Extracellular Potentials from Action Potentials of Anatomically Realistic Neurons and Neuronal Populations

by
Miguel Cavero

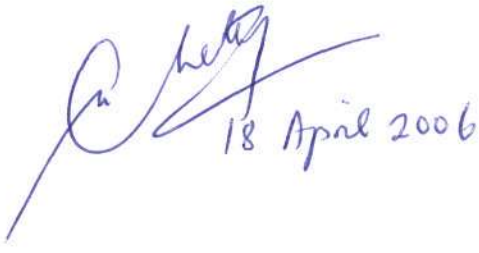
Submitted in partial fulfillment of the requirements
for the degree of Master of Science in the
School of Physics,
University of KwaZulu-Natal

Pietermaritzburg
December, 2005

Declaration

I declare that this work is a result of my own research, except where specifically indicated to the contrary, and has not been submitted for any other degree of examination to any other university.

Signed: M Cavers Date: 24-3-2006



Abstract

Extracellular potentials due to firing of action potentials are computed around cortical neurons and populations of cortical neurons. These extracellular potentials are calculated as a sum of contributions from ionic currents passing through the cell membrane at various locations using Maxwell's equations in the quasi-static limit. These transmembrane currents are found from simulations of anatomically reconstructed cortical neurons implemented as multi-compartmental models in the simulation tool NEURON.

Extracellular signatures of action potentials of single neurons are calculated both in the immediate vicinity of the neuron somas and along vertical axes. For the neuronal populations only vertical axis distributions are considered. The vertical-axis calculations were performed to investigate the contributions of action potential firing to laminar-electrode recordings. Results for high-pass (750 - 3000 Hz) filtered potentials are also given to mimic multi-unit activity (MUA) recordings.

Extracellular traces from single neurons and populations (both synchronous and asynchronous) of neurons are shown for three different neuron types: layer 3 pyra-

mid, layer 4 stellate and layer 5 pyramid cell. The layer 3 cell shows a 'closed-field' configuration, while the layer 5 pyramid demonstrates an 'open-field' appearance for singe neuron simulations which is less apparent in population simulations. The layer 4 stellate cell seems to fall somewhere in between the open- and closed-field scenarios. Comparing single neuron and synchronous populations, the amplitudes of the extracellular traces increase as population radii increase, though the shapes are generally similar. Asynchronous populations produce small amplitudes due to a time convolution of various neuron contributions.

Acknowledgements

Firstly and foremost, to my Lord and Saviour Jesus Christ, in whom I live and breathe and have my being. He is my source and my strength and I am nothing without Him.

I would like to thank my parents, who have given everything so that I can pursue my dreams. Mama y Papa, you push me to do my best and your support and encouragement always goes with me; so to you I dedicate this thesis, and do so proudly.

I am extremely grateful to my supervisors Dr Nithaya Chetty, Dr Gaute Einevoll, of the Norwegian University of Life Sciences in Ås, Norway, and Dr Rob Lindebaum, for their friendship and all their assistance. I would like to thank Nithaya for his patience with me and primarily for his guidance and motivation. Thank you to Rob for his support and help with various computational issues. I'd like to express my utmost gratitude to Gaute for the privilege of working with his research team in Norway. Jeg er dypt og inderlig takknemlig overfor ham, som har vrt til uvurderlig hjelp i arbeidet med denne rapporten.

Many thanks to Klas Pettersen for always providing rapid response to my questions and queries (over numerous e-mails!). His helpfulness, enthusiasm and friendship was always appreciated.

I would also like to express my appreciation to Belinda for her love and encouragement throughout; for inspiring me and for being my joy and delight.

To all my other friends who have stood by me: my flatmates Brendon and Dean, for their company in the good and bad times in all of our studies; the postgrads at the Physics Department, especially Trisha and Sarah, for walking together in this academic journey; and my friends in Norway, in particular Svein, Hanne and Ingvild.

Contents

Chapter 1

Introduction

In the last few decades our understanding of how the brain works has significantly increased. The new non-invasive imaging method *functional magnetic resonance imaging (fMRI)* has, when combined with alternative complementary techniques such as *positron-emission tomography (PET)*, *electroencephalography (EEG)* and *magnetoencephalography (MEG)*, given a wealth of new information about the function of the brain. A striking example is the first non-invasive mapping of the visual areas in the cortex in humans using fMRI [1]. However, the spatial resolution of these techniques is limited, and it has been difficult to firmly link the observed systems-level activity with neural activity of the underlying neuronal circuitry.

On the microscopic level invasive single-unit electrophysiological recordings have for more than half a century given significant insights into the function of single neurons. From a modelling point of view the crowning achievement was the establishment of the celebrated Hodgkin-Huxley equations describing action-potential

The remainder of the thesis is organised as follows. In Chapter 2 the physiological background is introduced, namely for the cortex and cortical neurons. The techniques used in measuring neural activities are also discussed. In Chapter 3 compartmental modelling is introduced, showing how the electrical properties of complex neurons can be described by mathematical models. A brief overview of the simulation tool NEURON is given in Chapter 4; the NEURON simulation tool was used in the simulations presented in this thesis. Chapter 5 shows how extracellular potentials outside of neurons can be calculated using data from the NEURON simulations. The results for the extracellular potential related to an action potential around a single neuron are given in Chapter 6. Results are presented for three different neuron types, where the extracellular potential is calculated at various positions around each cell. Potentials are calculated very close around the soma, as well as for points further away from the soma in a vertical axis. The extracellular traces are also filtered to produce multiple-unit activity (MUA) results. Extracellular potential traces around a population of both synchronous and asynchronous neurons for all three neuron types are given in Chapter 7. Results for various population sizes are shown. Chapter 8 gives a summary of the project and the results obtained.

Chapter 2

Physiological Background

2.1 Organisation of the Brain

The brain is the most complex structure known to man. It is the centre of information processing in the body and does so by the flow of small currents in its cells. Various invasive as well as noninvasive techniques have been developed to study not only animal but human brains as well. This chapter describes the structure of the cortex and the neuron and the methods used to measure electrical activity of the brain.

The adult human brain weighs approximately 1300 to 1400 g and is made up of about 10^{10} neurons in the cerebral cortex. The connections between neurons, called synapses, number about 10^{14}. The brain is divided up into two hemispheres and each of these is made up of four lobes: frontal, parietal, temporal and occipital [10], as shown in Figure 2.1.

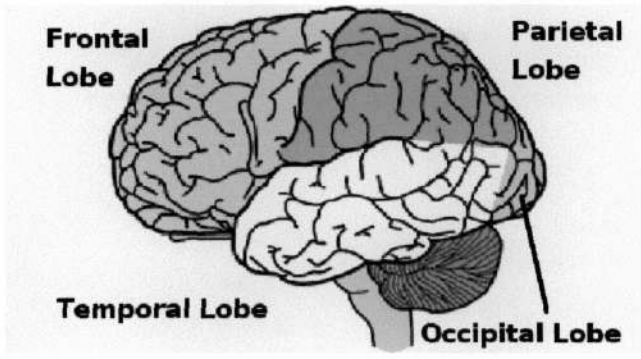


Figure 2.1:
Lateral view of the brain. The four lobes of the brain are shown: frontal, parietal, temporal and occipital. The folded surface of the brain is the cortex [11].

The part of the brain that is of most interest is the cortex, the processing part of the brain. The cortex is the outermost layer, characterised by its highly folded structure. It has a surface area of approximately 2500 cm^2 [10].

The brain consists of two main types of cells that are the fundamental constituents: neurons and glial cells. The latter provide structural support as well as nutrients from blood vessels to the brain tissue. Some glial cells form what are known as myelin sheaths. These cover the long axons of neurons, thereby assisting in the propagation of the electrical signals produced by neurons.

2.2 Cortex

The cerebral cortex, the outermost layer of grey matter, is responsible for higher brain functions that include muscle movement, thought and memory. Neurons in

the cortex are structured into six layers, although there are no distinct boundaries of the layers. Neurons and their axons and dendrites can extend to other cortical layers [12]:

layer I	molecular layer
layer II	external granular layer
layer III	external pyramidal layer
layer IV	internal granular layer
layer V	internal pyramidal layer
layer VI	multiform layer

Layer I is the most superficial layer while layer VI is the deepest. The height of each layer is not the same for all parts of the cortex. This system of labelling the layers of the cortex was done by the German Korbinian Brodmann in 1909. A picture of the structure of the macaque cortex is shown in Figure 2.2.

The cortex contains neurons that appear to group themselves in small columns. There are two main types of neurons found in the cortex: stellate and pyramidal cells. Stellate cells are the main interneurons of the cortex. An interneuron is a neuron in the cortex that connects other neurons; it is neither a sensory nor a motor neuron. They come in a variety of shapes and have axons that do not leave the cortex. They are typically small cells. Pyramidal cells, on the other hand, can be much larger. They usually have long apical dendrites that extend vertically up towards the cortex surface. Most pyramidal cells have long axons that extend to other areas

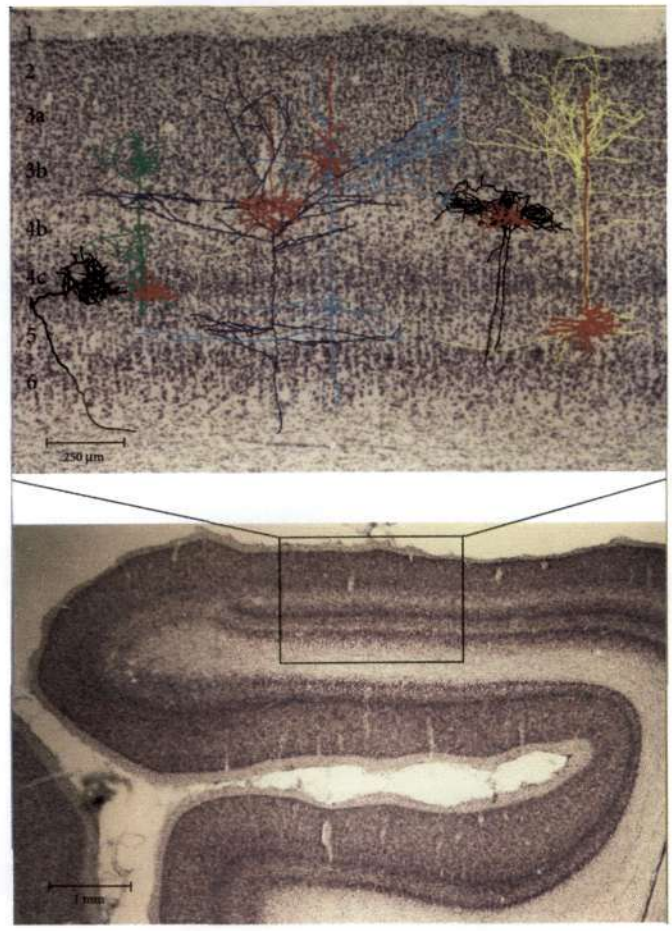


Figure 2.2:
Illustration of the structure of the macaque monkey cortex; the blow-up shows the different layers of the cortex as well as reconstructions of neurons. The red lines correspond to dendrites. The other colours show axons of the same neurons. Stellate cells (second from right) and pyramidal cells are visible [14].

in the cortex as well as subcortical regions [13].

Connections between neurons within the cortex have been studied using anatomical techniques. They have begun to show complex circuitry of the cortex by providing information of the afferent (inward carrying) inputs and efferent outputs from each layer of the cortex. Although the simulations presented later in this thesis do not include connections between neurons in a population, it is important to understand

how different groups or populations interact with one another. The neurons used in the simulations were pyramids from layers III and V and a stellate cell from layer IV.

The most studied input to the neocortex is the thalamus. It has been called the main relay to the cortex. The major areas in the cortex depend on a well defined group of cells in the thalamus [15]. Thalamic nuclei carry sensory information of one kind of sensory input to specific regions in the cortex. An example is the dorsal LGN (lateral geniculate nucleus) that project only to the primary visual cortex. Any particular region in the cortex also contains inputs from other cortical regions.

Layer V contains large pyramidal neurons that fire action potentials in bursts and small pyramids that exhibit regular spiking patterns. The dendrites of the latter extend to layer III and often not beyond. Those from upper layer V can project to the striatum, as for example in primates, while lower layer V cells project to non-specific thalamic nuclei, for example in rodents. In rats, layer V pyramids project to all layers in the cortex [16].

Layer IV spiny neurons have branches going to layer III, some within layer IV and some branches, less densely, reaching into the deeper layers. Layer IV spiny cells are sublayer specific. Layer III pyramid axons project to layer II, III and V. Axons pass through layers IV and VI without making connections in these layers [16].

Having briefly looked at the cortex in its layered structure, a typical cortical neuron

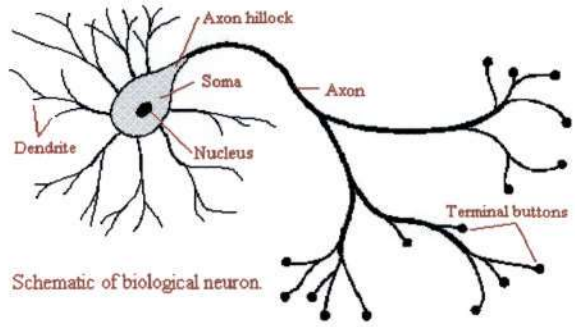


Figure 2.3:
Schematic representation of a neuron in the cortex, showing the various cellular specialisations: dendrites receive action potentials from other neurons; axons send action potentials generated in the neuron to other neurons in the cortex [17].

is now described in more detail.

2.3 Neurons

Neurons are the basic units for processing information in the brain. They are highly specialised cells with structures specifically used for the purpose of communicating with other neurons. A neuron is made up of a soma (or cell body), dendrites and an axon (see Figure 2.3). The soma is the place which contains the nucleus of the cell and where the neuron generates its electrical signals. Neurons fire action potentials, electrical pulses generated in the soma, which propagate along the axon to other neurons. The dendrites are branch-like extensions used to receive action potentials from other neurons. The axon is the output carrying part of a neuron. It is often a long, single fibre used to send an action potential to other neurons.

Electrical signals from other cells enter the neuron through its numerous dendrites. The dendrites transmit these signals to the soma, which can also receive electrical signals from other cells directly (e.g. when an axon from another neuron connects to the cell body itself). The soma receives constant bombardment of action potentials from other neurons and may generate its own action potential depending on the timing of the inputs of other neurons and how they affect the membrane potential of the soma. Under the appropriate conditions, the neuron will generate an action potential which typically occurs at the base of the axon. The action potential is a fluctuation in the potential across the cell membrane (approximately 100 mV) that lasts about 1 ms. The action potential propagates down the axon maintaining a constant velocity and amplitude, being actively regenerated along the axon by the appropriate ion channels..

Neurons are connected to each other via synapses (an example is shown in Figure 2.4). The point where the axon ends (or where it connects to another neuron) is called the axon terminal. There is a small gap or cleft between the terminal and the dendritic spine, an extension of a (postsynaptic) dendrite. While an action potential is an electrical signal that is transmitted along axons and dendrites, a synapse (generally) uses a chemical signal. Axon terminals contain vesicles that release neurotransmitter molecules when an action potential arrives at the terminal. The transmitter molecules are released into the synaptic cleft and cross this cleft to bind with receptors on the postsynaptic side. An action potential is then generated which propagates down the dendrite to the soma. Synapses can either be excitatory

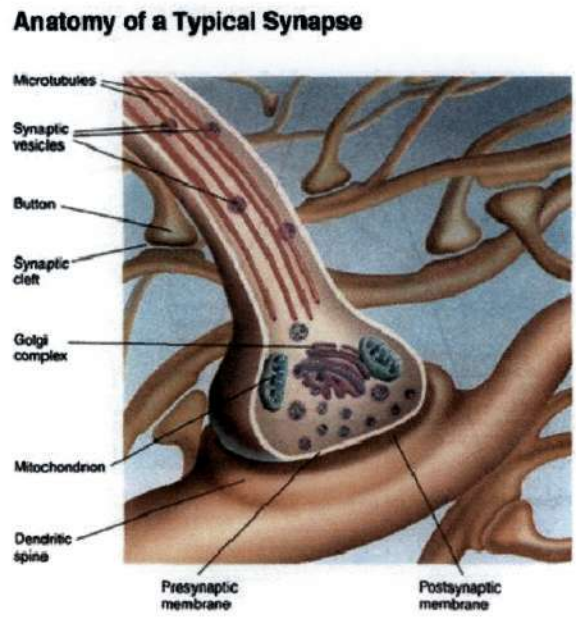


Figure 2.4:
Diagram of a typical synapse in the brain. An action potential arriving at the axon terminal causes synaptic vesicles to release the neurotransmitter molecules across the synaptic cleft. When these molecules bind to the receptors in the dendritic spine, an action potential is generated that propagates towards the soma of the postsynaptic neuron [18].

or inhibitory. An excitatory synapse is one that when active, increases the probability of the postsynaptic neuron to fire an action potential; similarly, inhibitory neurons decrease the probability for the postsynaptic neuron to for an action potential.

Action potentials are the result of changes in the potential across the cell membrane of a neuron. When the neuron is at rest, i.e. when it is inactive, the potential inside the neuron is approximately 70 mV lower than the potential outside in the extracellular medium. The cell membrane has numerous ion channels and ion pumps that span its surface. Ions flow in and out of a cell through these channels due to voltage

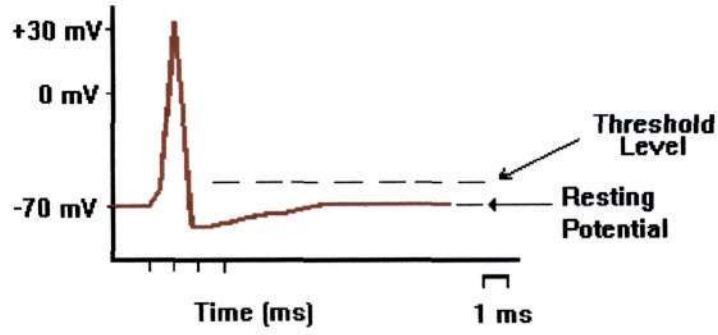


Figure 2.5:
Action potential in a neuron. The figure shows the membrane potential that starts at rest. An action potential is fired when the membrane potential exceeds a certain threshold value [19].

and concentration gradients, the most dominant being sodium (Na^+), potassium (K^+), calcium (Ca^{2+}) and chloride (Cl^-). The ion pumps use energy to actively move ions across a concentration gradient and maintain a lower potential inside the neuron relative to the outside. In its inactive state, the membrane is in dynamic equilibrium, so that the amount of current flowing into and out of the cell in the form of ions is balanced out and the excess negative charge inside the neuron is maintained.

Positively charged ions that flow into the cell cause the cell membrane to become less negative (and may cause the potential to become positive). This is called depolarisation. Positive ions flowing out of the cell make its membrane potential more negative; this is called hyperpolarisation.

When the neuron is being depolarised, the membrane potential may approach a certain threshold value. If the membrane potential does not reach this threshold level,

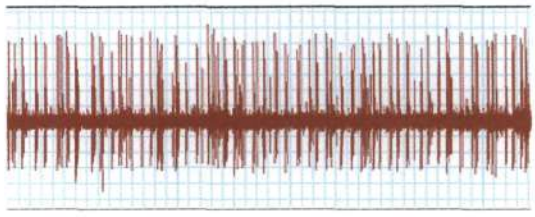


Figure 2.6:
An example of an extracellular recording. The recordings show action potentials but not subthreshold membrane potentials [21].

an outward current causes the membrane potential to become more negative and return to the resting potential value. If, however, the membrane potential exceeds the threshold level, then a positive feedback loop is initiated which further depolarises the membrane. The positive feedback increases the inward current rapidly causing the membrane to depolarise. An action potential is thus generated by the large change in the membrane potential [20].

2.4 Single-Unit Measuring Techniques

The electrical activity of single neurons has been recorded by surgically inserting electrodes in the cortex and placing them close to individual neurons. Measurements can be made intracellularly, i.e. making measurements from inside a neuron, or extracellularly by placing the electrode close to the neuron. Extracellular recordings allow one to see the action potentials fired by the neuron. Intracellular recordings show the subthreshold activity of the membrane potential, not seen extracellularly, although the superthreshold activity (the activity of the membrane potential above

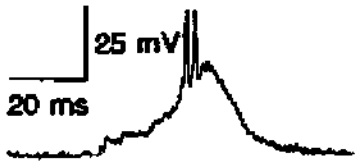


Figure 2.7:
An example of an intracellular recording, showing subthreshold membrane potentials. Two action potentials are also shown [22].

the threshold) is visible as well.

Making recordings from single neurons is a difficult task. Extracellular recordings have to be made very close to the neuron itself. The electrodes used in the single-unit measurements must also not cause damage to the neurons during experiments. Intracellular recordings are done by the use of a sharp electrode that has to penetrate the membrane of a neuron. Thin glass pipettes and microelectrodes have been developed to provide accurate measurements of electrical activity for neurons in most regions of the brain. Intracellular recordings are generally taken from the soma, although in some instances intracellular recordings have been made from dendrites and axons. Figures 2.6 and 2.7 show examples of extracellular and intracellular recordings respectively.

Extracellular single-unit recordings give reliable information about the occurrence of action potentials in single neurons (even though the detailed relationship between the shapes of the extracellularly and intracellularly recorded potentials is not straightforward [23]). One would also like to be able to simultaneously record the

firing activity of many neurons, i.e. neural populations, in the cortex simultaneously; laminar electrodes can be used to do this.

2.5 Laminar Electrodes

Techniques such as Positron Emission Tomography (PET) and functional Magnetic Resonance Imaging (fMRI) have been used to study the activity of the human cortex on a large scale. The use of single microelectrodes has led to recordings of electrical activity of single neurons. However, neurons in the cortex tend to be collected in groups in the form of columns. Cortical columns can be considered as the computational units of the brain, yet recordings from PET, fMRI or single microelectrodes cannot give information regarding these columns [6].

Multiple laminar electrodes have been developed to simultaneously record extracellular potentials across layers of the cortex. The electrode contains evenly spaced contact points that together span several layers of the cortex. Recordings are typically made at different depths of the cortex (since the neural activity is assumed to vary less within the plane of the layers [6]). Laminar electrodes also have the advantage of being able to measure activity close to different parts of the same neuron. They have been constructed for experiments in both humans and animals. The electrodes have to meet certain criteria, other than the obvious safety requirements.

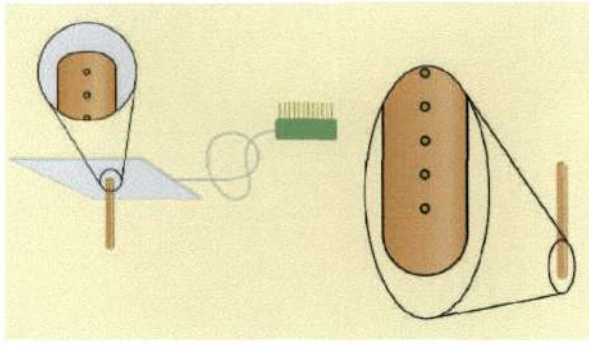


Figure 2.8:
Diagram of a multielectrode showing the electrode contacts on the shaft, connected to a silicon sheet base, and the cable that connects the electrode to the PC used for data acquisition via the connector [6, 25].

The experiments described in Ulbert *et al* [6] are the first set of recordings in humans. One of the motivations is to better understand neuronal processes during epileptic events. The laminar electrodes were used to record local field potentials (LFP) and multiple-unit activity (MUA). The LFP is the low frequency component of the electrical signals generated by neurons; the MUA is the high frequency part. The LFP is thought to reflect dendritic processing of synaptic inputs. The MUA is taken to reflect action potentials generated by neurons close to the measuring electrode [24]. Experiments performed in animals require a different setup process; the electrodes described here refer to experiments in humans.

An example of the type of multielectrodes used in experiments is shown in Figure 2.8. The electrode shafts can vary in length and have a typical diameter of 350 μm. They generally consist of 23 or 24 contact points with an intercontact distance that can be changed between 75 to 200 μm. Each contact has a width of 40 μm.

The focus of this study is on the use of laminar electrodes in animals such as rats. Experiments have been carried out using laminar electrodes to look at the relationship between neural activity and the hemodynamic response in rat barrel cortex [26]. A barrel is a well-defined area of the cortex which is affected by a particular whisker. Laminar electrodes of 24 contact points (with an intercontact spacing of 100 μm) were used to record activity in the lower layer II/III. Recordings were made for both MUA and LFP. The recorded signals were filtered at high frequencies (500 - 5000 Hz) to record MUA and at low frequencies (0.1 - 500 Hz) to record LFP.

Chapter 3

Compartmental Modelling

3.1 Cell Membrane

To design and construct models of neuronal systems, it is necessary to have a knowledge of the biophysical properties and mechanisms of neurons. The book Theoretical Neuroscience [27] was used as the main source for the background material. Neurons are highly specialised cells which receive, generate and transmit electrical signals. The starting point of modelling a neuron is to look at the structure of a neuron and in particular, the neuron cell membrane.

The purpose of the cell membrane is to enclose a neuron and to regulate the motion of substances in and out of a neuron. The movement of certain ions into and out of neurons results in the generation of electrical signals called action potentials. It is these action potentials that form the basis for transmitting information between neurons.

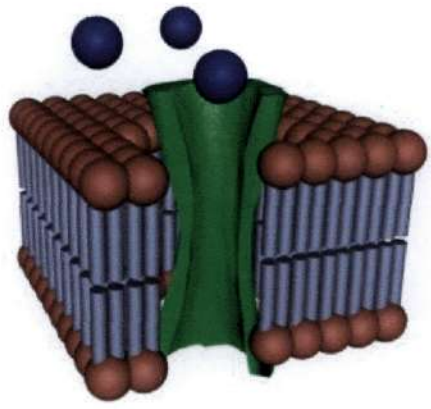


Figure 3.1:
Schematic diagram of a cell membrane showing the phospholipid bilayer nature of the membrane. The hydrophilic 'heads' are shown in red and the hydrophobic 'tails' in purple. A protein that spans the thickness of the membrane is also shown (green) with ions passing through the protein, in this case down a concentration gradient [28].

The cell membrane is a thin phospholipid bilayer, typically about 4nm in thickness. Each layer contains a phosphorus 'head' and a fatty acid chain 'tail' which is oriented in a way such that the head is aligned with either the extracellular or intracellular medium. The tail part remains in between the phosphorus layers. Other molecules can be found present in the membrane, for example, cholesterol and various proteins. Some proteins span the entire thickness of the membrane; such proteins allow the movement of ions or molecules across the membrane.

Cell membranes contain proteins that allow ions to pass through the membrane down an ion concentration gradient. These proteins are known as ion channels and several can be highly selective in terms of the ions that can pass through them; many allow only one type. Ions diffuse passively down a concentration gradient through ion channels. There are also proteins that are called ion pumps. These kinds of pro-

tein use adenosine triphosphate (ATP) to move ions across the membrane against a concentration gradient. The purpose of the ion pumps is to maintain different concentration levels of ions inside the neuron and in the extracellular medium.

A neuron at rest, or rather an inactive neuron, has a stable resting membrane potential. There is an excess of negative charges inside the neuron due to the ion pumps. Typically, the potential inside the neuron is 60 - 70 mV lower than the outside, and by convention the potential in the extracellular medium is taken to be 0. The potential difference across the membrane is called the membrane potential. A resting potential is obtained when the dynamic equilibrium state is reached where the flow of ions into the cell is equal to the flow out of the cell.

3.2 Modelling Single-Compartment Neurons

The structure of a cell membrane makes it an electrical insulator as it is impermeable to ions. The different concentrations of ions inside and outside of the cell cause charges of opposite sign to position themselves on the interior and exterior surface of the cell membrane. With a higher concentration of negative charge inside the neuron, negative ions align on the intracellular surface of the membrane, while positive ions align themselves on the extracellular surface. This makes the membrane act as a capacitor, separating charges on either side of it.

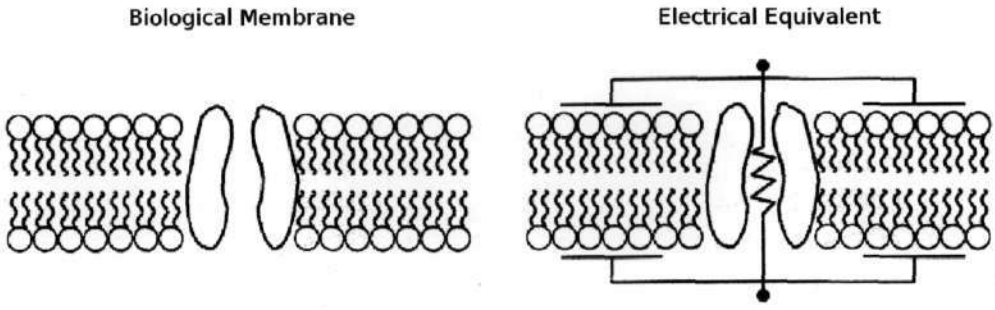


Figure 3.2:
Circuit representation of a cell membrane: the membrane is shown on the left and the equivalent circuit on the right, featuring a capacitance and a resistance in parallel.

Ions can only move in and out of a neuron due to the presence of ion channels and pumps. The presence of these channels decreases the membrane resistance. This resistance will determine how much the membrane potential will deviate from its current value due to current (in the form of ions) passing across the membrane.

The electrical properties exhibited by neuron membranes can be used to represent a neuron by a simple electrical circuit (see Figure 3.2). A mathematical model can then be created for the neuron.

Consider a one compartment neuron. In a one or single compartment neuron, the membrane potential is uniform over the entire membrane surface. Some neurons can be described in this simple manner; these neurons don't have narrow dendritic sections whose longitudinal resistances cause the membrane potential to vary substantially between different points. The negative charges on the inside of the neuron arrange themselves on the inner surface. Positive charges arrange themselves on the

outer surface, so that the cell creates a capacitance C_{m}. The potential difference across this membrane is V. The excess negative charge Q is then given by the equation for a capacitor:

$$Q = C_{\mathrm{m}} V. \tag{3.1}$$

It is now possible to calculate the current needed to change the membrane potential at a certain rate. The rate of flow of charge dQ/dt is the current that flows into the cell. This current is the amount needed to change the membrane potential of a cell with a capacitance C_{m} at a rate dV/dt and is given by

$$C_{\mathrm{m}} \frac{dV}{dt} = \frac{dQ}{dt}. \tag{3.2}$$

The membrane current is the current that flows across the membrane due to all of the ion channels in it. The contributions of all the different ion types are summed to give the total membrane current. This membrane current is typically used in the form of a membrane current per unit area and is denoted i_m.

Summing over all the different ion channel types gives the membrane current,

$$i_{\mathrm{m}} = \sum_i g_i (V - E_i). \tag{3.3}$$

The current depends on the different ion channel types i, each channel having its own reversal potential E_i. The reversal potential of a channel causes the membrane potential to move towards that reversal potential value [27]. There is an equilibrium situation when the currents flowing in or out of the neuron due to electrical forces is equal to the current flow due to diffusion gradients across the cell membrane. It can be seen that when this occurs, the current from channel i vanishes, since $V = E_i$.

Each channel type has a specific conductance per unit area g_i.

Neuron model complexity depends on the conductances that are used to make up the membrane equation. Simple models involve only a few conductance terms, whereas other models involve the inclusion of several different channels. Most conductances vary with time, but some are assumed to remain steady over time. The currents from these conductances are grouped into what is called a leakage current. This leakage term is expressed as $\overline{g}_{\mathrm{L}}(V - E_{\mathrm{L}})$. It is also known as a passive conductance, while variable conductances are called active. The line over $\overline{g}_{\mathrm{L}}$ is used to indicate that it is constant.

A very simple model, called the passive integrate-and-fire model, involves only a leakage term, so the membrane current becomes $i_{\mathrm{m}} = \overline{g}_{\mathrm{L}}(V - E_{\mathrm{L}})$. Other more complex models include the Hodgkin-Huxley model, where the membrane current depends on the leakage term as well as two active conductances. The membrane current in this example is

$$i_{\mathrm{m}} = \overline{g}_{\mathrm{L}}(V - E_{\mathrm{L}}) + \overline{g}_{\mathrm{K}} n^4 (V - E_{\mathrm{K}}) + \overline{g}_{\mathrm{Na}} m^3 h (V - E_{\mathrm{Na}}). \tag{3.4}$$

Here, V is the membrane voltage, E_{L}, E_{Na} and E_{K} are the reversal potentials for the leak, sodium and potassium conductances respectively and $\overline{g}_{\mathrm{L}}$, $\overline{g}_{\mathrm{Na}}$ and $\overline{g}_{\mathrm{K}}$ are their specific conductances. The factors n^4 and m^3h are called gating variables. They are used to model structural mechanisms in each channel that will allow ions to move through the channel. The two conductances are (1) a delayed-rectified K^+

conductance and (2) a transient Na^+ conductance, where each has its own reversal potentials and specific conductances. These conductances are examples of voltage-dependent conductances.

Referring back to Equation 3.2, the rate of change of the membrane potential is proportional to the sum of all the currents entering the neuron through membrane and synaptic conductances, as well as any current injected into the neuron by means of an electrode. Since the membrane current is typically expressed as a current per unit area, the terms in Equation 3.2 are also divided by the total surface area of the neuron. The equation relating the change in membrane potential and the current entering the neuron is

$$c_{\mathrm{m}} \frac{dV}{dt} = -i_{\mathrm{m}} + \frac{I_{\mathrm{e}}}{A}, \tag{3.5}$$

where I_{e} is the electrode current, A is the surface area of the neuron and c_{m} is the specific membrane capacitance, where $c_{\mathrm{m}} = C_{\mathrm{m}}/A$. The presence of the minus sign is a convention which is defined later. This equation uses Equation 3.3 as well as equations that define the specific conductances g_i for each channel in Equation 3.3.

Equation 3.5 is the basic equation for a neuron made up of one compartment. This neuron can be represented by a circuit that contains a capacitor connected in parallel to a series of resistors. The resistors depend on the conductances that are included in the membrane current. For example, a circuit representation of the passive integrate-and-fire model would contain a capacitor in parallel with one resistor, that of the leakage conductance. The Hodgkin-Huxley model could be represented

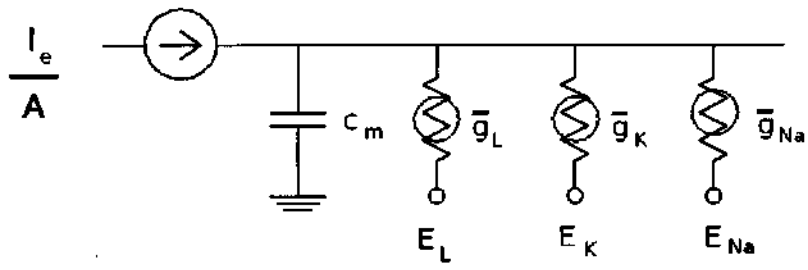


Figure 3.3:
Equivalent circuit for the Hodgkin-Huxley model. The circles over the resistors indicate a voltage-dependent conductance. I_e is the electrode current and A is the area of the neuron.

by the circuit shown in Figure 3.3.

More complex models can also be represented by equivalent circuits. These models may contain synaptic conductances and any number of additional active conductances.

3.3 Cable Equation

For simple neuronal models, or for point neurons, the membrane potential V is taken as a single value throughout the membrane. Neurons of course can have complicated morphologies with extensive dendritic trees. Neurons can have membrane potentials that change rapidly so that the potential can be different for different points along the cell membrane. In addition a varying membrane potential in the dendrites and the soma causes a longitudinal current to flow along the dendrite.

The axonal and dendritic branches are long and thin and can be viewed as narrow cables. Because the cables are thin enough to neglect radial variations in the membrane potential, the membrane potential in a cable is a function a longitudinal spatial coordinate and time, $V(x,t)$. Cable theory is then mathematical modelling of the membrane potential in neurons.

