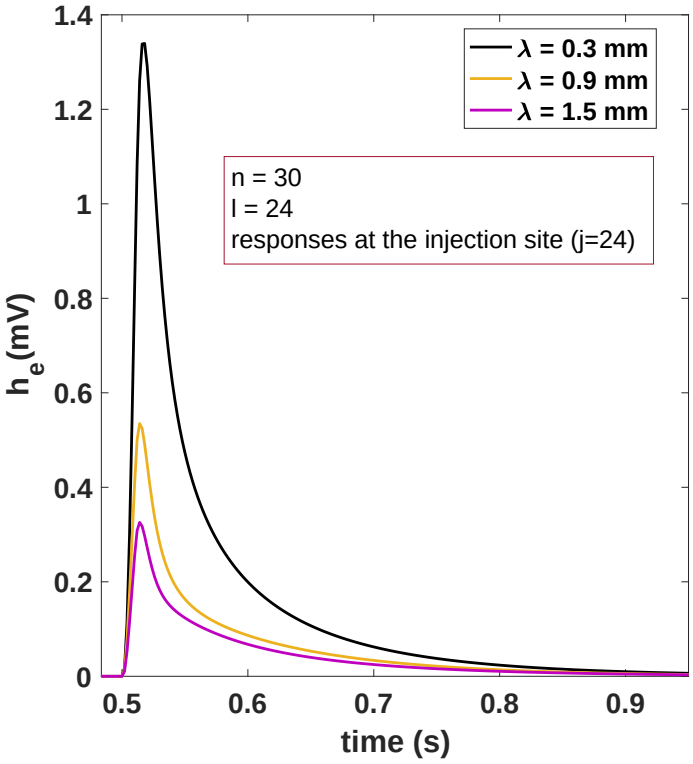

Figure 6.23: Lower values of λ lead to higher responses to a pulse (6.2) at the injection site. The system has 30 compartments and both soma and input are located at compartment 24.

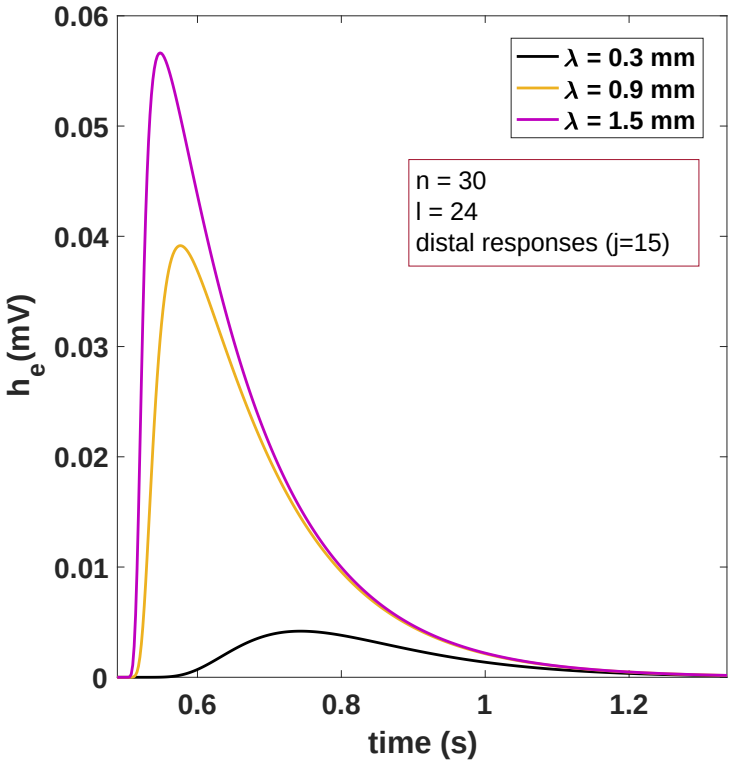

Figure 6.24: Higher values of λ enhance the propagation of the signal along the dendrite and lead to higher distal ($j = 15$) responses to a pulse (6.2) input located at compartment 24. The system has 30 compartments and both soma and input are located at compartment 24.

6.5 Stable points and linear spectra

The prediction of the power spectrum of frequencies is made from the Jacobian of the linearised system evaluated at a stable point. In this paragraph, the relationship occurring between the stable points and the linear spectra is illustrated. As merely illustrative cases, some spectra corresponding to limit cycles are shown at the end of this section.

Figures 6.25 and 6.26 show the first 2 seconds time series for 3 different values of the synaptic factor f_{syn} and the corresponding PSDs, for a system with the input located at the soma ($l = 24$, *control case*). As it can be seen in the figures, the convergence to a steady state solution and a corresponding spectrum with a peak in the alpha region are obtained for $f_{syn} = 9$. Higher values of the factor lead to self-sustained limit cycles (Fig. 6.25) and PSDs with peaks in the *gamma* band of frequency (Fig. 6.26). For lower values of the factor, the time series always converge to a stable point (see Fig. 6.27). The value $f_{syn} = 9$ is the critical value where a Hopf bifurcation occurrs.

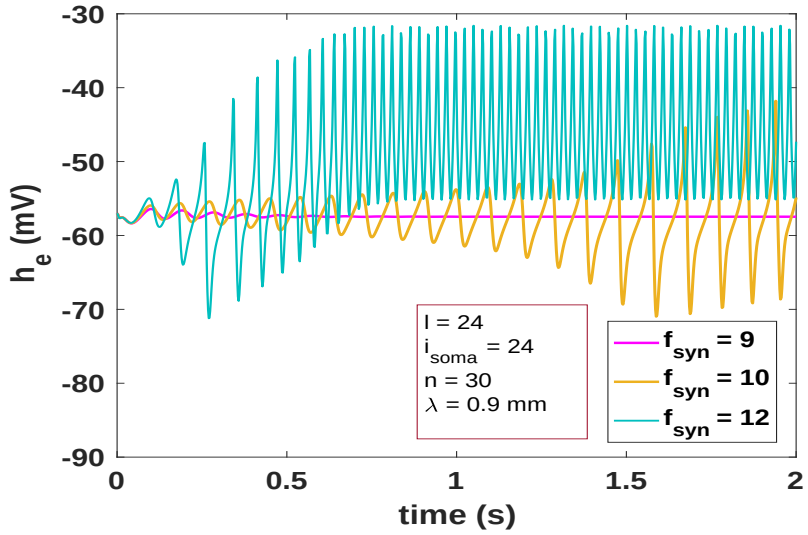

Figure 6.25: Time series for 3 different values of the synaptic factor. The system has 30 compartments and soma located at compartment 24. The input site is the soma and the synaptic length constant fixed in the 3 simulations ($\lambda = 0.09$ mm).

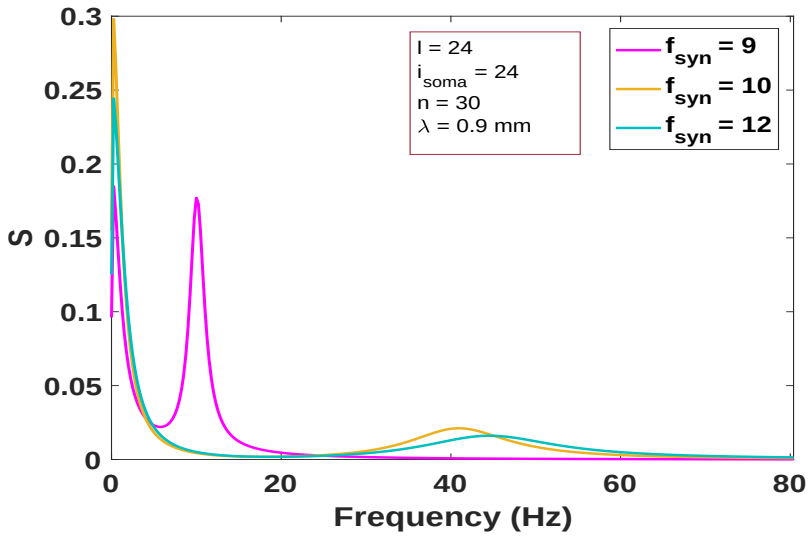

Figure 6.26: Linear PSDs, from Jacobian evaluation at stable ($f_{syn} = 9$) and unstable points.

Interestingly, for increasing factors, lower that the critical value, the corresponding spectra exhibit a peak that is progressively sharper and shifted towards higher *alpha* frequencies (Fig. 6.28). This result is also confirmed for different locations of the synaptic inputs, as shown in Fig. 6.29 where $l = 20$.

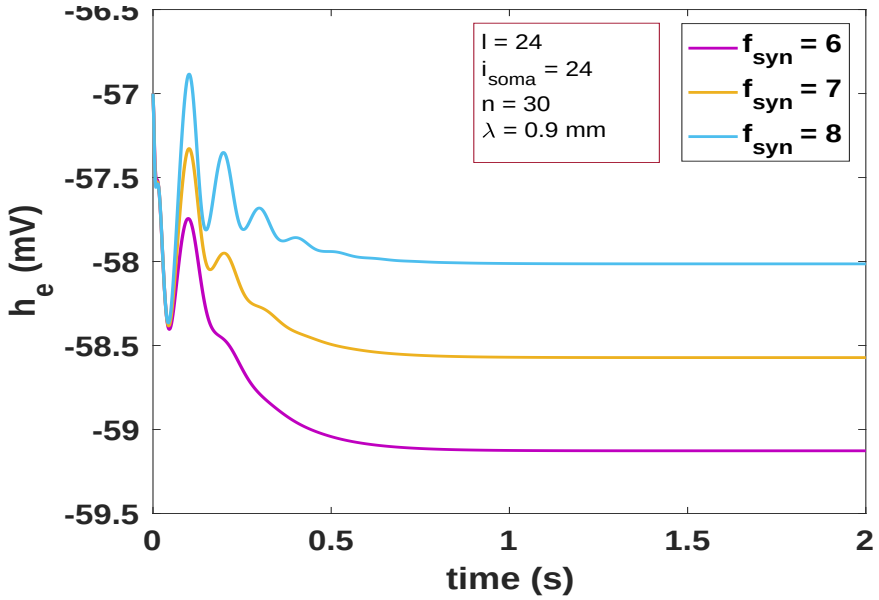

Figure 6.27: 3 time series converging towards stable equilibria for synaptic factors lower than the critical bifurcation value $f_{syn} = 9$. The system has 30 dendritic subsections, soma and input located at compartment 24, $\lambda = 0.09$ mm.

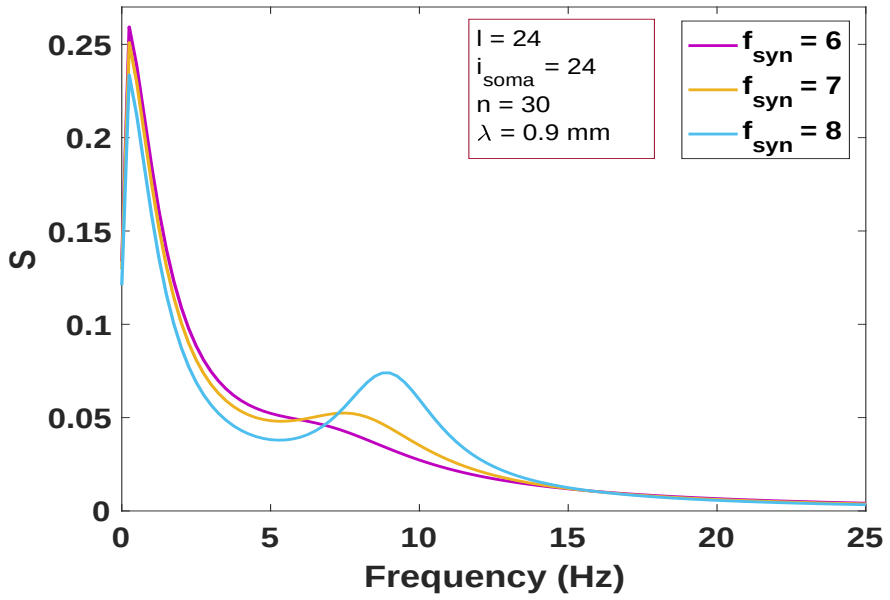

Figure 6.28: For increasing $f_{syn} < 9$ (corresponding to system configurations of Fig. 6.27), PSD peaks are progressively shifted towards the right.

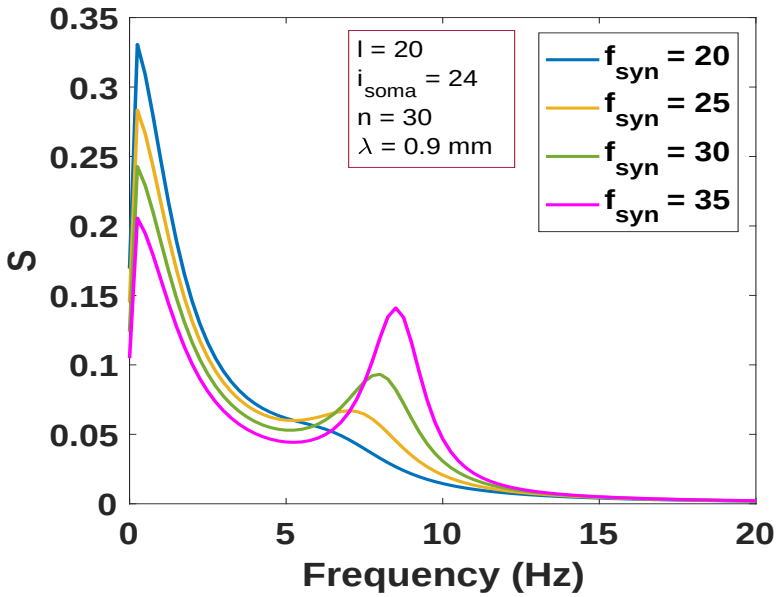

Figure 6.29: PSD peaks progressively shifted to the right for increasing f_{syn}. The system has 30 dendritic subsections with soma at compartment 24. The input is located in the proximal dendritic region ($l = 20$); $\lambda = 0.09$ mm.

As the location of the synaptic input is moved away from the soma, one can observe that the bifurcation occurs for higher values of f_{syn}. Figures 6.30 and 6.31 show the time series and the corresponding PSDs for a system where the input is located at compartment $l = 20$. The synaptic factor f_{syn} varies from 40 to 55. The alpha peak progressively moves towards higher frequencies and becomes sharper for values of f_{syn} approaching the critical value $f_{syn} = 58$. Figures 6.32 and 6.34 show the results for $f_{syn} \geq 58$ leading to self-sustained limit cycles. The sharp spectrum corresponds to the critical value, while the other spectra present small peaks in the *beta* range of frequencies. It must be noted, however, that in the case of limit cycles, the linearisation is not suitable to predict the dynamics of the system.

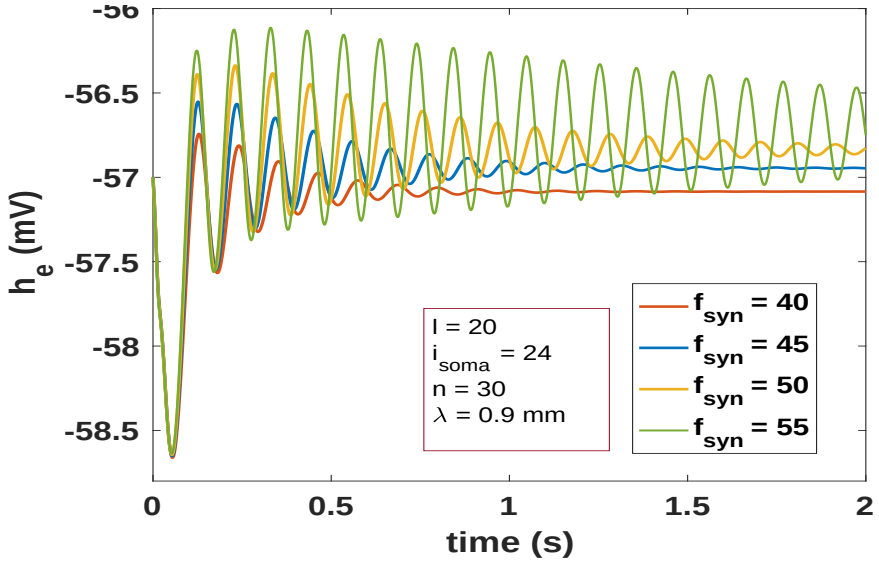

Figure 6.30: Time series for synaptic factors lower than the critical value. The input is located in a dendritic region proximal to the soma ($l = 20$).

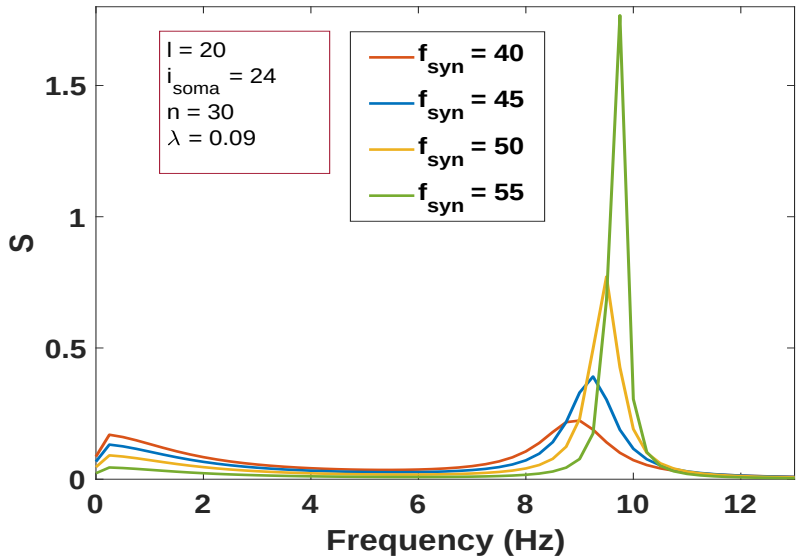

Figure 6.31: Linear PSDs corresponding to configurations shown in Fig. 6.30; *alpha* peaks move towards the right and become sharper for increasing f_{syn} approaching the critical value.

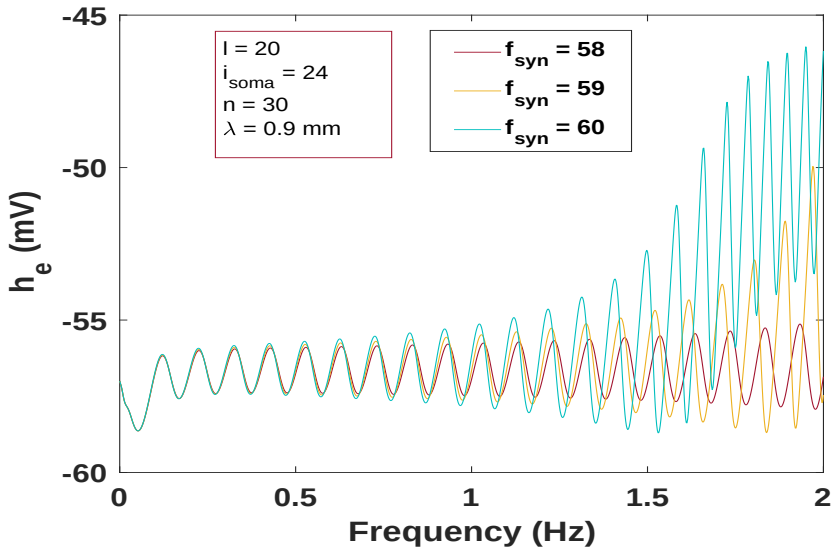

Figure 6.32: Time series obtained for synaptic factor greater or equal to the critical value ($f_{syn} \geq 58$) that are higher than the critical value. The input is located in a dendritical region proximal to the soma ($l = 20$).

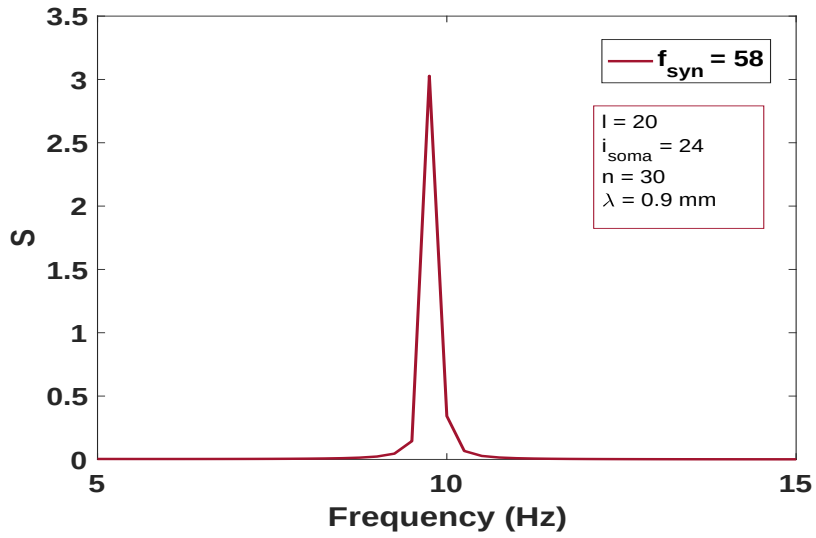

Figure 6.33: Linear PSD corresponding the critical value ($f_{syn} = 58$). The input is located in a dendritic region proximal to the soma ($l = 20$).

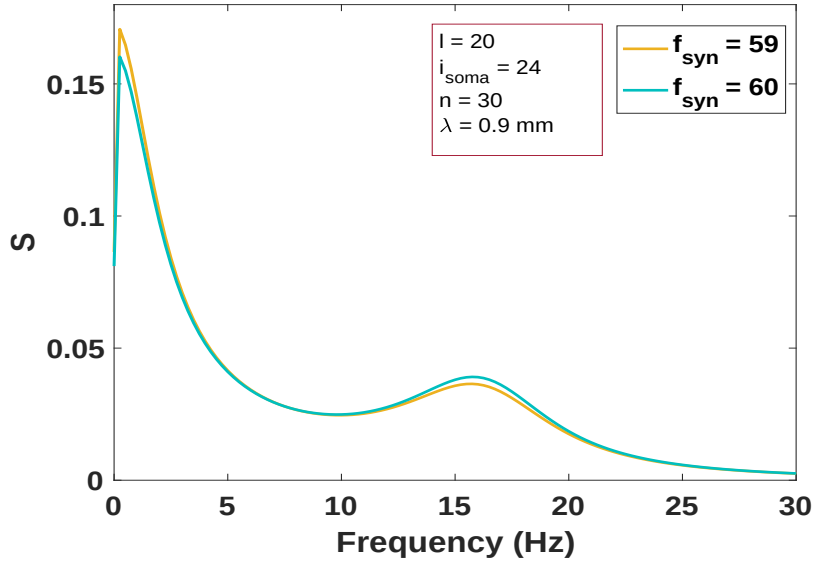

Figure 6.34: Linear PSD corresponding for synaptic factor greater than thee critical value ($f_{syn} = 58$). The input is located in a dendritic region proximal to the soma ($l = 20$).

6.6 PSDs depending on dendritic length constant, synaptic factor and location

The presence of PSDs with a peak in the alpha range of frequencies has been systematically investigated in this Section, with varying dendritic length constant λ, synaptic factor f_{syn}, and synaptic input location l. For this systematic study, the standard configuration ($n = 30$ dendritic compartments and soma located at compartment 24) with input location varying along the dendrite has been considered.

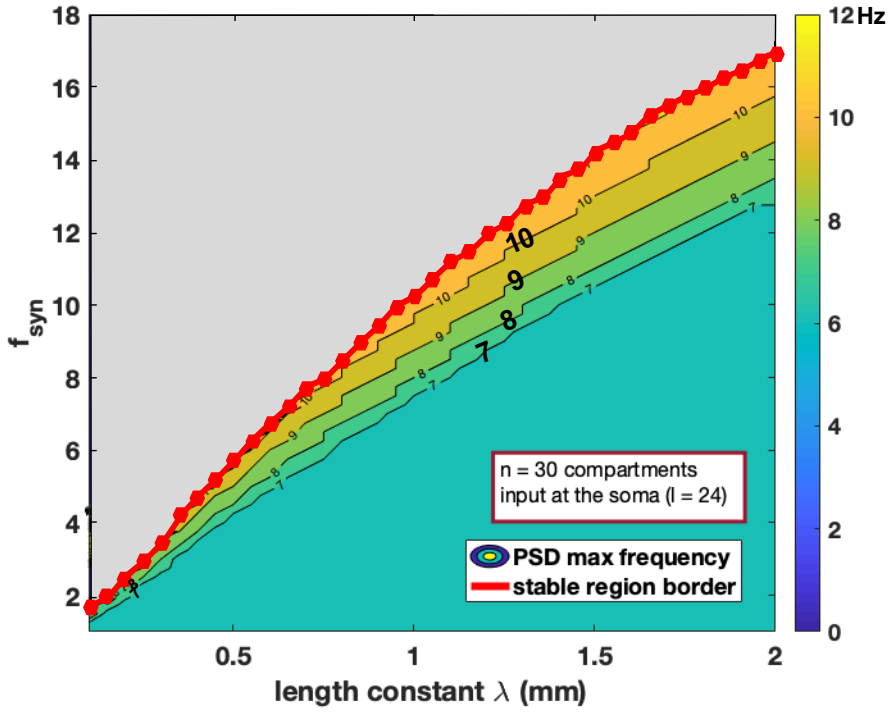

Figure 6.35: Frequency at which a peak in the PSD occurs, for a standard system ($n = 30$ and soma at compartment 24) with input at $l = 24$.

In Fig. 6.35, the frequency at which a peak in the PSD occurs (we are not considering the peaks in the δ region) is represented, as a function of the synaptic factor f_{syn} and dendritic length constant λ, in the case $l = 24$. The red line in the figure defines the up border of the "stable" region, i.e. the region where the steady

states are stable equilibrium points and the Jacobian evaluation at the equilibrium point is suitable to predict the system PSD. As expected, since the input is located at the soma, lower values of λ enhance the local excitability and lead to PSD with α peaks, for small values of synaptic factors ($f_{syn} = 1.5 - 2$). For the same reason, critical values of the synaptic factor (*red line*) are smaller for lower values of λ. For $\lambda = 2$ mm, factors in between 12 and 16 lead to PSD with peaks in between 7 and 11 Hz. For each value of λ, factors higher than the critical points lead to linear spectra with peaks in the β range. These results, confirm what we have illustrated in Section 6.5.

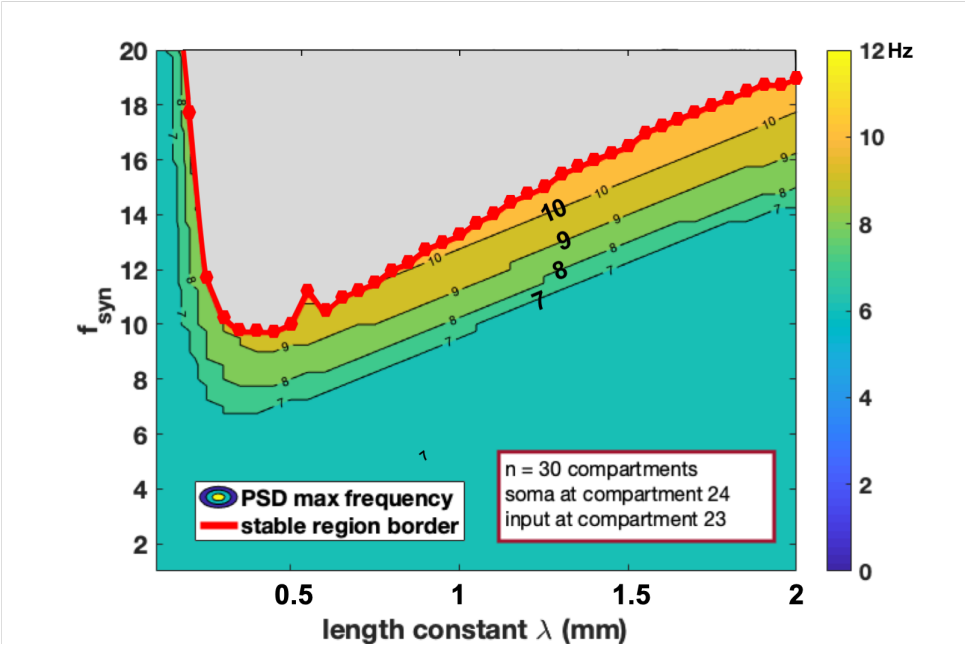

Figure 6.36: Frequency at which a peak in the PSD is occuring, for a standard system with input at $l = 23$.

Analogous graphs (Figs. 6.36-6.39) show the behaviour of the standard system, respectively for synaptic input locations $l = 23, 21, 20, 18$. Moving the input away from the soma progressively shifts the beginning of the unstable region towards higher frequencies: in the figures, the region on the left of the graphs (small λ) is stable for the represented values of the factor, and PSD have peaks at about 7 Hz. Interestingly, compared to Fig. 6.35, in the case $l = 23$ (Fig. 6.36), the stable region

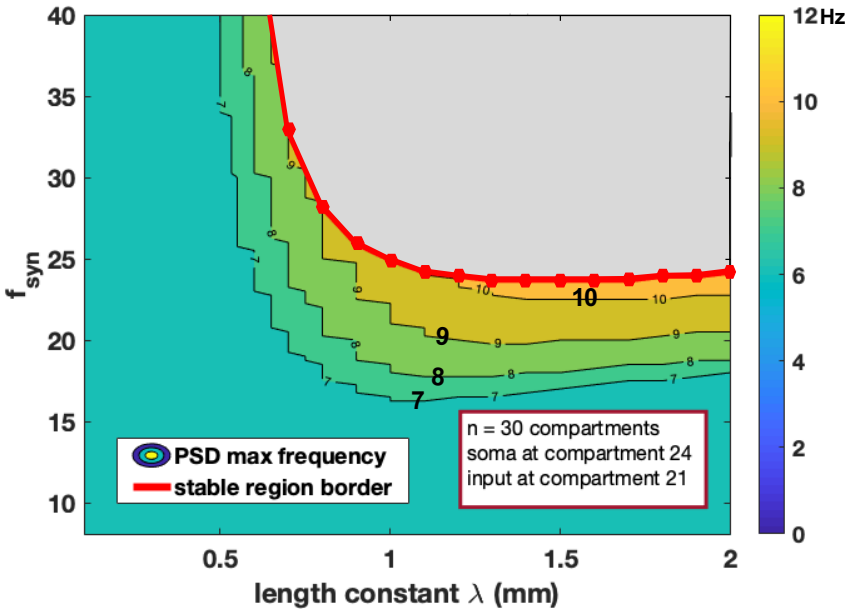

Figure 6.37: Frequency at which a peak in the PSD is occuring, for a standard system with input at $l = 21$.

