Norwegian University of Life Sciences
Faculty of Environmental Science and Technology
Department of Mathematical Sciences and Technology

Master Thesis 2015
60 credits

Biophysical Modeling of EEG Signals from Neurons in the Brain

Biofysisk modellering av EEG-signaler fra nerveceller i hjernen

Solveig Næss

Acknowledgements

This thesis is the fulfillment of my Master's degree at the Norwegian University of Life Sciences.

First of all, I would like to thank my inspiring supervisors, Prof. Gaute T. Einevoll and Dr. Torbjørn V. Ness. A special thanks to Gaute, for introducing me to the field of neuroscience, for sharing his enthusiasm and knowledge, and also for including me in the computational neuroscience community. My gratitude also goes to Torbjørn for being so generous with his time, for sharing his many ideas and for his patience teaching me programming. A big thanks also to Espen Hagen for answering my endless stream of emails about `LFPy`, and to Geir Halnes for good advice on `NEURON` issues.

Thanks also to Eirik, Nora, Karen Marie, my parents and my sister for all your support and useful comments on the manuscript. Lastly, the late hours in TF211 would never have been as enjoyable without my fellow students. You are awesome!

Ås, May 15th, 2015

Solveig Næss

Abstract

In order to understand the link between brain signal recordings, such as electrocorticography (ECoG) and electroencephalography (EEG), and the underlying neural activity, neuroinformatics tools play an important role. A great example of such a tool is the open-source `Python` package, `LFPy`, which can be used for numerical calculations of extracellular potentials, based on a well-established compartment-based forward-modeling scheme. In this project, detailed biophysical modeling was used to gain a better understanding of contributions from single neurons to measurable extracellular potentials. In particular, we addressed the following questions: *How do single neurons contribute to ECoG and EEG signals? And can these signals be modeled with the current dipole approximation?* `Python` tools for calculating neural axial currents and the current dipole moment of a neuron were developed, and further built on to calculate extracellular potentials from the current dipole approximation. These methods, in addition to the above-mentioned compartment-based forward model, were used for studying extracellular potentials from single-cell simulations. The two different models give similar results for computations of extracellular recordings from virtual electrodes placed several millimeters away from the neuron source. Thus, the dipole approximation cannot be used for predicting single-cell contributions to ECoG signals, since ECoG recordings are measured only some hundred micrometers away from the neuron. For modeling of single-cell EEG contributions, on the other hand, the current dipole approximation appears to be applicable.

Sammendrag

For å forstå sammenhengen mellom elektrisk aktivitet i nervecellene i hjernen og målinger som elektrokortikografi (ECoG) og elektroencefalogram (EEG), er informatikkverktøy en viktig bidragsyter. `LFPy` er et godt eksempel på et slikt verktøy, utviklet som en tilleggspakke i `Python` med åpen kildekode. Blant annet kan `LFPy` brukes til å regne ut elektriske potensialer med utgangspunkt i en veletablert metode for compartment-basert direkte modellering. I denne oppgaven blir detaljert biofysisk modellering av enkeltnerveceller brukt til å undersøke hvordan disse bidrar til målbare ekstracellulære potensialer. Særlig belyses følgende spørsmål: *På hvilken måte bidrar enkeltnerveceller til ECoG- og EEG-signaler? Og kan disse signalene modelleres ved hjelp av strømdipolmoment-tilnærmingen?* `Python`-verktøy for utregning av nevrale aksial-strømmer og strømdipolmoment fra nerveceller ble utviklet og videre brukt til å implementere dipol-basert direkte modellering i `Python`. Dette kan, sammen med den ovennevnte compartment-baserte modelleringsmetoden, brukes til å undersøke bidraget til ECoG- og EEG-signaler fra enkelthjerneceller. Resultatene fra de to forskjellige modellene viste seg å være like når målepunktene ble plassert flere millimeter unna nervecellen. Følgelig kan ikke strømdipolmoment brukes til å modellere enkeltcellebidrag til ECoG-signaler, siden nervecellen her ligger kun noen hundre mikrometer unna målepunktet. Bidrag til EEG-signaler kan derimot i stor grad forutsies ved hjelp av strømdipol-tilnærmingen.

Contents

Chapter 1

Introduction

Humans have been trying to figure out the underlying secrets of their own intelligence since ancient times [18]. Using the mind to understand how the brain forms thoughts, must be the ultimate puzzle and goes as far back as to the ancient Greece: Hippocrates, the father of western medicine, was one of the first to state that thoughts take place in the brain [8], and not in the heart, as was common belief. Lately, humans have been able to actually measure brain activity, and Richard Caton recorded electric signals from the top of animal brains already in 1875 [13]. Fifty years later, Hans Berger did the first electroencephalography (EEG) recording, i.e., he measured electric potentials on top of the scalp [9].

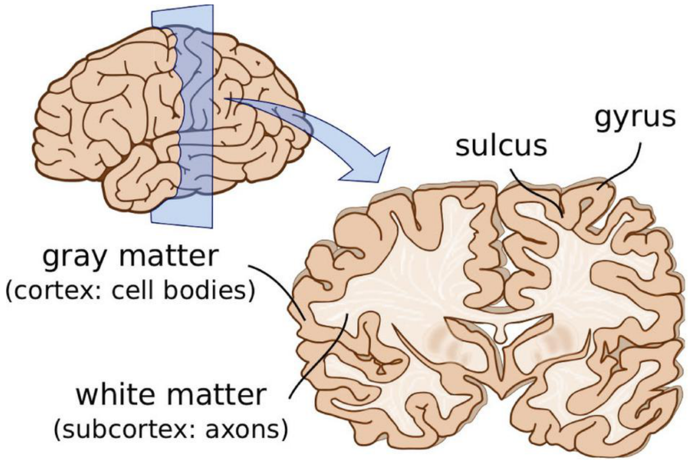

Figure 1.1: Schematic illustration of the human brain. The outer colored layer in the vertical cross section is the neocortex, or gray matter, containing $\sim 10^{10}$ neurons [32]. The folds in cortex are called sulci, and the regions between adjacent folds are called gyri. The subcortical layer, known as white matter, mainly consists of axons, i.e., cables connecting the neurons together. The figure is taken from Budday et al., 2014 [10].

The outer surface of the brain consists of a layer called neocortex, also known as gray matter. This bulky, folded structure covers the mammalian brain and is home to billions of brain cells [32], see Figure 1.1. The two main cell types are neuroglia and neurons. While glial cells are considered the support staff of the brain [42, 19], the neurons take care of information processing and storage [19, 22, 28].

Electric potentials in the brain arise from currents flowing in and out of neurons in neocortex. These potentials can be measured by placing electrodes on the top of cortex, ECoG (electrocorticography), on top of the skull, EEG (electroencephalography) or with MEG (magnetoencephalography), see explanation in Figure 1.2. What we know about neural activity today, however, is mainly results of intracranial recordings, i.e., from inserting electrodes into brain tissue [19, 32]. Such recordings are, for ethical reasons, not performed on humans, except in special clinical cases, as for example drug-resistant epilepsy or small brain tumors. Intracranial recordings from animals is an important contribution to brain research. Even though useful information can be extracted from animal studies, however, a model of the human brain can never be based solely on animal data [19]. Extracranial recording methods, like EEG and MEG, are hence of great importance because of their non-invasive nature.

EEG is also one of the simplest ways of measuring brain activity and is therefore widely used in psychology. The typical way of analyzing data from EEG recordings has been through looking for correlations between stimulus and measurements, or by comparing results from EEG recordings to other measures of brain activity. Lately, the importance of measurement physics has been emphasized, i.e., developing mathematical models for biological systems to get a better understanding of the underlying physics of what we measure [16, 26]. An example of this is a biophysical *forward-modeling scheme* which can be used to model both the underlying high-frequency part ($\gtrsim$ 500 Hz), known as *multi-unit activity* (MUA), and the low-frequency part, *local field potentials* (LFP), ($\lesssim$ 500 Hz) of electric potentials from neurons in the brain [38, 21, 25, 35, 36].

Multi-unit activity arises from neural spiking, and partly as a consequence of dendritic filtering, MUA signals decay rapidly with distance from the neural source. Because of this, MUAs can hold information about the spiking activity of a single neuron [35, 25]. LFPs, on the other hand, are presumably caused by synaptic activity in the neural dendrites and can travel farther than the high-frequency parts. Consequently, when recording LFP signals, the electrode will pick up contributions from whole populations of neurons. An advantage of this is that LFP recordings can be used for studying population activity [19, 32, 35, 26]. Compared to recording of spiking activity, LFP recordings are more stable and have therefore been suggested as a candidate for brain-computer interfaces, such as prosthetic devices [16].

Since EEG recordings register potentials relatively far from the neuron source,

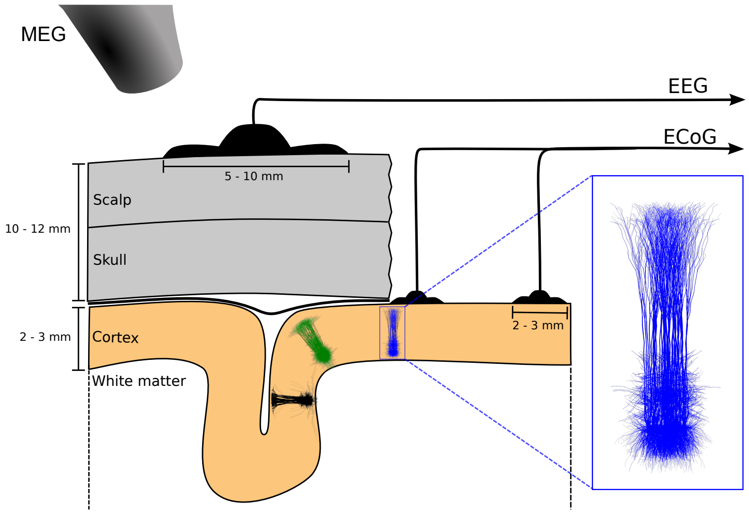

Figure 1.2: Extracellular potential recording methods. Illustration of the biophysics of ECoG, EEG and MEG recordings of electric and magnetic signals from neuron populations in neocortex. The strength and correlation of neural activity affect the measurable signals. Additionally, the recording position relative to the neuron is important in several ways: The farther away from the neural source, the weaker the measured signal. The angular orientation does, however, also affect the signal. For ECoG and EEG signals, properties of the brain components between the electrode and the source, such as conductivity, have a big impact on the signal strength. Courtesy of Torbjørn V. Ness.

such recordings mainly contain low-frequency potentials from populations of neurons (but see also [43]). Even if intracranial recordings can give more detailed information about single neurons, EEG signals can reveal new insight about neural networks and thereby the mechanisms controlling higher brain functions [19, 32].

In order to make comparisons between neuron models and measurements easier, it is important to develop new informatics tools. A great example of this is the open source `Python` package, `LFPy` [26, 3]. Among other things, this package can be used for calculating extracellular potentials numerically, based on the above-mentioned forward-modeling scheme.

In this project, detailed biophysical modeling was used to gain a better understanding of contributions from single neurons to measurable extracellular potentials. `Python` [6, 23] tools for calculating neural axial currents, the current dipole moment of a neuron and extracellular potentials based on the current dipole approximation, were developed. These methods, in addition to the well-established forward-modeling scheme, were thereafter used for studying extracellular potential contributions from single neurons. In particular, we wanted to look at single cell contributions to ECoG and EEG signals, and investigate whether these could be predicted by the current dipole approximation.

The following chapter gives an introduction to the electrical activity in neurons, before describing the theory behind models for computing extracellular potentials from single neurons. Chapter 3 gives an overview over the neuroinformatics tools used in the project, and a thorough explanation of how a `Python` tool based on the current dipole approximation was developed. The results of neuron simulations and virtual recordings are presented in Chapter 4 and discussed further in Chapter 5.

Chapter 2

Theory

This chapter explains the theoretical background of the electrical activity in neurons, before describing two biophysical models for calculations of extracellular potentials: the compartment-based and the dipole-based forward models.

2.1 Neural Electrical Activity

Neurons come in various shapes and sizes, and can be classified based on their morphology, like the star-shaped stellate cells and the pyramidal cells [8]. The *soma* is the neuron's cell body, see Figure 2.1, and contains the same organelles as other mammalian cells [30].

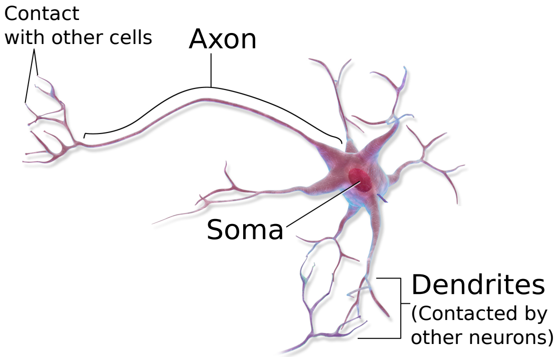

Figure 2.1: Simple illustration of neuron morphology. Synapses on the dendrites and the soma, receive input currents from other cells. If the soma membrane potential reaches a threshold value, an action potential will be fired, and a large fluctuation in membrane potential will be transported down the axon and submit an electric pulse to other neurons. Modified from Blausen.com staff. "Blausen gallery 2014". Wikiversity Journal of Medicine. DOI:10.15347/wjm/2014.010. ISSN 20018762

What mainly distinguishes the neuron from other cell types, are the *dendrites* and the *axon*, responsible for making the neurons capable of communicating with each other [8]. Dendrites are numerous, branchlike structures, that can receive inputs from other neurons, while the axon is a long cable passing the output on to other cells [14]. Neurons are hence capable of sending and receiving information, a characteristic arising from the electrical properties of the cell membrane.

The cell membrane is a five-nanometer thick lipid bilayer, effectively separating the cerebrospinal fluid surrounding the neurons from the cytoplasm inside the cells, see Figure 2.2 [40]. These fluids contain various types of ions, such as sodium (Na^+), potassium (K^+) and chloride (Cl^-), of which the carried charges are almost balanced on each side of the membrane. Because of the differing ion concentrations in the intracellular and extracellular media, ions want to diffuse down the concentration gradients, but are stopped by the membrane. Embedded in the membrane, however, are *ion channels*; pore-like protein structures letting certain ions through the membrane. Ion channels are categorized as either passive or active, meaning their conductance can be either static or dependent on membrane potential and other environmental variables [40].

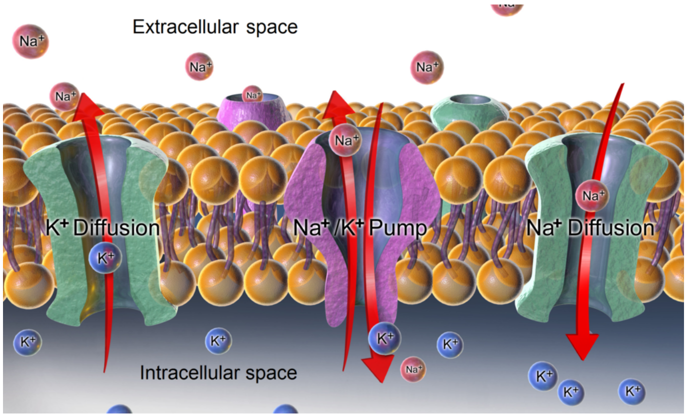

Figure 2.2: Illustration of the cell membrane. The cell membrane consists of a lipid bilayer, impermeable to water and ions, parting the extracellular cerebrospinal fluid from the intracellular cytoplasm. The green tunnels on the figure represent ion channels, letting certain types of ions through. The ion pumps, purple structures on the figure, set up ion concentration differences: there is typically much more sodium (Na^+) on the outside than on the inside of the cell membrane, and the opposite for potassium (K^+). Ions will diffuse through the ion channels in the direction of the concentration gradients, charging the membrane that acts as a parallel plate capacitor. The resulting electric potential over the cell membrane is called the membrane potential, and has a typical equilibrium value of -65 mV. The figure is taken from: The Blausen.com staff. "Blausen gallery 2014". Wikiversity Journal of Medicine. DOI:10.15347/wjm/2014.010. ISSN 20018762.

Due to the existence of ion channels, some ions can diffuse through the membrane and disturb the charge balance. There will typically be an excess of freely moving positive charges on the outside of the cell membrane attracting intracellular negative ions, and making the cell membrane act as a parallel plate capacitor. The ions will typically set up an electric potential difference over the cell membrane, so that the ionic currents due to diffusion and electrical drift will balance each other out. The typical value of this equilibrium membrane potential is often referred to as the resting potential and is about -65 mV [8, 40].

An axon splits into many branches to form thousands of connections, typically with the soma or dendrites of other neurons [14]. Consequently, the neurons in the brain are coupled together in highly complex networks, meaning neurons can be connected to their neighbors as well as cells on the opposite side of the brain [8]. These connections are called *synapses*, and the two neurons connected by a synapse are referred to as *presynaptic*, before the synapse, and *postsynaptic*, after the synapse. Synapses can be either electrical or chemical, their function, however, is to make currents flow into or out of the postsynaptic neuron. Chemical synapses do by far outnumber the electrical, and neurons can be classified based on which type of chemical synapses they hold. *Excitatory* neurons have chemical synapses, all of which contribute to opening of ion channels and an increase in membrane potential on the postsynaptic neuron. The chemical synapses of the *inhibitory* neurons, on the other hand, have the opposite effect, i.e., decreasing the postsynaptic neuron's membrane potential. If the membrane potential of a neuron exceeds a threshold, the neuron will fire an *action potential*, often referred to as a *spike*. A regenerative fluctuation in membrane potential will then propagate to the end of the neuron's axon, activating the synapses. After this flush of ions through the ion channels, membrane protein structures, called *ion pumps*, will pump ions against their concentration gradients to restore the membrane resting potential [14, 8, 40].

Ionic and capacitive currents are also known by the collective term *transmembrane currents*, i.e., currents crossing the cell membrane. In addition to affecting the membrane potential, the transmembrane currents generate an electric field in the extracellular medium. The *extracellular potential* can be measured relative to a reference electrode far away, by an electrode inserted into cortex, ECoG recordings or EEG recordings [32]. In the following two sections, the theory behind two ways of estimating extracellular potentials is explained, the compartment-based (CB) forward model and the dipole-based (DB) forward model. The term *forward* implies that the model is based on underlying neural dynamics and predicts what we can measure, as opposed to *inverse modeling*, predicting the neural dynamics from measurements.

2.2 Compartment-Based (CB) Forward Model

The compartment based forward model (CB model) is a well-founded biophysical two-step modeling scheme for precise calculations of extracellular potentials. The first step involves multi-compartmental modeling and incorporates the details of reconstructed neuron morphologies to calculate transmembrane currents. In the second step, volume conductor theory is used for calculating the contributions to extracellular potentials from the transmembrane currents found in step one.

2.2.1 Multi-Compartmental Modeling

When using multi-compartmental modeling to calculate transmembrane currents, neurons are first of all split into multiple cylindrical compartments, as illustrated in Figure 2.3.

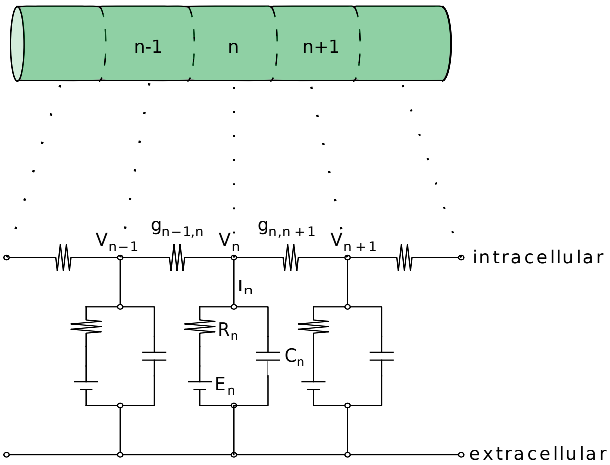

Figure 2.3: Multi-compartmental modeling illustrated by part of dendritic stick. A part of a neuron is split into cylindrical compartments, small enough to assume the membrane potential to be the same throughout each compartment. Each compartment is modeled as an electrical circuit, with membrane potential V_n, membrane resistance R_n, membrane capacitance C_n and equilibrium potential E_n. $g_{n,n+1}$ is the axial, intracellular conductance between compartments.

Since the cell membrane acts like both a resistor and a capacitor, each neuron compartment can be modeled as an electric RC-circuit driven by the voltage difference between the compartment's equilibrium potential and the actual membrane potential. The compartment size is chosen so small that the membrane potential can be assumed to be the same throughout each compartment. Because of its shape, the soma is conventionally modeled a single, isopotential compartment. Applying Kirchhoff's current law with respect to axial and transmembrane currents flowing into or out of one compartment, the following equation can be derived [14, 40]:

$$g_{n,n+1}(V_{n+1} - V_n) - g_{n-1,n}(V_n - V_{n-1}) = C_n \frac{dV_n}{dt} + \sum_j I_n^j. \tag{2.1}$$

Here, $g_{n,n+1}$ is the conductance between compartment n and compartment $n + 1$, i.e., the inversely proportional of the axial resistance. The membrane potential is denoted by V_n, and C_n is the membrane capacitance of compartment n. The left hand side represents the current flowing into or out of the compartment from its neighbor compartments, which by Kirchhoff's current law equals the current entering and escaping through the membrane. The first term on the right hand side is the capacitive current, whereas $\sum_j I_n^j$ is the sum of currents due to ions crossing the cell membrane through j types of ion channels. The right hand side of Equation (2.1) equals the total transmembrane current of compartment n, I_n [40, 26]:

$$I_n = C_n \frac{dV_n}{dt} + \sum_j I_n^j. \tag{2.2}$$

2.2.2 Extracellular Potentials from Volume Conduction

Thanks to volume conductor theory, it is possible to calculate the contribution from each transmembrane current in the vicinity of a virtual electrode, to the extracellular potential measured at the electrode location [19, 32, 15, 37]. To apply volume conductor theory, brain tissue should be envisioned as a smooth, three-dimensional volume conductor. Furthermore, transmembrane currents are to be understood as volume current sinks and sources [32, 26]. A *current source* means a current entering the extracellular medium. A current flowing into a cell, escaping the extracellular fluid, is similarly seen as a *current sink*.