Since current flows down a voltage potential gradient, the longitudinal current is given by Ohm's law. For a segment of cable, the longitudinal current I_{L} and the potential gradient are related by

$$\Delta V = V_2 - V_1 = -I_{\mathrm{L}} R_{\mathrm{L}}, \tag{3.6}$$

where R_{L} is the longitudinal resistance. The longitudinal resistance is proportional to the length of the cable segment and inversely proportional to the cross-sectional area of the segment. The longitudinal resistance is then

$$R_{\mathrm{L}} = r_{\mathrm{L}} \frac{L}{\pi a^2}, \tag{3.7}$$

where r_{L} is the constant of proportionality and is known as the intracellular resistivity and L and a are the length and radius of the cable segment respectively. A cable of length Δx is taken so that the potential difference across this length is

$$\Delta V = V(x + \Delta x) - V(x). \tag{3.8}$$

A convention is defined for the direction of the current flow: current that flows in the increasing x direction is positive [27]. The longitudinal current and potential difference are then related by

$$\Delta V = -I_{\mathrm{L}} R_{\mathrm{L}}, \tag{3.9}$$

and combining this relation with Equation 3.7 for a segment of length Δx gives the longitudinal current as

$$I_{\mathrm{L}} = -\frac{\pi a^2}{r_{\mathrm{L}}} \frac{\Delta V}{\Delta x}. \tag{3.10}$$

For an infinitesimally short cable, the limit is as $\Delta x \to 0$. The ratio ΔV to Δx then becomes a partial derivative $\partial V/\partial x$, as V can also depend on time [27]. The longitudinal current becomes

$$I_{\mathrm{L}} = -\frac{\pi a^2}{r_{\mathrm{L}}} \frac{\partial V}{\partial x}. \tag{3.11}$$

The current that flows across the membrane can do so through ion channels or an electrode. The former is expressed as the membrane current (per unit area) i_{m} multiplied by the surface area of the segment. Likewise, the current that enters the neuron through an electrode will be expressed as an electrode current per unit area i_{e}. The total electrode current is i_{e} multiplied by the surface area of the neuron. These currents, together with the longitudinal current described earlier, all contribute to the changing of the membrane potential V.

The capacitance determines the rate of change of the membrane potential due to these current contributions. The capacitance of a neuronal membrane is proportional to the size of the membrane in the cell, i.e. its surface area. The proportionality constant is the specific membrane capacitance c_{m}. For a cylindrical segment of length Δx and radius a with a specific membrane capacitance c_{m}, the capacitance is $2\pi a \Delta x c_{\mathrm{m}}$. The amount of current required to change the membrane current is then

$$2\pi a \Delta x c_{\mathrm{m}} \frac{\partial V}{\partial t}. \tag{3.12}$$

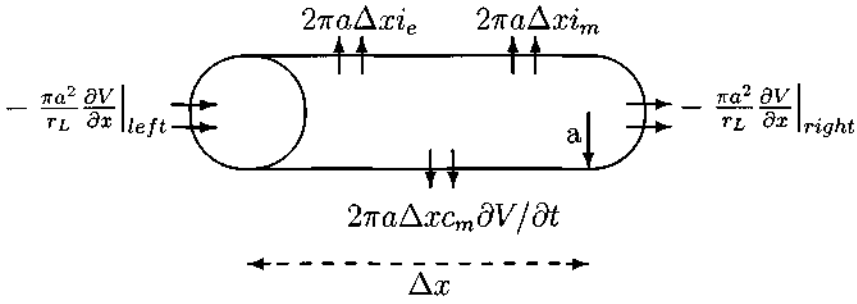


Figure 3.4:
Example of a cable segment of length Δx and radius a, showing the currents entering the cable segment. The current entering from other cable segments is in the form $-\pi a^2/r_{\mathrm{L}}$, current due to ion and synaptic channels is $2\pi a\Delta x i_{\mathrm{m}}$ and current entering through an electrode is $2\pi a\Delta x i_{\mathrm{e}}$. The membrane potential changes due to charge build-up, given in the term involving the specific membrane capacitance, c_{m}.

All the current contributions are summed together giving

$$2\pi a\Delta x c_{\mathrm{m}}\frac{\partial V}{\partial t} = -\left(\frac{\pi a^2}{r_{\mathrm{L}}}\frac{\partial V}{\partial x}\right)\Big|_{\mathrm{left}} + \left(\frac{\pi a^2}{r_{\mathrm{L}}}\frac{\partial V}{\partial x}\right)\Big|_{\mathrm{right}} - 2\pi a\Delta x(i_{\mathrm{m}} - i_{\mathrm{e}}) \qquad (3.13)$$

Both sides of the equation are divided by the term $2\pi a\Delta x$ giving

$$c_{\mathrm{m}}\frac{\partial V}{\partial t} = \frac{1}{2\pi a}\frac{1}{\Delta x}\left[\left(\frac{\pi a^2}{r_{\mathrm{L}}}\frac{\partial V}{\partial x}\right)\Big|_{\mathrm{right}} - \left(\frac{\pi a^2}{r_{\mathrm{L}}}\frac{\partial V}{\partial x}\right)\Big|_{\mathrm{left}}\right] - (i_{\mathrm{m}} - i_{\mathrm{e}}). \qquad (3.14)$$

The term involving the longitudinal currents is taken in the limit $\Delta x \to 0$ so that

$$\frac{1}{2\pi a}\frac{1}{\Delta x}\left[\left(\frac{\pi a^2}{r_{\mathrm{L}}}\frac{\partial V}{\partial x}\right)\Big|_{\mathrm{right}} - \left(\frac{\pi a^2}{r_{\mathrm{L}}}\frac{\partial V}{\partial x}\right)\Big|_{\mathrm{left}}\right] \to \frac{1}{2\pi a}\frac{\partial}{\partial x}\left(\frac{\pi a^2}{r_{\mathrm{L}}}\frac{\partial V}{\partial x}\right), \qquad (3.15)$$

as $\Delta x \to 0$. Substituting this result leads to the cable equation

$$c_{\mathrm{m}}\frac{\partial V}{\partial t} = \frac{1}{2\pi a r_{\mathrm{L}}}\frac{\partial}{\partial x}\left(a^2\frac{\partial V}{\partial x}\right) - i_{\mathrm{m}} + i_{\mathrm{e}}. \qquad (3.16)$$

The intracellular resistivity r_{L} can be taken out of the derivative as long as it is independent of position. The radius of the segment a may depend on x. The cable

equation is solved with appropriate boundary conditions. These include points at the end of a cable or where the cable branches into other cables. For points at the end of a cable, one possible option is to have no current flow out of the end of the cable. Other boundary conditions can be used as well. For a point where a cable branches out, charge must be conserved and the membrane potential V must be continuous. The following section describes how to solve the cable equation using multi-compartments.

3.4 Multi-Compartment Neurons

There are special cases for which the cable equation can be solved analytically. To look at more complex problems, for instance, dendrites with complicated branching structures and different radii, the membrane potential must be solved numerically [29]. The neuron is broken up into various compartments, as before, and the membrane potential $V(x,t)$ becomes a set of discrete, localised values. Thus each compartment has its own value for the membrane potential. Each compartment is assumed to be small enough so that the potential does not vary across the compartment.

The membrane potential for each compartment is found using an equation in the form of Equation 3.5. Thus there is a set of coupled differential equations to be solved numerically. The complexity of the multi-compartment model is determined

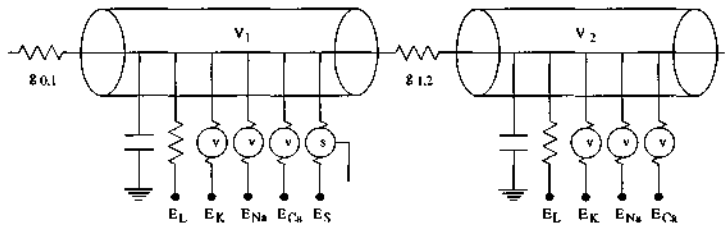


Figure 3.5:
Equivalent circuit for a multi-compartment neuron, showing two compartments. Each compartment contains active conductances (while one compartment contains an example of a synaptic conductance, with a reversal potential of E_S).

by the number of different ion channels and synaptic conductances that are included in each compartment.

The rate of change of the membrane potential for each individual compartment is given by summing the different currents that enter that compartment. For a compartment μ, the membrane current i_m^μ is summed with the current contributions from compartments $\mu + 1$ and $\mu - 1$. This is given by a coupling to each adjacent compartment and the membrane equation takes the form

$$c_m \frac{dV_\mu}{dt} = -i_m^\mu + \frac{I_e^\mu}{A_\mu} + g_{\mu,\mu+1}(V_{\mu,\mu+1} - V_\mu) + g_{\mu,\mu-1}(V_{\mu,\mu-1} - V_\mu), \qquad (3.17)$$

where the current injected into compartment μ is I_e^μ and the compartment has a surface area A_μ. The last two terms correspond to the two adjacent compartments. If a compartment branches into any number of other compartments, then there must be an equal number of coupling terms, i.e. one coupling term for every compartment that connects to the current one.

The coupling between adjacent compartments μ and μ' is described by the lon-

gitudinal conductance $g_{\mu,\mu'}$. The resistance between compartments μ and μ' is given by the intracellular resistivity r_{L}, the distance between the compartments and the cross-sectional area of the compartments (from equation 3.7). For two compartments μ, with length L_μ and radius a_μ, and μ', with corresponding length $L_{\mu'}$ and radius $a_{\mu'}$, the resistance between them is found by adding the resistances at the centre of each compartment:

$$\frac{r_{\mathrm{L}} L_\mu}{2\pi a_\mu^2} + \frac{r_{\mathrm{L}} L_{\mu'}}{2\pi a_{\mu'}^2}. \tag{3.18}$$

The current that flows between these two compartments is

$$I_{\mu,\mu'} = \frac{2\pi a_\mu^2 a_{\mu'}^2}{r_{\mathrm{L}}(L_\mu a_{\mu'}^2 + L_{\mu'} a_\mu^2)}(V_{\mu'} - V_\mu). \tag{3.19}$$

This must then be divided by the total surface area of compartment μ, which is $2\pi a_\mu L_\mu$ so that the term is given as a current per unit area. The current then becomes

$$i_{\mu,\mu'} = \frac{a_\mu a_{\mu'}^2}{r_{\mathrm{L}} L_\mu (L_\mu a_{\mu'}^2 + L_{\mu'} a_\mu^2)}(V_{\mu'} - V_\mu), \tag{3.20}$$

and it can be seen that $g_{\mu,\mu'}$ is

$$g_{\mu,\mu'} = \frac{a_\mu a_{\mu'}^2}{r_{\mathrm{L}} L_\mu (L_\mu a_{\mu'}^2 + L_{\mu'} a_\mu^2)}. \tag{3.21}$$

With this value for $g_{\mu,\mu'}$ the membrane current and voltage for the entire neuron is computed by solving Equation 3.17 for every compartment. Simulation programs such as NEURON [7] contain integration techniques that are used to create complex neuron models with large numbers of compartments.

Chapter 4

NEURON Simulation Tool

4.1 NEURON

The simulation tool NEURON [7] was developed by M.L. Hines and N.T. Carnevale to allow one to implement realistic models of neurons and networks of neurons. It is designed to specifically simulate neurons in such a way that a user can control how detailed a model is to be [30].

A simulation program such as NEURON has to solve the cable equation numerically, namely

$$c_{\mathrm{m}} \frac{\partial V}{\partial t} = \frac{1}{2\pi a r_{\mathrm{L}}} \frac{\partial}{\partial x} \left(a^2 \frac{\partial V}{\partial x} \right) - i_{\mathrm{m}} + i_{\mathrm{e}}. \tag{4.1}$$

As discussed previously, the cable equation is generally solved numerically. Neurons have tree-like structures, therefore various equations of the kind above are combined using appropriate boundary conditions. NEURON is very well suited to solve for the branching structure of neurons. The neuron is divided into many compartments,

each with its own membrane potential and parameters. This gives rise to a set of differential equations of the form

$$c_{\mathrm{m}_j}\frac{dV_j}{dt} + I_{\mathrm{ion}_j} = \sum_k \frac{V_k - V_j}{R_{jk}}, \tag{4.2}$$

where the capacitive component of the current ($c_{\mathrm{m}_j}dV_j/dt$) and the current through all the ion channels, I_{ion_j}, add up to give the total membrane current of compartment j. Here, V_j is the membrane voltage of compartment j and c_{m_j} is its membrane capacitance. The right-hand side of the equation is the total amount of current entering the compartment from neighbouring compartments.

To solve Equation 4.2 the membrane potential for the next time step Δt is found using the equation

$$V_j(t + \Delta t) = V_j(t) + \Delta V_j. \tag{4.3}$$

Rearranging Equation 4.2 updating the value of the potential is done using

$$\Delta V_j = \frac{1}{c_{\mathrm{m}_j}}\left[\sum_k g_{jk}(V_k(t) - V_j(t)) - I_{\mathrm{ion}_j}\right]\Delta t. \tag{4.4}$$

To improve on stability and accuracy the potential is found at a later time $t + z\Delta t$, therefore

$$\Delta V_j = \frac{1}{c_{\mathrm{m}_j}}\left[\sum_k g_{jk}(V_k(t + z\Delta t) - V_j(t + z\Delta t)) - I_{\mathrm{ion}_j}\right]\Delta t. \tag{4.5}$$

This equation must be solved to find V_j, the membrane potential for compartment j, using the approximation

$$V_j(t + z\Delta t) \approx V_j(t) + z\Delta V_j, \tag{4.6}$$

where ΔV_j is is the change in membrane potential during the time interval $z\Delta t$. Similar equations used to solve for V_k. The parameter z defines how many time steps forward in time the potentials V_j are calculated.

NEURON makes use of two integration methods, the backward or reverse Euler method and the central difference or Crank-Nicholson method. The reverse Euler method uses the value $z = 1$, whereas the Crank-Nicholson method uses the value $z = 0.5$. The user has a choice of using either of these methods when running a simulation.

The reverse Euler requires the time step to be as large as possible while at the same time maintaining accuracy. This is the default integrator for NEURON. The user can set a global parameter to make use of the Crank-Nicholson method, which is more accurate for small time steps as well as being more efficient.

4.2 Simulation Environment

Model neurons are constructed using cable compartments known as sections. A special feature of NEURON is keeping aspects such as the morphology of the cell being modelled separate from numerical aspects. The user can create sections such as a soma, an axon and dendrites by using simple commands. These sections can then be connected together, the user being able to specify the manner in which the

sections are joined together.

Each section has its own properties that need to be specified. Such properties include, for example, the length or diameter of the section or the number of segments the section will contain. The compartmentalisation parameter **nseg** defines how many subsections each section will be divided into. This is useful for having diameter variation over the length of a section. Splitting up a section into segments also allows for cell properties to change for different positions. Each segment has a centre or node; the internal voltage is defined at this point. The node is also the point where the transmembrane currents (for the whole surface area of the segment) are given. A specific section can be accessed for these properties to be set.

Sections in a modelled cell can also contain different Hodgkin-Huxley ion channels as well passive conductances. An insert command is used to add such mechanisms either to all or a specific section. The user is able to set parameters for ion channels within a given section, i.e. these parameters can vary for different sections.

The morphology or geometry of a cell may be specified using a list of 3-D spatial positions, which includes defining the diameter of the section. The NEURON program uses the positions list to create an anatomical reconstruction of a neuron. The cells from Mainen and Sejnowski [8] used this type of cell reconstruction to study the effect of morphology on firing rate.

The total time for the course of the simulation is then specified by the user. The time step can also be specified. The equations solved by NEURON's integration schemes are solved for the course of the time step. This step is repeated for the duration of the simulation, after every time step interval.

The hoc interpreter is used for the high level language NEURON incorporates for writing simulation programs. It contains pre-compiled code for certain computational procedures, such as integration of the cable equation or providing graphical results. All of the features mentioned previously can be defined using the hoc interpreter at the command prompt, or in .hoc files that are run in NEURON.

Chapter 5

Extracellular Potentials

5.1 Introduction

Action potentials generated in neurons cause potentials in the extracellular medium around them. Laminar electrodes measure neuronal activity experimentally by recording the electrical potentials in the extracellular medium at different cortical depths. Extracellular potentials can be calculated from action potentials from model neurons. First, the current sources from the neuron are computed and these are then used to calculate the extracellular potential at various positions. In Chapter 6 the extracellular potential is calculated around a single neuron. In Chapter 7 a population of identical neurons is also simulated and the extracellular potential is calculated for the population.

5.2 Quasi-Static Approximation

There are three separate assumptions that are made. The first assumption is based on the quasi-static approximation [10], relating the electric field and potential

$$\mathbf{E} = -\nabla V. \tag{5.1}$$

Here, $\mathbf{E}$ is the electric field and V the potential.

Secondly, the current density in the extracellular medium is a result of a current source density, namely [10]

$$\nabla \cdot \mathbf{J} = I_{\mathrm{m}}, \tag{5.2}$$

where $\mathbf{J}$ is the current density and I_{m} is the current source density. Lastly, the Lorentz force equation and the drift assumption, which says that the ionic velocity is proportional to the force, gives

$$\mathbf{J} = \sigma \mathbf{E}. \tag{5.3}$$

The constant of proportionality σ is the conductivity. It then follows that the potential around a point current source has the same form as that around a point charge, namely

$$V = \frac{I_{\mathrm{point}}}{4\pi\sigma_{\mathrm{ex}} r}, \tag{5.4}$$

where V is the potential at a distance r from the current source I_{point}. The conductivity σ_{ex} is the conductivity of the medium outside of a neuron.

5.3 Potential Around A Current Disk

To calculate the potential around a population of neurons, the potential is calculated first due to a current disk and then due to a cylinder. The potential at a point along the centre axis normal to the plane of a ring is

$$V_{\text{ring}} = \frac{1}{4\pi\sigma_{\text{ex}}} \frac{I_{\text{ring}}}{\sqrt{r^2 - (z - z_0)^2}}, \tag{5.5}$$

where z is the position of the extracellular potential being calculated, z_0 is the z-coordinate of the centre of the ring and r is the ring radius. The contribution to the potential from a differential ring is

$$dV_{\text{ring}} = \frac{1}{4\pi\sigma_{\text{ex}}} \frac{dI_{\text{ring}}}{\sqrt{r^2 - (z - z_0)^2}}, \tag{5.6}$$

where dI_{ring} can be written as $I_{\text{d}}dA$ where the current is once again expressed as a current density

$$I_{\text{d}} = \frac{I_{\text{disk}}}{\pi R_{\text{pop}}^2}, \tag{5.7}$$

where I_{disk} is the total current source in the disk of area πR_{pop}^2, R_{pop} is the radius of the population and dA is the infinitesimal area of the ring $dA = 2\pi r dr$. This can be done with the assumption that the current sources are in a homogeneous planar distribution in the cylinder. This gives

$$dV = \frac{1}{4\pi\sigma_{\text{ex}}} \frac{I_{\text{disk}}}{\pi R_{\text{pop}}^2} \frac{2\pi r\, dr}{\sqrt{r^2 + (z - z_0)^2}}. \tag{5.8}$$

Integrating from the centre $r = 0$ to the radius of the population $r = R_{\text{pop}}$ gives

$$V_{\text{disk}} = \frac{I_{\text{disk}}}{\pi R_{\text{pop}}^2} \frac{1}{2\sigma_{\text{ex}}} \left(\sqrt{R_{\text{pop}}^2 + (z - z_0)^2} - |z - z_0| \right). \tag{5.9}$$

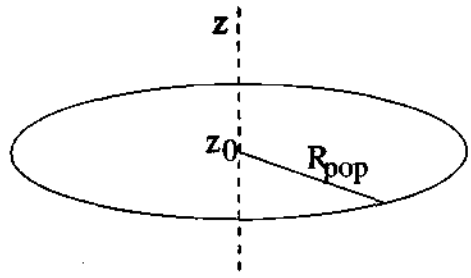


Figure 5.1:
The extracellular potential due to a current disk of radius of R_{pop} is calculated along the centre axis of the disk at the point z using Equation 5.9. The point z_0 is the position at the centre of the ring.

Figure 5.1 shows that the potential V_{disk} due to a current disk I_{disk} is calculated at a point z along the centre axis.

5.4 Extracellular Potential Around A Neuron

Equation 5.4 gives the extracellular potential a distance r away from a current source. The potential that is calculated is the result of all the transmembrane currents from the neuron in the simulation.

The currents from the nodes in the neuron are summed together using the principle of superposition. The potential at the point (x, y, z) is given by

$$V_{\text{neuron}}(x, y, z) = \sum_n \frac{I_n}{4\pi\sigma_{\text{ex}}\sqrt{(x - x_n)^2 + (y - y_n)^2 + (z - z_n)^2}} \tag{5.10}$$

where I_n is the current at the n-th node, and $n = 1, 2, ..., N$ and N is the total number of nodes in the neuron. The n-th node has position (x_n, y_n, z_n). Figure 5.2 shows that the potential V at an arbitrary position (x, y, z) is given by the contri-

Figure 5.2:
Diagram showing the point current sources of the layer 5 pyramid neuron (represented by dots) that contribute, by superposition, to the extracellular potential V calculated at some arbitrary point (x, y, z).

bution of all N current nodes.

5.5 Extracellular Potential For Neuronal Populations

A population of neurons is simulated from the data of a single neuron. The population has the following characteristics:

- the neurons in the population are situated in a cylinder in the cortex
- the centre axis of the cylinder is perpendicular to the cortex and the extracellular potentials are calculated along this axis
- the neurons are spread homogeneously in the horizontal plane of the cortical cylinder, with a cell density ρ

- the positions of the cells in the vertical direction are normally distributed (with a standard deviation σ_z)

When simulating the response of a population, the activity in the horizontal direction is assumed to be homogeneous. This means that the extracellular potential is calculated as a function of the depth of the cortex (i.e. the vertical z-direction).

The current from a disk is used to simulate a population of identical neurons. For a disk of infinitesimal thickness dz_0, the current is proportional to the current source density $\lambda_n(z,t)$. The current sources are discrete points, therefore the current source density is written as a continuous function by

$$\lambda(z_0,t) = \sum_n \lambda_n(z_0,t). \tag{5.11}$$

The linear current source density function λ_n is given by

$$\lambda_n(z,t) = I_n(t) f_n(z), \tag{5.12}$$

where $I_n(t)$ is the current from the n-th node at time t and $f_n(z)$ is the normalised Gaussian

$$f_n = \frac{1}{\sigma_z\sqrt{2\pi}} \exp\left(-\frac{(z-z_n)^2}{2\sigma_z^2}\right). \tag{5.13}$$

The Gaussian function gives a spread in the vertical alignment of the cells in the cylinder. The majority of the cells will have their somas at the same height in the cylinder, while others are either above or below, given by equation 5.13.

The current in the disk depends on the number of active cells. The density of

active cells in the disk is ρ. The current in the disk of area πR_{pop}^2 with thickness Δz around z_0 is

$$\Delta I_{\text{disk}}(z_0) = \rho \pi R_{\text{pop}}^2 \Delta z \; \lambda(z_0, t). \tag{5.14}$$

Integrating over the height of the cylinder gives an extracellular potential of

$$\begin{aligned} V(z,t) &= \int \rho \; \lambda(z_0,t) \; g(z,z_0) \; dz_0 \\ &= \rho \int \sum_n \lambda_n(z_0,t) \; g(z,z_0) \; dz_0 \end{aligned} \tag{5.15}$$

where

$$\lambda_n(z_0,t) = I_n(t) f_n(z) = \frac{I_n(t)}{\sqrt{2\pi}\sigma_z} \exp\left(-\frac{(z_0 - z_n)^2}{2\sigma_z^2}\right) \tag{5.16}$$

$$g(z,z_0) = \frac{1}{2\sigma_{\text{ex}}} \left(\sqrt{R_{\text{pop}}^2 + (z - z_0)^2} - |z - z_0|\right). \tag{5.17}$$

Such a simulation gives the potential of a synchronous population of identical neurons, i.e. all the cells are firing action potentials at the same time. This may not be a common scenario. A possibility is to use a Gaussian function such as Equation 5.13 to study a population of neurons firing at different times.

Chapter 6

Extracellular Signatures of Single-Neuron Action Potentials

6.1 NEURON Script

A NEURON script, in the hoc language, was written to simulate action potential firing in the three different cell types of Mainen and Sejnowski [8]. It was used to calculate the extracellular potential around a neuron during the firing of an action potential. The script is an extension to the NEURON simulation code provided by Mainen and Sejnowski [31] developed by Klas Pettersen and myself. Klas Pettersen extended the code to keep track of the transmembrane currents and the spatial positions of the nodes through which they enter and leave the neuron, as well as calculating the extracellular potential at any point due to all the currents. I made the code specific to each neuron type and calculated extracellular potentials from populations of cells.

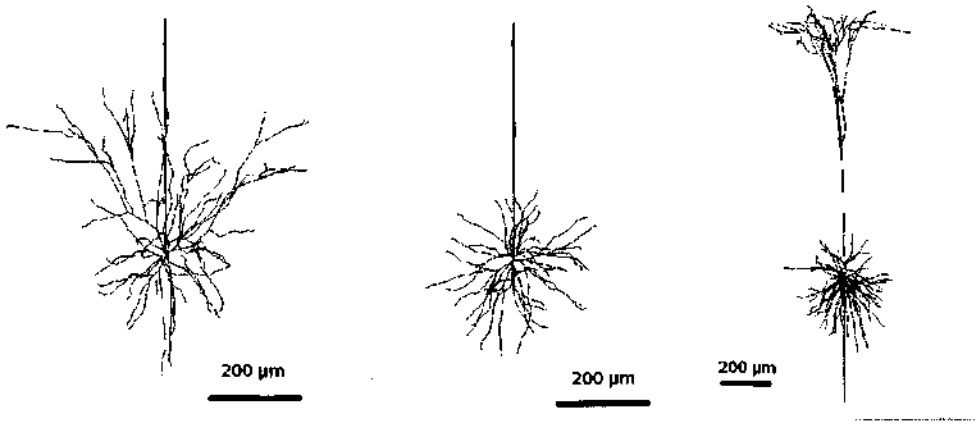


Figure 6.1:
Morphology of cat visual cortex cells (from left): a layer 3 pyramid, a layer 4 stellate and a layer 5 pyramid [8]. The straight lines are the modelled axons.

The NEURON script used in the simulation is the same for the different cells. One of the aims was to have a generic program that would be able to calculate extracellular potentials for any given neuron. The simulation consists of four parts: creating the morphology of the cell, inserting the various ion channels, running the simulation to generate an action potential in the cell and finally calculating the extracellular potential around the cell and writing this data to a file.

The cells provided by Mainen and Sejnowski [8] were created using data files provided by the authors [31]. These files contained the 3-D positions of the dendritic sections of the cells, as well as the diameters of each section. It also contains the information used to connect the various sections. The neurons were reconstructions of the following cells: a layer 3 pyramid, a layer 4 spiny stellate and a layer 5 pyramid, see Figure 6.1. All three cells are from cat visual cortex and each exhibits a

different dendritic morphology. Each section in the cell is set up in a way so that it does not have a length that exceeds 50 μm. The position of the soma was set to be at the origin.

The myelinated axon is created by the NEURON script as a long, straight cable of uniform diameter, extending straight in the vertical direction. The axon consists of an axon hillock, a section known as the initial segment, the myelinated axon and nodes of Ranvier, which are gaps in the myelinated sheath of the axon, see Figure 6.2. The axon hillock is the part of the axon that connects to the soma. It has the shape of a cone with a base that is four times as wide as the opposite end. The axon hillock is connected to the initial segment, followed by five long myelinated segments separated by the nodes of Ranvier. With the exception of the axon hillock, the entire axon is cylindrical with the same diameter in all the segments.

The neurons differed only in their structure and size and in the number of compartments. The neurons were constructed so that the dendrites were cylindrical compartments or segments, where no compartment exceeded 50 μm in length. The layer 3 pyramid and layer 4 stellate were approximately 800 μm in height and were made up of 257 and 199 compartments respectively. The layer 5 pyramidal cell was the largest of the three, approximately 1200 μm in height. It was made up of 479 compartments in the simulation. The diameters of each section varied throughout the neurons. The section diameters were obtained from the source data files.

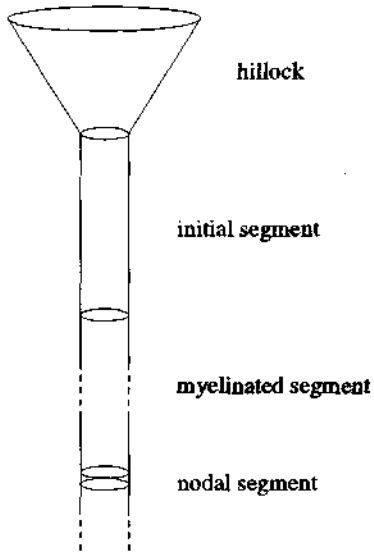


Figure 6.2:
Diagram of an axon, as it extends from the soma of the neuron. The conical axon hillock connects with the soma.

A current node is set up for each segment, see Figure 6.3. The node is a point in 3D space at the centre of the segment. The positions for every node are stored in a vector array. During the simulation, transmembrane currents are computed at these node points. The current nodes act as sources or sinks of current, which will give rise to the extracellular potential. The current from the node will be the total of the contributions from the different ion channels in the segment.

Once the cell structures are set up, the second part of the simulation involves inserting ion channels in the required compartments.

The original model used different voltage-dependent conductances that were inserted in different densities for each of the various segments. Sodium, potassium and cal-

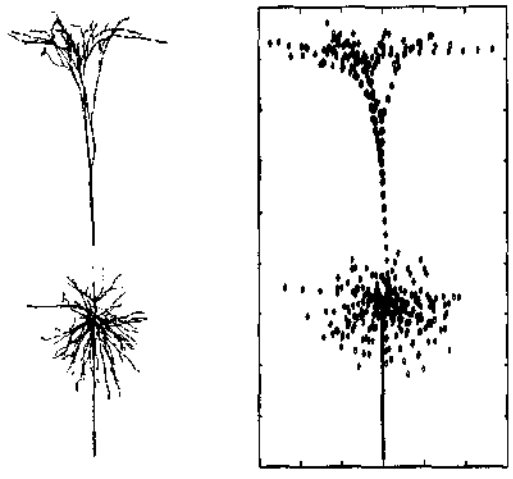


Figure 6.3:
The layer 5 pyramid cell [8] (left) and the node positions for the NEURON simulation of that cell. Transmembrane currents are computed at the node points shown.

cium channels were used. Fast Na^+ channels were inserted at a high density in the axon hillock and initial segment and at a low density in the soma and the dendrites. Fast K^+ channels were inserted in the axon and soma but not in the dendrites. Slow K^+ channels were used in the soma and dendrites (to generate more than one spike [8]). The soma and dendrites also included Ca^{2+}-dependent and high-threshold Ca^{2+} channels.

The model included an electrode in the soma that would cause it to generate an action potential. The electrode was removed and the method by Holt and Koch [9] was used to generate action potentials, where a new conductance was scattered throughout the cell to mimic a time-averaged synaptic conductance. The value of the specific conductance was set to 0.2 times the value of the leak conductance. The synaptic conductance also had a reversal potential value of 0 mV. The use of this

synaptic conductance provided the necessary current needed for the cell to fire an action potential. The time of firing an action potential was different for each neuron.

Some of the parameters that were defined for the NEURON simulation are listed in Table A.1 and Table A.2 in Appendix A. The time step in the simulation was chosen to be 0.01 ms. The initial value of the membrane potential was -70 mV.

The currents computed in each compartment of the neuron were then used to calculate the extracellular potential. The contribution to the potential from a compartment is then calculated using the equation

$$V(x,y,z) = \frac{1}{4\pi\sigma_{\text{ex}}} \frac{I_n}{\sqrt{(x-x_n)^2 + (y-y_n)^2 + (z-z_n)^2}}, \tag{6.1}$$

where the potential V is calculated at the point (x, y, z), σ_{ex} is the conductivity of the extracellular medium, I_n is the current from node n and (x_n, y_n, z_n) is the position of that node. This follows from Equation 5.4 for the potential due to a current point source.

The potential can be calculated in this manner at any instant in time; transmembrane currents at each time step are used to calculate the extracellular potential for the duration of the simulation. To replicate laminar electrode readings, this procedure is done for various spatial positions. The vertical direction (z-direction) was considered to be of special interest so the variation in potential with cortical depth above and below the neuron was emphasised in the studies.

The simulations typically ran for a time of 100 ms. During this time, several action potentials were fired by the cells, the number being different between the three neurons. (The layer 4 stellate cell fired two action potentials in the 100 ms period; three and four action potentials were fired by the layer 3 and 5 pyramids respectively, see Figure 6.4.) An action potential has a typical duration of about 1 ms. The extracellular potentials that will be shown in the results were calculated around the first action potential in each of the simulations. The potentials will be shown over a 3 ms time span around the time the action potential was fired by the cell.

The extracellular potential was obtained from the currents at the nodes in the cells. The node currents are stored in a vector array in the simulation. The NEURON script writes data from a simulation to a file for the extracellular potential to be calculated from these data files. The file contains a header with information about the simulation. The data it contains is the time step used in the simulation, the 3-D positions of the nodes in the cell and the current from each node at every time step in the simulation.

The data files produced by the NEURON simulations were used to calculate the extracellular potential around single neurons (for the three neurons by Mainen and Sejnowski). Extracellular potentials around populations of neurons were also calculated using the data files. The results of neuron population simulations are presented in Chapter 7.

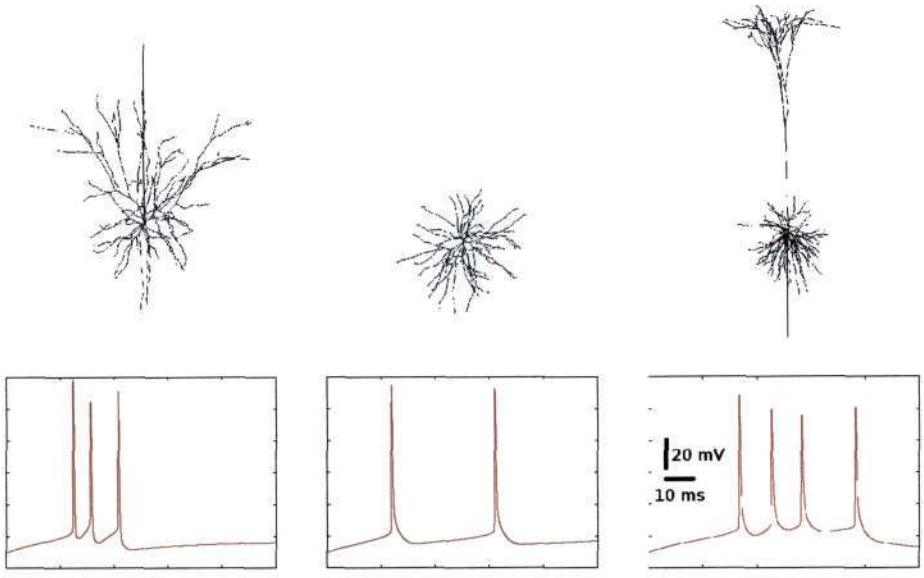


Figure 6.4:
Three cortical cell types used in the simulation, together with the soma potential for each cell: a layer 3 pyramid (left), a layer 4 stellate (middle) and a layer 5 pyramid (right) [8]. The resting potential for each cell is -70 mV. The plots show the spiking (firing of an action potential) rate of each cell, over a 100 ms simulation.

The results for simulations of single cells are presented here. Extracellular voltage traces are shown for various positions around three different types of neurons. In addition to the full extracellular potential, we also show results for high-band filtered potentials. The computed voltages are then filtered (750 - 3000 Hz) using a second-order Butterworth filter to mimic the multiple-unit activity (MUA).

An extracellular conductivity of 0.3 S/m [32] was chosen, but this number will only affect the overall amplitude of the calculated extracellular potentials. Each neuron had a resting membrane potential of -70 mV. Figure 6.4 shows the action potentials fired by each of the neurons used in the simulations.

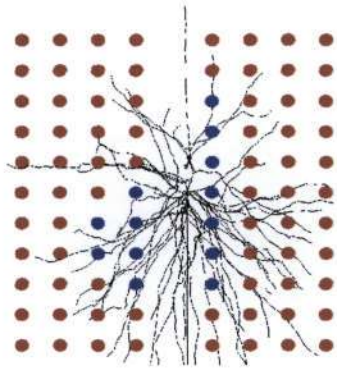


Figure 6.5:
Plot to show the positions of the calculated extracellular potentials around a layer 5 pyramid. The extracellular traces are shown in Figure 6.6. There are two scales for the traces, indicated by the two colours. The points are 50 μm apart horizontally and 40 μm apart vertically.