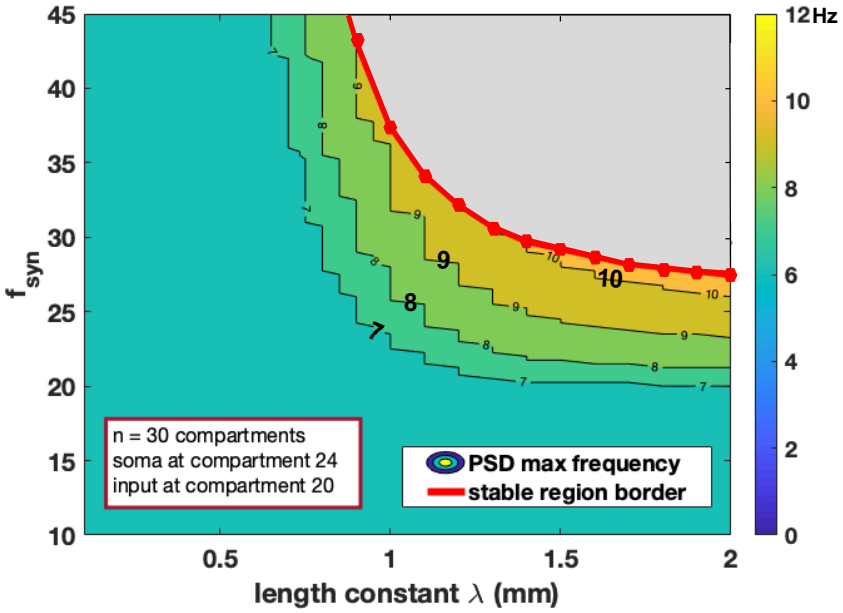

Figure 6.38: Frequency at which a peak in the PSD is occuring, for a standard system with input at $l = 20$.

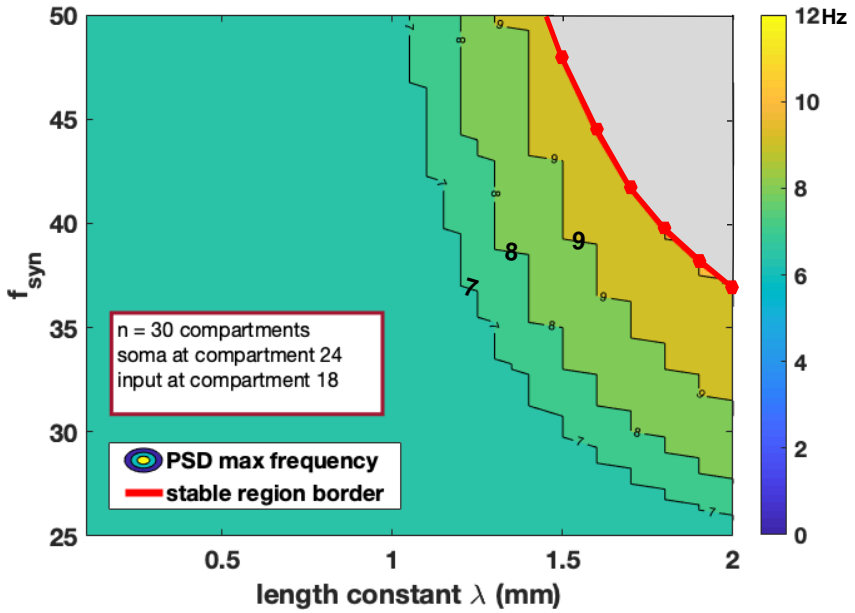

Figure 6.39: Frequency at which a peak in the PSD is occuring, for a standard system with input at $l = 18$.

border has a lower steepness (for $\lambda > 0.4$ mm). This slope almost falls to zero for $l = 21$ (Fig. 6.37 for $\lambda > 1.2$ mm) and becomes negative for $l \geq 20$ (Figs. 6.38-6.39, for all values of λ). This is confirming that higher values of λ correspond to a better propagation along the dendrite and enhance the "excitability" of the system when the input is progressively moved away from the soma. As an example, for $l = 18$ (Fig. 6.39), when λ is increasing from 1.5 mm to 2 mm, smaller factors are gradually needed to generate the α-rhythmicity, while for $\lambda < 1.1$ mm all factors considered (25-50) do not lead to PSDs similar to the ones encountered in humans. In cases as $l = 18$ (Fig. 6.39), to improve the excitability of the system and detect regions for "good" spectra, it is possible to consider multiple inputs, (i.e. inputs located in more than one compartment) and/or modify some parameters. In the following sections both these techniques are briefly explored.

6.7 Multiple synaptic inputs spectra

In the standard system (30 compartments, soma and input at compartment 24, dendritic length constant $\lambda = 0.9$ mm), the right factor leading to a spectrum with a shape typical of human spectra is $f^{3D}_{n=30} = 9$. The system exhibits lower alpha peaks or flat spectra for synaptic factors lower than 8 (see Section 6.5). However, considering 2 or more additional inputs, *located both in the apical and in the basal regions*, spectra with α-peaks are obtained for lower synaptic factors. Results are shown in Figs. 6.40 and 6.41 for $f_{syn} = 7$ and $f_{syn} = 4$, respectively.

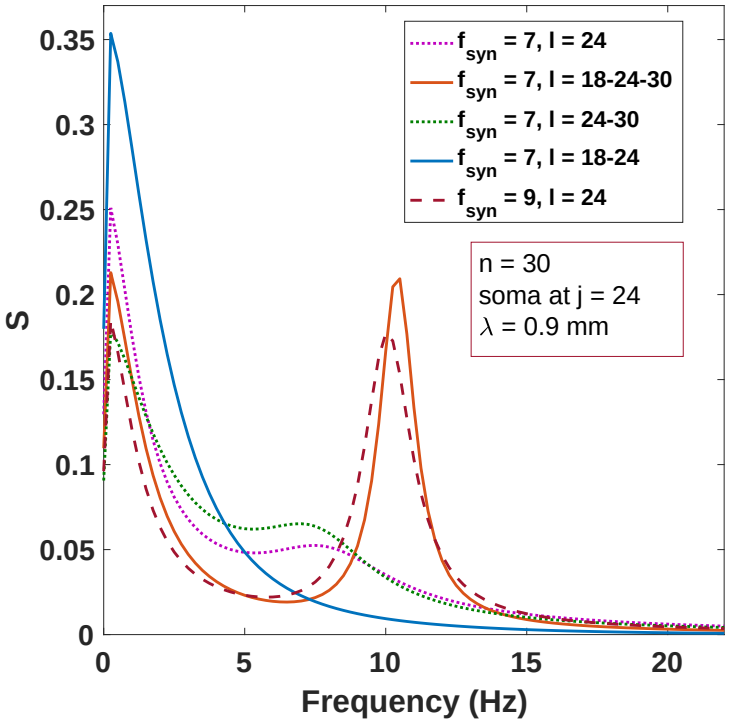

Figure 6.40: The standard system (30 compartments, soma and input located at compartment 24) with *one single input* exhibits an α-peak for $f_{syn} = 9$. Lower factors are needed if *multiple inputs* both in the apical and basal region are additionally considered.

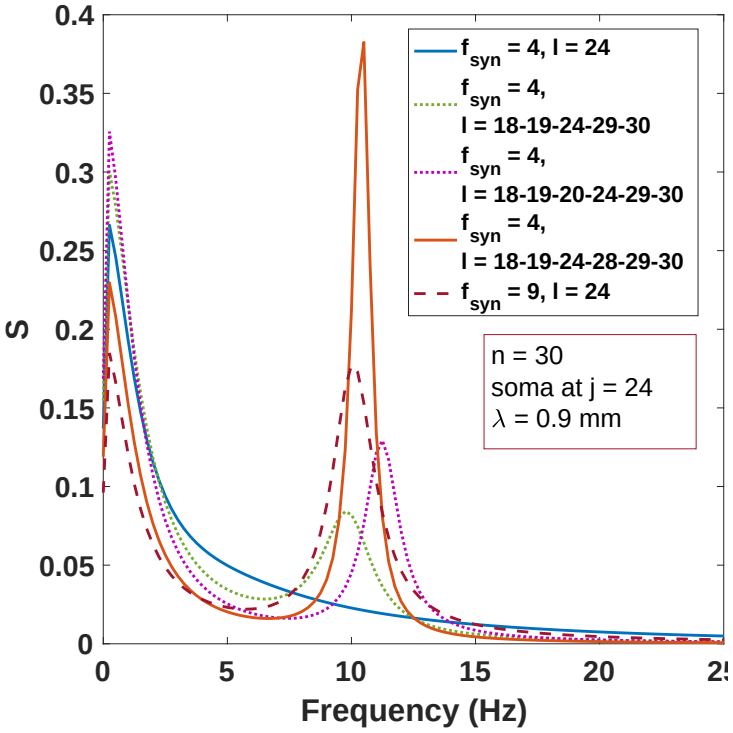

Figure 6.41: The standard system (30 compartments, soma and input located at compartment 24) with *one single input* exhibits an α-peak for $f_{syn} = 9$. Lower factors are needed if *multiple inputs* both in the apical and basal region are additionally considered.

6.8 New PSO spectra

Our results (Section 6.6) show that the standard system[4] ($n = 30$, soma at compartment 24) with synaptic input located at $l = 18$, presents spectra without a "proper" alpha peak for $\lambda < 1.1$ mm and synaptic factors up to 50 (Fig. 6.39). However, by changing some parameters, it is possible to enhance the "excitability" of the system and obtain PSDs with α-peaks for some values of the synaptic factor.

The parameters that are modified in this study are the mean resting membrane potential h_e^r and the time constant τ_e, since h_e^r influences the resting state of the dendritic h_e, and both h_e^r and τ_e affect the overall "excitability" of the system[5]. Firstly, some good regions have been found manually. Secondly, the PSO algorithm has been performed in the search-space S_2 (defined in Section 5.4.1):

$$S_2 = \{\tau_e, h_e^r, f_{syn}, \lambda\} \tag{6.24}$$

using the fitness function f_2, defined in Section 5.4.2.

The standard values for these two parameters are $h_e^r = -62.226$ mV and $\tau_e = 132.55$ ms. We found that, manually decreasing both parameters, within their physiological range of variability, can lead to "good" spectra (for values of $\lambda < 1.1$ mm) for factors that are lower for larger λ. Results are shown in Figs. 6.42 and 6.43, for λ respectively 0.5 and 1 mm.

Similar results are obtained from the PSO search, as shown in Figs. 6.44 and 6.45. The range of variability of the parameters is defined in table 5.1. For λ values lower than 1.1 mm have been used ($[0.6,\ 0.8]$ mm for simulation 1 shown in Fig. 6.44, $[0.4,\ 0.6]$ mm for simulation 2 shown in Fig. 6.45). The optima are obtained respectively for

$$[\tau_e, h_e^r, f_{syn}, \lambda] = [54.4 \text{ ms}, \ -60.0652 \text{ mV}, \ 38.0582, \ 0.727 \text{ mm}] \text{ and} \tag{6.25}$$

$$[\tau_e, h_e^r, f_{syn}, \lambda] = [37.8 \text{ ms}, \ -60.00 \text{ mV}, \ 31.7285, \ 0.60 \text{ mm}] \tag{6.26}$$

The spectra in the figures confirm that, by varying the parameters τ_e and h_e^r, good spectra can be obtained even for values of $\lambda < 1.1$ mm and input located

[4]The values for the parameters of the model are illustrated in Table 3.3.

[5]For a more detailed discussion about the impact of these parameters on the alpha rhythm, see Section 4.5.

at compartment $l = 18$. This result can likely be extended for configurations with input located in the distal regions, opportunely varying the parameters in the search spaces.

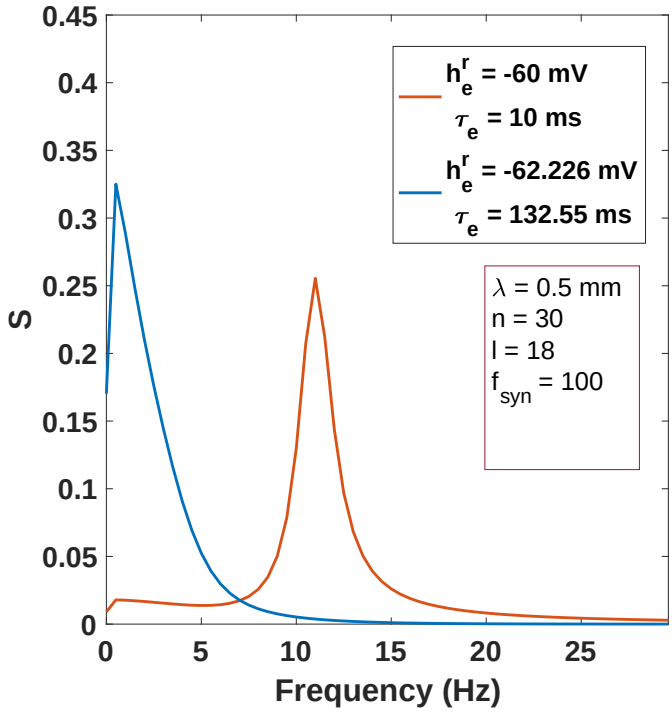

Figure 6.42: Different values for h_e^r and τ_e can lead to good spectra in the standard systems, with input at $l = 18$ and $f_{syn} = 30$, while standard values $h_e^r = -62.226$ mV and $\tau_e = 132.55$ ms lead to flat spectra.

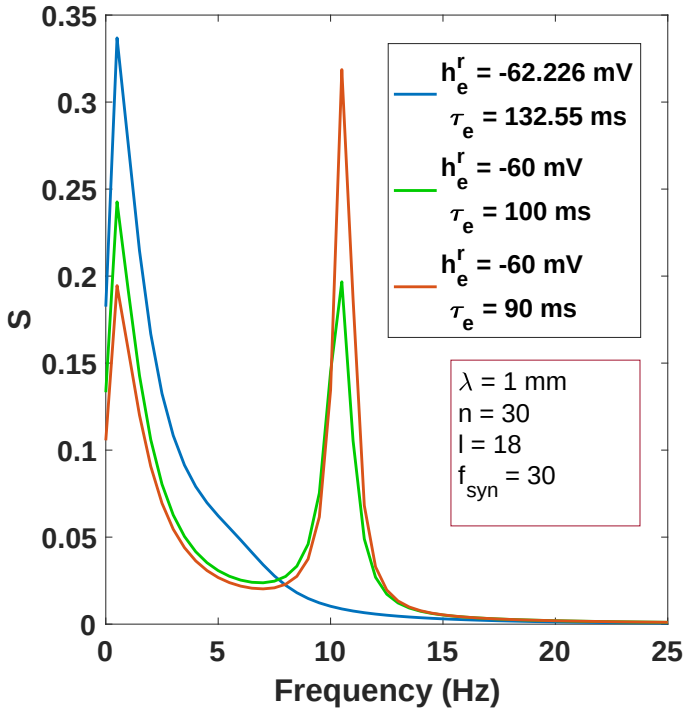

Figure 6.43: Different values for h_e^r and τ_e can lead to good spectra in the standard systems, with input at $l = 18$ and $f_{syn} = 30$, while standard values $h_e^r = -62.226$ mV and $\tau_e = 132.55$ ms lead to flat spectra.

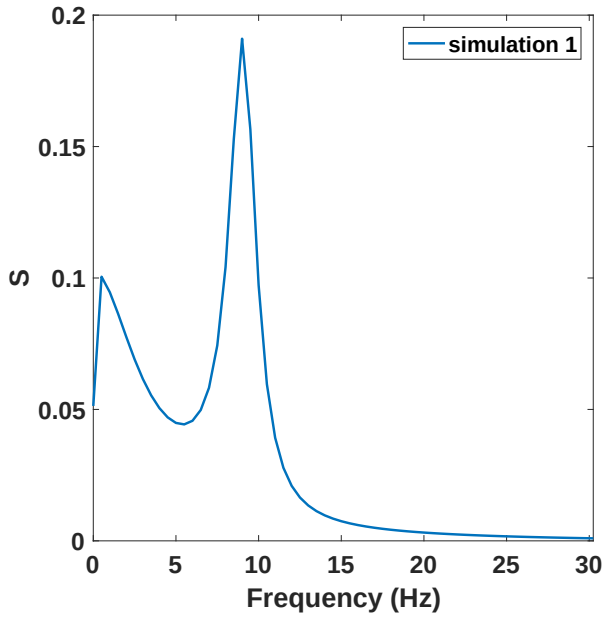

Figure 6.44: Optimal spectrum obtained with the PSO in the search space S_2 and $\lambda \in [0.6,\ 0.8]$ mm.

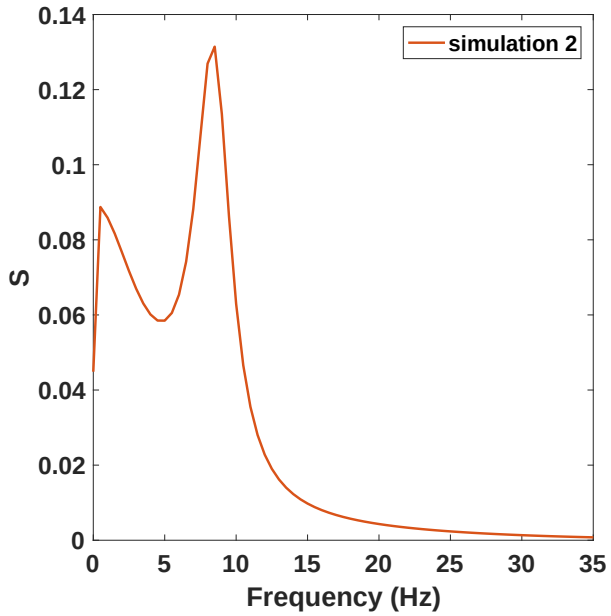

Figure 6.45: Optimal spectrum obtained with the PSO in the search space S_2 and $\lambda \in [0.4,\ 0.6]$ mm.

6.9 Discussion

We have formulated a NFM with an extended dendritic tree that is able to capture the chief properties of human alpha activity, which is conceived as the main rhythm of spontaneous EEG [248]. Pivotal to this endeavor has been testing the new model dynamics to adopt plausible parameters. In particular, for the purpose of characterizing alpha activity in terms of dendritic tree mechanisms, the research effort has focused on dendritic parameters introduced in the 3D geometry. Our study outlines plausible strength, density and distribution of synapses over the dendritic domains and points out the potential role in NF models of a dendritic length constant varying with the distance from the soma. Furthermore, other parameters of the model have been opportunely chosen, using the PSO technique, to give rise to alpha rhythmicity. From a more generic perspective, we have proposed a possible relation between the genesis of alpha activity and well known dendritic mechanisms like *dendritic democracy* [31] and *dendritic coincidence detection* [134]. Moreover, our results predict the role of a progressively increasing synaptic conductance on the occurrence of abrupt changes in the alphoid dynamics [248], mathematically corresponding to Hopf bifurcations.

Dendritic democracy and alpha-rhythm

The concept of *dendritic democracy* relies on the essential fact that the efficacy of synaptic input is normalised at the soma [31]. To put it in another way, the excitatory post-synaptic potentials (EPSPs) have the same somatic amplitude, independently of the location of the synapses where the EPSP is initiated. Dendrites behave as leaky electric cables filtering the signals passing through them, causing attenuation of EPSPs at the soma generated by inputs in the distal dendritic domains [150]. Since the somatic response is nevertheless normalized, physiological mechanisms compensating for the filtering behaviour of the dendrite are required. The first evidence that synaptic efficacy could be scaled with the distance from the soma has been provided by the pioneering work of Jack and Redman [249] on CA1 pyramidal neurons. After a couple of decades, the work of Magee and Cook [152]

proved that the compensation is more likely due to an increase in synaptic conductance, precluding the hypothesis that voltage-gated channels could be responsible for the genesis of this phenomenon. Nevertheless, voltage dependent channels have a role in the amplification of synaptic responses when multiple synapses are activated [152, 176]. Over the years a variety of studies have provided further experimental evidence for the scaling of synaptic efficacy with the distance from the soma [32, 153, 250] and, although several mechanisms can be involved in the genesis of the compensation, the scaling of synaptic strength over the dendritic membrane is hypothesized to be the main cause underlying the dendritic democracy. The increase of synaptic conductance could in turn originate from several intrinsic phenomena like self-regulation activated through back propagation of action potential through the dendritic tree [149] or synaptic plasticity [251, 252].

Independent of the intrinsic mechanisms responsible for the regulation of synaptic efficacy with distance, the crucial strengthening effect of the synaptic conductance can be reproduced through modelling choices boosting the synaptic inputs, especially in the case of single synaptic inputs. The effect of a strengthened synaptic input with distance has been studied here, proving that when a single pulse input is provided to the 3D NF model, an increasing synaptic input is necessary to render the somatic response independent of the dendritic input location.

Furthermore, the present work speculatively identifies the dendritic democracy as a key mechanism responsible for the genesis of alpha-rhythmicity. Compared to the 2D model [36], the present 3D NF model including the dendritic dimension necessitates an additional *synaptic strengthening factor* to reproduce the alpha band activity, and this compensating mechanism has to increase with the distance of the synaptic input. Our results also prove that the necessary factor *linearly* increases with the number of dendritic compartments used for the spatial discretisation. Furthermore, multiplying the synaptic strength f_{syn} by a factor k is equivalent to considering k simultaneuos inputs in adjacent locations, according to the idea that synaptic democracy can be achieved by either increasing the synaptic conductance or by synchronous synaptic activation [31]. All these mechanisms (*necessity of intrinsic synaptic factor, linear increase of the factor with the number of com-*

partments and equivalence of strengthened and synchronous synaptic activation) are preserving not only the somatic responses to pulse inputs but more importantly the occurrence of alpha rhythmicity.

Our systematic study on the occurrence of alpha rhythm depending on synaptic factor and dendritic length constant (Section 6.6) suggests that the synaptic strength has to increase linearly in the proximal region, while for input in the central apical domains the scale is likely to be faster than linear and other parameters of the model have to be modified to achieve the alphoid PSD, as proved using the PSO technique. These results are qualitatively in agreement with the work of Timofeeva et al. [34], providing a mathematical exploration of the "democtratic" phenomenon. In this paper, the scientists adopted a passive model of a dendritic tree and studied a variety of key measures of synaptic compensation, finding that the linear scale of synaptic strength required in the proximal region becomes supralinear in distal region. Moreover, beyond some critical distance, besides the increase in the synaptic strength, other mechanisms like active currents must be involved to compensate for the input distance.

Dendritic coincidence detection

Our results suggest that, during asynchronous synaptic activation of proximal and central dendritic domains, a factor strengthening the synaptic efficacy is essential to achieve the alphoid dynamics. However, beyond a critical distance[6], a simultaneous modification of other parameters is needed to achieve PSDs with peaks in the alpha-band (Section 6.8 and Chapter 5), together with significantly high factors[7]. Interestingly, during more synchronous activation of both basal and apical domains, reasonably lower factors are sufficient for the genesis of alpha rhythm, even without a further modification of other parameters of the system. Tipically, dendritic coincident activation is studied for assessing the occurrence of action po-

[6]The critical distance from the soma is 0.8 mm, corresponding to an input located at compartment 18. The parameters of the 3D NF model are illustrated in Table 3.3.

[7]In Chapter 5, the synaptic factor f_{syn} has value up to 1000 in many of the optimal solutions provided by the PSO technique.

tentials in single neuron models including active properties [134]. Here, coincidence detection has been identified as a potential mechanism for the genesis of the alpha activity. Broadly speaking, the general conditions that implicate the onset of action potential in active models seem to promote the alpha activity in passive models of the dendritic tree.

Dendritic length constant

Our study of physiologically admissible parameters giving rise to alpha-dynamical states outlines possible inferences about the "suitable" value for the dendritic length constant λ, i.e. values enhancing the onset of alpha rhythm. Both the exploration of parameter spaces conducted in Chapter 5 and the numerical analysis illustrated in Sections 6.4 and 6.6 provide good reasons to think that, in agreement with the physiological role of λ [163], higher values of the length constant promote the alpha rhythm when distal synaptic inputs are active, while lower values are more plausible in the presence of proximal inputs. It is important to note that this observation is pertinent in systems where inputs are activated in single dendritic domains and a *unique constant value of* λ *is assumed over the entire dendritic tree*. This particular estimation is coherent with the idea of dendritic democracy [31], in that a higher λ enhances the propagation, preventing the loss of signal spreading towards the soma, while lower values for the length constant impede the propagation of somatic inputs towards the tree.

Synaptic strength and linear stability

Numerical analysis has revealed that increasing the synaptic strength generates a progressive shift of the dominant PSD peak towards higher frequencies. This also generally produces progressively sharper alpha resonances. If one further increases the synaptic factor the equilibrium loses stability in a Hopf bifurcation, normally occurring at a frequency of about 11-12 Hz. In [248], Liley and Bojak provided a theory for the genesis of alpha rhythm, studying the dynamics of the Liley model [36, 42, 253]. Although their model do not include a dendritic dimension, our results are in agreement with their findings. The Liley model, in fact, exhibits an analogous

loss of stability in Hopf bifurcations if one increases the extra-cortical input p_{ee}. This drives the model into oscillatory states at a frequency of about 11Hz. Indeed, both the synaptic factor f_{syn} and the extra-cortical input p_{ee} strengthen the intensity of the synaptic currents, driving the models into comparable dynamical states.

Chapter 7

Conclusions and future work

Although some NF studies have partially explored the role of the denrite [138–140, 140], NF theory has mainly used models in which the dendritic dimension is collapsed into a point and the cortex is modelled as a two-dimensional sheet, neglecting all the dendritic properties essential to model the cortical dynamics.

We have implemented and studied a three-dimensional cortical NF model including the dendritic dimension. Our model represents the dynamics of a square portion of the cortex (from 1 mm^2 to 256 mm^2) corresponding to the activity of a number of cortical microcolumns (from 1 to 256). The model is an extension of the 2D NF model [36], and is capable of reproducing the alpha rhythmicity typical for the human cortical activity at rest.

Both the non-linear and the linear power spectral densities (PSDs) of the mean membrane potential have been calculated. The linear approximation of the PSD has been found to be a good approximation of the non-linear one close to stable fixed point, allowing a significant decrease of the computational time. The procedure applied to find the linear PSD using the spatial Fourier transform is an extension to the 3D geometry of the one used in [36] for a two-dimensional sheet. In the present work, we have also developed a mathematical formulation to find the linear PSD, without using the spatial Fourier transform. This formula is applied in systems including several macrocolumns and it provides an accurate prediction as judged by

comparing with spectra derived from a time series computation.

Naively extending the 2D NF model to a 3D geometry (with all the 2D parameters unchanged), leads to "flat" power spectra. However, we found that the alpha rhythmicity can be reproduced by introducing a factor strengthening the synaptic input. The impact of the factor on the dendritic dynamics has been investigated in detail and its introduction has been motivated by taking into account the effective membrane surface where the input is applied. This allowed us to find an important linear relationship between the strength of the synaptic input needed to elicit a particular response and the extent of the surface where the input is applied. Furthermore, by applying a simple conversion formula for the synaptic factor, it is possible to elicit the same response in systems with different number of dendritic compartments. Moreover, by applying the same formula, the alpha rhythm is also preserved. The phenomenon of "dendritic democracy" [31] indicates the normalization of inputs arriving at the soma from different dendritic locations. Our results are in line with this mechanism compensating for the dendritic filtering [30] and suggest that, to some extent, the dendritic democracy can be linked with the occurrence of alpha rhythmicity.

Next, we have found that the alpha rhythm in the 3D model can be reproduced by introducing the synaptic factor and simultaneously changing some of the 2D parameters. This simultaneous variation of some parameters is particularly needed in configurations where the input is not located at the soma. The particle swarm optimization (PSO) technique has been used to detect these parameter sets for single microcolumn 3D systems. We implemented from scratch the PSO algorithm and explored different search spaces. The fitness function used has been progressively improved to better evaluate the similarity of the spectra obtained with those encountered in humans. Additionally, this study could be extended by considering different topologies for the communication of the particles in the swarm or by investigating other parameter spaces. Moreover, further fitness functions could be developed and the technique could be extended to multiple microcolumn systems.

The alpha rhythm is easily reproduced if the synaptic input is located at the soma or at locations proximal to the soma, for specific values of the synaptic factor f_{syn}^{3D}, depending on the intrinsic properties of the 3D model (in particular the dendritic length constant and the extent of the area where the input is applied). Interestingly, we found that synaptic factors lower than f_{syn}^{3D} can lead to spectra with peaks in the alpha band if two simultaneous inputs are additionally applied to the system, one in the basal domain and another in the distal dendrite. For all the three synaptic inputs we have considered the same synaptic factor. Many other configurations could be investigated involving simultaneous inputs in several locations and with different intensity.

A systematic study on the PSDs has been carried out for different values of synaptic factors, synaptic locations and dendritic length constant. This study revealed that higher factors are needed to reproduce the alpha rhythmicity in systems where the synaptic input is progressively moved away from the soma, and confirmed the physiological role of the dendritic length constant in dendritic dynamics[1]. Results also suggest that higher factors, different parameters and multiple inputs can generate alphoid spectrograms for distal synaptic input locations.

One of the topics that could be explored in the future is the implementation of more realistic synaptic configurations where inhibitory and excitatory inputs are applied separately in one or multiple locations. This would allow one to consider inhibitory populations located in different cortical layers.

In addition, the results from NFMs including the dendritic dimension could be compared to experimental data from local field potential (LFP) measurements.

Our 3D NF model is a valid starting point for any study that wants to include the dendritic properties in the NF cortical dynamics. As an example, one can develop a procedure for assimilating to the 3D-NFM experimental data from depth electrode recordings. Other synaptic properties, depending on the dendritic tree, could be

[1] A larger dendritic length constant enhances the propagation of the signal along the cable.

potentially included in the model, together with a more complex structure for the dendritic cable. As an example, one could take into account different physiological values for the dendritic length constant, depending on more realistic dendritic tree geometries). Our expectation is that, due to their capability to incorporate a more detailed dendritic microcircuitry and experimental depth data, 3D-NFMs will be widely used in the future.

Chapter 8

Appendix A

8.1 Time series, non-linear and linear prediction

The following scripts and functions illustrate how the 3D model with one or multiple microcolumns have been implemented and their activity studied in terms of non-linear and linear power spectral density.

8.1.1 single microcolumn 3D system

Script *main_3D*

- The parameters defining the dimension of the grid and the geometry of the dendrite (number of compartments of the discretization, location of soma and synaptic input) are assigned. Other parameters of the system are defined using the function *par_set*.
- The stable point of the system is calculated using the function *sing_point*.
- Specifications for Euler method are assigned.
- Parameters for the cortical input modelled as white noise are defined. The input is defined inside the function *system_3D*.
- Finally, the a 200s simulation in run using the forward Euler method.

Function *par_set*

The parameters of the system are defined, according to Table 3.3.

Function *sing_point*

The stable point of the system is calculated using a procedure described in Section 3.2.9.

