KU LEUVEN
FACULTEIT INGENIEURSWETENSCHAPPEN

Real-time detection of voltage patterns in the brain

Tomas Fiers

Thesis submitted for the degree of
Master of Science in
Biomedical Engineering

Thesis supervisors:
Prof. dr. ir. A. Bertrand
Prof. dr. F. Kloosterman

Assessors:
Prof. dr. ir. R. Puers
Dr. eng. J. Couto

Mentors:
Ir. J. Wouters
Dott. D. Ciliberti

Academic year 2018 – 2019

Contents

Abbreviations

BPTT Backpropagation through time. See section 9.3.

BPF Band-pass filter. See chapter 7.

BCI Brain-computer interface. See section 1.1.

CA1 "Cornu Ammonis", subregion 1. Area in the hippocampus where voltages are recorded from (see fig. 3.1).

CA3 "Cornu Ammonis", subregion 3. Area in the hippocampus (see fig. 3.1). CA3 sends many axons (called "Schafer collaterals") to CA1.

ch. Channel.

CSD Current source density. The rate at which the positive charge surplus of a small volume decreases, $-\partial\rho/\partial t$. By charge conservation, equal to the net rate $\nabla \cdot \mathbf{j}$ at which positive charges flow out of the small volume. See section 2.2.

FIR Finite impulse response. A linear filter whose output is a convolution of the input signal with some kernel.

GEVal Generalized eigenvalue. See section 8.2.

GEVec Generalized eigenvector. See section 8.2.

GRU Gated recurrent unit. See section 9.2.

IIR Infinite impulse response. A filter whose output at each timestep is a linear combination of both input samples and previous output samples.

IQR Interquartile range. A measure of the spread of a set of one-dimensional values, that is robust to outliers. Difference between the 75th and the 25th data percentile.

KDE Kernel density estimate.

LFP Local field potential. The extracellular electric potential (see chapter 2).

LTP Long-term potentiation. See section 4.5.

RMS Root-mean-square. $\sqrt{\langle x_t^2 \rangle}$ for a signal x_t.

RNN Recurrent neural network. See chapter 9.

SNR Signal-to-noise ratio. See section 8.2.

SOTA State of the art. The algorithm currently used for SWR detection, namely an online single channel band-pass filter.

STDP Spike-timing-dependent plasticity. See section 4.5.

SWR Sharp wave-ripple. The pattern in the LFP that we want to detect in real-time. See chapter 3.

VHC Ventral hippocampal commisure. See section 4.7.

Symbols

Notation

y	Scalars are denoted in lowercase italic.
$\mathbf{z}$	Vectors are denoted in lowercase boldface.
$\mathbf{A}$	Matrices are denoted in uppercase boldface.
$\langle\cdot\rangle$	Time-average of a signal.
$\lvert\cdot\rvert$	Number of elements in a set; Magnitude.
$\odot$	Elementwise multiplication. ("Hadamard product").
$\sigma(\cdot)$	Sigmoid 'squashing' function, $\mathbb{R} \to (0, 1)$. $\sigma(x) = \frac{1}{1+\exp(-x)}$.
$\tanh(\cdot)$	Hyperbolic tangent, $\mathbb{R} \to (-1, 1)$. $\tanh(x) = 2\ \sigma(x) - 1$.

Signals

$\mathbf{z}_t$	Digitized LFP sample at discrete time step t. $\mathbf{z}_t \in \mathbb{R}^C$, with C the number of channels (i.e. the number of electrodes simultaneously recorded from). Input to an SWR detection algorithm.
o_t	Output signal of an SWR detection algorithm, $\in \mathbb{R}$.
n_t	'Envelope'. Transformation of o_t, so that it is constrained to $\mathbb{R}^+$. Should be high when the corresponding input sample $\mathbf{z}_t$ is part of an SWR segment, and low when it is not. $n_t = \lvert o_t \rvert$ for online linear filters; $n_t = \sigma(o_t)$ for the RNN's of chapter 9.
y_t	Binary target signal, used when training data-driven SWR detection algorithms. We define $y_t = 1$ when the corresponding input sample $\mathbf{z}_t$ is part of an SWR segment, and $y_t = 0$ when it is not.

Measures & parameters

T	Detection threshold applied to the envelope n_t. $T \in [\min n_t,\ \max n_t]$. Each threshold T yields a different P-value, R-value, F_1-value, etc.
P	Precision. Also known as positive predictive value. The fraction of correct detections, out of all detections.
R	Recall. Also known as sensitivity, hit rate, or true positive rate. The fraction of detected reference SWR segments, out of all reference SWR segments.
F_β	F-score: weighted harmonic mean of recall and precision. $F_\beta = \frac{(1+\beta^2)PR}{\beta^2 P+R}$. Measures detection performance "for a user who attaches β times as much importance to recall as to precision." [1]
F_1	F-score where recall and precision are weighted equally.
f_s	Sampling frequency of a signal.

Chapter 1

Introduction

1.1 Closed-loop neuroscience

This thesis discusses signal-processing software for a brain-computer interface (BCI).[1] Most BCI research is focused on restoring or partially replacing a damaged nervous system after a stroke or a traumatic injury [2]. The BCI discussed in this thesis however is used for fundamental brain research into memory & learning, using laboratory animals such as rats or macaques.

On a high level, the discussed BCI works as follows (see also fig. 1.1): an electrode is surgically implanted into the animal's brain, where it records the electric field potential. (Chapter 2 discusses the origin of this signal). During an experiment, the BCI continuously tests whether it can detect a particular pattern in the recording. (This pattern, called the *sharp wave-ripple*, is discussed in chapter 3). Each detection of the pattern triggers a feedback action. This is often a current injection that temporarily disables the specific brain area where the pattern is recorded from.

This experimental technique enables scientists to test their hypotheses about sharp wave-ripples (SWR's), the affected brain area, and in general, about brain mechanisms such as learning & memory. (This is discussed further in chapter 4).

1.2 Problem statement

This thesis investigates whether the current sharp wave-ripple detection algorithm can be improved upon. That is: can we find an online algorithm that detects SWR's with less latency than the current online algorithm, while being at least equally sensitive and precise?

(In chapters 5 and 6, we discuss these performance criteria in more detail).

[1] A BCI is a device that reads out brain activity, which is used to control some automated action. The brain receives feedback about the performed action, either through normal sensory input, or through direct neural stimulation, thereby forming a closed loop.