6.2 Layer 5 Neuron

6.2.1 Extracellular Potential Close to the Soma

The extracellular potential was calculated for points close to the soma of a layer 5 pyramid cell, partially in order to compare with results from other simulations [9]. The extracellular potential was calculated for a set of grid points around the soma. The positions of the grid points are shown in Figure 6.5, and the voltage traces for each of these points is shown in Figure 6.6.

The voltage traces presented here (and from here onwards) are the extracellular potentials at various positions indicated by the coloured circles next to the picture of the neuron. The position of the circles shows the position where the extracellular potential was computed. The traces shown are the potentials around the first action

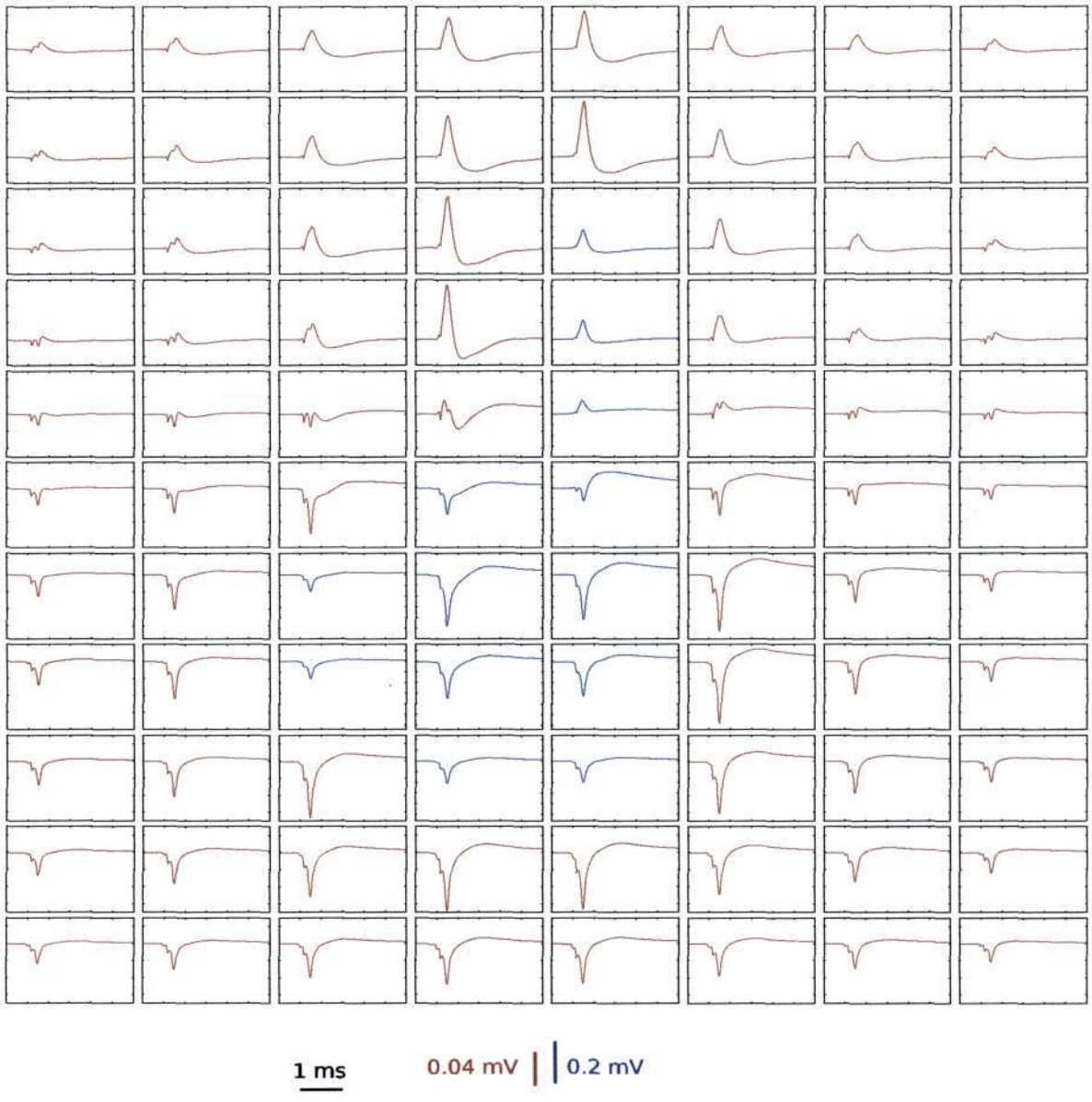


Figure 6.6:
The extracellular traces around the layer 5 pyramid in Figure 6.5. All the traces have a duration of 3ms; however there are two different voltage scales indicated by the two colours.

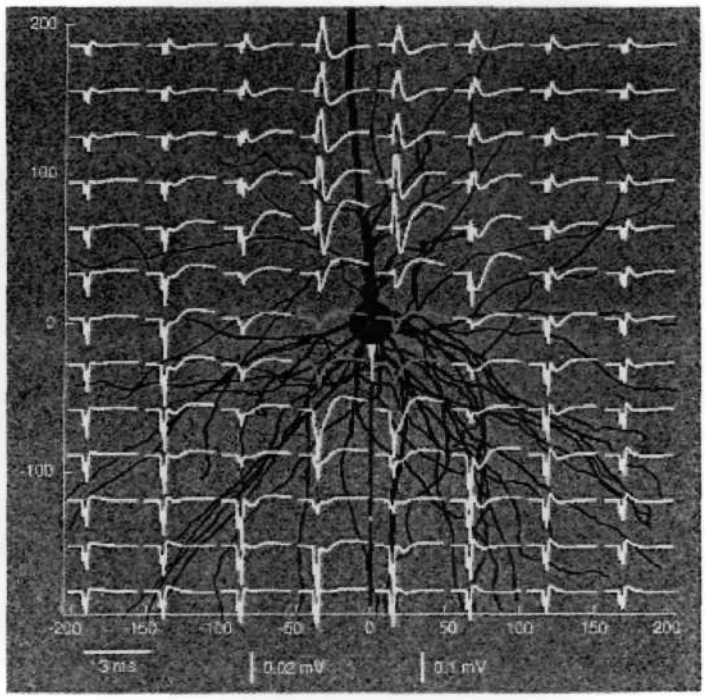


Figure 6.7:
Plots of extracellular potentials by Holt and Koch [9]. The distances in the x and y axes are in μm. The traces shown were calculated at the point at the centre of the trace.

potential fired by the respective neuron. The traces shown have a total duration of 3 ms. The traces are on two different scales, indicated by the scale bar of matching colour below the picture of the neuron.

This set of results was used to compare qualitatively with simulations done by Holt and Koch [9] in which the authors calculated extracellular potentials in the vicinity of the same layer 5 pyramid neuron used here. Note, however, that our choice of grid points is not identical to the choice of Holt and Koch. Another difference is that Holt and Koch used the line source method to calculate the extracellular potential, different to the point source method presented here. The point source method and line source method are expected to give somewhat different results close to a current

source node. For points further away from the neuron, the results will become more similar. Figure 6.7 shows the set of results by Holt and Koch for the extracellular potentials around the soma of a cortical neuron.

The plots from Figure 6.6 are found to be in good qualitative agreement with the results of Holt and Koch shown in Figure 6.7.

The largest potentials are generally found close to the soma and the base of the axon. The voltage traces exhibit one or two initial negative peaks due to currents in the initial segment and axon hillock. Current enters the neuron mainly through the axon hillock. This is seen in the large negative peaks for voltage traces close to the hillock.

As current enters the neuron in one place, by charge conservation it must leave the neuron at another point. The dendritic branches act as the main source for current to flow out of the neuron. An initial positive peak is seen clearly for this neuron, which has two main sets of dendritic branches. The initial positive peak can be seen close to the main apical dendrite, the main branch that goes roughly upwards from the soma. The extracellular traces close to the apical dendrite have not been shown.

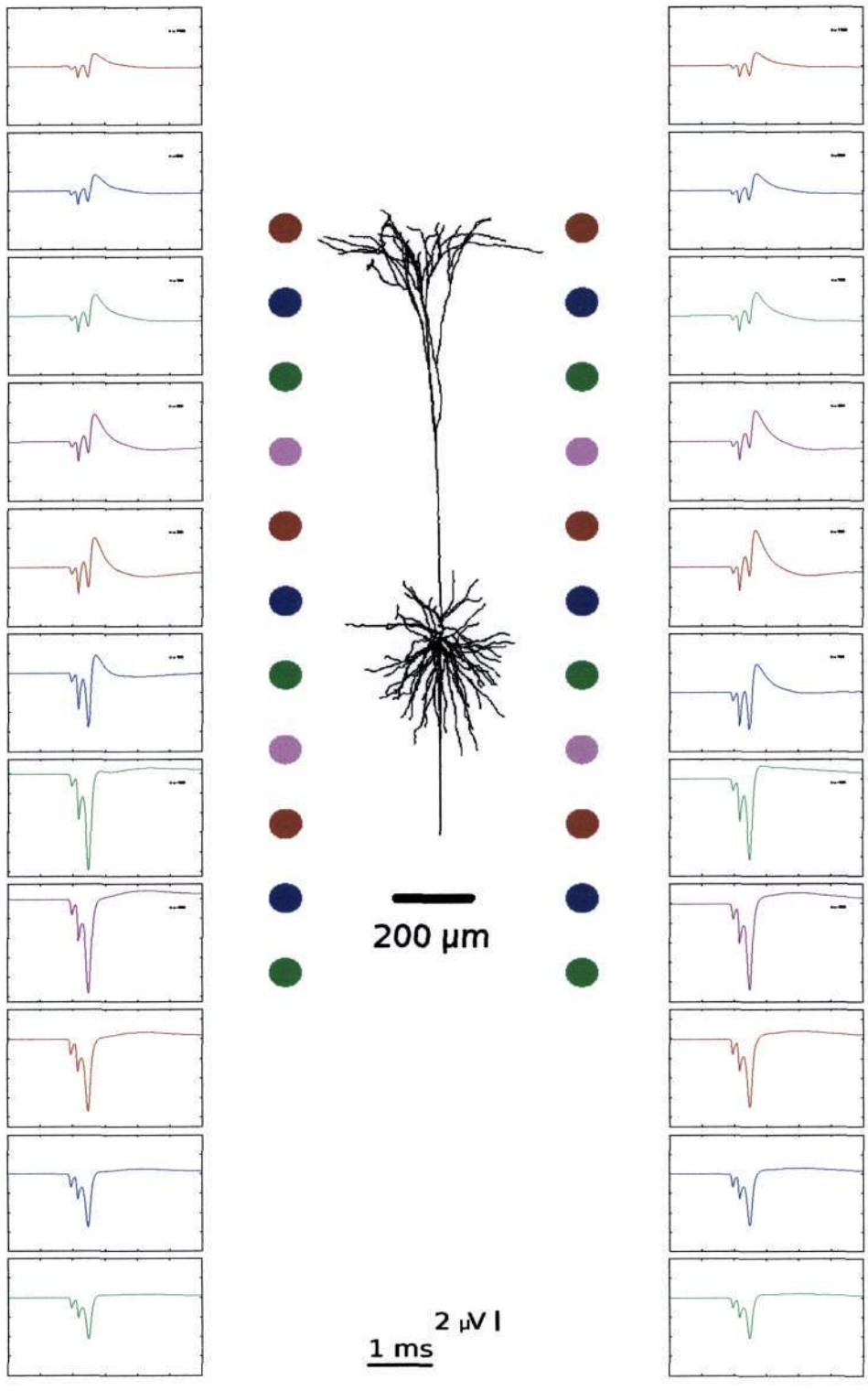


Figure 6.8:
The extracellular potential for a layer 5 pyramid cell. The extracellular potential traces were calculated at the points shown, 400 μm either side of the cell.

6.2.2 Extracellular Potential Distant From Soma

Figure 6.8 shows the extracellular potential at different heights next to the layer 5 pyramid. This shows the extracellular traces further away from the soma, and shows how the neuron can contribute to the extracellular potential recorded by more distant laminar electrodes. The potentials were calculated at 400 μm away from the neuron at either side, each position being 200 μm apart.

The potentials with the largest amplitudes are found close to the soma and below near the axon. The two initial negative peaks are seen more visibly for these positions. Above the soma, on the side of the apical dendrite, a positive peak is seen which does not occur below the soma. The potential decays for positions further away from the soma.

6.2.3 Contributions From Various Neuron Parts

Plots to show the contribution due to currents in the soma, axon hillock and initial segment are shown in Figures 6.9, 6.10 and 6.11. Extracellular potentials were calculated at different cortical depths using only currents from these parts of the cell individually. The extracellular traces due to currents from the soma, axon hillock and initial segment are each shown to the same scale.

The plots show that the currents that most dominate are those from the axon hillock

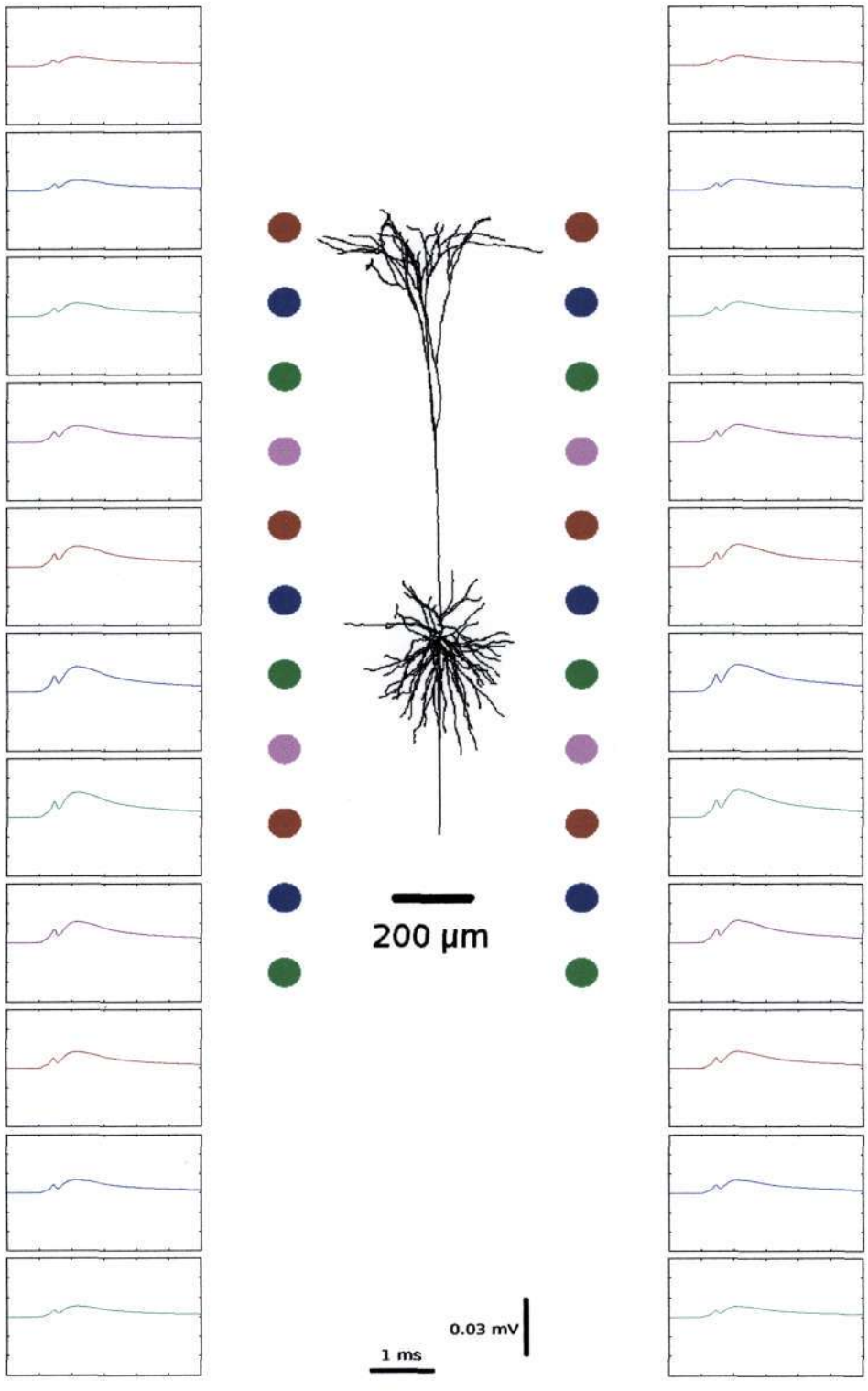


Figure 6.9:
Extracellular potential traces for a layer 5 pyramid neuron, due to transmembrane currents in the soma compartment.

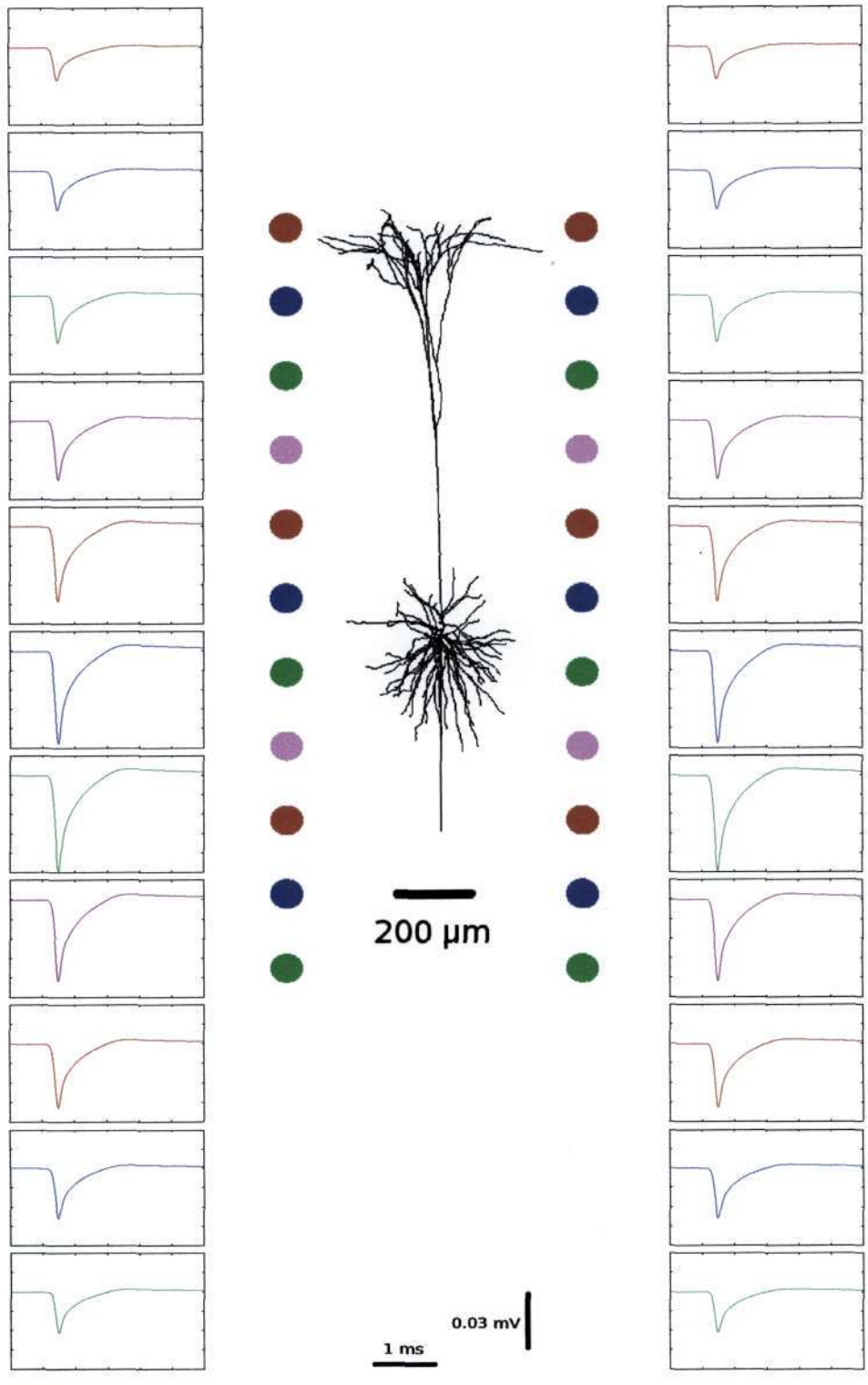


Figure 6.10:
Extracellular potential traces for a layer 5 pyramid neuron, due to transmembrane currents in the axon hillock.

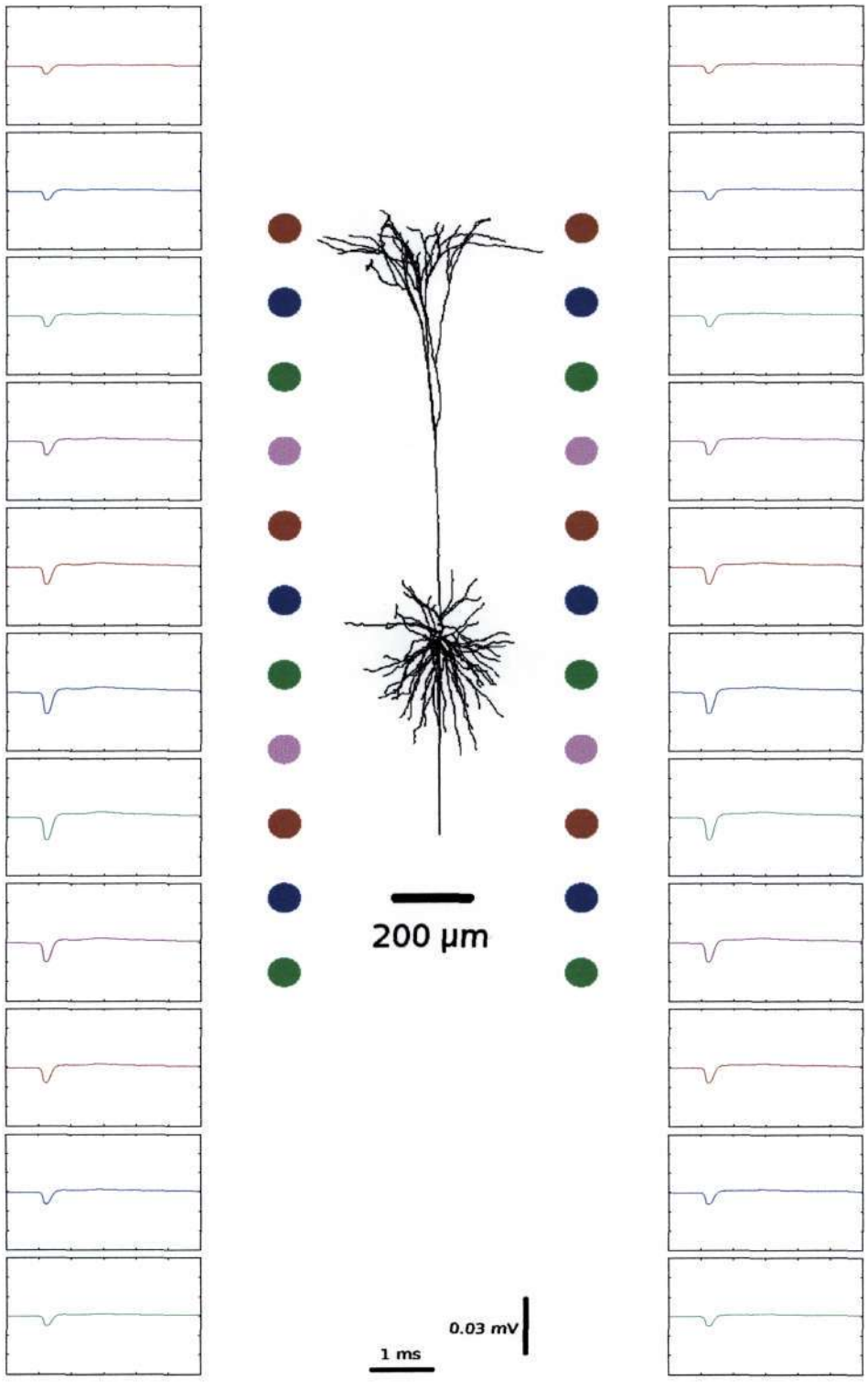


Figure 6.11:
Extracellular potential traces for a layer 5 pyramid neuron, due to transmembrane currents in the initial segment.

and, to a lesser extent, the soma. The contribution to the total extracellular potential by the initial segment is small compared to the soma and axon hillock. It contributes to the first negative peak in the extracellular trace.

Figures 6.9, 6.10 and 6.11 show larger amplitudes than for the total extracellular trace (shown in Figure 6.8). This is due to the cancellation caused by currents from the rest of the neuron. The extracellular potential due to all other currents in the neuron, i.e. currents apart from the soma, axon hillock and initial segment, has a large positive peak which is similar in amplitude to the extracellular trace due to the axon hillock (not shown).

6.2.4 High-Pass Filtered Extracellular Potential

The filtered voltage traces for the layer 5 pyramid are shown in Figure 6.12. The extracellular traces are high-pass filtered between 750 and 3000 Hz. For MUA recordings, the signals are high-pass filtered as well, showing the action potential firing of neurons.

The amplitudes of the traces are once again largest when close to the soma. Amplitudes are higher below the soma where the axon is, as compared to above the soma where the primary dendrite lies. Below the soma, the first negative peak is also evident.

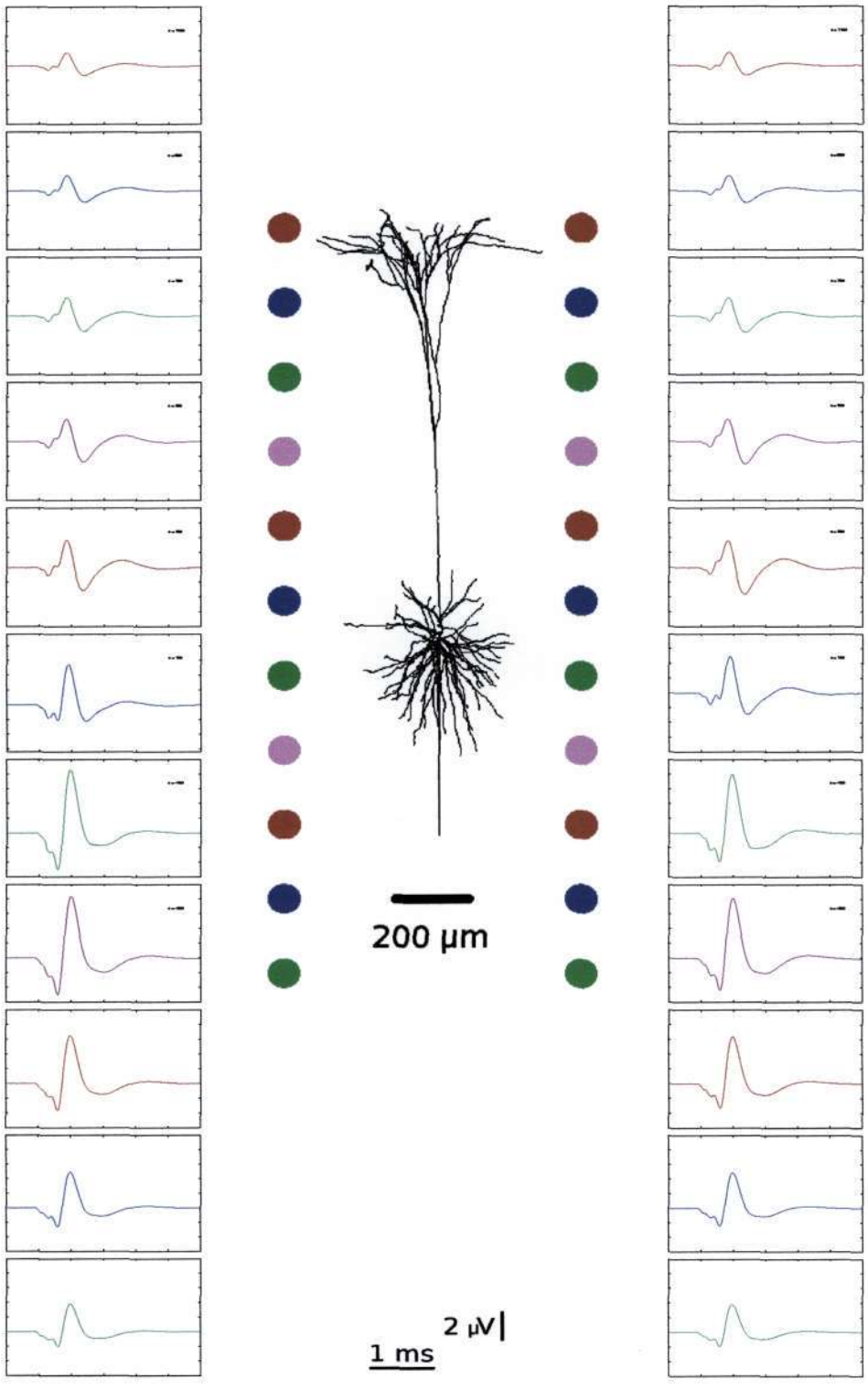


Figure 6.12:
The filtered (750 - 3000 Hz) extracellular potential for a layer 5 pyramid cell.

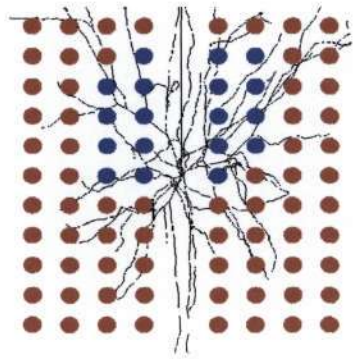


Figure 6.13:
Positions for which the extracellular potential is calculated around the soma of a layer 3 pyramid cell. Note that there are two voltage scales, indicated by the two different colours. The traces at these positions is shown in Figure 6.14. Each position is 40 μm above and below from the next position, and 50 μm to the side of the adjacent point.

6.3 Layer 3 Neuron

6.3.1 Extracellular Potential Close to the Soma

The extracellular traces around a layer 3 pyramid cell are presented here. First, the extracellular potential is calculated at at points close to the soma of the neuron, shown in Figure 6.13. The extracellular traces at these positions is shown in Figure 6.14.

The extracellular traces for points close to the soma (Figure 6.14) differ from the results for the layer 5 pyramid (Figure 6.6), in that there is no polarity difference for positions above and below the soma.

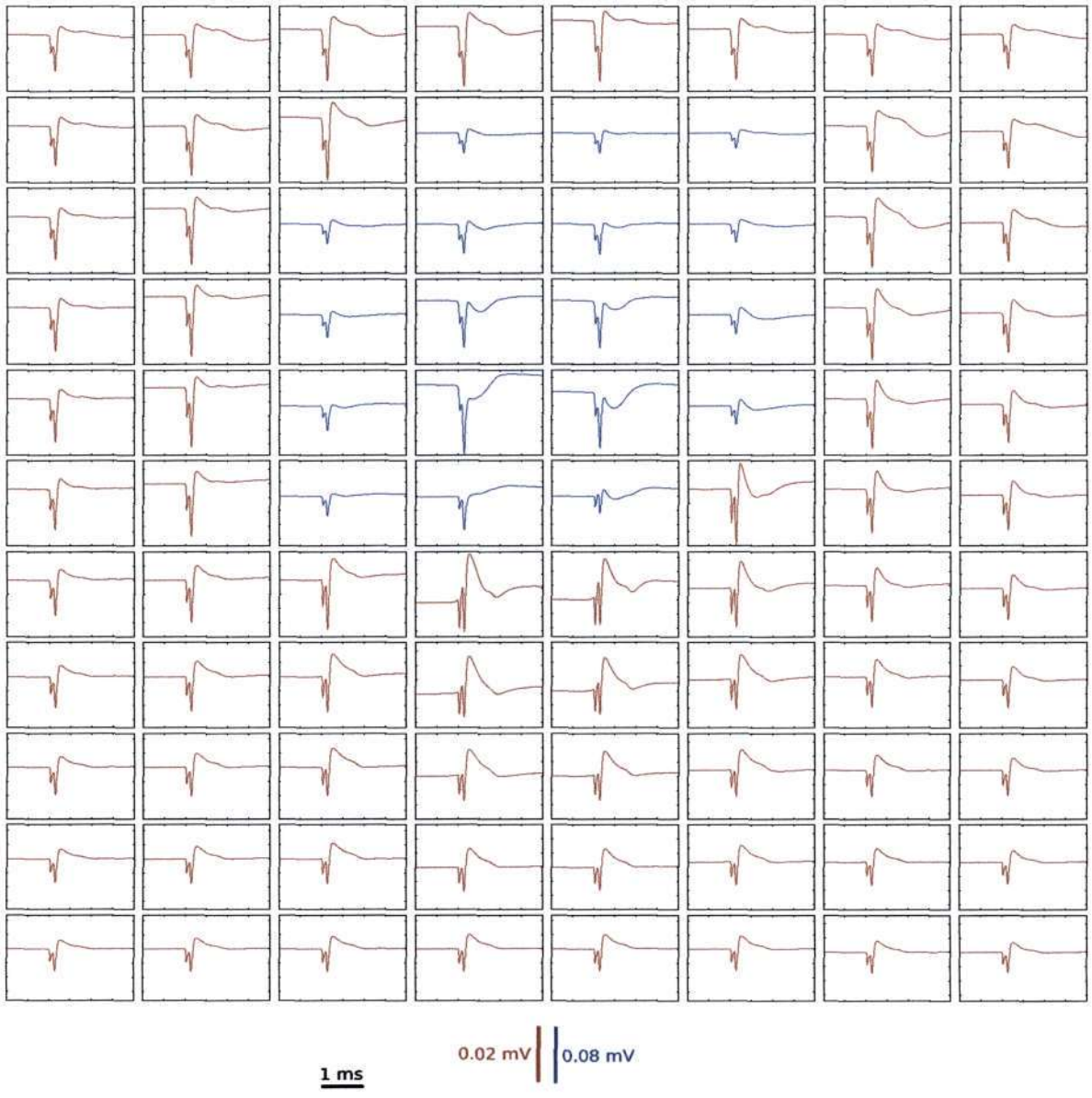


Figure 6.14:
Plots to show the extracellular potential traces around a layer 3 pyramid cell. All traces have a duration of 3 ms; note the different voltage scales.

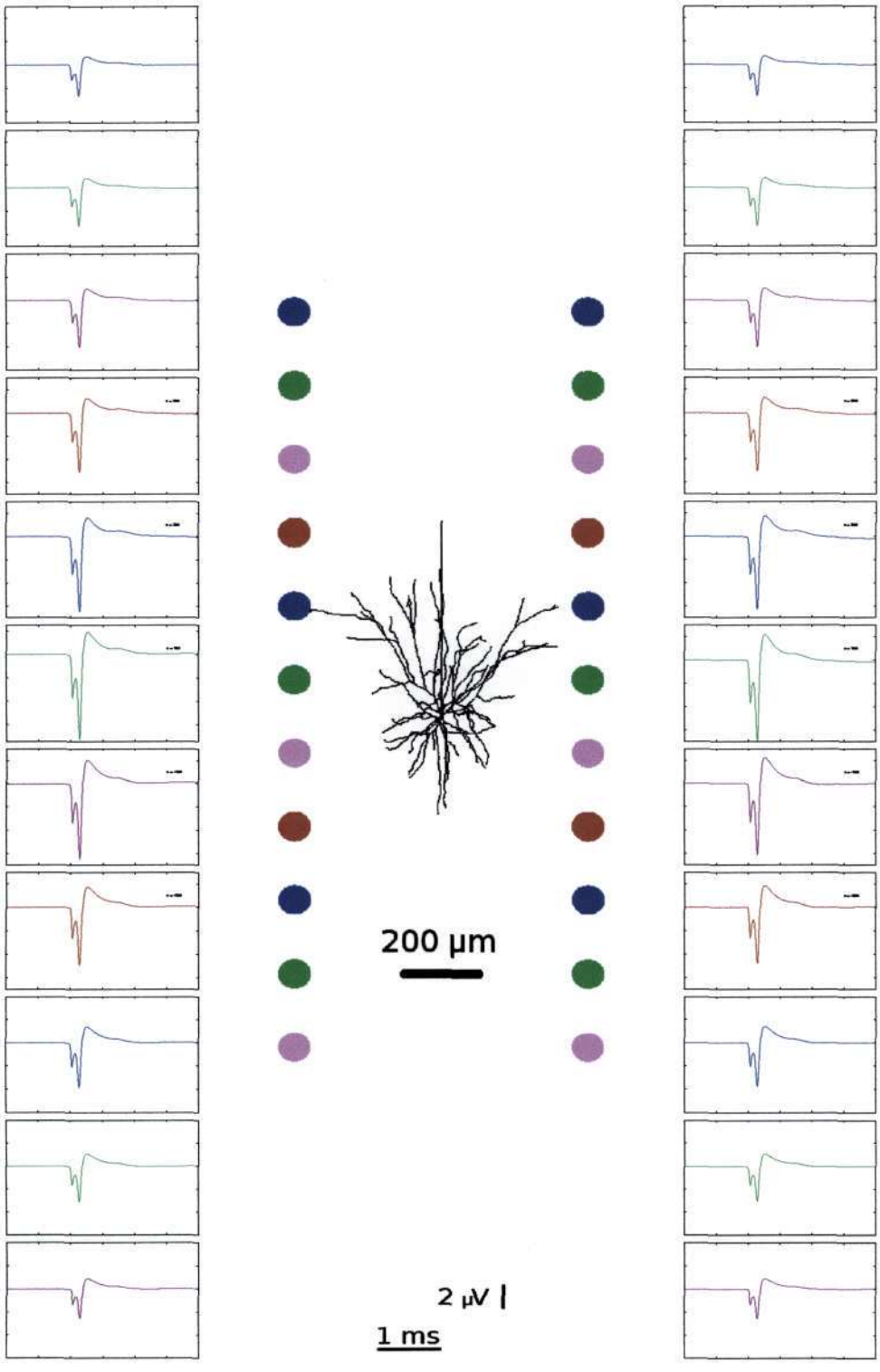


Figure 6.15:
The extracellular potential for a layer 3 pyramid cell. Extracellular potential traces are computed 400 μm from the neuron, at the points shown next to the cell, 200 μm apart from each other.