- The parameters of the system are defined.
- The membrane potential, both excitatory and inhibitory, are initialised to -62mV, while all the other variables are set to zero.
- The system is run for 3 seconds and temporary "stable" values are assigned to all variables. Usually, the temporary value is the mean of the last 0.5 seconds of the first test simulation.
- A loop of test simulation is run, until the convergence rate of the membrane potential h_e is lower than 10^{-4} (this means that the difference in between the new "stable" value for h_e and the current initial condition for h_e is less than 10^{-4}). The initial condition for each simulation is the temporary "stable" point of the previous test simulation. At each iteration a new stationary point is calculated. The loop ends is the convergence condition is not satisfied within 5 iterations, meaning that the system most likely exhibiting a limit cycle, rather than converging to a stable point.

Function *forward_euler*

This function is implemented to numerically solve ordinary differential equation (ODE) systems of the form

$$\mathbf{y}(t)' = \mathbf{F}(t)$$

using forward Euler method [178]. The function $\mathbf{F}$, the initial condition $y(0)$, the number N of iteration, the time interval and the size of the output $\mathbf{y}$ are opportunely defined as input parameters (see comments in the code for more explanations). The output are the solution $\mathbf{y}(T)$ and the time vector T.

Function *system_3D*

This function defines the 3D system

$$\mathbf{y}(t)' = \mathbf{F}(t) + \mathbf{P}$$

for one single microcolumn. The system is solved using the forward Euler method (see script *main_3D* and function *forward_euler*). The input parameters of the function are

- variable "t" representing time
- the state variable "y"
- the set "param_set" of parameters for the system
- the parameters "stdev_pee, p_eet, Noise" used to define the extracortical input as white noise

For more details, see the comments in the code.

Function *jacobian_and_linear_PSD*

This function calculates:

- the Jacobian matrix "J" of the linearized system

  $$\mathbf{S}'(t) = \mathbf{J}\ \mathbf{s} + \mathbf{P}$$

  where J is the Jacobian evaluated at the singular point "SP", $\mathbf{s}$ is the perturbation defined as $\mathbf{s} = \mathbf{S} - \mathbf{S}^*$ (S^* is the singular point "SP") and P is the extracortical input vector. The state variable $\mathbf{S}$ is defined as a symbolic vector (see more details in the code comments) and the Jacobian is calculated using the MatLab function *jacobian*;

- the linear spectrum "Spe" of single microcolumn system using formula (3.73); right and left eigen-matrices and eigen-values are previously computed using the MatLab functions *eig* and *diag*;

- the linear PSD "Spe_norm", normalised with respect to the area under the spectrum curve.

Input parameters are:
- the angular frequency "omega" defined as $\omega = 2\pi f$, where f is the frequency
- the singular point "SP"
- the set of parameters "param_set"
- the number "nex" of excitatory dendritic segments
- the input location "ind_input"
- the somatic location "ihe_soma"
- the electrotonic length constant "lam"

Function *manual_jacobian*

This function calculates:

- the Jacobian matrix "J" of the linearized system

$$\mathbf{S}'(t) = \mathbf{J}\ \mathbf{s} + \mathbf{P} \tag{8.1}$$

  where J is the Jacobian evaluated at the singular point "SP", $\mathbf{s}$ is the perturbation defined as $\mathbf{s} = \mathbf{S} - \mathbf{S}^*$ (S^* is the singular point "SP") and P is the extracortical input vector. The Jacobian is calculated from scratch, by computing the all the derivatives

$$\frac{\partial \mathbf{F}(\mathbf{S})}{\partial \mathbf{S}}$$

- the linear spectrum "Spe" of single microcolumn system using formula (3.73); right and left eigen-matrices and eigen-values are previously computed using the MatLab functions *eig* and *diag*;

- the linear PSD "Spec_norm", normalised with respect to the area under the spectrum curve.

Input parameters are:
- the singular point "SP"
- the set of parameters "param_set"

```
%************************************************
%% Function "jacobian_and_linear_PSD"
%************************************************

%-------------------------------------------------------------------------------------------
%% This function calculates the Jacobian "J" and the linear spectrum for a system with
% one microcolumn, using the formula 3.76 of the thesis (the 2 spectra
% are identical). The linearised system is dS/dt = J*s + P, where
        % - P is the extracortical
        % - s is the perturbation s = S-S^* (S^* is the singular point "SP")
        % - J is the Jacobian matrix evaluated at a singular point "SP".
%  The Jacobian is calculated using the MatLab function "jacobian".
%-------------------------------------------------------------------------------------------

%-------------------------------------------------------------------------------------------
%% State variable S
%-------------------------------------------------------------------------------------------
        % The state variable S is defined as a symbolic vector

        % S = [he_1 he_2 ... he_nex hi S3 S14 S15 ... S13 S14]

        % -------
        % he
        % -------
        % he(i) = he over the dendrite for i in {1,...,nex}
        % (note that hhe(i) = he(i)' (transpose), see below)


        % -------
        % hi
        % -------
        % hi(1) = hi


        % -------
        % I_lk
        % -------
        % S(3) = Iee, S(4) = Iee_t  S(5) = Iei, S(6) = Iei_t
        % S(7) = Iie, S(8) = Iie_t, S(9) = Iii, S(10) = Iii_t


        % -------
        % Phi_ek
        % -------
        % S(11) = Phi_ee ,  S(12) = Phi_eet, S(13) = Phi_ei  ,S(14) = Phi_eit


%-------------------------------------------------------------------------------------------
%% Inputs - Outputs
%-------------------------------------------------------------------------------------------
        % "param_set" is the set of parameters of the system
        % "omega" is the angular frequency, defined by omega = 2*pi*f, where f is the
        % frequency
        % "SP" is the singular point

        % The Jacobian matrix is calculated using the MatLab function "jacobian"
        % The spectrum "Spe" is calculated using the formula 3.76 of my thesis
        % The normalised spectrum "Spe_norm" is computed dividing the Spectrum by
```

```
        % its area

        % "nex" is the number of dendritic excitatory compartments

        % "ind_input" is the index for the synaptic input
        % "ihe_soma" is the index for the somatic compartment
        % "lam" is the dendritic length constant

        % "J" is the Jacobian matrix evaluated at the singular point SP
%-------------------------------------------------------------------------------------------

function [Spe, Spe_norm, J] = jacobian_and_linear_PSD(omega,SP,param_set,nex,ind_input,↙
ihe_soma,lam)


%-------------------------------------------------------------------------------------------
%% Parameters
%-------------------------------------------------------------------------------------------

        %% system dimension
        n_var = nex + 13; % number of variables for each microcolumn

        %% constants from file

        % mV
        her = param_set(1);
        hir = param_set(2);

        % mV
        q1 = param_set(3); % h_ee_eq
        q2 = param_set(4); % h_ei_eq
        q3 = param_set(5); % h_ie_eq
        q4 = param_set(6); % h_ii_eq

        %mV
        Gamma1 = param_set(7);
        Gamma2 = param_set(8);
        Gamma3 = param_set(9);
        Gamma4 = param_set(10);
        Gamma = [Gamma1 Gamma2 Gamma3 Gamma4];

        % 1/s
        gamma1 = param_set(11);
        gamma2 = param_set(12);
        gamma3 = param_set(13);
        gamma4 = param_set(14);
        gamma = [gamma1 gamma2 gamma3 gamma4];

        % 1
        Nb1 = param_set(15);
        Nb2 = param_set(16);
        Nb3 = param_set(17);
        Nb4 = param_set(18);

        % 1
        Na1 = param_set(19);
        Na2 = param_set(20);
```

```
        % 1/s
        Sem = param_set(21);
        Sim = param_set(22);

        % mV
        mue = param_set(23);
        mui = param_set(24);

        % mV
        se = param_set(25);
        si = param_set(26);

        pee = param_set(27);

        % s
        te = param_set(31);
        ti = param_set(32);

        % 1/cm
        lambda_ee = param_set(33);
        lambda_ei = param_set(34);
        lambda = [lambda_ee lambda_ei];

        e = exp(1);
        f = e*gamma.*Gamma;

        % cm/s
        v = param_set(35);


        vv = 1/dx^2*3/2*v^2*F;
        b = -sqrt(2);
        r_abs = 0;

        %% dendritic parameters
        % cm
        lenght = 0.4; % lenght of the dendrite
        dz = lenght/nex; % lenght of a compartment
        ce = her/te;
        Ke = lam^2/(te*dz^2);

%-------------------------------------------------------------------------------------------
%% Symbolic vectors
%-------------------------------------------------------------------------------------------

        % he
        he = sym('he%d_%d',[1, nex]);
        for i = 1:np
            hhe(:,i) = he(i,:);
        end

        % hi
        hi = sym('hi%d',[1,1]);

        % I_lk and Phi_ek
        S = sym('S%d_%d',[1 14]);

        % State variable
```

```
        SS = horzcat(he(1,1:nex),hi(1),S(1,3:14));

%-----------------------------------------------------------------------------------------
%% Symbolic Jacobian
%-----------------------------------------------------------------------------------------

    system =  [Ke*(hhe(2,i)-hhe(1,i))-1/te*hhe(1,i)+ce;
         Ke*(hhe(1,i)-2*hhe(2,i)+hhe(3,i))-1/te*hhe(2,i)+ce;
         Ke*(hhe(2,i)-2*hhe(3,i)+hhe(4,i))-1/te*hhe(3,i)+ce;
         Ke*(hhe(3,i)-2*hhe(4,i)+hhe(5,i))-1/te*hhe(4,i)+ce;
         Ke*(hhe(4,i)-2*hhe(5,i)+hhe(6,i))-1/te*hhe(5,i)+ce;
         Ke*(hhe(5,i)-2*hhe(6,i)+hhe(7,i))-1/te*hhe(6,i)+ce;
         Ke*(hhe(6,i)-2*hhe(7,i)+hhe(8,i))-1/te*hhe(7,i)+ce;
         Ke*(hhe(7,i)-2*hhe(8,i)+hhe(9,i))-1/te*hhe(8,i)+ce;
         Ke*(hhe(8,i)-2*hhe(9,i)+hhe(10,i))-1/te*hhe(9,i)+ce;
         Ke*(hhe(9,i)-2*hhe(10,i)+hhe(11,i))-1/te*hhe(10,i)+ce;
         Ke*(hhe(10,i)-2*hhe(11,i)+hhe(12,i))-1/te*hhe(11,i)+ce;
         Ke*(hhe(11,i)-2*hhe(12,i)+hhe(13,i))-1/te*hhe(12,i)+ce;
         Ke*(hhe(12,i)-2*hhe(13,i)+hhe(14,i))-1/te*hhe(13,i)+ce;
         Ke*(hhe(13,i)-2*hhe(14,i)+hhe(15,i))-1/te*hhe(14,i)+ce;
         Ke*(hhe(14,i)-2*hhe(15,i)+hhe(16,i))-1/te*hhe(15,i)+ce;
         Ke*(hhe(15,i)-2*hhe(16,i)+hhe(17,i))-1/te*hhe(16,i)+ce;
         Ke*(hhe(16,i)-2*hhe(17,i)+hhe(18,i))-1/te*hhe(17,i)+ce;
         Ke*(hhe(17,i)-2*hhe(18,i)+hhe(19,i))-1/te*hhe(18,i)+ce;
         Ke*(hhe(18,i)-2*hhe(19,i)+hhe(20,i))-1/te*hhe(19,i)+ce;
         Ke*(hhe(19,i)-2*hhe(20,i)+hhe(21,i))-1/te*hhe(20,i)+ce;
         Ke*(hhe(20,i)-2*hhe(21,i)+hhe(22,i))-1/te*hhe(21,i)+ce;
         Ke*(hhe(21,i)-2*hhe(22,i)+hhe(23,i))-1/te*hhe(22,i)+ce;
         Ke*(hhe(22,i)-2*hhe(23,i)+hhe(24,i))-1/te*hhe(23,i)+ce;
         Ke*(hhe(23,i)-2*hhe(24,i)+hhe(25,i))-1/te*hhe(24,i)+ce+1/te*((q1-he(i,↙
ind_input))/(abs(her-q1))*S(i,3)+(q3-he(i,ind_input))/(abs(q3-her))*S(i,7));
         Ke*(hhe(24,i)-2*hhe(25,i)+hhe(26,i))-1/te*hhe(25,i)+ce;
         Ke*(hhe(25,i)-2*hhe(26,i)+hhe(27,i))-1/te*hhe(26,i)+ce;
         Ke*(hhe(26,i)-2*hhe(27,i)+hhe(28,i))-1/te*hhe(27,i)+ce;
         Ke*(hhe(27,i)-2*hhe(28,i)+hhe(29,i))-1/te*hhe(28,i)+ce;
         Ke*(hhe(28,i)-2*hhe(29,i)+hhe(30,i))-1/te*hhe(29,i)+ce;
         Ke*(hhe(nex-1,i)-hhe(nex,i))-1/te*hhe(nex,i)+ce;
         1/ti*(-(hi(i)-hir)+(q2-hi(i))/(abs(q2-hir))*S(i,5)+(q4-hi(i))/(abs(q4-hir))*S(i,↙
9));
         S(i,4);
        -2*gamma(1)*S(i,4)-(gamma(1))^2*S(i,3)+f(1)*(Nb1*Sem./(1+(1-r_abs*Sem)*exp(b*(he↙
(i,ihe_soma)-mue)/se))+S(i,11));
        S(i,6);
        -2*gamma(2)*S(i,6)-(gamma(2))^2*S(i,5)+f(2)*(Nb2*Sem./(1+(1-r_abs*Sem)*exp(b*(he↙
(i,ihe_soma)-mue)/se))+S(i,13));
        S(i,8);
        -2*gamma(3)*S(i,8)-(gamma(3))^2*S(i,7)+f(3)*Nb3*Sim./(1+(1-r_abs*Sim)*exp(b*(hi↙
(i)-mui)/si));
        S(i,10);
        -2*gamma(4)*S(i,10)-(gamma(4))^2*S(i,9)+f(4)*Nb4*Sim./(1+(1-r_abs*Sim)*exp(b*(hi↙
(i)-mui)/si));
        S(i,12);
        -2*v*lambda(1)*S(i,12)+v^2*lambda(1)^2*Na1*Sem./(1+(1-r_abs*Sem)*exp(b*(he(i,↙
ihe_soma)-mue)/se))+vv(i,:)*S(:,11);
        S(i,14);
        -2*v*lambda(2)*S(i,14)+v^2*lambda(2)^2*Na2*Sem./(1+(1-r_abs*Sem)*exp(b*(he(i,↙
ihe_soma)-mue)/se))+vv(i,:)*S(:,13);];
```

```
    J = jacobian(system,SS);

%-------------------------------------------------------------------------------------------
%% Jacobian evaluated at the sungular point SP
%-------------------------------------------------------------------------------------------
    J_star = double(subs(J,SS,SP));

%-------------------------------------------------------------------------------------------
%% PSD prediction
%-------------------------------------------------------------------------------------------
        % eigenvalues
        [R,D,Lm] = eig(J_star);
        eig_values = diag(D);
        L = Lm';
        E = L*R;
        en = diag(E);

        % indexes
        i1 = ihe_soma; % index for somatic he
        i4 = i1+(nex-ihe_soma+3); % index for I_eet

        % matrix P, formula to calculate the PSD from the time Fourier trnsform
        % of the linearised system
        Pc = zeros(n_var,n_var,size(omega,2));
        Spe = zeros(1,size(omega,2));
        for j = 1:size(omega,2)
            Pc(:,:,j) = (R*diag(1./(en.*(omega(1,j)*1i-eig_values))))*L;
            Spe(1,j) = (abs(Pc(i1,i4,j))*abs(f(1)*0.1*pee))^2;
        end

        % single side spectrum
        Spe(2:end-1) = 2*Spe(2:end-1);

        % normalisation
        Spe = Spe./500;
        Spe_norm = Spe./area(Spe,1,501,0.5);

end
```

```
%**************************************
%% Function "par_set"
%**************************************

%-------------------------------------------------------------------------------------------
%% This function defines the parameters of the system, according to the table 3.3 of my
% thesis
%-------------------------------------------------------------------------------------------

function param_set = par_set(nex,ind_input,ihe_soma,fsyn,lam)

        %% Parameters
        her = -62.226;
        hir = -65.666;

        q1 = -18.038; % h_ee_eq
        q2 = -16.554; % h_ei_eq
        q3 = -81.796; % h_ie_eq
        q4 = -78.995; % h_ii_eq

        Gamma1 = 0.10631;
        Gamma2 = 0.64105;
        Gamma3 = 0.46477;
        Gamma4 = 0.28663;

        gamma1 = 291.50;
        gamma2= 697.76;
        gamma3 = 458.67;
        gamma4 = 82.330;

        Nb1 = 2185.8;
        Nb2 = 3749.8;
        Nb3 = 466.30;
        Nb4 = 160.69;

        Na1 = 4611.6;
        Na2 = 1372.4;

        Sem = 196.08;
        Sim = 454.40;

        mue = -45.104;
        mui = -43.910;

        se = 3.8420;
        si = 4.5793;

        pee = 6603.4;
        pei = 2625.7;
        pie = 0;
        pii = 0;

        te = 132.55/1000;
        ti = 135.91/1000;

        Lee = 0.92809;
        Lei = 0.92809;
        v = 684.24;
```

```
        b = -sqrt(2);
        r_abs = 0;

        %% dendritic parameters

        % cm
        lenght = 0.4; % lenght of the dendrite
        dz = lenght/nex; % lenght of a compartment

        % parameters used to define the dendritic system
        ce = her/te;
        Ke = lam^2/(te*dz^2);
        Be = -(1/te+2*Ke);

        %% param_set
        param_set = zeros(1,47+size(ind_input,2));
        param_set(1,1:6) = [her, hir, q1, q2, q3 ,q4 ];
        param_set(1,7:14) = [Gamma1, Gamma2, Gamma3, Gamma4, gamma1, gamma2, gamma3,↙
gamma4];
        param_set(1,15:20) = [Nb1, Nb2, Nb3, Nb4, Na1, Na2];
        param_set(1,21:24) = [Sem, Sim, mue, mui];
        param_set(1,25:30) = [se, si, pee, pei, pie, pii];
        param_set(1,31:35) = [te, ti, Lee, Lei, v];
        param_set(1,36:37) = [b, r_abs];
        param_set(1,38:40) = [ce, Ke, Be];
        param_set(1,41:44) = [dz, lenght, nex, nex + 2];
        param_set(1,45:47) = [lam, fsyn, ihe_soma];
        param_set(1,48:end) = ind_input;
end
```

```
%**************************************
%% Function "sing_point"
%**************************************

%-------------------------------------------------------------------------------------------
%% This function is calculating the singular point of one microcolumn 3D system
% A detailed description of the procedure is provided at the beginning of the Appendix
% and in section 3.6.1
%-------------------------------------------------------------------------------------------

function [stable_point,n_iter,T_test,Y_test] = sing_point(param_set)

    %% Dendritic parameters and dimension of the system
    nex = param_set(43);
    ihe_soma = param_set(47);
    num_var = nex + 13;
    np = 1; % number of points in the grid

    %% Parameters for numerical method
    pps = 10000; % points per second for the euler integration, time step = 1/pps
    eff_pps = 500; % points per second to be effectively saved
    tinit_sim = 0;
    tfinal_sim = 3;
    time_sim = tfinal_sim-tinit_sim;
    N_sim = time_sim*pps; % number of euler loop iterations
    % Initial Condition
    IC(1:nex+1) = -62;
    IC(nex+2:nex+13) = 0;


    %% Parameters for the extra-cortical input
    Noise = randn(N_sim,np);
    p_eet = linspace(0,tfinal_sim-tinit_sim,N_sim);
    stdev_pee = 0;

    %-------------------------------------------------------------------------------------------
    %% First test simulation and temporary stable point
    %-------------------------------------------------------------------------------------------

    disp('sp iteration')
    ii=0;
    disp(ii)
    [T_test,Y_test] = forward_euler(@(t,y) system_3D(t,y,param_set,stdev_pee, p_eet,↙
Noise),0,IC,time_sim,N_sim,pps,eff_pps);
    Y_test = Y_test';


    stable_point = zeros(1,num_var);
        for i = 1:num_var
            stable_point(1,i) = mean(Y_test(1251:1501,i));
        end

    %-------------------------------------------------------------------------------------------
    %% The test simulation is iteratively run until the convergence rate is less than
    % 10^-4, for a maximum of 5 iterations (otherwise a limit cycle is detected)
    %-------------------------------------------------------------------------------------------
        while (abs(stable_point(ihe_soma)-Y_test(1,ihe_soma)) > 10^-4 && ii<6)
            ii = ii+1;
```

```
            disp('sp iteration')
            disp(ii)

            % Parameters for numerical method
            tfinal_sim = 2; % duration of the test simulation
            time_sim = tfinal_sim-tinit_sim;
            N_sim = time_sim*pps; % number of euler loop iterations

            % Initial Conditions (IC)
            IC = stable_point; % The IC is the temporary stable point calculated in the↙
previous test simulation

            % Parameters for extra-cortical input
            Noise = randn(N_sim,np);
            p_eet = linspace(0,tfinal_sim-tinit_sim,N_sim);
            stdev_pee =0;

            % Test simulation
            [T_test,Y_test] = forward_euler(@(t,y) system_3D(t,y,param_set,stdev_pee,↙
p_eet, Noise),0,IC,time_sim,N_sim,pps,eff_pps);
            Y_test = Y_test';

            % Stable point update
            stable_point = zeros(1,num_var);
                for i = 1:num_var
                    stable_point(1,i) = mean(Y_test(501:1001,i));
                    %stable_point(1,i)= Y_test(end,i);
                end
        end
        n_iter = ii;
end
```

```
%**************************************
%% Function "forward_euler"
%**************************************

%% This function implements Euler's method to numerically solve the system y'(t) = F(t),
% with initial condition y(0) = yinit.

%-------------------------------------------------------------------------------------------
%% Input parameters
%-------------------------------------------------------------------------------------------
        % "F" is the system to be integrated,
        % "yinit" is the initial contition for y(t)
        % "tinit" and "tfinal" define the time interval (tfinal-tinit) of the simulation
        % "points_per_sec" is the number of Euler iterations per second
        % "N" is the total number of iteration (defined in "main_2D" as
        % "N =  time_sim*pps;" i.e. duration of the simulation (in seconds) * number
        % of iteration per second)

        % Note that the time step is defined by h = (tfinal - tinit)/N;

%-------------------------------------------------------------------------------------------
%% Output
%-------------------------------------------------------------------------------------------
        % "y" (function of time) is the solution of the system y'(t) = F(t)
        % "y_new" is the solution "y" resized
        % (while "y" has size N, the output "y_new", has lower size
        % "size(y_new) = time_sim * eff_points_per_sec", i.e. duration of the
        % simulation * number of points per second to be stored in y_new)

function [t_new,y_new] = forward_euler(F, tinit, yinit, tfinal, N, points_per_sec,↙
eff_points_per_sec)
    % Time step
    h = (tfinal - tinit)/N;
    % Points to be saved
    nmod = points_per_sec/eff_points_per_sec;

    % Initialising t, y, t_new and y_new
    t = [tinit zeros(1,N)];
    y = zeros(size(yinit,2),N+1);
    y(:,1) = yinit;
    t_new = [tinit zeros(1,N/nmod)];
    y_new = zeros(size(yinit,2),N/nmod + 1);
    y_new(:,1) = yinit;

    % Calculating y using Euler Method
    k = 1;
        for i = 1:N
            t(i+1) = t(i) + h;
            y(:,i+1) = y(:,i) + h*F(t(i),y(:,i));
            if (mod(i,nmod) == 0)
              y_new(:,k+1) = y(:,i+1);
              t_new(1,k+1) = t(i+1);
              k = k + 1;
            end
        end
end
```

```
%**************************************
%% Function "system_3D"
%**************************************

%-------------------------------------------------------------------------------------------
%% This function defines the 3D system y'(t) = F(t) for one single microcolumn.
% This system is then numerically solved to find the solution y(t)
%-------------------------------------------------------------------------------------------

%-------------------------------------------------------------------------------------------
%% Dendritic parameters
%-------------------------------------------------------------------------------------------
        % "nex" is the number of excitatory compartments
        % "ihe_soma" is the index for the soma
        % "ind_input" is the index for the location of the input
        % "f_syn" is the synaptic factor
        % "lam" is the dendritic length constant

%-------------------------------------------------------------------------------------------
%% Inputs
%-------------------------------------------------------------------------------------------

        %% state variale "y"
        % y is a vector variable, whose components are the variables describing one
        % microcolumn:

        % ---(mean excitatory membrane potential)---
        % y(1) = h_e (first compartment)
        % ...
        % y(ind_input) =  h_e (at input location)
        % ...
        % y(ihe_soma) = h_e (at the soma)
        % ...
        % y(nex) = h_e (last compartment)

        % ---(mean inhibitory membrane potential)---
        %  y(nex + 1)= h_i

        %  ---(synaptic currents)---
        % y(nex+2) = Iee, y(nex+3) = Iee_t  y(nex+4) = Iei, y(nex+5) = Iei_t
        % y(nex+6) = Iie, y(nex+7) = Iie_t, y(nex+8) = Iii, y(nex+9) = Iii_t
        % ---(cortical inputs)---
        % y(nex+10) = Phi_ee ,  y(nex+11) = Phi_eet, y(nex+12) = Phi_ei  ,y(nex+13) =↙
Phi_eit


        %% "t" is the time variable

        %% "param_set" is the set of parameters

        %% "stdev_pee, p_eet, Noise" are the parameters to define the extra-cortical
        % input as white noise.
%-------------------------------------------------------------------------------------------

function dy = system_3D(t,y,param_set,stdev_pee, p_eet,Noise)

        %% Dendritic parameters
        nex = param_set(43);
```

```
ihe_soma = param_set(47);
ind_input = param_set(48:end);
fsyn = param_set(46);

%% Parameters

% number of variables describing one microcolumn (size of y)
num_var = nex + 13;

% s
tau_e = param_set(31);
tau_i = param_set(32);

% mV
h_er = param_set(1);
h_ir = param_set(2);

% mV
h_eeeq = param_set(3);
h_eieq = param_set(4);
h_ieeq = param_set(5);
h_iieq = param_set(6);
h_eq = [h_eeeq h_eieq h_ieeq h_iieq];

% 1/s
gamma_ee = param_set(11);
gamma_ei = param_set(12);
gamma_ie = param_set(13);
gamma_ii = param_set(14);
gamma = [gamma_ee gamma_ei gamma_ie gamma_ii];

% 1
N_beta_ee = param_set(15);
N_beta_ei = param_set(16);
N_beta_ie = param_set(17);
N_beta_ii = param_set(18);
N_beta = [N_beta_ee N_beta_ei N_beta_ie N_beta_ii];

% 1/s
S_max_e = param_set(21);
S_max_i = param_set(22);
S_max = [S_max_e S_max_i];

% mV
mu_bar_e = param_set(23);
mu_bar_i = param_set(24);
mu_bar = [mu_bar_e mu_bar_i];

% mV
sigmac_e = param_set(25);
sigmac_i = param_set(26);
sigmac = [sigmac_e sigmac_i];

% 1/cm
lambda_ee = param_set(33);
lambda_ei = param_set(34);
lambda = [lambda_ee lambda_ei];
```

```
% mV
Gamma_ee = param_set(7);
Gamma_ei = param_set(8);
Gamma_ie = param_set(9);
Gamma_ii = param_set(10);
Gamma = [Gamma_ee Gamma_ei Gamma_ie Gamma_ii];

% cm/s
v = param_set(35);

% 1
N_alfa_ee = param_set(19);
N_alfa_ei = param_set(20);
N_alfa = [N_alfa_ee N_alfa_ei];

%
f = exp(1)*gamma.*Gamma;

% cm
% dx = 0.1;
% auxiliary parameter
% vv = 1/dx^2*3/2*v^2*F;


%% Extra-cortical input implemented as white noise
% 1/s
p_eebar = 6603.4;
p_ee = (p_eebar*(1+stdev_pee*interp1(p_eet,Noise,t)))';
p_ee(p_ee<0) = 0;
p_ei = 2625.7;
p_ie = 0;
p_ii =0;

%% Auxiliary dendritic parameters
ce = param_set(38);
Ke = param_set(39);
%Be = param_set(40);

%% INDEXES to access all the variables of the state vector y
i1 = ihe_soma; % somatic he
i2 = nex+1; % hi
i3 = nex+2; % Iee
i4 = nex+3; % Ieet
i5 = nex+4; % Iei
i6 = nex+5; % Ieit
i7 = nex+6; % Iie
i8 = nex+7; % Iiet
i9 = nex+8; % Iii
i10 = nex+9; % Iiit
i11 = nex+10; % Phi_ee
i12 = nex+11; % Phi_eet
i13 = nex+12; % Phi_ei
i14 = nex+13; % Phi_eit

%% Dendritic matrix and Input
Ae = zeros(nex-2,nex);
indhe = 1:1:nex;
for j  = 1:nex-2
```

```
            Ae(j,j) = 1;
            Ae(j,j+1) = -2;
            Ae(j,j+2) = 1;
        end
        Input = fsyn*(psi(y(ind_input),h_eq(1),h_er).*y(i3) + psi(y(ind_input),h_eq(3),↙
h_er).*y(i7));
        Je = zeros(nex,1);
        Je(ind_input) = Input;
        % note that Je has size nex, then Je(ind_input) = Input. However, when
        % summed in the system we consider Je(2:end-1)

        %↙
------------------------------------------------------------------------------------------
        %% ONE MICROCOLUMN 3D SYSTEM
        %↙
------------------------------------------------------------------------------------------
        dy = zeros(num_var,1);

        % excitatory mean membrane potential he along the dendrite with sealed end↙
boundary conditions
        dy(1) = Ke*(y(2)-y(1)) -1/tau_e*y(1)+ce;
        dy(2:nex-1) = Ke*(Ae*y(indhe))-1/tau_e*y(2:nex-1) + ce + 1/tau_e*Je(2:end-1);
        dy(nex) = Ke*(y(nex-1)-y(nex)) -1/tau_e*y(nex)+ce;

        % hi
        dy(i2) = 1/tau_i*(-(y(i2)-h_ir)+ psi(y(i2),h_eq(2),h_ir)*y(i5)+ psi(y(i2),h_eq↙
(4),h_ir)*y(i9));

        % I_lk
        dy(i3) = y(i4);
        dy(i4) = -2*gamma(1)*y(i4)-(gamma(1))^2*y(i3)+f(1)*(N_beta(1)*S(y(i1),S_max(1),↙
mu_bar(1),sigmac(1))+y(i11)+p_ee);
        dy(i5) = y(i6);
        dy(i6) = -2*gamma(2)*y(i6)-(gamma(2))^2*y(i5)+f(2)*(N_beta(2)*S(y(i1),S_max(1),↙
mu_bar(1),sigmac(1))+y(i13)+p_ei);
        dy(i7) = y(i8);
        dy(i8) = -2*gamma(3)*y(i8)-(gamma(3))^2*y(i7)+f(3)*(N_beta(3)*S(y(i2),S_max(2),↙
mu_bar(2),sigmac(2))+p_ie);
        dy(i9) = y(i10);

        % Phi_ek
        dy(i10) = -2*gamma(4)*y(i10)-(gamma(4))^2*y(i9)+f(4)*(N_beta(4)*S(y(i2),S_max(2),↙
mu_bar(2),sigmac(2))+p_ii);
        dy(i11) = y(i12);
        dy(i12) = -2*v*lambda(1)*y(i12)-v^2*lambda(1)^2*y(i11)+v^2*lambda(1)^2*N_alfa(1)↙
*S(y(i1),S_max(1),mu_bar(1),sigmac(1)); %+vv*y(i11);
        dy(i13) = y(i14);
        dy(i14) = -2*v*lambda(2)*y(i14)-v^2*lambda(2)^2*y(i13)+v^2*lambda(2)^2*N_alfa(2)↙
*S(y(i1),S_max(1),mu_bar(1),sigmac(1)); %+vv*y(i13);
        %↙
------------------------------------------------------------------------------------------
end

%% Auxiliary functions: psi and S, defined by Eqs. 3.5 and 3.6 of my thesis
function y = psi(h,eq,r)
    y = (eq-h)/abs(eq-r);
end
```

```
function y = S(h,S_max,mu_ba,sigmac)
    b = -sqrt(2);
    r_abs=0;
    % y = 0;
    y = S_max./(1+(1-r_abs*S_max)*exp(b*(h-mu_ba)/sigmac));
end
```

```
%***********************************************
%% Function "jacobian_and_linear_PSD"
%***********************************************

%-------------------------------------------------------------------------------------------
%% This function calculates the Jacobian "J" and the linear spectrum for a system with
% one microcolumn, using the formula 3.73 of the thesis (the 2 spectra
% are identical). The linearised system is dS/dt = J*s + P, where
        % - P is the extracortical
        % - s is the perturbation s = S-S^* (S^* is the singular point "SP")
        % - J is the Jacobian matrix evaluated at a singular point "SP".
%  The Jacobian is calculated using the MatLab function "jacobian".
%-------------------------------------------------------------------------------------------

%-------------------------------------------------------------------------------------------
%% State variable S
%-------------------------------------------------------------------------------------------
        % The state variable S is defined as a symbolic vector

        % S = [he_1 he_2 ... he_nex hi S3 S14 S15 ... S13 S14]

        % -------
        % he
        % -------
        % he(i) = he over the dendrite for i in {1,...,nex}
        % (note that hhe(i) = he(i)' (transpose), see below)


        % -------
        % hi
        % -------
        % hi(1) = hi


        % -------
        % I_lk
        % -------
        % S(3) = Iee, S(4) = Iee_t  S(5) = Iei, S(6) = Iei_t
        % S(7) = Iie, S(8) = Iie_t, S(9) = Iii, S(10) = Iii_t


        % -------
        % Phi_ek
        % -------
        % S(11) = Phi_ee ,  S(12) = Phi_eet, S(13) = Phi_ei  ,S(14) = Phi_eit


%-------------------------------------------------------------------------------------------
%% Inputs - Outputs
%-------------------------------------------------------------------------------------------
        % "param_set" is the set of parameters of the system
        % "omega" is the angular frequency, defined by omega = 2*pi*f, where f is the
        % frequency
        % "SP" is the singular point

        % The Jacobian matrix is calculated using the MatLab function "jacobian"
        % The spectrum "Spe" is calculated using the formula 3.73 of my thesis
        % The normalised spectrum "Spe_norm" is computed dividing the Spectrum by
```

```
        % its area

        % "nex" is the number of dendritic excitatory compartments

        % "ind_input" is the index for the synaptic input
        % "ihe_soma" is the index for the somatic compartment
        % "lam" is the dendritic length constant

        % "J" is the Jacobian matrix evaluated at the singular point SP
%-------------------------------------------------------------------------------------------

function [Spe, Spe_norm, J] = jacobian_and_linear_PSD(omega,SP,param_set,nex,ind_input,↙
ihe_soma,lam)


%-------------------------------------------------------------------------------------------
%% Parameters
%-------------------------------------------------------------------------------------------

        %% system dimension
        n_var = nex + 13; % number of variables for each microcolumn

        %% constants from file

        % mV
        her = param_set(1);
        hir = param_set(2);

        % mV
        q1 = param_set(3); % h_ee_eq
        q2 = param_set(4); % h_ei_eq
        q3 = param_set(5); % h_ie_eq
        q4 = param_set(6); % h_ii_eq

        %mV
        Gamma1 = param_set(7);
        Gamma2 = param_set(8);
        Gamma3 = param_set(9);
        Gamma4 = param_set(10);
        Gamma = [Gamma1 Gamma2 Gamma3 Gamma4];

        % 1/s
        gamma1 = param_set(11);
        gamma2 = param_set(12);
        gamma3 = param_set(13);
        gamma4 = param_set(14);
        gamma = [gamma1 gamma2 gamma3 gamma4];

        % 1
        Nb1 = param_set(15);
        Nb2 = param_set(16);
        Nb3 = param_set(17);
        Nb4 = param_set(18);

        % 1
        Na1 = param_set(19);
        Na2 = param_set(20);
```

```
        % 1/s
        Sem = param_set(21);
        Sim = param_set(22);

        % mV
        mue = param_set(23);
        mui = param_set(24);

        % mV
        se = param_set(25);
        si = param_set(26);

        pee = param_set(27);

        % s
        te = param_set(31);
        ti = param_set(32);

        % 1/cm
        lambda_ee = param_set(33);
        lambda_ei = param_set(34);
        lambda = [lambda_ee lambda_ei];

        e = exp(1);
        f = e*gamma.*Gamma;

        % cm/s
        v = param_set(35);


        vv = 1/dx^2*3/2*v^2*F;
        b = -sqrt(2);
        r_abs = 0;

        %% dendritic parameters
        % cm
        lenght = 0.4; % lenght of the dendrite
        dz = lenght/nex; % lenght of a compartment
        ce = her/te;
        Ke = lam^2/(te*dz^2);

%-------------------------------------------------------------------------------------------
%% Symbolic vectors
%-------------------------------------------------------------------------------------------

        % he
        he = sym('he%d_%d',[1, nex]);
        for i = 1:np
            hhe(:,i) = he(i,:);
        end

        % hi
        hi = sym('hi%d',[1,1]);

        % I_lk and Phi_ek
        S = sym('S%d_%d',[1 14]);

        % State variable
```

```
        SS = horzcat(he(1,1:nex),hi(1),S(1,3:14));

%-------------------------------------------------------------------------------------------
%% Symbolic Jacobian
%-------------------------------------------------------------------------------------------

    system =  [Ke*(hhe(2,i)-hhe(1,i))-1/te*hhe(1,i)+ce;
         Ke*(hhe(1,i)-2*hhe(2,i)+hhe(3,i))-1/te*hhe(2,i)+ce;
         Ke*(hhe(2,i)-2*hhe(3,i)+hhe(4,i))-1/te*hhe(3,i)+ce;
         Ke*(hhe(3,i)-2*hhe(4,i)+hhe(5,i))-1/te*hhe(4,i)+ce;
         Ke*(hhe(4,i)-2*hhe(5,i)+hhe(6,i))-1/te*hhe(5,i)+ce;
         Ke*(hhe(5,i)-2*hhe(6,i)+hhe(7,i))-1/te*hhe(6,i)+ce;
         Ke*(hhe(6,i)-2*hhe(7,i)+hhe(8,i))-1/te*hhe(7,i)+ce;
         Ke*(hhe(7,i)-2*hhe(8,i)+hhe(9,i))-1/te*hhe(8,i)+ce;
         Ke*(hhe(8,i)-2*hhe(9,i)+hhe(10,i))-1/te*hhe(9,i)+ce;
         Ke*(hhe(9,i)-2*hhe(10,i)+hhe(11,i))-1/te*hhe(10,i)+ce;
         Ke*(hhe(10,i)-2*hhe(11,i)+hhe(12,i))-1/te*hhe(11,i)+ce;
         Ke*(hhe(11,i)-2*hhe(12,i)+hhe(13,i))-1/te*hhe(12,i)+ce;
         Ke*(hhe(12,i)-2*hhe(13,i)+hhe(14,i))-1/te*hhe(13,i)+ce;
         Ke*(hhe(13,i)-2*hhe(14,i)+hhe(15,i))-1/te*hhe(14,i)+ce;
         Ke*(hhe(14,i)-2*hhe(15,i)+hhe(16,i))-1/te*hhe(15,i)+ce;
         Ke*(hhe(15,i)-2*hhe(16,i)+hhe(17,i))-1/te*hhe(16,i)+ce;
         Ke*(hhe(16,i)-2*hhe(17,i)+hhe(18,i))-1/te*hhe(17,i)+ce;
         Ke*(hhe(17,i)-2*hhe(18,i)+hhe(19,i))-1/te*hhe(18,i)+ce;
         Ke*(hhe(18,i)-2*hhe(19,i)+hhe(20,i))-1/te*hhe(19,i)+ce;
         Ke*(hhe(19,i)-2*hhe(20,i)+hhe(21,i))-1/te*hhe(20,i)+ce;
         Ke*(hhe(20,i)-2*hhe(21,i)+hhe(22,i))-1/te*hhe(21,i)+ce;
         Ke*(hhe(21,i)-2*hhe(22,i)+hhe(23,i))-1/te*hhe(22,i)+ce;
         Ke*(hhe(22,i)-2*hhe(23,i)+hhe(24,i))-1/te*hhe(23,i)+ce;
         Ke*(hhe(23,i)-2*hhe(24,i)+hhe(25,i))-1/te*hhe(24,i)+ce+1/te*((q1-he(i,↙
ind_input))/(abs(her-q1))*S(i,3)+(q3-he(i,ind_input))/(abs(q3-her))*S(i,7));
         Ke*(hhe(24,i)-2*hhe(25,i)+hhe(26,i))-1/te*hhe(25,i)+ce;
         Ke*(hhe(25,i)-2*hhe(26,i)+hhe(27,i))-1/te*hhe(26,i)+ce;
         Ke*(hhe(26,i)-2*hhe(27,i)+hhe(28,i))-1/te*hhe(27,i)+ce;
         Ke*(hhe(27,i)-2*hhe(28,i)+hhe(29,i))-1/te*hhe(28,i)+ce;
         Ke*(hhe(28,i)-2*hhe(29,i)+hhe(30,i))-1/te*hhe(29,i)+ce;
         Ke*(hhe(nex-1,i)-hhe(nex,i))-1/te*hhe(nex,i)+ce;
         1/ti*(-(hi(i)-hir)+(q2-hi(i))/(abs(q2-hir))*S(i,5)+(q4-hi(i))/(abs(q4-hir))*S(i,↙
9));
         S(i,4);
        -2*gamma(1)*S(i,4)-(gamma(1))^2*S(i,3)+f(1)*(Nb1*Sem./(1+(1-r_abs*Sem)*exp(b*(he↙
(i,ihe_soma)-mue)/se))+S(i,11));
        S(i,6);
        -2*gamma(2)*S(i,6)-(gamma(2))^2*S(i,5)+f(2)*(Nb2*Sem./(1+(1-r_abs*Sem)*exp(b*(he↙
(i,ihe_soma)-mue)/se))+S(i,13));
        S(i,8);
        -2*gamma(3)*S(i,8)-(gamma(3))^2*S(i,7)+f(3)*Nb3*Sim./(1+(1-r_abs*Sim)*exp(b*(hi↙
(i)-mui)/si));
        S(i,10);
        -2*gamma(4)*S(i,10)-(gamma(4))^2*S(i,9)+f(4)*Nb4*Sim./(1+(1-r_abs*Sim)*exp(b*(hi↙
(i)-mui)/si));
        S(i,12);
        -2*v*lambda(1)*S(i,12)+v^2*lambda(1)^2*Na1*Sem./(1+(1-r_abs*Sem)*exp(b*(he(i,↙
ihe_soma)-mue)/se))+vv(i,:)*S(:,11);
        S(i,14);
        -2*v*lambda(2)*S(i,14)+v^2*lambda(2)^2*Na2*Sem./(1+(1-r_abs*Sem)*exp(b*(he(i,↙
ihe_soma)-mue)/se))+vv(i,:)*S(:,13);];
```

```
    J = jacobian(system,SS);

%-------------------------------------------------------------------------------------------
%% Jacobian evaluated at the sungular point SP
%-------------------------------------------------------------------------------------------
    J_star = double(subs(J,SS,SP));

%-------------------------------------------------------------------------------------------
%% PSD prediction
%-------------------------------------------------------------------------------------------
        % eigenvalues
        [R,D,Lm] = eig(J_star);
        eig_values = diag(D);
        L = Lm';
        E = L*R;
        en = diag(E);

        % indexes
        i1 = ihe_soma; % index for somatic he
        i4 = i1+(nex-ihe_soma+3); % index for I_eet

        % matrix P, formula to calculate the PSD from the time Fourier trnsform
        % of the linearised system
        Pc = zeros(n_var,n_var,size(omega,2));
        Spe = zeros(1,size(omega,2));
        for j = 1:size(omega,2)
            Pc(:,:,j) = (R*diag(1./(en.*(omega(1,j)*1i-eig_values))))*L;
            Spe(1,j) = (abs(Pc(i1,i4,j))*abs(f(1)*0.1*pee))^2;
        end

        % single side spectrum
        Spe(2:end-1) = 2*Spe(2:end-1);

        % normalisation
        Spe = Spe./500;
        Spe_norm = Spe./area(Spe,1,501,0.5);

end
```

```
%%************************************
%% Function "manual_jacobian"
%%************************************

%%-------------------------------------------------------------------------------------
%% This function calculates the Jacobian "J" and the spectrum "Spectrum" for a system↙
with
% one microcolumn, using the formula 3.73 of the thesis (the 2 spectra
% are identical). The linearised system is dS/dt = J*s + P, where
        % - P is the extracortical
        % - s is the perturbation s = S-S^* (S^* is the singular point "SP")
        % - J is the Jacobian matrix evaluated at a singular point "SP".

% The Jacobian is calculated manually,
% i.e. explicitely calculating the derivatives
% \partial (F(S))/ \partial S = \partial (\partial St/ partial t)/ \partial S
%-------------------------------------------------------------------------------------

%%-------------------------------------------------------------------------------------
%% Inputs - Outputs
%%-------------------------------------------------------------------------------------
        % "param_set" is the set of parameters of the system
        % "SP" is the singular point

        % S is the state variable of the system, where
        % S(1) = he at the soma, S(2) = hi
        % S(3) = Iee, S(4) = Iee_t  S(5) = Iei, S(6) = Iei_t
        % S(7) = Iie, S(8) = Iie_t, S(9) = Iii, S(10) = Iii_t
        % S(11) = Phi_ee ,  S(12) = Phi_eet, S(13) = Phi_ei  ,S(14) = Phi_eit

        % The outputs are "J", "Spectrum" and "Spec_norm" that is the
        % spectrum normalised with the respect to the area under the curve

%%-------------------------------------------------------------------------------------


function [J, Spectrum, Spec_norm]= manual_jacobian(param_set, SP)

%%-------------------------------------------------------------------------------------
%% Parameters
%%-------------------------------------------------------------------------------------
    %% system dimension
    nex = param_set(43);
    num_var = nex + 13;

    %% singular point
    S = SP;

    %% dendritic parameters
    fsyn = param_set(46);
    ind_input = param_set(48:end); % indexes for the synaptic inputs
    ihe_soma = param_set(47);

    %% constants

    % mV
    her = param_set(1);
    hir = param_set(2);
```

```
% mV
q1 = param_set(3); % h_ee_eq
q2 = param_set(4); % h_ei_eq
q3 = param_set(5); % h_ie_eq
q4 = param_set(6); % h_ii_eq

%mV
Gamma1 = param_set(7);
Gamma2 = param_set(8);
Gamma3 = param_set(9);
Gamma4 = param_set(10);

% 1/s
gamma1 = param_set(11);
gamma2 = param_set(12);
gamma3 = param_set(13);
gamma4 = param_set(14);

% 1
Nb1 = param_set(15);
Nb2 = param_set(16);
Nb3 = param_set(17);
Nb4 = param_set(18);

% 1
Na1 = param_set(19);
Na2 = param_set(20);

% 1/s
Sem = param_set(21);
Sim = param_set(22);

% mV
mue = param_set(23);
mui = param_set(24);

% mV
se = param_set(25);
si = param_set(26);

pee = param_set(27);
% pei = param_set(28);
% pie = param_set(29);
% pii = param_set(30);

% s
te = param_set(31);
ti = param_set(32);

% 1/cm
lambda1 = param_set(33);
lambda2 = param_set(34);

e = exp(1);

% cm/s
v = param_set(35);
```

```
    b = param_set(36);
    % r_abs = param_set(37);

    % ce = param_set(38); % ce = her/te
    Ke = param_set(39);


    % dz = param_set(41);
    % lenght = param_set(42);

    %% indexes to access all variables
    i1 = ihe_soma;
    i2 = nex+1;
    i3 = nex+2;
    i4 = nex+3;
    i5 = nex+4;
    i6 = nex+5;
    i7 = nex+6;
    i8 = nex+7;
    i9 = nex+8;
    i10 = nex+9;
    i11 = nex+10;
    i12 = nex+11;
    i13 = nex+12;
    i14 = nex+13;

%-------------------------------------------------------------------------------------------
%% Jacobian
%-------------------------------------------------------------------------------------------

    J = zeros(num_var,num_var);

    %-----------------------
    %% Dendritic derivatives
    %-----------------------

        %% dF1:nex_dS1:nex
        J(1:nex,1:nex) = diag((-1/te-2*Ke)*ones(nex,1)) + diag(Ke*ones(nex-1,1),1)+diag↙
(Ke*ones(nex-1,1),-1);
        J(1,1) = -1/te-Ke;
        J(nex,nex) = -1/te-Ke;

        %% dF(indinput)/d(hindinput)_S3_S7
        for j = 1:size(ind_input,2)
            J(ind_input(j),ind_input(j)) = -2*Ke-1/te + fsyn*1/te*(-S(i3)/abs(her-q1)-S↙
(i7)/abs(her-q3));
            J(ind_input(j),i3) = fsyn*1/te*(q1-S(ind_input(j)))/abs(her-q1);
            J(ind_input(j),i7) = fsyn*1/te*(q3-S(ind_input(j)))/abs(her-q3);
        end

    %-----------------------
    %% Other derivatives
    %-----------------------

        %% dFi2/dS2
        J(i2,i2) = dFi2_dhi(S(i5),S(i9),ti,hir,q2,q4);

        %% dFi2/dS5_9
```

```
J(i2,i5) = dFi2_dS59(S(i2), ti, q2, hir);
J(i2,i9) = dFi2_dS59(S(i2), ti, q4, hir);

%% dFi3/dS4
J(i3,i4) = 1;

%% dFi4/dS1_3_4_11_
J(i4,i1) = dI_dh(e,gamma1,Gamma1,Nb1,Sem,b,se,mue,S(i1));
%- e*gamma*Gamma*Nb1*Sem*b/se*exp(b*(S1-mue)/se)/(1 + exp(b(S1-mue)/se))^2;
J(i4,i3) = - gamma1^2;
J(i4,i4) = - 2*gamma1;
J(i4,i11) = e*gamma1*Gamma1;

%% dFi5/dS6
J(i5,i6) = 1;

%% dFi6/dS1_5_6_13
J(i6,i1) = dI_dh(e,gamma2,Gamma2,Nb2,Sem,b,se,mue,S(i1));
J(i6,i5) = - gamma2^2;
J(i6,i6) = - 2*gamma2;
J(i6,i13) = e*gamma2*Gamma2;

%% dFi7/dS8
J(i7,i8) = 1;

%% dFi8/dS2_7_8
J(i8,i2) = dI_dh(e,gamma3,Gamma3,Nb3,Sim,b,si,mui,S(i2));
J(i8,i7) = - gamma3^2;
J(i8,i8) = - 2*gamma3;

%% dFi9/dS10
J(i9,i10) = 1;

%% dFi10/dS2_9_10
J(i10,i2) = dI_dh(e,gamma4,Gamma4,Nb4,Sim,b,si,mui,S(i2));
J(i10,i9) = - gamma4^2;
J(i10,i10) = - 2*gamma4;

%% dFi11/dS12
J(i11,i12) = 1;

%% dFi12/dS1_11_12
J(i12,i1) = dPhi_dh(v,lambda1,Na1,Sem,b,se,mue,S(i1));
J(i12,i11) = - v^2*lambda1^2;
J(i12,i12) = - 2*v*lambda1;

%% dFi13/dS14
J(i13,i14) = 1;

%% dFi14/dS1_13_14
J(i14,i1) = dPhi_dh(v,lambda2,Na2,Sem,b,se,mue,S(i1));
J(i14,i13) = - v^2*lambda2^2;
J(i14,i14) = - 2*v*lambda2;

%% debugging
% if  (is_nan_inf(J) == 1)
%     nan_or_inf_J = 1;
%     freq = 0:0.5:250;
```

```
        %     omega = freq*2*pi;
        %     Spectrum = zeros(1,size(omega,2));
        %     eig_values_manual  = 0;
        %     return
        % else
        %     nan_or_inf_J = 0;
        % end

%-------------------------------------------------------------------------------------------
%% Eigenvalues and PSD calculations
%-------------------------------------------------------------------------------------------

    %% eig_values calculation
    [R,D,Lm] = eig(J);
    eig_values_manual = diag(D);

    %% calculation of the spectrum
    freq = 0:0.5:250;
    omega = freq*2*pi;

    L = Lm';
    E = L*R;
    en = diag(E);

    i_Ieet = nex + 3;

    Pc = zeros(nex+1+12,nex+1+12,size(omega,2));
    Spectrum = zeros(1,size(omega,2));
    for j = 1:size(omega,2)
        Pc(:,:,j) = (R*diag(1./(en.*(omega(1,j)*1i-eig_values_manual))))*L;
        Spectrum(1,j) = (abs(Pc(ihe_soma,i_Ieet,j))*abs(e*gamma1*Gamma1*0.1*pee))^2;
    end

    Spectrum = 2*Spectrum./500;

    % single side spectrum
    Spectrum(1) = Spectrum(1)/2;
    Spectrum(end) = Spectrum(end)/2;

    Spec_norm = Spectrum./area(Spe,1,501,0.5);

end

%% debugging
    % function y = is_nan_inf(M)
    %     if sum(sum(isinf(M)))>0 || sum(sum(isnan(M)))>0
    %         y = 1;
    %     else
    %         y = 0;
    %     end
    % end

%-------------------------------------------------------------------------------------------
%% Auxiliary functions to compute the derivatives
%-------------------------------------------------------------------------------------------
        function y =  dFi2_dhi(S5,S9,ti,hir,q2,q4)
            y = 1/ti*(- 1 - S5/abs(q2-hir) - S9/abs(q4-hir));
        end
```

```
function y = dFi2_dS59(S,t,q,hr)
    y = 1/t*(q-S)/abs(q-hr);
end

function y = dI_dh(e,g,G,N,S,b,s,m,S1_2)
    y = -e*g*G*N*S*b/s*exp(b*(S1_2-m)/s)/(1 + exp(b*(S1_2-m)/s))^2;
end

function y = dPhi_dh(v,L,N,S,b,s,m,S1)
    y = - v^2*L^2*N*S*b/s*exp(b*(S1-m)/s)/(1+exp(b*(S1-m)/s))^2;
end
```