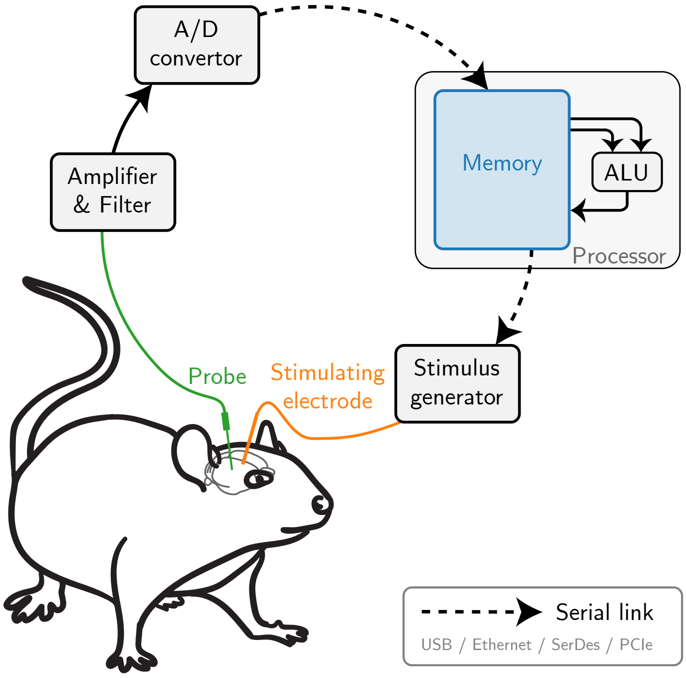


Figure 1.1: **A closed-loop brain-computer interface in neuroscience**. Arrows show the direction of data flow. ALU: arithmetic logic unit.

1.3 Motivation

This question is motivated by two factors. First, current online detection algorithms add a considerable latency: the SWR event is often detected after a third of the event has already passed (see figs. 7.4 and 7.5 from chapter 7, where existing online SWR detection algorithms are evaluated). This detection latency might compromise experimental power.

Second, most current detection algorithms make use of only a single voltage signal. Recently however, more and more recordings are being made with *neural probes*, which simultaneously record the voltage at multiple spatially distributed points [3], [4]. We investigate whether we can use these spatially distributed multichannel recordings to improve online SWR detection.

In chapters 8 and 9, two existing signal detection techniques are adapted to create new online SWR detection algorithms. These algorithms are analyzed, their parameters are explored, and their performance in online SWR detection is compared with the state-of-the-art. Chapter 10 summarizes the findings, gives recommendations, and discusses limitations of this thesis, as well as directions for further work.

Chapter 2

The local field potential

This thesis discusses software for processing intracranial voltage recordings. This chapter describes the nature of such voltage signals.

At every location in the brain, the electric field vector $\mathbf{E}$ (in units of N/C or V/m) points in the direction that a positive charge would be pushed towards, if it was placed at that location. The electric potential ϕ is defined[1] such that:

$$\mathbf{E} = -\boldsymbol{\nabla}\phi, \tag{2.1}$$

i.e. such that the electric field points from locations of high potential to locations of lower potential. The electric field potential ϕ is a scalar field, in units of V. When it is measured extracellularly, neuroscientists refer to it as the *local field potential* (LFP) – especially when only frequencies below about 500 Hz are considered.

We now discuss two models used to estimate the LFP ϕ. Both models calculate ϕ based on the mobile charge density ρ at each location in the brain and throughout time. (ρ is the net positive surplus of mobile charges, in units of C/m^3. Biological tissue contains a high number of such mobile charges. These are mostly ions, with Na^+, Cl^-, and K^+ the most abundant ones [7]).

Both models consist of a single divergence equation (namely eqs. (2.2) and (2.7)), which, together with eq. (2.1) and an assumption of uniform and isotropic brain tissue, leads in both cases to a simple closed form equation to calculate the LFP ϕ (namely eqs. (2.3) and (2.8)).

2.1 Model 1 (Gauss's law)

The first model is a direct translation of the first of Maxwell's equations (Gauss's law), at macroscopic scale:[2]

$$\boldsymbol{\nabla} \cdot \varepsilon\mathbf{E} = \rho, \tag{2.2}$$

[1] The electric field potential ϕ is only defined when the electric and magnetic fields do not vary too quickly over time, i.e. when $\partial\mathbf{E}/\partial t \approx 0$ and $\partial\mathbf{B}/\partial t \approx 0$. It then directly follows from Maxwell's equations that both fields become decoupled, and that the electric field becomes irrotational ($\boldsymbol{\nabla} \times \mathbf{E} = 0$), so that an electric potential may be defined as in eq. (2.1). This so called *quasi-static* assumption is met in electrophysiological conditions [5], [6].

where ε is the local tissue permittivity, in units of $\mathrm{F/m}$ and generally a 3×3 matrix.

When we assume the permittivity ε to be uniform and isotropic throughout the tissue (i.e. ε is a constant scalar)[3], eq. (2.2) has the following solution for the LFP ϕ [8]:

$$\phi = \frac{1}{4\pi\varepsilon} \int_V \frac{\rho \, \mathrm{d}V}{r}, \tag{2.3}$$

where we summate over the entire brain volume V, with $\rho \, \mathrm{d}V$ the surplus mobile charge in a small volume $\mathrm{d}V$ of tissue (in Coulomb), and r the distance of this small volume to the point where ϕ is calculated.

In other words, positive charge surplusses increase the nearby LFP, negative charge surplusses decrease it, and their effects summate linearly, weighted inverse-proportionally by distance. This model explains for example the resting "membrane potential" of neurons, where ϕ is lower inside the cell than outside: neurons at rest contain an excess of negative charges [12].

Although this model is arguably the most physically correct, it is often difficult to apply in practice: charge distributions ρ are already quite complex in even the simplest of electronic circuits [13, chapter 8]. For the highly entangled topologies of brain tissue (see fig. C.3), estimating charge distributions is intractable.

The next model is more useful in practice, as it calculates the LFP based not on charges, but on currents. Currents (and specifically transmembrane currents) are the bread and butter of electrophysiology [14].