6.3.2 Extracellular Potential Distant From the Soma

Figure 6.15 shows the extracellular potential at different heights 400 μm at either side of the neuron.

The initial negative peaks and the positive peak are visible in the layer 3 neuron's traces. Once more, the largest amplitudes are close to the soma after which the potentials decay.

6.3.3 High-Pass Filtered Extracellular Potential

Figure 6.16 shows the filtered extracellular traces around the layer 3 pyramid. The traces all exhibit the same shape differing only in amplitude.

Note that the amplitudes for these traces are smaller than the filtered extracellular traces for the layer 5 pyramid (Figure 6.12).

6.4 Layer 4 Neuron

6.4.1 Extracellular Potential Close to the Soma

Extracellular traces around a layer 4 stellate cell are presented here. Figure 6.18 shows the traces close to the soma of the stellate cell.

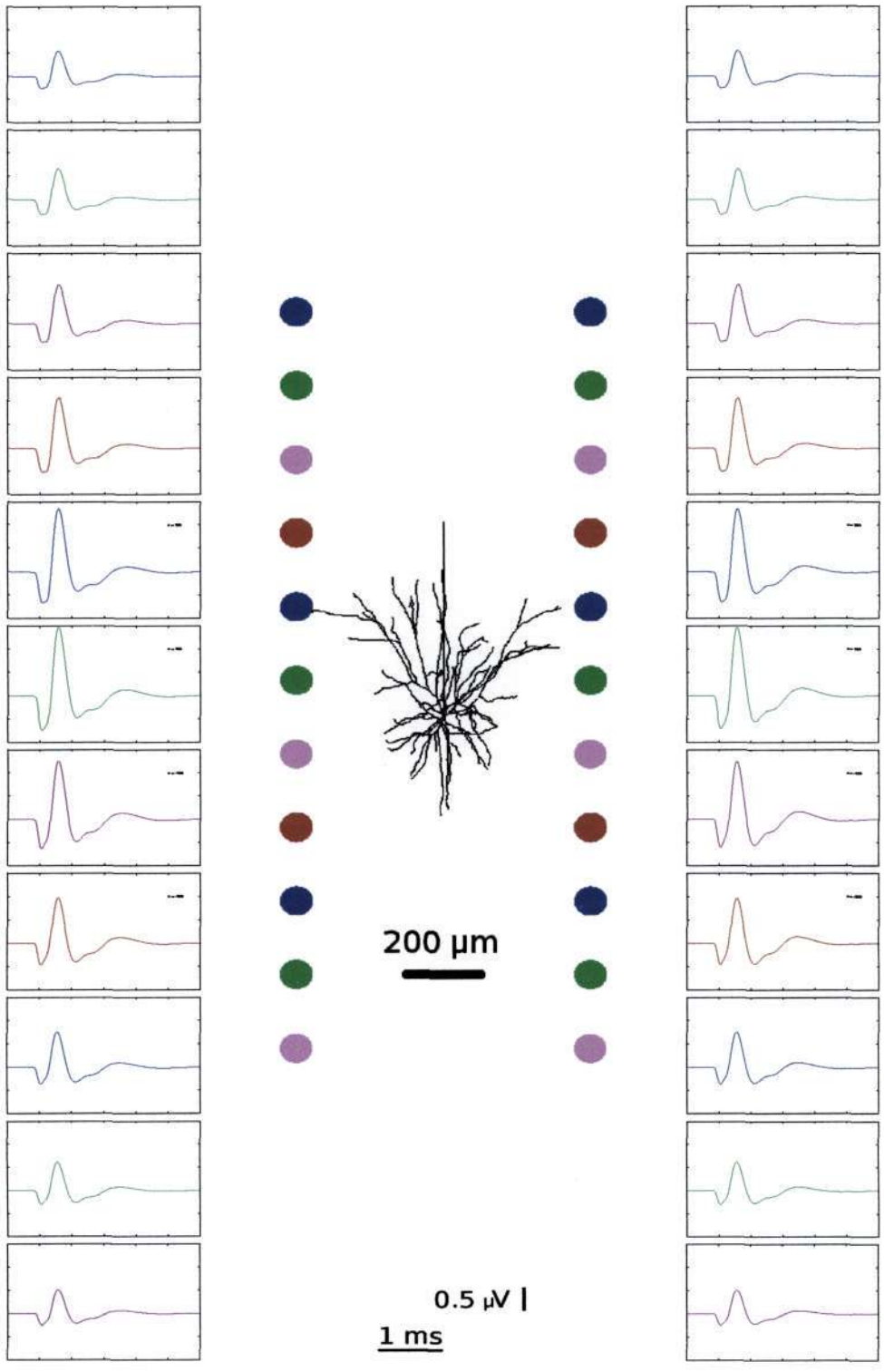


Figure 6.16:
The filtered extracellular potential for a layer 3 pyramid cell. The voltage traces from the active model were filtered (750 - 3000 Hz) using a Butterworth filter.

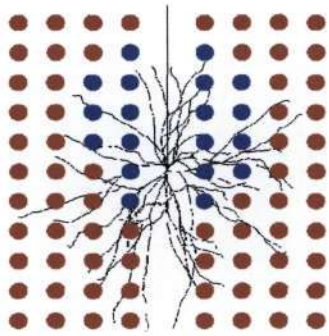


Figure 6.17:
Extracellular traces calculated at points close to the soma of a layer 4 stellate neuron. The traces are shown in Figure 6.18 at the positions given in Figure 6.17.

Figure 6.18 shows the shift in the polarity of the traces above and below the soma is less evident than for the traces around the soma of a layer 5 pyramid (Figure 6.6).

6.4.2 Extracellular Potential Distant From the Soma

Figure 6.19 shows the extracellular potential around a layer 4 stellate neuron. As with the layer 3 pyramid, the axon of this cell rises above the soma.

The potential is very small when far away from the soma. The positive peak is visible close to the soma but more for potentials below the soma. For points above the soma, on the side of the axon, the positive peak is less pronounced whereas the negative peaks are larger.

Figure 6.18:
Plots to show the extracellular potential traces around a layer 4 stellate cell. All traces have a duration of 3 ms; note the different voltage scales.

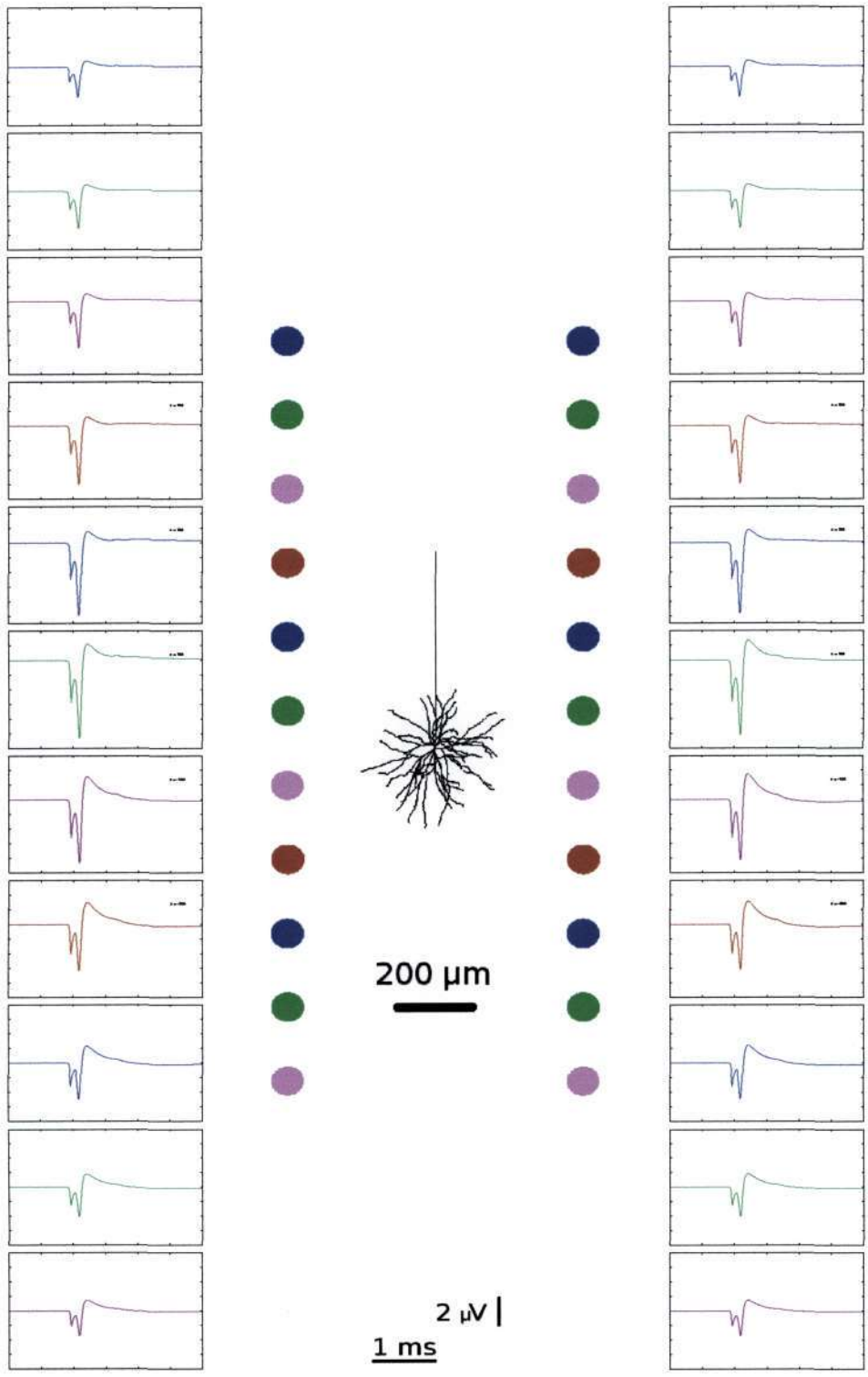


Figure 6.19:
The extracellular potential for a layer 4 stellate cell. The extracellular potential traces were calculated at the points shown next to the cell.

6.4.3 High-Pass Filtered Extracellular Potential

Figure 6.20 shows the filtered extracellular potential around a layer 4 stellate neuron. The traces have been filtered between 750 and 3000 Hz to mimic MUA recordings.

The high-pass filtered extracellular traces for the three neurons are shown in Figures 6.12, 6.16 and 6.20. Comparing the data with the unfiltered traces (Figures 6.8, 6.15 and 6.19) it can be seen that the traces decay more rapidly for the unfiltered data, moving further away from the soma. The traces have similar amplitudes at distances up to 300 μm from the soma, for both the unfiltered and filtered data, after which there is decay in the amplitude. The decay is less for the filtered data.

The fast oscillations in the early dip of the unfiltered extracellular traces is not present in the filtered traces. This may be due to the high-frequency cut-off at 3000 Hz.

6.5 Summary

Here we contrast the extracellular traces from the three different neurons. Notice that the layer 5 pyramid in Figure 6.6 has a spatial distribution that is similar to the distribution around a dipole that is positioned at the soma and oriented in the z-axis (vertical). The layer 3 pyramid (Figure 6.14) does not have such a dipole distribution. The extracellular potential close to the soma of a layer 4 stellate cell

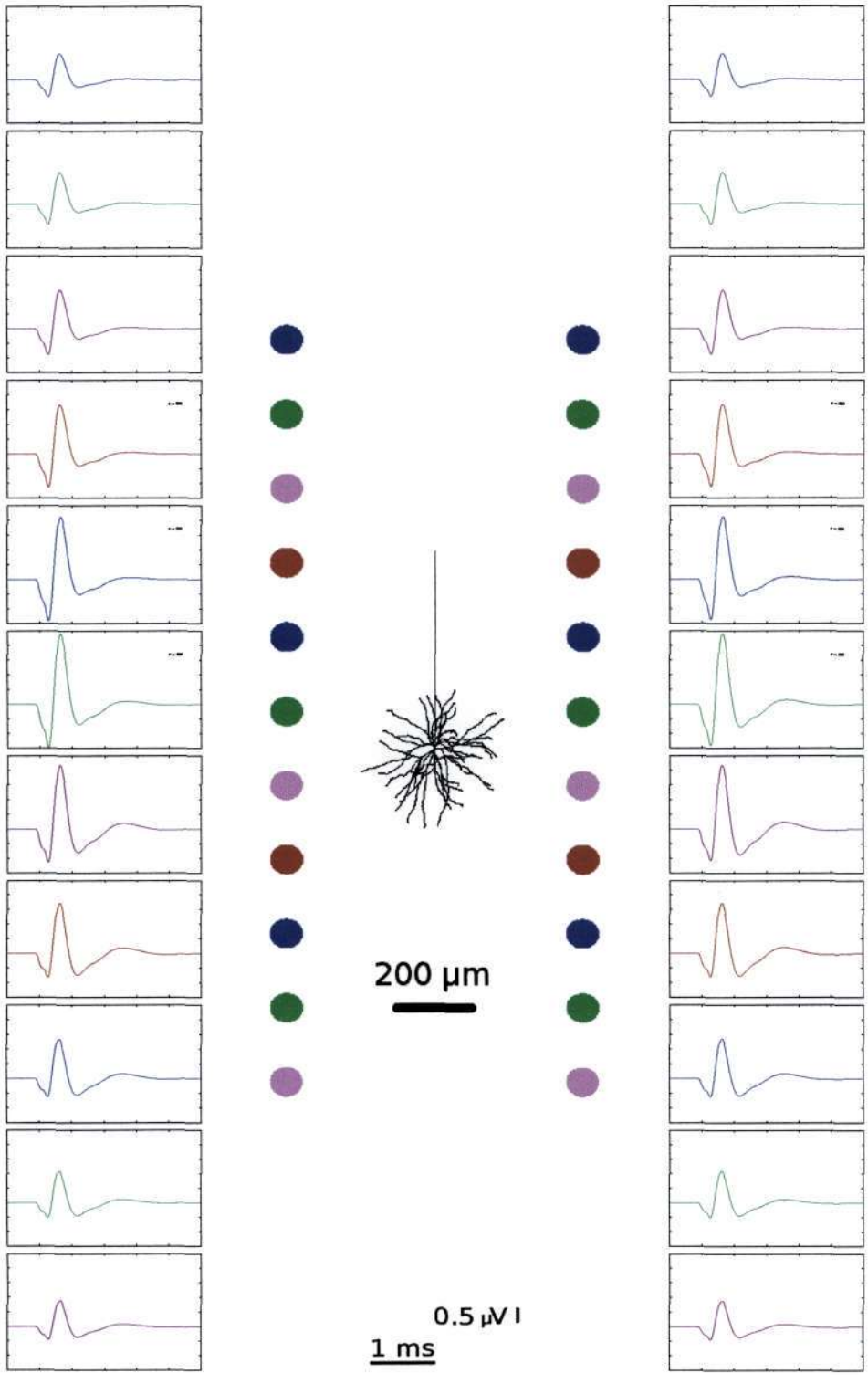


Figure 6.20:
The filtered (750 - 3000 Hz) extracellular potential for a layer 4 stellate cell.

(shown in Figure 6.18) shows a sort of intermediate case, in between the dipole nature of the layer 5 potentials and the non-dipole potentials of the layer 3 pyramid. This shows that the dendritic structure of the neuron determines the extracellular signature of the spike.

The layer 5 pyramid has a large apical dendrite which spreads out into a large structure above the soma while the axon extends below the soma. The dendritic structure of the layer 3 pyramid extends over the same space as the axon, giving it a more spherically symmetric structure. The layer 4 stellate cell has a large portion of its axon outside of the dendritic branches. The dipolar structure of the layer 5 pyramid is commonly known as an open field situation; the other cells and their more stellar shaped structures are called closed fields.

Figures 6.6, 6.14 and 6.18 show that the extracellular potential has peaks of magnitude of about 0.1 mV for distances of 100 μm or less away from the soma. This is comparable to experiments by Somogyvári *et al*, where a multielectrode was used to measure extracellular potentials in cat primary auditory cortex [33]. The laminar electrode used in those experiments are estimated to be at most 110 μm from the cell body. The decay of the extracellular traces with distance from the central maximum is also evident in the results of Somogyvári *et al*.

The results of Somogyvári *et al* [33] show that most of the traces do not change sign above and below the trace with the maximum amplitude. This is true for po-

sitions that are not very close to the soma. The extracellular traces in Figure 6.6, for the layer 5 pyramid cell, are quite different from Somogyvári's results. These results more resemble the traces in Figure 6.14, for the layer 3 pyramid. The cells of Somogyvári *et al* stretch from layer 2/3 to layer q6 [33], yet the results appear similar to the data from the layer 3 pyramid simulations. (The traces from the layer 4 stellate cell, in Figure 6.18, have shapes that are somewhere in between the shapes for the layer 5 and 3 pyramids.)

Chapter 7

Extracellular Potentials Around Populations

7.1 Extracellular Potentials From Single-Neuron Data

Extracellular potentials for populations of identical neurons are presented here. Data files were generated by the NEURON simulations (of single neurons) and were used to compute the extracellular potentials of a population of identical neurons. To simulate extracellular potentials of a population of neurons in a cylinder, the data files containing membrane currents from the nodes in the neurons were used..

In Chapter 5, it was shown how to calculate the extracellular potential around a single neuron using current sources. The potential at a point outside of a neuron is calculated in Equation 5.10 using superposition. The vertical alignment of the

population of neurons in the cylinder is given by a normalised Gaussian function (Equation 5.13) that gives a spread in the vertical alignment of the neurons in the population cylinder. The extracellular potential at a height z in the cylinder is given by Equation 5.15.

The potential was calculated using the equation

$$V(z,t) = \rho \int \sum_n \lambda_n(z_0,t)\; g(z,z_0)\; dz_0, \tag{7.1}$$

where

$$\lambda_n(z_0,t) = I_n(t) f_n(z) = \frac{I_n(t)}{\sqrt{2\pi}\sigma_z} \exp\left(-\frac{(z_0 - z_n)^2}{2\sigma_z^2}\right) \tag{7.2}$$

$$g(z,z_0) = \frac{1}{2\sigma_{\mathrm{ex}}}\left(\sqrt{R_{\mathrm{pop}}^2 + (z-z_0)^2} - |z - z_0|\right). \tag{7.3}$$

The potential is calculated along the centre of the cylinder, at a height z. The height of the cylinder was 1800 μm, with z from 1200 μm to -600 μm. The radius of the cylinder is R_{pop}. The extracellular potential was calculated at 9 points, with a virtual space of 200 μm between each point.

The extracellular conductivity of the medium was taken to be 0.3 Sm^{-1} as before. Not all the cells in a cylinder are thought to be involved in firing of action potentials at the same time. Only a proportion of the cells were involved and that was given by the factor ρ. This cell density factor, ρ, was given a value of 1×10^{-4} $\mu\mathrm{m}^{-2}$ (or 100 active cells per mm^{-2}) throughout the simulations. The standard deviation σ_z, which can also be taken as a variation in the morphology of the cells, was given a value of 100 μm. Five different cylinder sizes were used in the population:

generation and propagation in neurons [2]. This model arguably represents the biggest success in theoretical biology and has opened (for mathematical analysis) a wide variety of diverse membrane phenomena. One example is the compartmental modelling of propagation of synaptic signals to the neuron soma, which is crucial for understanding the action-potential generation and thus the information processing properties of a single neuron [3, 4]. At present one has the necessary mathematical machinery to, in principle, simulate signal processing in biologically realistic neural networks.

Broadly speaking, two types of electrical single-unit recordings are done. In *intracellular* recordings one inserts the electrode into a cell and measures transmembrane electrical events directly. In *extracellular* recordings, on the other hand, electric fields due to neuronal activity are measured between two points in the extracellular space rather than across the cell membrane. The signals recorded are called *extracellular field potentials.*

When an extracellular electrode is placed very close to the soma (cell body) of a particular neuron, the signal it picks up will be dominated by currents from the soma and parts of the dendrites close to the soma, see Figure 1.1. An action potential from this neuron will thus give a clear recording signature. Extracellular recordings have thus become a standard tool for measuring the firing of neuronal action potentials. The method is not limited to studies of such single-neuron events, however. It can also be used to study the activity in groups or *populations* of neu-

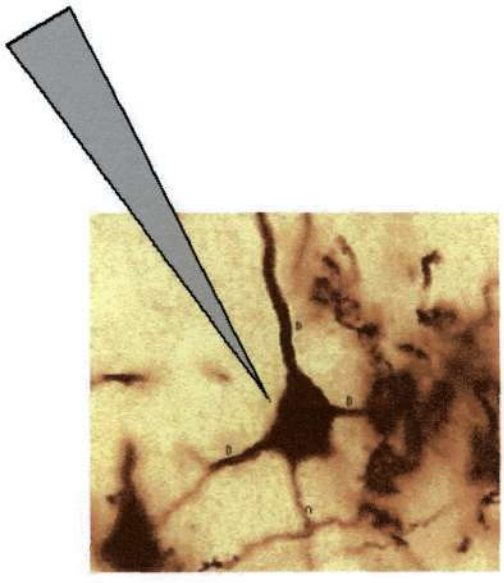


Figure 1.1:
Single unit method for measuring electrical activity from one neuron. A schematic representation of a microelectrode placed in the extracellular medium close to the soma of a neuron to record action potentials fired by the neuron.

rons, but the interpretation of such recordings in terms of the underlying neural activity is often difficult.

To extract more information about neural activity, various types of *multi-electrodes* are now also used. In such electrodes, the extracellular potentials are simultaneously recorded at several spatial positions, and more correlated data thus becomes available. EEG recordings are in fact large scale variations of multi-electrode extracellular measurements where the electrodes are positioned outside the brain.

A long term goal would be to be able to use electrical data (EEG, MEG) on the macroscopic systems level ($\sim$ 10 cm) to infer the underlying activity at the micro-

scopic level, i.e., at the level of individual neurons ($\sim$ 10 μm). A natural strategy when pursuing this goal is to first try to understand electrical activity at an intermediate, i.e., *mesoscopic* level ($\sim$ 1 mm). So-called *laminar electrodes* are used for measurements of cortical activity at this length scale.

The cortex in mammals is typically a couple of millimetres thick, and has a characteristic layered structure where there is much less variation in the distribution of neurons laterally than in the vertical direction. It is customary to divide the cortical sheet into six layers, and different neuron types are preferentially located in the various layers [5]. Laminar electrodes are typically inserted into the cortex perpendicular to its plane, and they record the extracellular potential at many cortical depths (typical spacing of 100 μm) [6]. The low-frequency part of the potentials (*Local Field Potential, LFP*) contains information about dendritic processing of synaptic inputs in populations of neurons, while the high-frequency part (*Multi-Unit Activity, MUA*) is thought to reflect firing of action potentials in such cortical populations. Laminar-electrode recordings in the cortex are schematically illustrated in Figure 1.2.

The Computational Neuroscience group at the Norwegian University of Life Sciences (UMB) has started a collaborative project with professor Anders Dale and his group at University of California, San Diego to develop methods for better interpreting laminar-electrode data. The somatosensory cortex (barrel cortex) of the rat has been chosen as a model system, and laminar-electrode data has been recorded

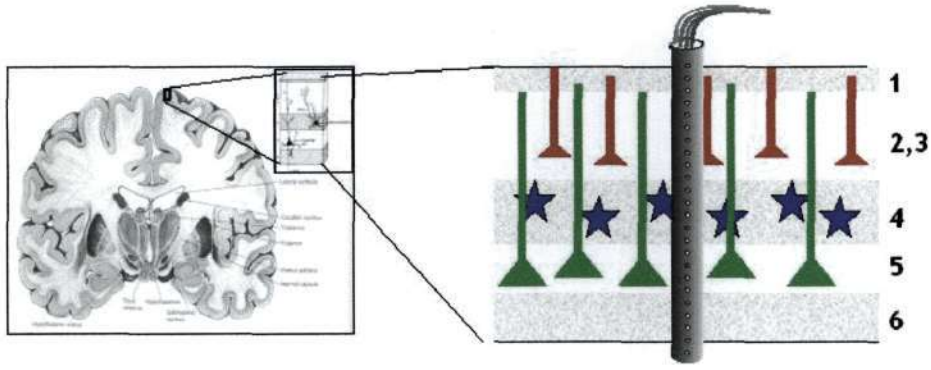


Figure 1.2:
Diagram of a laminar electrode used to measure action potentials fired by a population of neurons. The electrodes have numerous contact points that simultaneously record neural activity at various depths of the cortex. The diagram shows different types of cells (pyramids in red and green and stellates in blue) that are found in the six layers of the cortex, labelled 1 through to 6.

following whisker stimulation. Several modelling projects at UMB have been initiated to study such data, and the present thesis project is one of the them.

The focus of the present project is on action potentials and the extracellular potentials they make. The MUA extracted from laminar-electrode data is thought to originate from various cortical populations firing more or less synchronously. MUA data represents a sum of contributions from numerous firing neurons surrounding the electrode contacts, and the direct relationship with the firing of action potentials in various populations is not easy to establish. However, the *forward* problem of calculating the extracellular potential around a particular firing neuron is well-posed and in principle straightforward to do.

In the present thesis we study the extracellular fields generated by action potentials

for a set of different types of cortical neurons. The contribution to the extracellular potential from neurons depends on the spatial and temporal distribution of ion currents going through the neuronal cell membrane. Detailed anatomically reconstructed neurons must thus be considered, and so called *compartmental modelling* must be used. In such modelling the dendritic tree is described as a branched cable structure, where each cable or cylinder, in principle, can have different diameters and membrane properties. The simulation tool NEURON [7] is used in the simulation. The three reconstructed cortical model neurons used in the project corresponds to the model neurons described and used by Mainen and Sejnowski [8].

Calculations of extracellular potentials during action-potential firing from *individual neurons* have been pursued previously [9]. Our particular aim here is to elucidate the contributions to MUA recorded by laminar electrodes. We have therefore further studied what effect high-pass filtering of the extracellular data (as done when extracting the MUA) have on the predicted extracellular response. The MUA recorded from one of the electrode contacts in the laminar electrode generally stems from numerous neurons. We have therefore also studied extracellular responses of *populations* of neurons firing synchronously. Such a population response will depend on the spatial distribution of neurons in the synchronously firing cortical population. While our present results are limited to calculating the extracellular response at the centre axis of a cylindrical population, the scheme can be straightforwardly generalised to other geometries.

$R_{\text{pop}} = 100\ \mu\text{m}$, 400 μm, 800 μm, 1000 μm and 2000 μm.

This model assumes synchronous neurons. Extracellular potentials around different neuron populations were calculated. The filtered traces are then shown for each neuron population. The potential is calculated at different heights around the time of the firing of an action potential by the neurons in the cylinder. Extracellular traces for varying cylinder radii are shown.

7.1.1 Asynchronous Populations

The population models described above consist of synchronously firing neurons. In reality neurons fire action potentials at different times relative to each other.

A very simple asynchronous population model involves assuming a temporal jitter in the firing of action potentials in a population of neurons. One way of doing this is using a Gaussian distribution of (action potential) firing times

$$\rho(t') = \frac{1}{\sqrt{\pi}\sigma_t} \exp\left(-\frac{t'^2}{\sigma_t^2}\right), \tag{7.4}$$

where t' is the firing times and σ_t is the measure of the temporal spread of firing. The asynchronous population $V_{\text{async}}(t)$ is calculated using the synchronous population extracellular potentials in equation 7.1, using a time convolution as follows

$$V_{\text{async}}(t) = \int_{t'=-\infty}^{t'=\infty} \rho(t')\ V(t-t')\ dt', \tag{7.5}$$

where $V(t-t')$ is the synchronous population potential.

Computationally it is not necessary to calculate the integral in Equation 7.5 over an infinite time, as the Gaussian distribution goes to zero for large $|t-t'|$. The limits of integration are hence chosen as $-4\sigma_t$ to $4\sigma_t$, which provides a time filter of adequate size. The asynchronous population potential then becomes

$$V_{\text{async}}(t) = \int_{t'=-4\sigma_t}^{t'=4\sigma_t} \rho(t')\ V(t-t')\ dt'. \tag{7.6}$$

The asynchronous population model was used to investigate populations of different cylinder radii as well as with varying temporal spread, σ_t. The plots presented in this chapter show extracellular potentials for populations of radii 50 and 100 μm, for $\sigma_t = 1$ ms and $\sigma_t = 2$ ms.

7.2 Layer 5 Populations

7.2.1 Synchronous Populations

The following plots (Figures 7.1 and 7.2) show the extracellular potential around a population of synchronously firing layer 5 pyramidal neurons, for varying cylinder radii.

The extracellular potential is largest close to the region where the somas lie. The potential decays very rapidly at either ends of the cylinder for a small cylinder radius.

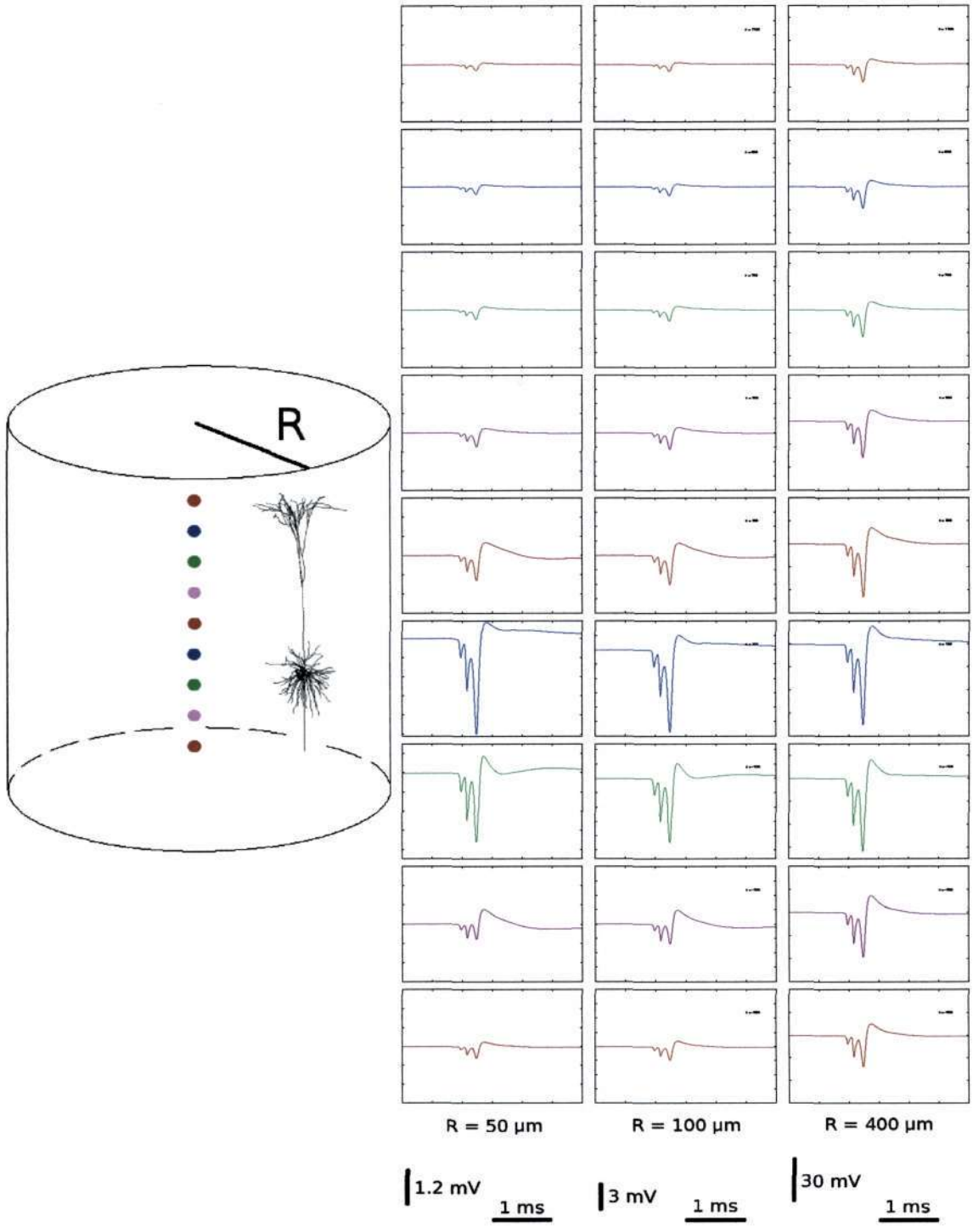


Figure 7.1:
Extracellular potential for a synchronous population of layer 5 pyramids in a cortical cylinder for various radii (50, 100 and 400 μm).

The decay rate is faster than for a single layer 5 pyramid neuron (see Figure 6.8).

The extracellular potential is larger, however, for larger radii. For wider columns, the decay rate of the amplitude with respect to height is less. For large radii, the systems begins to look like infinite planes where the potential outside infinite planes of current is a constant.

Figures 7.1 and 7.2 show that for populations in cylinders of large radii, the amplitudes of the extracellular potentials are very high, of about 40 mV or higher. For populations of radii 800 μm, 1000 μm and 2000 μm, the amplitude is greater than 60 mV. Extracellular potentials from the experiments by Somogyvári *et al* [33] have amplitudes in the order of 0.1 mV. The amplitudes here depend on the value of the cell density factor ρ. A smaller density would result in smaller amplitudes for the population traces. The extracellular traces in Figures 7.1 and 7.2 are due to a completely synchronous population. Large completely synchronous populations are very unlikely, so extracellular traces with high amplitudes as those calculated would therefore not likely occur.

The traces in Figure 7.2 show that the only difference in the extracellular potential for large radii is the amplitude. The shapes of the traces are all very similar. For a population of radius 800 μm and beyond the shape of the potential at the cylinder axis varies little with height.