Moreover, the CB model is based on several assumptions, the first being the quasistatic approximation of Maxwell's equations. This implies that the magnetic and electric fields effectively decouple, an assumption that seems to be well justified, [19, 37]. Secondly, the extracellular medium is assumed to be linear, ohmic, isotropic, homogeneous and frequency independent [32, 19]. Consequently, the relationship between current density, $\mathbf{j}$, and the electric field, $\mathbf{E}$, is linear, i.e., $\mathbf{j} = \sigma\mathbf{E}$, where the extracellular conductivity, σ, is a real scalar. The assumption that the conductivity is real reflects the negligibility of the capacitive properties of the extracellular medium, which consequently can be thought of as ohmic. The homogeneous, isotropic and frequency-independent properties of the extracellular medium are implied by the constant conductivity and hence extracellular-medium independence of location, direction and frequency [32, 27, 17, 37].

Based on this framework, we get an equation for the extracellular potential contribution from a single compartment [19, 32]

$$\phi_n(\mathbf{r}, t) = \frac{1}{4\pi\sigma} \frac{I_n(t)}{|\mathbf{r} - \mathbf{r_n}|}. \tag{2.3}$$

Here $\phi_n(\mathbf{r}, t)$ denotes the extracellular potential contribution from compartment n to a virtual electrode located at position $\mathbf{r}$ at time t. The location of the center of compartment n, where the transmembrane current source I_n can escape or enter, is $\mathbf{r_n}$, see Figure 2.4.

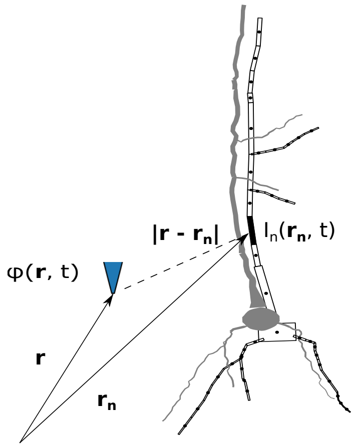

Figure 2.4: Compartment-based forward modeling. A neuron branch is divided into multiple compartments indexed by n. Each compartment has a center location $\mathbf{r}_n$ where a transmembrane current $I_n(\mathbf{r}_n, t)$ can escape or enter. The virtual electrode (blue triangle) records extracellular potential contributions from the neural compartments. The potential contribution depends on the electrode position, $\mathbf{r}$, relative to the compartment location, as well as the extracellular conductivity and the transmembrane currents. The figure is modified from Figure 2a in Lindén et al., 2014 [26].

Due to ohmic properties of the extracellular fluids, electric potentials add linearly and the total extracellular potential can be expressed as

$$\phi(\mathbf{r}, t) = \frac{1}{4\pi\sigma} \sum_{n=1}^{N} \frac{I_n(t)}{|\mathbf{r} - \mathbf{r_n}|}. \tag{2.4}$$

An important thing to note here is the underlying *point-source approximation*, meaning that transmembrane currents can only escape or enter a compartment through a source point located in the compartment center. An alternative approach is the *line-source approximation* where I_n is assumed to be equally distributed along the compartment's centerline axis. In the equation below, Equation (2.4) is integrated along the center-line axis for each compartment, to get an expression for the extracellular potential based on the line source approximation, ϕ_{LS} [21, 35, 26]:

$$\phi_{LS}(\mathbf{r},t) = \frac{1}{4\pi\sigma}\sum_{n=1}^{N} I_n(t)\int \frac{dr_n}{|\mathbf{r}-\mathbf{r_n}|} = \frac{1}{4\pi\sigma}\sum_{n=1}^{N} I_n(t)\frac{1}{\Delta s_n}\log\left|\frac{\sqrt{h_n^2+\rho_n^2}-h_n}{\sqrt{l_n^2+\rho_n^2}-l_n}\right|. \tag{2.5}$$

Here, Δs_n is the length of compartment n, ρ_n is the radial distance, perpendicular to the compartment length, h_n is the longitudinal distance to the bottom of the compartment, and $l_n = h_n + \Delta s_n$ is the longitudinal distance to the top of the compartment.

The line-source approximation is more accurate for electrodes measuring close to the neuron; however, the two methods will converge toward the same result for large distances [21, 22].

2.3 Dipole-Based (DB) Forward Model

In the same way as electric charges can create charge multipoles, a combination of current sinks and sources can set up *current multipoles* [32]. Extracellular potentials, ϕ, can be precisely described by a multipole expansion, i.e., the sum of contributions to electrical potentials from the different orders of current multipoles [32]:

$$\phi(r) = \frac{C_{monopole}}{r} + \frac{C_{dipole}}{r^2} + \frac{C_{quadrupole}}{r^3} + \frac{C_{octupole}}{r^4} + ...,$$

where r is the distance from the current source to the virtual measuring point and the $C_{multipole}$-terms represent the contributions from the corresponding multipoles. Since current conservation applies in neural tissue, there is no such thing as a current monopole contribution to extracellular potentials. When r gets sufficiently large, the quadrupole and octupole terms etc., are negligible compared to the current dipole contribution, hence the extracellular potential can be modeled based on the contribution from the current dipole moment only [32]. Here, a sufficiently large r means approximately $3-4$ times the dipole length, i.e., the distance between the current source and the current sink. This section gives an introduction to current dipole moments before explaining how to incorporate these in the DB model to approximate extracellular potentials.

2.3.1 Current Dipole Moments

The simplest model possible for creating a current dipole is the two-compartment model of a neural stick. Figure 2.5 illustrates a two-compartment neuron model, where the upper compartment represents the dendrites and the lower compartment represents the soma. A current I enters the dendritic compartment through a single point. From here, the current flows intracellularly, and out through the middle of the bottom compartment. The current entering point, marked with a minus sign, is a current sink, while the current escaping point, plus, is a current source. The current flowing inside the neuron between the sink and the source travels along the distance vector $\mathbf{d}$. Note that, analogue to electric circuit currents, we do not picture a single ion moving through the whole circuit.

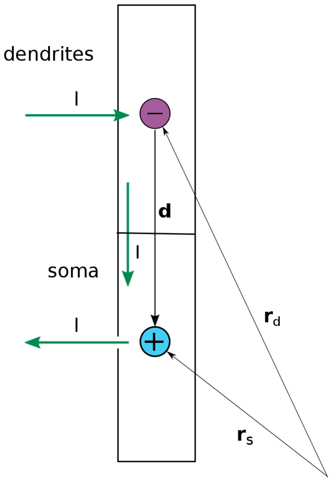

Figure 2.5: Current dipole moment from a two-compartment stick. A two-compartmental neuron model, where the upper compartment represents the dendrites, and the lower represents the soma. Using the point-source approximation, current I flows into the cell through a point at the center of the dendritic compartment. According to volume conductor theory, this point is a current sink, represented by a purple dot at position $\mathbf{r}_d$. The current moves intracellularly and out through the middle point on the bottom compartment, representing a current source, see blue dot with position vector $\mathbf{r}_s$. The current dipole moment $\mathbf{p}$ can be calculated from the axial current, I, and the distance vector, $\mathbf{d}$, pointing from the current sink to the current source: $\mathbf{p} = I\mathbf{d}$, analogue to a charge dipole moment.

Two similar approaches can be used for calculating the current dipole moment from a neuron. The first is completely analogue to charge dipole moments, where charge is replaced by current. The relation below shows how to calculate the current dipole moment, $\mathbf{p}$, from the axial current and the distance between the sink and the source in a two-compartment model,

$$\mathbf{p} = I\mathbf{d}. \tag{2.6}$$

From this, we get the equation for the total current dipole moment from an N-compartment neuron, based on axial currents, by summing up dipole moment contributions from all axial currents between neuron compartments

$$\mathbf{p}_{axial} = \sum_{n=1}^{N-1} I^n_{axial}(t)\mathbf{d}_n. \tag{2.7}$$

Here, $\mathbf{p}_{axial}$ is the total current dipole moment calculated from axial currents, and $\mathbf{d}_n$ is the distance vector going from the start point to the end point of each axial current $I^n_{axial}(t)$.

The current dipole moment can also be calculated from transmembrane currents, see Figure 2.5:

$$\mathbf{p} = I\mathbf{d} = I(\mathbf{r}_s - \mathbf{r}_d) = I\mathbf{r}_s - I\mathbf{r}_d = I_s\mathbf{r}_s + I_d\mathbf{r}_d. \tag{2.8}$$

Here, I_s is the current flowing out of the soma compartment, and I_d is the current flowing out of the dendritic compartment, thus $I_s = I$ and $I_d = -I$. Generalizing this to an N-compartment neuron model, we get the following equation for current dipole moments based on transmembrane currents, $\mathbf{p}_{trans}$:

$$\mathbf{p}_{trans} = \sum_{i=1}^{N} I^i_{trans}(t)\mathbf{r}_i. \tag{2.9}$$

The transmembrane current of compartment i, $I^i_{trans}(t)$, is a function of time, t, and $\mathbf{r}_i$ is the middle position of the compartment.

2.3.2 Point-Source vs. Line-Source Dipole Moment

Considering the CB model, the line-source approximation gives a more accurate estimate than the point-source approximation. This section goes into the question of whether the choice of point-source or line-source approximation has an effect on the calculation of current dipole moments.

Transmembrane currents can be expressed as the spatial integral over the linear current density i. Following, the current dipole moment equation from transmembrane currents (2.9) is split into x-, y- and z- components, so that $\mathbf{p}(t) = p_x(t)\hat{\mathbf{x}} + p_y(t)\hat{\mathbf{y}} + p_z(t)\hat{\mathbf{z}}$, and each direction component is written as a function of i:

$$\begin{aligned} p_x(t) &= \sum_{n=1}^{N} \int x_n i_n(x,t) dx, \\ p_y(t) &= \sum_{n=1}^{N} \int y_n i_n(y,t) dy, \\ p_z(t) &= \sum_{n=1}^{N} \int z_n i_n(z,t) dz, \end{aligned} \tag{2.10}$$

where N is the total number of compartments.

Next, an example for applying the point-source approximation to calculate the current dipole moment from a dendritic stick is outlined. We assume a straight multi-compartmental dendrite model with N compartments, each of length ΔL, elongated in the z-direction only. Its linear current density for the point-source approximation can be expressed as follows

$$i_n(z,t) = I_n(t)\delta(z - z_n), \tag{2.11}$$

where $I_n(t)$ is the space-independent current component and z_n is the middle position of compartment n, i.e., where current can leave or enter. Plugging this into Equation (2.10), and integrating over the length of each compartment, ΔL, the following expression for p_z appears:

$$p_z = \sum_{n=1}^{N} \int_{z_n - \frac{\Delta L}{2}}^{z_n + \frac{\Delta L}{2}} z_n I_n(t) \delta(z - z_n) dz = \sum_{n=1}^{N} z_n I_n(t). \tag{2.12}$$

When calculating the current dipole moment using the line-source approximation, the linear current density takes the following form:

$$i_n(z,t) = \frac{I_n(t)}{\Delta L}. \tag{2.13}$$

Inserting this into Equation (2.10) gives

$$\begin{aligned} p_z &= \sum_{n=1}^{N} \int_{z_n - \frac{\Delta L}{2}}^{z_n + \frac{\Delta L}{2}} z \frac{I_n(t)}{\Delta L} dz \\ &= \sum_{n=1}^{N} \frac{I_n(t)}{\Delta L} \Big[\frac{1}{2} z^2\Big]_{z_n - \frac{\Delta L}{2}}^{z_n + \frac{\Delta L}{2}} \\ &= \sum_{n=1}^{N} z_n I_n(t). \end{aligned} \tag{2.14}$$

Hence, we have found that the point-source and the line-source approximations will give the exact same results when calculating current dipole moments. The simpler point-source approximation is therefore preferable.

2.3.3 Extracellular Potentials from Current Dipole Moments

When modeling extracellular potentials with a virtual electrode placed far away from the neuron source ($r >> d$, where $r = |\mathbf{r}|$ is the distance from the dipole to the electrode and d is the dipole length), the potentials can be accurately approximated with the following equation [35, 32]:

$$\phi(\mathbf{r}, t) = \frac{1}{4\pi\sigma} \frac{|\mathbf{p}(t)| \cos\theta}{|\mathbf{r}|^2}. \tag{2.15}$$

Here $|\mathbf{p}(t)|$ denotes the magnitude of the total current dipole moment from a neuron in the vicinity of a virtual electrode at position $\mathbf{r}$ relative to the dipole position. The angle between $\mathbf{p}$ and $\mathbf{r}$ is denoted by θ, and σ is the conductivity of the extracellular medium. The approximation is expected to be good when $r > 3d$ or $4d$ [32].

Conductivity of Extracellular Media

The conductivity of the extracellular medium in cortex is in this thesis set to a constant, ~ 0.3 S/m [19, 32, 27]. In the brain, conductivity can, however, depend on various properties of the extracellular media. For example can white matter be anisotropic, meaning currents experience a higher conductivity when traveling along nerve fibers, than across [32]. When recording EEG signals, the conductivity differs significantly for cortex, skull and scalp [32], and assuming the extracellular medium to be homogeneous is therefore a substantial simplification. It is also debated whether conductivity is frequency dependent [37, 27]. The extracellular conductivity is, however, for the most part only expected to affect the amplitude of the measured signal.

Chapter 3

Methods

In order to study single-cell contributions to extracellular potentials, neuroinformatics tools such as NEURON and LFPy can be applied. NEURON is a widely used simulation environment for building and using computational neuron models [11, 12]. The Python package, LFPy, runs on top of NEURON [5, 11], and can use NEURON's compartmental modeling tools to simulate extracellular potentials. Extracellular potential simulations in LFPy are based on the CB (compartment-based) model described in Section 2.2. There is, however, no available module for calculations of extracellular potentials with the DB (dipole-based) model.

This chapter first describes how extracellular potentials can be computed with the CB model and LFPy. Further, we give a detailed explanation of how a new Python simulation tool, based on the DB model, was developed from transmembrane currents (TC) and axial currents (AC). In the last section, cell models used for method validation and simulations are presented.

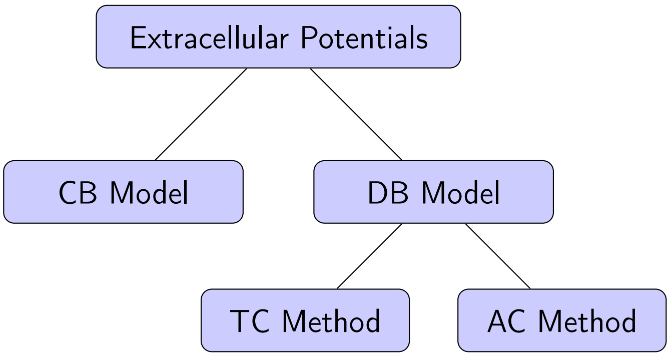

Figure 3.1: How to calculate extracellular potentials. Two different models can be used for calculating extracellular potentials from a neuron: the CB (compartment-based) model or the DB (dipole-based) model. The DB model can be based on either transmembrane currents, i.e., the TC method, or axial currents, i.e., the AC method.

3.1 Compartment-Based (CB) Forward Model with LFPy

In this section, we give a short introduction to a selection of essential LFPy-classes, and how these can be used for calculating extracellular potentials.

First of all, a neuron object is established with the cell class `LFPy.Cell`. This class can load a neuron morphology file and be used to set a number of properties needed to specify the neuron's inherent dynamics [26]. Since LFPy runs on top of NEURON, the cell morphology is organized in the NEURON way: the neuron is split into continuous unbranched cables called *sections*, and each section is split into a number of smaller compartments, referred to as *segments* [11]. All segments are given an index, starting with the root segment, index 0, and segments within a section are indexed by consecutive numbers. The soma is normally defined as the root segment and does always consist of a single segment within a single section [11].

The `LFPy.Synapse` class is used for creating a synaptic input current activated by spike trains. The synapse is placed on a cell by assigning a cell compartment index to the synapse class [26]. In order to record what is going on in the extracellular medium, an electrode class is needed: `LFPy.RecExtElectrode`. Keyword arguments are given to specify which cell to record from and where in space the electrode should be located [26].

Simulation of the cell is done by calling `LFPy.Cell.simulation()`, and after simulating a cell with synapses and electrodes, LFPy can calculate the extracellular potentials with `LFPy.RecExtElectrode.calc()`. The calculations are based on the CB model described in Section 2.2 [26]. Figure 3.2 is an example of how one can import a neuron morphology into LFPy and calculate extracellular potentials using the above-mentioned classes and methods.

3.2 New Python Module for the Dipole-Based (DB) Forward Model

To calculate extracellular potential contributions from single cells based on the DB model, one first has to compute the neuron's current dipole moment. As explained in Section 2.3, current dipole moments can be calculated using two different methods, the transmembrane current (TC) method and the axial current (AC) method, cf. Equation (2.9) and (2.7). The following two sections go into the implementation of the TC method and the AC method in Python. The TC method is largely based on LFPy, while the AC method was built up from scratch in Python, and is therefore described in a highly detailed manner to make it easy for other students to build on the work. The third section explains how Equation (2.15) was implemented in

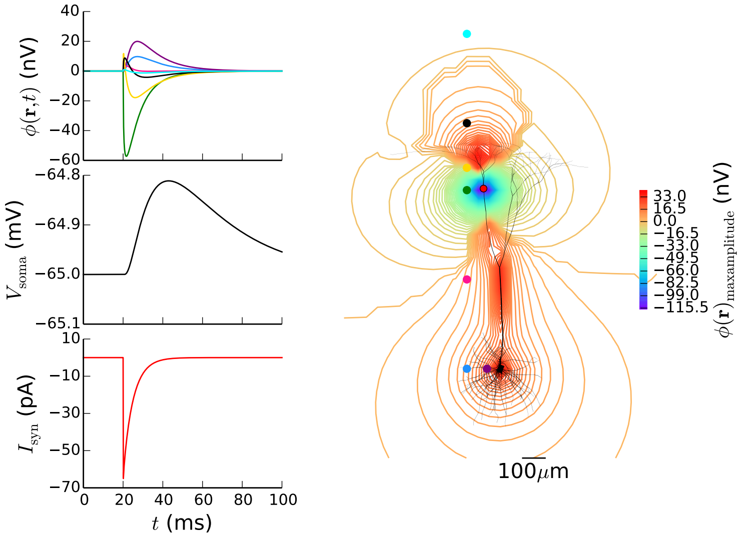

Figure 3.2: Extracellular potentials calculated with LFPy. Multiple virtual electrodes record extracellular potentials around a neuron with one excitatory, conductance-based synapse. The neuron model is a passive layer-5 pyramidal cell from cat visual cortex, ref. Mainen and Sejnowski, 1996 [29]. The synapse is located at the red dot and the synaptic input current, I_{syn}, is plotted in the lower left plot. Membrane potential recorded in the soma, V_{soma}, is shown in the middle left panel. Maximum amplitudes of extracellular potentials, $\phi(\mathbf{r})$, are shown as equipotential lines surrounding the neuron plot on the right. Seven virtual electrodes are represented as colored dots and the potentials recorded are plotted in the upper left panel. Each colored curve refers to the electrode of the same color. The figure is inspired by Figure 3 in Lindén et al., 2014 [26].

`Python` to finally calculate the potentials from the current dipole moments.

3.2.1 Current Dipole Moments from Transmembrane Currents (TC)

In `LFPy`, finding transmembrane currents is straightforward because of the following `LFPy.Cell` class attribute:

```
# list of transmembrane currents for all segments in cell
i_trans = cell.imem
```

`cell.imem` is an array of transmembrane currents flowing into or out from all compartments. If the point source approximation is specified, one segment only gets one transmembrane current. Further, all the segment positions are calculated using the following code:

```
segment_positions = zip(cell.xmid, cell.ymid, cell.zmid)
```

Now, the current dipole moment can easily be calculated from transmembrane currents applying Equation (2.9), i.e., by multiplying transmembrane currents with their position vectors.

3.2.2 Current Dipole Moments from Axial Currents (AC)

There is no built-in method for calculating axial currents in `LFPy`. From the theory chapter, we know that the axial current between two neighbor compartments is given by the first term on the left hand side of Equation (2.1)

$$I_{seg} = \frac{V_{seg} - V_{par}}{r_{axial}}. \tag{3.1}$$

Here, $I_{seg} = I^n_{axial}$ is the current flowing into a segment n, from its parent segment $n-1$. Further, $V_{seg} = V_n$ and $V_{par} = V_{n-1}$ are the membrane potentials measured in the middle of the segment and the parent segment, respectively. The axial resistance between the two compartment midpoints is denoted by $r_{axial} = \frac{1}{g_{n-1,n}}$. The membrane potential in the middle of each neuron compartment is stored in the `LFPy` Cell attribute `cell.vmem` after the neuron simulation. The axial resistance, on the other hand, proved to be more difficult to find. The following five examples illuminate the difficulties related to calculating inner axial resistances and axial currents of a neuron.

Dendritic Stick

We start with the simplest possible example, namely a single, unbranched dendrite illustrated in Figure 3.3. In order to get properties like axial resistance from the dendritic stick model, it turned out to be useful to dig into `NEURON` through `Python`, in addition to using `LFPy` tools.