2.2 Model 2 (Current source density analysis)

This is the "standard model of electric potentials in biological tissue" [15]. Unlike the previous model, it is mostly empirical [15]. In fact, the assumptions on which it rests are strictly incompatible with eqs. (2.2) and (2.3), and lead to a paradox when considered jointly. Bedard and Destexhe explore this paradox [15], and propose a more general formulation of CSD analysis that 'solves' the paradox. In the remainder of this chapter however, we will present the 'classic' CSD model, as it is presented in most electrophysiology texts [5], [6], [16]–[19].

We first define the so called *current source density*, I, as the rate at which the net charge surplus inside a small volume decreases (units $\mathrm{A/m^3}$):

$$I = -\frac{\partial \rho}{\partial t} \tag{2.4}$$

[2] Gauss's law in its pure form ($\boldsymbol{\nabla} \cdot \mathbf{E} = \rho_{\text{total}}/\varepsilon_0$) considers both mobile and "bound" charges: $\rho_{\text{total}} = \rho_{\text{free}} + \rho_{\text{bound}}$. In eq. (2.2), we only explicitly consider mobile charges $\rho = \rho_{\text{free}}$, while bound charges are subsumed in the tissue permittivity ε. This simplification is allowed if we assume that "tissue polarisation is proportional to electric field strength" [8]. This is a common assumption in physics, and is largely valid for brain tissue in normal conditions [5].

[3] The uniform and isotropic permitivitty assumption is questionable. Roughly, brain tissue consists of a dense alternation of two types of tissue (see fig. C.3): the seawater-like fluid inside and in between cells, and the fatty membranes around cells, organelles, and vesicles. The former has a permittivity ε of about 15 times larger than the latter [9]–[11]. Additionally, the strongly non-random organization of some brain regions may challenge the isotropy assumption. Nevertheless, this uniform and isotropic permittivity assumption is often made [5], [6].

Charge conservation (which is a direct consequence of Maxwell's equations) dictates that

$$\boldsymbol{\nabla} \cdot \mathbf{j} = I, \tag{2.5}$$

where $\mathbf{j}$ is the current density (units of $\mathrm{A/m^2}$, and $\mathbf{j} = \rho\mathbf{v}$ with $\mathbf{v}$ the local velocity of mobile charges). In other words, the decrease in net positive charges in a small volume (I) equals the net rate at which positive charges flow out of this volume ($\boldsymbol{\nabla} \cdot \mathbf{j}$).

Next (and this is the assumption challenged in [15]), we assume that the extracellular medium is *resistive* or *ohmic*:

$$\mathbf{j} = \sigma\mathbf{E}, \tag{2.6}$$

where σ is the local tissue conductivity, in units of S/m and generally a 3×3 matrix. In other words, current is assumed to be linearly related to the electric field. (This needn't be the case in reality: the electric field dictates the acceleration of mobile charges, while the current density describes their velocity. These are instantaneously indepedent.)

Combining charge conservation and the ohmic assumption leads to the governing equation for CSD analysis:

$$\boldsymbol{\nabla} \cdot \sigma\mathbf{E} = I. \tag{2.7}$$

Note the similarity with eq. (2.2). And analogously as in eqs. (2.2) and (2.3), when we assume the conductivity σ to be uniform and isotropic throughout the tissue (i.e. σ is a constant scalar)[4], eq. (2.7) has the following solution for the LFP ϕ [6]:

$$\phi = \frac{1}{4\pi\sigma} \int_V \frac{I\,\mathrm{d}V}{r}. \tag{2.8}$$

In other words, so called "current sources" ($I > 0$) increase the nearby LFP, current sinks ($I < 0$) decrease it, and their effects again summate linearly, weighted inverse-proportionally by distance.

2.3 Transmembrane currents

Active transmembrane currents are the main physical mechanism of signal propagation in the nervous system.[5] At synapses, excitatory neurons cause a brief inward current in their target neuron, while inhibitory neurons cause a brief outward current. Along axons, action potentials ("spikes") propagate by a cascade of inward currents.

Each of these active currents leads to corresponding passive currents in other parts of the membrane, forming closed current loops. An excitatory, inward current at a synapse corresponds to outward currents at distal parts of the neuron (most notably at the cell body). The inward, active currents of a spike are surrounded by outward, passive currents.

[4] Again, a questionable assumption: (extra)cellular plasma has a conductivity σ of about 10^6 times higher than the lipid bilayer membranes in between [5], [9]–[11], [20].

[5] With 'active', we mean here that the current is caused by the opening of ion channels that are located at the same position in the membrane as the current.

2.4 Forward CSD model of the LFP

In CSD analysis, both active and passive transmembrane currents are used to estimate the LFP, by making use of the following 'trick': the existence of the inside of the neuron is ignored. An inward current for example is then modelled as a current sink $I_m < 0$ in the extracellular space, i.e. an accumulation of mobile charges. In reality, charges do not quite accumulate, but rather dissipate in the neuron. Or, equivalently, a proportional current source $I'_m = -I_m$ exists inside the neuron, at virtually the same location as I_m, cancelling it out. In CSD analysis however, only the current sink I_m (and not the current source I'_m) would be used for calculating the LFP ϕ.

The justification for ignoring the inside of the neuron is often as follows (paraphrasing from [16], a foundational CSD theory paper): "The extracellular space is independent of the intracellular space, because its boundaries (the cell membranes) have a very high resistance compared to the extracellular space." However, electric insulators (high resistance areas) do not shield the electric field, nor the electric field potential ϕ. [6] [7]

Nonetheless, in the so called 'forward' CSD model (estimating the LFP from transmembrane currents), eq. (2.8) is used to calculate the LFP based on 'unpaired' transmembrane current source densities I_m. From eq. (2.8), an inward current then corresponds to a drop in the nearby LFP, and an outward current to an increase. This technique is applied in section 3.2 to explain the physical basis of the sharp wave-ripple motif. Another application of this technique is e.g. `LFPy`, a software package used to numerically simulate the LFP [18].

Notwithstanding the theoretical problems outlined here and in [15], CSD analysis seems to provide relatively accurate results in practice [15], [19]. As mentioned earlier, Bedard and Destexhe [15] give a derivation of a CSD-like model that does not rest upon the 'classic' derivation as presented here, instead concluding: "[..] the results obtained with the classic CSD analysis are perfectly consistent with *ionic diffusion* because diffusion gives a source term which is very close to the phenomenological model of current source density introduced by Pitts and Mitzdorf, but in a manner consistent with Maxwell-Gauss law." (Pitts [21] and Mitzdorf [16] refers to the classical CSD model).