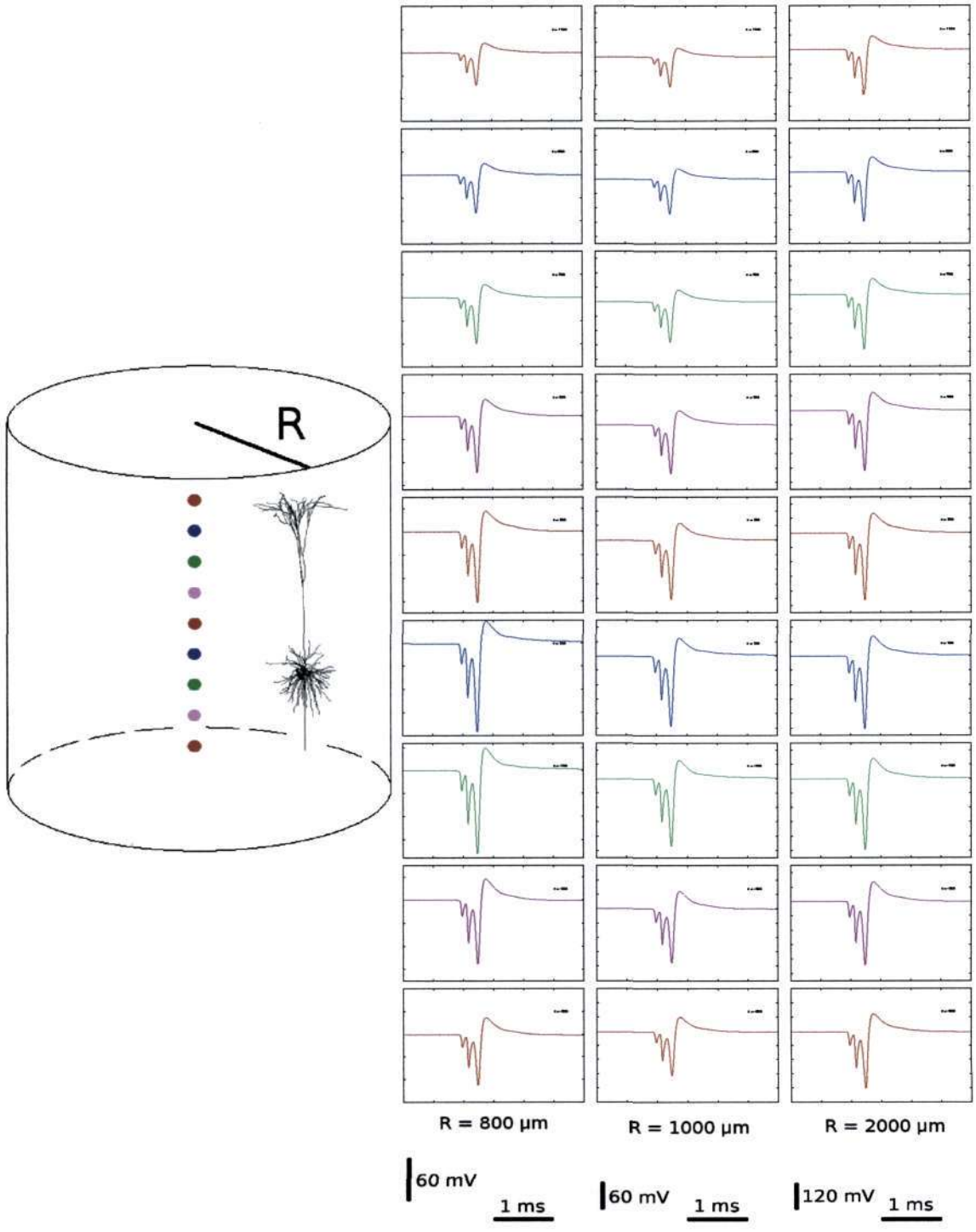


Figure 7.2:
Extracellular potential for a synchronous population of layer 5 pyramids in a cortical cylinder for various radii (800, 1000 and 2000 μm).

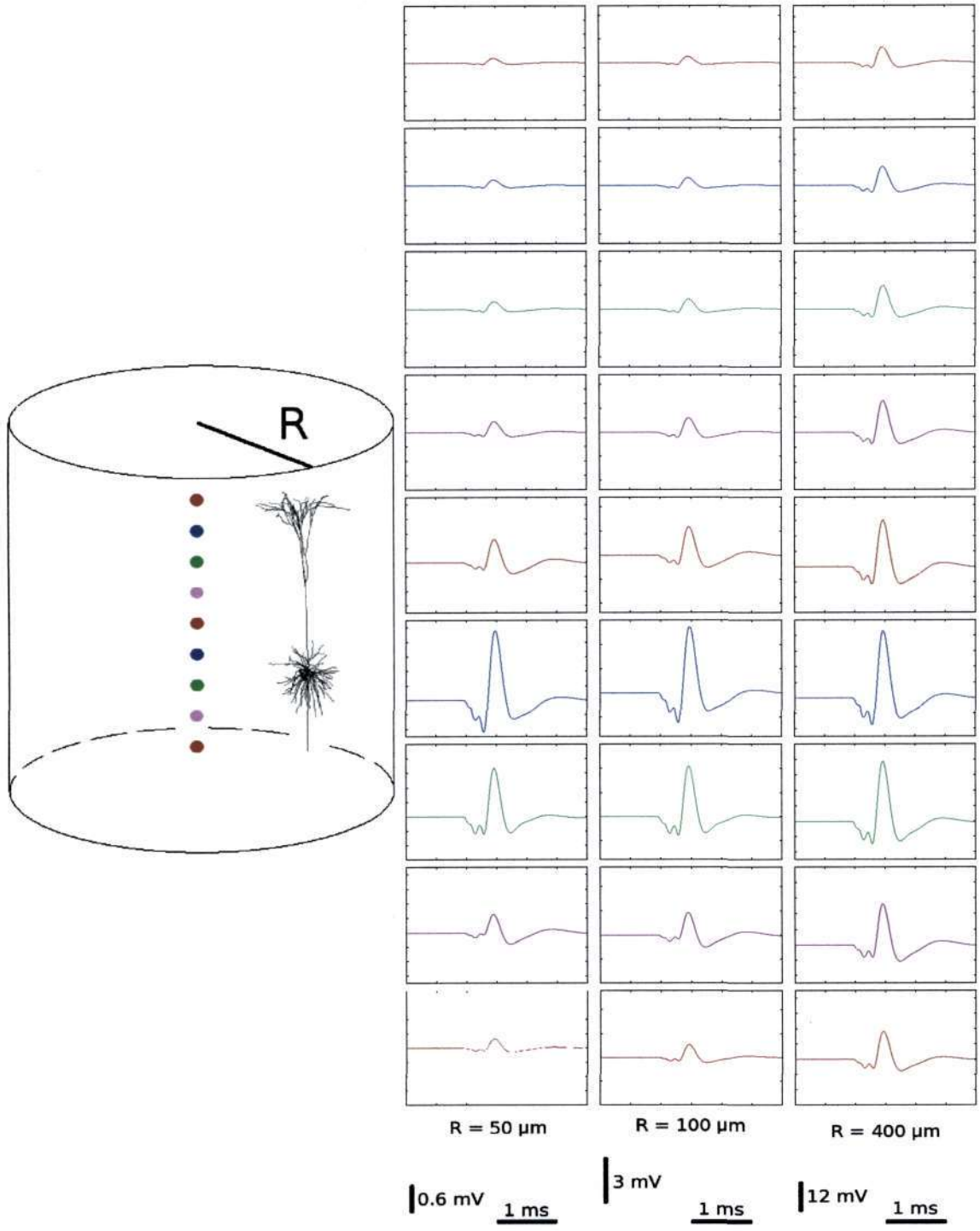


Figure 7.3:
High-pass filtered extracellular potential traces for a synchronous population of layer 5 pyramids in a cortical cylinder of varying radii (50, 100 and 400 μm).

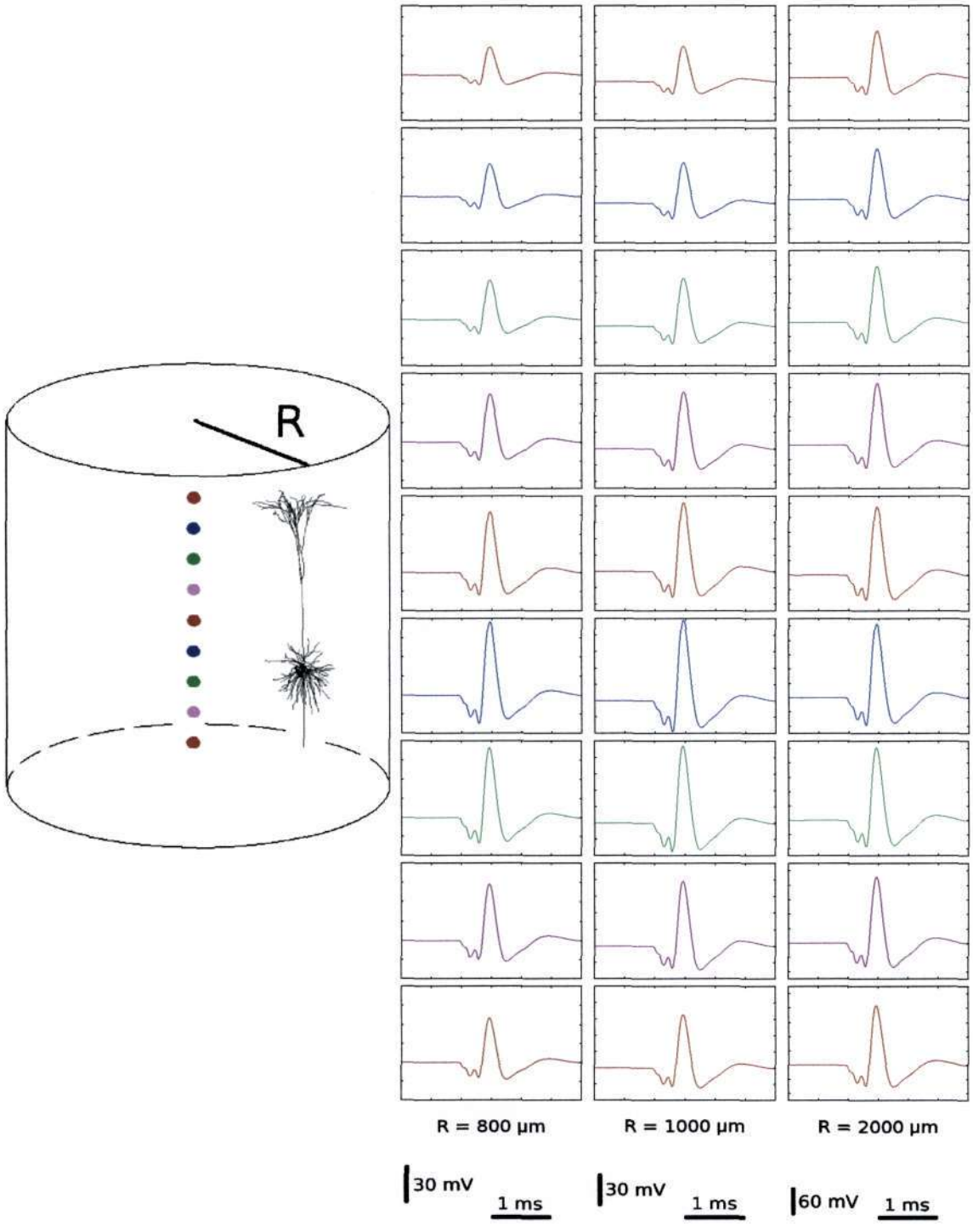


Figure 7.4:
High-pass filtered extracellular potential traces for a synchronous population of layer 5 pyramids in a cortical cylinder of varying radii (800, 1000 and 2000 μm).

7.2.2 High-Pass Filtered Synchronous Populations

Figures 7.3 and 7.4 show the filtered extracellular traces for synchronous populations of varying cylinder radii.

The traces show that for the smaller radii (Figure 7.3) there is a rapid decay for distances further away from the soma. This decay rate is not so pronounced for populations of large radii, e.g. for R equal to 400, 800, 1000 and 2000 μm.

7.2.3 Asynchronous Populations

The extracellular traces for an asynchronous population are shown in Figures 7.5 and 7.6. The three columns show traces for different values of σ_t; σ_t was set to 0, 1 and 2 ms. The different values of σ_t makes it necessary to vary the time window used to compute the extracellular potential.

The traces have a time interval of 3, 8 and 19 ms for σ_t values of 0, 1 and 2 ms respectively. This is because the trace has $4\sigma_t$ ms on either side of the 3 ms where the action potential most likely takes place. A σ_t value of 0 ms corresponds to a completely synchronous population, hence the trace lasts only 3 ms. For the other two σ_t values, the most likely time an action potential occurs is in the middle 3 ms of the trace.

The two sets of plots show the extracellular traces for populations of neurons where the cortical cylinder has radius 50 μm (Figure 7.5) and 100 μm (Figure 7.6).

7.2.4 High-Pass Filtered Asynchronous Populations

Figures 7.7 and 7.8 show the extracellular traces for an asynchronous population of layer 5 pyramid cells, which have been filtered (750 - 3000 Hz) using a high-pass Butterworth filter.

The amplitudes of the traces in Figures 7.7 and 7.8 for σ_t values of 1 and 2 ms are very small. This a reflection that a high-pass filtering as well as low-pass filtering has been done, the latter due to the time convolution with the synchronous population extracellular potential function.

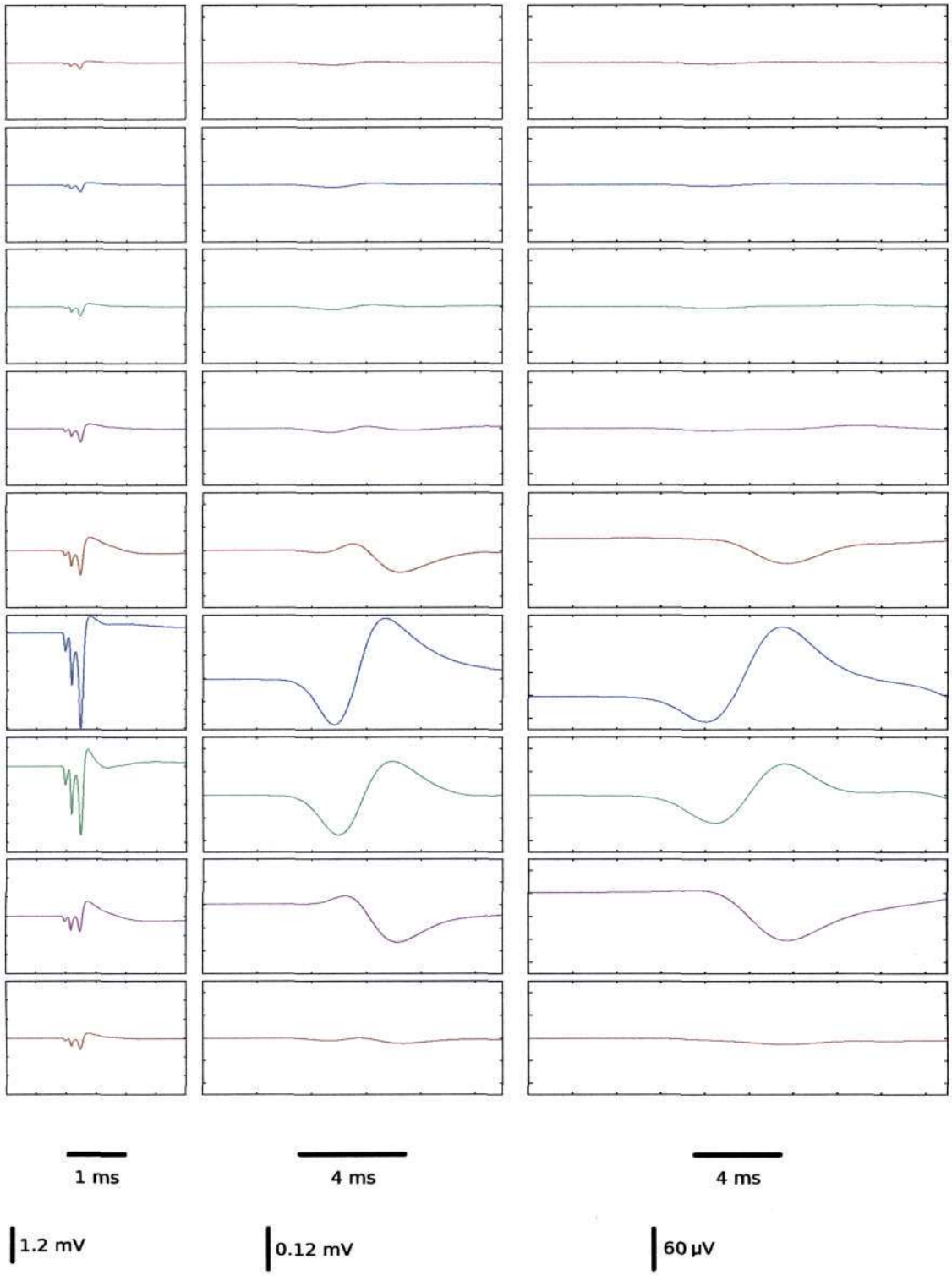


Figure 7.5:
Extracellular potential traces for an asynchronous population of layer 5 pyramids in a cortical cylinder of radius 50 μm; σ_t has values 0, 1 and 2 ms.

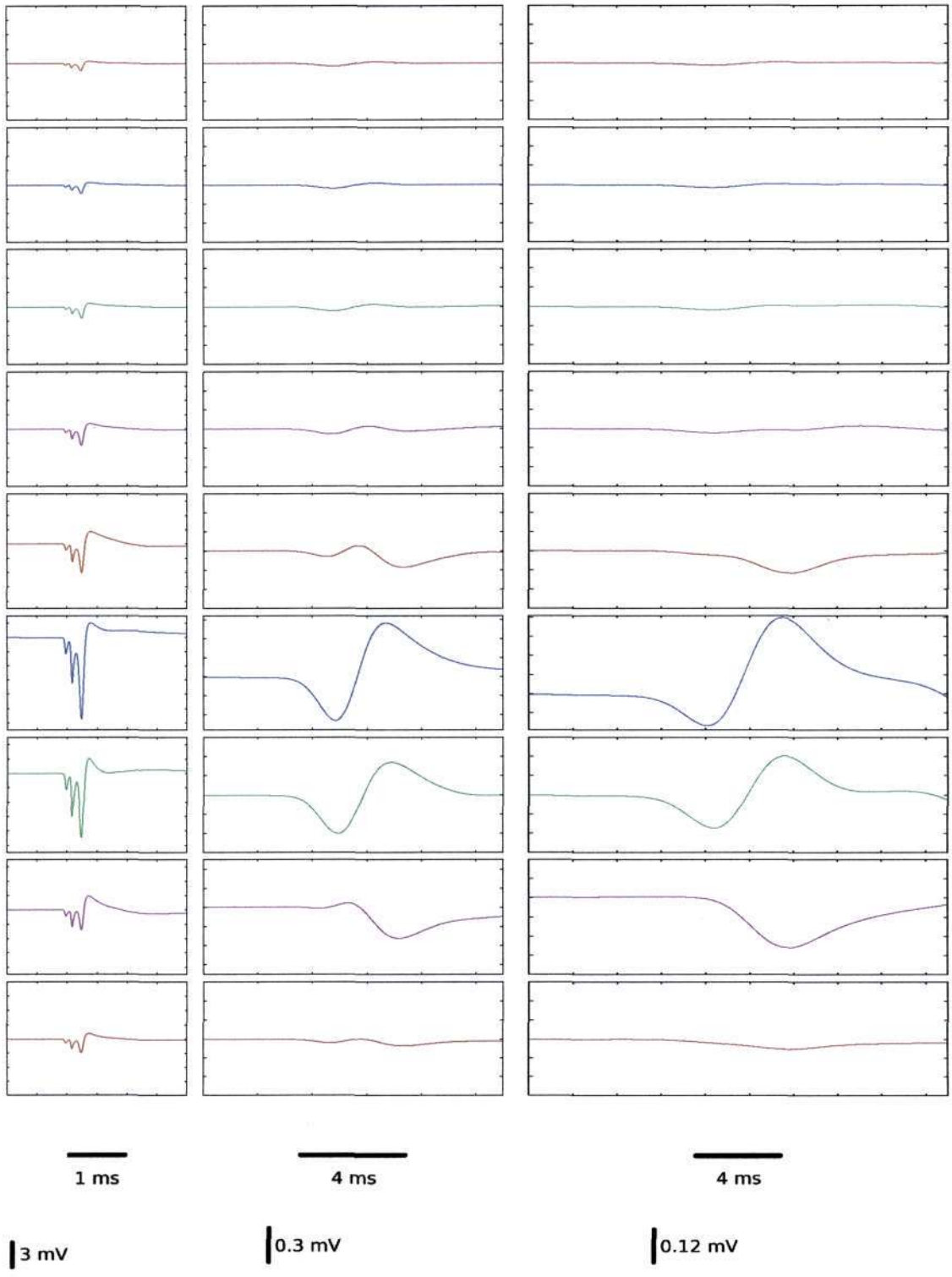


Figure 7.6:
Extracellular potential traces for an asynchronous population of layer 5 pyramids in a cortical cylinder of radius 100 μm; σ_t has values 0, 1 and 2 ms.

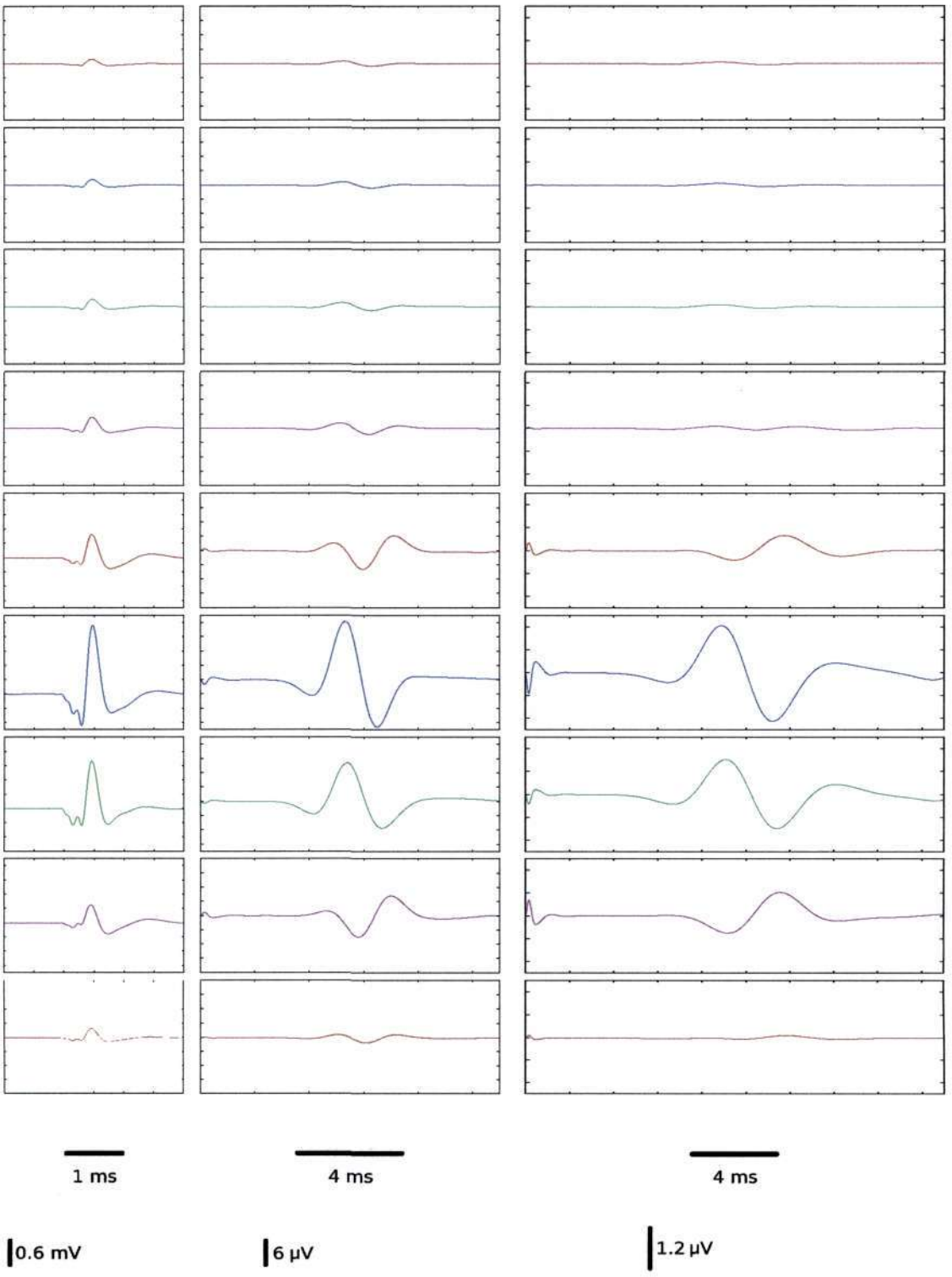


Figure 7.7:
High-pass filtered extracellular potential traces for an asynchronous population of layer 5 pyramids in a cortical cylinder of radius 50 μm; σ_t has values 0, 1 and 2 ms.

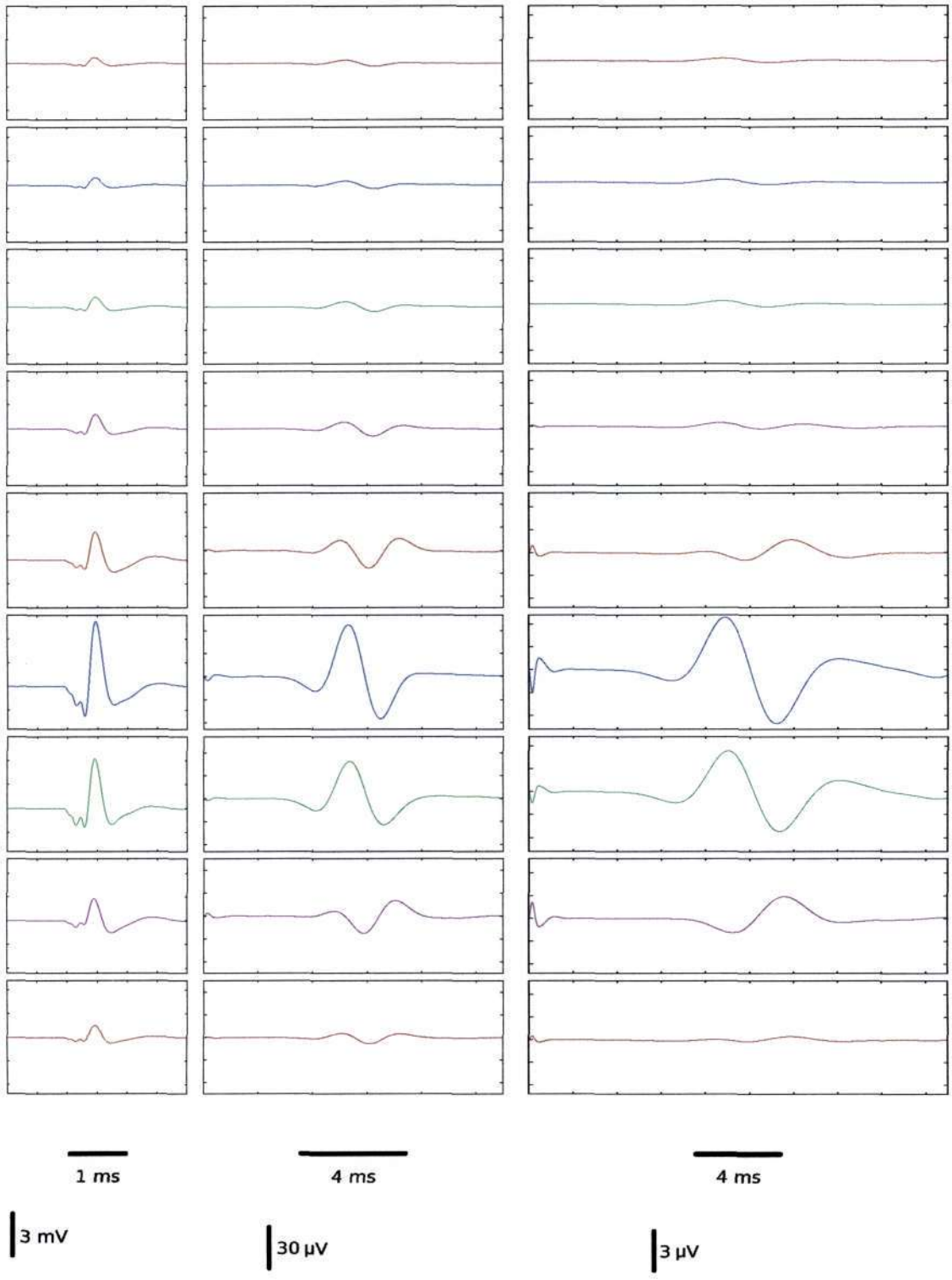


Figure 7.8:
High-pass filtered extracellular potential traces for an asynchronous population of layer 5 pyramids in a cortical cylinder of radius 100 μm; σ_t has values 0, 1 and 2 ms.

7.3 Layer 3 Populations

7.3.1 Synchronous Populations

Figures 7.9 and 7.10 show the extracellular potential around a population of synchronously firing layer 3 pyramidal neurons, for varying cylinder radii.

The traces show that for small population diameters, the decay rate is fast for positions at the extremities of the cylinder. The amplitudes increase with population size and once more the shape of the extracellular traces show little difference as the population radius goes beyond 800 μm.

The traces for the layer 3 pyramid look similar to the corresponding traces of the layer 5 pyramid (Figures 7.1 and 7.2). Yet extracellular traces for single neurons of these two cell types are qualitatively different (see Figures 6.6, for the layer 5 pyramid, and 6.14, for the layer 3 pyramid, in Chapter 6). There is a change in polarity for traces above and below the soma for a single layer 5 pyramid which is not present for a population of such neurons. The polarity change is not evident in the layer 3 pyramid for traces away from the soma. This is related to the position of the axon relative to the soma of the two neuron types.

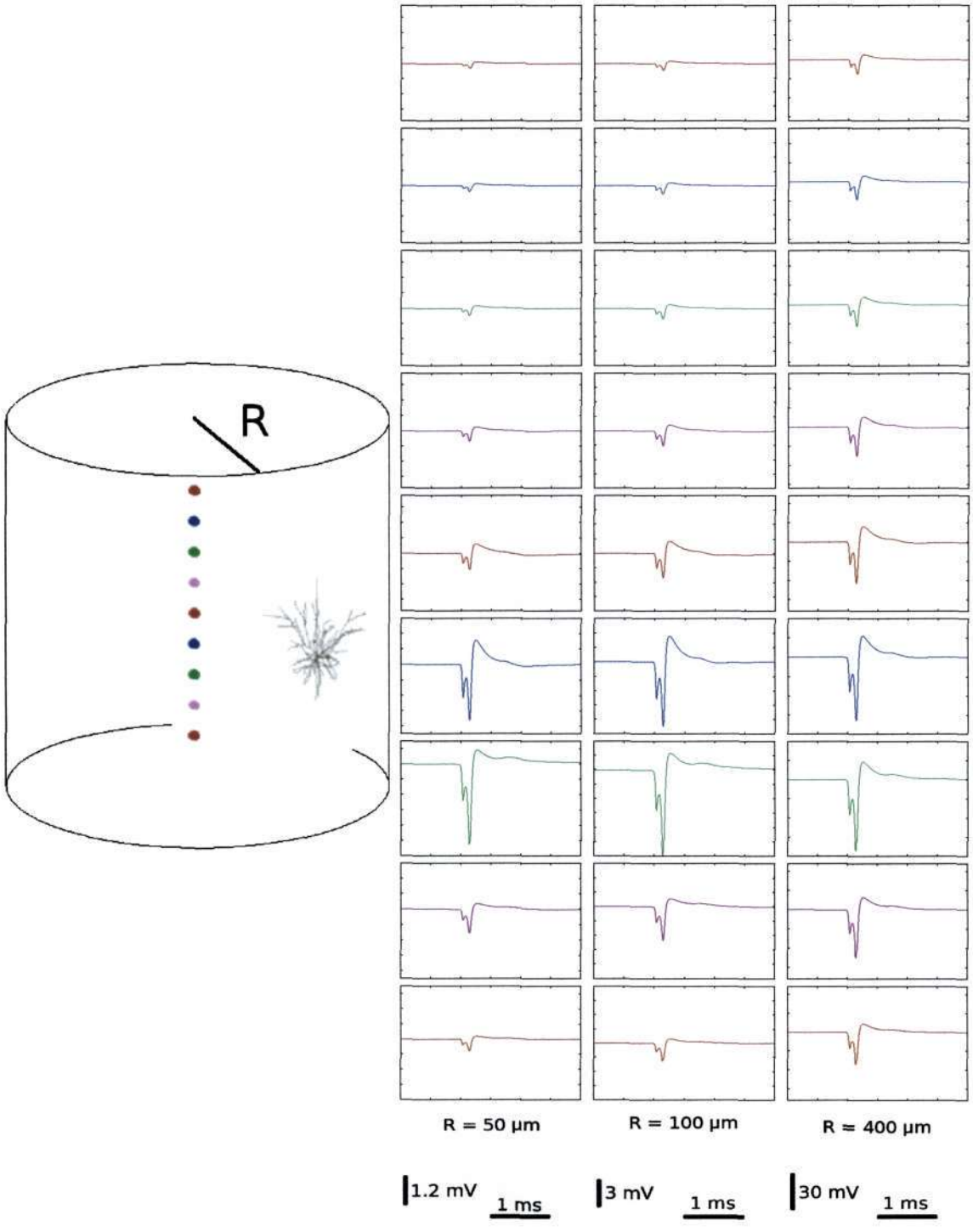


Figure 7.9:
Extracellular potential for a synchronous population of layer 3 pyramids in a cortical cylinder for various radii (50, 100 and 400 μm).

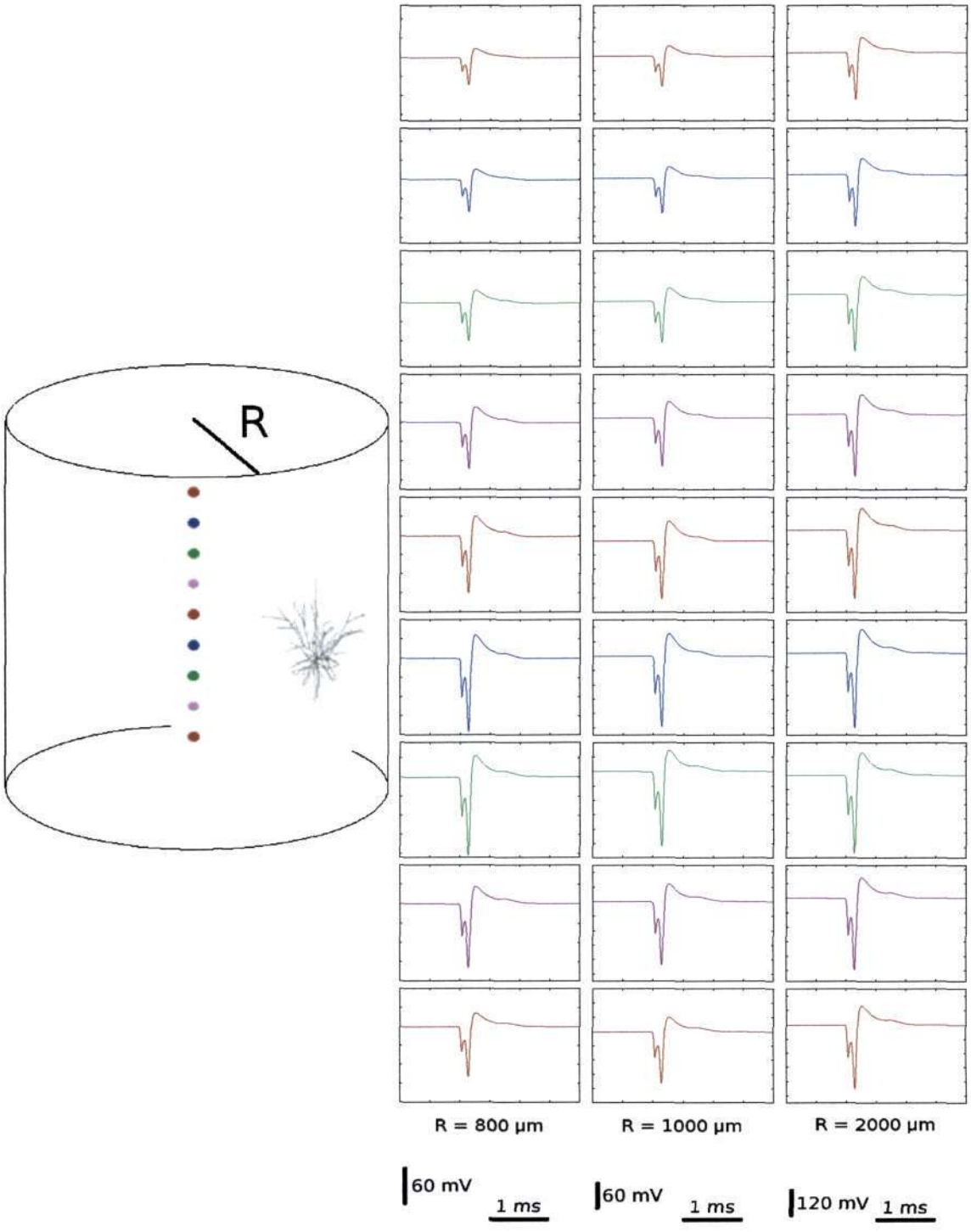


Figure 7.10:
Extracellular potential for a synchronous population of layer 3 pyramids in a cortical cylinder for various radii (800, 1000 and 2000 μm).