As mentioned above, `NEURON` organizes neural models into sections and segments. Furthermore, a segment is modeled equivalent to an electrical compartment described in Section 2.2.1: i.e., in `NEURON` a segment is treated as an electrical circuit. The dendritic stick in this example is a one-section model with four segments. The following code finds the axial resistance between two segments [3, 4]:

```
import neuron as nrn

for sec in nrn.h.allsec():
    for seg in sec:
        x = seg.x                  # segment middle position
        seg_ri = nrn.h.ri(x)       # axial resistance from
                                   # preceding seg to seg
```

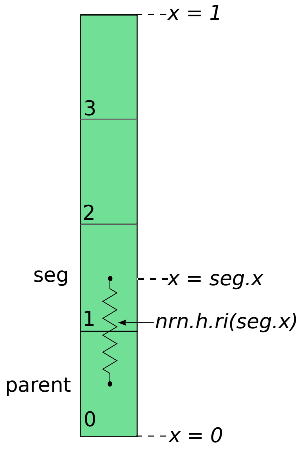

Figure 3.3: Illustration of dendritic stick. Simple neuron model consisting of 1 dendrite with 4 compartments. The dendrite is a `NEURON` section and each compartment is a segment. Within a section, segments are indexed by consecutive numbers. All segments except segment 0 has a parent segment. i.e., segment 0 is the root segment, and is not growing out of any other segment. The zigzag-line illustrates the axial resistance along the path going from the node in the middle of a segment, to the node in the middle of its parent. `nrn.h.ri(seg.x)` returns the axial resistance.

Here, `nrn.h.allsec()` is a list of all sections in the specified cell[1]. In `NEURON` each section is given a *continuous arc length*, `x`, ranging from 0 to 1. By looping over all segments in the section, the arc length of the center of each segment can be found with the `NEURON` attribute `x = seg.x`. Next, the path length is inserted into the `NEURON` function `nrn.h.ri(x)`. The following is key: `nrn.h.ri(x)` returns the axial resistance along the path between position x of the segment, `seg`, and the middle of its *parent segment* [11]. A parent segment in `NEURON` is the segment that the concerned segment "grows out of". In Figure 3.3 segment 0 is the parent segment of segment 1. Segment 1 is the parent of segment 2 etc. Thus `nrn.h.ri(seg.x)` returns the axial resistance between the center of the segment `seg` and the center of its parent segment[2].

`NEURON` calculates `nrn.h.ri()` from the following equation, which depends on

[1] The `NEURON` method `nrn.h.allsec()` is equivalent to `cell.allseclist` in `LFPy`. It turned out, however, that looping over the `LFPy` list was problematic when using certain `LFPy` tools, such as `cell.get_idx()`, that internally loop over the same list. `nrn.h.allsec()` is thus a safer choice.

[2] The `NEURON` function `nrn.h.ri(seg.x)` is equivalent to the `LFPy` method `seg.ri()`. `nrn.h.ri()` has advantages needed for more complicated cases. We therefore introduce the `NEURON`-version only, to avoid confusion.

the length, diameter and inner resistance of the concerned piece of neural cable [11]

$$r_{axial} = \frac{R_a l}{\pi (\frac{d}{2})^2}. \tag{3.2}$$

Here, r_{axial} is the total axial resistance of a cable with length l, R_a is the inner resistivity and d is the cable diameter. For neurons with varying segment diameters, NEURON will integrate up this formula to get the total resistance. The axial currents of a dendritic stick can easily be found by plugging r_{axial} into Equation (3.1).

Ball-and-Stick Neuron

In this example we study the axial resistance of the two-section ball-and-stick neuron illustrated in Figure 3.4.

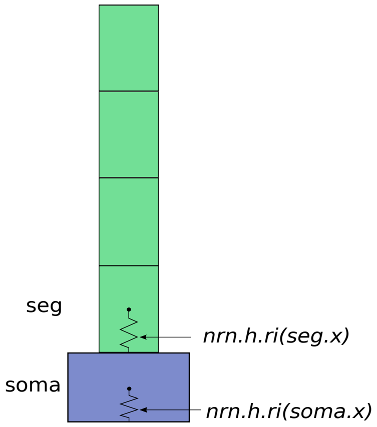

Figure 3.4: Illustration of ball-and-stick neuron. The ball-and-stick neuron has two sections: the green dendrite and the blue soma. Segment 0 and 1 are both bottom segments, i.e., the first segment within their sections. `nrn.h.ri(seg.x)` returns the axial resistance from the middle point to the starting point of the segment `seg`. To get the axial resistance between the `seg` and `soma`, one can simply add `nrn.h.ri(soma.x)` to `nrn.h.ri(seg.x)`.

When applying `nrn.h.ri(seg.x)` to the bottom dendrite segment, segment 1, one would expect the axial resistance along the path from the middle of section 1 to the middle of section 0 to be computed. It appears, however, that NEURON creates a "ghost segment" of zero length at the bottom of each section. Thus, the ghost segment, and not segment 0, is functioning as the parent segment of segment 1. Therefore, `nrn.h.ri(seg.x)` returns the axial resistance from the middle to the bottom of the given segment exclusively, whenever the segment is the first segment of a section. The first segment in each section is here referred to as a *bottom segment*, and accordingly, the soma is always a bottom segment. `nrn.h.ri(seg.x)` and `nrn.h.ri(soma.x)` are illustrated in Figure 3.4. For this relatively simple example, the two resistances could be added to get the sought-after axial resistance between the two nodes. Further, Equation (3.1) gives the axial currents of the ball-and-stick neuron.

Neuron with Varying Segment Diameters

Our improvised solution to the previous example is no longer valid as soon as the neuron model gets more complicated. Real neuron morphologies do not have constant segment diameters, and we see from Figure 3.5 that the resistance of the lower half can no longer replace the resistance of the upper half of a segment.

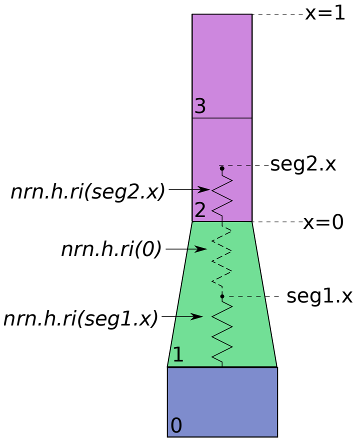

Figure 3.5: Illustration of ball-and-stick neuron with varying segment diameters. This ball-and-stick neuron has three sections: the blue soma, one green dendrite section and one purple dendrite section. The diameter of the green dendrite segment, compartment 1, is not constant and the axial resistance of the lower part is therefore not equal to the upper-part resistance. The upper part resistance is illustrated by the dotted zigzag line and can be found with `nrn.h.ri(0)`. The 0 and the 1 on the right-hand side of the neuron mark the start and end points of the purple section. By adding the resistance of the upper part of `seg1` to the lower part resistance of `seg2`, we get the total axial resistance between the two segments [4].

In the code snippet below, we show how to find the axial resistance between the compartments of a neuron with varying diameter.

```
import neuron as nrn
for sec in nrn.h.allsec():
    new_sec = True
    # no need to go through soma, since soma is an orphan
    if 'soma' not in sec.name():
        for seg in sec:
            if new_sec:
                # axial resistance from segment middle point
                # to segment start:
                segment_ri = nrn.h.ri(seg.x)
                # NEURON way of finding axial resistance from
                # segment at position 0 = start to parent mid:
                parent_ri = nrn.h.ri(0)
                total_ri = segment_ri + parent_ri
                new_sec = False
            else:
              total_ri = nrn.h.ri(seg.x)
```

It appears that when `x = 0` is inserted into `nrn.h.ri()`, `NEURON` is no longer distracted by the ghost segment. This way, one can calculate the axial resistance from the bottom of the section we are looping over, to the middle of its parent segment. By adding `nrn.h.ri(0)` to `nrn.h.ri(seg.x)` we get the axial resistance from the middle of a bottom segment to the middle of its parent segment.

Ball-and-Y Neuron

For the ball-and-Y neuron illustrated in Figure 3.6 we can use the same way of finding the axial resistances as for the preceding example. Figuring out the paternity of a neuron segment, however, is no longer straightforward. Because a segment is not necessarily an only child, another issue is that the current flowing out of a parent segment, I_{par} can be divided. One part will flow into the segment we are concerning, I_{seg}, and the rest, I_{sib}, flows into its sibling segment. From now on, all axial currents are therefore spilt into two parts. One is the current going from the parent middle to the parent end. The other is going from the parent end, i.e., the segment start, to the segment middle.

Assuming that a dendrite never splits into more than two branches, and given that we know the membrane potentials at the middle position of all segments and the axial resistance everywhere, axial currents can accordingly be found from the following equations:

$$I_{par} = I_{seg} + I_{sib}, \tag{3.3}$$

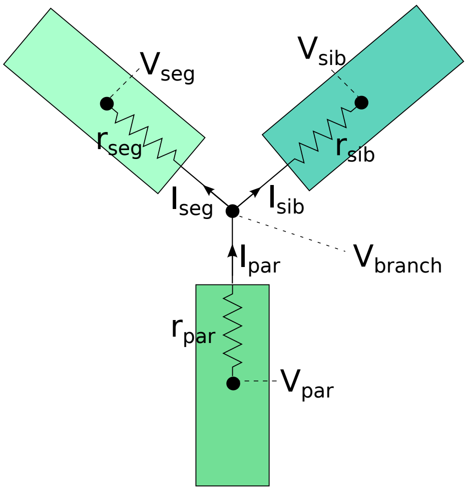

Figure 3.6: Dendrite branching. Ball-and-y neuron with three sections, each consisting of one segment. The upper left segment is here referred to as the segment, or *seg*, the upper right is the sibling, *sib*, and the lower segment is the parent, *par*. The membrane potentials measured at the midpoint of each compartment are denoted by V_{seg}, V_{sib} and V_{par}. Further, r_{seg} is the axial resistance along the path from a segment's midpoint to the branch node and V_{branch} is the membrane potential at the branching point. The current flowing out of the parent segment, I_{par}, splits into I_{seg} and I_{sib}, the currents entering the segment and the sibling, respectively.

and

$$I_{seg} = \frac{V_{seg} - V_{branch}}{r_{seg}}, \quad I_{sib} = \frac{V_{sib} - V_{branch}}{r_{sib}}, \quad I_{par} = \frac{V_{branch} - V_{par}}{r_{par}}, \tag{3.4}$$

where V_{seg}, V_{sib}, V_{par} and V_{branch} are the membrane potentials at the segment midpoint, the sibling midpoint, the parent midpoint and the branchpoint, respectively. Combining Equations (3.3) and (3.4) we get the following expression for V_{branch}:

$$V_{branch} = \frac{\frac{V_{par}}{r_{par}} + \frac{V_{seg}}{r_{seg}} + \frac{V_{sib}}{r_{sib}}}{\frac{1}{r_{par}} + \frac{1}{r_{seg}} + \frac{1}{r_{sib}}}. \tag{3.5}$$

Setting V_{branch} into Equation (3.4) gives us all axial currents related to a branch point.

Next, we have to figure out the neuron's family relations. Because of the branching, the parent segment of a bottom segment is no longer necessarily the preceding segment, index wise. In order to use the equations above, the index of each bottom segment, its parent segment and potential sibling segment must be determined. Within sections, i.e., for segments that are not bottom segments, these indices are trivial to find (due to no branching within segments and consecutive indexing). The following code can be used for finding the bottom segment indices:

```
for sec in nrn.h.allsec():
    seg_idx = cell.get_idx(section=sec.name())[0]
```

`cell.get_idx()` returns an array containing indices for all segments in the section. The first index segment in the array refers to the bottom segment index, selected with `[0]`. Next, a *section reference* is created in order to check whether a bottom segment has a parent and get the parent segment and its index:

```
for sec in nrn.h.allsec():
    if ’soma’ not in sec.name():
        secref =  nrn.h.SectionRef(sec.name())
        not_orphan = secref.has_parent()        # boolean
        parentseg = secref.parent()
        parentsec = parentseg.sec
        parent_idx = cell.get_idx(section=parentsec.name())[-1]
```

`NEURON` has a set of tools only accessible from section references, `nrn.h.SectionRef()`, such as `secref.has_parent()`, that can be used to make sure we do not try to find the axial current from the soma to its non-existent parent. A section reference can also give us the parent segment of a section: `secref.parent()`. `segment.sec` is a `NEURON` trick for getting the section holding a segment. Since the parent segment is the top segment of its section, the last index of the `cell.get_idx()` list is chosen, with `[-1]`.

There are several ways of finding the sibling of a bottom segment, however, it was found that the cleanest way to do this, was by once again taking advantage of a section reference method, to find the children sections of each section.

```
for sec in nrn.h.allsec():
    secref = nrn.h.SectionRef(sec.name())
    for child in secref.child:
        child_idx = cell.get_idx(section = child.name())[0]
```

`secref.child` is a list of sections holding the children segments of a parent segment. `cell.get_idx()` can then give us the index of the children segments. Storing these segment indices in a dictionary, we can for each bottom segment pop the sibling segment index via its parent section. Now that the family ties of the bottom segments are figured out, the axial currents can be calculated for branching dendrites.

Dendrite Soma Connection

When the code was tested on more complicated morphologies, see Chapter 4, errors occurred due to a misunderstanding related to how dendrites are connected to the soma. Based on the recommendations in the `NEURON` documentation [11], the assumption was that a section attachment is always created so that the top or the bottom of a section is connected to the top or the bottom of another. In a file specifying the morphology in `NEURON`'s native `hoc` language, this would for instance look like:

```
connect dend(0), soma(1)
```

where 0 and 1 refer to the minimum and the maximum arc length of the sections. It appeared, however, that when it comes to connecting a dendrite to the soma, it is customary to connect the dendritic section to the center of the soma section [20, 29]:

```
connect dend(0), soma(0.5)
```

The axial current flowing between the soma and the connected segment, $I_{somatoseg}$, can therefore be computed without taking the axial resistance of the soma into account:

```
v_seg = cell.vmem[segment_idx] # segment membrane potential
v_soma = cell.vmem[parent_idx] # soma membrane potential
r_seg = nrn.h.ri(seg.x)        # axial resistance from segment
                               # start to segment mid
i_soma_to_seg = (v_seg - v_soma)/r_seg.
```

This means that the axial resistance of the soma does not contribute to the total axial resistance felt by a current flowing from the middle of the soma into a dendrite.

In order to work out the current dipole moment contribution from each axial current, the distance vectors traveled by each small current must be calculated.

```
dseg = [cell.xmid[seg_idx] - cell.xstart[seg_idx],
                cell.ymid[seg_idx] - cell.ystart[seg_idx],
                cell.zmid[seg_idx] - cell.zstart[seg_idx]]
dpar = [cell.xstart[seg_idx] - cell.xmid[parent_idx],
                cell.ystart[seg_idx] - cell.ymid[parent_idx],
                cell.zstart[seg_idx] - cell.zmid[parent_idx]]
```

Here `dseg` is the distance vector from the bottom to the middle of the segment. `dpar` is the distance vector from the middle to the top of the parent segment.

Now, the current dipole moment contribution from half a neural segment, can be found by multiplying its length, `dseg`, with the axial current flowing inside, I_{seg}. Summing up all contributions, seeing Equation(2.7), we can compute the total current dipole moment from a complex neuron morphology.

Based on this, two python modules were written: One returning the axial currents and distance vectors between all adjacent compartments of a neuron, and another taking currents and distances and returning current dipole moments. The second takes both transmembrane and axial currents, according to Equations (2.9) and (2.7). This was important so as to enable internal checks of the code. The full code is included in Appendix B.

3.2.3 Extracellular Potentials from Current Dipole Moments

In order to approximate extracellular potentials with Equation (2.15), the orientation of the current dipole moment in space needs to be considered. The spatial orientation of the current dipole moment vector can be specified by two angles: θ_0 and ϕ_0, where θ_0 is the angle between $\mathbf{p}$ and the z-axis, and ϕ_0 is the angle between $\mathbf{p}$ and the x-axis. Note that the letter ϕ is used for both electric potential and radial angle, however, it should be clear from the context which is which.

The current dipole approximation for electric potentials is given by Equation (2.15):

$$\phi(\mathbf{r},t) = \frac{1}{4\pi\sigma}\frac{|\mathbf{p}(t)|\cos\theta}{|\mathbf{r}|^2}.$$

Once the current dipole moment $\mathbf{p}(t)$ from a neuron is calculated, we can estimate the extracellular potentials from the cell at any point in space. The extracellular conductivity σ is constant, see Section 2.3.3, and the length of the distance vector from the dipole vector to the electrode, $\mathbf{r}$, is known, see Figure 3.7. Before plugging $|\mathbf{p}(t)|$ and $|\mathbf{r}| = r$ into Equation (2.15), $\cos\theta(t)$ must be calculated. Here, $\theta(t)$ is the angle between $\mathbf{p}(t)$ and $\mathbf{r}$. As illustrated in Figure 3.7, θ can be calculated by transforming θ_0 and θ_e, where θ_e is the angle between $\mathbf{r}$ and the z-axis. A simpler way of doing this, however, is by calculating $\cos\theta(t)$ directly from the definition of the dot product

$$\mathbf{r}\cdot\mathbf{p}(t) = |\mathbf{r}||\mathbf{p}(t)|\cos\theta(t)$$

$$\Downarrow \tag{3.6}$$

$$\cos\theta(t) = \frac{\mathbf{r}\cdot\mathbf{p}(t)}{|\mathbf{r}||\mathbf{p}(t)|}. \tag{3.7}$$

Based on this, the current dipole approximation was implemented in `Python`. First of all, the location of the current dipole moment, $\mathbf{r}_{dipole}$, was defined as the middle position between the soma and the mean synapse location:

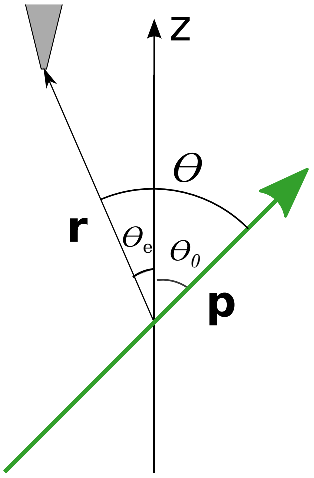

Figure 3.7: Orientation of current dipole moment relative to electrode position. Illustration of the angle θ between the current dipole moment $\mathbf{p}$ and the distance vector from the electrode measure point to the middle of the current dipole, $\mathbf{r}$. $\theta = \theta_0 + \theta_e$ where θ_0 is the angle between $\mathbf{p}$ and the z-axis, and θ_e is the angle between $\mathbf{r}$ and the z-axis.

```
syninds = cell.synidx
r_soma_syns = [cell.get_intersegment_vector(idx0 = 0,
               idx1 = i) for i in syninds]

r_mean = np.average(r_soma_syns, axis = 0)
r_dip = r_mean/2. + cell.somapos
```

Here, `r_soma_syns` is a list of all distance vectors from the soma to the neuron's synapses. The average distance vector, `r_mean`, is divided by two, to get the middle position. The soma position vector, `cell.somapos`, is added, giving `r_dip`, the position vector from the origin to the current dipole moment vector. The distance vector from the current dipole moment to the electrode, **r**, can now easily be found:

```
r = r_dip - r_e
```

Next, $\cos\theta(t)$, $\theta(t)$ and the extracellular potential $\phi(t)$ were calculated:

```
cos_theta = np.dot(P, r)/(np.linalg.norm(r)*np.linalg.norm(P
            , axis = 1))
cos_theta = np.nan_to_num(cos_theta)
theta = np.arccos(cos_theta)
phi = 1./(4*np.pi*sigma)*np.linalg.norm(P,
          axis = 1)*cos_theta/np.sum(r**2)*1E6
```

When dividing by the current dipole moment, `P`, in the first line, we're actually doing division by zero, since the current dipole moment is zero before the first input current is initiated. This problem is fixed by np.nan_to_num. The extracellular potential, phi, is converted from millivolts to nanovolts by multiplying with 1E6.

The two Python classes with tools for calculating current dipole moments and extracellular potentials from the current dipole approximation can be found in Appendices B and C.

3.3 Cell Models

This section describes the neuron models studied in the results: two idealized models and two anatomically reconstructed models.

Idealized Neuron Models

By idealized neuron models, we mean simple neuron morphologies well suited for conceptual understanding and method validation, like the simple morphologies used for method development in Section 3.2.2, see Figures 3.3-3.6. Here, two such models are presented: a ball-and-stick neuron and a ball-and-Y neuron. Both neuron models were manually implemented in NEURON's native hoc language. The morphologies were specified by setting the size of each section and determining how the sections are to be connected. The ball-and-stick neuron model has two sections only; a soma and a dendrite. Both sections have constant diameters. The ball-and-Y neuron, on the contrary, has varying section diameters in all sections except for the soma. The dendrite consists of five sections with two branch points. Both neuron models are passive, and use LFPy's default model parameters shown in Table 3.1.

Anatomically Reconstructed Neuron Models

More complex neuron models can be found in the literature, such as the neuron model from the article by Mainen and Sejnowski [29], henceforth referred to as the MS neuron. This is a digital reconstruction of a pyramidal cell from layer-5 in cat visual cortex. This model possesses active conductances, however, for simplicity, the default passive LFPy parameters in Table 3.1 are used instead.

The Hay neuron is a model of a pyramidal neuron from layer-5b in rat cortex [20]. Like the MS neuron, this is a neuron with a complex morphology and even though active conductances are available, these are here replaced by the passive parameters in Table 3.1.

Table 3.1: Model Parameters. Default LFPy model parameters [5].