[6] Another reason the inside of neurons is ignored might possibly be notation: at synapses, there are real currents, often denoted by I; but there are no real current sources or sinks, also often denoted by I. This may cause confusion.

[7] A final reason to not only consider the extracellular space, is that there is not that much of it: only between 10% and 20% of hippocampus volume is extracellular space **Sykova1997a**. See also fig. C.3.

Chapter 3

Sharp wave-ripples

The sharp wave-ripple (SWR) is a well-known motif of the local field potential. Section 3.1 shows where SWR's are found in the brain and what they look like, while section 3.2 briefly describes their physiology and how they are generated. Section 3.3 shows how SWR's can be recorded, and describes the data set analyzed in this thesis. The next chapter explains why SWR's are worth investigating in the first place.

3.1 Anatomy of the SWR

Sharp wave-ripples are a spatiotemporal pattern of the LFP of the *hippocampus*. Figure 3.1 (left) shows the location of the hippocampus in the human and the rat brain.[1,2] (The data analyzed in this thesis was recorded from a rat (see section 3.3) which is why we focus on this animal).

More specifically, SWR's are observed in *area CA1* of the hippocampus (see the right-hand side of fig. 3.1). This brain area is highly organized: most neuron cell bodies are concentrated in a thin layer, the *pyramidal cell layer*. (This is the darkly colored C-shape in the stained brain slices). The dendrites of these neurons are located in the layers above and below this pyramidal cell layer, more specifically in the *stratum oriens* above, and the *stratum radiatum* and *stratum lacunosum-moleculare* below. (These layers have a lighter color in the stained brain slices).

The "ripple" part of a sharp wave-ripple is observed in and around the pyramidal cell layer of CA1, and consists of a $\pm$ 40 ms long oscillation of the extracellular electric field potential, at a frequency of 100 – 200 Hz (period of 5 – 10 ms). See the time series in fig. 3.2 for two examples. See also table A.1 for a comparison of reported ripple bands in the literature.

[1] Strictly, a brain has two hippocampi, one in each hemisphere, and mirrored with respect to each other. We will quite inconsistently refer to them in both singular and plural.

[2] In the rat brain, the hippocampi are the size of two cooked grains of rice, and are shaped as two upright bananas joined at their ends. In the human brain, the hippocampi are about 7 cm long, lying lengthwise at the base of the brain, at approximately eye height [22], [23]. They are shaped like two seahorses (which is the namesake of the hippocampus – Greek for seahorse), or like a ram's horns (which is the namesake of the CA regions in the hippocampus – these stand for Cornu Ammonis, or Ammon's Horn, where Ammon is an Egyptian deity sometimes taking the form of a ram.)