7.3.2 High-Pass Filtered Synchronous Populations

Figures 7.11 and 7.12 show the filtered extracellular traces for synchronous populations of layer 3 pyramids of varying cortical cylinder radii. The filtered traces all show a very similar shape. The amplitude increases as the population radius increases, with very small amplitudes far away from the soma for populations of small radius.

7.3.3 Asynchronous Populations

The extracellular traces for asynchronous layer 3 pyramids are shown in Figures 7.13 and 7.14. The two sets of traces show the extracellular potentials for populations of radius 50 μm and 100 μm respectively.

Because of the amplitudes of the traces, the positive and negative peaks in the traces are only visible close to the region of the soma positions; further away from the soma positions the amplitudes are very small in comparison. The shapes differ from those of the layer 5 pyramid in that there is another spike in the trace. Figure 6.4 shows the action potentials in the soma of the three neurons used in the simulations. The extracellular potential was calculated around the first of the three spikes fired by the layer 3 pyramid. The second action potential contributes to the activity shown in the traces of Figures 7.13 and 7.14.

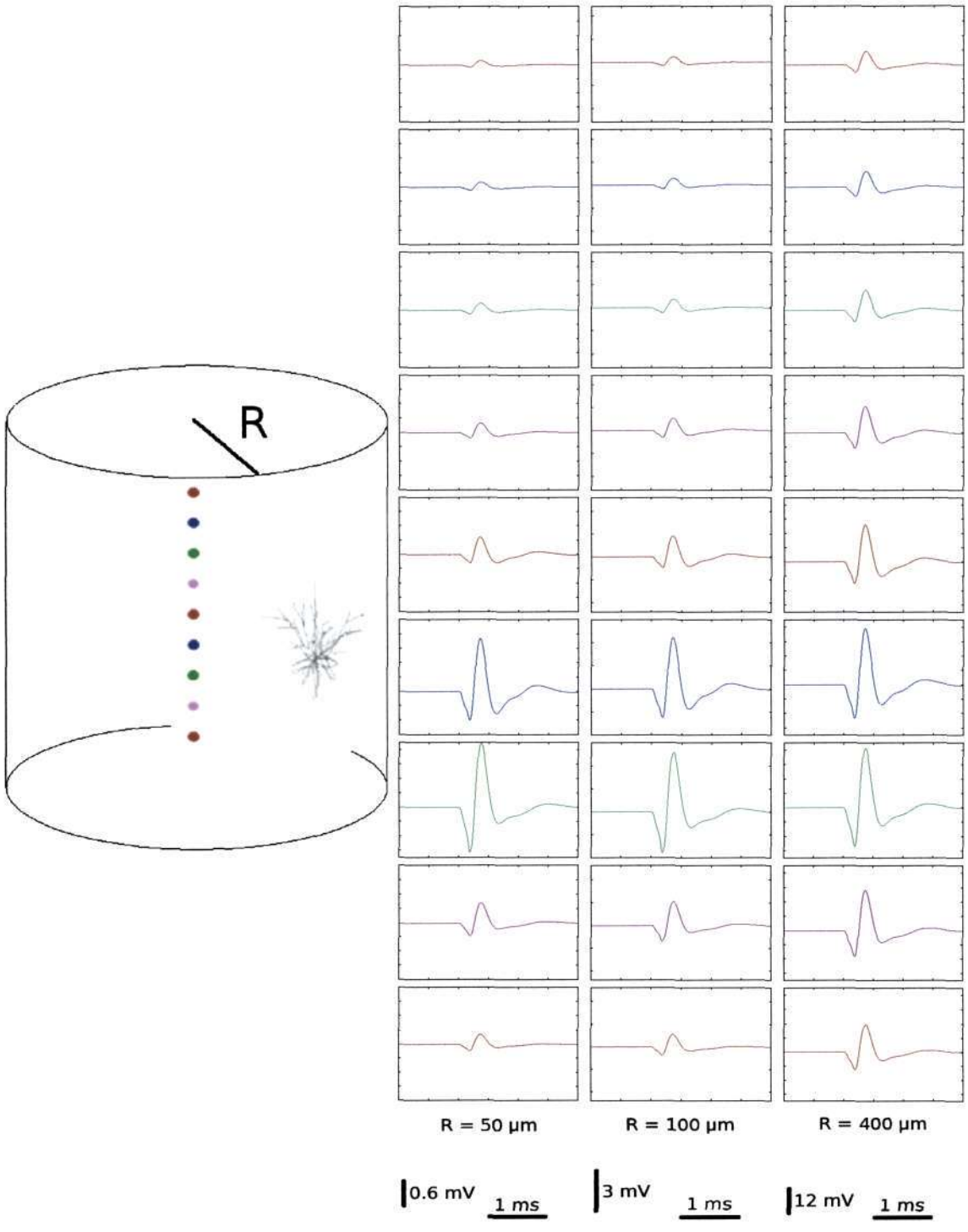


Figure 7.11:
High-pass filtered extracellular potential for a synchronous population of layer 3 pyramids in a cortical cylinder for various radii (50, 100 and 400 μm).

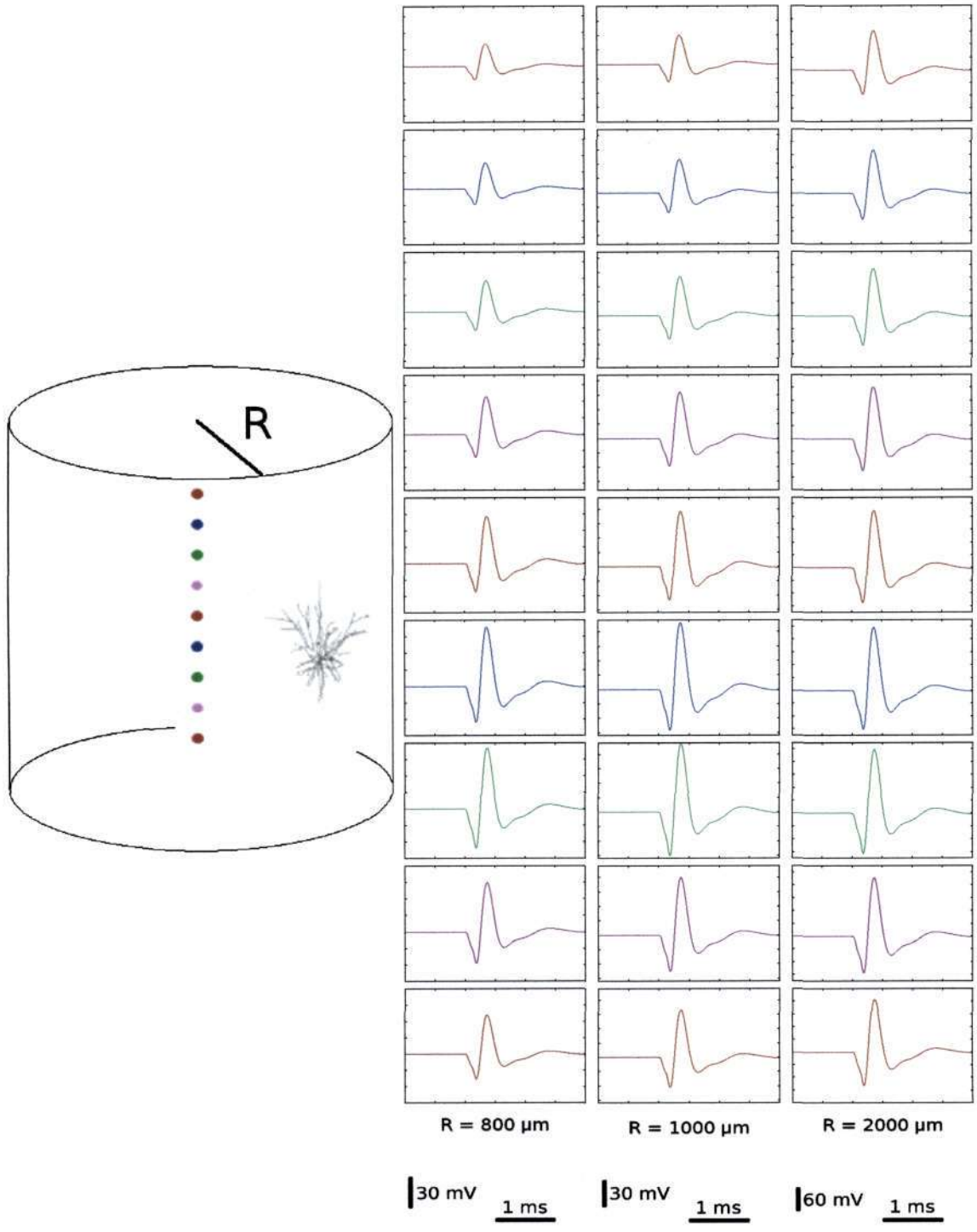


Figure 7.12:
High-pass filtered extracellular potential for a synchronous population of layer 3 pyramids in a cortical cylinder for various radii (800, 1000 and 2000 μm).

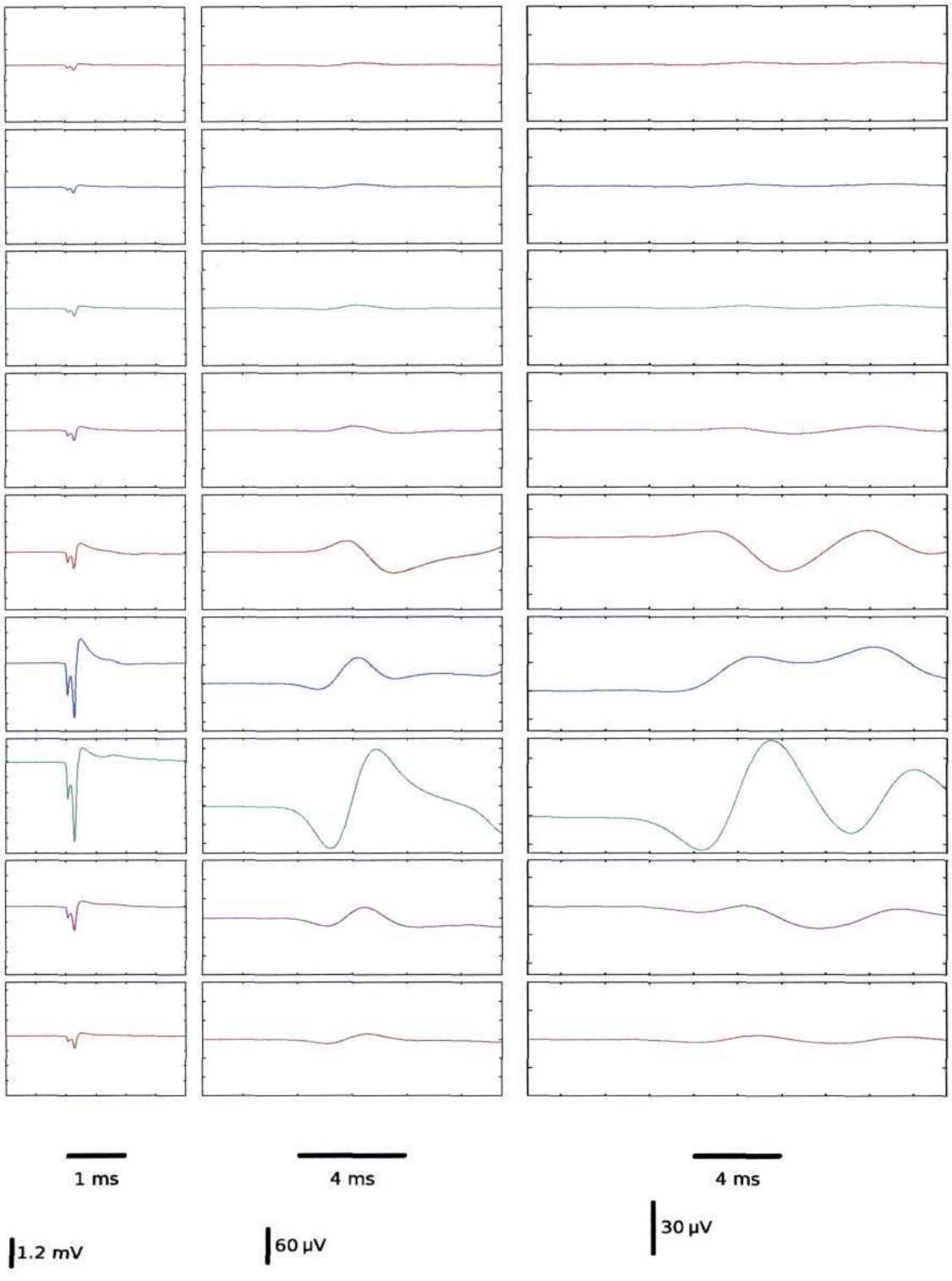


Figure 7.13:
Extracellular potential traces for an asynchronous population of layer 3 pyramids in a cortical cylinder of radius 50 μm; σ_t has values 0, 1 and 2 ms.

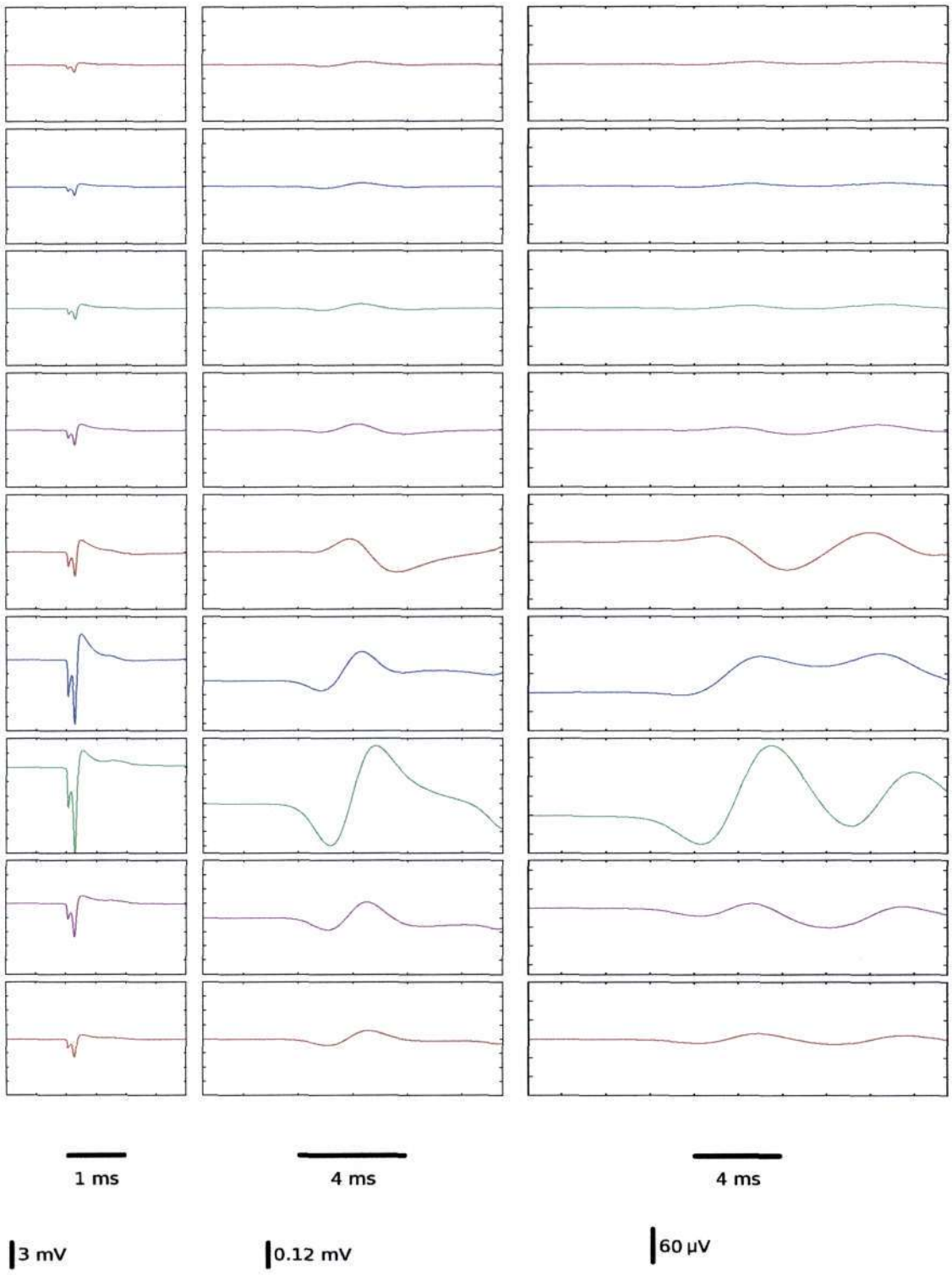


Figure 7.14:
Extracellular potential traces for an asynchronous population of layer 3 pyramids in a cortical cylinder of radius 100 μm; σ_t has values 0, 1 and 2 ms.

7.3.4 High-Pass Filtered Asynchronous Populations

The filtered extracellular traces are shown is Figure 7.15 and Figure 7.16, for a population of radii 50 μm and 100 μm respectively.

The traces are different in shape as compared to, for example, the filtered -traces for the layer 5 pyramid (Figures 7.7 and 7.8). This is due once more to the second action potential present in the unfiltered trace.

Both populations (for R equal to 50 and 100 μm) show similar shapes, and once again the difference between the two is the amplitude of each trace. For these small population radii, the potential decays very rapidly with vertical distance from the most likely position of the soma.

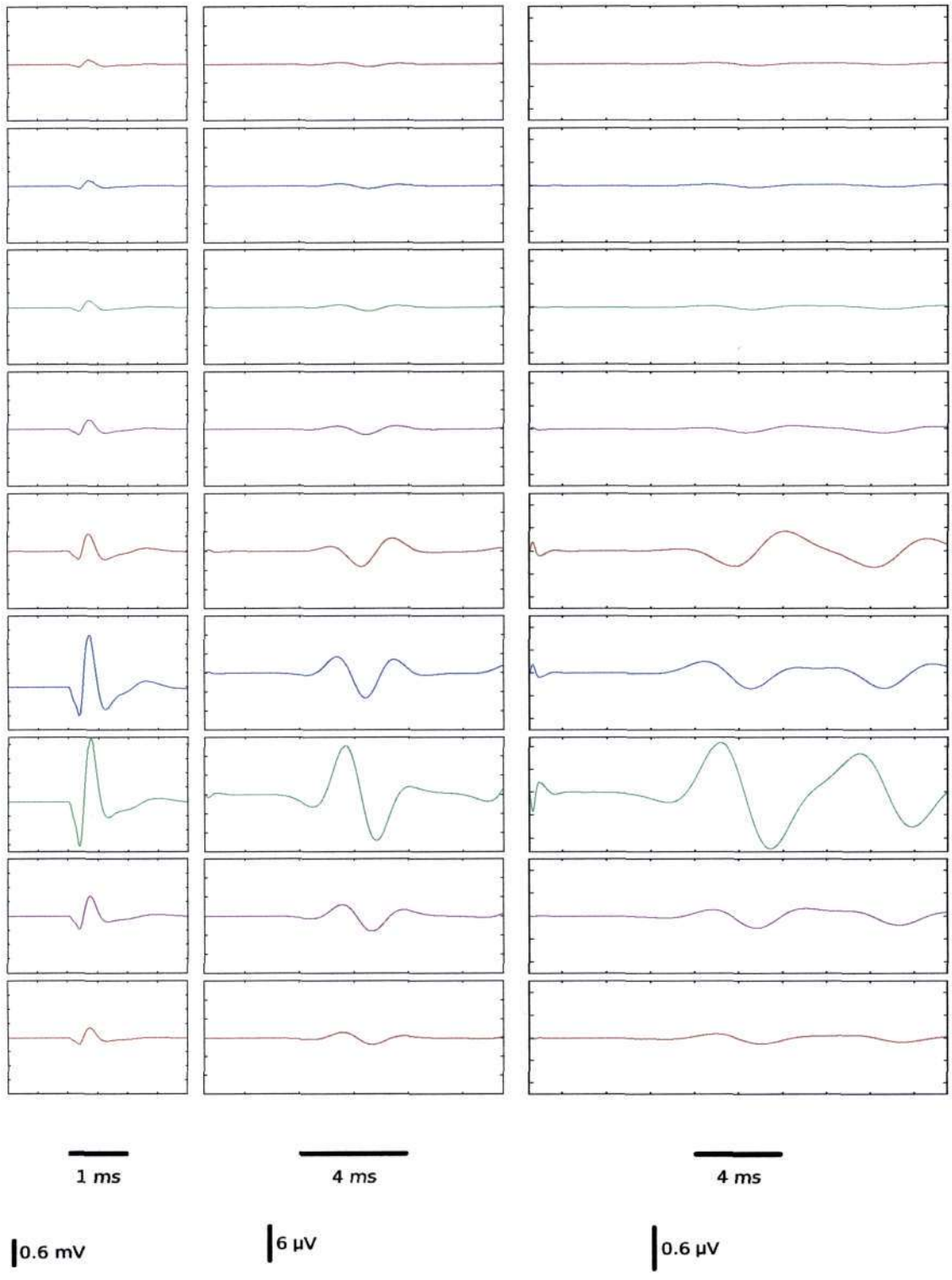


Figure 7.15:
High-pass filtered extracellular potential traces for an asynchronous population of layer 3 pyramids in a cortical cylinder of radius 50 μm; σ_t has values 0, 1 and 2 ms.

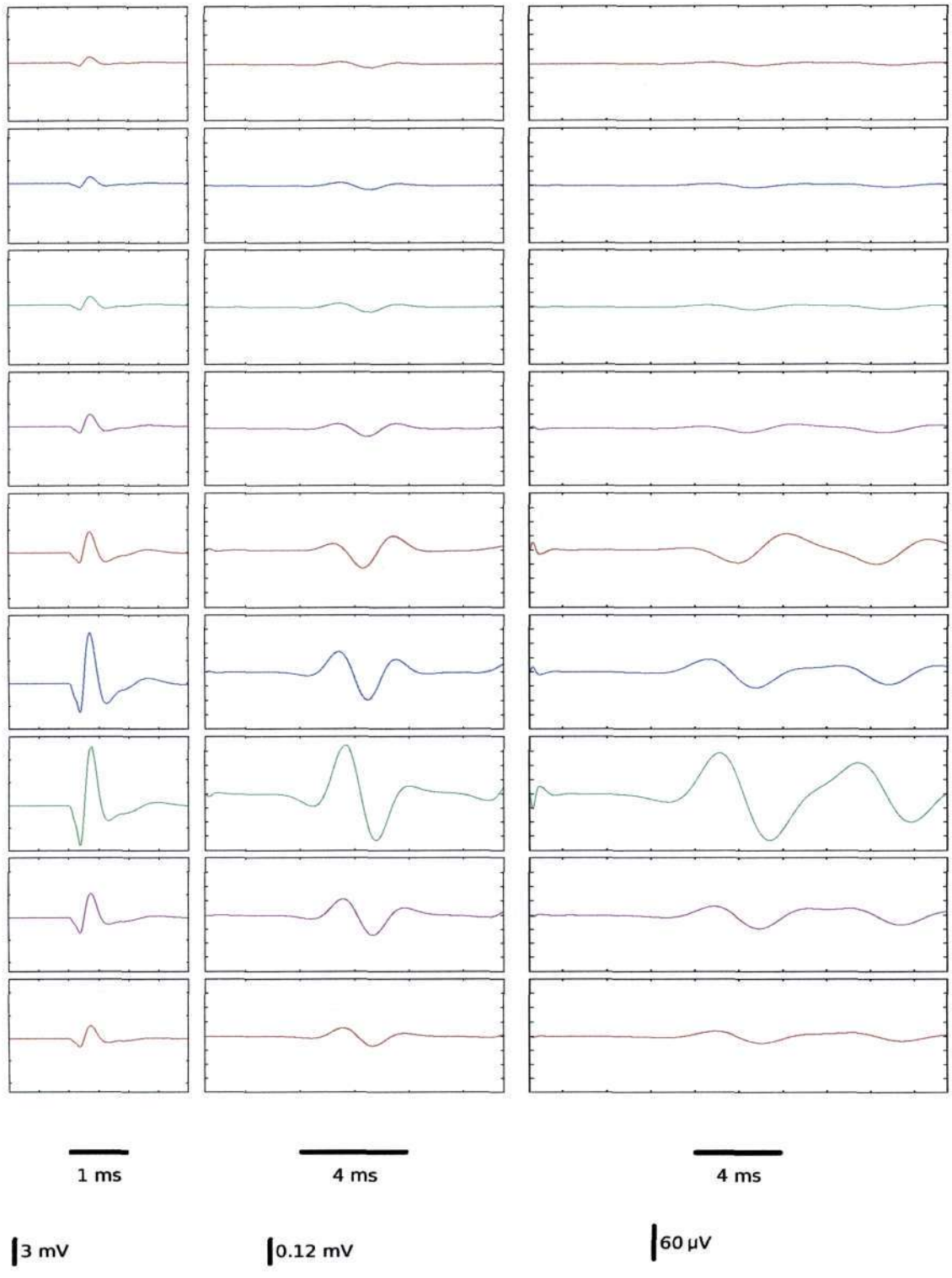


Figure 7.16:
High-pass filtered extracellular potential traces for an asynchronous population of layer 3 pyramids in a cortical cylinder of radius 100 μm; σ_t has values 0, 1 and 2 ms.

7.4 Layer 4 Populations

7.4.1 Synchronous Populations

Figures 7.17 and 7.18 show the extracellular potential around a population of synchronously firing layer 4 stellate neurons, for varying cylinder radii.

For the small radii the potential decreases faster with vertical distance from the soma. The shape of the traces varies little for larger populations (for R equal to 400, 800, 1000 and 2000 μm), the only difference being the amplitude. There is a point where the potential reaches a standard shape; here the system is said to approach the 'infinite-plane' limit.

The majority of current nodes in the cell are found in the region around the soma. These sources/sinks are small and 'point-like' for small R, causing the potential to decay rapidly away from the soma. For large R, the sources/sinks are 'plane-like' and therefore the potential decays slower.

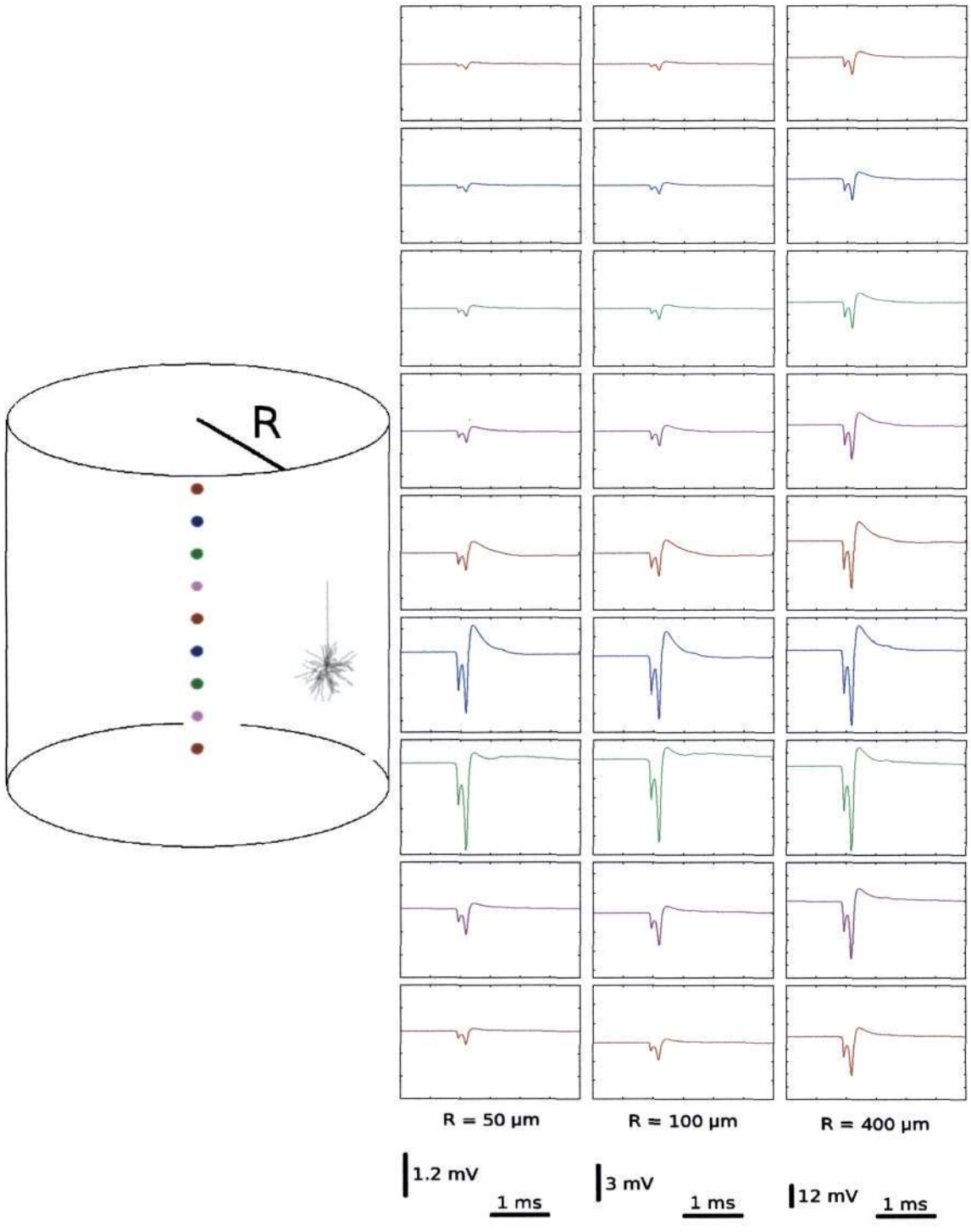


Figure 7.17:
Extracellular potential for a synchronous population of layer 4 stellates in a cortical cylinder for various radii (50, 100 and 400 μm).

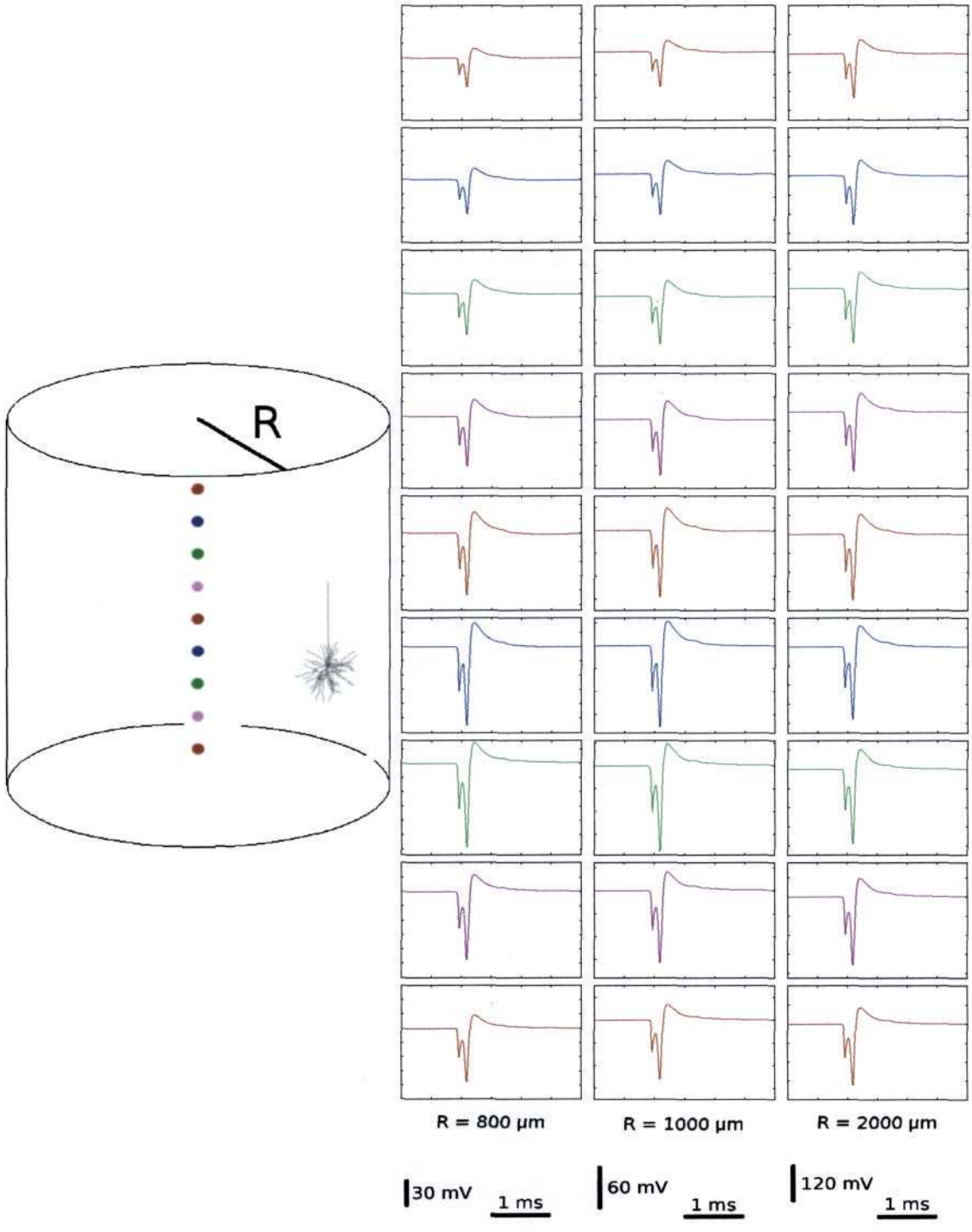


Figure 7.18:
Extracellular potential for a synchronous population of layer 4 stellates in a cortical cylinder for various radii (800, 1000 and 2000 μm).

7.4.2 High-Pass Filtered Synchronous Populations

Figures 7.19 and 7.20 show the filtered extracellular traces for synchronous populations of varying cylinder radii.

7.4.3 Asynchronous Populations

The results of the asynchronous layer 4 stellate populations are shown in Figure 7.21 and Figure 7.22.

The shape of the traces are similar to those of the layer 5 pyramid cell (Figures 7.5 and 7.6). The decay rate of the traces is faster in the layer 4 stellate, due to the size of the neuron.

7.4.4 High-Pass Filtered Asynchronous Populations

The filtered extracellular traces for asynchronous layer 4 stellate neurons are shown in Figures 7.23 and 7.24.

The filtered traces are very small in amplitude, and they decrease for large σ_t. This is due to the fact that the trace is both high-pass and low-pass filtered by the MUA filtering and time convolution respectively.