	Symbol	Code	Value	Units
Membrane resistance	r_m	`cell.rm`	30000	Ωcm^2
Axial resistance	r_{axial}	`cell.Ra`	150	Ωcm
Membrane capacitance	c_m	`cell.cm`	1	$\mu\text{F}/\text{cm}^2$

3.4 Synaptic Models

In this project, only exponential input currents are used, meaning that when a synaptic current is initiated, the opening of ion channels at the spike time t_s is simulated by a discontinuous jump followed by an exponential decay. This current can be modeled in two different ways, the first with a *conductance-based input current*:

$$I_{conductance}(t, t_s) = \bar{g}_s e^{-\frac{(t-t_s)}{\tau_m}} (V(t) - E_{syn}). \tag{3.8}$$

Here, t_s is the time of the incoming spike, g_s is the maximum conductance, τ is the membrane time constant, V is the membrane potential and E_{syn} is the synapse reversal potential. The second way of modeling this, is with a *current-based input current*

$$I_{current}(t, t_s) = \bar{I}_s e^{-\frac{(t-t_s)}{\tau_m}}, \tag{3.9}$$

where $\bar{I}_s$ is the maximum input current. Here, the change in membrane potentials with time is not taken into account.

The time of spiking can be chosen in several different ways. The spike times can for example be set by hand or drawn from a stationary Poisson distribution, using the stationary Poisson input generator in LFPy: `LFPy.inputgenerators.stationary_poisson()`. In this project we use two different types of synaptic input only:

- Conductance-based exponential synaptic input, with one spike set by hand
- Current-based exponential synaptic input, with spike times drawn from a stationary Poisson distribution with an average of five spikes per simulation

Chapter 4

Results

The extracellular potentials from neurons in the brain can be studied using the compartment-based (CB) model and the dipole-based (DB) model, described in Chapter 3. Since the DB model is based on current dipole moments, the first section of this chapter looks into how the methods for calculating current dipole moments were validated. Further, virtual recordings of extracellular potentials from single cells were performed with intracranial as well as ECoG and EEG electrodes. The underlying neuron simulations were carried out for different synapse positions and a varying number of synapses. By comparing the extracellular potentials calculated with the two models, this chapter is meant to give an overview over the applicability of the DB model.

4.1 Validation of Current Dipole Moment Methods

There are two ways of calculating the current dipole moment from a single neuron: the transmembrane current (TC) method and the axial current (AC) method. The TC method, described in Section 3.2.1, is based on Equation 2.9, while the AC method is based on Equation 2.7, see Section 3.2.2. The two methods were compared by calculating current dipole moments from simple idealized neuron models and anatomically reconstructed neurons. Here, the results for four different morphologies are presented.

4.1.1 Idealized Neuron Models

The current dipole moments were calculate for two idealized neuron models: the ball-and-stick and the ball-and-Y neuron. The differences between the two models are described in Section 3.4, in short the ball-and-Y morphology is slightly more complicated, making it comparable to anatomically reconstructed models.

Simulations lasting 100 ms were carried out for both neurons, where each neuron had an excitatory synapse placed on an apical dendrite, see Figures 4.1 and 4.2. Furthermore, exponential, conductance-based input currents, see Section 3.4, were initiated after 10 ms, generating fluctuations in the membrane potentials. The resulting transmembrane currents and membrane potentials were calculated for all segments in the cell. By studying these, it is apparent that excitatory synapses set up currents flowing into the cell, resulting in negative transmembrane currents and an increase in membrane potential around the synapse, see I_{syn} and V_{syn} in panels B and C. The amplitudes of I_{soma} and V_{soma} are smaller, because only a fraction of the current flowing into the cell around the synapse reaches the soma before re-entering the extracellular medium. The somatic transmembrane current is positive, due to Kirchhoff's current law.

From transmembrane currents and membrane potentials, current dipole moments were computed with the TC and the AC methods. Studying panels D and E on the two figures, it is clear that the results from the two models are indistinguishable, with a relative error $\sim 10^{-10}$. The relative error has sharp peak at time $t = 10$ ms, before it seemingly flattens out. The relative error curves are, however, not smooth. For $t \approx 40 - 50$ ms, we can see small oscillations due to round-off errors, explained by the very small size of the signal, see scales in panels D and E.

In addition to investigating the magnitude of the current dipole moment, the direction of $\mathbf{p}(t)$ was studied for the ball-and-Y neuron, see panels F and G on Figure 4.2. Since this neuron morphology is extended in the xz-plane only, the radial angle of the current dipole moment, ϕ_0, is zero throughout the whole simulation. The angle between $\mathbf{p}(t)$ and the z-axis is given by θ_0, and $\bar{\theta_0}$ is the average θ_0 summed over the time when $|\mathbf{p}(t)| \neq 0$. After the synapse is activated, θ_0 increases to ~ 2.57 radians, close to $\bar{\theta}$. Both θ_0 and ϕ_0 are set to 0 when $|\mathbf{p}| = 0$.

The current dipole moment from a ball-and-stick neuron has already been calculated analytically, ref. Pettersen et al., 2014 [34]. This has, however, not been done for more complicated morphologies, such as the ball-and-Y neuron.

The two simulations described here are only two examples of many simulations carried out for similar morphologies, giving equally small errors. From this it is safe to conclude that the TC and AC methods are applicable for calculating current dipole moments from idealized neuron morphologies.

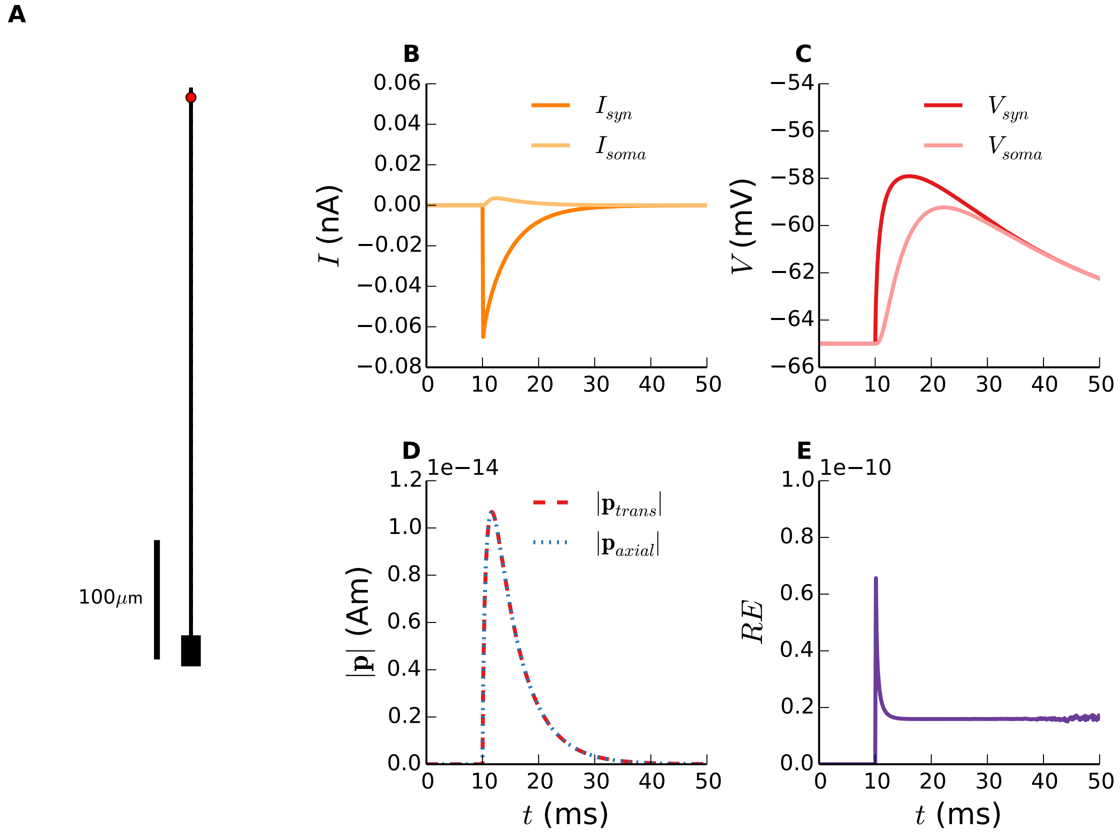

Figure 4.1: Current dipole moment from ball-and-stick neuron. **A:** Ball-and-stick neuron morphology with an excitatory synapse placed on an apical dendrite, illustrated by a red dot. During the 100 ms simulation an exponential, conductance-based synaptic input was initiated after 10 ms. Neuron parameters are listed in Section 3.3. **B:** Transmembrane currents recorded in the synapse compartment, $I_{syn}(t)$, and the soma, $I_{soma}(t)$. **C:** Membrane potentials, $V_{syn}(t)$ and $V_{soma}(t)$, measured in the synapse and the soma compartments, respectively. **D:** Current dipole moment amplitude computed with the TC method, $|\mathbf{p}_{trans}(t)|$ and the AC method, $|\mathbf{p}_{axial}(t)|$. **E:** Relative error, RE, comparing $|\mathbf{p}_{axial}(t)|$ to $|\mathbf{p}_{trans}(t)|$.

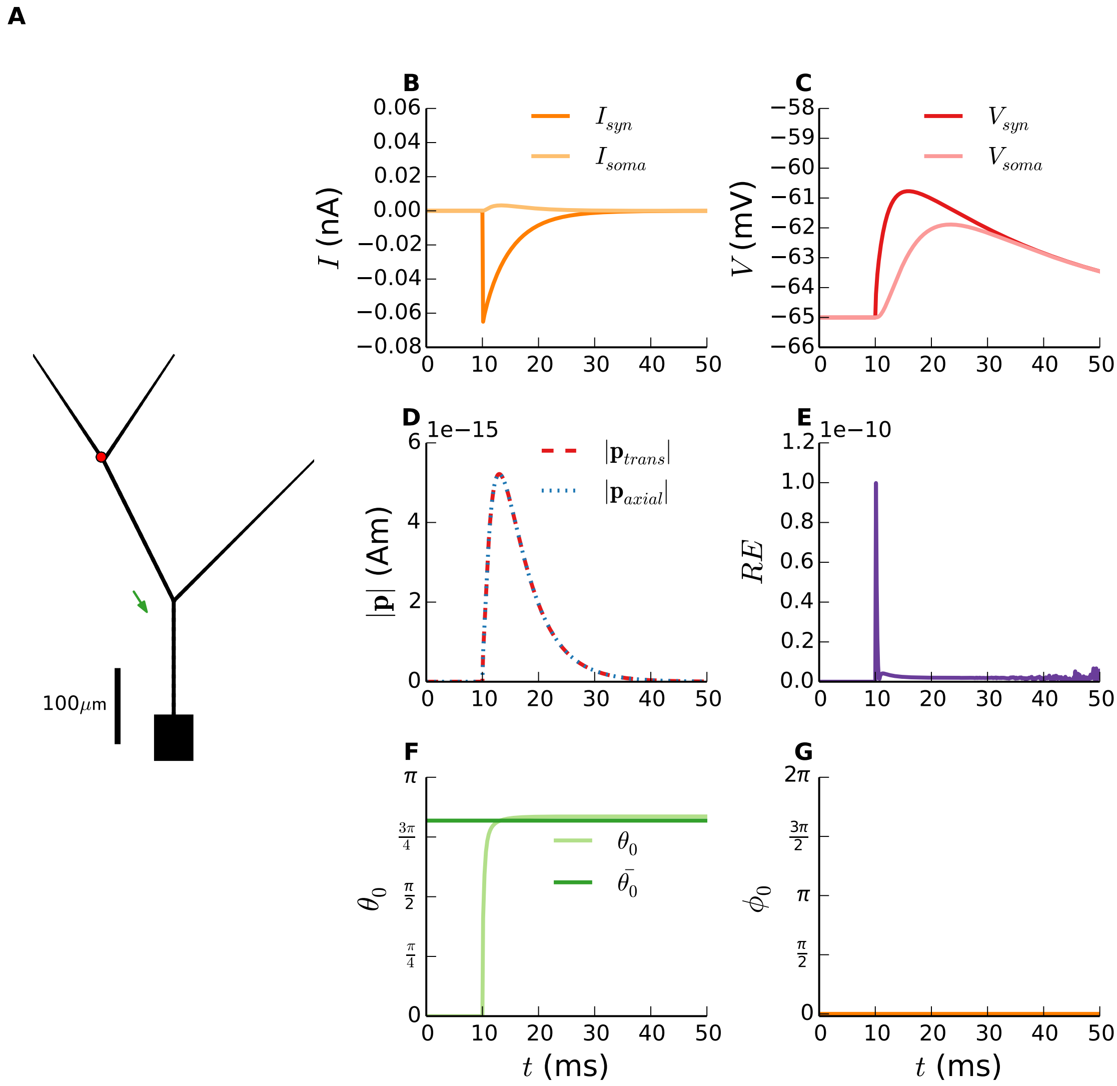

Figure 4.2: Current dipole moment from ball-and-Y neuron. **A:** Ball-and-Y neuron morphology, described in Section 3.3, with an excitatory synapse placed on an apical dendrite, marked by a red dot on the figure. The synaptic input is exponential and conductance-based, with a spike initiated at time $t_s = 10$ ms. The orientation of the resulting current dipole moment is illustrated by a green arrow. **B:** Transmembrane currents in the synaptic and the somatic compartments, denoted by $I_{syn}(t)$ and $I_{soma}(t)$. **C:** Membrane potentials recorded in the synapse compartment, $V_{syn}(t)$, and in the soma compartment, $V_{soma}(t)$. **D:** The magnitude of the total current dipole moment, computed with the TC method, $|\mathbf{p}_{trans}(t)|$ and the AC method, $|\mathbf{p}_{axial}(t)|$. **E:** The relative error, RE, between the current dipole moments in panel **D**. **F:** The angle between the current dipole moment vector and the z-axis, $\theta_0(t)$, and the mean value, $\bar{\theta_0}$, averaged over $\theta_0(t)$ for $t > t_s$. **G:** The angle between the radial component of the current dipole moment and the x-axis, ϕ_0.

4.1.2 Anatomically Reconstructed Neuron Models

In order to test whether the TC and the AC methods are applicable for more complex neuron morphologies, current dipole moments from the Mainen and Sejnowski (MS) neuron and the Hay neuron, described in Section 3.4, were calculated. The simulations were carried out in the same way as in the previous section, with the exact same synaptic excitatory input currents.

The results are shown in Figures 4.3 and 4.4, where panel A shows the neuron morphology and the excitatory synapse placed on an apical dendrite. Examples of transmembrane currents and membrane potentials are shown in panels B and C, while D shows the magnitude of the current dipole moments calculated with the TC and the AC methods. Panels D and E illustrate how $|\mathbf{p}_{trans}(t)|$ and $|\mathbf{p}_{axial}(t)|$ are almost identical, with a relative error $\sim 10^{-10}$.

The orientation of $\mathbf{p}$ in space was calculated for both morphologies and shown in panels F and G. For both cells, the current sink is located above the current source on the neuron and the dendrites are mainly elongated in the z-direction. Therefore, the current dipole moments point downwards, see green arrow in panel A, and θ_0 is close to $\bar{\theta_0} \sim \pi$. The radial angle, ϕ_0 varies with time, due to currents spreading as a function of time in the neuron morphology, where dendrites are pointing in various radial directions. For the MS neuron, ϕ_0 approaches 3.26, while ϕ_0 goes to 3.76 radians for the Hay neuron.

Simulations for the same morphologies with different synapse positions gave similar results. Thus, current dipole moments from neuron simulations with anatomically reconstructed morphologies can be calculated with the TC and the AC methods. From this, it is also clear that axial currents from a neuron can be computed by using the code for axial currents that the AC method is based on. Now that the methods for calculating current dipole moments have been validated, extracellular potentials can be estimated using the DB model as well as the CB model.

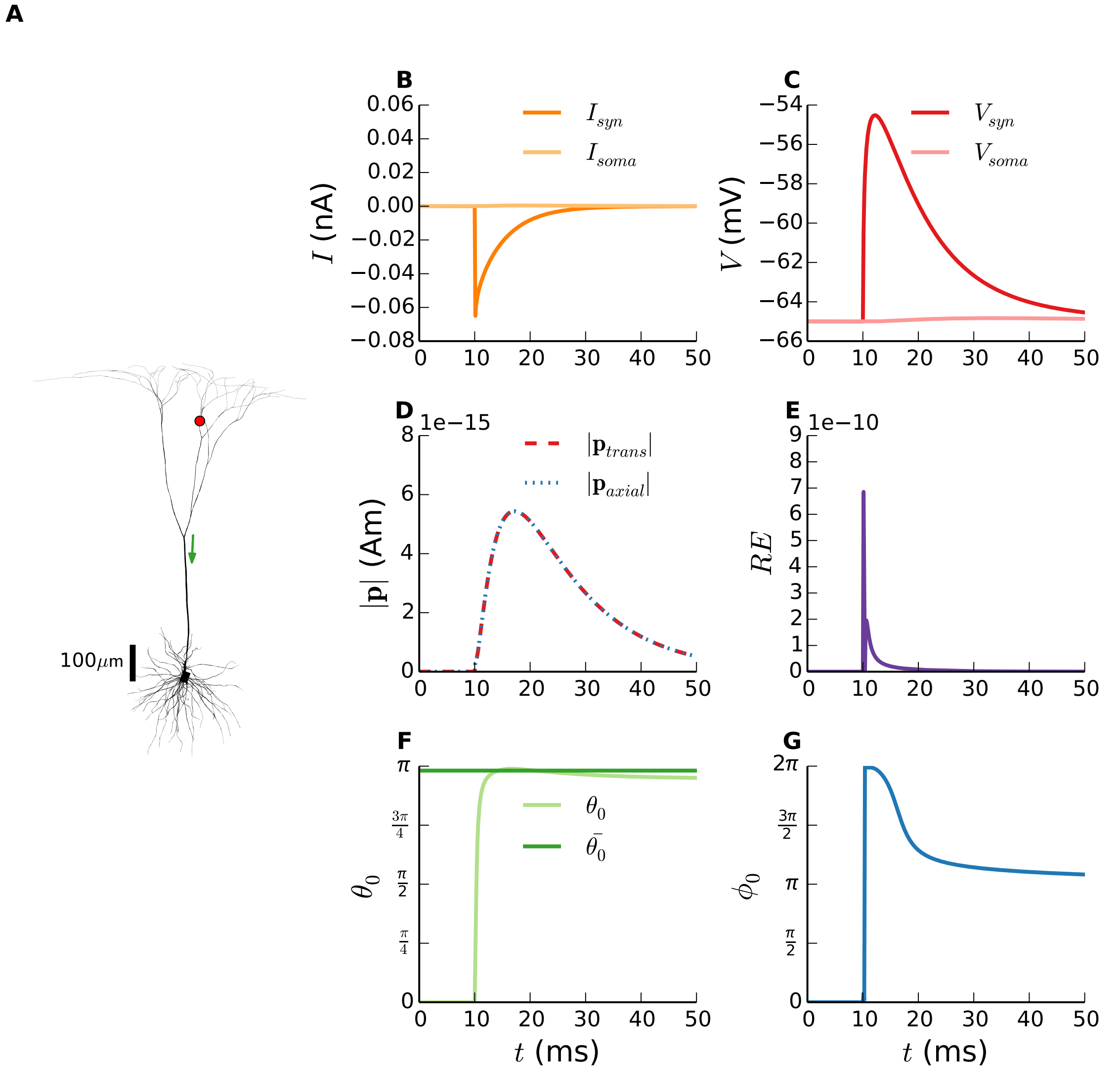

Figure 4.3: Current dipole moment from MS neuron. **A:** Neuron morphology with a red dot representing an excitatory, exponential, conductance-based synapse activated at time $t_s = 10$ ms. For morphology details and passive membrane properties, see Section 3.3 and Table 3.1. The orientation of the current dipole moment resulting from the simulation is shown as a green arrow. **B:** Transmembrane currents measured in the synaptic and the somatic compartments. **C:** Membrane potential recordings from the synapse compartment, $V_{syn}(t)$ and the soma compartment, $V_{soma}(t)$. **D:** Magnitude of current dipole moment, $|\mathbf{p}_{trans}(t)|$ and $|\mathbf{p}_{axial}(t)|$, computed with the TC method and the AC method, respectively. **E:** The relative error, RE, between the current dipole moment calculations in panel **D**. **F:** Angle between the current dipole moment and the z-axis, $\theta_0(t)$, and $\bar{\theta_0}$ averaged over $t > t_s$. **G:** Angle between radial component of current dipole moment and the x-axis, ϕ_0.