8.1.2 Multiple microcolumn 3D system

Function *system_3D_full_grid*

This function defines the 3D system

$$\mathbf{Y}(t)' = \mathbf{F}(t) + \mathbf{P}$$

for a system with multiple microcolumns. The state variable $\mathbb{Y}$ is obtained by concatenation of state variables $\mathbf{y}$ describing single microcolumns. The system is solved using the forward Euler method (see script *main_3D* and function *forward_euler*). The input parameters of the function are

- variable "t" representing time
- the state variable "Y"
- the set "param_set" of parameters for the system
- the parameters "stdev_pee, p_eet, Noise" used to define the extracortical input as white noise
- the matrix "F", defining the coefficients for the 5-points stencil approximation
- the number "n_var" of variables for each microcolumn

For more details, see the comments in the code.

Function *spatial_coupling_matrix*

This function is creating a matrix F to implement the 5-points stencil approximation for the cortical inputs Φ_{ek}. Let

$$\mathbf{\Phi_{ek}} = [\Phi^1_{ek}, \Phi^2_{ek}, ...\Phi^{p*p}_{ek}]$$

be the vector of the cortical connections of all the population (where Φ^i_{ek} is the cortical connection of the microcolumn i) and F_i be the i-th row of F. Then the multiplication $F_i \ * \ \mathbf{\Phi_{ek}}$ gives the 5-points stencil approximation for microcolumn i:

$$F_i \ * \ \mathbf{\Phi_{ek}} = [\Phi^u_{ek} + \Phi^r_{ek} + \Phi^l_{ek} + \Phi^d_{ek} - 4\Phi^i_{ek}]_i$$

where u, r, l, d stand for the *upper, right, left, down* nearest neighbour microcolomns.

Function *non_linear_spectrum_full_grid*

This function calculates the non-linear PSD for a 3D multiple microcolumn system. The spectrum is obtained by summing and then averaging the PSD of each microcolumn calculated using the function *win_spec* described above. Input parameters are: - the solution "Y" of the system $\mathbf{Y}'(t) = \mathbf{F}(\mathbf{Y})$ solved with Euler method (function *forward_euler*)

- the time vector T (obtained solving the system with Euler method)
- the number "np" of microcolumns
- the number "nex" of excitatory dendritic segments
- the somatic location "ihe_soma"

Both the PSD and the normalised PSD (with respect to the area under the spectrum curve) are calculated. The output "f" is the frequency.

Function *jacobian_and_linear_PSD_full_grid*

This function is used to compute the spectrum for a 3D system with multiple microcolumn, using equation (3.71) and (3.72). The 2 evaluations give identical spectra. The function, analogous to the function *jacobian_and_linear_PSD* for one microcolumn described above, calculates

- the Jacobian matrix "J" of the linearized system

$$\mathbf{S}'(t) = \mathbf{J}\ \mathbf{s} + \mathbf{P}$$

  where J is the Jacobian evaluated at the singular point "SP", $\mathbf{s}$ is the perturbation defined as $\mathbf{s} = \mathbf{S} - \mathbf{S}^*$ (S^* is the singular point "SP") and P is the extracortical input vector. The state variable $\mathbf{S}$ is defined as a symbolic vector (see more details in the code comments) and is obtained by concatenation of state variables of single microcolumns; the Jacobian is calculated using the MatLab function *jacobian*;

- the linear spectra "spe371, spe372" are calculated using formulas (3.71) and (3.72); right and left eigen-matrices and eigen-values are previously computed using the MatLab functions *eig* and *diag*;

- the linear PSDs “spe371_norm, spe372_norm”, normalised with respect to the area under the spectrum curve.