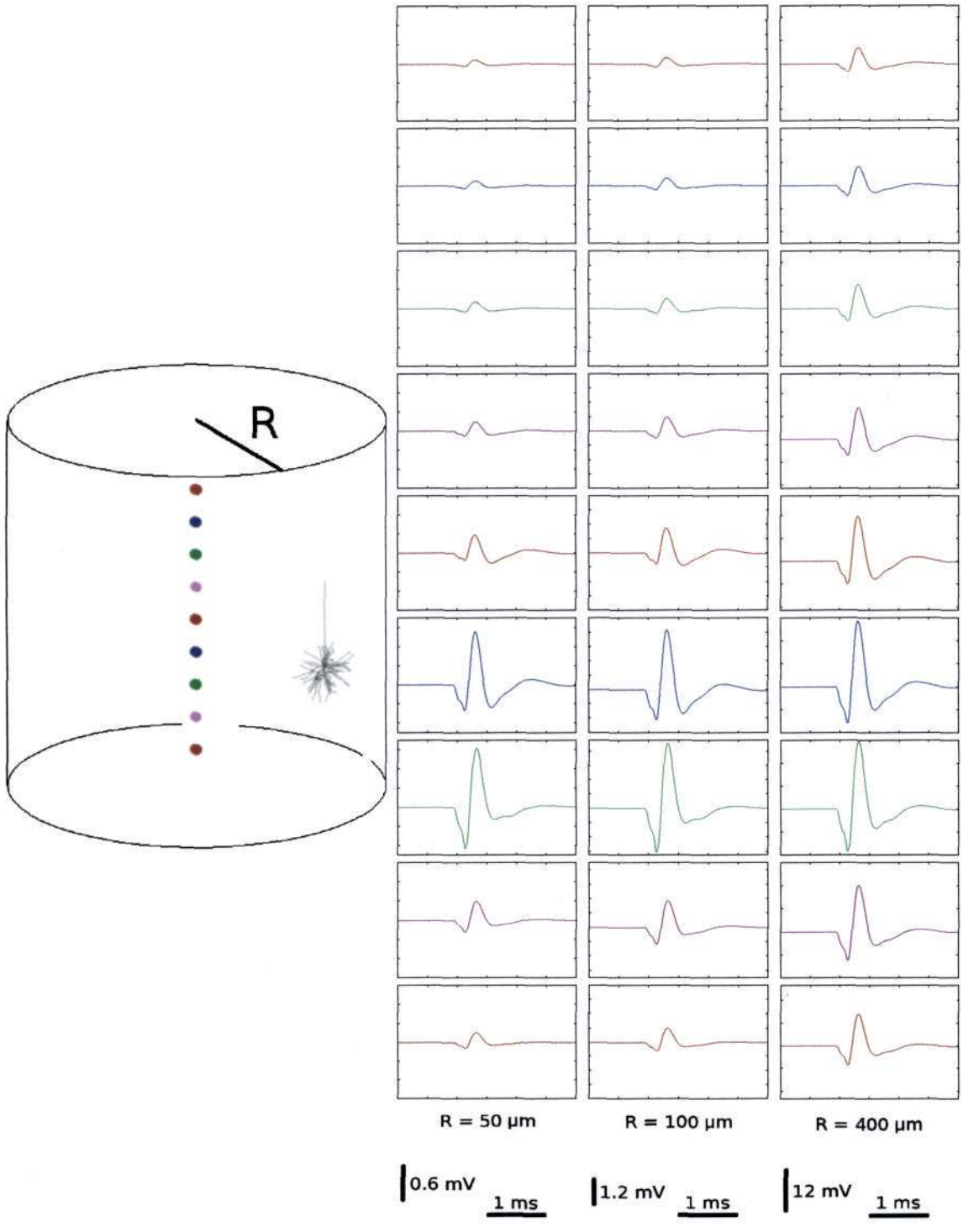


Figure 7.19:
High-pass filtered extracellular potential for a synchronous population of layer 4 stellates in a cortical cylinder for various radii (50, 100 and 400 μm).

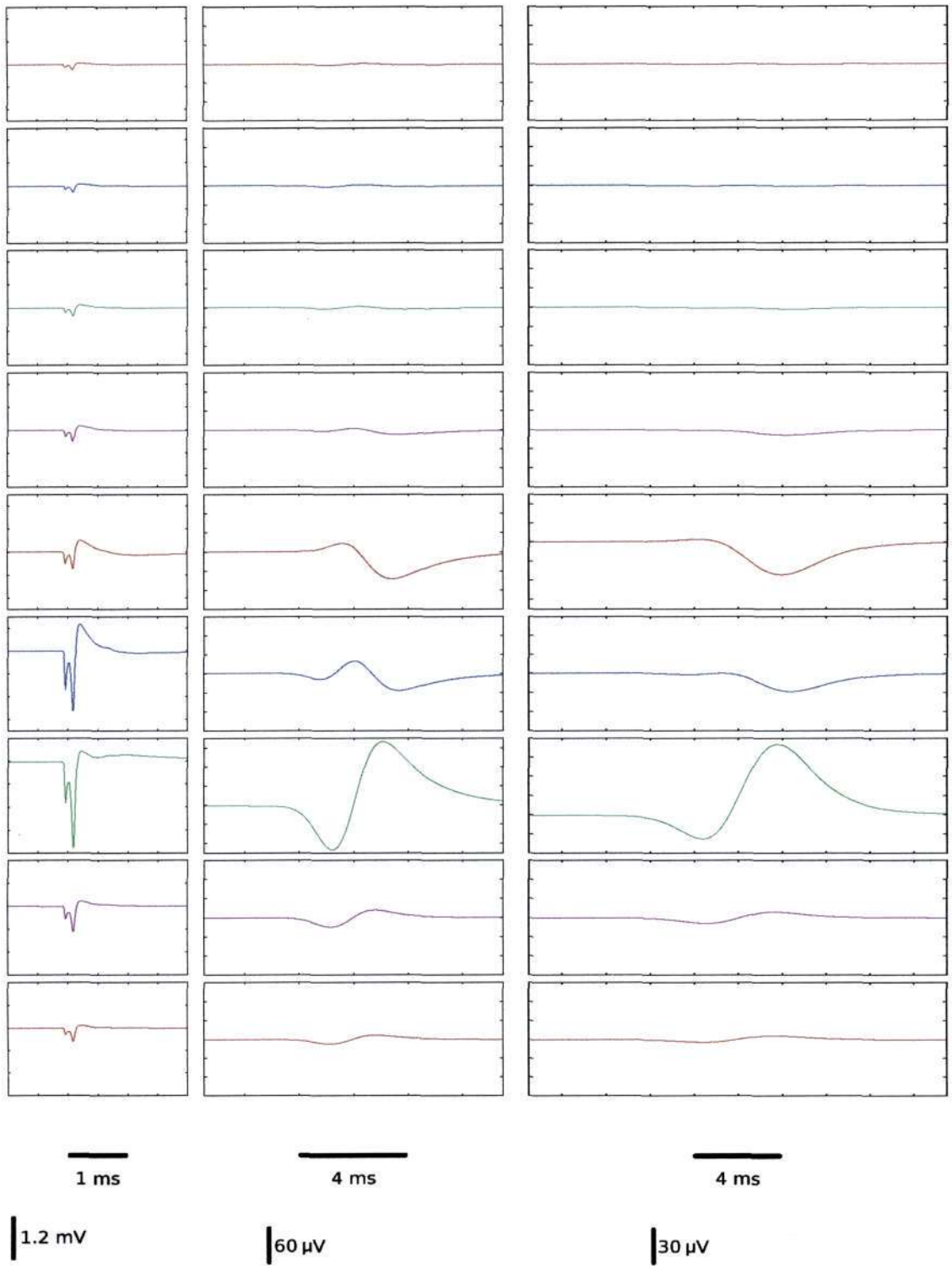


Figure 7.21:
Extracellular potential traces for an asynchronous population of layer 4 stellates in a cortical cylinder of radius 50 μm; σ_t has values 0, 1 and 2 ms.

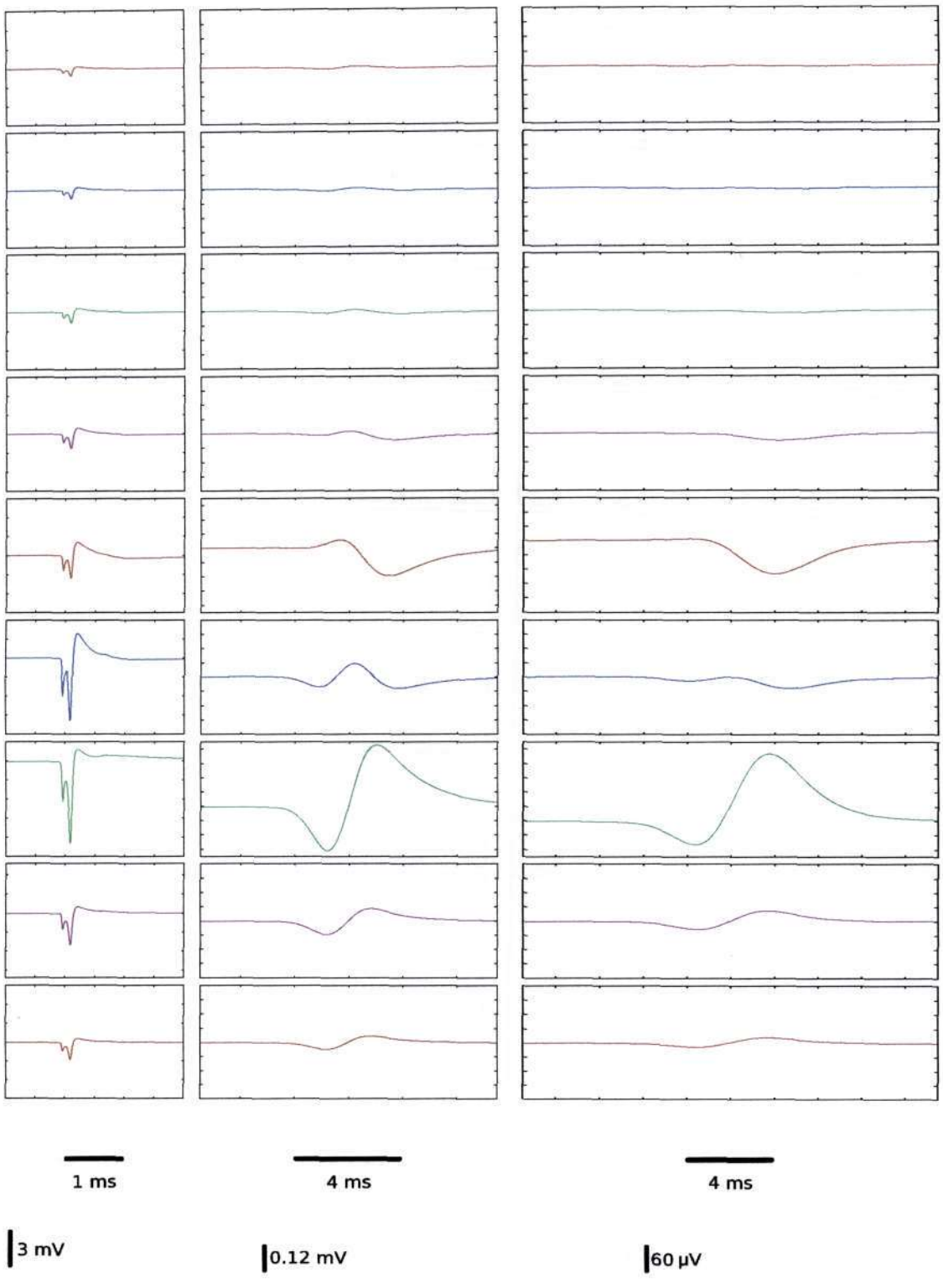


Figure 7.22:
Extracellular potential traces for an asynchronous population of layer 4 stellates in a cortical cylinder of radius 100 μm; σ_t has values 0, 1 and 2 ms.

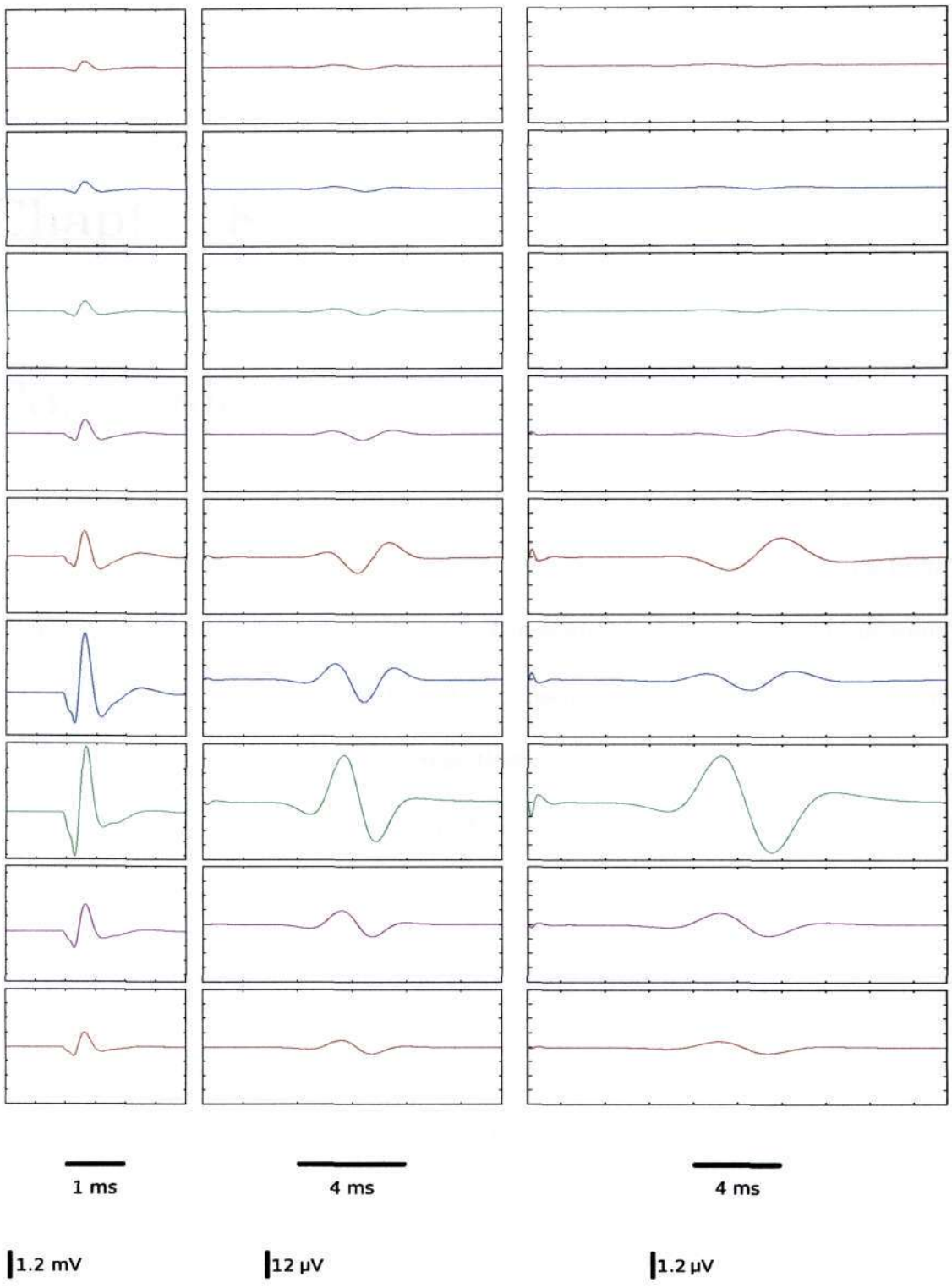


Figure 7.24:
High-pass filtered extracellular potential traces for an asynchronous population of layer 4 stellates in a cortical cylinder of radius 100 μm; σ_t has values 0, 1 and 2 ms.

In experiments the extracellularly recorded signatures of action potentials are commonly high-pass filtered to distinguish them from other neural processes. Such filtered potentials, called MUA in the laminar-electrode setting, have thus also been calculated using a high-pass Butterworth filter (750 - 3000 Hz).

Using compartmental modelling, the neuron is constructed as series of connected cables which form individual compartments. These compartments are then nodes for currents to either move in or out of the neuron. The current nodes are either sources or sinks and it was seen that these determine the overall shape of the extracellular trace. The layer 5 pyramid produced extracellular traces that had peaks that changed sign depending on whether the traces were computed above or below the soma (see Figure 6.6). The corresponding traces for the layer 3 pyramid show no such 'polarity' changes (Figure 6.14). Comparing these plots with traces from the layer 4 stellate neuron (Figure 6.18) one can see that it falls somewhere in between the layer 5 and 3 cells. The difference comes from the position of soma, the region where the majority of the current nodes are around, and the axon that carries the action potential fired by the neuron. The different scenarios correspond to the axon extending away from the dendritic branching structure around the soma (layer 5 cell, 'open-field' configuration [8]), the dendrites almost totally enveloping the axon (layer 3 cell, 'closed-field' configuration) and the dendrites only partially enveloping the axon (layer 4 cell, see Figure 6.1).

The results of the simulations close to the soma of the neurons are in good agreement with the modelling results obtained for the layer 5 cell by Holt and Koch [9]. Recently, Somogyvári *et al* [33] measured the extracellular action potential signatures of 13 individual cells using laminar electrodes. They measured potentials with peak amplitude of about 0.1 mV, in qualitative agreement with our model results.

Neuron populations were simulated in cylinders, where the extracellular potential was calculated along the centre axis of the cylinder. The response due to a population of completely synchronous neurons is calculated using Equation 5.15 and Equation 7.6 for an asynchronous population. In general for the synchronous population, the decay in the amplitude of the traces is fast with vertical distance for small populations (radii 50 and 100 μm). This rate of decay is not so pronounced for larger populations. The traces all look similar for large populations but with different amplitudes reaching an 'infinite-plane' limit. When comparing with the single neuron results for the layer 5 cell, we find that the prominent 'open-field' asymmetry of the potential above and below the soma has been significantly reduced for the layer 5 population. Asynchronous populations produce traces in our model that are much smaller in amplitude and more extended in time than synchronous populations, reflecting the effect of a temporal convolution of contributions from several sources.

The MUA was simulated by filtering the extracellular traces. The filtered results again show rapid decay in amplitudes of the traces for small populations. The traces

have similar shape for the three neurons, although the amplitudes are smaller. The filtered traces for asynchronous populations results in the smallest amplitudes, due to the fact that there is high-pass filtering as well as low-pass filtering (from the time convolution in Equation 7.6).

Laminar electrodes provide information that can help one understand neuronal activity. By simulating extracellular potentials due to single neurons as well as populations it could be possible to further understand the correlation between action potentials and the extracellular responses they generate.

The main result of this thesis is the demonstration of how the extracellular signatures of action potentials can be calculated from reconstructed realistic neurons. Since the contributions from the individual neurons add linearly, the calculation of responses from many neurons (i.e., a population) is in principle straightforward. Here we have looked into a particular population model where a cylindrical firing activity pattern is assumed, but the present approach can straightforwardly be modified to other situations as well. As more experimental data becomes available, one should thus be in the position to use this modelling approach to learn more about single neuron and neuron population activity.

Appendix A

Activation Variables and Parameters Used In Simulations

The current from each channel is given by an equation of the form

$$I = \overline{g}\, a^x b\, (V - E), \tag{A.1}$$

where $\overline{g}$ is the specific conductance, a and b are the activation and inactivation variables respectively with x-order kinetics, V is the membrane potential of the compartment and E is the reversal potential of the channel.

The activation and inactivation variables for the ion channels are calculated using a first-order reaction scheme given by

$$\tau_a(V)\frac{da}{dt} = a_\infty(V) - a, \tag{A.2}$$

which describes the rate at which the activation variable changes with time. Here a_∞ and τ_a are

$$a_\infty(V) = \frac{\alpha(V)}{\alpha(V)+\beta(V)} \tag{A.3}$$

$$\tau_a = \frac{1}{\alpha+\beta}, \tag{A.4}$$

the steady state value and the time constant respectively. The variable a approaches the value a_∞ exponentially with time constant τ_∞ when V remains fixed [27].

The rate functions for the different current types are as follows, where the potential V is given in mV [8]:

I_{Na} activation ($x = 3$):

$$\alpha = \frac{0.182(V+25)}{1-\exp(-(V-25)/9)} \tag{A.5}$$

$$\beta = \frac{-0.124(V+25)}{1-\exp((V+25)/9)}. \tag{A.6}$$

I_{Na} inactivation:

$$\alpha = \frac{0.024(V+40)}{1-\exp(-(V-40)/5)} \tag{A.7}$$

$$\beta = \frac{-0.009(V+65)}{\exp((V+65)/5)} \tag{A.8}$$

$$b_{\mathrm{infinity}} = \frac{1}{1+\exp((V+55)/6.2)}. \tag{A.9}$$

I_{ca} inactivation ($x = 2$):

$$\alpha = \frac{0.055(V+27)}{1-\exp(-(V+27)/3.8)} \tag{A.10}$$

$$\beta = 0.94\exp((V+75)/17). \tag{A.11}$$

I_{Ca} activation:

$$\alpha = 4.57 \times 10^{-4} \exp(-(V-25)/9) \tag{A.12}$$

$$\beta = \frac{0.0065(V+25)}{1+\exp(-(V+15)/28)}. \tag{A.13}$$

I_{Kv} activation ($x = 1$):

$$\alpha = \frac{0.02(V-25)}{1-\exp(-(V-25)/9)} \tag{A.14}$$

$$\beta = \frac{-0.002(V-25)}{1-\exp((V-25)/9)}. \tag{A.15}$$

I_{Km} activation ($x = 1$):

$$\alpha = \frac{1 \times 10^{-4}(V+30)}{1-\exp(-(V+30)/9)} \tag{A.16}$$

$$\beta = \frac{-1.10^{-4}(V+30)}{1-\exp((V-30)/9)}. \tag{A.17}$$

I_{KCa} activation ($x = 1$):

$$\alpha([\mathrm{Calcium}^{2+}]_{\mathrm{i}}) = 0.001 \times [\mathrm{Calcium}^{2+}]_{\mathrm{i}} \tag{A.18}$$

$$\beta = 0.02. \tag{A.19}$$

Reversal Potential	$E_{\text{leak}} = -70$ mV
	$E_{\text{K}} = -90$ mV
	$E_{\text{Na}} = 50$ mV
	$E_{\text{Ca}} = 140$ mV
	$E_{\text{syn}} = 0$ mV
Specific membrane capacitance	$C_{\text{m}} = 0.75\ \mu\text{Fcm}^{-2}$
	myelinated axon, $C_{\text{m}} = 0.02\ \mu\text{Fcm}^{-2}$
Specific membrane resistance	$R_{\text{m}} = 30\ \text{k}\Omega\text{cm}^2$
	axon nodes, $R_{\text{m}} = 50\ \text{k}\Omega\text{cm}^2$
Specific axial resistance	150 Ωcm

Table A.1:
Values of certain parameters used in the NEURON simulation: specific membrane resistance and capacitance, specific axial resistance and reversal potentials for the different ion channels in the simulation [8]. The synaptic conductance inserted throughout the neuron to generate action potentials in the cell has a reversal potential of E_{syn}.

Conductance densities	Dendrites:	$\overline{g}_{\mathrm{Na}} = 20$ pS
		$\overline{g}_{\mathrm{Ca}} = 0.3$ pS
		$\overline{g}_{\mathrm{KCa}} = 3$ pS
		$\overline{g}_{\mathrm{Kv}} = 0.1$ pS
	Soma:	$\overline{g}_{\mathrm{Na}} = 20$ pS
		$\overline{g}_{\mathrm{Ca}} = 0.3$ pS
		$\overline{g}_{\mathrm{KCa}} = 3$ pS
		$\overline{g}_{\mathrm{Kv}} = 200$ pS
	Axon hillock, initial segment	$\overline{g}_{\mathrm{Kv}} = 2000$ pS
		$\overline{g}_{\mathrm{Na}} = 30000$ pS
	Nodes of Ranvier	$\overline{g}_{\mathrm{Na}} = 30000$ pS

Table A.2:
Conductance densities for the different sections in the simulation [8]. The densities are the same for the three neuron types.

Appendix B

NEURON And Octave Simulation Code

B.1 Layer 5 Pyramid Cell Simulation Code

```
/* ------------------------------------------------------------
L5P.hoc

Program to set up a layer 5 pyramid cell from Mainen and Sejnowski.
The neuron contains synaptic conductances that are used to generate action
potentials. The extracellular potential can be calculated at any point; the
positions of each current node and the current entering/leaving the cell at
each time step are written to a file.

------------------------------------------------------------ */

tstop = 100    // Simulation time in ms
dt = 0.01
steps_per_ms = 1/dt

secondorder = 0
total_segments = 479
rmin = 0
graphcounter = 0
objref st, sh, sh2, dendritic, dendritic_only
```

```
objref currentsource[tstop/dt+1]
objref geom[3]
objref xpot

for i = 0,2 {
    // Store 3D positions of each current node; 1 point in each segment
    geom[i] = new Vector(total_segments)
}

// Store extracellular potential at each time step
xpot = new Vector(tstop/dt + 1)

for i = 0,tstop/dt {
    // Store currents in each current node at every time step
    currentsource[i] = new Vector(total_segments)
}

//------------------------
// Declare default constants

x_measurepoint = 10
y_measurepoint = 0
z_measurepoint = 0
xconductivity = 0.3    // S/m, e.g. Ranck (1963)

// ------------------------

create soma
access soma

//Set up the properties for the active and passive membrane conductances

ra = 150
global_ra = ra
rm = 30000
c_m = 0.75
cm_myelin = 0.04
g_pas_node = 0.02

v_init = -70
celsius = 37

Ek = -90
Ena = 60
```

```
gna_dend = 20
gna_node = 30000
gna_soma = gna_dend

gkv_axon = 2000
gkv_soma = 200

gca = 0.3
gkm = 0.1
gkca = 3

gca_soma = gca
gkm_soma = gkm
gkca_soma = gkca

n_axon_seg = 5
create iseg, hill, myelin[n_axon_seg], node[n_axon_seg]

conductance_type = 0

// Construct neuron
// ------------------------------------------------
// procedure create_axon

proc create_axon() {
     create iseg, hill, myelin[n_axon_seg], node[n_axon_seg]
     soma {
          equiv_diam = sqrt(area(0.5)/(4 * PI))
          // area = equiv_diam ^ 2 * 4 * PI
     }

     if (numarg()) equiv_diam = $1

     // total number of segments is 40 for myelinated axon

     iseg {    // initial segment between hillock + myelin
          // new position for section iseg

          pt3dclear()
          pt3dadd(0,-39.7838,0,equiv_diam/10)
          pt3dadd(0,-54.7838,0,equiv_diam/10)

          // — Original code
          nseg = 5
          L = 15
```

```
    diam = equiv_diam/10    // see Sloper and Powell 1982, Fig.71
    // — Original code
}

hill {

    //new position for section hill
    pt3dadd(0,-29.7838,0,4*iseg.diam)
    pt3dadd(0,-39.7838,0,  iseg.diam)
    nseg = 5

    }

// construct myelinated axon with nodes of ranvier

for i = 0,n_axon_seg-1 {

    myelin[i] {    // myelin element

            // new positions for sections myelin
            myelin_const = 54.7838 + (i * 100) + (i * 1.0)
            myelin_const = -myelin_const
            pt3dadd(0,myelin_const,    ,0,iseg.diam)
            pt3dadd(0,myelin_const-100,0,iseg.diam)
            nseg = 5

    }

    node[i] {    // nodes of Ranvier

            // new positions for sections node
            node_const = 154.7838 + (i * 1.0) + (i * 100)
            node_const = -node_const
            pt3dadd(0,node_const  ,0,iseg.diam*0.75)
            pt3dadd(0,node_const-1,0,iseg.diam*0.75)
            nseg = 1

    }

}

soma connect hill(0), 0.5
hill connect iseg(0), 1
iseg connect myelin[0](0), 1
myelin[0] connect node[0](0), 1
```

```
    for i = 0,n_axon_seg-2 {
        node[i] connect myelin[i+1](0), 1
        myelin[i+1] connect node[i+1](0), 1
    }
}

// end procedure create_axon
spine_dens = 1
spine_area = 0.83    // um^2 — K Harris

//------------------------------------------------------------
// procedure add_spines

proc add_spines() {local a
    forsec $o1 {
            a = 0
            for (x) a = a + area(x)
            F = (L * spine_area * spine_dens + a)/a
            L = L * F^(2/3)
            for (x) diam(x) = diam(x) * F^(1/3)
    }
}

// end procedure add_spines

//------------------------------------------------------------
// procedure init_cell

proc init_cell() {

    forall {
            insert pas
            Ra = ra
            cm = c_m
            g_pas = 1/rm
            e_pas = v_init
    }

    // synaptic conductances (to make the neuron fire automatically)

    forall {
            insert pas2
            Ra = ra
            cm = c_m
```

```
        g_pas2 = 1/rm * 0.2
        e_pas2 = 0
}

// exceptions along the axon
forsec "myelin" cm = cm_myelin
forsec "node" g_pas = g_pas_node

// Insert na+ channels
forall insert na
forsec dendritic gbar_na = gna_dend
forsec "myelin" gbar_na = gna_dend
hill.gbar_na = gna_node
iseg.gbar_na = gna_node
forsec "node" gbar_na = gna_node

// Insert kv delayed rectifier channels
iseg  insert kv gbar_kv = gkv_axon
hill  insert kv gbar_kv = gkv_axon
soma  insert kv gbar_kv = gkv_soma

// dendritic channels
forsec dendritic {
        insert km   gbar_km = gkm
        insert kca  gbar_kca = gkca
        insert ca   gbar_ca = gca
        insert cad
}

soma {
     gbar_na = gna_soma
     gbar_km = gkm_soma
     gbar_kca = gkca_soma
     gbar_ca = gca_soma
}

forall if(ismembrane("k_ion")) ek = Ek

forall if(ismembrane("na_ion")) {
        ena = Ena
        // seems to be necessary for 3d cells to shift Na
        // kinetics -5 mV
        vshift_na = -5
}
```

```
        forall if(ismembrane("ca_ion")) {
                eca = 140
                ion_style("ca_ion",0,1,0,0,0)
                vshift_ca = 0
        }
}

// end procedure init_cell

// ------------------------------------------------------------
// procedure load_cell

proc load_cell() {

        aspiny = 0
        forall delete_section()
        xopen("cells/temp-cell5.hoc")     // Contains cell geometry
        // Contains the layer 5 pyramid geometry
        access soma

        dendritic = new SectionList()

        // make sure no compartments exceed 50 uM length
        forall {
                diam_save = diam
                n = L/50
                nseg = n + 1
                if (n3d() == 0) diam = diam_save
                dendritic.append()
        }

        dendritic_only = new SectionList()
        forsec dendritic dendritic_only.append()
        soma dendritic_only.remove()

        create_axon()
        init_cell()
        if (!aspiny) add_spines(dendritic_only, spine_dens)
        xopen("L5P-moduleA.hoc")

}

// end procedure load_cell

load_cell()
```

B.2 Module For Layer 5 Pyramid Cell Simulation Code

```
/* ------------------------------------------------------------
L5P-moduleA.hoc
Module for program L5P.hoc
Extracellular potentials for a given point are calculated and either printed.
to a file or graph; the currents entering/leaving each node, for each time
step, are written to a file along with the 3D positions of each node

------------------------------------------------------------ */


tstop = 100
dt = 0.01

// ------------------------------------------------------------
// procedure xpot_simulation

proc xpot_simulation() {
     for i = 0,tstop/dt {
         compute_currentsources(i)
         fadvance()
     }
}

// ------------------------------------------------------------
// procedure init_geom

// Set up the 3D positions of each current node; 1 per segment.

proc init_geom() {

     j = 0
     soma for (x){
          if (x != 0) {
             if (x != 1) {
                xlength = x3d(n3d() - 1) - x3d(0)
                ylength = y3d(n3d() - 1) - y3d(0)
                zlength = z3d(n3d() - 1) - z3d(0)

                geom[0].x[j] = x3d(0) + xlength * x
```

```
            geom[1].x[j] = y3d(0) + ylength * x
            geom[2].x[j] = z3d(0) + zlength * x

            j = j + 1
         }
      }
}

for k = 0,n_axon_seg-1 {
    node[k] for (x) {
            if (x != 0) {
               if (x != 1) {
                  xlength = x3d(n3d() - 1) - x3d(0)
                  ylength = y3d(n3d() - 1) - y3d(0)
                  zlength = z3d(n3d() - 1) - z3d(0)

                  geom[0].x[j] = x3d(0) + xlength * x
                  geom[1].x[j] = y3d(0) + ylength * x
                  geom[2].x[j] = z3d(0) + zlength * x

                  j = j + 1
               }
            }
    }
}

hill for (x) {
      if (x != 0) {
         if (x != 1) {
            xlength = x3d(n3d() - 1) - x3d(0)
            ylength = y3d(n3d() - 1) - y3d(0)
            zlength = z3d(n3d() - 1) - z3d(0)

            geom[0].x[j] = x3d(0) + xlength * x
            geom[1].x[j] = y3d(0) + ylength * x
            geom[2].x[j] = z3d(0) + zlength * x

            j = j + 1
         }
      }
}

iseg for (x) {
      if (x != 0) {
         if (x != 1) {
```

```
            xlength = x3d(n3d() - 1) - x3d(0)
            ylength = y3d(n3d() - 1) - y3d(0)
            zlength = z3d(n3d() - 1) - z3d(0)

            geom[0].x[j] = x3d(0) + xlength * x
            geom[1].x[j] = y3d(0) + ylength * x
            geom[2].x[j] = z3d(0) + zlength * x

            j = j + 1
        }
    }
}

for k = 0,8 {
    dend1[k] for (x) {
            if (x != 0) {
               if (x != 1) {
                  xlength = x3d(n3d() - 1) - x3d(0)
                  ylength = y3d(n3d() - 1) - y3d(0)
                  zlength = z3d(n3d() - 1) - z3d(0)

                  geom[0].x[j] = x3d(0) + xlength * x
                  geom[1].x[j] = y3d(0) + ylength * x
                  geom[2].x[j] = z3d(0) + zlength * x

                  j = j + 1
               }
            }
    }
}

for k = 0,12 {
    dend2[k] for (x) {
            if (x != 0) {
               if (x != 1) {
                  xlength = x3d(n3d() - 1) - x3d(0)
                  ylength = y3d(n3d() - 1) - y3d(0)
                  zlength = z3d(n3d() - 1) - z3d(0)

                  geom[0].x[j] = x3d(0) + xlength * x
                  geom[1].x[j] = y3d(0) + ylength * x
                  geom[2].x[j] = z3d(0) + zlength * x

                  j = j + 1
               }
```

```
            }
    }
}

for k = 0,4 {
    dend3[k] for (x) {
            if (x != 0) {
               if (x != 1) {
                  xlength = x3d(n3d() - 1) - x3d(0)
                  ylength = y3d(n3d() - 1) - y3d(0)
                  zlength = z3d(n3d() - 1) - z3d(0)

                  geom[0].x[j] = x3d(0) + xlength * x
                  geom[1].x[j] = y3d(0) + ylength * x
                  geom[2].x[j] = z3d(0) + zlength * x

                  j = j + 1
               }
            }
    }
}

for k = 0,6 {
    dend4[k] for (x) {
            if (x != 0) {
               if (x != 1) {
                  xlength = x3d(n3d() - 1) - x3d(0)
                  ylength = y3d(n3d() - 1) - y3d(0)
                  zlength = z3d(n3d() - 1) - z3d(0)

                  geom[0].x[j] = x3d(0) + xlength * x
                  geom[1].x[j] = y3d(0) + ylength * x
                  geom[2].x[j] = z3d(0) + zlength * x

                  j = j + 1
               }
            }
    }
}

for k = 0,0 {
    dend5[k] for (x) {
            if (x != 0) {
               if (x != 1) {
                  xlength = x3d(n3d() - 1) - x3d(0)
```

```
                ylength = y3d(n3d() - 1) - y3d(0)
                zlength = z3d(n3d() - 1) - z3d(0)

                geom[0].x[j] = x3d(0) + xlength * x
                geom[1].x[j] = y3d(0) + ylength * x
                geom[2].x[j] = z3d(0) + zlength * x

                j = j + 1
            }
        }
    }
}

for k = 0,10 {
    dend6[k] for (x) {
            if (x != 0) {
               if (x != 1) {
                xlength = x3d(n3d() - 1) - x3d(0)
                ylength = y3d(n3d() - 1) - y3d(0)
                zlength = z3d(n3d() - 1) - z3d(0)

                geom[0].x[j] = x3d(0) + xlength * x
                geom[1].x[j] = y3d(0) + ylength * x
                geom[2].x[j] = z3d(0) + zlength * x

                j = j + 1
            }
        }
    }
}

for k = 0,12 {
    dend7[k] for (x) {
            if (x != 0) {
               if (x != 1) {
                xlength = x3d(n3d() - 1) - x3d(0)
                ylength = y3d(n3d() - 1) - y3d(0)
                zlength = z3d(n3d() - 1) - z3d(0)

                geom[0].x[j] = x3d(0) + xlength * x
                geom[1].x[j] = y3d(0) + ylength * x
                geom[2].x[j] = z3d(0) + zlength * x

                j = j + 1
            }
```

```
            }
    }
}

for k = 0,4 {
    dend8[k] for (x) {
            if (x != 0) {
               if (x != 1) {
                  xlength = x3d(n3d() - 1) - x3d(0)
                  ylength = y3d(n3d() - 1) - y3d(0)
                  zlength = z3d(n3d() - 1) - z3d(0)

                  geom[0].x[j] = x3d(0) + xlength * x
                  geom[1].x[j] = y3d(0) + ylength * x
                  geom[2].x[j] = z3d(0) + zlength * x

                  j = j + 1
               }
            }
    }
}

for k = 0,4 {
    dend9[k] for (x) {
            if (x != 0) {
               if (x != 1) {
                  xlength = x3d(n3d() - 1) - x3d(0)
                  ylength = y3d(n3d() - 1) - y3d(0)
                  zlength = z3d(n3d() - 1) - z3d(0)

                  geom[0].x[j] = x3d(0) + xlength * x
                  geom[1].x[j] = y3d(0) + ylength * x
                  geom[2].x[j] = z3d(0) + zlength * x

                  j = j + 1
               }
            }
    }
}

for k = 0,10 {
    dend10[k] for (x) {
             if (x != 0) {
                if (x != 1) {
                   xlength = x3d(n3d() - 1) - x3d(0)
```

```
                        ylength = y3d(n3d() - 1) - y3d(0)
                        zlength = z3d(n3d() - 1) - z3d(0)

                        geom[0].x[j] = x3d(0) + xlength * x
                        geom[1].x[j] = y3d(0) + ylength * x
                        geom[2].x[j] = z3d(0) + zlength * x

                        j = j + 1
                    }
                }
        }
    }

    for k = 0,82 {
        dend11[k] for (x) {
                if (x != 0) {
                    if (x != 1) {
                        xlength = x3d(n3d() - 1) - x3d(0)
                        ylength = y3d(n3d() - 1) - y3d(0)
                        zlength = z3d(n3d() - 1) - z3d(0)

                        geom[0].x[j] = x3d(0) + xlength * x
                        geom[1].x[j] = y3d(0) + ylength * x
                        geom[2].x[j] = z3d(0) + zlength * x

                        j = j + 1
                    }
                }
        }
    }

}
```