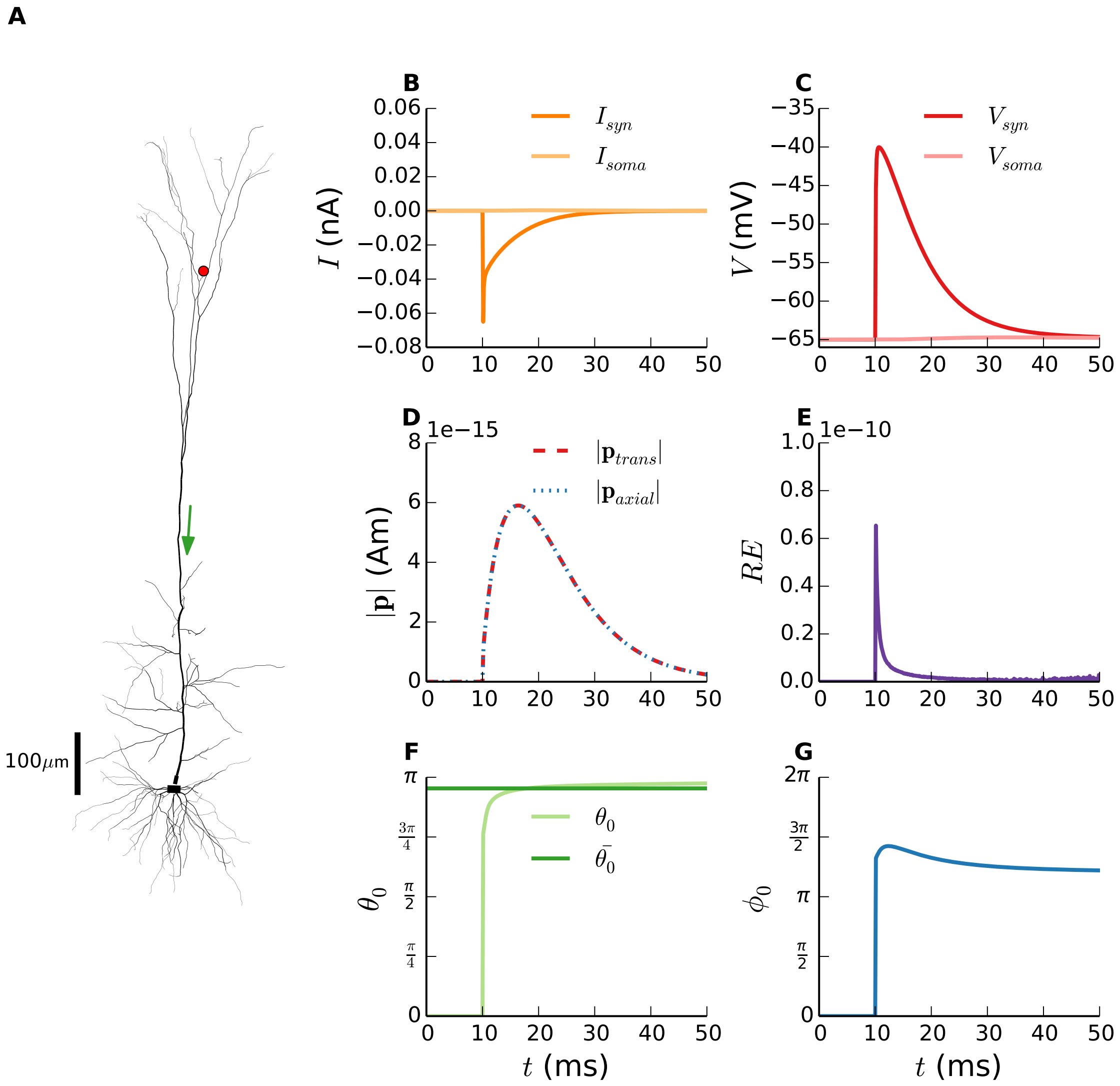

Figure 4.4: Current dipole moment from Hay neuron. **A:** Hay neuron morphology with an excitatory synapse placed on an apical dendrite, illustrated by the red dot. At time $t_s = 10$ ms, an exponential, conductance-based synaptic input current is initiated, and the resulting current dipole moment is illustrated by the green arrow. **B:** Transmembrane currents, $I_{syn}(t)$ and $I_{soma}(t)$, recorded in the synaptic and the somatic compartments, respectively. **C:** Membrane potentials measured in the soma compartment, $V_{soma}(t)$, and the synapse compartment, $V_{syn}(t)$. **D:** The magnitude of the current dipole moments, $|\mathbf{p}_{trans}(t)|$ and $|\mathbf{p}_{axial}(t)|$, computed with the TC and the AC methods. **E:** The relative error, RE, between the current dipole moment calculations in panel **D**. **F:** The angle, $\theta_0(t)$ between the z-axis and the current dipole moment vector. The average angle over time $t > t_s$, $\bar{\theta_0}$ is also shown. **G:** The angle between the radial component of the current dipole moment and the z-axis, ϕ_0.

4.2 Extracellular Potentials

In this section extracellular potentials are measured by virtual electrodes positioned at different locations relative to single neurons. The virtual recordings are carried out using the CB and the DB models. The CB model is here representing the ground truth, and the recordings are meant to test whether the DB model can predict extracellular potentials measured at ECoG and EEG planes.

4.2.1 Single Synapse with Static Location

The extracellular potentials from a simulation of the MS cell, see Section 3.3, were calculated using both the CB and the DB models. The neuron simulation lasted 100 ms, and an excitatory synapse placed on an apical dendrite generated an exponential synaptic input current, see Section 3.4, after 10 ms. The resulting extracellular potentials were calculated for different placements of virtual electrodes relative to the neuron.

Vertical Cross Section

A 31 × 31 grid of virtual point electrodes was positioned in a 2 by 2 cm vertical cross section of cortex and skull. Further, an MS neuron was placed in cortex, so that the soma was located in the origin, $(0, 0, 0)$, corresponding to the center of the virtual electrode grid. The cell morphology elongates in the z-direction, such that the top of the cell was located 100 μm below the cortex-skull interface.

Extracellular potentials were calculated with both the CB and the DB models and plotted together with absolute and relative errors in Figure 4.5. Both extracellular potentials and error plots are shown for time point $t = t_{pmax}$, when the current dipole moment is at its strongest.

A few things are important to note from studying the extracellular potentials in panels B and C, regarding the current dipole approximation. First of all, the figure-8 shapes of the electric potentials are very similar to the characteristic shape of a dipole-field, similar to the field around an antenna. The strength of the potentials is higher close to the neurons in the middle of the plot, and decreases with the distance to the measuring point, r. Moreover, the shapes of the plots in B and C are very similar. From this it appears that current dipoles are well suited for approximating electric potentials far away from a neuron. The green arrow in panel A illustrates the current dipole moment vector pointing down from the apical dendrites towards the soma. This is also reflected in the electric potential plots, where the upper half is negative, because the apical dendrites with the excitatory synapse act like a current sink. Further, currents are escaping through the lower half of the cell, turning this part into a current source generating a positive potential, red on the plot.

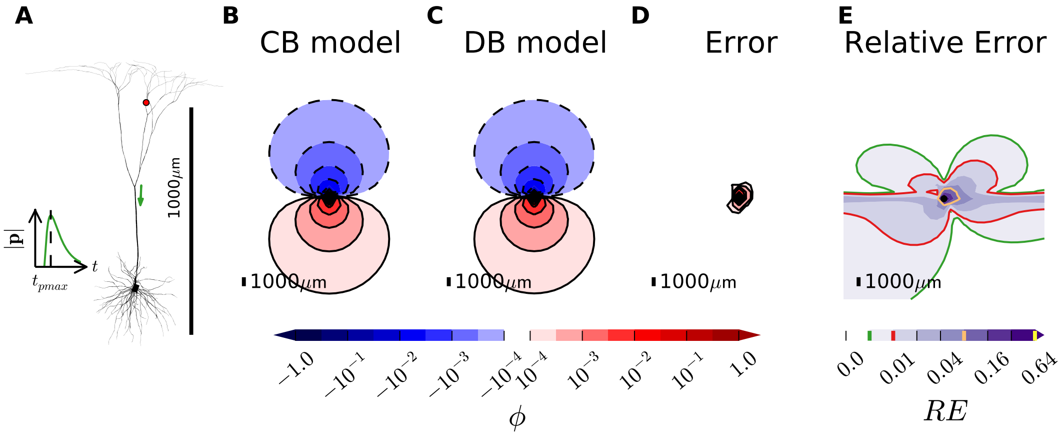

Figure 4.5: Contribution to extracellular potentials measured in vertical cross section from MS neuron. **A:** MS neuron morphology, see Section 3.3, shown with the red dot illustrating an excitatory synapse. An exponential, conductance-based input current was generated at $t_s = 10$ ms, and the magnitude of resulting current dipole moment, $\mathbf{p}(t)$, is shown in the plot on the left, and peaks at time $t = t_{pmax}$. The orientation of the current dipole vector at this time point is illustrated by the green arrow. All the remaining panels show results for time point $t = t_{pmax}$ measured by a 2×2 cm grid of electrodes placed in an xz-plane, where the MS neuron is located in the grid's center. **B:** Extracellular potentials calculated with the CB model. The scale is $\log_{10}$-based and normalized, such that 1.0 corresponds to 39 nV. **C:** Extracellular potentials calculated with the DB model. **E:** The absolute error between the extracellular potential calculations from the two different models in panels **B** and **C**. The relative error, plotted on a base-2 logarithmic scale. The green, red and orange lines mark the 0.005, 0.01 and 0.08 contours.

Even though the extracellular potentials look equal from the 2 by 2 cm plots in panels B and C, calculating the absolute and the relative error, reveals the differences between the two models. The absolute error in panel D is less than 1/10000 of the maximal potential measured, except within a radius, of $\sim 4 - 5$ mm, of the neuron. In comparison, the size of the MS neuron is approximately 1 mm.

The relative error is plotted in Figure 4.5E. Except from within a very small area with diameter ~ 3 mm around the neuron, the relative error is less than 0.08, corresponding to 8 %. The relative error plot is almost butterfly shaped, spreading out more in the xy-direction than in the z-direction. Moving the virtual electrode horizontally away from the neuron, the relative error stays between 0.02 and 0.04 even tens of mm away. This is the region where the extracellular potential is the lowest, and crosses zero. Moving in the z-direction away from the xy-plane, the relative error decay seems to depend on θ_0, the angle between the current dipole moment, $\mathbf{p}$, and the z-axis. The relative error appears to decrease rapidly directly above and below the neuron, i.e., when the measuring point is close to the z-axis. When increasing the angle θ_0, however, the relative error experiences a slower decay. This may partly be caused by the dipole-shaped extracellular potentials; the extracellular potential is lower when the angle θ_0 is big. The effect wares off when $\theta_0 \to \pi$, which is the radial direction where also the decrease in electric potential is the lowest.

Drawing a 4 mm radius circle around the neuron, we see from panel E, that the relative error is less than 0.02 outside of this region. Here, the extracellular potential is less than 1/10000 of its maximum value. In short, the DB model has an error below 2% when the distance between the neuron and the measuring point is more than 4 mm. Similar conclusions can be drawn from measurements done at other time points t in the simulation.

ECoG Plane

The objective of this section was first of all to test the accuracy of the DB model at the top of cortex, i.e., to check whether the dipole approximation is appropriate for estimating ECoG signals. Secondly, the shape and strength of the electric potentials were studied. Following the same procedure as in the previous section, a grid of virtual electrodes was placed in a 2 by 2 cm area, this time in an xy-plane 100 μm above the top of the MS neuron, symbolizing an ECoG recording. The simulation was run, and extracellular potentials calculated. The neuron morphology, the current dipole moment from the simulation, and the electrodes are shown in Figure 4.6A.

The extracellular potentials calculated with the CB and the DB model are illustrated in Figures 4.6B and C respectively. As expected, based on the previous section, the extracellular potentials are negative and decrease radially with distance from the center of the plot, i.e., the grid point located directly above the neuron. Studying panels B and C, the potential plots appear to be very alike. Further, the absolute errors were calculated and plotted in panel D, revealing that the difference between the measurements from the CB and the DB model is less than $\frac{1}{10000}$ of the maximal potential measured, as long as the electrode is more than 5 mm away from the point directly above the neuron. Since the signals decrease rapidly with distance from the middle of the grid, the signal strength is very small at measuring points where the absolute error is small. Hence, it makes more sense to study the relative error to get an understanding of the accuracy of the dipole method.

The relative error was calculated for all electrodes in the grid, and is shown in Figure 4.6E. Studying the plot, we see that the relative error varies significantly with electrode position, and that for points near the grid center and for small values of y,

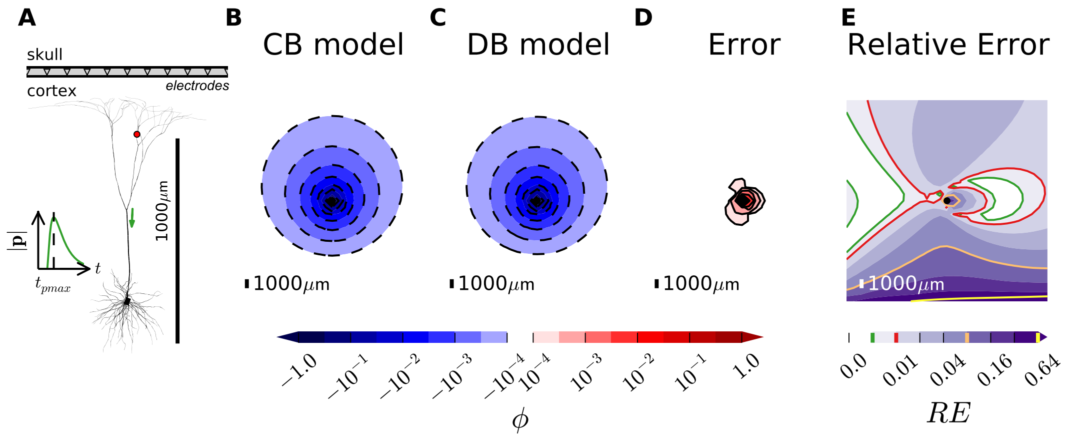

Figure 4.6: ECoG contribution from MS neuron. **A:** The virtual experiment set-up consists of the MS neuron plotted in the xz-plane with an conductance-based excitatory synapse placed on an apical dendrite. See 3.3 for more details about the neuron model. A 100 ms simulation, with a single synaptic input after 10 ms, was run, and the magnitude of the resulting current dipole moment, $|\mathbf{p}(t)|$, is plotted in the small graph on the left. Here, t_{pmax} is the simulation time point where $|\mathbf{p}(t)|$ peaks. The orientation of $\mathbf{p}(t_{pmax})$ in the xz-plane is illustrated by the green arrow. All values in panels **B-E** are shown for time $t = t_{pmax}$. The gray bar on the top illustrates the grid of ECoG-electrodes on top of cortex, only a 100 μm away from the top of the neuron. There are 31×31 point electrodes distributed in a 2 by 2 cm area in the xy-plane. The center of the electrode grid is the grid point straight above the soma, marked by a black dot in panel **B-E**. **B:** Extracellular potentials calculated with the CB model. The scale is base-10 logarithmic, and 1.0 on the scale corresponds to 4.0 nV. **C:** Extracellular potentials calculated with the DB model. **D:** Absolute errors, calculated to show the difference between the results from the CB and the DB models. **E:** The relative error is plotted with a base-2 log scale. The green, red, orange and yellow lines correspond to the $0.005, 0.01, 0.08$ and 0.64 contours.

the relative error is big, i.e., > 0.08. From these measurements it seems as though ECoG recordings are too close to the neuron to be predicted with the DB model.

EEG Plane

Next, it was tested whether the DB model is applicable for modeling of EEG signals. The same simulation and virtual experiment set-up as in the previous section were used. However, since the thickness of the human skull and scalp combined, is approximately 1.1 cm [33], the electrode grid was now placed in the xy-plane at z

$\sim$ 1.2 cm. This way, the virtual measurements symbolized EEG recordings. The soma of the MS neuron was now located 1.2 cm below the center of the electrode grid, and the top of the cell was still $100\mu m$ below the cortex-skull interface.

Extracellular potentials were calculated with the CB and the DB models for all electrodes in the grid, on top of the scalp. The virtual recording set-up, the cell morphology and $|\mathbf{p}(t)|$ are shown in Figure 4.7A. All plots, except panel F shows time point $t = t_{pmax}$, i.e., the time when the current dipole moment is at its strongest. The extracellular potentials are plotted in panels B and C, and show that the electric potentials calculated were, like the ECoG signals, still negative. The shape of the electric potential plot is circular for both models. The potential is strong close to the center of the grid and decreases radially for electrodes farther away from the center. The decrease is, however, not as rapid as for the ECoG measurements: note that the scale is now linear and not $\log_{10}$-based. Comparing extracellular potentials from the two models, it seems as though the DB model gives a good approximation.

The absolute error is plotted in panel D and at first sight; it appears that the error is large, having the error plots from the previous sections in mind. Studying the scales, however, we see that this is only a trick, due to the fact that the extracellular potentials vary a lot less with position in the grid electrode plane, than for the ECoG recordings. To get an impression of whether the current dipole approximation is accurate for EEG recordings, we need to study the relative error. The relative error shown in panel E is less than 0.005 for all virtual electrodes, except for the ones in a small region to the right of the middle point. The relative root mean square error, ($RRMSE$) over all time points, was calculated and plotted in panel F. The $RRMSE$ does never exceed 0.01. With this, it seems like the DB model can predict EEG signals from single cells very accurately.

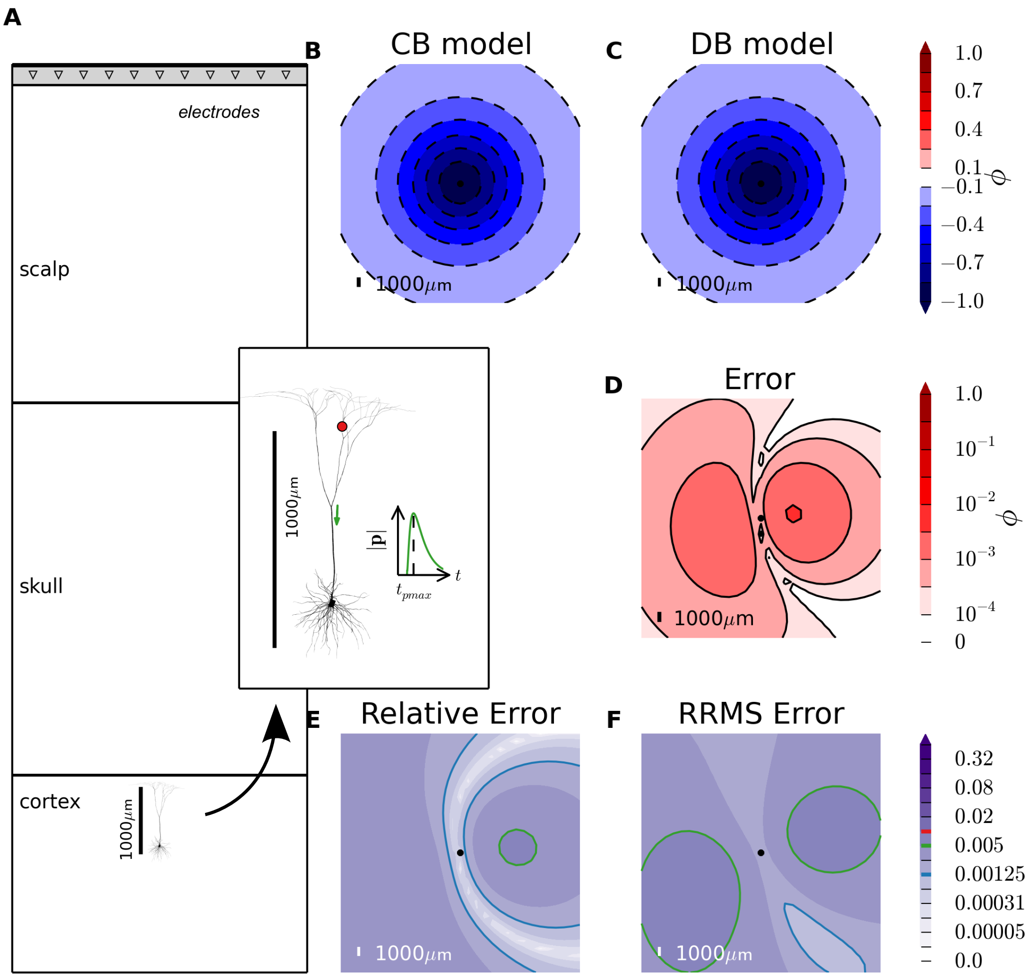

Figure 4.7: EEG contributions from MS neuron with one apical synaptic input. **A:** The virtual recording set-up is illustrated in the xz-plane. The skull and scalp thickness is here 1.1 cm and the top of the MS neuron is located $\sim 100\ \mu$m below the top of neocortex. The soma is located in the origin, 1.2 cm below the electrode grid on top of the scalp, where 31×31 electrodes are distributed in a 2×2 cm area in the xy-plane. A conductance-based, excitatory synapse is placed on an apical dendrite, marked by the red dot, and a synaptic input arrives after 10 ms. The resulting current dipole moment for the neuron is calculated, and the magnitude, $|\mathbf{p}(t)|$ is plotted in the small panel on the right. $|\mathbf{p}(t)|$ peaks at $t = t_{pmax}$, and all panels, except panel **F**, show data calculated at t_{pmax}. The orientation of $\mathbf{p}(t_{pmax})$ in the xz-plane is illustrated by the green arrow. **B:** Extracellular potentials calculated with the CB model, plotted on a normalized linear scale, where 1.0 corresponds to 0.011 nV. **C:** Extracellular potentials computed with the DB model. **D:** Absolute errors, showing the difference between the results in panels **B** and **C**, plotted on a base-10 logarithmic scale. **E:** Relative error plotted with a log_2-scale. The green line represents the 0.005 error contour, and the blue line corresponds to 0.00125. **F:** Relative root mean square (RRMS) error over the entire time signal.

4.2.2 Single Synapse with Varying Location

To assure that the results from Section 4.2 are applicable also for other morphologies and synapse locations, the ECoG and EEG calculations from the preceding section were carried out for the MS neuron and the Hay neuron for various locations of synaptic input. For each cell, 24 random synaptic input positions were used, and the same 100 ms simulation with one excitatory, exponential synaptic input after 10 ms, as in the previous sections, was applied for all trials. Instead of showing the whole electrode grid, only the results for the electrode at the center of each grid are plotted. The synapse locations are here represented by two coordinates relative to the soma: ρ_{syn}, the radial distance from the soma to the synapse, and z_{syn}, the vertical distance from the soma to the synapse.

ECoG Point

The ECoG signals were calculated with the CB and the DB models for a point electrode located 100 μm straight above the top of neuron. The results for the MS neuron and the Hay neuron are shown in Figures 4.8 and 4.9, respectively. The virtual experiment set-up is illustrated in the left panel on the figure. For both cells, 24 simulations were run for 24 different synapse locations, giving 24 pairs of measured potentials for the two different models. In order to check how the extracellular potential measurements change with synapse location, the amplitude of the measured potentials calculated with the CB model, was plotted as functions of ρ_{syn} and z_{syn}. Studying the ρ_{syn}-dependence shown in panel A in Figures 4.8 and 4.9, no clear trend with ρ is visible. For the z_{syn}-dependence in panel B, on the other hand, extracellular potentials appear to increase significantly with z_{syn} for $z_{syn} > 800$ μm.