Input parameters are:
- the angular frequency “omega” defined as $\omega = 2\pi f$, where f is the frequency
- the singular point “SPn” of the full grid system
- the set of parameters “param_set”
- the matrix “F”, defining the coefficients for the 5-points stencil approximation
- the number “nex” of excitatory dendritic segments
- the number “np” of microcolumns
- the input location “ind_input”
- the somatic location “ihe_soma”
- the electrotonic length constant “lam”

For more details, see the code and comments below.

```
%***********************************************
%% Function "system_3D_full_grid"
%***********************************************

%-------------------------------------------------------------------------------------------
%% This function defines the 3D multiple microcolumns system Y'(t) = F(t)
% The microcolumns are spread on a square discretised pxp points grid. Each
% point represents a microcolumn.
% This system has to be numerically solved to find the solution Y(t)
%-------------------------------------------------------------------------------------------

%-------------------------------------------------------------------------------------------
%% Inputs
%-------------------------------------------------------------------------------------------

        %------------------------------
        %% State variable "Y"
        %------------------------------
        % Y is a vector variable obtained by concatenation of pxp "y" vectors.
        % Each "y" vector contains the variables describing the dynamics of one
        % single microcolumn, then "Y" contains the variables for all the pxp
        % microcolumns of the grid system.

        %------------------------------
        %% For one microcolumn system
        %------------------------------
        % the state variale "y" is a vector variable, whose components are the
        % variables describing the dynamics of one population:

        % ---(mean excitatory membrane potential)---
        % y(1) = h_e (first compartment)
        % ...
        % y(ind_input) =  h_e (at input location)
        % ...
        % y(ihe_soma) = h_e (at the soma)
        % ...
        % y(nex) = h_e (last compartment)

        % ---(mean inhibitory membrane potential)---
        %  y(nex + 1)= h_i

        %  ---(synaptic currents)---
        % y(nex+2) = Iee, y(nex+3) = Iee_t  y(nex+4) = Iei, y(nex+5) = Iei_t
        % y(nex+6) = Iie, y(nex+7) = Iie_t, y(nex+8) = Iii, y(nex+9) = Iii_t

        % ---(cortical inputs)---
        % y(nex+10) = Phi_ee ,  y(nex+11) = Phi_eet
        % y(nex+12) = Phi_ei, y(nex+13) = Phi_eit
        %-------------------------------------------------------------------------------------

        %-------------------------------------------------------------------
        %% For a generic microcolumn in a multiple microcolumns system
        %-------------------------------------------------------------------

        % ___________________________________________________________________
        % IMPORTANT: Y(i_X) is a vector addressing the componenets X of the
        % single microcolumn state variables y(i) for i in {1,...,pxp}
        % ___________________________________________________________________
```

```
        % ---(mean excitatory membrane potential)---
        % Y(i_firsts) = h_e (in the first compartment of all microcolumns)
        % ...
        % Y(i_input) =  h_e (at input location, for all microcolumns)
        % ...
        % Y(i1) = h_e (at the soma of all microcolumns)
        % ...
        % Y(i_lasts) = h_e (in the last compartment of all microcolumns)


        % ---(mean inhibitory membrane potential) in all microcolumns ---
        % Y(i2) = h_i

        %  ---(synaptic currents) in all microcolumns ---
        % Y(i3) = Iee, Y(i4) = Iee_t  Y(i5) = Iei, Y(i6) = Iei_t
        % Y(i7) = Iie, Y(i8) = Iie_t, Y(i9) = Iii, Y(i10) = Iii_t

        % ---(cortical inputs) in all microcolumns ---
        % Y(i11) = Phi_ee ,  Y(i12) = Phi_eet, Y(i13) = Phi_ei  ,Y(i14) = Phi_eit

        % where the variables i_firsts, i1, i_input, i_lasts,, i1, ..., i14
        % address respectively
            % - all the variables in Y corresponding to the mean membrane
            %   potentials h_e in the first comaprtment
            % - ...
            % - all the variables in Y corresponding to the cortical inputs Phi_eit
        %-------------------------------------------------------------------------


        %------------------------------
        %% Other inputs
        %------------------------------
        % "t" is the time variable
        % "param_set" is the set of parameters
        % "stdev_pee, p_eet, Noise" are the parameters to define the extra-cortical
        % input as white noise.

        % "np" is the number of microcolumns
        % "F" is the matrix defining the 5-points stencil approximation for
        % the cortical inputs (see function "spatial_coupling_matrix")
        % "n_var" is the number of variables per microcolumn (n_var = size(y))
%-----------------------------------------------------------------------------------

function dY = system_3D_full_grid(t,Y,param_set,stdev_pee, p_eet,Noise,np,F,n_var)

        %% debugging variable
        %persistent kkk

%-----------------------------------------------------------------------
%% Parameters
%-----------------------------------------------------------------------
        %% Dendritic parameters
        nex = param_set(43);
        ihe_soma = param_set(47);
        ind_input = param_set(48:end);
        fsyn = param_set(46);
```

```
%% Other parameters
% s
tau_e = param_set(31);
tau_i = param_set(32);

% mV
h_er = param_set(1);
h_ir = param_set(2);

% mV
h_eeeq = param_set(3);
h_eieq = param_set(4);
h_ieeq = param_set(5);
h_iieq = param_set(6);
h_eq = [h_eeeq h_eieq h_ieeq h_iieq];

% 1/s
gamma_ee = param_set(11);
gamma_ei = param_set(12);
gamma_ie = param_set(13);
gamma_ii = param_set(14);
gamma = [gamma_ee gamma_ei gamma_ie gamma_ii];

% 1
N_beta_ee = param_set(15);
N_beta_ei = param_set(16);
N_beta_ie = param_set(17);
N_beta_ii = param_set(18);
N_beta = [N_beta_ee N_beta_ei N_beta_ie N_beta_ii];

% 1/s
S_max_e = param_set(21);
S_max_i = param_set(22);
S_max = [S_max_e S_max_i];

% mV
mu_bar_e = param_set(23);
mu_bar_i = param_set(24);
mu_bar = [mu_bar_e mu_bar_i];

% mV
sigmac_e = param_set(25);
sigmac_i = param_set(26);
sigmac = [sigmac_e sigmac_i];

% 1/cm
lambda_ee = param_set(33);
lambda_ei = param_set(34);
lambda = [lambda_ee lambda_ei];

% mV
Gamma_ee = param_set(7);
Gamma_ei = param_set(8);
Gamma_ie = param_set(9);
Gamma_ii = param_set(10);
Gamma = [Gamma_ee Gamma_ei Gamma_ie Gamma_ii];
```

```
        % cm/s
        v = param_set(35);

        % 1
        N_alfa_ee = param_set(19);
        N_alfa_ei = param_set(20);
        N_alfa = [N_alfa_ee N_alfa_ei];

        %
        f = exp(1)*gamma.*Gamma;

        % cm
        dx = 0.1;

        vv = 1/dx^2*3/2*v^2*F;

%-----------------------------------------------------------------------
%% Extra-cortical input implemented as white noise
%-----------------------------------------------------------------------
        % 1/s
        p_eebar = param_set(27); %p_eebar = 6603.4;
        p_ei = param_set(28); %p_ei = 2625.7;
        p_ee = (p_eebar*(1+stdev_pee*interp1(p_eet,Noise,t)))';
        p_ee(p_ee<0) = 0;
        p_ie = 0;
        p_ii =0;

%-----------------------------------------------------------------------
%% INDEXES to access all the variables in Y
%-----------------------------------------------------------------------
        [i_firsts,i1,i_lasts,i_input,i2,i3,i4,i5,i6,i7,i8,i9,i10,i11,i12,i13,i14] =↙
create_indexes(np, nex, ihe_soma,ind_input);

%-----------------------------------------------------------------------
%% Parameters and matrix for the dendritic system
%-----------------------------------------------------------------------
        ce = param_set(38);
        Ke = param_set(39);

        Ae = zeros(nex-2,nex);
        for j  = 1:nex-2
            Ae(j,j) = 1;
            Ae(j,j+1) = -2;
            Ae(j,j+2) = 1;
        end

%-----------------------------------------------------------------------
%% Synaptic input
%-----------------------------------------------------------------------
        Input = fsyn*(psi(Y(i_input),h_eq(1),h_er).*Y(i3) + psi(Y(i_input),h_eq(3),h_er).↙
*Y(i7));
        Je = zeros(n_var*np,1);
        Je(i_input) = Input;
        % note that Je has the same size as A, then Je(i_input) = Input.
        % In the sum below we then consider Je(2:nex-2) in each point, that means


%-----------------------------------------------------------------------
```

```
%% debugging point after nn iterations
%-------------------------------------------------------------------
        % nn = 1107;
        % if isempty(kkk)
        %     kkk = 0;
        % end
        %
        % if kkk == nn
        %     kkk = nn;
        % end

%-------------------------------------------------------------------------------------------------
%% 3D GRID SYSTEM
%-------------------------------------------------------------------------------------------------
dY = zeros(n_var*np,1);

% Dendritic he system with sealed end boundary conditions
dY(i_firsts) = Ke*(Y(i_firsts+1)-Y(i_firsts)) -1/tau_e*Y(i_firsts)+ce;
for i = 1:np
    dY(i_firsts(i)+1:i_lasts(i)-1) = Ke*(Ae*Y(i_firsts(i):i_lasts(i)))-1/tau_e*Y(i_firsts↙
(i)+1:i_lasts(i)-1) + ce + 1/tau_e*Je(i_firsts(i)+1:i_lasts(i)-1);
end

dY(i_lasts) = Ke*(Y(i_lasts-1)-Y(i_lasts)) -1/tau_e*Y(i_lasts)+ce;

% Inhibitory membrane potential hi
dY(i2) = 1/tau_i*(-(Y(i2)-h_ir)+ psi(Y(i2),h_eq(2),h_ir).*Y(i5)+ psi(Y(i2),h_eq(4),h_ir).↙
*Y(i9));

% Synaptic currents I_lk
dY(i3) = Y(i4);
dY(i4) = -2*gamma(1)*Y(i4)-(gamma(1))^2*Y(i3)+f(1)*(N_beta(1)*S(Y(i1),S_max(1),mu_bar(1),↙
sigmac(1))+Y(i11)+p_ee);
dY(i5) = Y(i6);
dY(i6) = -2*gamma(2)*Y(i6)-(gamma(2))^2*Y(i5)+f(2)*(N_beta(2)*S(Y(i1),S_max(1),mu_bar(1),↙
sigmac(1))+Y(i13)+p_ei);
dY(i7) = Y(i8);
dY(i8) = -2*gamma(3)*Y(i8)-(gamma(3))^2*Y(i7)+f(3)*(N_beta(3)*S(Y(i2),S_max(2),mu_bar(2),↙
sigmac(2))+p_ie);
dY(i9) = Y(i10);

% Cortical inputs
dY(i10) = -2*gamma(4)*Y(i10)-(gamma(4))^2*Y(i9)+f(4)*(N_beta(4)*S(Y(i2),S_max(2),mu_bar↙
(2),sigmac(2))+p_ii);
dY(i11) = Y(i12);
dY(i12) = -2*v*lambda(1)*Y(i12)-v^2*lambda(1)^2*Y(i11)+v^2*lambda(1)^2*N_alfa(1)*S(Y(i1),↙
S_max(1),mu_bar(1),sigmac(1)) + vv*Y(i11);
dY(i13) = Y(i14);
dY(i14) = -2*v*lambda(2)*Y(i14)-v^2*lambda(2)^2*Y(i13)+v^2*lambda(2)^2*N_alfa(2)*S(Y(i1),↙
S_max(1),mu_bar(1),sigmac(1)) + vv*Y(i13);
%-------------------------------------------------------------------------------------------------
%kkk = kkk+1;
end

%-------------------------------------------------------------------------------------------------
%% AUXILIARY FUNCTIONS
%-------------------------------------------------------------------------------------------------
        function y = psi(h,eq,r)
```

```
        y = (eq-h)/abs(eq-r);

    end

function y = S(h,S_max,mu_ba,sigmac)
    b = -sqrt(2);
    r_abs=0;
    %y = 0;
    y = S_max./(1+(1-r_abs*S_max)*exp(b*(h-mu_ba)/sigmac));
end
```

```
%***********************************************
%% Function "spatial_coupling_matrix"
%***********************************************

%-------------------------------------------------------------------------------------------
%% This function is creating a matrix used to implement the 5 points stencil
% approximation for the cortical inputs.

        % p*p =  number of points of the grid
        % Phi_ek = [Phi_ek(1), ..., Phi_ek(p*p)]
        % F = F(i,j)
        % F(i,:)*Phi_ek = Phi_ek_upper + Phi_ek_right + Phi_ek_left + Phi_ek_down
        % -4*Phi_ek  % this is the stencil approximation
%-------------------------------------------------------------------------------------------

function F = spatial_coupling_matrix(p)

    F = zeros(p,p);
    r = 1;
        for i = 1:p
            for j = 1:p
                F(r,(i-1)*p+j) = -4;
                if (i==1) F(r,(p-1)*p+j) = 1;
                else F(r,(i-2)*p+j) = 1;
                end
                if (j==1) F(r,(i-1)*p+(j-1+p)) = 1;
                else F(r,(i-1)*p+j-1) = 1;
                end
                if (j==p) F(r,(i-1)*p+(j+1-p)) = 1;
                else F(r,(i-1)*p+j+1) = 1;
                end
                if (i==p) F(r,i*p+j-p*p) = 1 ;
                else F(r,i*p+j) = 1;
                end
                r = r+1;
            end
        end

end
```

```
%****************************************
%% Function "non_linear_spectrum_full_grid"
%****************************************

%-------------------------------------------------------------------------------------------
%% This function calculates the non-linear spectrum "spec" for a system with p*p = np
% microcolumns. The spectrum is calculated for each microcolumn using the
% function "win_spec"; the PSD of the system is then obtained averaging all the PSDs
% from all the microcolumns.
%-------------------------------------------------------------------------------------------

%-------------------------------------------------------------------------------------------
%% Inputs - Outputs
%-------------------------------------------------------------------------------------------
        % "spec_norm" is the spectrum normalised with respect to the area
        % under the PSD
        % "f" is the frequency

        % "T" is the time vector
        % "Y" is the state variable of the full system. It is obtained by
        % concatenation of state variables of the single microcolumns.

        % "np" is the number of microcolumns, distributed on a square grid
        % (np = p*p)

        % "nex" is the number of dendritic excitatory compartments
        % "ihe_soma" is the index for the somatic compartment

function [spec,spec_norm,f] = non_linear_spectrum_full_grid(Y,T,np,nex,ihe_soma)

    % dimension of the system
    n_var = nex + 13;
    indexes = 1:numel(Y);

    % i1 is a vector of indexes to access the soma of each microcolumn in Y
    i1 = zeros(np,1);
    positions_he_in_Y = mod(indexes,n_var)==ihe_soma;
    k=1;
    for i=1:np*n_var
        if (positions_he_in_Y(i)==1)
            i1(k)=i;
            k = k+1;
        end
    end

    % sum of the spectra of all microcolumns, calculated with win_spec
    [spec,f] = win_spec(Y(i1(1),:)',1000,2);
    for i = 2:size(i1)
        spec = spec + win_spec(Y(i1(i),:)',1000,2);
    end
    % average
    spec = spec./(np);
    %normalisation
    spec_norm = spec./area(spec,1,size(spec,2),f(2));

end
```

```
%***********************************************
%% Function "jacobian_and_linear_PSD_full_grid"
%***********************************************

%-------------------------------------------------------------------------------------------
%% This function calculates the Jacobian "J" and the linear spectrum for a system with
% p*p = np microcolumns, using the formula 3.71 and 3.72 of the thesis (the 2 spectra
% are identical). The linearised system is dS/dt = J*s + P, where
        % - P is the extracortical
        % - s is the perturbation s = S-S^* (S^* is the singular point "SP")
        % - J is the Jacobian matrix evaluated at a singular point "SP".
%  The Jacobian is calculated using the MatLab function "jacobian".
%-------------------------------------------------------------------------------------------

%-------------------------------------------------------------------------------------------
%% State variable S
%-------------------------------------------------------------------------------------------

% The state variable "SS" of the full system is defined as a symbolic vector
    % "np" is the number of microcolumn
    % S = [[he1_1 he1_2 ... he1_nex hi_1 S1_3 S1_4 S1_5 ... S1_13 S1_14]
    %       -----state vector for microcolumn 1-----
    %      ...
    %     [he_np_1 he_np_2 ... he_np_nex hi_1 Snp1_3 ... Snp_14]]
    %       -----state vector for microcolumn np-----


    % -------
    % he
    % -------
    % he(i,1:nex) = he over the dendrite from compartment 1 to nex, for
    % microcolumn i, for i in {1,...,np}
    % (note that hhe(:,i) = he(i,:), see below)

    % -------
    % hi
    % -------
    %hi(i) = hi of microcolumn i, for i in {1,...,np}


    % -------
    % I_lk
    % -------
    % S(i,3) = Iee, S(i,4) = Iee_t  S(i,5) = Iei, S(i,6) = Iei_t
    % S(i,7) = Iie, S(i,8) = Iie_t, S(i,9) = Iii, S(i,10) = Iii_t
    % for microcolumn i, for i in {1,...,np}

    % -------
    % Phi_ek
    % -------
    % S(i,11) = Phi_ee ,  S(i,12) = Phi_eet, S(i,13) = Phi_ei  ,S(i,14) = Phi_eit
    % for microcolumn i, for i in {1,...,np}


%-------------------------------------------------------------------------------------------
%% Inputs - Outputs
%-------------------------------------------------------------------------------------------
        % "param_set" is the set of parameters of the system
```

```
        % "omega" is the angular frequency, defined by omega = 2*pi*f, where f is the
        % frequency
        % "SPn" is the singular point for the entire grid of microcolumns

        % "F" is the matrix defining the 5-points stencil approximation for
        % the cortical inputs (see function "spatial_coupling_matrix")

        % "nex" is the number of dendritic excitatory compartments
        % "np" is the number of microcolumns, distributed on a square grid (np = p*p)

        % "ind_input" is the index for the synaptic input
        % "ihe_soma" is the index for the somatic compartment
        % "lam" is the dendritic length constant

        % "spe371" spectrum calculated with formula 3.71 of the thesis
        % "spe371" spectrum calculated with formula 3.71 of the thesis
        % "spe371_norm" and "spe372_norm" are the spectra normalised with
        % respect to the area under the spectrum curve

        % "J" is the Jacobian matrix evaluated at the singular point SP
%-------------------------------------------------------------------------------------

function [spe371,spe371_norm,spe372,spe372_norm,J] = jacobian_and_linear_PSD_full_grid↙
(omega,SPn,param_set,F,nex,np,ind_input,ihe_soma,lam)


%-------------------------------------------------------------------------------------
%% Parameters
%-------------------------------------------------------------------------------------

        %% grid dimension
        dx = 0.1; % distance between two microcolumns in the derection perpendicular
        % to the dendritic extension
        n_var = nex + 13; % number of variables for each microcolumn

        %% constants from file

        % mV
        her = param_set(1);
        hir = param_set(2);

        % mV
        q1 = param_set(3); % h_ee_eq
        q2 = param_set(4); % h_ei_eq
        q3 = param_set(5); % h_ie_eq
        q4 = param_set(6); % h_ii_eq

        %mV
        Gamma1 = param_set(7);
        Gamma2 = param_set(8);
        Gamma3 = param_set(9);
        Gamma4 = param_set(10);
        Gamma = [Gamma1 Gamma2 Gamma3 Gamma4];

        % 1/s
        gamma1 = param_set(11);
        gamma2 = param_set(12);
        gamma3 = param_set(13);
```

```
gamma4 = param_set(14);
gamma = [gamma1 gamma2 gamma3 gamma4];

% 1
Nb1 = param_set(15);
Nb2 = param_set(16);
Nb3 = param_set(17);
Nb4 = param_set(18);

% 1
Na1 = param_set(19);
Na2 = param_set(20);

% 1/s
Sem = param_set(21);
Sim = param_set(22);

% mV
mue = param_set(23);
mui = param_set(24);

% mV
se = param_set(25);
si = param_set(26);

pee = param_set(27);

% s
te = param_set(31);
ti = param_set(32);

% 1/cm
lambda_ee = param_set(33);
lambda_ei = param_set(34);
lambda = [lambda_ee lambda_ei];

e = exp(1);
f = e*gamma.*Gamma;

% cm/s
v = param_set(35);


vv = 1/dx^2*3/2*v^2*F;
b = -sqrt(2);
r_abs = 0;

%% dendritic parameters
% cm
lenght = 0.4; % lenght of the dendrite
dz = lenght/nex; % lenght of a compartment
ce = her/te;
Ke = lam^2/(te*dz^2);


%-----------------------------------------------------------------------------------------
%% Symbolic vectors
%-----------------------------------------------------------------------------------------
```

```
        % he
        he = sym('he%d_%d',[np, nex]);
        for i = 1:np
            hhe(:,i) = he(i,:);
        end

        %hi
        hi = sym('hi%d',[np,1]);

        % I_lk and Phi_ek
        S = sym('S%d_%d',[np 14]);

        % State variable for the full system
        SS = horzcat(he(1,1:nex),hi(1),S(1,3:14));
        for i = 2:np
            SS = horzcat(SS, he(i,1:nex),hi(i),S(i,3:14));
        end


%-------------------------------------------------------------------------------------------
%% Symbolic Jacobian
%-------------------------------------------------------------------------------------------
system = [];
for i = 1:np
    fadd =  [Ke*(hhe(2,i)-hhe(1,i))-1/te*hhe(1,i)+ce;
         Ke*(hhe(1,i)-2*hhe(2,i)+hhe(3,i))-1/te*hhe(2,i)+ce;
         Ke*(hhe(2,i)-2*hhe(3,i)+hhe(4,i))-1/te*hhe(3,i)+ce;
         Ke*(hhe(3,i)-2*hhe(4,i)+hhe(5,i))-1/te*hhe(4,i)+ce;
         Ke*(hhe(4,i)-2*hhe(5,i)+hhe(6,i))-1/te*hhe(5,i)+ce;
         Ke*(hhe(5,i)-2*hhe(6,i)+hhe(7,i))-1/te*hhe(6,i)+ce;
         Ke*(hhe(6,i)-2*hhe(7,i)+hhe(8,i))-1/te*hhe(7,i)+ce;
         Ke*(hhe(7,i)-2*hhe(8,i)+hhe(9,i))-1/te*hhe(8,i)+ce;
         Ke*(hhe(8,i)-2*hhe(9,i)+hhe(10,i))-1/te*hhe(9,i)+ce;
         Ke*(hhe(9,i)-2*hhe(10,i)+hhe(11,i))-1/te*hhe(10,i)+ce;
         Ke*(hhe(10,i)-2*hhe(11,i)+hhe(12,i))-1/te*hhe(11,i)+ce;
         Ke*(hhe(11,i)-2*hhe(12,i)+hhe(13,i))-1/te*hhe(12,i)+ce;
         Ke*(hhe(12,i)-2*hhe(13,i)+hhe(14,i))-1/te*hhe(13,i)+ce;
         Ke*(hhe(13,i)-2*hhe(14,i)+hhe(15,i))-1/te*hhe(14,i)+ce;
         Ke*(hhe(14,i)-2*hhe(15,i)+hhe(16,i))-1/te*hhe(15,i)+ce;
         Ke*(hhe(15,i)-2*hhe(16,i)+hhe(17,i))-1/te*hhe(16,i)+ce;
         Ke*(hhe(16,i)-2*hhe(17,i)+hhe(18,i))-1/te*hhe(17,i)+ce;
         Ke*(hhe(17,i)-2*hhe(18,i)+hhe(19,i))-1/te*hhe(18,i)+ce;
         Ke*(hhe(18,i)-2*hhe(19,i)+hhe(20,i))-1/te*hhe(19,i)+ce;
         Ke*(hhe(19,i)-2*hhe(20,i)+hhe(21,i))-1/te*hhe(20,i)+ce;
         Ke*(hhe(20,i)-2*hhe(21,i)+hhe(22,i))-1/te*hhe(21,i)+ce;
         Ke*(hhe(21,i)-2*hhe(22,i)+hhe(23,i))-1/te*hhe(22,i)+ce;
         Ke*(hhe(22,i)-2*hhe(23,i)+hhe(24,i))-1/te*hhe(23,i)+ce;
         Ke*(hhe(23,i)-2*hhe(24,i)+hhe(25,i))-1/te*hhe(24,i)+ce+1/te*((q1-he(i,↙
ind_input))/(abs(her-q1))*S(i,3)+(q3-he(i,ind_input))/(abs(q3-her))*S(i,7));
         Ke*(hhe(24,i)-2*hhe(25,i)+hhe(26,i))-1/te*hhe(25,i)+ce;
         Ke*(hhe(25,i)-2*hhe(26,i)+hhe(27,i))-1/te*hhe(26,i)+ce;
         Ke*(hhe(26,i)-2*hhe(27,i)+hhe(28,i))-1/te*hhe(27,i)+ce;
         Ke*(hhe(27,i)-2*hhe(28,i)+hhe(29,i))-1/te*hhe(28,i)+ce;
         Ke*(hhe(28,i)-2*hhe(29,i)+hhe(30,i))-1/te*hhe(29,i)+ce;
         Ke*(hhe(nex-1,i)-hhe(nex,i))-1/te*hhe(nex,i)+ce;
         1/ti*(-(hi(i)-hir)+(q2-hi(i))/(abs(q2-hir))*S(i,5)+(q4-hi(i))/(abs(q4-hir))*S(i,↙
9));
```

```
         S(i,4);
        -2*gamma(1)*S(i,4)-(gamma(1))^2*S(i,3)+f(1)*(Nb1*Sem./(1+(1-r_abs*Sem)*exp(b*(he↙
(i,ihe_soma)-mue)/se))+S(i,11));
        S(i,6);
        -2*gamma(2)*S(i,6)-(gamma(2))^2*S(i,5)+f(2)*(Nb2*Sem./(1+(1-r_abs*Sem)*exp(b*(he↙
(i,ihe_soma)-mue)/se))+S(i,13));
        S(i,8);
        -2*gamma(3)*S(i,8)-(gamma(3))^2*S(i,7)+f(3)*Nb3*Sim./(1+(1-r_abs*Sim)*exp(b*(hi↙
(i)-mui)/si));
        S(i,10);
        -2*gamma(4)*S(i,10)-(gamma(4))^2*S(i,9)+f(4)*Nb4*Sim./(1+(1-r_abs*Sim)*exp(b*(hi↙
(i)-mui)/si));
        S(i,12);
        -2*v*lambda(1)*S(i,12)+v^2*lambda(1)^2*Na1*Sem./(1+(1-r_abs*Sem)*exp(b*(he(i,↙
ihe_soma)-mue)/se))+vv(i,:)*S(:,11);
        S(i,14);
        -2*v*lambda(2)*S(i,14)+v^2*lambda(2)^2*Na2*Sem./(1+(1-r_abs*Sem)*exp(b*(he(i,↙
ihe_soma)-mue)/se))+vv(i,:)*S(:,13);];
    system = [system; fadd];
end

J = jacobian(system,SS);

%-------------------------------------------------------------------------------------------
%% Jacobian evaluated at the sungular point SPn
%-------------------------------------------------------------------------------------------
    J_star = double(subs(J,SS,SPn));

%-------------------------------------------------------------------------------------------
%% PSD prediction
%-------------------------------------------------------------------------------------------
    % eigenvalues
    [R,D,Lm] = eig(J_star);
    eig_values = diag(D);
    L = Lm';
    E = L*R;
    en = diag(E);

    % indexes to access the variables he_soma and Phi_eit in Y
    indexes = 1:n_var*np;
    i1 = zeros(np,1);
    positions_hesoma_in_Y = mod(indexes,n_var)==ihe_soma;
    k=1;
    for i=1:np*n_var
        if (positions_hesoma_in_Y(i)==1)
            i1(k)=i;
            k = k+1;
        end
    end
    i4 = i1+(nex-ihe_soma+3);

    %-------------------------------------------------------------
    %% PSDs from all points (as sum of PSD of single microcolumns)
    %-------------------------------------------------------------
    % spectrum 3.71: sum all the spectra of all the macrocolumns
    spe371 = spe_from1point(1,np, omega,eig_values,R,en,L,pee,f,n_var,ihe_soma,i4);
    for p_spe = 2:np
        spe371 = spe371 + spe_from1point(p_spe,np, omega,eig_values,R,en,L,pee,f,n_var,↙
```

```
ihe_soma,i4);
    end

    % spectrum 3.72: spectrum from one microcolumn multiplied by the number of
    % points
    spe372 = np*spe_from1point(1,np, omega,eig_values,R,en,L,pee,f,n_var,ihe_soma,i4);

    % calculating single side spectra
    spe371(2:end-1) = spe371(2:end-1)*2;
    spe372(2:end-1) = spe372(2:end-1)*2;

    % normalised spectra
    spe371_norm = spe371./area(spe371,1,size(spe371,2),f(2));
    spe372_norm = spe372./area(spe372,1,size(spe372,2),f(2));


    %-------------------------------------------------------------
    %% function calculation the PSD for single microcolumn
    %-------------------------------------------------------------
    % Here, for each row "he_row" the indexes "Phi_eit_col" address
    % the corresponding column elements of the row s (they are the non-zero
    % elements of the extra-cortical input vector P
    function spe = spe_from1point(p_spe,np, omega,eig_values,R,en,L,pee,f,n_var,ihe_soma,↙
i4)
        he_row = n_var*(p_spe-1) + ihe_soma;
        Pc = zeros((n_var)*np,(n_var)*np,size(omega,2));
        spe = zeros(1,size(omega,2));

        for kk = 1:np
            Phi_eit_col = i4(kk);
            for jj = 1:size(omega,2)
                Pc(:,:,jj) = (R*diag(1./(en.*(omega(1,jj)*1i-eig_values))))*L;
                spe(1,jj) = spe(1,jj) + abs((Pc(he_row,Phi_eit_col,jj))*abs(f(1)*0.↙
1*pee))^2;
            end
        end
    end

end
```

8.2 Particle Swarm Optimization (PSO) algorithm

The following scripts and functions illustrate how the Particle Swarm algorithm has been implemented and run.

Script *run_swarm_fit1*

This script is used to run the PSO algorithm varying the location of the synaptic inputs.

- The geometry of the NFM is defined. Initial and final input locations are set
- The fitness function and the search space are defined
- Initial positions for the particles are set
- A loop of PSO simulations is run
- Outputs are stored in files

For more details, see comments embedded in the code below.

Function *particle_swarm*

This function is the PSO algorithm described in Chapter 5, with global topology defined in Section 5.3 and velocities updated using formula (5.6). The main steps are illustrated below

- Initialization:
  - The particles are initialized at uniform random positions $\mathbf{x}$ (corresponding to "**X**" in the code)
  - The best position $\mathbf{p}$ ("**PX**") and the velocity $\mathbf{v}$ ("**V**") of each particle are initialized to zero
  - The fitness evaluation $\mathbf{f}(\mathbf{p})$ ("**FPX**") is initialized to Inf, for each particle
- At each iteration:

- For all particles:
  - The fitness function is evaluated at $\mathbf{x}$; let $\mathbf{f}(\mathbf{x})$ be this evaluation (**FPX** in the code)
  - If $\mathbf{f}(\mathbf{x}) < \mathbf{f}(\mathbf{p})$ then both $\mathbf{p}$ and $\mathbf{f}(\mathbf{p})$ are updated ($\mathbf{p} = \mathbf{x}$ and $\mathbf{f}(\mathbf{p}) = \mathbf{f}(\mathbf{x})$)
- The global best position $\mathbf{g}$ ("**bestx**") is calculated as the position corresponding to the minimum evaluation ("**fbestx**"); $\mathbf{f}(\mathbf{g}) = \min(\mathbf{f}(\mathbf{p}))$
- The positions $\mathbf{x}$ and velocities $\mathbf{v}$ are updated according to formula (5.6) (particles converge towards the global best)
- If a particle is outside the allowable bounds then its position is set to the closest bound and its velocity to zero
- The PSO stops if for more than n ("max_stall_iteration") iterations the best evaluation does not differ from the previous best evaluation more than the tolerance μ ("tolerance"); see the code and comments for more details

The inputs of the function are:
- the fitness function
- the number of parameters "n_param"
- the number of particles "n_particles"
- the initial positions for the particles "X_init"
- the total number of PSO iterations "n_iter"
- the tolerance "tolerance"
- the maximum number "max_stall_iteration" of allowed consecutive iterations leading to a similar evaluations (see explanation above)

The outputs of the function are:
- the best position "bestx"
- the best evaluation "fbestx"
- the number of fitness calls "n_fitness_call"

- the matrix “ALL” where the results of all iterations are stored (see embedded comments for more details)

For a complete explanation of the algorithm see also Section 5.3; more details about the implementation are illustrated in the comments embedded in the code below.

Function *fitness*

The fitness function evaluates the similarity between the PSD at position **x** of the search space and typical PSDs encountered in humans. This evaluation is based on some criteria described in Section 5.4.2. A “cost” is associated to each criterium and the fitness evaluation is the averaged sum of all the costs[1].

The inputs of the functions are:
- the position **x** of the particle in the search space
- the number “nex” of excitatory dendritic segments
- the input location “ind_input”
- the somatic location “ihe_soma”
- the electrotonic length constant “lam”

The outputs are:
- the fitness evaluation “f_min”
- the PSD for the current position **x**
- the vector of costs “all_costs”

A number of preliminary conditions are checked before computing the best evaluation. These conditions correspond to cases where the linearization and linear PSD prediction cannot be performed (see comments embedded in the code). For more details about the criteria see both Section 5.4.2 and comments embedded in the code below.