// end procedure init_geom

// ---

// procedure compute_currentsources

// Compute current entering/leaving each current node at every time step.

```
proc compute_currentsources() {local xlength, ylength, zlength
 i = $1
 j = 0

 soma for (x) {
```

```
 if (x != 0) {
  if (x != 1) {
  currentsource[i].x[j]=(i_cap(x)+i_pas(x)+ica(x)+ik(x)+ina(x)+i_pas2(x))*are
  // 10^-11 Ampere
  j = j + 1
  }
 }
}

for k = 0,4 {
 node[k] for (x) {
  if (x != 0) {
   if (x != 1) {
    currentsource[i].x[j] = (i_cap(x)+i_pas(x)+ina(x)+i_pas2(x))*area(x)
    j = j + 1
   }
  }
 }
}

for k = 0,4 {
 myelin[k] for (x) {
  if (x != 0) {
   if (x != 1) {
    currentsource[i].x[j] = (i_cap(x)+i_pas(x)+ina(x)+i_pas2(x))*area(x)
    j = j + 1
   }
  }
 }
}

hill[k] for (x) {
 if (x != 0) {
  if (x != 1) {
   currentsource[i].x[j] = (i_cap(x)+i_pas(x)+ik(x)+ina(x)+i_pas2(x))*area(x)
   j = j + 1
  }
 }
}

iseg[k] for (x) {
 if (x != 0) {
  if (x != 1) {
   currentsource[i].x[j] = (i_cap(x)+i_pas(x)+ik(x)+ina(x)+i_pas2(x))*area(x)
   j = j + 1
```

```
  }
 }
}

for k = 0,8 {
 dend1[k] for (x) {
  if (x != 0) {
   if (x != 1) {
   currentsource[i].x[j]=(i_cap(x)+i_pas(x)+ica(x)+ik(x)+ina(x)+i_pas2(x))*ar
   j = j + 1
   }
  }
 }
}

for k = 0,12 {
 dend2[k] for (x) {
  if (x != 0) {
   if (x != 1) {
   currentsource[i].x[j]=(i_cap(x)+i_pas(x)+ica(x)+ik(x)+ina(x)+i_pas2(x))*ar
   j = j + 1
   }
  }
 }
}

for k = 0,4 {
 dend3[k] for (x) {
  if (x != 0) {
   if (x != 1) {
   currentsource[i].x[j]=(i_cap(x)+i_pas(x)+ica(x)+ik(x)+ina(x)+i_pas2(x))*ar
   j = j + 1
   }
  }
 }
}

for k = 0,6 {
 dend4[k] for (x) {
  if (x != 0) {
   if (x != 1) {
   currentsource[i].x[j]=(i_cap(x)+i_pas(x)+ica(x)+ik(x)+ina(x)+i_pas2(x))*ar
   j = j + 1
   }
  }
```

```
 }
}

for k = 0,0 {
 dend5[k] for (x) {
  if (x != 0) {
   if (x != 1) {
   currentsource[i].x[j]=(i_cap(x)+i_pas(x)+ica(x)+ik(x)+ina(x)+i_pas2(x))*a
   j = j + 1
   }
  }
 }
}

for k = 0,10 {
 dend6[k] for (x) {
  if (x != 0) {
   if (x != 1) {
   currentsource[i].x[j]=(i_cap(x)+i_pas(x)+ica(x)+ik(x)+ina(x)+i_pas2(x))*a
   j = j + 1
   }
  }
 }
}

for k = 0,12 {
 dend7[k] for (x) {
  if (x != 0) {
   if (x != 1) {
   currentsource[i].x[j]=(i_cap(x)+i_pas(x)+ica(x)+ik(x)+ina(x)+i_pas2(x))*a
   j = j + 1
   }
  }
 }
}

for k = 0,4 {
 dend8[k] for (x) {
  if (x != 0) {
   if (x != 1) {
   currentsource[i].x[j]=(i_cap(x)+i_pas(x)+ica(x)+ik(x)+ina(x)+i_pas2(x))*a
   j = j + 1
   }
  }
 }
```

```
 }

 for k = 0,4 {
  dend9[k] for (x) {
   if (x != 0) {
    if (x != 1) {
    currentsource[i].x[j]=(i_cap(x)+i_pas(x)+ica(x)+ik(x)+ina(x)+i_pas2(x))*ar
    j = j + 1
    }
   }
  }
 }

 for k = 0,10 {
  dend10[k] for (x) {
   if (x != 0) {
    if (x != 1) {
    currentsource[i].x[j]=(i_cap(x)+i_pas(x)+ica(x)+ik(x)+ina(x)+i_pas2(x))*ar
    j = j + 1
    }
   }
  }
 }

 for k = 0,82 {
  dend11[k] for (x) {
   if (x != 0) {
    if (x != 1) {
    currentsource[i].x[j]=(i_cap(x)+i_pas(x)+ica(x)+ik(x)+ina(x)+i_pas2(x))*ar
    j = j + 1
    }
   }
  }
 }

}

// end procedure compute_currentsources

// ------------------------------------------------------------
// procedure change

// Compute the extracellular potential at the point (EP_x,EP_y,EP_z)

proc change() {local v_extracellular, j, r, rmin, k
```

```
  EP_x = $1
  EP_y = $2
  EP_z = $3
  sigma = $4

  print "xcell point:  (",EP_x,",",EP_y,",",EP_z,")"
  print "conductivity:  ", sigma

  for i = 0,tstop/dt {
    v_ex = 0
    for j = 0,total_segments-1 {
     r=sqrt((EP_x-geom[0].x[j])^2+(EP_y-geom[1].x[j])^2+(EP_z-geom[2].x[j])^2)
     if (i == 0) {
        if (j == 0) {
           rmin = r
           k = j
        }
        if (r < rmin) {
           rmin = r
           k = j
        }
     }

     v_ex = v_ex + currentsource[i].x[j]/(4 * PI * sigma * r)*10^-2   // mV
    }

    xpot.x[i] = v_ex
  }

  print "distance from measurepoint to nearest nodepoint:  ", rmin, "um"
  print "The node point is nodepoint ", k," at location:"
  print "(",geom[0].x[k],",",geom[1].x[k],",",geom[2].x[k],")"
  paintgraph(tstop, 2 /*, EP_x, EP_y, EP_z */)
  graphcounter = graphcounter + 1

// end procedure change

// ------------------------------------------------------------
// procedure paintgraph

// Display graph of extracellular trace from procedure change

proc paintgraph() { local i, ymin, ymax
     tstop = $1
     graphcolor = $2
```

```
    ymin = 0
    ymax = 0

    g[graphcounter] = new Graph()
    for i = 0,tstop/dt {
        if (xpot.x[i] > ymax) ymax = xpot.x[i]
        if (xpot.x[i] < ymin) ymin = xpot.x[i]
    }

    g[graphcounter].size(39, 45, -0.35, 0.35)    // focus graph window
    g[graphcounter].color(graphcolor)
    g[graphcounter].beginline()
    g[graphcounter].label("xcell")

    for i = 0,tstop/dt {
        t = i * dt
        g[graphcounter].line(t, xpot.x[i])
    }

}

// end procedure paintgraph

// ---------------------------------------------------
// procedure make_demopanel

proc make_demopanel() {

    xpanel("Simulations of extracellular fields outside L5P cell")
    xvalue("xpos", "x_measurept")
    xvalue("ypos", "y_measurept")
    xvalue("zpos", "z_measurept")
    xvalue("xconductivity", "xcond")
    xbutton("xfield","change(x_measurept,y_measurept,z_measurept,xcond)")
    xpanel()

}

// end procedure make_demopanel

// ---------------------------------------------------
// procedure Get_File_Names()

proc Get_File_Names() {
```

```
    Num = $1
    strdef tmp

    ropen("filenamelist13.dat")

    // Read in the marker for the number of files in the list.
    getstr(tmp)
    //print tmp

    // Read in the number of filenames in the list.
    temp2 = fscan()

    // Read the required filename
    for k = 1,Num {
        getstr(File_Name, 1)
    }

    ropen()
}

// end procedure Get_File_Names()

// ---------------------------------------------------------
// procedure EP_One_Axis()

proc EP_One_Axis() {

  strdef afilename
  Num_files = 11
  sigma = xconductivity
  EP_x = 200
  EP_z = 0

  EP_y = 200

  for i = 1,Num_files {

    Get_File_Names(i)
    print File_Name, " ", i
    afilename = File_Name

    // Print EP to file, afilename.
    print "Writing the extracellular potential to file ..."
    print "EP at point:  ( ", EP_x,", ", EP_y,", ", EP_z,")"
```

```
    wopen(afilename)

    fprint("L5P\n")
    fprint("EP Syn\n")
    fprint("This file contains the extracellular potential at:\n")
    fprint("( %9.5f, %9.5f, %9.5f )\n", EP_x, EP_y, EP_z)
    fprint("for a L5 pyramidal neuron, excited by synaptic\n")
    fprint("conductances throughout the neuron.\n")
    fprint("Miguel Cavero 12-10-2005.\n")
    fprint("----------------------------\n")
    fprint("STARTDATA\n")

    fprint("%g\n", dt)

    for j = 0,tstop/dt {

      V_ex = 0
      for k = 0,total_segments-1 {
        R=sqrt((EP_x-geom[0].x[k])^2+(EP_y-geom[1].x[k])^2+(EP_z-geom[2].x[k])

        V_ex = V_ex + currentsource[j].x[k]/(4 * PI * sigma * R)*10^-2    //m\
      }

      fprint("%5.2f %16.11f\n", j*dt, V_ex)

    }

    wopen()
    print "Finished writing to file."

    EP_y = EP_y - 40
  }
}

// end procedure EP_One_Axis()

// ------------------------------------------------------
// procedure show_cell

proc show_cell() {

// Display the neuron and the points at which the extracellular potential
// is calculated.
```

```
    xpt = -400
    ypt = 1100

    sh = new PlotShape()
    sh.size(-600,600,-600,1200)

    sh.mark(-400, 1100, "O", 8, 2, 1)
    sh.mark(-400,  900, "O", 8, 3, 1)
    sh.mark(-400,  700, "O", 8, 4, 1)
    sh.mark(-400,  500, "O", 8, 7, 1)
    sh.mark(-400,  300, "O", 8, 2, 1)
    sh.mark(-400,  100, "O", 8, 3, 1)
    sh.mark(-400, -100, "O", 8, 4, 1)
    sh.mark(-400, -300, "O", 8, 7, 1)
    sh.mark(-400, -500, "O", 8, 2, 1)

    sh.mark(400, 1100, "O", 8, 2, 1)
    sh.mark(400,  900, "O", 8, 3, 1)
    sh.mark(400,  700, "O", 8, 4, 1)
    sh.mark(400,  500, "O", 8, 7, 1)
    sh.mark(400,  300, "O", 8, 2, 1)
    sh.mark(400,  100, "O", 8, 3, 1)
    sh.mark(400, -100, "O", 8, 4, 1)
    sh.mark(400, -300, "O", 8, 7, 1)
    sh.mark(400, -500, "O", 8, 2, 1)

}

// end procedure show_cell

// ------------------------------------------------------------
// procedure print_data_file

// Print currents and current nodes to a data file.

proc print_data_file() {

  print "Writing current sources to file ...\n"
  wopen("L5P-Syn-CurrentSources-10msB.dat")

  // Print the heading.
  fprint("This file contains membrane currents and their respective\n")
  fprint("positions for a L5 pyramid neuron, excited by synaptic\n")
  fprint("conductances.\n")
  fprint("Miguel Cavero 12-10-2005.\n")
```

```
    fprint("------------------------------\n")
    fprint("STARTDATA\n")

    // Print dt.
    fprint("%g\n", dt)

    // Print the geometry of the neuron.
    for i = 0,2 {
        for j = 0,total_segments-1 {
            fprint("%9.5f\t", geom[i].x[j])
        }
        fprint("\n")
    }

    // Print the currents going out of each node for every dt.
    for i = 3000,4000 {
        for j = 0,total_segments-1 {
            fprint("%16.11f\t",currentsource[i].x[j])
        }
        fprint("\n")
    }

    wopen()
    print "Finished writing current sources."

}

// end procedure print_data_file

// ------------------------------------------------------------
// ------------------------------------------------------------

init()
init_geom()

xpot_simulation()

//show_cell()
//EP_One_Axis()
//print_data_file()
```

B.3 Octave Synchronous Population Simulation Code

```
#-------------------------------------------------------------
#
# sync_pop_sim.m
#
# Program to calculate the extracellular potential for a population of
# synchronously firing neurons, of varying population radii.
#
#-------------------------------------------------------------

1;

#-------------------------------------------------------------

function y3 = f3(z0)

  global R;
  global z_n;
  global z;
  global sigma_z;

  A = exp(-1 * (z0 - z_n)^2 / (2 * sigma_z^2));
  B = sqrt(R^2 + (z - z0)^2) - abs(z - z0);

  y3 = A * B;

endfunction

#-------------------------------------------------------------

# Function to calculate the extracellular potential at a depth z.

function Get_EP(filename, seg, I, pos, dt)

  global R;
  global z_n;
  global z;
  global sigma_z;
  global sigma_ex;
  global rho;

  lim1 = -600;
```

```
  lim2 = 1200;

  CONST = rho * (1 / (sqrt(2 * pi) * sigma_z)) * (1 / (2 * sigma_ex));

  FID2 = fopen(filename, 'w');    # Output file

  # Print header.
  fprintf (FID2, "Data file containing the extracellular potential around\n")
  fprintf (FID2, "a cortical cylinder containing identical synchronous\n");
  fprintf (FID2, "layer 3 pyramid neurons.\n");
  fprintf (FID2, "The cylinder has a radius %i, a sigma_z value of\n", R);
  fprintf (FID2, "%f.\n", sigma_z);
  fprintf (FID2, "The potential is calculated at a depth %i.\n", z);
  fprintf (FID2, "STARTDATA\n");
  fprintf (FID2, "%f\n", dt);

  for t = 350:650

      EP = 0;

      for i = 1:seg

          z_n = pos(3,i);    # z-coordinate of n-th segment
          I_n = I(t+1,i);    # current from n-th segment at time t

          [INT, ier, nfun, err] = quad("f3", lim1, lim2);
          EP = EP + (I_n * INT);

      endfor

      EP = EP * CONST;
      fprintf(FID2, "%f %f\n", t*dt, EP);

  endfor

  fclose (FID2)

endfunction
```

#---

Function to write various data files containing the extracellular potential
due to a population of identical synchronously firing cortical neurons, at
varying cortical depths.

```
function [COUNT] = Laminar_Calculation()

  global z;

  # Read in data from the input file.
  printf("Reading in data ...\n")
  [I_n,pos,dt]=read_input_data2('L3P-Syn-CurrentSources-10msB.dat');
  printf("Finished reading data.\n")

  # Read in output file names.
  FID1 = fopen('InputDataFiles/DataFilenames1.dat', 'r');

  # Marker
  tline = fgetl(FID1);
  # Read the number of file names.
  tline = fgetl (FID1);
  [num_files CNT] = sscanf(tline, "%i");

  # Calculate the extracellular potential for num_files depths and output to
  # file.

  COUNT = 1;
  total_seg = 257;    # L3P - 257; L4S - 199; L5P - 479
  z = 1100;    # Point on vertical axis where extracellular potential
               # is calculated.

  for i = 1:num_files

      # Read in file name
      filename = fgetl (FID1);
      # filename = "L3P-SyncEPz-500R50t0.dat"
      Get_EP(filename, total_seg, I_n, pos, dt);

      COUNT = COUNT + 1;

      z = z - 200;

  endfor

  fclose(FID1)

endfunction

#-------------------------------------------------------------------
```

```
global R;
global z_n;
global z;
global sigma_z;
global sigma_ex;
global rho;

sigma_z = 100;
sigma_ex = 0.3;
rho = 1;
R = 50;

clock()

A = Laminar_Calculation();
A

clock()

printf("Done.\n")
```

B.4 Octave Asynchronous Population Simulation Code

```
#-------------------------------------------------------------
#
# pop_sim-async4.m
#
# Script to simulate asynchronous neuron population spiking, using a data file
# of current sources for cortical neurons.
#
#-------------------------------------------------------------

1;

#-------------------------------------------------------------

function y3 = f3(z0)

  global z_n;
  global R;
  global z;
  global sigma_z;

  A = exp (-1 * (z0 - z_n)^2 / (2 * sigma_z^2));
  B = sqrt (R^2 + (z - z0)^2) - abs(z - z0);

  y3 = A * B;

endfunction

#-------------------------------------------------------------

# Function to calculate the extracellular potential at a depth z.

function Get_EP(filename, seg, I_n, pos, dt)

  global R;
  global z_n;
  global z;

  global sigma_z;
  global sigma_ex;
  global sigma_t;
  global rho;
```

```
lim1 = -600;
lim2 = 1200;

time_filter = 4 * sigma_t;

CONST = rho * (1/(sqrt(2 * pi) * sigma_z)) * (1/(2 * sigma_ex));
CONST = CONST * (1/(sigma_t * sqrt(pi)));

FID2 = fopen (filename, 'w');    # Output file

# Print header.
fprintf (FID2, "Data file containing the extracellular potential\n");
fprintf (FID2, "around a cortical cylinder containing identical\n");
fprintf (FID2, "asynchronous layer 4 stellate neurons.\n");
fprintf (FID2, "The cylinder has a radius %i,a sigma_z value of\n", R);
fprintf (FID2, "%f and a sigma_t value of %f.\n", sigma_z, sigma_t);
fprintf (FID2, "The potential is calculated at a depth %i.\n", z);
fprintf (FID2, "STARTDATA\n");
fprintf (FID2, "%f\n", dt);

for i = 1:seg

    z_n = pos(3,i);    # z-coordinate of n-th segment
    # time = t - t_prime + 1;

    [INT1, ier, nfun, err] = quad ("f3", lim1, lim2);

    if (i == 1)
       INT = [INT1];
    else
       INT = [INT; INT1];
    endif

endfor

for t = 850:2750

EP = 0;

    for t_prime = -time_filter:time_filter

        X = 0;
        time = t - t_prime + 1;
```

```
            for i = 1:seg
                X = X + I_n(time,i) * exp(-1 * t_prime^2/sigma_t^2) * INT(i);
            endfor

            EP = EP + X;

        endfor

        EP = EP * CONST;
        fprintf (FID2, "%f %f\n", t*dt, EP);

    endfor

    fclose(FID2)

endfunction

#-------------------------------------------------------------------

# Function to write various data files containing the extracellular potential
# due to a population of identical asynchronous cortical neurons.

function [COUNT] = Laminar_Calculation()

    global z;

    # Read in data from the input file.
    printf ("Reading in data ...\n")
    [I_n,pos,dt]=read_input_data2('L4S-Syn-CurrentSources-36msB.dat');
    printf ("Finished reading data.\n")

    # Read in the output file names.
    FID1 = fopen ('InputDataFiles/DataFilenames2.dat', 'r');

    # Marker
    tline = fgetl (FID1);
    # Read the number of file names.
    tline = fgetl (FID1);
    [num_files CNT] = sscanf (tline, "%i");

    # Calculate the extracellular potential for num_files depths and
    # output to file.

    COUNT = 1;
    total_seg = 199;     # L3P - 257; L4S - 199; L5P - 479
```

```
  z = -100;     # Point on vertical axis where extracellular potential
                # is calculated.

  for i = 1:num_files

      # Read in file name
      filename = fgetl (FID1)
      # filename = "L3P-ASynEPz300R100t2.dat";
      Get_EP(filename, total_seg, I_n, pos, dt);

      COUNT = COUNT + 1;

      z = z - 200;

      endfor

  fclose (FID1)

endfunction

global R;
global z_n;

global sigma_z;
global sigma_ex;
global sigma_t;
global rho;

sigma_z = 100;
sigma_ex = 0.3;
rho = 1;
R = 50;     # R1 = 50; R2 = 100; R3 = 200; R4 = 500
sigma_t = 2 /0.01;     # T1 = 1; T2 = 2; T3 = 4

clock()
A = Laminar_Calculation();
A
clock()

printf ("Done.\n")
```

Bibliography

[1] M.I. Sereno, A.M. Dale, J.B. Reppas, K.K. Kwong, J.W. Belliveau, T.J. Brady, B.R. Rosen, and R.B. Tootell. Borders of multiple visual areas in humans revealed by functional magnetic resonance imaging. *Science*, 268:889 – 893, 1995.

[2] A.L. Hodgkin and A.F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology*, 117:500 – 544, 1952.

[3] C. Koch and I. Segev. *Methods in Neuronal Modelling.* The MIT Press, 1998.

[4] J.M. Bower and D. Beeman. *The Book of Genesis: Exploring Realistic Neural Models with the General Neural Simulation System.* Springer, New York, 1998.

[5] Rodney Douglas, Henry Markram, and Kevan Martin. *The Synaptic Organisation of the Brain.* Oxford University Press, New York, 2004.

[6] István Ulbert, Eric Halgren, Gary Heit, and George Karmos. Multiple microelectrode-recordings system for human intracortical applications. *Journal of Neuroscience Methods*, pages 69 – 79, 2001.

[7] NEURON at Yale. http://www.neuron.yale.edu/neuron/.

[8] Zachary F. Mainen and Terrence J. Sejnowski. Influence of dendritic structure on firing pattern in model neocortical neurons. *Nature*, 382:363 – 366, 1996.

[9] Gary R. Holt and Christof Koch. Electrical interations via the extracellular potential near cell bodies. *Journal of Computational Neuroscience*, 6:169 – 184, 1999.

[10] Matti Hämäläinen, Riitta Hari, Risto J. Ilmoniemi, Jukka Knuutila, and Olli V. Lounasmaa. Magnetoencephalography - theory, instrumentation and applications to noninvasive studies of the working human brain. *Reviews of Modern Physics*, 65:413 – 497, 1993.

[11] Brain Connection. http://www.brainconnection.com/topics/?main=gal/cns-home.

[12] Karl Zilles. Receptor fingerprinting of cortical areas in the human brain.

[13] Cerebral Cortex. http://www.geocities.com/medinotes/cerebralcortex.htm.

[14] Christof Koch. *The Quest for Consciousness.* Roberts and Company Publishers, Englewood, Colorado, 2004.

[15] S. Murray Sherman and R.W. Guillery. *Exploring the Thalamus.* Academic Press, 2001.

[16] Alex M. Thomson and A. Peter Bannister. Interlaminar connections in the neocortex. *Cerebral Cortex*, 13:5 –14, 2003.

[17] Bryn Mawr College. http://serendip.brynmawr.edu/bb/kinser/nerve7.html.

[18] Jacob L. Driesen. http://www.driesen.com/synapse.htm.

[19] University of Washington. http://faculty.washington.edu/chudler/ap.html.

[20] Christoff Koch. *Biophysics of Computation - Information Processing in Single Neurons.* The Oxford University Press, 1999.

[21] ADInstruments. http://www.adinstruments.com/research/rapps/xtracellular.html.

[22] Didier Pinault, Nathalie Leresche, Stéphane Charpier, Jean-Michel Deniau, Christian Marescaux, Marguerite Vergnes, and Vincenzo Crunelli. Intracellular recordings in thalamic neurons during spontaneous spike and wave discharges in rats with absence epilepsy. *Journal of Physiology*, 509:449 – 456, 1998.

[23] Darrell A. Henze, Zsolt Borhegyi, Jozsef Csicsvari, Akira Mamiya, Kenneth D. Harris, and György Buzsáki. Intrecellular features predicted by extracellular recordings in the hippocampus in vivo. *Journal of Neurophysiology*, 84:390 – 400, July 2000.

[24] David J. Heeger and David Ress. What does fmri tell us about neuronal activity? *Nature Reviews Neuroscience*, 3:3822 – 3827, 2002.

[25] Neurotrack. http://www.neurotrack.com/prod01.htm.

[26] Anna Devor, István Ulbert, Andrew K. Dunn, Suresh N. Narayanan, Stephanie R. Jones, Mark L. Andermann, David A. Boas, and Anders M. Dale.

Coupling of the cortical hemodynamic response to cortical and thalamic neuronal activity. *Proceedings of the National Academy of Sciences*, 10:3822 – 3827, 2005.

[27] Peter Dayan and L.F. Abbott. *Theoretical Neuroscience - Computational and Mathematical Modelling of Neural Systems.* The MIT Press, 2001.

[28] Royal Society of Chemistry. http://www.chemsoc.org.

[29] L.F. Abbott, Edward Farhi, and Sam Gutmann. The path integral for dendritic trees. *Biological Cybernetics*, 66:49 – 60, 1991.

[30] M.L. Hines and N.T. Carnevale. The neuron simulation environment. *Neural Computation*, 9:1179 – 1209, 1997.

[31] Zachary F. Mainen and Terrence J. Sejnowski NEURON Simulation Code. http://www.cnl.salk.edu/ zach/patdemo.html.

[32] J.B. Ranck. Specific impedance of rabbit cerebral cortex. *Experimental Neurology*, 7:144 – 152, 1963.

[33] Zoltán Somogyvári, László Zalányi, István Ulbert, and Péter Érdi. Model-based source localisation of extracellular action potentials. *Journal of Neuroscience Methods*, 2005.

Resumé

Miguel Cavero was born in Lima, Peru on the 12th April 1978. He has lived most of his life in Africa, attending high school at Midlands Christian College in Gweru, Zimbabwe. Miguel completed his B.Sc. degree from the University of KwaZulu-Natal, in Pietermaritzburg, in 2001 majoring in physics and computer science. The following year he obtained an Honours degree in physics. Miguel will continue with a Ph.D. at the University on completion of his M.Sc. degree.

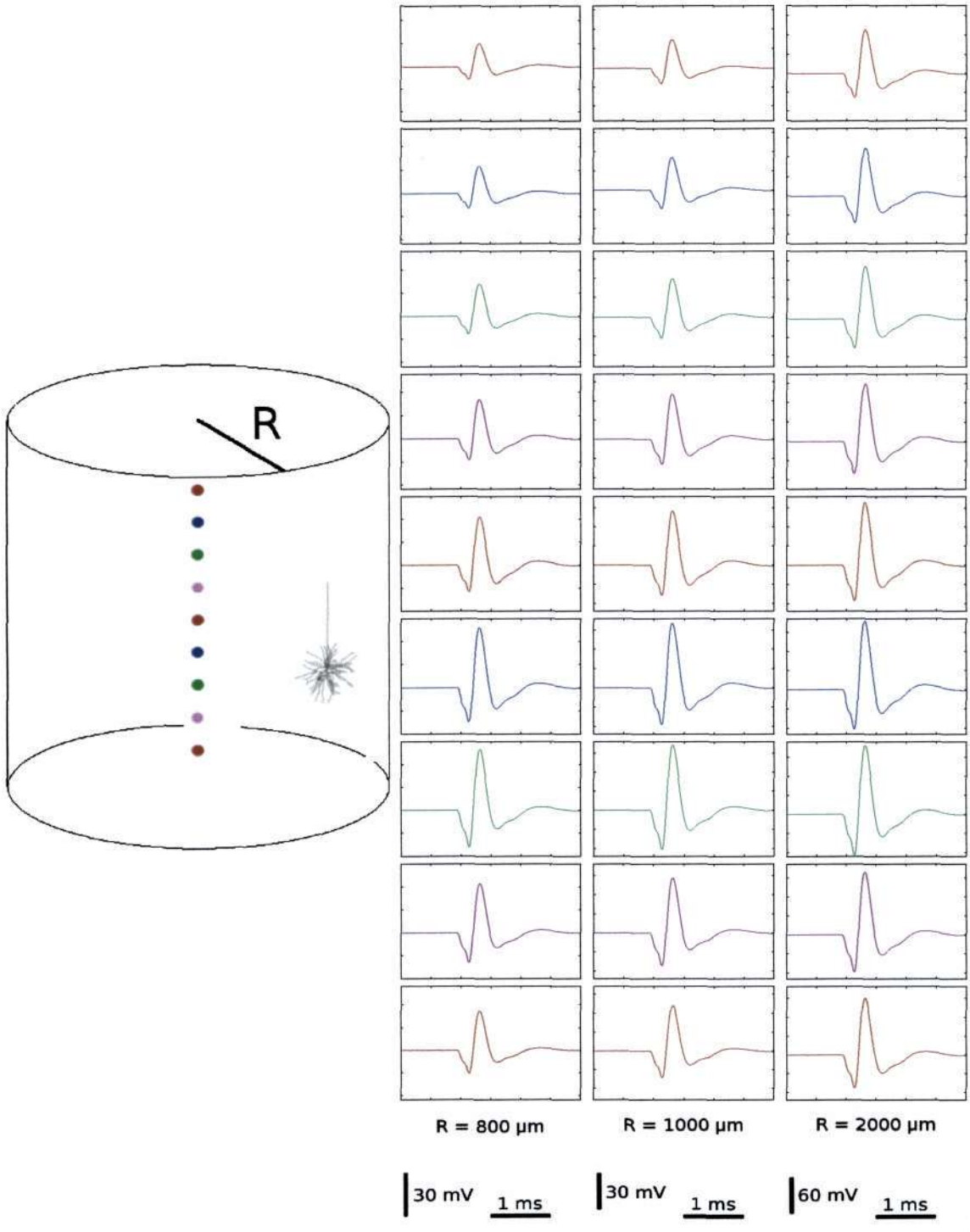


Figure 7.20:
High-pass filtered extracellular potential for a synchronous population of layer 4 stellates in a cortical cylinder for various radii (800, 1000 and 2000 μm).

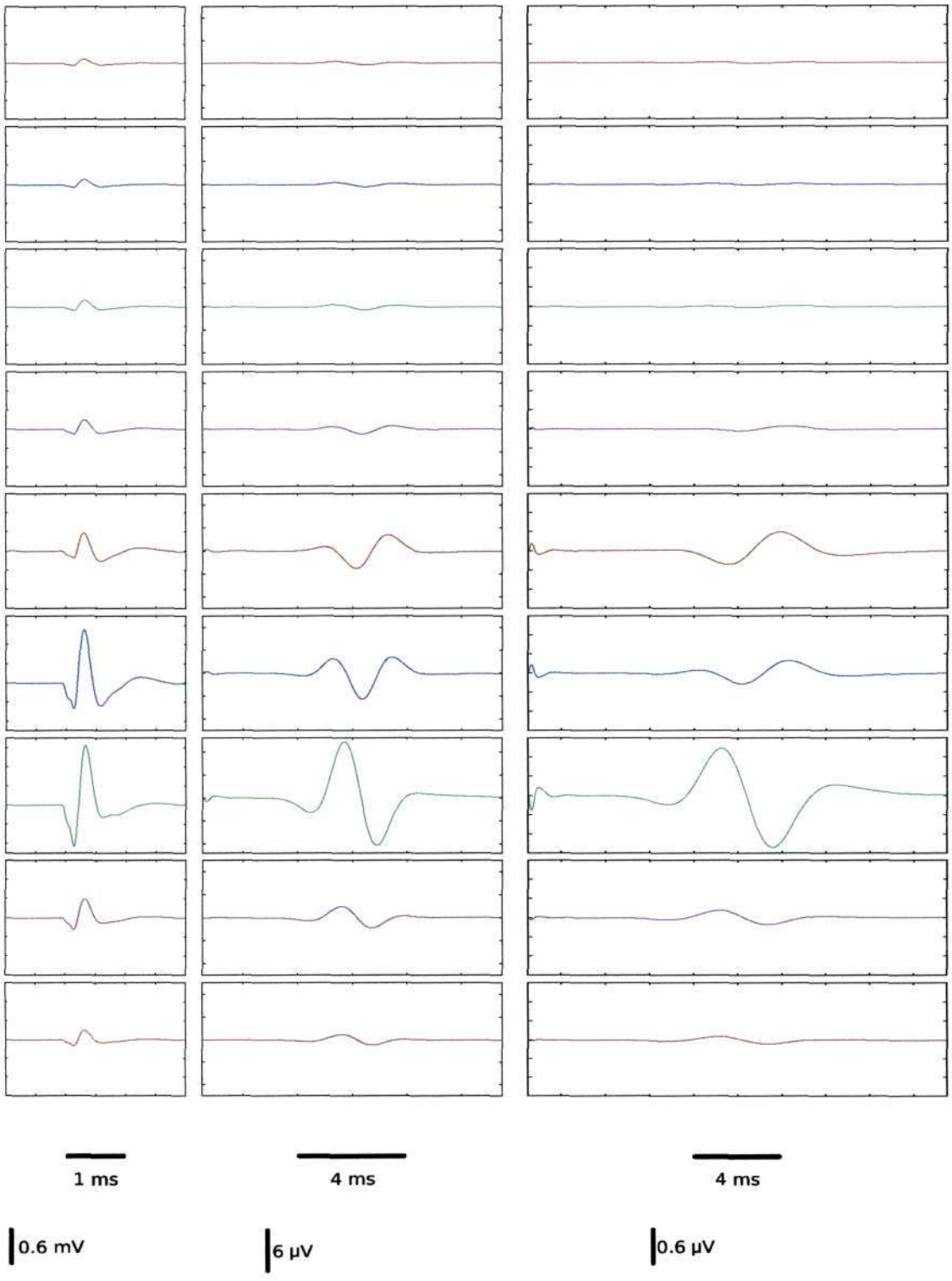


Figure 7.23:
High-pass filtered extracellular potential traces for an asynchronous population of layer 4 stellates in a cortical cylinder of radius 50 μm; σ_t has values 0, 1 and 2 ms.

Chapter 8

Conclusion

The goal of this M.Sc. thesis was to simulate extracellular potentials arising from action potentials fired by realistic neurons. The anatomically realistic neurons were taken from Mainen and Sejnowski [8] and were used to compute extracellular traces of three different types of cortical neurons, that is, a layer 5 pyramidal cell, a layer 3 pyramidal cell and a layer 4 stellate cell. A set of extracellular potential signatures of action potentials has been calculated:

- potentials from a single neuron in the vicinity of the soma,
- potentials from a single neuron further away from the soma along a vertical axis (mimicking laminar electrode recordings),
- potentials from a synchronously firing population of neurons in a cylindrical geometry, and
- potentials from an asynchronously firing population of neurons in a cylindrical geometry.