To compare the dipole approximation to the ground truth, absolute errors (E), relative root mean square errors with respect to time ($RRMSE$), and relative errors (RE) were calculated and plotted as functions of ρ_{syn} and z_{syn}. Studying panels C-F, it is clear that the $RRMSE$ and the RE measurements are very high for both the MS and the Hay morphologies, i.e., there are relative errors > 1 for both cells. The absolute errors are also overall high, with no apparent trends with respect to synapse location, see panels G and H.

When it comes to synapse location dependency for relative errors and $RRMSE$, it does not seem like the errors are correlated with ρ_{syn}, apart from the observation that higher relative error values seem to occur when ρ_{syn} is close to zero. This is more likely explained by the neuron morphology, and the fact that with respect to z_{syn}, the relative error seems to go up for values corresponding to the middle height of the neuron. For these z-values, the synapse will be located at the thick apical dendrite, where ρ_{syn} is restricted to values close to 0. When an excitatory synapse is placed at the middle height of a neuron, current will spread relatively equally upwards to the apical dendrites and down towards the soma and basal dendrites. Thus, the electric potentials will be negative around the middle of the cell, and positive in the areas near the top and the bottom of the cell. In this case, the neuron acts like a quadrupole. The dipole contribution to the electric potential is hence very small, and the dipole approximation is not a good approximation. From Section 2.3 we know that the quadrupole contribution to the extracellular potential falls off as $1/r^3$, i.e., much faster than the dipole contribution. Hence, the extracellular potentials are low for these z-values.

From this it is clear that our conclusion about dipole approximation for ECoG-signals in Section 4.2.1 still holds. Current dipoles are hence not useful for estimating extracellular potentials measured on top of cortex.

EEG Point

The same experiments as described in the previous section were conducted for EEG signals and illustrated in Figures 4.10 and 4.11. The recording set-up is shown in the panel on the left. An extracranial point electrode was now placed a distance 1.1 cm above the top of the neuron, i.e., on top of the scalp. The exact potentials recorded were calculated for both cells and studied in figure panels A and B. The strength of the EEG signal does not seem to be correlated with the radial synapse distance from the soma, ρ_{syn}. On the contrary, the electric potentials seem to go down for z-values in the range $\sim [0, 300]$ μm.

Extracellular potentials were also calculated using the DB model, and compared to the CB model by looking at absolute errors, relative RMS errors with respect to time and relative errors. The absolute errors seem to be higher for synapse locations near the middle of the neuron, than top or bottom locations.

For both cells, the relative errors and relative RMS errors are much lower than for the ECoG recordings. The majority of the synapse locations give relative RMS and relative errors < 0.05. The two error types follow very similar patterns. The errors do not seem to be correlated with ρ_{syn} at all. Like in the previous example, there are no very high relative errors for large values of ρ. We can not conclude anything from this, because of the low number of measurements with big values for ρ_{syn}. Studying the z_{syn} dependency of the relative error reveals that the higher relative errors > 0.05 occur when the extracellular potential is low. To conclude, the DB model can be used to predict EEG contributions from single cells with one synapse, however, the accuracy may exceed 5% for extracellular potentials with small amplitudes.

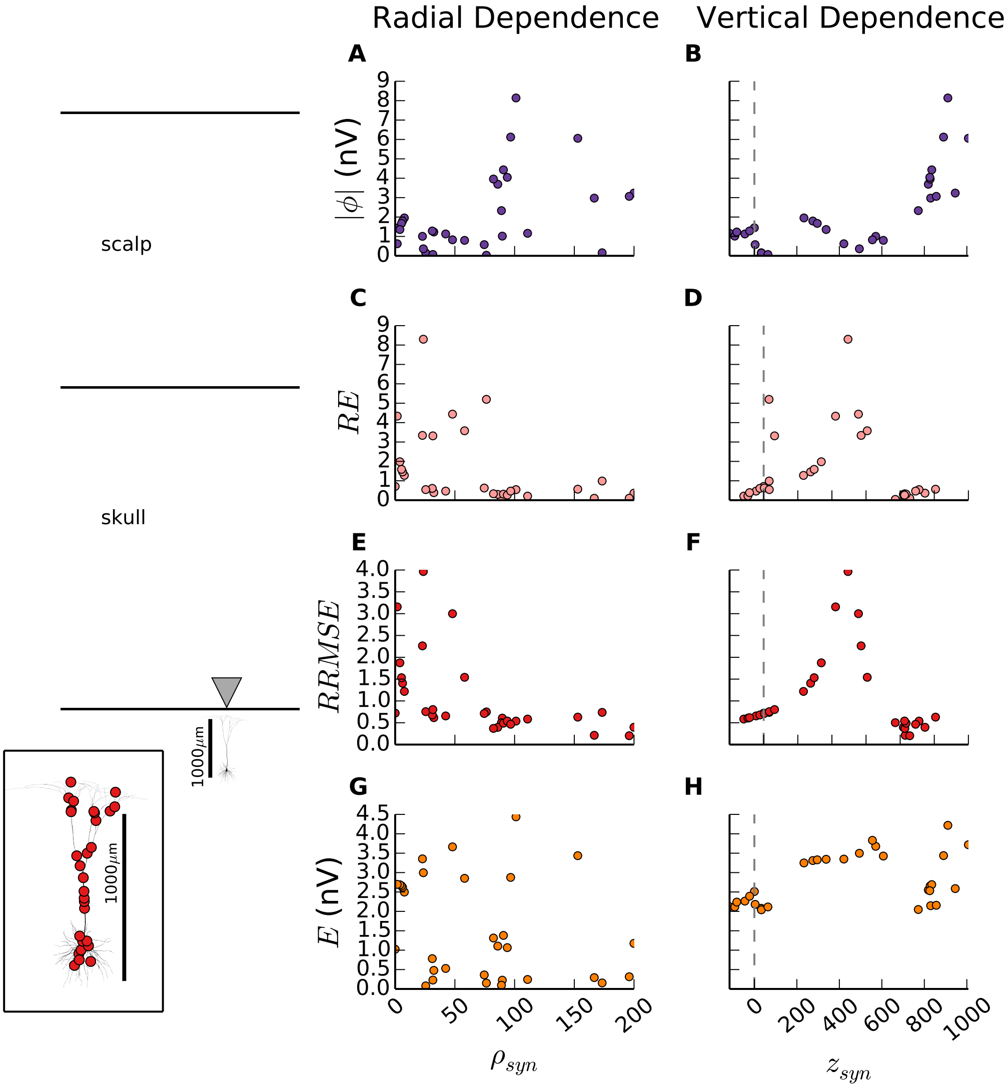

Figure 4.8: ECoG contributions from MS neuron with different synapse locations. Same virtual recording set-up as in 4.6, however, instead of a whole plane of electrodes, there is only one electrode recording from 100 μm straight above the neuron. The 24 synapse locations are shown on the left. One 100 ms simulation was run for each synapse location, the synapses all being conductance based and excitatory, with an input initiated after 10 ms. Synapse location is expressed with two coordinates: vertical distance from the soma, i.e., the xy-plane, z_{syn} and radial distance from the soma, i.e., distance from the z-axis, ρ_{syn}. **A:** Extracellular potentials measured by the ECoG electrode is plotted as a function of ρ_{syn}. **B:** The potential dependence on z_{syn}. **C,D:** Relative errors, RE, plotted as functions of ρ_{syn} and z_{syn}. **E,F:** Relative root mean square errors, $RRMSE$. **G,H:** Absolute errors, E, as functions of z_{syn} and ρ_{syn}.

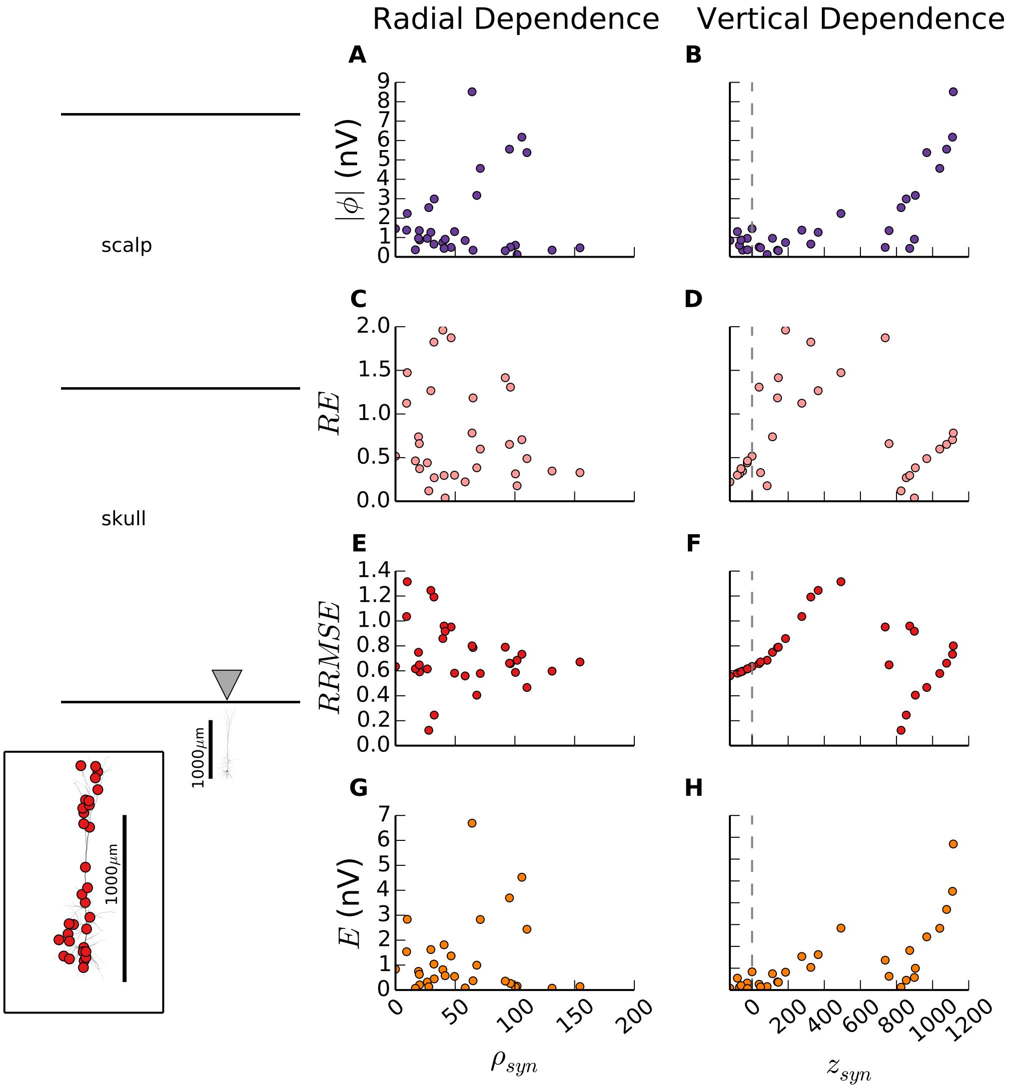

Figure 4.9: ECoG contributions from Hay neuron with different synapse locations. A point electrode records extracellular potentials a distance 100 μm straight above the neuron. One simulation is run for each of the 24 synapse locations shown on the left. All synapses are conductance based and excitatory, initiating an input current after 10 ms. Synapse location is described by two coordinates: z_{syn}, the vertical distance from the soma, i.e., the xy-plane, and ρ_{syn}, the radial distance from the soma, i.e., distance from the z-axis. **A:** Extracellular potentials measured by the ECoG electrode plotted as a function of ρ_{syn}. **B:** The potential dependence on z_{syn}. **C,D:** Relative errors, RE, plotted as functions of ρ_{syn} and z_{syn}. **E,F:** Relative root mean square errors, $RRMSE$. **G,H:** Absolute errors, E, shown as functions of z_{syn} and ρ_{syn}.

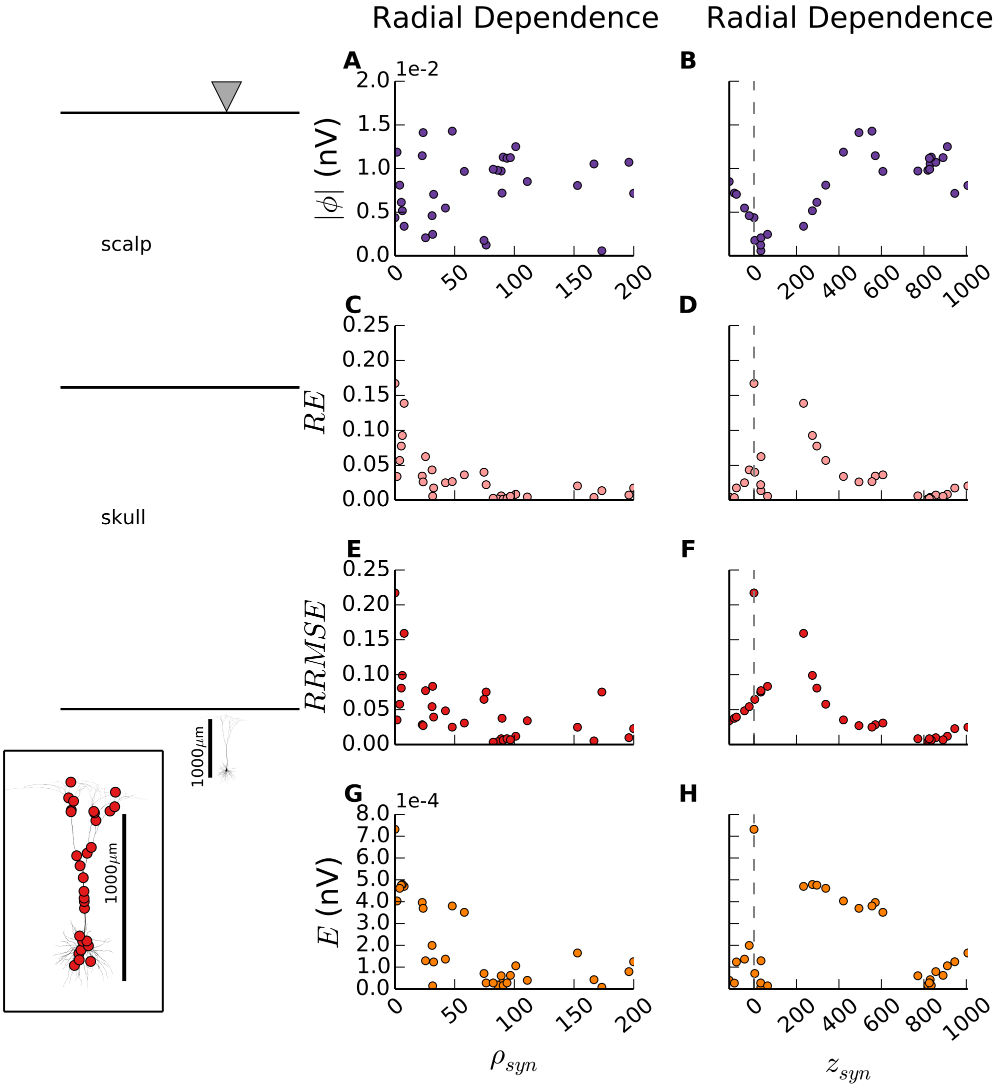

Figure 4.10: EEG contributions from MS neuron with different synapse locations. Virtual recording set-up with one point electrode on top of the scalp, directly above the MS neuron, 1.2 cm above the soma. The 24 red dots represent 24 randomly generated synapse locations. One 100 ms simulation was run for each synapse. All synapses are excitatory and conductance based, initiating an input current after 10 ms. Synapse locations are given by ρ_{syn}, the radial distance from the soma and z_{syn}, the vertical distance from soma. **A:** Extracellular potentials calculated with the CB model, measured by the EEG electrode and plotted as a function of ρ_{syn}. **B:** Extracellular potential dependence on z_{syn}. **C,D:** Relative errors, RE, as functions of ρ_{syn} and z_{syn}. **E,F:** Relative root mean square errors, $RRMSE$, plotted against synapse location. **G,H:** Absolute errors plotted as functions of synapse locations, E(ρ_{syn}) and E(z_{syn}).

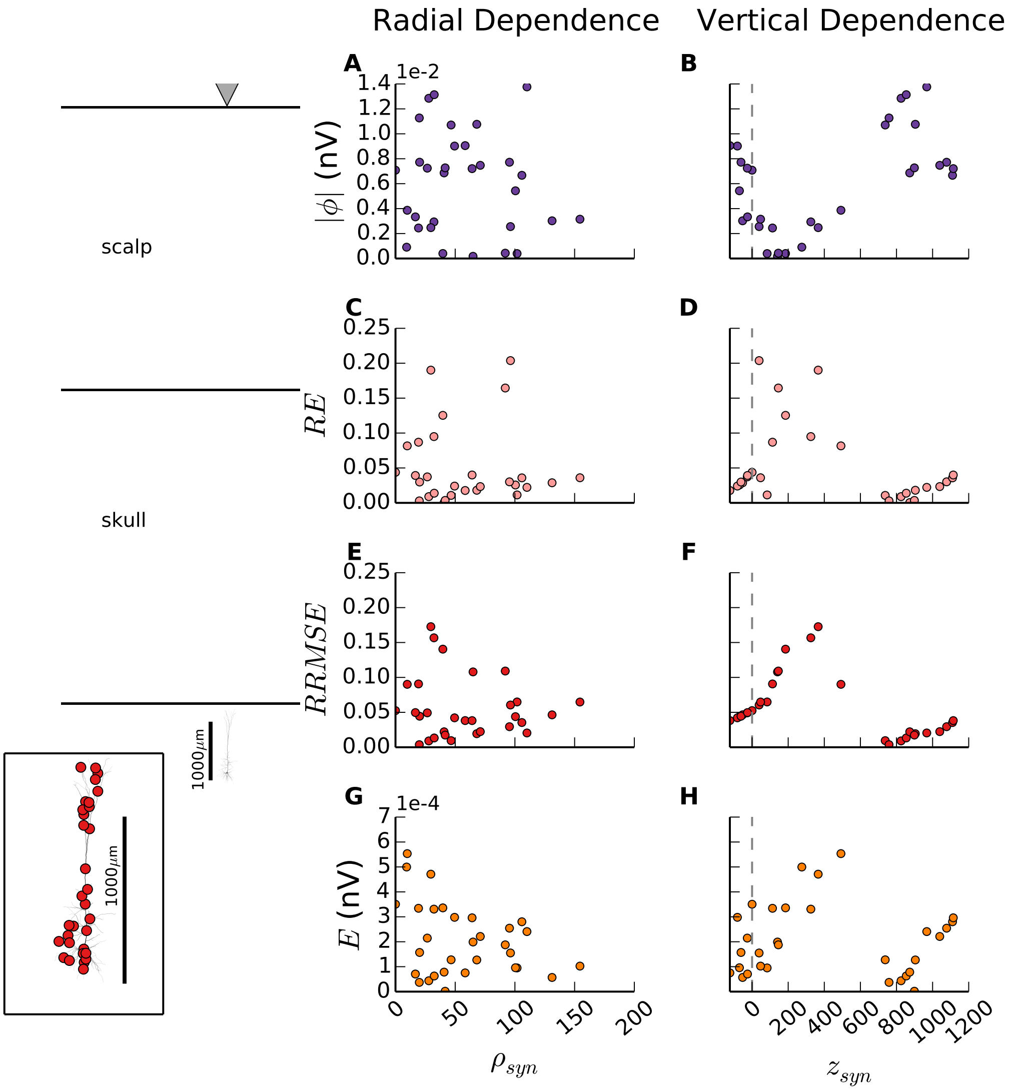

Figure 4.11: EEG contributions from Hay neuron with different synapse locations. The left panel shows the virtual recording set-up with a point electrode on top of the scalp, at $z \approx 1.2$ cm, straight above the neuron. The small lower left panel shows the Hay neuron morphology with 24 randomly generated synapse locations, red dots. Simulations lasting 100 ms were carried out for all synapses, with a conductance-based, excitatory input current initiated after 10 ms. Synapse locations are given by ρ_{syn} and z_{syn}, the radial and vertical distances from the soma, respectively. **A:** Extracellular potentials calculated with the CB model, measured by the EEG electrode and plotted as a function of ρ_{syn}. **B:** The extracellular potential dependence on z_{syn}. **C,D:** Relative errors, RE, as functions of ρ_{syn} and z_{syn}. **E,F:** Relative root mean square errors, $RRMSE$, plotted against synapse location. **G,H:** Absolute errors plotted as functions of synapse locations, E(ρ_{syn}) and E(z_{syn}).

4.2.3 More Synapses per Neuron

In contrast to the preceding virtual neuron simulations, a pyramidal neuron can receive input from thousands of synapses distributed all over the neuron morphology [40, 8]. In order to study how a neuron responds to more than one synaptic input, three simulations with different distributions of 100 excitatory synapses were tested on the MS cell. The synapses were randomly distributed on apical dendrites, basal dendrites, and finally all over the cell. All synapses were driven by a stationary Poisson input, see Section 3.4, over a 400 ms interval with an average firing rate of 5 spikes. The virtual recording set-up was exactly the same as in Section 4.2.1, with the top of the MS cell placed $\sim$ 1.1 cm below the center of an electrode grid on top of the scalp, see schematic illustration in Figure 4.7A.

Apical Synapse Positions

The MS cell had 100 synapses randomly distributed on apical dendrites, i.e., on segments located above the $z = 850$ μm -plane. The morphology and synapses are shown in panel A on Figure 4.12, together with the amplitude of the current dipole moment. The rapid variations in $|\mathbf{p}(t)|$ is due to the spiking of the many synapses at different points in time. The time point $t = t_{pmax}$ marks when the current dipole moment amplitude is at its strongest, and the plots on the right all show data for this time point.