[1]The PSO algorithm make the particles converge towards the minimum fitness evaluation.

```
%**************************************
%% Script "run_swarm_fit1"
%**************************************

%-------------------------------------------------------------------------------------------
%% This script is used to run the Particle Swarm Opimization (PSO) Algorithm.
%-------------------------------------------------------------------------------------------

    % 1.
        % The geometry of the NF model is set.
        % The swarm is run n times varying the location of the synaptic input from
        % "init_comp" to "final_comp" (n = final_comp-init_comp+1)

    % 2.
        % The fitness function and its input parameters are assigned.

    % 3.
        % The search space is defined.
        % The particles are initially randomly distributed in the search space

    % 4.
        % The output matrix "M" is initialized. M contains:
            % the best position "bestx"
            % the evaluation "fbestx" of the fitness function at bestx
            % the number of times the fitness has been run "n_fitness_call"
            % the vector "cost_vector" of the costs (the fitness function minimizes
            % the sum of the costs)
            % the time "el_time" needed for the swarm to find the solution
            % the best power spectrum
        % FOR ALL THE INVESTIGATED COMPARTMENTS


    % 5.
        % A loop is designed to run the swarm algorithm n times, varying the location
        % of the synaptic input from "init_comp" to "final_comp"
        % (n = final_comp-init_comp+1)
        % For each swarm simulation the particles have random initial positions

        % The output of each simulation is stored in the file "ALL_x.mat",
        % where x is the current location of the input

        % The outputs for all the input compartments are stored in the matrix M

%-----------------------------------------------------
%% 1. geometry of the NF model
%-----------------------------------------------------
    % number of excitatory dendritic compartments
    nex = 30;
    % somatic location
    ihe_soma = 24;
    % dendritic length constant
    lam = 0.09;
    % frequency
    freq = 0:0.5:250;

    %% investigated dendritic compartments
    numb_investigated_compartments = 1;
    init_comp = 22; % compartment where the synaptic input is located
```

```
    final_comp = 22; % compartment where the synaptic input is located

%-------------------------------------------------
%% 2. Fitness function
%-------------------------------------------------
    fitn = @fitness_1;
    tolerance = 10^(-3);
    max_stall_iteration = 20;

%-------------------------------------------------------------------------
%% 3. Parameter space
%-------------------------------------------------------------------------
    % number of particles in the swarm
    n_particles = 40;
    % number of iterations
    n_iter = 500;
    % dimension of the search space
    n_param = 6;

        % search space parameters:
        % hee_eq, gamma_ee, gamma_ei, her, te, fsyn
        lb = [-20, 100, 100, -80, 5/1000, 8];
        ub = [10, 1000, 1000, -60, 150/1000, 1000];
        db = ub-lb; % legal interval for x

    %% random initial condition for the particles
    X_init = repmat(lb,n_particles,1) + (repmat(db,n_particles,1) .* rand(n_particles,↙
n_param));

%-------------------------------------------------------------------------
%% 4. Output Initialization
%-------------------------------------------------------------------------

    M = zeros(nex,n_param+3+18+501);
    % bestx, fbestx, n_fitness_call, el_time(i)
    % 18 for cost vector
    % 501 for spectrum
    el_time = zeros(1,numb_investigated_compartments);


%-------------------------------------------------------------------------
%% 5. Loop running the PSO Algorithm and saving the outputs
%-------------------------------------------------------------------------

tic
for i = init_comp:final_comp

    % location for the synaptic input
    ind_input = i;
    ind_input

    tic;
    % running the swarm
    [bestx, fbestx,n_fitness_call, ALL]= particle_swarm(@(x) fitn(x,nex, ind_input,↙
ihe_soma,lam), n_param, n_particles, X_init, lb, ub, n_iter, max_stall_iteration,↙
tolerance);

    % saving output for corrent input location
```

```
    save(['ALL_' num2str(i) '.mat'],'ALL','bestx','fbestx','n_fitness_call')

    % finding the cost vector at bestx
    [yy,best_spe_norm, cost_vector] = fitn(bestx, nex, ind_input, ihe_soma,lam);

    % initial condition for next simulation
    X_init = repmat(lb,n_particles,1) + (repmat(db,n_particles,1) .* rand(n_particles,↙
n_param));

    % simulation time
    el_time(i) = toc

    % the output matrix M containing the bests for all the compartments is
    % updated
    M(i,:) = [bestx, fbestx, n_fitness_call, el_time(i), cost_vector,best_spe_norm];
end
toc
```

```
%***************************************
%% Function "particle_swarm"
%***************************************

%-------------------------------------------------------------------------------------------
% This function is the particle swarm algorithm with global topology
% (see Section 5.3 for more details)
% and velocity function defined by Eq. (5.6)
%-------------------------------------------------------------------------------------------

%-------------------------------------------------------------------------------------------
%% Inputs
%-------------------------------------------------------------------------------------------

% The inputs of the function are

    % the "fitness" function evaluating the similarity in between the PSD
    % at position x of the search space and typical PSDs encountered
    % in humans

    % the dimension of the search space "n_param"

    % the number of particles "n_particles"

    % initial positions "X_init" for all the particles

    % lower and upper bounds ("lb" and "ub") for the particles

    % the number of iteration "n_iter" of the swarm simulation

    % the maximum number of iterations "max_stall_iteration"
    % producing a best solution that differ from the previous solution
    % by "tolerance"
    % In the case "bests difference < tolerance" for more than
    % max_stall_iteration, the MatLab function fminsearch is used and
    % the PSO is stopped

%-------------------------------------------------------------------------------------------
%% Outputs
%-------------------------------------------------------------------------------------------

% The outputs are:

    % the best position "bestx"

    % the best fitness evaluation "fbestx"

    % the number of fitness calls "n_fitness_call"

    % The matrix ALL storing the results of all iterations:
    %    ALL(row i) = [iteration index i, particle index j, current position X
    %                  fitness evaluation, all_costs vector (see fitness function),
    %                 PSD];


function [bestx, fbestx,n_fitness_call,ALL]= particle_swarm(fitness, n_param,↙
n_particles, Xinit, lb, ub, n_iter, max_stall_iteration, tolerance)
```

```
%-------------------------------------------------------------------------------------------
    %% parameters for fminsearch
    %% if tolerance is not set
%-------------------------------------------------------------------------------------------
    if (nargin < 8)
        tolerance = eps
        max_stall_iteration = Inf
    end

    n_stall = 0;
    fbestx_prev = inf;

%-------------------------------------------------------------------------------------------
    %% parameters for the Converegence
    w_soc = 1.3;
    w_cog = 2.8;
%-------------------------------------------------------------------------------------------

%-------------------------------------------------------------------------------------------
    %% Initial Conditions and Parameters
%-------------------------------------------------------------------------------------------

    % Positions
    X = Xinit;
    % Velocities
    V = zeros(n_particles,n_param); % zero initial velocity

    % Fitness evaluation in all positions X
    PX = zeros(size(X));
    % Best position encountered for each particle
    FPX = inf(1, n_particles); % FPX(i) is the minimum fitness for the i-th particle

    % number of fitness calls
    n_fitness_call = 0;

    % Ouptput vector
    ALL = zeros(n_iter*n_particles,2+1+n_param+17+501);

%-------------------------------------------------------------------------------------------
%% Swarm iterations
%-------------------------------------------------------------------------------------------
for i = 1:n_iter

    %-----------------------------------------------------------
    % Find the best evaluation for each particle
    %-----------------------------------------------------------

    for j = 1:n_particles
        % printing
        PRINT_PART = sprintf('iteration %d, particle %d, current position  %4.4f  %4.4f↙
%4.4f  %4.4f ',i,j,X(j,:));
        disp(PRINT_PART)

        %% Evaluate the fitness for all particles
        [tempFit,Spe,cost_vector, eigen_values] = fitness(X(j,:));

        % printing
        PRI_FIT=sprintf('fitness evaluation is ');
```

```
        disp(PRI_FIT)
        disp(tempFit)

        %% store the outputs in ALL
        ALL((i-1)*n_particles+j,:) = [i,j,X(j,:),tempFit,cost_vector,Spe];


        %% Update the best for each particle j
        % If the current evaluation is less than previous evaluation
        % then the best for particle j is updated
        if (tempFit<FPX(j))
            PX(j,:) = X(j,:); % matrices of best positions so far of all the particles
            FPX(j) = tempFit;
        end
    end

    %-----------------------------------------------------------
    % Update the number of fitness call
    %-----------------------------------------------------------
    n_fitness_call = n_fitness_call + n_particles;


    %-----------------------------------------------------------
    % Find the global best evaluation for all particles
    % (minumum of all evaluations)
    %-----------------------------------------------------------

    [fbestx, best_x_ind] = min(FPX);
    bestx = PX(best_x_ind, :);

    %-----------------------------------------------------------
    % Use fminsearch if n_stall==max_stall_iteration
    %-----------------------------------------------------------

    % update n_stall
    if (abs(fbestx-fbestx_prev) < tolerance)
        n_stall = n_stall+1;
    else
        n_stall = 0;
    end
    % find the minumum using a matlab function
    if (n_stall == max_stall_iteration)
        min_matlab = 1
        [bestx, fbestx] = fminsearch(fitness, bestx);
        n_fitness_call = n_fitness_call + 1;
        return
    end

    %-----------------------------------------------------------
    % Update the particles positions and velocities according
    % to formula described in Section 5.3
    %-----------------------------------------------------------

    % update the current global best evaluation
    fbestx_prev = fbestx;

    % constriction factor
    w = w_cog + w_soc;
```

```
    w0 = 2/abs(2-w-sqrt((w-4)*w)); % setting w0 to have convergence of the swarm

    %% Velocities update
    %V formula 1
    %V = w0*V + w_soc*rand()*(repmat(bestx,n_particles,1) - X) + w_cog*rand()*(PX - X);

    %V formula 2
    %V = V + w0*(V + w_soc*rand()*(repmat(bestx,n_particles,1) - X) + w_cog*rand()*(PX -↙
X));

    %V formula 3
    V = w0*(V + w_soc*rand()*(repmat(bestx,n_particles,1) - X) + w_cog*rand()*(PX - X));

    %% Position Update
    X = X + V;

    %-------------------------------------------------------------
    % Check that all particles are inside the bounds
    %-------------------------------------------------------------

    XX = min( max(X, repmat(lb,n_particles,1)), repmat(ub,n_particles,1) );

    % if one of the particles is outside the boundaries all the velocities
    % are set to zero
    if (isequal(X,XX)==0)
        X = XX;
        V = zeros(n_particles,n_param);
    end

end


end
```

```
%**************************************
%% Function "fitness_1"
%**************************************

%-------------------------------------------------------------------------------------------
%% This function evaluates the similarity between the PSD (at position x of the PSO
% search space) and typical PSDs encountered in humans
%-------------------------------------------------------------------------------------------

%-------------------------------------------------------------------------------------------
%% Inputs
%-------------------------------------------------------------------------------------------
% The similarity is evaluated at position x of the search space and for a configuration
% defined by the parameters:

        % "nex" = number of dendritic excitatory compartments
        % "ind_input" = index for the synaptic input
        % "ihe_soma" = index for the somatic compartment
        % "lam" = dendritic electrotonic length constant

%-------------------------------------------------------------------------------------------
%% Outputs
%-------------------------------------------------------------------------------------------

    % "SPE_NORM" is the normalised spectrum at position x and corresponding to
    % the configuration define by input parameters above

    % "f_min" is the fitness evaluation, normally varying from 0 to "high_cost,
    % where high_cost = 100.

    % The global variable "fitnessCounter" is the number of fitness call for the PSO
    % simulation

    %-----------------------------------------------------------------------
    %% Costs and fitness evaluation
    %-----------------------------------------------------------------------
    % The similarity of PSDs is calculated considering a number of criteria that
    % the current spectrum should fulfill (defined in Section 5.4.2).
    % A "cost" is associated to each criterium
    % (the cost is zero if the criterium is satisfied and increases linearly
    % otherwise (see function "cost_lines" below)).

    % For a detailed description of the criteria see the code below and the
    % explanation provided in Section 5.4.2

    % f_min is the averaged sum of all costs.
    % The costs are stored in a 13 dimension vector "costs"

    % The components of costs are:
            %1 cost1_Se                      criteria 1
            %2 cost1_Si
            %3 cost_delta                    criteria 2
            %4 cost_theta
            %5 cost_alpha
            %6 cost_beta

            %7 cost_ratio_td                 criteria 3
            %8 cost_ratio_ta
```

```
            %9 cost_SEF_90                  criterium 4

            %10 cost_alphamax_deltamax      criteria 5
            %11 cost_thetamin_alphamax
            %12 cost_thetamin_deltamax
            %13 cost_thetamin_deltamin

    %--------------------------------------------------------------------
    %% Preliminarly costs: flag variables
    %--------------------------------------------------------------------
    % A number of 'flag' variables control the behaviour of the system.
    % The flagged cases (flag = 1) correspond to configurations
    % presenting numerical problems (as computations involving 'NaN' or 'Inf'),
    % or to configurations where the linearization cannot be performed, as the
    % case where positive eigenvalues occur. In these cases, the fitness
    % evaluation is set to high_cost and both the eigenvalues and the spectrum
    % are set to zero.

    % Each flag variable is associated to a "n_*" variable keeping track of how many
    % fitness calls, in one PSO simulation, presented flags equal to 1

        % Before calculating the Jacobian, eigenvalues and PSD:
        %-------------------------------------------------------

        %% Preliminarly cost 1.
            % "flag_limit_cycle" is 1 if the position x corresponds to a limit
            % cycle, 0 otherwise
            % the corresponding 'n_*' variable is "n_limit_cycle"

        %% Preliminarly cost 2.
            % "flag_NanInf_SP" is 1 if the singular point SP
            % (for the current configuration) contains any 'NaN' or 'Inf'
            % the corresponding 'n_*' variable is "n_nan_inf_SP"

        % After calculating the Jacobian, eigenvalues and PSD:
        %------------------------------------------------------

        %% Preliminarly cost 3.
            % "flag_NanInf_J" is 1 if the Jacobian matrix J of the linearized
            % system evaluated at the singular point SP contains any
            % 'Not a Number' or 'Infinite'

        %% Preliminarly cost 4.
            % "flag_eigen" is 1 if there is at least one positive eigenvalue,
            % 0 otherwise

    %--------------------------------------------------------------------
    %% IMPORTANT: both "costs" and Preliminarly costs are stored in the
    % variable "all_costs"
    % all_costs = [flag_limit_cycle, flag_NanInf_SP, flag_NanInf_J,
    %              flag_pos_eigenv, costs]
    %--------------------------------------------------------------------

function [f_min,SPE_NORM,all_costs, eigen_values] = fitness_1(x,nex, ind_input, ihe_soma,↙
lam)

%----------------------------------
```

```
%% Global variables
%-----------------------------------
    global n_nan_inf_SP
    global n_eigv_pos
    global n_nan_inf_J
    global n_limit_cycle

    global fitnessCounter
    fitnessCounter = fitnessCounter +1;

%-----------------------------------
%% Parameters and singular point
%-----------------------------------

    % Parameters of the search space
    hee_eq = x(1);
    gamma_ee = x(2);
    fsyn = x(3);

    % set of param initialised depending on search space
    param_set = par_set(nex,ind_input,ihe_soma,hee_eq,gamma_ee,fsyn,lam);

    % Singular point SP
    [SP,n_iter] = sing_point(param_set);

    % Parameters
    Sem = param_set(1,21);
    Sim = param_set(1,22);
    mue = param_set(1,23);
    mui = param_set(1,24);
    se = param_set(1,25);
    si = param_set(1,26);
    he_soma = SP(1,ihe_soma);
    hi = SP(1,nex+1);

    % Minimum and Maximum cost
    high_cost = 150;
    max_cost = 100;

    % Initialization of cost vector
    all_costs = zeros(1, 17);

%-------------------------------------------------------------------------
%% Preliminarly costs 1-2-3: flag variables --- Jacobian, eigenvalues
% and PSD
%-------------------------------------------------------------------------

    %-------------------------------------------------------------------
    %% Preliminarly cost 1.
    % Check if the point x correspond to a limit cycle
    %-------------------------------------------------------------------
    flag_limit_cycle = 0;
    if n_iter>5
        f_min = high_cost;
        n_limit_cycle = n_limit_cycle + 1;
        flag_limit_cycle = 1;
        SPE_NORM = zeros(1,501);
        eigen_values = zeros(45,1);
```

```
        all_costs(4) = flag_limit_cycle;
        return;
    end

    %-------------------------------------------------------------------
    %% Preliminarly cost 2.
    % Check if there are NaN or Inf elements in the singular point SP
    %-------------------------------------------------------------------
    flag_NanInf_SP = 0;
    if (sum(isnan(SP)>=1) || (sum(isinf(SP)>=1)) )
        f_min = high_cost;
        SPE_NORM = zeros(1,501);
        flag_NanInf_SP = 1;
        n_nan_inf_SP =n_nan_inf_SP+1;
        eigen_values = zeros(45,1);
        all_costs(1) = flag_NanInf_SP;
        return;
    end

    %-------------------------------
    %% Jacobian and Spectrum
    %-------------------------------
    [freq, SPE_NORM, eigen_values,nan_or_inf_J] = JacobianoManual(param_set, SP);
    delta_f = freq(2); % frequency step, freq is the vector freq = 0:0.5:250;


    %-------------------------------------------------------------------
    %% Preliminarly cost 3.
    % Check if there are NaN or Inf elements in the singular point SP
    %-------------------------------------------------------------------
    flag_NanInf_J = 0;
    if (nan_or_inf_J == 1)
            f_min = high_cost;
            SPE_NORM = zeros(1,501);
            flag_NanInf_J = 1;
            n_nan_inf_J =n_nan_inf_J+1;
            all_costs(2) = flag_NanInf_J;
        return;
    end

%-------------------------------------------------------------------
%% CRITERIA AND COSTS
%-------------------------------------------------------------------

    %----------------------------------------
    %% criteria 1: 0.1<= firing rate S<=20
    %----------------------------------------
        A1 = [0,1];
        A2 = [50,1];
        B1 = [0.1,0];
        B2 = [20,0];

        cost1_Se = cost_lines(A1,B1,A2,B2,S(he_soma,Sem,mue,se));

        cost1_Si = cost_lines(A1,B1,A2,B2,S(hi,Sim,mui,si));


    %----------------------------------------
```

```
%% frequency bands
%----------------------------------------
    init_delta = find(freq==0);
    final_delta = find(freq==3.5);
    init_theta = find(freq==4);
    final_theta = find(freq==7.5);
    init_alpha = find(freq==8);
    final_alpha = find(freq==12.5);
    init_beta = find(freq==13);
    final_beta = find(freq==30);


%----------------------------------------
%% criteria 2
%----------------------------------------

    % area in the delta region should be in between 15% and 50%
    A1 = [0,1];
    A2 = [1,1];
    B1 = [0.15,0];
    B2 = [0.5,0];
    area_delta = area(SPE_NORM, init_delta, final_delta, delta_f);
    cost_delta = cost_lines(A1,B1,A2,B2,area_delta);

    % area in the theta region should be in between 10% and 25%
    A1 = [0,1];
    A2 = [1,1];
    B1 = [0.1,0];
    B2 = [0.25,0];
    area_theta = area(SPE_NORM, init_theta, final_theta, delta_f);
    cost_theta = cost_lines(A1,B1,A2,B2,area_theta);

    % area in the alpha region should be in between 15% and 40%
    A1 = [0,1];
    A2 = [1,1];
    B1 = [0.15,0];
    B2 = [0.4,0];
    area_alpha = area(SPE_NORM, init_alpha, final_alpha, delta_f);
    cost_alpha = cost_lines(A1,B1,A2,B2,area_alpha);

    % area in the beta region should be in between 15% and 40%
    A1 = [0,1];
    A2 = [1,1];
    B1 = [0.15,0];
    B2 = [0.4,0];
    area_beta = area(SPE_NORM, init_beta, final_beta, delta_f);
    cost_beta = cost_lines(A1,B1,A2,B2,area_beta);

%----------------------------------------
%% criteria 3
%----------------------------------------

    % the ratio theta/delta should be smaller than 0.6
    A1 = [0,0];
    B1 = [0,0];
    A2 = [5,1];
    B2 = [0.6,0];
    ratio_theta_delta = area_theta/area_delta;
```

```
        cost_ratio_td = cost_lines(A1,B1,A2,B2,ratio_theta_delta);

        % the ratio theta/alpha should be smaller than 0.7
        A1 = [0,0];
        B1 = [0,0];
        A2 = [5,1];
        B2 = [0.7,0];
        ratio_theta_alpha = area_theta/area_alpha;
        cost_ratio_ta = cost_lines(A1,B1,A2,B2,ratio_theta_alpha);

    %----------------------------------------
    %% criteria 4
    %----------------------------------------

        % find freq for which 90% of area is before
        SEF_90 = 0;
        iii = 1;
         while SEF_90 < 0.9
             iii = iii + 1;
             SEF_90 = area(SPE_NORM, 1, iii, delta_f);
         end
        i_SEF_90 = iii;

        i_SEF_min = find(freq == 12);
        i_SEF_max = find(freq == 21);

        A1 = [1,1]; % note that here A1(1) cannot be 0, the minumum is 1 since it is the↙
index of a vector
        A2 = [final_beta,1];
        B1 = [i_SEF_min,0];
        B2 = [i_SEF_max,0];

        cost_SEF_90 = cost_lines(A1,B1,A2,B2,i_SEF_90);

    %----------------------------------------
    %% criteria 5
    %----------------------------------------

    delta_max = max(SPE_NORM(init_delta:final_delta));
    alpha_max = max(SPE_NORM(init_alpha:final_alpha));
    theta_min = min(SPE_NORM(init_theta:final_theta)) ;
    delta_min = min(SPE_NORM(init_delta:final_delta));

    % 1/3 delta_max <= alpha_max <= 5 delta_max
    A1 = [0,1];
    B1 = [1/3*delta_max,0];
    A2 = [10*delta_max,1];
    B2 = [5*delta_max,0];
    cost_alphamax_deltamax = cost_lines(A1,B1,A2,B2,alpha_max);

    % theta_min <= 1/2 alpha_max
    A1 = [0,0];
    B1 = [0,0];
    A2 = [alpha_max,1];
    B2 = [1/2*alpha_max,0];
    cost_thetamin_alphamax = cost_lines(A1,B1,A2,B2,theta_min);

    % theta_min <= 1/2 delta_max
```

```
    A1 = [0,0];
    B1 = [0,0];
    A2 = [delta_max,1];
    B2 = [1/2*delta_max,0];
    cost_thetamin_deltamax = cost_lines(A1,B1,A2,B2,theta_min);

    % theta_min <= 0.9 delta_min
    A1 = [0,0];
    B1 = [0,0];
    A2 = [delta_min,1];
    B2 = [0.9*delta_min,0];
    cost_thetamin_deltamin = cost_lines(A1,B1,A2,B2,theta_min);

% ************************************************
%% FITNESS EVALUATION f_min and vectors of costs
% ************************************************

    %-------------------------------------------------------------------
    %% Preliminarly cost 1.
    % Check if the point x correspond to a limit cycle
    %-------------------------------------------------------------------
    flag_pos_eigenv = 0;
    if max(real(eigen_values))>=  -10^-9
        flag_pos_eigenv = 1;
        n_eigv_pos = n_eigv_pos+1;
        f_min = max_cost*(1+max(real(eigen_values)));
    else
        f_min = max_cost*(cost1_Se + cost1_Si +  cost_delta + cost_theta + cost_alpha +↙
cost_beta + cost_ratio_td + cost_ratio_ta + cost_SEF_90 + cost_alphamax_deltamax +↙
cost_thetamin_alphamax + cost_thetamin_deltamax + cost_thetamin_deltamin)/13;
    end

    %--------------------------------
    %% Averaged costs and all_costs
    %--------------------------------
        costs = max_cost/13*[cost1_Se, cost1_Si, cost_delta, cost_theta, cost_alpha,↙
cost_beta, cost_ratio_td, cost_ratio_ta, cost_SEF_90, cost_alphamax_deltamax,↙
cost_thetamin_alphamax, cost_thetamin_deltamax, cost_thetamin_deltamin];

        all_costs = [flag_NanInf_SP, flag_NanInf_J, flag_pos_eigenv, flag_limit_cycle,↙
costs]; % size(all_costs,2) = 17;

%-------------------------------------------------------------------
%% Auxiliary functions
%-------------------------------------------------------------------

    % firing rate
    function y = S(h,S_max,mu_ba,sigmac)
        b = -sqrt(2);
        r_abs = 0;
        y = S_max./(1+(1-r_abs*S_max)*exp(b*(h-mu_ba)/sigmac));
    end

    % line through points A and B
    function y = line(A,B,z)
        mAB = (B(2)-A(2))/(B(1)-A(1));
        y = B(2) + mAB*(z-B(1));
    end
```

```
    % trapezoid area under the spectrum
    function y = area(fx, xinit, xfinal, deltax)
        y = 1/2*deltax*(fx(xinit) + fx(xfinal));
        i = 1;
        while i < xfinal - xinit
            y = y + deltax*fx(xinit+i);
            i = i + 1;
        end
    end

    %% Functions to calculate the costs
    function y = cost_lines(A1,B1,A2,B2,z)
        x_MIN = A1(1);
        x_min = B1(1);
        x_max = B2(1);
        x_MAX = A2(1);

        if A1(1)==B1(1) && A1(2)==B1(2) % case 1 line
            if z <= x_max
                y = 0;
            else
                if  z <= x_MAX
                    y = line(A2,B2,z);
                else
                    y = 1;
                end
            end
        end

        if (A1(1)==B1(1))==0 || (A1(2)==B1(2))==0 % case 2 or 3 lines
            if z<=x_MIN
                y = 1;
            else
                if z<=x_min
                    y = line(A1,B1,z);
                else
                    if z<=x_max
                        y = 0;
                    else
                        if z<=x_MAX
                            y = line(A2,B2,z);
                        else
                            y = 1;
                        end
                    end
                end
            end
        end
    end
end
```

```
%**************************************
%% Function "fitness_2"
%**************************************

%-------------------------------------------------------------------------------------------
%% This function evaluates the similarity between the PSD (at position x of the PSO
% search space) and typical PSDs encountered in humans

% The function is similar to fitness_1
% Criteria 5 are substituted by alternative criteria, more suitable to
% evaluate the similarity in the case the PSD does not exhibit an alpha
% peak.
% Criteria 2 are modified according to the explanation provided in
% Section 5.4.2
%-------------------------------------------------------------------------------------------

%-------------------------------------------------------------------------------------------
%% Inputs
%-------------------------------------------------------------------------------------------
% The similarity is evaluated at position x of the search space and for a configuration
% defined by the parameters:

        % "nex" = number of dendritic excitatory compartments
        % "ind_input" = index for the synaptic input
        % "ihe_soma" = index for the somatic compartment
        % "lam" = dendritic electrotonic length constant

%-------------------------------------------------------------------------------------------
%% Outputs
%-------------------------------------------------------------------------------------------

    % "SPE_NORM" is the normalised spectrum at position x and corresponding to
    % the configuration define by input parameters above

    % "f_min" is the fitness evaluation, normally varying from 0 to "high_cost,
    % where high_cost = 100.

    % The global variable "fitnessCounter" is the number of fitness call for the PSO
    % simulation

    %-----------------------------------------------------------------------
    %% Costs and fitness evaluation
    %-----------------------------------------------------------------------
    % The similarity of PSDs is calculated considering a number of criteria that
    % the current spectrum should fulfill (defined in Section 5.4.2).
    % A "cost" is associated to each criterium
    % (the cost is zero if the criterium is satisfied and increases linearly
    % otherwise (see function "cost_lines" below)).

    % For a detailed description of the criteria see the code below and the
    % explanation provided in Section 5.4.2

    % f_min is the averaged sum of all costs.
    % The costs are stored in a 13 dimension vector "costs"

    % The components of costs are:
            %1 cost1_Se                    criteria 1
            %2 cost1_Si
```

```
        %3 cost_delta                   criteria 2
        %4 cost_theta
        %5 cost_alpha
        %6 cost_beta

        %7 cost_ratio_td                criteria 3
        %8 cost_ratio_ta

        %9 cost_SEF_90                  criterium 4

        %10 cost_alphamax_deltamax      criteria 5
        %11 cost_thetamin_alphamax
        %12 cost_thetamin_deltamax
        %13 cost_thetamin_deltamin

%-------------------------------------------------------------------------
%% Preliminarly costs: flag variables
%-------------------------------------------------------------------------
% A number of 'flag' variables control the behaviour of the system.
% The flagged cases (flag = 1) correspond to configurations
% presenting numerical problems (as computations involving 'NaN' or 'Inf'),
% or to configurations where the linearization cannot be performed, as the
% case where positive eigenvalues occur. In these cases, the fitness
% evaluation is set to high_cost and both the eigenvalues and the spectrum
% are set to zero.

% Each flag variable is associated to a "n_*" variable keeping track of how many
% fitness calls, in one PSO simulation, presented flags equal to 1

    % Before calculating the Jacobian, eigenvalues and PSD:
    %-------------------------------------------------------

    %% Preliminarly cost 1.
        % "flag_limit_cycle" is 1 if the position x corresponds to a limit
        % cycle, 0 otherwise
        % the corresponding 'n_*' variable is "n_limit_cycle"

    %% Preliminarly cost 2.
        % "flag_NanInf_SP" is 1 if the singular point SP
        % (for the current configuration) contains any 'NaN' or 'Inf'
        % the corresponding 'n_*' variable is "n_nan_inf_SP"

    % After calculating the Jacobian, eigenvalues and PSD:
    %-------------------------------------------------------

    %% Preliminarly cost 3.
        % "flag_NanInf_J" is 1 if the Jacobian matrix J of the linearized
        % system evaluated at the singular point SP contains any
        % 'Not a Number' or 'Infinite'

    %% Preliminarly cost 4.
        % "flag_eigen" is 1 if there is at least one positive eigenvalue,
        % 0 otherwise

%-------------------------------------------------------------------------
%% IMPORTANT: both "costs" and Preliminarly costs are stored in the
% variable "all_costs"
% all_costs = [flag_limit_cycle, flag_NanInf_SP, flag_NanInf_J,
```

```
    %              flag_pos_eigenv, costs]
    %-------------------------------------------------------------------------

function [f_min,SPE_NORM,cost_vector, eigen_values] = fitness_2(x,nex, ind_input,↙
ihe_soma,lam)

%-----------------------------------
%% Global variables
%-----------------------------------
    global n_nan_inf_SP
    global n_eigv_pos
    global n_nan_inf_J
    global n_limit_cycle

    global fitnessCounter
    fitnessCounter = fitnessCounter +1;

%-----------------------------------
%% Parameters and singular point
%-----------------------------------
    hee_eq = x(1);
    gamma_ee = x(2);
    fsyn = x(3);

    param_set = par_set(nex,ind_input,ihe_soma,hee_eq,gamma_ee,fsyn,lam);
    [SP,n_iter] = sing_point(param_set);

    Sem = param_set(1,21);
    Sim = param_set(1,22);
    mue = param_set(1,23);
    mui = param_set(1,24);
    se = param_set(1,25);
    si = param_set(1,26);
    he_soma = SP(1,ihe_soma);
    hi = SP(1,nex+1);

    high_cost = 150;
    max_cost = 100;

    cost_vector = zeros(1, 17);

%-------------------------------------------------------------------------
%% Preliminarly costs 1-2-3: flag variables --- Jacobian, eigenvalues
% and PSD
%-------------------------------------------------------------------------

    %---------------------------------------------------------------
    %% Preliminarly cost 1.
    % Check if the point x correspond to a limit cycle
    %---------------------------------------------------------------
    flag_limit_cycle = 0;
    if n_iter>5
        f_min = high_cost;
        n_limit_cycle = n_limit_cycle + 1;
        flag_limit_cycle = 1;
        SPE_NORM = zeros(1,501);
        eigen_values = zeros(45,1);
        all_costs(4) = flag_limit_cycle;
```

```
        return;
    end

    %-------------------------------------------------------------------
    %% Preliminarly cost 2.
    % Check if there are NaN or Inf elements in the singular point SP
    %-------------------------------------------------------------------
    flag_NanInf_SP = 0;
    if (sum(isnan(SP)>=1) || (sum(isinf(SP)>=1)) )
        f_min = high_cost;
        SPE_NORM = zeros(1,501);
        flag_NanInf_SP = 1;
        n_nan_inf_SP =n_nan_inf_SP+1;
        eigen_values = zeros(45,1);
        all_costs(1) = flag_NanInf_SP;
        return;
    end

    %-------------------------------
    %% Jacobian and Spectrum
    %-------------------------------
    [freq, SPE_NORM, eigen_values,nan_or_inf_J] = JacobianoManual(param_set, SP);
    delta_f = freq(2); % frequency step, freq is the vector freq = 0:0.5:250;

    %-------------------------------------------------------------------
    %% Preliminarly cost 3.
    % Check if there are NaN or Inf elements in the singular point SP
    %-------------------------------------------------------------------
    flag_NanInf_J = 0;
    if (nan_or_inf_J == 1)
            f_min = high_cost;
            SPE_NORM = zeros(1,501);
            flag_NanInf_J = 1;
            n_nan_inf_J =n_nan_inf_J+1;
            all_costs(2) = flag_NanInf_J;
        return;
    end

%-----------------------------------------------------------------------
%% CRITERIA AND COSTS
%-----------------------------------------------------------------------↙
-------------------------------------------------------------

    %----------------------------------------
    %% criteria 1: 0.1<= firing rate S<=20
    %----------------------------------------
        A1 = [0,1];
        A2 = [50,1];
        B1 = [0.1,0];
        B2 = [20,0];

        cost1_Se = cost_lines(A1,B1,A2,B2,S(he_soma,Sem,mue,se));

        cost1_Si = cost_lines(A1,B1,A2,B2,S(hi,Sim,mui,si));

    %----------------------------------------
    %% frequency bands
    %----------------------------------------
```

```
        init_delta = find(freq==0);
        final_delta = find(freq==3.5);
        init_theta = find(freq==4);
        final_theta = find(freq==7.5);
        init_alpha = find(freq==8);
        final_alpha = find(freq==12.5);
        init_beta = find(freq==13);
        final_beta = find(freq==30);

    %----------------------------------------
    %% criteria 2
    %----------------------------------------

        %-----------------------------------------------------------------
        % area in the delta region should be in between 15% and 50%
        %-----------------------------------------------------------------
        A1 = [0,1];
        A2 = [1,1];
        B1 = [0.15,0];
        B2 = [0.5,0];
        area_delta = area(SPE_NORM, init_delta, final_delta, delta_f);
        cost_delta = cost_lines(A1,B1,A2,B2,area_delta);

        %-----------------------------------------------------------------
        % area in the theta region should be in between 10% and 25%
        %-----------------------------------------------------------------
        A1 = [0,1];
        A2 = [1,1];
        B1 = [0.08,0]; %++++++++ modified from fit 1 0.1 to 0.08
        B2 = [0.25,0];
        area_theta = area(SPE_NORM, init_theta, final_theta, delta_f);
        cost_theta = cost_lines(A1,B1,A2,B2,area_theta);

        %-----------------------------------------------------------------
        % area in the alpha region should be in between 30% and 50%
        %-----------------------------------------------------------------
        % for simulation of Chapter 5
        % A1 = [0,50];
        % for simulation of Chapter 6
        A1 = [0,1]; % same as fit 1ù

        A2 = [1,1];

        %+++++++++++++++++++++++++++++++++++++++++++ modified from fit 1
        %B1 = [0.15,0];
        %B2 = [0.4,0];
        B1 = [0.3,0];
        B2 = [0.5,0];
        %+++++++++++++++++++++++++++++++++++++++++++
        area_alpha = area(SPE_NORM, init_alpha, final_alpha, delta_f);
        cost_alpha = cost_lines(A1,B1,A2,B2,area_alpha);

        %-----------------------------------------------------------------
        % area in the beta region should be in between 15% and 40%
        %-----------------------------------------------------------------

        %+++++++++++++++++++++++++++
        A1 = [0,0.5]; % modified 1-->0.5
```

```
        A2 = [1,1];
        %++++++++++++++++++++++++++++
        modified
        %B1 = [0.15,0];
        %B2 = [0.4,0];
        B1 = [0.05,0];
        B2 = [0.25,0];
        %++++++++++++++++++++++++++++
        area_beta = area(SPE_NORM, init_beta, final_beta, delta_f);
        cost_beta = cost_lines(A1,B1,A2,B2,area_beta);

    %----------------------------------------
    %% criteria 3
    %----------------------------------------

        % the ratio theta/delta should be smaller than 0.6
        A1 = [0,0];
        B1 = [0,0];
        A2 = [5,1];
        B2 = [0.6,0];
        ratio_theta_delta = area_theta/area_delta;
        cost_ratio_td = cost_lines(A1,B1,A2,B2,ratio_theta_delta);

        % the ratio theta/alpha should be smaller than 0.7
        A1 = [0,0];
        B1 = [0,0];
        A2 = [5,1];
        B2 = [0.7,0];
        ratio_theta_alpha = area_theta/area_alpha;
        cost_ratio_ta = cost_lines(A1,B1,A2,B2,ratio_theta_alpha);

    %----------------------------------------
    %% criteria 4
    %----------------------------------------

        % find freq for which 90% of area is before
        SEF_90 = 0;
        iii = 1;
         while SEF_90 < 0.9
             iii = iii + 1;
             SEF_90 = area(SPE_NORM, 1, iii, delta_f);
         end
        i_SEF_90 = iii;

        i_SEF_min = find(freq == 12);
        i_SEF_max = find(freq == 21);

        A1 = [1,1]; % note that here A1(1) cannot be 0, the minumum is 1 since it is the↙
index of a vector
        A2 = [final_beta,1];
        B1 = [i_SEF_min,0];
        B2 = [i_SEF_max,0];

        cost_SEF_90 = cost_lines(A1,B1,A2,B2,i_SEF_90);

    %----------------------------------------
    %% criteria 5 substituted
    %----------------------------------------
```

```
        % Defining some critical requencies
        max_after4Hz = max(SPE_NORM(init_theta:final_beta));
        A1 = [init_theta,1];
        B1 = [init_alpha,0];
        A2 = [final_beta,1];
        B2 = [final_alpha,0];

        % If there is no peak in the alpha region...
        if max_after4Hz > SPE_NORM(final_theta) && (find(SPE_NORM == max_after4Hz) <↙
init_alpha  || find(SPE_NORM  == max_after4Hz) > final_alpha)
            i_max_after4Hz = find(SPE_NORM == max_after4Hz);

            cost_peak_not_in_alpha = cost_lines(A1,B1,A2,B2,i_max_after4Hz);
            cost_alphamax_deltamax = cost_peak_not_in_alpha;
            cost_thetamin_alphamax = cost_peak_not_in_alpha;
            cost_thetamin_deltamax = cost_peak_not_in_alpha;
            cost_thetamin_deltamin = cost_peak_not_in_alpha;
        else
            %... if there is a peak in the alpha region criteria 5 are as
            % in fitness 1

            delta_max = max(SPE_NORM(init_delta:final_delta));
            alpha_max = max(SPE_NORM(init_alpha:final_alpha));
            theta_min = min(SPE_NORM(init_theta:final_theta)) ;
            delta_min = min(SPE_NORM(init_delta:final_delta));

            % 1/3 delta_max <= alpha_max <= 5 delta_max
            A1 = [0,1];
            B1 = [1/3*delta_max,0];
            A2 = [10*delta_max,1];
            B2 = [5*delta_max,0];
            cost_alphamax_deltamax = cost_lines(A1,B1,A2,B2,alpha_max);


            % theta_min <= 1/2 alpha_max
            A1 = [0,0];
            B1 = [0,0];
            A2 = [alpha_max,1];
            B2 = [1/2*alpha_max,0];
            cost_thetamin_alphamax = cost_lines(A1,B1,A2,B2,theta_min);


            % theta_min <= 1/2 delta_max
            A1 = [0,0];
            B1 = [0,0];
            A2 = [delta_max,1];
            B2 = [1/2*delta_max,0];
            cost_thetamin_deltamax = cost_lines(A1,B1,A2,B2,theta_min);


            % theta_min <= 0.9 delta_min
            A1 = [0,0];
            B1 = [0,0];
            A2 = [delta_min,1];
            B2 = [0.9*delta_min,0];
            cost_thetamin_deltamin = cost_lines(A1,B1,A2,B2,theta_min);
        end
```

```
% ************************************************
%% FITNESS EVALUATION f_min and vectors of costs
% ************************************************

    %-------------------------------------------------------------------
    %% Preliminarly cost 1.
    % Check if the point x correspond to a limit cycle
    %-------------------------------------------------------------------
    flag_pos_eigenv = 0;
    if max(real(eigen_values))>=  -10^-9
        flag_pos_eigenv = 1;
        n_eigv_pos = n_eigv_pos+1;
        f_min = max_cost*(1+max(real(eigen_values)));
    else
        f_min = max_cost*(cost1_Se + cost1_Si +  cost_delta + cost_theta + cost_alpha +↙
cost_beta + cost_ratio_td + cost_ratio_ta + cost_SEF_90 + cost_alphamax_deltamax +↙
cost_thetamin_alphamax + cost_thetamin_deltamax + cost_thetamin_deltamin)/13;
    end

    %-------------------------------
    %% Averaged costs and all_costs
    %-------------------------------
        costs = max_cost/13*[cost1_Se, cost1_Si, cost_delta, cost_theta, cost_alpha,↙
cost_beta, cost_ratio_td, cost_ratio_ta, cost_SEF_90, cost_alphamax_deltamax,↙
cost_thetamin_alphamax, cost_thetamin_deltamax, cost_thetamin_deltamin];

        all_costs = [flag_NanInf_SP, flag_NanInf_J, flag_pos_eigenv, flag_limit_cycle,↙
costs]; % size(all_costs,2) = 17;

%-------------------------------------------------------------------
%% Auxiliary functions
%-------------------------------------------------------------------

    % firing rate
    function y = S(h,S_max,mu_ba,sigmac)
        b = -sqrt(2);
        r_abs = 0;
        y = S_max./(1+(1-r_abs*S_max)*exp(b*(h-mu_ba)/sigmac));
    end

    % line through points A and B
    function y = line(A,B,z)
        mAB = (B(2)-A(2))/(B(1)-A(1));
        y = B(2) + mAB*(z-B(1));
    end

    % trapezoid area under the spectrum
    function y = area(fx, xinit, xfinal, deltax)
        y = 1/2*deltax*(fx(xinit) + fx(xfinal));
        i = 1;
        while i < xfinal - xinit
            y = y + deltax*fx(xinit+i);
            i = i + 1;
        end
    end

    %% Functions to calculate the costs
```

```
    function y = cost_lines(A1,B1,A2,B2,z)
        x_MIN = A1(1);
        x_min = B1(1);
        x_max = B2(1);
        x_MAX = A2(1);

        if A1(1)==B1(1) && A1(2)==B1(2) % case 1 line
            if z <= x_max
                y = 0;
            else
                if  z <= x_MAX
                    y = line(A2,B2,z);
                else
                    y = 1;
                end
            end
        end

        if (A1(1)==B1(1))==0 || (A1(2)==B1(2))==0 % case 2 or 3 lines
            if z<=x_MIN
                y = 1;
            else
                if z<=x_min
                    y = line(A1,B1,z);
                else
                    if z<=x_max
                        y = 0;
                    else
                        if z<=x_MAX
                            y = line(A2,B2,z);
                        else
                            y = 1;
                        end
                    end
                end
            end
        end
    end
end
```