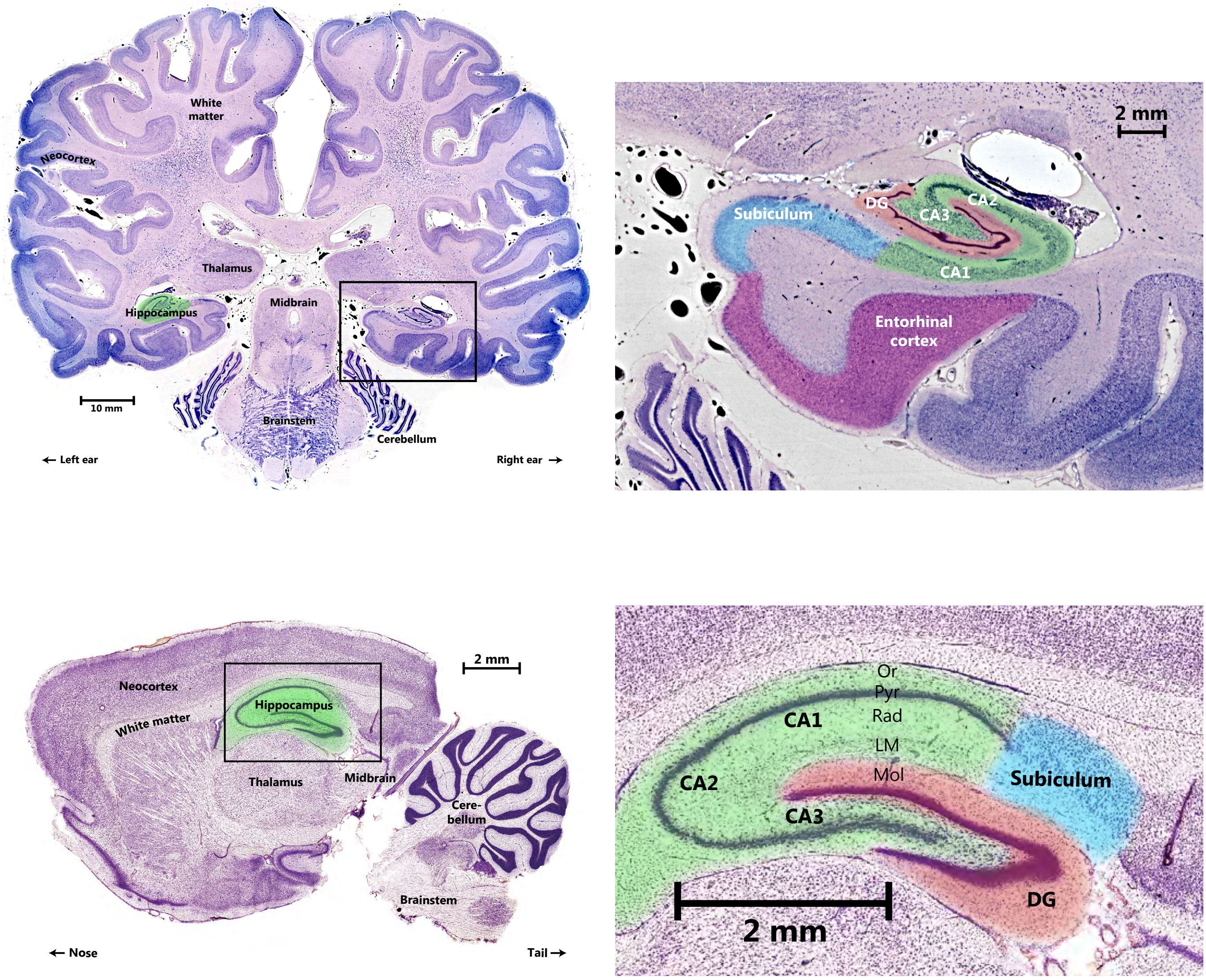


Figure 3.1: **Location of relevant brain structures in the human and the rat brain**. Coronal section of a human brain (top) and sagittal section of a rat brain (bottom). Both slices are Nissl-stained, which means that each purple dot represents a cell body. (This can be more clearly seen in fig. C.1, which is a further zoom-in of the top-right panel). Lighter areas indicate regions that consist mostly of axons and dendrites. Black boxes mark location of the zoomed-in micrographs at the right hand side. Note the correspondence in shape and constituents between the human and the rat hippocampus. Note also the grossly enlarged neocortex in the human brain relative to the size of other brain structures, when compared with the rat brain. DG: dentate gyrus. CA1/2/3: Cornu Amonis region 1/2/3. Or: stratum oriens. Pyr: pyramidal cell layer. Rad: stratum radiatum. LM: stratum lacunosum-moleculare. Mol: molecular layer of the dentate gyrus. Human slice from [24] (30 mm posterior to the anterior commisure), with [23] as a guide for labelling. Rat slice from [25] (plate 175 – 3.4 mm lateral).

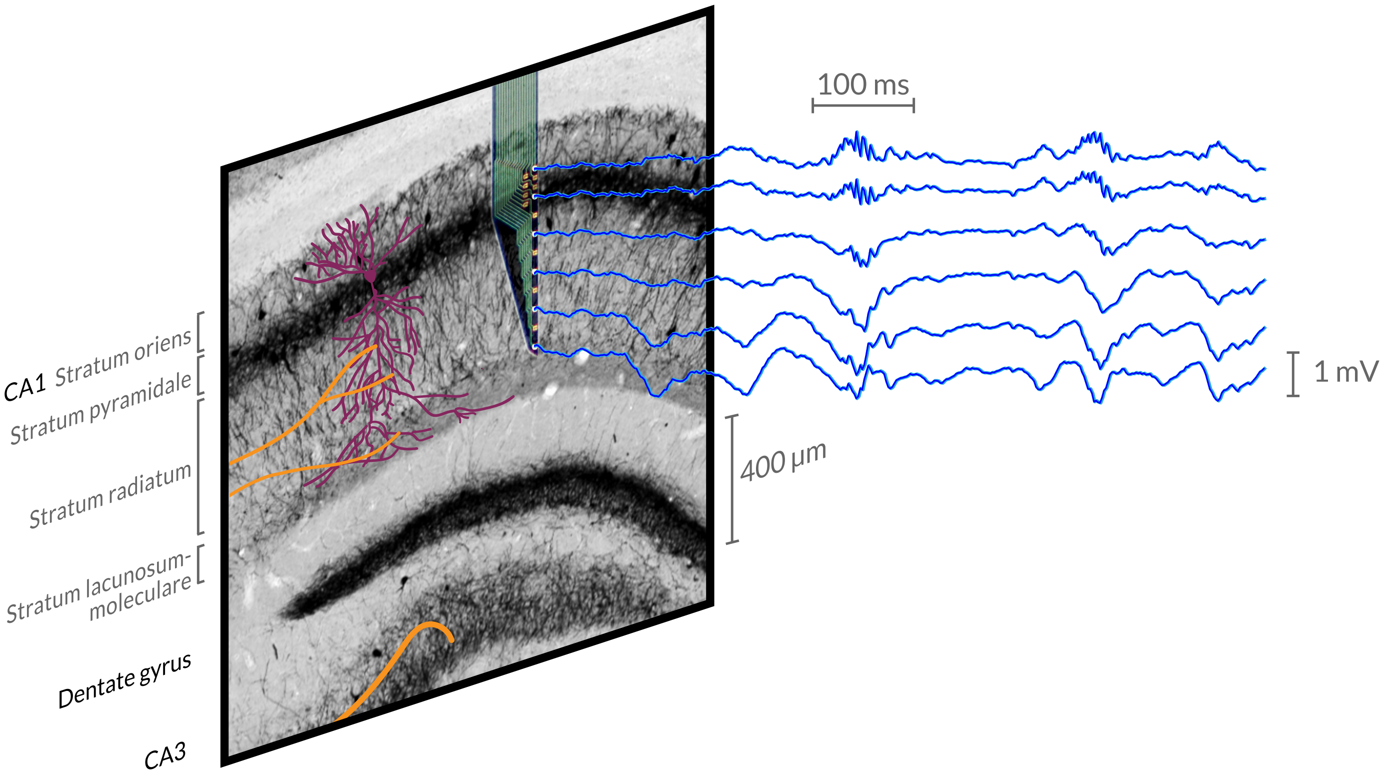


Figure 3.2: **Recording sharp wave-ripples**. Composite image showing estimated location of the probe tip in area CA1 of the rat hippocampus. Indicated on the left are hippocampal regions (black text) and layers of region CA1 (grey text). Overlaid on the micrograph of a slice of rat hippocampus (black and white picture) are a cartoon of a possible CA1 pyramidal neuron (purple), and an axon (orange) leaving a neuron in CA3 to synapse onto the apical dendrites of the CA1 neuron. Also overlaid is a photograph of the "L-style" probe from [3], source of the data analyzed in this thesis. Gold squares are its 16 electrodes. Time series (right) show example voltage recordings from five of these electrodes (which are marked with a white dot). The displayed data shows two sharp wave-ripple events. Note the ripples in and around the pyramidal cell layer ("stratum pyramidale"). Note also the phase reversal of the sharp wave: from strongly negative in the stratum radiatum, to near zero power in the middle of the pyramidal cell layer, to positive in the stratum oriens. (Coronal slice of rat hippocampus from [26]. Location: 2.98 mm posterior to bregma. Slice stained for parvalbumin, which highlights interneuron somata and neurites. Anterior-posterior and medial-lateral location of the probe were estimated based on the location of probe "L2" given in [3]. Depth of probe estimated from the characteristics of the time-series recorded from each electrode. CA1 pyramidal cell morphology from [27]).