Extracellular potentials were calculated with the CB and the DB model, and can be studied in panels B and C. All in all, the results appear to be very similar for the two models. The absolute error is plotted on a log_{10}-based scale in panel D, and E shows the relative error. The relative error is overall very small, with a maximum less than 0.02. Similar results were generated for different points in time. We can conclude that the DB model can very well predict extracellular potentials from a neuron simulation with many synapses placed on apical dendrites.

Basal Synapse Positions

To check what the extracellular potentials looked like for 100 basal dendrites, synapses were randomly spread out on segments of the MS neuron, located below the $z = 0$ -plane. The results are shown in Figure 4.12. Extracellular potentials, absolute errors and relative errors were plotted for time point $t = t_{pmax}$. An important thing to note here, is that the scale for plot B and C is linear, whereas the absolute error is plotted with a base-10 logarithmic scale. The absolute error is thus not as big as it looks. The maximum relative error is < 0.04, with the biggest error measured straight above the neuron. The EEG contribution from pyramidal neurons with 100 excitatory synapses on basal dendrites can be modeled with 96% accuracy using the DB model.

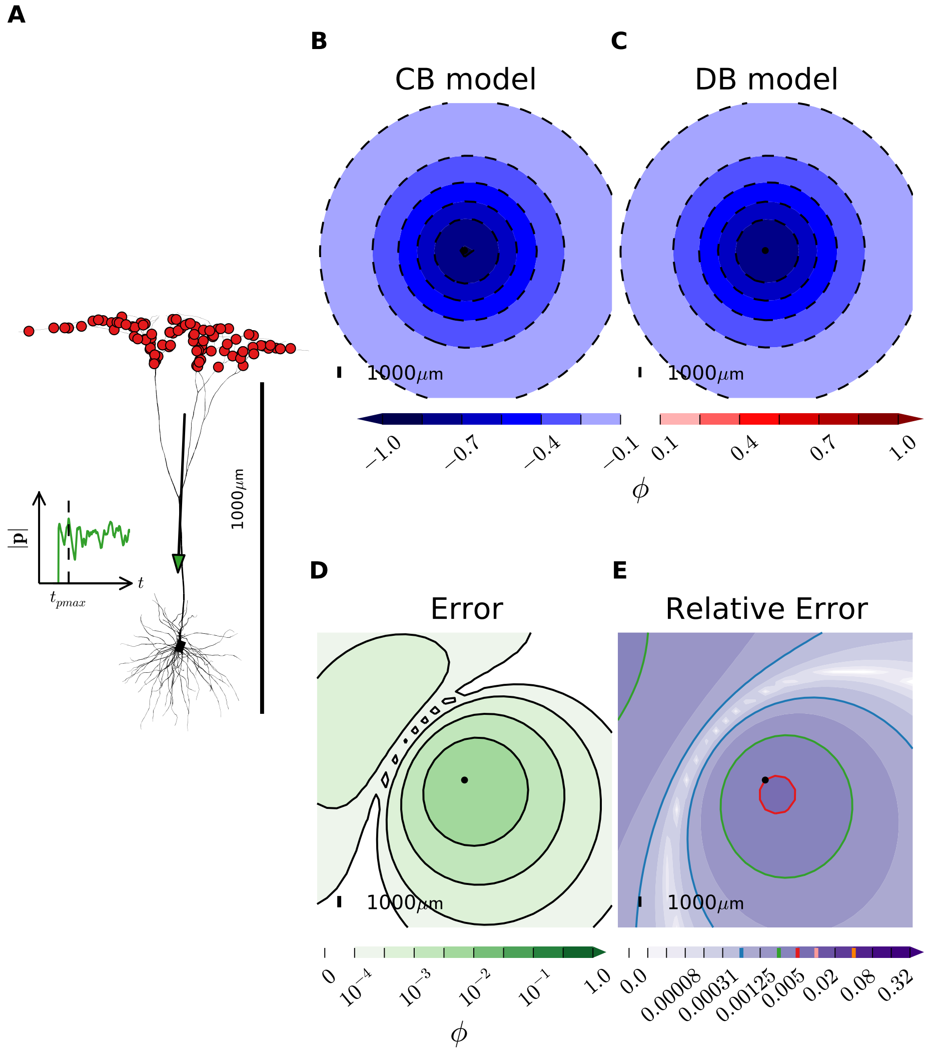

Figure 4.12: EEG contribution from MS neuron with 100 apical synapses. **A:** MS neuron with 100 excitatory synapses randomly spread out on the apical dendrites, i.e., on segments positioned at $z > 850$ μm. Current-based stationary Poisson synaptic inputs, see Section 3.4, were used for the 400 ms simulation. The resulting current dipole moment, $\mathbf{p}$, is a function of time, peaking at time $t = t_{pmax}$. The magnitude, $|\mathbf{p}(t)|$ and the spacial orientation of $\mathbf{p}(t_{pmax})$ are illustrated by the small graph on the left and the green arrow on the morphology, respectively. The remaining plots all show data for time point $t = t_{pmax}$ in a 2×2 cm grid of virtual electrodes, a distance ~ 1.2 cm above the top of the neuron, representing an EEG recording. The black dot in the middle of each plot marks the point straight above the neuron. **B:** Extracellular potentials calculated with the CB model. **C:** Extracellular potentials calculated with the DB model. The scale used for panels **B** and **C** is linear and normalized, such that 1.0 corresponds to 0.003nV. **D:** The difference between the CB and the DB models illustrated by the absolute error on a $\log_{10}$-based scale, where 1.0 represents 0.003nV. **E:** Relative error shown on a base-2 log scale. The blue, green and red lines correspond to the contours marking a relative error of 0.00125, 0.005 and 0.01, showing that the highest relative error is less than 0.02 and can be seen inside the red contour.

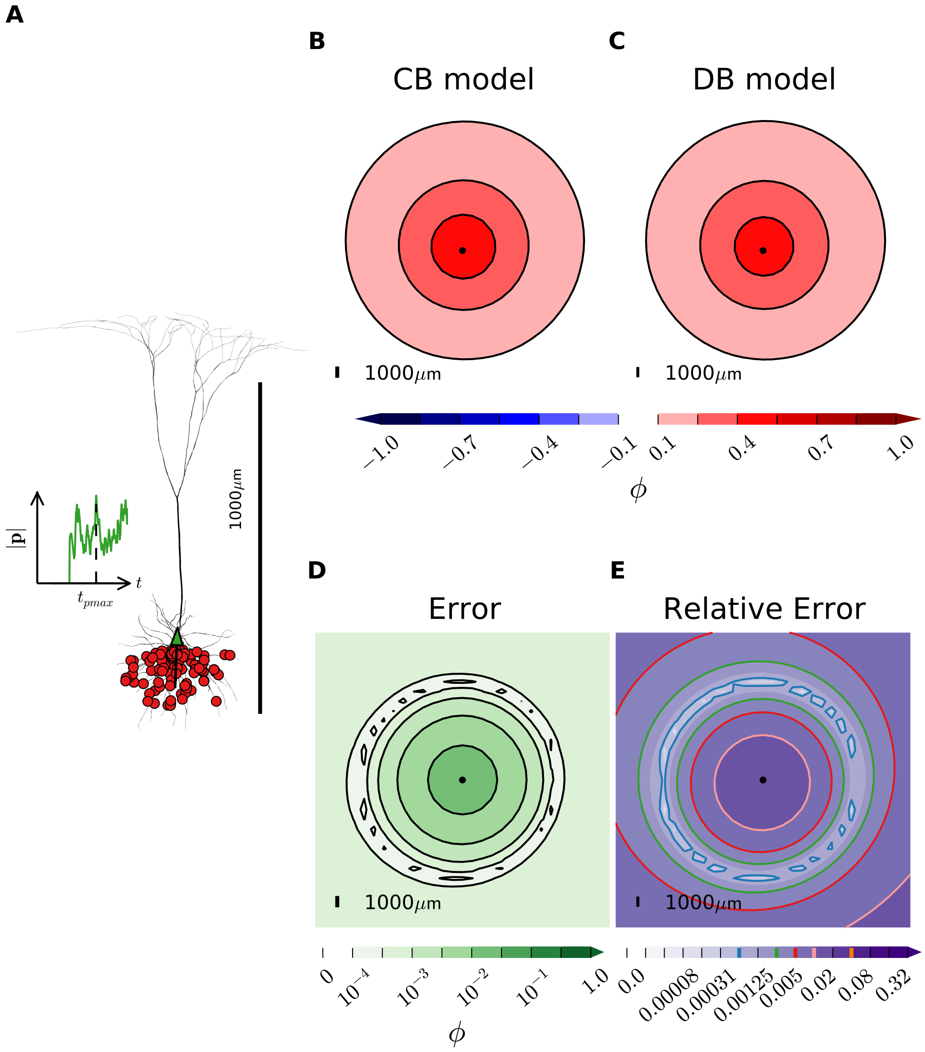

Figure 4.13: EEG contribution from MS neuron with 100 basal synapses. **A:** Neuron morphology of MS neuron with 100 excitatory synapses, represented by red dots, randomly distributed on the basal dendrites, i.e., on segments below $z = 0$. A 400 ms simulation with stationary Poisson synaptic input currents, as described in Section 3.4, gave the current dipole moment $\mathbf{p}(t)$, and the magnitude is shown in the green graph on the left. The current dipole moment reaches its maximum amplitude at time point $t = t_{pmax}$. The spatial orientation of $\mathbf{p}(t_{pmax})$ is illustrated by the green arrow near the soma. The remaining plots all show values for time point $t = t_{pmax}$ at the EEG plane, a 2 by 2 cm grid of virtual electrodes located a distance $\sim$ 12 mm above the top of the neuron. The black dots mark the center of the virtual electrode grid, i.e., the position directly above the neuron. **B:** Extracellular potentials calculated with the CB model as a function of space, i.e., each point represents the potential measured at a point in the EEG plane. The scale is linear and normalized, such that 1.0 corresponds to 0.003 nV. **C:** Extracellular potentials calculated with the DB model. **D:** Absolute error illustrating the difference between potentials calculated from the two models, shown on a base-10 log scale, where 1.0 equals 0.003 nV. **E:** Relative error plot with a $\log_2$-based scale. The blue, green, red and pink contours illustrate 0.00125, 0.005, 0.01 and 0.02 relative error borders. The maximum relative error calculated is less than 0.04.

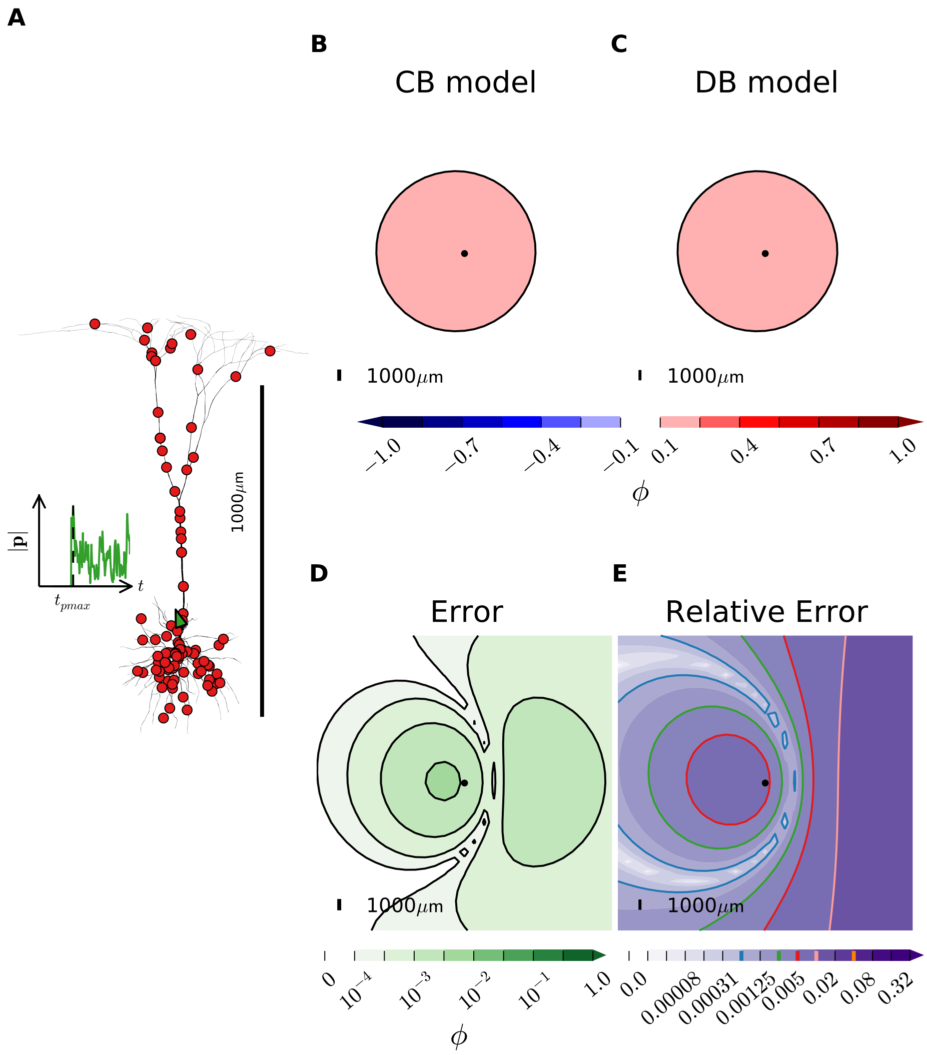

Figure 4.14: EEG contributions from MS neuron with 100 synapses distributed all over the morphology. **A:** An MS neuron with 100 excitatory synapses with Poisson input currents, see Section 3.4, randomly distributed all over the morphology. The magnitude of resulting current dipole moment, $\mathbf{p}(t)$, shown in the plot on the left, peaks at time $t = t_{pmax}$. The spatial orientation of $\mathbf{p}(t_{pmax})$ is illustrated by the tiny green arrow above the soma. All the remaining plots show data for the time point $t = t_{pmax}$. The plots represent data recorded by 2×2 cm areas of 31×31 virtual electrodes located in the xy-plane at the top of the scalp, i.e., 12 mm above the top of the neuron. The black dot located in the origin, marks the point straight above the neuron. **B:** Extracellular potentials calculated by the CB model with a linear, normalized scale, where 1.0 corresponds to 0.003 nV. **C:** Extracellular potentials approximated with the DB model. **D:** The absolute error representing the difference between the calculations from the two different models, plotted with a base-10 log scale, on which 1.0 = 0.003 nV. The relative error shown on a $\log_2$-based scale, where the blue, green, red and pink lines correspond to the 0.00125, 0.005, 0.01 and 0.02 contours. The maximum relative error is ~ 0.04.

Randomly Spread-Out Synapse Positions

In this final experiment 100 excitatory synapses were randomly spread out all over the MS neuron. The synapse distribution on the neuron morphology in addition to the current dipole moment are illustrated in Figure 4.14. Extracellular potentials were calculated with the CB and the DB models and shown in panels B and C, respectively. The signal strength is in this case far lower than for the two preceding examples. This is most likely due to the current dipole moment being very small when the excitatory synapses are distributed all over the neuron. The absolute error is plotted with a $\log_{10}$-scale in panel D, and panel E shows the relative error plotted with a base-2 logarithmic scale. Except for the region on the right, the relative error does not exceed 2%. The maximum relative error measured is, however, less than 0.04. Similar results were found for equivalent simulations with different spike times.

Note that EEG electrodes are much bigger than point electrodes, with diameters up to 1 cm [31, 32]. To make the virtual experiments more EEG-like, we averaged the potentials from the DB and CB models over a disk with 1 cm radius at the scalp surface. The relative errors for the apical input, basal input and homogeneous input then changed from 0.010, 0.03 and 0.012 measured at the point straight above the neuron to 0.009, 0.03 and 0.009 for "large EEG electrodes", respectively.

Chapter 5

Discussion

Findings

A tool for calculating axial currents in cortical neurons, based on membrane potentials and axial resistances generated by `NEURON` and `LFPy`, was implemented in `Python`. Further, axial currents and transmembrane currents from `LFPy` were used to compute the current dipole moments from single neurons. Here, the axial currents were important for enabling an internal check, meaning that current dipole moments could be calculated in two different ways and compared to each other. Further, `Python` modules for computing current dipole moments and extracellular potentials, based on the current dipole approximation, were developed.

When compared to the compartment-based forward model, in `LFPy`, it turned out that the two methods give similar results for virtual electrodes placed several millimeters away from the neuron source. Thus, the dipole approximation cannot be used for predicting single cell contributions to ECoG signals. For modeling of single cell EEG contributions, on the other hand, the current dipole approximation appears to be applicable.

Challenges

The biggest challenge faced during the work with developing the `Python` methods, was finding the axial resistances of a complex neuron morphology, e.g. getting a grip on `NEURON`'s mysterious ghost segments and how to find the resistance of a sibling segment.

Notes on Simplifications

A couple of major simplifications were done with respect to conductivity and the volume conductor model. First of all, the extracellular medium consisting of cortex, skull and scalp is assumed to be homogeneous, i.e., all parts are assumed to have the conductivity. In addition, we don't take anisotropy into account, meaning conductivity is assumed not to be dependent on the spatial direction. In reality, the conductivity is significantly higher when moving along than across the axons in white matter. The study by Vorweck et al., 2014 [43] takes the three-compartment head model, with brain, skull and scalp, as a starting point for investigating which other conductive compartments to include in head volume conductor modeling. They suggest to additionally include the cerebrospinal fluid and to distinguish between gray and white matter when modeling EEG signals [43].

When modeling EEG signals in this project, values for human skull and scalp thickness have been used to illustrate the distance from the neuron to the measuring point. The MS neuron and the Hay neuron studied in the project are, however, not human cells, but rather a cat and a rat cell. Human cortex is thicker than rat and cat cortex [7], and since pyramidal cells can extend through all layers of cortex [8], human pyramidal cells are longer than rat and cat cells, and will therefore produce current dipole moments with longer dipole lengths. The accuracy of the current dipole approximation depends on the dipole length, and simulations equivalent to the ones studied here, will give a larger error for human pyramidal cells.

In order to use the current dipole approximation, the location of the current dipole needs to be estimated. In this project the dipole location was defined as the mean position between the soma and the synapses. A better approximation could be to calculate the center of cell membrane area, and find the middle position between this point and the average synapse position.

Advantages of using Axial Currents and Current Dipoles

When comparing the current dipole approximation to the compartment-based forward formula, there appears to be several great advantages of using the former. One of them being that the new method for calculating extracellular potentials can be based on axial and not transmembrane currents. As described in Section 3.2.2, the axial currents are calculated from the neuron's membrane potentials and inner axial resistances. The axial resistances are constant properties of the neuron morphology, and the neuron's membrane potentials are key data likely to be calculated in any simulation. This way, no large amount of extra data is needed for calculating axial currents, and hence the current dipole moment, and the current dipole approximation is therefore suitable for implementing into already existing neuron simulations. Axial currents, can further be used for calculating transmembrane currents, whenever these are needed but not calculated in the simulation. In Lindén et al. [26], a problem in `LFPy` related to parallelization and calculations of signals from networks of neurons was reported. This problem was caused by an extracellular mechanism in `NEURON` needed to extract transmembrane currents. Using axial currents instead may be a way of avoiding the issue.

A plus of using current dipoles is storage: Let us assume a neuron has 1000 segments and we run a one-time-step simulation. Calculating extracellular potentials from transmembrane currents, we would have to store one current value per segment, giving a total of 1000 numbers to store. In contrast, the current dipole moment for the same neuron simulation requires three numbers, one for each direction of the 3D current dipole vector. Hence, the current dipole approximation requires much less storage, not to forget, however, that the approximation is restricted to extracortical recordings. Besides, if the current dipole moments are going to be used for computing EEG signals, we're only interested in low-frequency parts of the signals, and a downsampling can be done to further decrease the amount of data needed.

Another advantage is that when working with neuron populations, it is a lot easier to analyze the magnitude and orientation of the current dipole moments than to work with thousands of transmembrane currents. With only two vectors per neuron, containing $|\mathbf{p}(t)|$ and $\theta(t)$ we can get a good understanding of the activity of a whole neuron population. It is also conceivable that these two vectors will behave similarly across subpopulations of neurons, in such a way that we can use averages for these populations, further simplifying the link between neural activity and EEG.

Dipole localization is already used in EEG analysis [39, 24] and up until now, to the best of our knowledge, the precise link between the localized dipoles and the neural activity has not been thoroughly explored. Our hope is that the approach described in this project can contribute to clarifying this relation.

Outlook

To improve the simulations done in this project, we suggest the following changes: A more realistic head volume conductor model, e.g., a cortex-skull-scalp model, or the five-compartment model suggested by Vorweck et al. [43]. In addition, a more accurate method for positioning of the current dipole could be implemented in the method.

Further, the methods described in this thesis could be tested against experiments such as Teleńcuk et al., 2014 [41]. In this study, the correlation between single-cell cortical action potentials and epidural EEG was investigated, by recording epidural EEG and single-unit activity from macaque cortex. In an analogue virtual experiment, extracellular potentials could be estimated with the compartment-based forward-modeling scheme, and compared to the results of the experimental recordings.

Implementing the `Python` modules developed in this project in the Human Brain Project and BrainScales [2, 1], would open for comparisons between simulated EEG signals from large neural networks and experimental data from real EEG recordings. This could potentially lead to even better neuron models and a better understanding of neural network dynamics, which can bring us closer to figuring out the underlying biophysics of higher brain functions.

Bibliography

[1] Brainscales. 'https://brainscales.kip.uni-heidelberg.de/'.

[2] Human brain project. 'https://www.humanbrainproject.eu/'.