Bibliography

[1] R. F. Betzel and D. S. Bassett. Multi-scale brain networks. *NeuroImage*, 160:73–83, 2017.

[2] A. L. Allegra Mascaro, L. Silvestri, L. Sacconi, and F. Pavone. Towards a comprehensive understanding of brain machinery by correlative microscopy. *Journal of biomedical optics*, 20:61105, 6 2015.

[3] P. C. Bressloff and S. Coombes. Physics of the Extended Neuron. *International Journal of Modern Physics B*, 11(20):2343–2392, 8 1997.

[4] W. Gerstner and W. Kistler. *Spiking Neuron Models: An Introduction*. Cambridge University Press, New York, NY, USA, 2002.

[5] Peiji LiangSi WuFanji Gu. *Single Neuron Models In: An Introduction to Neural Information Processing*. Springer, Dordrecht, 2016.

[6] H. Markram, E. Muller, S. Ramaswamy, M. Reimann, M. Abdellah, C. Sanchez, A. Ailamaki, L. Alonso-Nanclares, N. Antille, S. Arsever, G. Kahou, T. Berger, A. Bilgili, N. Buncic, A. Chalimourda, G. Chindemi, J.-D. Courcol, F. Delalondre, V. Delattre, S. Druckmann, R. Dumusc, J. Dynes, S. Eilemann, E. Gal, M. Gevaert, J.-P. Ghobril, A. Gidon, J. Graham, A. Gupta, V. Haenel, E. Hay, T. Heinis, J. Hernando, M. Hines, L. Kanari, D. Keller, J. Kenyon, G. Khazen, Y. Kim, J. King, Z. Kisvarday, P. Kumbhar, S. Lasserre, J.-V. Le Bé, B. Magalhães, A. Merchán-Pérez, J. Meystre, B. Morrice, J. Muller, A. Muñoz-Céspedes, S. Muralidhar, K. Muthurasa, D. Nachbaur, T. Newton, M. Nolte, A. Ovcharenko, J. Palacios, L. Pastor, R. Perin, R. Ranjan, I. Riachi,

J.-R. Rodríguez, J. Riquelme, C. Rössert, K. Sfyrakis, Y. Shi, J. Shillcock, G. Silberberg, R. Silva, F. Tauheed, M. Telefont, M. Toledo-Rodriguez, T. Tränkler, W. Van Geit, J. Díaz, R. Walker, Y. Wang, S. Zaninetta, J. DeFelipe, S. Hill, I. Segev, and F. Schürmann. Reconstruction and Simulation of Neocortical Microcircuitry. *Cell*, 163(2):456–492, 2015.

[7] Q. J. M. Huys, M. B. Ahrens, and L. Paninski. Efficient Estimation of Detailed Single-Neuron Models. *Journal of Neurophysiology*, 96(2):872–890, 8 2006.

[8] W. J. Freeman. Tutorial on Neurobiology: from single neurons to brain chaos. *International Journal of Bifurcation and Chaos*, 02(03):451–482, 9 1992.

[9] W. Freeman. *Neurodynamics: An Exploration in Mesoscopic Brain Dynamics*. 2000.

[10] W. J. Freeman. *Mass Action in the Nervous System*. Elsevier, 1975.

[11] Hans Liljenström. Mesoscopic brain dynamics., 2012.

[12] K. J. Friston, L. Harrison, and W. Penny. Dynamic causal modelling. *NeuroImage*, 19(4):1273–1302, 8 2003.

[13] K. E. Stephan, W. D. Penny, R. J. Moran, H. E. M. den Ouden, J. Daunizeau, and K. J. Friston. Ten simple rules for dynamic causal modeling. *NeuroImage*, 49(4):3099–3109, 2 2010.

[14] Stephen Grossberg. *Studies of Mind and Brain*. D. Reideil Publishing Company, Dordrecht, Boston, London, 1982.

[15] Erol Basar. *Memory and Brain Dynamics*. Boca Raton: CRC Press,, 2004.

[16] S. Grossberg. The Link between Brain Learning, Attention, and Consciousness. *Consciousness and Cognition*, 8(1):1–44, 1999.

[17] T. Yamanaka and N. Nukina. ER Dynamics and Derangement in Neurological Diseases. *Frontiers in Neuroscience*, 12:91, 2018.

[18] The Dynamics of Neurological Disease: Current Views and Key Issues. In *Neural Dynamics of Neurological Disease*, chapter 1, pages 1–17. John Wiley & Sons, Ltd, 2017.

[19] The fine structure of the nervous system: The neurons and supporting cells, by Alan Peters, Sanford L. Palay, and Henry Def. Webster, 395 pp, W. B. Saunders Company, Philadelphia, 1976. *Annals of Neurology*, 4(6):588, 1978.

[20] T. E. Abdel-Maguid and D. Bowsher. Classification of neurons by dendritic branching pattern. A categorisation based on Golgi impregnation of spinal and cranial somatic and visceral afferent and efferent cells in the adult human. *Journal of anatomy*, 138 (Pt 4)(Pt 4):689–702, 6 1984.

[21] L. Costa, K. Zawadzki, M. Miazaki, M. Viana, and S. Taraskin. Unveiling the Neuromorphological Space. *Frontiers in Computational Neuroscience*, 4:150, 2010.

[22] R. Parekh and G. Ascoli. Neuronal Morphology Goes Digital: A Research Hub for Cellular and System Neuroscience. *Neuron*, 77(6):1017–1038, 2013.

[23] Douglas C. Wallace. Bioenergetic origins of complexity and disease. *Cold Spring Harbor symposia on quantitative biology*, 76:1–16, 2011.

[24] R. H. Edwards. The Neurotransmitter Cycle and Quantal Size. *Neuron*, 55(6):835–858, 2007.

[25] T. C. Südhof. Neuroligins and neurexins link synaptic function to cognitive disease. *Nature*, 455:903, 10 2008.

[26] K. Matsuda and M. Yuzaki. Cbln1 and the Delta2 Glutamate Receptor—An Orphan Ligand and an Orphan Receptor Find Their Partners. *The Cerebellum*, 11(1):78–84, 2012.

[27] H. Takahashi and A. M. Craig. Protein tyrosine phosphatases PTPδ, PTPσ, and LAR: presynaptic hubs for synapse organization. *Trends in Neurosciences*, 36(9):522–534, 2013.

[28] T. C. Südhof. Molecular Neuroscience in the 21st Century: A Personal Perspective. *Neuron*, 96(3):536–541, 2017.

[29] M. Häusser and B. Mel. Dendrites: bug or feature? *Current opinion in neurobiology*, 13(3):372–83, 6 2003.

[30] S. N. H. M. Stuart, G. *Dendrites*. Oxford University Press, 2016.

[31] M. Häusser. Synaptic function: Dendritic democracy. *Current Biology*, 11(1):R10–R12, 2001.

[32] B. K. Andrasfalvy and J. C. Magee. Distance-Dependent Increase in AMPA Receptor Number in the Dendrites of Adult Hippocampal CA1 Pyramidal Neurons. *J. Neurosci.*, 21(23):9151–9159, 12 2001.

[33] M. A. Smith, G. C. R. Ellis-Davies, and J. C. Magee. Mechanism of the distance-dependent scaling of Schaffer collateral synapses in rat CA1 pyramidal neurons. *The Journal of physiology*, 548(Pt 1):245–258, 4 2003.

[34] Y. Timofeeva, S. J Cox, S. Coombes, and K. Josic. Democratization in a passive dendritic tree: An analytical investigation. *Journal of computational neuroscience*, 25:228–244, 3 2008.

[35] D. Liley, B. Foster, and I. Bojak. Co-operative populations of neurons: mean field models of mesoscopic brain activity. *Computational Systems NeurobiologyLiley, D., Foster, B., & Bojak, I. (2012). Co-operative populations of neurons: mean field models of mesoscopic brain activity. Computational Systems Neurobiology, 1–50. Retrieved from http://link.springer.com/chapter/10.*, pages 1–50, 2012.

[36] I. Bojak and D. Liley. Modeling the effects of anesthesia on the electroencephalogram. *Physical Review E*, 71(4):041902, 4 2005.

[37] W. Rall. Theoretical significance of dendritic trees for neuronal input-output relations. 1 1964.

[38] I. Bojak and M. Breakspear. Neuroimaging, Neural Population Models for. In D. Jaeger and R. Jung, editors, *Encyclopedia of Computational Neuroscience*, pages 1–29. Springer New York, New York, NY, 2013.

[39] G. Perea, M. Navarrete, and A. Araque. Tripartite synapses: astrocytes process and control synaptic information. *Trends in neurosciences*, 32(8):421–31, 8 2009.

[40] G. Perea and A. Araque. GLIA modulates synaptic transmission. *Brain research reviews*, 63(1-2):93–102, 5 2010.

[41] G. Deco, V. K. Jirsa, P. A. Robinson, M. Breakspear, and K. Friston. The dynamic brain: from spiking neurons to neural masses and cortical fields. *PLoS computational biology*, 4(8):e1000092, 1 2008.

[42] D. Liley, P. Cadusch, and M. Dafilis. A spatially continuous mean field theory of electrocortical activity. 8 2009.

[43] H. R. Wilson and J. D. Cowan. Excitatory and inhibitory interactions in localized populations of model neurons. *Biophysical journal*, 12(1):1–24, 1 1972.

[44] H. R. Wilson and J. D. Cowan. A mathematical theory of the functional dynamics of cortical and thalamic nervous tissue. *Kybernetik*, 13(2):55–80, 9 1973.

[45] S. Amari. Homogeneous nets of neuron-like elements. *Biological cybernetics*, 17(4):211–20, 1 1975.

[46] S. Amari. Dynamics of pattern formation in lateral-inhibition type neural fields. *Biological cybernetics*, 27(2):77–87, 8 1977.

[47] R. L. Beurle. Properties of a Mass of Cells Capable of Regenerating Pulses. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 240(669):55–94, 8 1956.

[48] J. S. Griffith. A field theory of neural nets: I: Derivation of field equations. *The Bulletin of Mathematical Biophysics*, 25(1):111–120, 3 1963.

[49] J. S. Griffith. A field theory of neural nets: II. Properties of the field equations. *The Bulletin of Mathematical Biophysics*, 27(2):187–195, 6 1965.

[50] F. H. Lopes da Silva, A. Hoeks, H. Smits, and L. H. Zetterberg. Model of brain rhythmic activity. *Kybernetik*, 15(1):27–37, 1974.

[51] B. H. Jansen and V. G. Rit. Electroencephalogram and visual evoked potential generation in a mathematical model of coupled cortical columns. *Biological cybernetics*, 73(4):357–66, 9 1995.

[52] P. A. Robinson, C. J. Rennie, J. J. Wright, H. Bahramali, E. Gordon, and D. L. Rowe. Prediction of electroencephalographic spectra from neurophysiology. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 63(2 Pt 1):021903, 2 2001.

[53] F. Wendling, A. Hernandez, J.-J. Bellanger, P. Chauvel, and F. Bartolomei. Interictal to ictal transition in human temporal lobe epilepsy: insights from a computational model of intracerebral EEG. *Journal of clinical neurophysiology : official publication of the American Electroencephalographic Society*, 22(5):343–56, 10 2005.

[54] V. Jirsa and H. Haken. Field Theory of Electromagnetic Brain Activity. *Physical review letters*, 77(5):960–963, 7 1996.

[55] S. Coombes, N. Venkov, L. Shiau, I. Bojak, D. Liley, and C. Laing. Modeling electrocortical activity through improved local approximations of integral neural field equations. *Physical Review E*, 76(5):051901, 11 2007.

[56] S. Coombes. Large-scale neural dynamics: simple and complex. *NeuroImage*, 52(3):731–9, 9 2010.

[57] M. L. Steyn-Ross, D. A. Steyn-Ross, J. W. Sleigh, and D. T. Liley. Theoretical electroencephalogram stationary spectrum for a white-noise-driven cortex:

evidence for a general anesthetic-induced phase transition. *Physical review. E, Statistical physics, plasmas, fluids, and related interdisciplinary topics*, 60(6 Pt B):7299–311, 12 1999.

[58] M. L. Steyn-Ross, D. A. Steyn-Ross, and J. W. Sleigh. Modelling general anaesthesia as a first-order phase transition in the cortex. *Progress in biophysics and molecular biology*, 85(2-3):369–85, 1 2004.

[59] D. T. Liley, P. J. Cadusch, and J. J. Wright. A continuum theory of electrocortical activity. *Neurocomputing*, 26-27:795–800, 6 1999.

[60] A. J. K. Phillips and P. A. Robinson. A quantitative model of sleep-wake dynamics based on the physiology of the brainstem ascending arousal system. *Journal of biological rhythms*, 22(2):167–79, 4 2007.

[61] D. A. Steyn-Ross, M. L. Steyn-Ross, J. W. Sleigh, M. T. Wilson, I. P. Gillies, and J. J. Wright. The sleep cycle modelled as a cortical phase transition. *Journal of biological physics*, 31(3-4):547–69, 12 2005.

[62] M. Steyn-Ross, D. Steyn-Ross, J. Sleigh, M. Wilson, and L. Wilcocks. Proposed mechanism for learning and memory erasure in a white-noise-driven sleeping cortex. *Physical Review E*, 72(6):061910, 12 2005.

[63] M. T. Wilson, M. L. Steyn-Ross, D. A. Steyn-Ross, and J. W. Sleigh. Going beyond a mean-field model for the learning cortex: second-order statistics. *Journal of biological physics*, 33(3):213–46, 6 2007.

[64] M. L. Steyn-Ross, D. A. Steyn-Ross, M. T. Wilson, and J. W. Sleigh. Modeling brain activation patterns for the default and cognitive states. *NeuroImage*, 45(2):298–311, 4 2009.

[65] M. L. Steyn-Ross, D. A. Steyn-Ross, J. W. Sleigh, and M. T. Wilson. A mechanism for ultra-slow oscillations in the cortical default network. *Bulletin of mathematical biology*, 73(2):398–416, 2 2011.

[66] G. Deco and E. T. Rolls. Neurodynamics of biased competition and cooperation for attention: a model with spiking neurons. *Journal of neurophysiology*, 94(1):295–313, 7 2005.

[67] E. Mavritsaki, D. Heinke, H. Allen, G. Deco, and G. W. Humphreys. Bridging the gap between physiology and behavior: evidence from the sSoTS model of human visual attention. *Psychological review*, 118(1):3–41, 1 2011.

[68] N. Brunel and X.-J. Wang. Effects of Neuromodulation in a Cortical Network Model of Object Working Memory Dominated by Recurrent Inhibition. *Journal of Computational Neuroscience*, 11(1):63–85.

[69] J. A. S. Kelso. *Dynamic Patterns: The Self-organization of Brain and Behavior*. 1997.

[70] K. J. Friston. Transients, metastability, and neuronal dynamics. *NeuroImage*, 5(2):164–71, 2 1997.

[71] S. L. Bressler and J. A. Kelso. Cortical coordination dynamics and cognition. *Trends in cognitive sciences*, 5(1):26–36, 1 2001.

[72] K. J. Friston. The labile brain. I. Neuronal transients and nonlinear coupling. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 355(1394):215–36, 2 2000.

[73] W. J. Freeman and M. D. Holmes. Metastability, instability, and state transition in neocortex. *Neural networks : the official journal of the International Neural Network Society*, 18(5-6):497–504, 1.

[74] F. H. Lopes da Silva, W. Blanes, S. N. Kalitzin, J. Parra, P. Suffczynski, and D. N. Velis. Dynamical diseases of brain systems: different routes to epileptic seizures. *IEEE transactions on bio-medical engineering*, 50(5):540–8, 5 2003.

[75] F. Wendling, J. J. Bellanger, F. Bartolomei, and P. Chauvel. Relevance of nonlinear lumped-parameter models in the analysis of depth-EEG epileptic signals. *Biological Cybernetics*, 83(4):367–378, 9 2000.

[76] P. Robinson, C. Rennie, and D. Rowe. Dynamics of large-scale brain activity in normal arousal states and epileptic seizures. *Physical Review E*, 65(4):041924, 4 2002.

[77] M. Breakspear, J. A. Roberts, J. R. Terry, S. Rodrigues, N. Mahant, and P. A. Robinson. A unifying explanation of primary generalized seizures through nonlinear brain modeling and bifurcation analysis. *Cerebral cortex (New York, N.Y. : 1991)*, 16(9):1296–313, 9 2006.

[78] M. A. Kramer, H. E. Kirsch, and A. J. Szeri. Pathological pattern formation and cortical propagation of epileptic seizures. *Journal of the Royal Society, Interface / the Royal Society*, 2(2):113–27, 3 2005.

[79] P. Suffczynski, F. Lopes da Silva, J. Parra, D. Velis, and S. Kalitzin. Epileptic transitions: model predictions and experimental validation. *Journal of clinical neurophysiology : official publication of the American Electroencephalographic Society*, 22(5):288–99, 10 2005.

[80] B. Molaee-Ardekani, P. Benquet, F. Bartolomei, and F. Wendling. Computational modeling of high-frequency oscillations at the onset of neocortical partial seizures: from 'altered structure' to 'dysfunction'. *NeuroImage*, 52(3):1109–22, 9 2010.

[81] F. Marten, S. Rodrigues, O. Benjamin, M. P. Richardson, and J. R. Terry. Onset of polyspike complexes in a mean-field model of human electroencephalography and its application to absence epilepsy. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 367(1891):1145–61, 3 2009.

[82] I. Bojak and D. Liley. Self-organized 40Hz synchronization in a physiological theory of EEG. *Neurocomputing*, 70(10-12):2085–2090, 6 2007.

[83] P. Robinson. Patchy propagators, brain dynamics, and the generation of spatially structured gamma oscillations. *Physical Review E*, 73(4):041904, 4 2006.

[84] J. J. Wright. EEG simulation: variation of spectral envelope, pulse synchrony and approximately 40 Hz oscillation. *Biological cybernetics*, 76(3):181–94, 3 1997.

[85] C. J. Rennie, J. J. Wright, and P. A. Robinson. Mechanisms of Cortical Electrical Activity and Emergence of Gamma Rhythm. *Journal of Theoretical Biology*, 205(1):17–35, 7 2000.

[86] C. J. Honey, R. Kötter, M. Breakspear, and O. Sporns. Network structure of cerebral cortex shapes functional connectivity on multiple time scales. *Proceedings of the National Academy of Sciences of the United States of America*, 104(24):10240–5, 6 2007.

[87] G. Deco, V. Jirsa, A. R. McIntosh, O. Sporns, and R. Kötter. Key role of coupling, delay, and noise in resting brain fluctuations. *Proceedings of the National Academy of Sciences of the United States of America*, 106(25):10302–7, 6 2009.

[88] A. Ghosh, Y. Rho, A. R. McIntosh, R. Kötter, and V. K. Jirsa. Noise during rest enables the exploration of the brain's dynamic repertoire. *PLoS computational biology*, 4(10):e1000196, 10 2008.

[89] G. Deco, V. K. Jirsa, and A. R. McIntosh. Emerging concepts for the dynamical organization of resting-state activity in the brain. *Nature reviews. Neuroscience*, 12(1):43–56, 1 2011.

[90] C. J. Honey, O. Sporns, L. Cammoun, X. Gigandet, J. P. Thiran, R. Meuli, and P. Hagmann. Predicting human resting-state functional connectivity from structural connectivity. *Proceedings of the National Academy of Sciences of the United States of America*, 106(6):2035–40, 2 2009.

[91] A. Babajani, M.-H. Nekooei, and H. Soltanian-Zadeh. Integrated MEG and fMRI model: synthesis and analysis. *Brain topography*, 18(2):101–13, 1 2005.

[92] A. Babajani and H. Soltanian-Zadeh. Integrated MEG/EEG and fMRI model based on neural masses. *IEEE transactions on bio-medical engineering*, 53(9):1794–801, 9 2006.

[93] A. Babajani-Feremi, H. Soltanian-Zadeh, and J. E. Moran. Integrated MEG/fMRI model validated using real auditory data. *Brain topography*, 21(1):61–74, 9 2008.

[94] A. Babajani-Feremi and H. Soltanian-Zadeh. Multi-area neural mass modeling of EEG and MEG signals. *NeuroImage*, 52(3):793–811, 9 2010.