The "sharp wave" part is observed simultaneously, in all layers of CA1, but most markedly in the stratum radiatum. In the stratum radiatum and the stratum lacunosum-moleculare, the sharp wave consists of a strong potential drop, followed by a return to the baseline potential. Together these form a downwards peak of $\pm$ 40 ms wide. Throughout the pyramidal cell layer, the polarity of this peak gradually reverses. The sharp wave has near zero power in the middle of the pyramidal cell layer, and consists of an upwards peak near the top of the pyramidal cell layer and in the stratum oriens above.

3.2 Physiology of the SWR

Sharp wave-ripples are observed during deep sleep[3] and during certain awake states, such as drinking, eating, grooming, and staying still. They then occur at a rate of between one every ten seconds, to twice per second [28]. In contrast, SWR's almost never occur during exploratory behaviours such as walking, running, sniffing, or rearing[4]. (During these behaviours, the hippocampal LFP is dominated by a $\pm$ 8 Hz oscillation – the so called θ-state. See fig. 4.1 for an example). The aggregated firing rate of neurons in the hippocampus is highest during SWR's. SWR's have so far been observed in all investigated mammalian species, including humans [29].

SWR's are generated as follows [28], [29]. Large groups of excitatory neurons in area CA3 of the hippocampus simultaneously activate; i.e. send an action potential along their axons. These axons (called "Schaffer collaterals") end up in the stratum radiatum of area CA1. There, they stimulate the dendrites of CA1 neurons: at each synapse, a small current enters the CA1 neuron. In accordance with the LFP model of section 2.4, these small, simultaneous, inward currents result in a drop in the LFP. This is the sharp wave. (See fig. 3.2 for SWR examples, and the corresponding anatomy).

As described in section 2.3, the active, inward currents in the stratum radiatum give rise to passive, outward currents in other parts of the CA1 neurons. These outward currents occur mostly in the cell bodies, which are concentrated in the pyramidal cell layer. This is why the 'polarity' of the sharp wave reverses when going from the stratum radiatum, through the pyramidal cell layer, up to the stratum oriens: the outward currents result in an increase in the LFP. Around the bottom-middle of the pyramidal cell layer, the effects of the inward and the outward currents cancel out, resulting in an absence of the sharp wave. Nearer the stratum oriens, the outward currents dominate, resulting in the upwards sharp wave.

The ripple oscillation is generated through delayed inhibition. The so called CA1 *pyramidal neurons*, that are activated in the sharp wave, synapse onto inhibitory *interneurons*, which are also present in CA1. The axons of these interneurons synapse back onto the pyramidal cell bodies. This recurrent connection pattern enables the ripple oscillation: sharp wave-activated pyramidal cells activate interneurons, which silence the pyramidal cells. This deactivates the interneurons, allowing the pyramidal cells to activate again. They then activate the interneurons, and so on. The intrinsic frequency of this system

[4]standing up on the hind legs

[4]Namely so called *slow-wave sleep* (SWS). The LFP then has higher power in the δ-band, at $\pm$ 1 Hz (i.e. the slow waves). SWS is contrasted with rapid eye movement (REM) sleep, where the LFP is dominated by θ waves at $\pm$ 8 Hz – The same θ waves as during exploratory behaviours.

is the ripple frequency, 100 – 200 Hz. The LFP is low when the pyramidal cells are active (spikes and excited synapses correspond to inward currents). It is high when the interneurons are active (inhibited synapses correspond to outward currents).

3.3 Recording SWR's

Sharp wave-ripples can be recorded by making a small opening in the skull and inserting an electrode in the brain, ending in area CA1. All analyses in this thesis were carried out on such a recording, made in 2014 by Michon et al., and described in [3].

The data consists of a 34-minute long, multi-channel time-series of the extracellular electric field potential in area CA1 of a rat, made with a flexible, silicon-based, multi-electrode probe. (See fig. C.2 for photos of the full probe and the implant). Specifically, we used data from the probe labelled "L2" in [3]. Figure 3.2 shows the layout of the electrodes on this probe, and the approximate location of the probe with respect to the different structures in the hippocampus. The probe has 16 electrodes, 8 of which are placed in a linear array (spaced 50 µm apart), and 8 placed in a cluster (of 100 µm $\times$ 40 µm, with an inter-electrode distance of $\pm$ 25 µm). The electrodes cover a total depth of 500 µm and span all layers of area CA1. The recording was made while the rat was at rest. This makes it likely that sharp wave-ripples are observed [29].

We detected SWR's in this recording according to the offline procedure described in chapter 5. Figure 3.3 shows where the resulting SWR's occur along the recording, as well as the distribution of SWR durations. The last 40% of the recording (i.e. the last 14 minutes) were used to evaluate online detection algorithms (as in chapter 6). The other 60% were used to train data-driven algorithms (chapters 8 and 9).[5]

[5]The 60-40 division was chosen arbitrarily – under the constraint that there are sufficient amounts of both training and test data.