[3] LFPy 1.0.0 documentation. 'http://compneuro.umb.no/LFPy/classes.html'.

[4] NEURON 7.3 documentation: Conceptual overview of sections. 'http://www.neuron.yale.edu/neuron/static/new_doc/modelspec/programmatic/topology/geometry.html#'.

[5] NEURON documentation. 'http://www.neuron.yale.edu/neuron/docs'.

[6] Python documentation. 'https://www.python.org/doc/'.

[7] Moshe Abeles. *Corticonics: Neural circuits of the cerebral cortex.* Cambridge University Press, 1991.

[8] Mark F. Bear, Barry W. Connors, and Michael A. Paradiso. *NEUROSCIENCE Exploring the Brain.* Lippincott Williams & Wilkins, Philadelphia, 2007.

[9] Hans Berger. Über das Elektreenkephalogramm des Menschen. *Arch. Psychiatr. Nervenkr.*, (87), 1929.

[10] Silvia Budday, Charles Raybaud, and Ellen Kuhl. A mechanical model predicts morphological abnormalities in the developing human brain. *Scientific reports*, 4, 2014.

[11] Nicholas T. Carnevale and Michael L. Hines. *The NEURON Book.* Cambridge University Press, Cambridge, 2006.

[12] Ted Carnevale. Neuron simulation environment. *Scholarpedia 2*, 2007.

[13] Richard Caton. The electric currents of the brain. *British Medical Journal*, (2), 1875.

[14] Peter Dayan and L. F. Abbott. *Theoretical Neuroscience, Computational and Mathematical Modeling of Neural Analysis.* The MIT Press, Cambridge, Massachusetts, 2001.

[15] Gaute T. Einevoll, Dieter Jaeger (Ed), and Ranu Jung (Ed). *Encyclopedia of Computational Neuroscience, Forward Modeling of Extracellular Potentials.* Springer Reference, New York, 2013.

[16] Gaute T. Einevoll, Christoph Kayser, Nikos K. Logothetis, and Stefano Panzeri. Modelling and analysis of local field potentials for studying the function of cortical circuits. *Nature Reviews, Neuroscience*, 14, 2013.

[17] Gaute T. Einevoll, Henrik Lindén, Tom Tetzlaff, Szymon Łęski, Klas H. Pettersen, Rodrigo Quian Quiroga (Ed), and Stefano Panzeri (Ed). *Principles of Neural Coding, Local Field Potentials Biophysical Origin and Analysis.* CRC Press, Florida, 2013.

[18] Stanley Finger and Tilli Tansey. Origins of neuroscience: A history of explorations into brain function. *Trends in Neurosciences*, 17(7), 1994.

[19] M. Hämäläinen, R. Haari, R. J. Ilmoniemi, J. Knuutila, and O. V. Lounasmaa. Magnetoencephalography — theory, instrumentation, and application to noninvasive studies of the working human brain. *Reviews of Modern Physics*, 65, 1993.

[20] Etay Hay, Sean Hill, Felix Schürmann, Henry Markram, and Idan Segev. Models of neocortical layer 5b pyramidal cells capturing a wide range of dendritic and perisomatic active properties. *PLoS computational biology*, 7(7), 2011.

[21] Gary R. Holt and Christof Koch. Electrical interactions via the extracellular potential near cell bodies. *Journal of Computational Neuroscience*, 6, 1999.

[22] Christof Koch and Idan Segev. The role of single neurons in information processing. *nature neuroscience*, 3, 2000.

[23] Hans Petter Langtangen. *A primer on scientific programming with Python.* Springer, 2009.

[24] R. M. Leahy, J. C. Mosher, M. E. Spencer, M. X. Huang, and J. D. Lewine. A study of dipole localization accuracy for MEG and EEG using a human skull phantom. *Electroencephalography and clinical neurophysiology*, 107(2), 1998.

[25] H. Lindén, K. H. Pettersen, and G. T. Einevoll. Intrinsic dendritic filtering gives low-pass power spectra of local field potentials. *Springer Science+Business Media*, 29, 2010.

[26] Henrik Lindén, Espen Hagen, Szymon Łęski, Klas H. Pettersen Eivind S. Norheim, and Gaute T. Einevoll. LFPy: a tool for biophysical simulation of extracellular potentials generated by detailed model neurons. *Front. Neuroinform*, 7, 2014.

[27] Nikos K. Logothetis, Christoph Kayser, and Axel Oeltermann. In vivo measurement of cortical impedance spectrum in monkeys: Implications for signal propagation. *Neuron*, 55(5), 2007.

[28] Michael London and Michael Häusser. Dendritic computation. *Annual Review of Neuroscience*, 28(1), 2005.

[29] Z. F. Mainen and T. J. Sejnowski. Influence of dendritic structure on firing pattern in model neocortical neurons. *Nature*, 382, 1996.

[30] Philip Nelson. *Biological physics.* WH Freeman New York, 2004.

[31] P. L. Nunez and R. Srinivasan. Electroencephalogram. 2(2):1348, 2007.

[32] Paul L. Nunez and Ramesh Srinivasan. *Electric Fields of the Brain.* Oxford University Press, New York, 2006.

[33] William H. Oldendorf and Youichi Iisaka. Interference of scalp and skull with external measurements of brain isotope content: Part 1. isotope content of scalp and skull. Technical report, Wadsworth Hospital, Los Angeles. Univ. of California, Los Angeles, 1969.

[34] Klas Pettersen, Henrik Lindén, Tom Tetzlaff, and Gaute T. Einevoll. Power laws from linear neuronal cable theory: Power spectral densities of the soma potential, soma membrane current and single-neuron contribution to the EEG. *PLoS Comput Biol*, 10, 2014.

[35] Klas H. Pettersen and Gaute T. Einevoll. Amplitude variability and extracellular low-pass filtering of neuronal spikes. *Biophysical Journal*, 94, 2008.

[36] Klas H. Pettersen, Espen Hagen, and Gaute T. Einevoll. Estimation of population firing rates and current source densities from laminar electrode recordings. *Journal of Computational Neuroscience*, 24(3), 2008.

[37] Klas H. Pettersen, Henrik Lindén, Anders M Dale, and Gaute T Einevoll. Extracellular spikes and CSD. *Handbook of Neural Activity Measurement*, 2012.

[38] Wilfrid Rall. Electrophysiology of a dendritic neuron model. *Biophysical journal*, 2, 1962.

[39] R. R. Ramírez. Source localization. *Scholarpedia*, 3(11), 2008.

[40] David Sterratt, Bruce Graham, Andrew Gillies, and David Willshaw. *Principles of Computational Modeling in Neuroscience*. Cambridge University Press, Cambridge, 2011.

[41] Bartosz Teleńczuk, Stuart Baker, Richard Kempter, and Gabriel Curio. Correlates of a single cortical action potential in the epidural EEG. *NeuroImage*, 2014.

[42] Michael C. Trachtenberg and Daniel A. Pollen. Neuroglia: biophysical properties and physiologic function. *Science*, 167(3922), 1970.

[43] Johannes Vorwerk, Jae-Hyun Cho, Stefan Rampp, Hajo Hamer, Thomas R. Knösche, and Carsten H. Wolters. A guideline for head volume conductor modeling in EEG and MEG. *NeuroImage*, 100(0), 2014.

Appendix A

Orientation of Current Dipole Moment in Space

The spatial orientation of a current dipole moment can be defined by the two angles shown in Figure A.1: the angle between $\mathbf{p}$ and the z-axis: θ_0, and the angle between the radial component of the current dipole moment: $\mathbf{p}_\rho$ and the x-axis: ϕ_0.

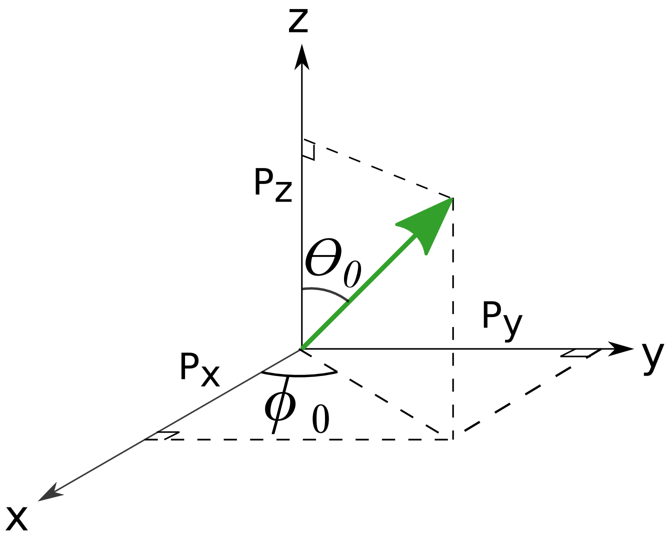

Figure A.1: Orientation of Current Dipole Moment in Space. Illustration of the current dipole moment vector in space. Here, θ_0 is the angle between $\mathbf{p}$ and the z-axis and ϕ_0 is the angle between the radial component of $\mathbf{p}$ and the x-axis.

The angles can easily be found by decomposing $\mathbf{p}$:
$\mathbf{p}(t) = \mathbf{p}_x(t) + \mathbf{p}_y(t) + \mathbf{p}_z(t) = p_x(t)\hat{\mathbf{x}} + p_y(t)\hat{\mathbf{y}} + p_z(t)\hat{\mathbf{z}}$, and the definition of the dot product

$$\begin{aligned}
\hat{\mathbf{z}} \cdot \mathbf{p}(t) &= |\hat{\mathbf{z}}|\ |\mathbf{p}(t)| \cos\theta_0(t) \\
&\Downarrow \qquad \text{(A.1)} \\
\theta_0(t) &= \cos^{-1}\Big(\frac{\hat{\mathbf{z}} \cdot \mathbf{p}(t)}{|\hat{\mathbf{z}}||\mathbf{p}(t)|}\Big) \qquad \text{(A.2)} \\
&= \cos^{-1}\Big(\frac{\mathbf{p}_{\mathbf{z}}(t)}{|\mathbf{p}(t)|}\Big), \qquad \text{(A.3)}
\end{aligned}$$

$$\begin{aligned}
\hat{\mathbf{x}} \cdot (\mathbf{p}_x(t) + \mathbf{p}_y(t)) &= |\hat{\mathbf{x}}|\ |(\mathbf{p}_x(t) + \mathbf{p}_y(t))| \cos\phi_0(t) \\
&\Downarrow \qquad \text{(A.4)} \\
\phi_0(t) &= \cos^{-1}\Big(\frac{\hat{\mathbf{x}} \cdot (\mathbf{p}_x(t) + \mathbf{p}_y(t))}{|\hat{\mathbf{x}}||(\mathbf{p}_x(t) + \mathbf{p}_y(t))|}\Big) \qquad \text{(A.5)} \\
&= \cos^{-1}\Big(\frac{\mathbf{p}_x(t)}{|(\mathbf{p}_x(t) + \mathbf{p}_y(t))|}\Big), \qquad \text{(A.6)}
\end{aligned}$$

where $\theta_0 \in [0, \pi]$ and $\phi_0 \in [0, 2\pi]$. Now the current dipole moment can be completely described by $|\mathbf{p}|$, θ_0 and ϕ_0.

Appendix B

Dipole Class

```
# -*- coding: utf-8 -*-
"""
Created on Mon May 11 09:28:16 2015

@author: solveig
"""

# -*- coding: utf-8 -*-
"""
Created on Wed Dec 17 14:53:06 2014

@author: solveig
"""
import numpy as np
import neuron as nrn

class Dipolefixed:
    """Calculate currents and current dipole moment from cell."""

    def __init__(self, cell, first_sp = None):
        """Initialize cell, # timesteps, # sections in cell."""
        self.cell = cell
        self.time = len(cell.tvec)
        self.numsecs = len(cell.allsecnames)
        self.vlist = cell.vmem
        self.children_dict = self.children_dictionary()
        self.ri_list = self.axial_resistance()
        if not first_sp:
            first_sp = cell.sptimeslist[0][0]
        self.startstep = int((self.time - 1)/(self.cell.tstopms-
                        self.cell.tstartms)*np.floor(first_sp) + 1)
    def transmembrane_currents(self):
        """Return midpoint locations and transmembrane currents.

           Returns
           -------
           r_seg : ndarray [microm]
               Array containing location vectors for midpoints of
               all segments in cell.

           i_trans : ndarray [nA]
               Array of transmembrane currents through midpoints of
               all compartments in cell.
        """

        r_seg = np.array([self.cell.get_intersegment_vector(idx0 =
                        0, idx1 = i) for i in range(
                        self.cell.totnsegs)])
        r_seg += self.cell.somapos

        i_trans = self.cell.imem

        return r_seg, i_trans

    def axial_currents(self):
        """Return magnitude and distance traveled by axial currents.

           Returns
           -------
           i_axial : ndarray [nA]
               Array of axial currents, I(t) going from compartment
               end to mid/ compartment mid to start for all
               comparment halves in cell.
           d_list : ndarray [microm]
               Array of distance vectors traveled by each axial
               current in i_axial.
        """

        iaxial = []
        d_list = []

        dseg = zip((self.cell.xmid - self.cell.xstart),
                   (self.cell.ymid - self.cell.ystart),
                   (self.cell.zmid - self.cell.zstart))
        dpar = zip((self.cell.xend - self.cell.xmid),
                   (self.cell.yend - self.cell.ymid),
                   (self.cell.zend - self.cell.zmid))

        for secnum, sec in enumerate(nrn.h.allsec()):
            if not nrn.h.SectionRef(sec.name()).has_parent():
                # skip soma, since soma is an orphan
                continue

            bottom_seg = True

            parentseg = nrn.h.SectionRef(sec.name()).parent()
            parentsec = parentseg.sec

            branch = len(self.children_dict[
                              parentsec.name()])> 1

            parent_idx = self.cell.get_idx(section =
                                           parentsec.name())[-1]
            seg_idx = self.cell.get_idx(section=sec.name())[0]


            # we only need parent_ri calculated for bottom-segments
            # that aren’t children of soma.
            parent_ri = (nrn.h.ri(0) if bottom_seg and
                         not ’soma’ in parentsec.name() else 0)

            for seg in sec:
                iseg, ipar = self.parent_and_segment_i(seg_idx,
                               parent_idx, parent_ri,bottom_seg,
                               branch, sec, parentsec)

                if bottom_seg and ’soma’ in parentsec.name():
                    # if a seg is connencted to soma, it is
                    # connected to the middle of soma,
                    # and dpar needs to be altered.
                    dpar[parent_idx] = [(self.cell.xstart[seg_idx] -
                                        self.cell.xmid[parent_idx]),
                                        (self.cell.ystart[seg_idx] -
                                        self.cell.ymid[parent_idx]),
                                        (self.cell.zstart[seg_idx] -
                                        self.cell.zmid[parent_idx])]

                d_list.append(dpar[parent_idx])
                d_list.append(dseg[seg_idx])
                iaxial.append(ipar)
                iaxial.append(iseg)

                parent_idx = seg_idx
                seg_idx += 1
#                counter += 2
                branch = False
                bottom_seg = False
                parent_ri = 0
        return np.array(d_list), np.array(iaxial)
```

```
    def axial_resistance(self):
        """Return NEURON axial resistance for all cell compartments.

           Returns
           -------
           ri_list : ndarray [MOhm]
               Array containing nrn.h.ri(seg.x) for all segments in
               cell. nrn.h.ri(seg.x) is the axial resistance from
               the middle of the segment to the middle of its parent
               segment. If seg is the first/ bottom segment in a
               section, nrn.h.ri(seg.x) is the axial resistance from
               the middle to the start of the segment, seg, only.
        """

        ri_list = np.zeros(self.cell.totnsegs)
        comp = 0
        for sec in nrn.h.allsec():
            for seg in sec:
                ri_list[comp] = nrn.h.ri(seg.x)
                comp += 1
        return ri_list

    def children_dictionary(self):
        """Return dictionary with children seg indices for all secs.

           Returns
           -------
           children_dict : dictionary
               Dictionary containing a list for each section,
               with the segment index of all the section’s children.
               The dictionary is needed to find the
               sibling of a segment.
        """

        children_dict = {}
        for sec in nrn.h.allsec():
            children_dict[sec.name()] = []
            for child in nrn.h.SectionRef(sec.name()).child:
                # add index of first segment of each child
                children_dict[sec.name()].append(self.cell.get_idx(
                                        section = child.name())[0])

        return children_dict

    def parent_and_segment_i(self, seg_idx, parent_idx,
                             parent_ri, bottom_seg, branch,
                             sec, parentsec):
        """Return current from segmid to start and parentend to mid.
```

```
           Parameters
           ----------
           seg_idx : int
           parent_idx : int
           parent_ri : float [MOhm]
           bottom_seg : boolean
           branch : boolean
           sec : nrn.Section object
           parentsec : nrn.Section object

           Returns
           -------
           iseg : ndarray [nA]
               ndarray containing axial currents from segment middle
               to segment start for all segments in sec.
           ipar : ndarray [nA]
               ndarray containing axial currents from
               parent segment end to parent segment middle
               forall parent segments in cell.

        """
        seg_ri = self.ri_list[seg_idx]
        vpar = self.vlist[parent_idx]
        vseg = self.vlist[seg_idx]
        if bottom_seg and branch and not 'soma' in parentsec.name():
            # segment is a bottom_seg with siblings and a parent
            # hat is not soma. need to calculate ipar and iseg
            # separately.

            [[sib_idx]] = np.take(self.children_dict[
                                parentsec.name()],
                                np.where(self.children_dict[
                                parentsec.name()]
                              != seg_idx))
            sib_ri = self.ri_list[sib_idx]
            vsib = self.vlist[sib_idx]

            if np.abs(parent_ri) < 1e-8:
                raise RuntimeError("Zero parent ri")

            v_branch = (vpar/parent_ri + vseg/seg_ri +
                        vsib/sib_ri)*(1./(1./parent_ri +
                        1./seg_ri + 1./sib_ri))
            # only a fraction of ipar is added for each parent,
            # since children can have the same parent
            # and ipar should only be counted once.
            ipar = (vpar -
                v_branch)/parent_ri/len(self.children_dict[
                                        parentsec.name()])
            iseg = (v_branch - vseg)/seg_ri
        else:
            ri = (parent_ri + seg_ri)
            iseg = (vpar - vseg)/ri
            ipar = iseg

        return iseg, ipar

    def current_dipole_moment(self, dist, current):
        """Return current dipole moment vector P and P_tot.

           Parameters
           ----------
           current : ndarray [nA]
               Either an array containing all transmembrane currents
               from all compartments of the cell. Or an array of all
               axial currents between compartments in cell.
           dist : ndarray [microm]
               When input current is an array of axial currents,
               the dist is the length of each axial current.
               When current is the an array of transmembrane
               currents, dist is the position vector of each
               compartment middle.

           Returns
           -------
           P : ndarray [10^-15 mA]
               Array containing the current dipole moment for all
               timesteps in the x-, y- and z-direction.
           P_tot : ndarray [10^-15 mA]
               Array containing the magnitude of the
               current dipole moment vector for all timesteps.
        """

        P = np.dot(current.T, dist)
        P[:self.startstep] = 0.
        P_tot = np.sqrt(np.sum(P**2, axis=1))
        return P, P_tot
```

Appendix C

CalcLFP Class

```
# -*- coding: utf-8 -*-
"""
Created on Mon Feb 23 12:20:17 2015

@author: solveig
"""

import numpy as np

class CalcLFP:
    '''Calculate extracellular potentials from cell.'''

    def __init__(self, cell, X, Y, Z, ppidx = [], first_sp = None):
        '''Initialize cell, X, Y, Z and dipole midpoint r_mid.'''
        # conversion factors:
        self.k1 = 1E6 # from mV to nV

        self.cell = cell
        self.time = len(cell.tvec)
        self.totnsegs = cell.totnsegs
        self.X = X
        self.Y = Y
        self.Z = Z
        syninds = cell.synidx + ppidx
        r_soma_syns = [self.cell.get_intersegment_vector(idx0 = 0,
                       idx1 = i) for i in syninds]
        self.r_mid = np.average(r_soma_syns, axis = 0)
        self.r_mid = self.r_mid/2. + self.cell.somapos
        if not first_sp:
            first_sp = cell.sptimeslist[0][0]
        self.startstep = int((self.time - 1)/(
                             self.cell.tstopms -
                             self.cell.tstartms)*np.floor(
                             first_sp) + 1)


    def grid_lfp_theta(self, P, sigma):
        '''Return array phi(t) for points in XYZ-grid,timedep theta.

           Parameters
           ----------
           P : ndarray [1E-15 mA]
               Array containing the current dipole moment for
               all timesteps in the x-, y- and z-direction.
           sigma : float [ohm/m]
               Extracellular Conductivity.

           Returns
           -------
           theta : ndarray [radians]
               Angle between phi(t) and distance vector from
               electrode to current dipole location,
               calculated for all timesteps.
           grid_LFP : ndarray [nV]
               Array containing the current dipole moment at all
               points in X-, Y-, Z-grid for all timesteps.
        '''
        gridpoints = zip(self.X.flatten(), self.Y.flatten(),
                         self.Z.flatten())
        grid_LFP = np.zeros((len(gridpoints), self.time))
        for j in range(len(gridpoints)):
            dist = gridpoints[j] - self.r_mid
            cos_theta = np.dot(P, dist)/(np.linalg.norm(
                        dist)*np.linalg.norm(P, axis = 1))
            cos_theta = np.nan_to_num(cos_theta)
            theta = np.arccos(cos_theta)
            grid_LFP[j, :] = 1./(4*np.pi*sigma)*np.linalg.norm(P,
                             axis = 1)*cos_theta/np.sum(
                             dist**2)*self.k1

        return grid_LFP, theta
```

Norwegian University
of Life Sciences

Postboks 5003
NO-1432 Ås, Norway
+47 67 23 00 00
www.nmbu.no