[95] J. Riera, E. Aubert, K. Iwata, R. Kawashima, X. Wan, and T. Ozaki. Fusing EEG and fMRI based on a bottom-up model: inferring activation and effective connectivity in neural masses. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 360(1457):1025–41, 5 2005.

[96] J. J. Riera, X. Wan, J. C. Jimenez, and R. Kawashima. Nonlinear local electrovascular coupling. I: A theoretical model. *Human brain mapping*, 27(11):896–914, 11 2006.

[97] J. J. Riera, J. C. Jimenez, X. Wan, R. Kawashima, and T. Ozaki. Nonlinear local electrovascular coupling. II: From data to neuronal masses. *Human brain mapping*, 28(4):335–54, 4 2007.

[98] R. C. Sotero, N. J. Trujillo-Barreto, Y. Iturria-Medina, F. Carbonell, and J. C. Jimenez. Realistically Coupled Neural Mass Models Can Generate EEG Rhythms. *Neural Computation*, 19(2):478–512, 2 2007.

[99] R. C. Sotero and N. J. Trujillo-Barreto. Biophysical model for integrating neuronal activity, EEG, fMRI and metabolism. *NeuroImage*, 39(1):290–309, 1 2008.

[100] P. A. Valdes-Sosa, J. M. Sanchez-Bornot, R. C. Sotero, Y. Iturria-Medina, Y. Aleman-Gomez, J. Bosch-Bayard, F. Carbonell, and T. Ozaki. Model driven EEG/fMRI fusion of brain oscillations. *Human brain mapping*, 30(9):2701–21, 9 2009.

[101] T. Deneux and O. Faugeras. EEG-fMRI fusion of paradigm-free activity using Kalman filtering. *Neural computation*, 22(4):906–48, 4 2010.

[102] I. Bojak, T. F. Oostendorp, A. T. Reid, and R. Kötter. Connecting mean field models of neural activity to EEG and fMRI data. *Brain topography*, 23(2):139–49, 6 2010.

[103] I. Bojak, T. F. Oostendorp, A. T. Reid, and R. Kötter. Towards a model-based integration of co-registered electroencephalography/functional magnetic resonance imaging data with realistic neural population meshes. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 369(1952):3785–801, 10 2011.

[104] D. Barwick. Hans Berger on the electroencephalogram of man. The fourteen original reports on the human electroencephalogramTranslated and edited by P. Gloor, xi + 350 pages, 175 illustrations, Elsevier, Amsterdam, 1969, Dfl 90.—. 13(4):507, 1 1971.

[105] E. D. Adrian and B. H. C. Matthews. The Berger Rhythm: potential changes from the occipital lobes in man. *Brain*, 57(4):355–385, 1934.

[106] L. R. Silva, Y. Amitai, and B. W. Connors. Intrinsic oscillations of neocortex generated by layer 5 pyramidal neurons. *Science (New York, N.Y.)*, 251(4992):432–5, 1 1991.

[107] R. R. Llinás. The intrinsic electrophysiological properties of mammalian neurons: insights into central nervous system function. *Science (New York, N.Y.)*, 242(4886):1654–64, 12 1988.

[108] S. W. Hughes and V. Crunelli. Thalamic mechanisms of EEG alpha rhythms and their pathological implications. *The Neuroscientist : a review journal bringing neurobiology, neurology and psychiatry*, 11(4):357–72, 8 2005.

[109] S. W. Hughes and V. Crunelli. Just a phase they're going through: the complex interaction of intrinsic high-threshold bursting and gap junctions in the

generation of thalamic alpha and theta rhythms. *International journal of psychophysiology : official journal of the International Organization of Psychophysiology*, 64(1):3–17, 4 2007.

[110] P. L. Nunez, B. M. Wingeier, and R. B. Silberstein. Spatial-temporal structures of human alpha rhythms: theory, microcurrent sources, multiscale measurements, and global binding of local networks. *Human brain mapping*, 13(3):125–64, 7 2001.

[111] C. Ciulla, T. Takeda, and H. Endo. MEG Characterization of Spontaneous Alpha Rhythm in the Human Brain. *Brain Topography*, 11(3):211–222, 2 1999.

[112] D. T. Liley, D. M. Alexander, J. J. Wright, and M. D. Aldous. Alpha rhythm emerges from large-scale networks of realistically coupled multicompartmental model cortical neurons. *Network (Bristol, England)*, 10(1):79–92, 2 1999.

[113] A. van Rotterdam, F. H. Lopes da Silva, J. van den Ende, M. A. Viergever, and A. J. Hermans. A model of the spatial-temporal characteristics of the alpha rhythm. *Bulletin of mathematical biology*, 44(2):283–305, 1 1982.

[114] P. Robinson, C. Rennie, and J. Wright. Propagation and stability of waves of electrical activity in the cerebral cortex. *Physical Review E*, 56(1):826–840, 7 1997.

[115] J. Wolfe, A. R. Houweling, and M. Brecht. Sparse and powerful cortical spikes. *Current opinion in neurobiology*, 20(3):306–12, 6 2010.

[116] K. W. Kaplan. H. Haken, Synergetics. An Introduction. Nonequilibrium Phase Transitions and Self-Organization in Physics, Chemistry, and Biology (2nd Edition). XI + 355 S., 152 Abb. Berlin-Heidelberg-New York 1978. Springer-Verlag. DM 66,00. *Zeitschrift für allgemeine Mikrobiologie*, 20(9):600–600, 1 2007.

[117] D. C. Van Essen. A Population-Average, Landmark- and Surface-based (PALS) atlas of human cerebral cortex. *NeuroImage*, 28(3):635–62, 11 2005.

[118] B. Pakkenberg and H. J. Gundersen. Neocortical neuron number in humans: effect of sex and age. *The Journal of comparative neurology*, 384(2):312–20, 7 1997.

[119] Y. Tang, J. R. Nyengaard, D. M. De Groot, and H. J. Gundersen. Total regional and global number of synapses in the human brain neocortex. *Synapse (New York, N.Y.)*, 41(3):258–73, 9 2001.

[120] S. Herculano-Houzel. The human brain in numbers: a linearly scaled-up primate brain. *Frontiers in human neuroscience*, 3:31, 1 2009.

[121] K. Fleischhauer, H. Petsche, and W. Wittkowski. Vertical bundles of dendrites in the neocortex. *Zeitschrift für Anatomie und Entwicklungsgeschichte*, 136(2):213–23, 1 1972.

[122] The Human Central Nervous System, 4th ed. *American Journal of Neuroradiology*, 29(5):e39–e39, 2 2008.

[123] V. B. Mountcastle. Modality and topographic properties of single neurons of cat's somatic sensory cortex. *Journal of neurophysiology*, 20(4):408–34, 7 1957.

[124] V. B. Mountcastle. The columnar organization of the neocortex. *Brain : a journal of neurology*, 120 (Pt 4:701–22, 4 1997.

[125] E. G. Jones, H. Burton, and R. Porter. Commissural and cortico-cortical "columns" in the somatic sensory cortex of primates. *Science (New York, N.Y.)*, 190(4214):572–4, 11 1975.

[126] J. Szentágothai. The modular architectonic principle of neural centers. *Reviews of physiology, biochemistry and pharmacology*, 98:11–61, 1 1983.

[127] P. S. Goldman and W. J. Nauta. Columnar distribution of cortico-cortical fibers in the frontal association, limbic, and motor cortex of the developing rhesus monkey. *Brain research*, 122(3):393–413, 2 1977.

[128] H. Markram. Fixing the location and dimensions of functional neocortical columns. *HFSP journal*, 2(3):132–5, 6 2008.

[129] Martini and F. E. Al. *Anatomy and Physiology' 2007 Ed.2007 Edition*. Rex Bookstore, Inc., 2007.

[130] H. Markram, M. Toledo-Rodriguez, Y. Wang, A. Gupta, G. Silberberg, and C. Wu. Interneurons of the neocortical inhibitory system. *Nature reviews. Neuroscience*, 5(10):793–807, 10 2004.

[131] V. Braitenberg and A. Schüz. *Cortex: Statistics and Geometry of Neuronal Connectivity*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998.

[132] F. Lopes da Silva. *EEG - fMRI*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[133] G. Buzsáki, C. A. Anastassiou, and C. Koch. The origin of extracellular fields and currents–EEG, ECoG, LFP and spikes. *Nature reviews. Neuroscience*, 13(6):407–20, 6 2012.

[134] N. Spruston. Pyramidal neurons: dendritic structure and synaptic integration. *Nature reviews. Neuroscience*, 9(3):206–21, 3 2008.

[135] W.-J. Gao and Z.-H. Zheng. Target-specific differences in somatodendritic morphology of layer V pyramidal neurons in rat motor cortex. *The Journal of comparative neurology*, 476(2):174–85, 8 2004.

[136] E. M. Kasper, A. U. Larkman, J. Lübke, and C. Blakemore. Pyramidal neurons in layer 5 of the rat visual cortex. I. Correlation among cell morphology, intrinsic electrophysiological properties, and axon targets. *The Journal of comparative neurology*, 339(4):459–74, 1 1994.

[137] B. W. Mel. Information Processing in Dendritic Trees. *Neural Computation*, 6(6):1031–1085, 11 1994.

[138] P. Bressloff and B. De Souza. Neural pattern formation in networks with dendritic structure. *Physica D: Nonlinear Phenomena*, 115(1-2):124–144, 4 1998.

[139] P. R. Coombes S., beim Graben P. *Tutorial on Neural Field Theory*. Berlin, Heidelberg, 2014.

[140] P. C. Bressloff. Spatiotemporal dynamics of continuum neural fields. *Journal of Physics A: Mathematical and Theoretical*, 45(3):033001, 1 2012.

[141] I Segev and John Rinzel and GM Shepherd, editor. *The theoretical foundation of dendritic function: Selected papers of Wilfrid Rall with commentaries*. MIT Press, 1994.

[142] L. J. Cauller and B. W. Connors. Functions of very distal dendrites: experimental and computational studies of layer 1 synapses on neocortical pyramidal cells. pages 199–229, 4 1992.

[143] L. J. Cauller and B. W. Connors. Synaptic physiology of horizontal afferents to layer I in slices of rat SI neocortex. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 14(2):751–62, 2 1994.

[144] M. E. Larkum, W. Senn, and H.-R. Lüscher. Top-down dendritic input increases the gain of layer 5 pyramidal neurons. *Cerebral cortex (New York, N.Y. : 1991)*, 14(10):1059–70, 10 2004.

[145] S. Gasparini, M. Migliore, and J. C. Magee. On the initiation and propagation of dendritic spikes in CA1 pyramidal neurons. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 24(49):11046–56, 12 2004.

[146] A. Losonczy and J. C. Magee. Integrative properties of radial oblique dendrites in hippocampal CA1 pyramidal neurons. *Neuron*, 50(2):291–307, 4 2006.

[147] Z. F. Mainen and T. J. Sejnowski. Influence of dendritic structure on firing pattern in model neocortical neurons. *Nature*, 382(6589):363–6, 7 1996.

[148] M. Egelhaaf, J. Haag, and A. Borst. Processing of synaptic information depends on the structure of the dendritic tree. *Neuroreport*, 6(1):205–8, 12 1994.

[149] G. Stuart, N. Spruston, B. Sakmann, and M. Häusser. Action potential initiation and backpropagation in neurons of the mammalian CNS. *Trends in neurosciences*, 20(3):125–31, 3 1997.

[150] G. Stuart and N. Spruston. Determinants of voltage attenuation in neocortical pyramidal neuron dendrites. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 18(10):3501–10, 5 1998.

[151] S. R. Williams and G. J. Stuart. Role of dendritic synapse location in the control of action potential output. *Trends in neurosciences*, 26(3):147–54, 3 2003.

[152] J. C. Magee and E. P. Cook. Somatic EPSP amplitude is independent of synapse location in hippocampal pyramidal neurons. *Nature neuroscience*, 3(9):895–903, 9 2000.

[153] D. A. Nicholson, R. Trana, Y. Katz, W. L. Kath, N. Spruston, and Y. Geinisman. Distance-dependent differences in synapse number and AMPA receptor expression in hippocampal CA1 pyramidal neurons. *Neuron*, 50(3):431–42, 5 2006.

[154] S. R. Williams and G. J. Stuart. Dependence of EPSP efficacy on synapse location in neocortical pyramidal neurons. *Science (New York, N.Y.)*, 295(5561):1907–10, 3 2002.

[155] S. Gasparini and J. C. Magee. State-dependent dendritic computation in hippocampal CA1 pyramidal neurons. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 26(7):2088–100, 2 2006.

[156] G. Ariav, A. Polsky, and J. Schiller. Submillisecond precision of the input-output transformation function mediated by fast sodium dendritic spikes in basal dendrites of CA1 pyramidal neurons. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 23(21):7750–8, 8 2003.

[157] G. Stuart, J. Schiller, and B. Sakmann. Action potential initiation and propagation in rat neocortical pyramidal neurons. *The Journal of physiology*, 505 (3):617–32, 12 1997.

[158] J. Schiller, Y. Schiller, G. Stuart, and B. Sakmann. Calcium action potentials restricted to distal apical dendrites of rat neocortical pyramidal neurons. *The Journal of physiology*, 505 (3):605–16, 12 1997.

[159] B. M. Kampa and G. J. Stuart. Calcium spikes in basal dendrites of layer 5 pyramidal neurons during action potential bursts. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 26(28):7424–32, 7 2006.

[160] M. E. Larkum, K. M. M. Kaiser, and B. Sakmann. Calcium electrogenesis in distal apical dendrites of layer 5 pyramidal cells at a critical frequency of back-propagating action potentials. *Proceedings of the National Academy of Sciences*, 96(25):14600–14604, 12 1999.

[161] G. J. Stuart and M. Häusser. Dendritic coincidence detection of EPSPs and action potentials. *Nature neuroscience*, 4(1):63–71, 1 2001.

[162] M. E. Larkum, J. J. Zhu, and B. Sakmann. A new cellular mechanism for coupling inputs arriving at different cortical layers. *Nature*, 398(6725):338–41, 3 1999.

[163] H. C. Tuckwell. Introduction to Theoretical Neurobiology: Volume 1, Linear Cable Theory and Dendritic Structure (Cambridge Studies in Mathematical Biology): 9780521022224: Medicine & Health Science Books @ Amazon.com.

[164] W. Rall. Theory of physiological properties of dendrites. *Annals of the New York Academy of Sciences*, 96:1071–92, 3 1962.

[165] J. C. Magee and D. Johnston. A synaptically controlled, associative signal for Hebbian plasticity in hippocampal neurons. *Science (New York, N.Y.)*, 275(5297):209–13, 1 1997.

[166] S. Watanabe, D. A. Hoffman, M. Migliore, and D. Johnston. Dendritic K+ channels contribute to spike-timing dependent long-term potentiation in hippocampal pyramidal neurons. *Proceedings of the National Academy of Sciences of the United States of America*, 99(12):8366–71, 6 2002.

[167] B. M. Kampa, J. J. Letzkus, and G. J. Stuart. Requirement of dendritic calcium spikes for induction of spike-timing-dependent synaptic plasticity. *The Journal of physiology*, 574(Pt 1):283–90, 7 2006.

[168] F. G. Pike, R. M. Meredith, A. W. Olding, and O. Paulsen. Rapid report: postsynaptic bursting is essential for 'Hebbian' induction of associative long-term potentiation at excitatory synapses in rat hippocampus. *The Journal of physiology*, 518 (Pt 2:571–6, 7 1999.

[169] K. M. Stiefel, F. Tennigkeit, and W. Singer. Synaptic plasticity in the absence of backpropagating spikes of layer II inputs to layer V pyramidal cells in rat visual cortex. *The European journal of neuroscience*, 21(9):2605–10, 5 2005.

[170] S. Remy and N. Spruston. Dendritic spikes induce single-burst long-term potentiation. *Proceedings of the National Academy of Sciences of the United States of America*, 104(43):17192–7, 10 2007.

[171] N. L. Golding, N. P. Staff, and N. Spruston. Dendritic spikes as a mechanism for cooperative long-term potentiation. *Nature*, 418(6895):326–31, 7 2002.

[172] J. J. Letzkus, B. M. Kampa, and G. J. Stuart. Learning rules for spike timing-dependent plasticity depend on dendritic synapse location. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 26(41):10420–9, 10 2006.

[173] P. J. Sjöström and M. Häusser. A cooperative switch determines the sign of synaptic plasticity in distal dendrites of neocortical pyramidal neurons. *Neuron*, 51(2):227–38, 7 2006.

[174] B. M. Kampa, J. J. Letzkus, and G. J. Stuart. Dendritic mechanisms controlling spike-timing-dependent synaptic plasticity. *Trends in neurosciences*, 30(9):456–63, 9 2007.

[175] D. Johnston, J. C. Magee, C. M. Colbert, and B. R. Cristie. Active properties of neuronal dendrites. *Annual review of neuroscience*, 19:165–86, 1 1996.

[176] M. Häusser, N. Spruston, and G. J. Stuart. Diversity and dynamics of dendritic signaling. *Science (New York, N.Y.)*, 290(5492):739–44, 10 2000.

[177] G. G. Turrigiano. Homeostatic plasticity in neuronal networks: the more things change, the more they stay the same. *Trends in neurosciences*, 22(5):221–7, 5 1999.

[178] K. A. Atkinson. *An Introduction to Numerical Analysis*. 1989.

[179] A. Naka and H. Adesnik. Inhibitory Circuits in Cortical Layer 5. *Frontiers in Neural Circuits*, 10:35, 2016.

[180] I. Bojak and D. T. J. Liley. Axonal Velocity Distributions in Neural Field Equations. *PLoS Computational Biology*, 6(1):e1000653, 1 2010.

[181] Gordon M. Shepherd. *The Synaptic Organization of the Brain*. Oxford University Press, 2004.

[182] Rodney J. Douglas and Kevan A. C. Martin and David Whitteridge. A Canonical Microcircuit for Neocortex. *Neural Computation*, 1:480–488, 1989.

[183] T. McKenna, J. Davis, and S. F. Zornetzer, editors. *Single Neuron Computation*. Academic Press Professional, Inc., San Diego, CA, USA, 1992.

[184] P. C. Bressloff. New Mechanism for Neural Pattern Formation. *Phys. Rev. Lett.*, 76(24):4644–4647, 6 1996.

[185] W. Rall. Theory of physiological properties of dendrites. *Annals of the New York Academy of Sciences*, 96:1071–1092, 3 1962.

[186] R. Terjung, editor. *Comprehensive Physiology*. John Wiley & Sons, Inc., Hoboken, NJ, USA, 1 2011.

[187] W. R. Holmes. Equivalent Cylinder Model (Rall). In D. Jaeger and R. Jung, editors, *Encyclopedia of Computational Neuroscience*, pages 1–3. Springer New York, New York, NY, 2013.

[188] W. Rall. Electrophysiology of a Dendritic Neuron Model. *Biophysical Journal*, 2(2, Part 2):145–167, 1962.

[189] P. Fatt and B. Katz. The effect of inhibitory nerve impulses on a crustacean muscle fibre. *The Journal of Physiology*, 121(2):374–389, 8 1953.

[190] J. S. Coombs, J. C. Eccles, and P. Fatt. The inhibitory suppression of reflex discharges from motoneurones. *The Journal of Physiology*, 130(2):396–413, 11 1955.

[191] A. L. Hodgkin and B. Katz. The effect of sodium ions on the electrical activity of the giant axon of the squid. *The Journal of Physiology*, 108(1):37–77, 3 1949.

[192] P. Fatt and B. Katz. An analysis of the end-plate potential recorded with an intra-cellular electrode. *The Journal of Physiology*, 115(3):320–370, 11 1951.

[193] A. L. Hodgkin and A. F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, 117(4):500–544, 8 1952.

[194] S. S. Stefanou, G. Kastellakis, and P. Poirazi. Creating and Constraining Compartmental Models of Neurons Using Experimental Data. In A. Korngreen, editor, *Advanced Patch-Clamp Analysis for Neuroscientists*, pages 325–343. Springer New York, New York, NY, 2016.

[195] H. A. Lloyd, R. W. A. Hugh, and A. E. Douglas. The electrical constants of a crustacean nerve fibre. *Proceedings of the Royal Society of London. Series B - Biological Sciences*, 133(873):444–479, 12 1946.

[196] L. Davis and R. Lorente de No. Contribution to the mathematical theory of the electrotonus. *Studies from the Rockefeller institute for medical research. Reprints. Rockefeller Institute for Medical Research*, 131:442–496, 1947.

[197] W. R. Holmes. Cable Equation. In D. Jaeger and R. Jung, editors, *Encyclopedia of Computational Neuroscience*, pages 1–13. Springer New York, New York, NY, 2013.

[198] Ernst Niebur. Neuronal cable theory. *Scholarpedia*, 3:2674, 2008.

[199] Z. S. e. a. Lodish H, Berk A. *Molecular Cell Biology. Section 21.4, Neurotransmitters, Synapses, and Impulse Transmission.* 4th edition, 2000.

[200] S. Kim. Action potential modulation in CA1 pyramidal neuron axons facilitates OLM interneuron activation in recurrent inhibitory microcircuits of rat hippocampus. *PloS one*, 9(11):e113124–e113124, 11 2014.

[201] K. R. Michalikova M, Remme MW. Spikelets in Pyramidal Neurons: Action Potentials Initiated in the Axon Initial Segment That Do Not Activate the Soma. *PLoS Comput Biol*, 2017.

[202] H. Meffin and T. Kameneva. The electrotonic length constant: A theoretical estimate for neuroprosthetic electrical stimulation. *Biomedical Signal Processing and Control*, 6(2):105–111, 2011.

[203] D. R. Merrill, M. Bikson, and J. G. R. Jefferys. Electrical stimulation of excitable tissue: design of efficacious and safe protocols. *Journal of Neuroscience Methods*, 141(2):171–198, 2005.

[204] B. Coburn. Neural modeling in electrical stimulation. *Critical reviews in biomedical engineering*, 17(2):133–178, 1989.

[205] D. Kabaso, P. J. Coskren, B. I. Henry, P. R. Hof, and S. L. Wearne. The electrotonic structure of pyramidal neurons contributing to prefrontal cortical circuits in macaque monkeys is significantly altered in aging. *Cerebral cortex (New York, N.Y. : 1991)*, 19(10):2248–2268, 10 2009.

[206] W. R. Holmes. Electrotonic Length, Formulas and Estimates. In D. Jaeger and R. Jung, editors, *Encyclopedia of Computational Neuroscience*, pages 1–3. Springer New York, New York, NY, 2013.

[207] W. Rall. Time Constants and Electrotonic Length of Membrane Cylinders and Neurons. *Biophysical Journal*, 9(12):1483–1508, 1969.

[208] W. Rall. *Core Conductor Theory and Cable Properties of Neurons.*, volume 1. 1977.

[209] T. H. Brown, R. A. Fricke, and D. H. Perkel. Passive electrical constants in three classes of hippocampal neurons. *Journal of Neurophysiology*, 46(4):812–827, 10 1981.

[210] W. R. Holmes, I. Segev, and W. Rall. Interpretation of time constant and electrotonic length estimates in multicylinder or branched neuronal structures. *Journal of Neurophysiology*, 68(4):1401–1420, 10 1992.

[211] W. R. Holmes and W. Rall. Electrotonic length estimates in neurons with dendritic tapering or somatic shunt. *Journal of Neurophysiology*, 68(4):1421–1437, 10 1992.

[212] G. Major, J. D. Evans, and J. J. Jack. Solutions for transients in arbitrarily branching cables: II. Voltage clamp theory. *Biophysical journal*, 65(1):450–468, 7 1993.

[213] G. Major, J. D. Evans, and J. J. Jack. Solutions for transients in arbitrarily branching cables: I. Voltage recording with a somatic shunt. *Biophysical journal*, 65(1):423–449, 7 1993.

[214] P. Welch. The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, 15(2):70–73, 1967.

[215] R. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. In *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, pages 39–43, 1995.

[216] M. R. AlRashidi and M. E. El-Hawary. A Survey of Particle Swarm Optimization Applications in Electric Power Systems. *IEEE Transactions on Evolutionary Computation*, 13(4):913–918, 2009.

[217] M. Omran. *Particle swarm optimization methods for pattern recognition and image processing* . 1 2005.

[218] A. D. Rosli, N. S. Adenan, H. Hashim, N. E. Abdullah, S. Sulaiman, and R. Baharudin. Application of Particle Swarm Optimization Algorithm for Optimizing ANN Model in Recognizing Ripeness of Citrus. *IOP Conference Series: Materials Science and Engineering*, 340:012015, 2018.

[219] D. Singh, S. Prasad, and S. Srivastava. Implementation of artificial intelligence cognitive neuroscience neuron cell using adaptive velocity threshold particle swarm optimization (AVT-PSO) on FPGA. In *2017 6th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, pages 548–552, 2017.

[220] S. Poungponsri and X. Yu. Electrocardiogram (ECG) signal modeling and noise reduction using wavelet neural networks. In *2009 IEEE International Conference on Automation and Logistics*, pages 394–398, 2009.

[221] B. A. Garro and R. Vázquez. *Designing Artificial Neural Networks Using Particle Swarm Optimization Algorithms*, volume 2015. 6 2015.

[222] N. P. Suraweera and D. N. Ranasinghe. A Natural Algorithmic Approach to the Structural Optimisation of Neural Networks. In *2008 4th International Conference on Information and Automation for Sustainability*, pages 150–156, 2008.

[223] S. de-los-cobos silva, A. Terceño, M. Angel Gutiérrez-Andrade, E. A. Rincón-García, P. Lara-Velázquez, and M. Aguilar-Cornejo. *Particle swarm optimization: An alternative for parameter estimation in regression*, volume 18. 1 2013.

[224] A. Alfi and H. Modares. *System identification and control using adaptive particle swarm optimization*, volume 35. 3 2011.

[225] R.-J. Ma, N.-Y. Yu, and J.-Y. Hu. *Application of Particle Swarm Optimization Algorithm in the Heating System Planning Problem*, volume 2013. 7 2013.

[226] M. Hajihassani, D. Jahed Armaghani, and R. Kalatehjari. *Applications of Particle Swarm Optimization in Geotechnical Engineering: A Comprehensive Review*. 9 2017.

[227] Z. Wu, Z. Ni, L. Gu, and X. Liu. *A Revised Discrete Particle Swarm Optimization for Cloud Workflow Scheduling*. 12 2010.

[228] Riccardo Poli and James Kennedy and Tim Blackwell. *Particle swarm optimization – An Overview*. 2007.

[229] Riccardo Poli. Analysis of the Publications on the Applications of Particle Swarm Optimisation. 2008.

[230] R. C. Eberhart and Y. Shi. Comparing inertia weights and constriction factors in particle swarm optimization. In *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512)*, volume 1, pages 84–88, 2000.

[231] Eberhart and Y. Shi. Particle swarm optimization: developments, applications and resources. In *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*, volume 1, pages 81–86, 2001.

[232] S. W. Yudong Zhang and G. Ji. A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications. 2015.

[233] A. Banks, J. Vincent, and C. Anyakoha. *A review of particle swarm optimization. Part II: hybridisation, combinatorial, multi-criteria and constrained optimization, and indicative application*, volume 7. 3 2008.

[234] Y. Shi and R. Eberhart. A modified particle swarm optimizer. In *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*, pages 69–73, 1998.

[235] M. Clerc and J. Kennedy. The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6(1):58–73, 2002.

[236] D. E. Gbenga and E. I. Ramlan. Understanding the Limitations of Particle Swarm Algorithm for Dynamic Optimization Tasks: A Survey Towards the Singularity of PSO for Swarm Robotic Applications. *ACM Comput. Surv.*, 49(1):8:1–8:25, 7 2016.

[237] D. Bratton and J. Kennedy. Defining a Standard for Particle Swarm Optimization. In *2007 IEEE Swarm Intelligence Symposium*, pages 120–127, 2007.

[238] M. Oliveira, D. Pinheiro, B. Andrade, C. Bastos-Filho, and R. Menezes. Communication Diversity in Particle Swarm Optimizers. In M. Dorigo, M. Birattari, X. Li, M. López-Ibáñez, K. Ohkura, C. Pinciroli, and T. Stützle, editors, *Swarm Intelligence*, pages 77–88, Cham, 2016. Springer International Publishing.

[239] J. Kennedy. Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance. In *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, volume 3, pages 1931–1938, 1999.

[240] E. Safavieh, A. Gheibi, M. Abolghasemi, and A. Mohades. *Particle swarm optimization with Voronoi neighborhood*. 11 2009.

[241] J. Kennedy. The particle swarm: social adaptation of knowledge. In *Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC '97)*, pages 303–308, 1997.

[242] P. N. Suganthan. Particle swarm optimiser with neighbourhood operator. In *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, volume 3, pages 1958–1962, 1999.

[243] A. D. Bird and H. Cuntz. Optimal Current Transfer in Dendrites. *PLOS Computational Biology*, 12(5):e1004897–, 5 2016.

[244] H. Cuntz, A. Borst, and I. Segev. Optimization principles of dendritic structure. *Theoretical biology & medical modelling*, 4:21, 2007.

[245] M. H. P. Kole, S. Hallermann, and G. J. Stuart. Single I_h Channels in Pyramidal Neuron Dendrites: Properties, Distribution, and Impact on Action Potential Output. *The Journal of Neuroscience*, 26(6):1677, 8 2006.

[246] M. Wolff, W. Vogel, and B. V. Safronov. Uneven distribution of K+ channels in soma, axon and dendrites of rat spinal neurones: functional role of the soma in generation of action potentials. *The Journal of physiology*, 509 (Pt 3)(Pt 3):767–776, 6 1998.

[247] S. I. Motipally, K. M. Allen, D. K. Williamson, and G. Marsat. Differences in sodium channel densities in the apical dendrites of pyramidal cells of the electrosensory lateral line lobe. *bioRxiv*, page 592758, 1 2019.

[248] D.T.J. Liley and I. Bojak and M.P. Dafilis and L. van Veen and F. Frascoli and B.L. Foster. *Bifurcations and state changes in the human alpha rhythm: Theory and experiment.* Springer, New York, 2009.

[249] J. J. Jack, S. J. Redman, and K. Wong. The components of synaptic potentials evoked in cat spinal motoneurones by impulses in single group Ia afferents. *The Journal of Physiology*, 321(1):65–96, 12 1981.

[250] M. Migliore, M. Ferrante, and G. A. Ascoli. Signal Propagation in Oblique Dendrites of CA1 Pyramidal Cells. *Journal of Neurophysiology*, 94(6):4145–4155, 12 2005.

[251] C. C. Rumsey and L. F. Abbott. Equalization of Synaptic Efficacy by Activity- and Timing-Dependent Synaptic Plasticity. *Journal of Neurophysiology*, 91(5):2273–2280, 5 2004.

[252] C. C. Rumsey and L. F. Abbott. Synaptic Democracy in Active Dendrites. *Journal of Neurophysiology*, 96(5):2307–2318, 11 2006.

[253] David T. J. Liley and Peter J. Cadusch and James J. Wright. A continuum theory of electro-cortical activity. *Neurocomputing*, 26-27:795–800, 1999.