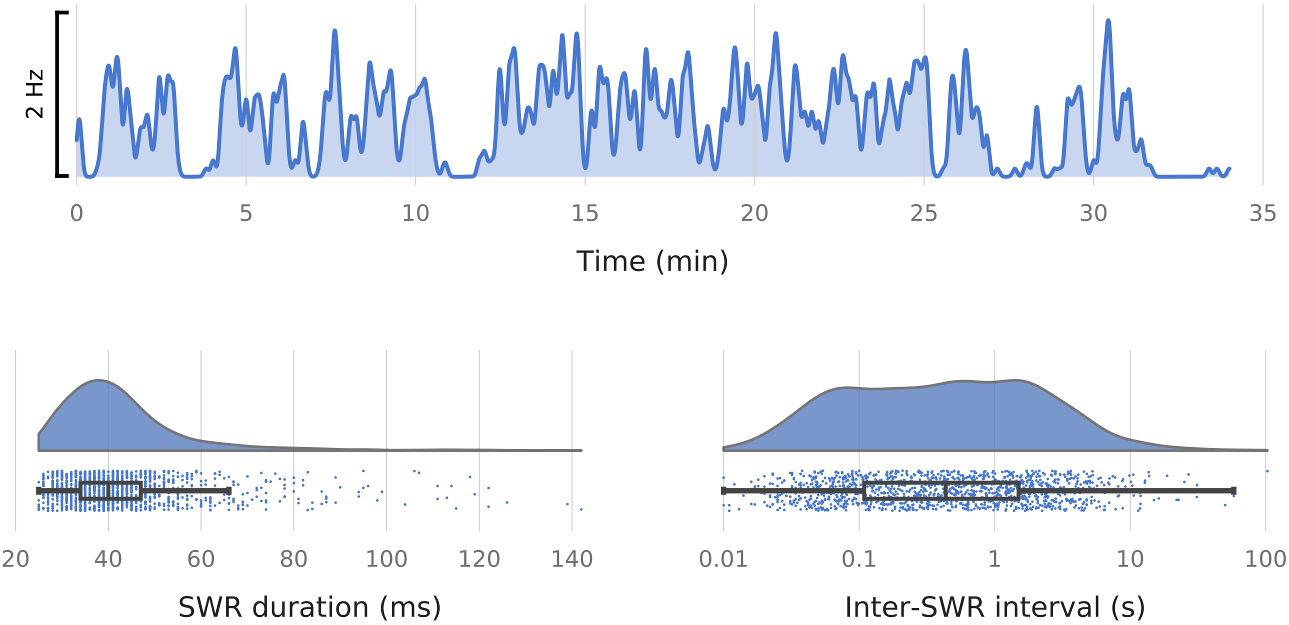


Figure 3.3: **Statistics of the recorded SWR's**. SWR locations and extents calculated offline according to the procedure described in chapter 5. *Top*: SWR frequency along the duration of the recording. Kernel density estimate with Gaussian kernel, $\sigma = 4$ seconds. *Bottom*: Distribution of SWR duration, and duration between SWR's. (Boxplot whiskers indicate 1.5 times the interquartile range).

Chapter 4

Scientific importance of SWR's

Sharp wave-ripples play an important role in learning and memory.[1] The evidence for this has been accumulating over the years, and has mostly been associative / correlational. The kind of closed-loop experiments discussed in this thesis however have recently been providing more direct evidence.

Most of this chapter is based on the review of Girardeau and Zugaro [28], and to a lesser extent on [29]–[32].

4.1 Overview of evidence

We first summarize what is known about the importance of sharp wave-ripples. In the following sections, we expand on each of these points.

1. The hippocampus – where SWR's are recorded from – is necessary for consolidating memories.
2. Memory consolidation happens mostly during sleep. SWR's also occur mostly during sleep.
3. The high firing frequency of neurons during SWR's is ideal for strengthening synapses.
4. Awake neural firing patterns are later replayed, in both the hippocampus and the neocortex. These replays occur mostly during SWR's.
5. Forcibly silencing the hippocampus during SWR's deteriorates performance on a newly learned task. Silencing the hippocampus outside SWR's has no such effect.

[1] In this section, when we refer to "SWR's", we mean "SWR's, or the mechanism that generates them". – The SWR pattern in the LFP could be regarded as an epiphenomenon. Additionally, when we refer to "memory", we more specifically mean so called *declarative* memory. This excludes conditioned and instinctive emotional responses, "muscle memory" (habits and motor skills), and habituation or sensitization of the senses.

The last two points have only been convincingly demonstrated in rats, but are likely true for other mammals as well.

4.2 Memory consolidation

SWR's are thought to be pertinent mainly in the process of *memory consolidation*, where some short-term memories are stabilized into long-term memories. Single-cell recordings in primates have hinted that short-term memories exist as positive feedback loops of firing neurons in the neocortex ("reverberations", or persistent activity). Long-term memories on the other hand are likely to exist as strenghtened or newly built physical connections between neurons in the neocortex. [32]–[34]

The case of Henry Molaison (known as "patient H.M." up until his death in 2008) illustrates the importance of memory consolidation, and of the role of the hippocampus in this process. In an attempt to cure his epilepsy, most of Molaison's hippocampi and the adjacent entorhinal cortices were surgically removed, leaving the rest of the neocortex intact. Before the surgery, Molaison had no memory problems. After the surgery however – although his epileptic seizures decreased – he could not form new long-term memories.[2] Curiously though, Molaison still had both intact short-term memory (remembering new information for up to minutes after), and largely intact long-term memory for events that happened before the surgery. Similar symptoms have been observed in other clinical cases and in lesioned animal experiments. These symptoms are consistent with a model of memory consolidation where both short-term and long-term memories are stored in and retrieved by the neocortex, but where the hippocampus is needed to convert short-term into new long-term memories. [32], [33]

This memory consolidation is thought to take place mostly during sleep. In this so called *two-stage model* of memory consolidation, new information is first input to the hippocampus during the awake state. In the second stage, during subsequent sleep, this information is consolidated 'offline' to the neocortex, by the hippocampus. As mentioned in section 3.2, SWR's occur mostly during sleep.

4.3 Hippocampo-cortical connections

At a more detailed anatomic and physiological level, there is indeed quite some evidence for this model of memory consolidation. The main input to the hippocampus is the entorhinal cortex, which relays deeply processed sensory information from the rest of the neocortex. In turn, the hippocampus has many output projections to the neocortex (in large part to the prefrontal and the anterior cingulate cortices, where both short- and long-term memories are thought to mostly reside).

It has been discovered in rats that during SWR's, neocortical neurons indeed tend to fire shortly after hippocampal neurons [35]. Additionally, awake firing patterns in the neocortex have been found to be *replayed* later, specifically during SWR's [36]. (More about replay in section 4.4. That section discusses replay in the hippocampus itself. Hippocampal replay is more established in the literature than neocortical replay).

[2] For example, the doctors working with Molaison had to re-introduce themselves on every occasion they saw him [34].

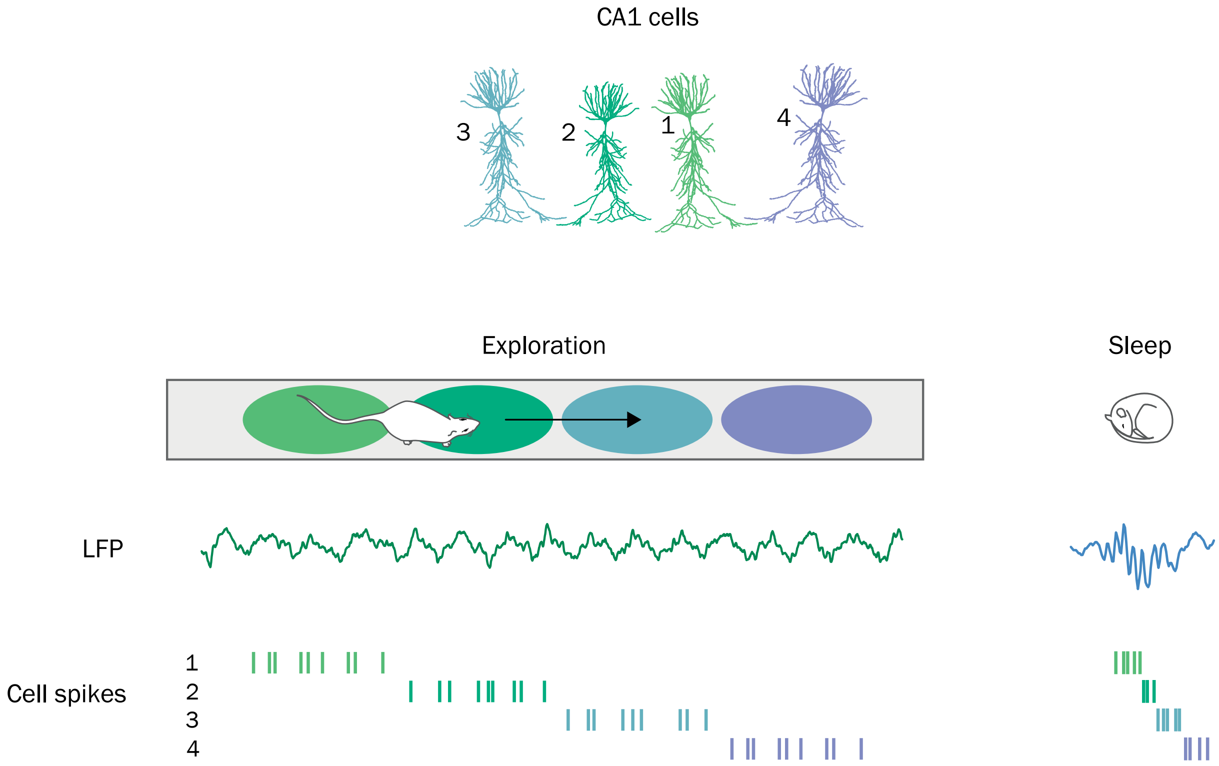


Figure 4.1: **Replay**. *Adapted from Girardeau and Zugaro [28].* See text for details.

Finally, during alert wakefulness, the hippocampus contains higher levels of neuromodulators that enhance the influence of external inputs relative to internal activity. During sleep on the other hand, the hippocampus contains lower concentrations of such neuromodulators, and there is simply less sensory input. This favors endogenous activity; the highly recurrent connectivity of the hippocampus is then thought to spontaneously generate SWR's.

4.4 Replay

Around the turn of the millenium, a memory-related discovery was made in the hippocampus at a quite detailed and mechanistic level, yielding new evidence for the importance of sharp wave-ripples. This phenomenon, called *replay*, was discovered through simultaneous recordings of the spiking activity of multiple individual neurons in the hippocampus.[3] See fig. 4.1 for a schematic example of replay.

When a rat is placed in a maze, there is a good chance that a given hippocampal neuron is a so called *place cell*, meaning that it only fires around a certain location in the maze.[4].

[3]Such recordings are classically made with tetrodes – four tightly wound electrodes. More recently, multi-electrode neural probes are also being used. Each electrode picks up the spikes from multiple nearby neurons; and each neuron's spikes are visible on multiple electrodes. Each neuron's spikes leave a distinct signature on the multi-electrode recordings. The process of finding these signatures and using them to group recorded spikes by their generating neuron is called *spike sorting*.

When the rat walks around the maze, a set of place cells will therefore fire in sequence, reflecting the path that the rat traversed.

When the rat later rests, these place cell firing sequences occur again, even though the rat stays in the same place. These sequence reactivations are called *replay*. They are compressed in time versus the original sequences, and, importantly, occur mostly during sharp wave-ripples. This time-scale means that reactivated neurons (and their downstream projections) are likely to become more tightly connected, as is explained in section 4.5.

It has been hypothesized that not only spatial trajectories are replayed during SWR's, but in general any type of episodic memory – especially in primates [28].

4.5 Synaptic plasticity

Long-term memories are formed by changing the strength of synapses between neurons, which is known as *synaptic plasticity*.[5]

Long-term potentiation (LTP) is any biomolecular process that strengthens synapses based on recent presynaptic spiking activity. It is often induced after the presynaptic neuron has fired at a relatively high frequency (a so called *tetanus* or burst of spikes).

Spike-timing dependent plasticity (STDP) is a phenomenon where synapses are strengthened or weakened based on both pre- and postsynaptic firing activity. When the postsynaptic neuron fires shortly after the presynaptic neuron, the synapse is strengthened. When it fired shortly before the presynaptic neuron however, the synapse is weakened. STDP is a translation of Hebb's 'rule' that "neurons that fire together, wire together" (extended with Hebb's intuitions about temporal precedence) [38].

As mentioned earlier, many neurons in area CA1 of the hippocampus 1) fire at high frequency during SWR's (100–200 Hz), and 2) have projections (i.e. axons) to the neocortex. This is ideal to induce LTP in downstream neocortical neurons. Additionally, the temporal firing sequences replayed at ripple-speed in the hippocampus (see section 4.4) are ideal to connect neurons through STDP, both within the hippocampus and within the neocortex.

4.6 Planning & decision-making

We briefly mention a possible complementary role for SWR's and replay, besides learning & memory.

Analyzing place cell spiking during awake SWR's revealed sped-up mental trajectories

[4]The area where it fires is called the neuron's *place field*. To be precise, the neuron can fire anywhere in the maze, but has an increased firing *rate* in its place field. Finding the rat's position based on neuron firing rates is called *neural population decoding*, and is typically performed using a Bayesian framework to invert the empirical place field encodings $p(\text{fire} \mid \text{position})$[37].

[5]This strengthening is measured as an increased postsynaptic current or potential-difference, for an identical presynaptic spike; i.e. the 'weight' or the gain of the connection has persistently increased.

through the maze, similar to sleep replay. These awake place cell sequences are generally more varied than sleep replay sequences: the replay can be in reverse (i.e. the place cells activate in reverse order than the navigated path), especially when the rat is consuming its reward. Or the place cell sequence can seemingly trace out a path not navigated before (this has been termed "preplay"). Additionally, while traversing a maze, place cell sequences have been observed to trace out paths ahead of the rat – especially when the animal stops at a fork in the maze.

Such observations are less well established than sleep replay, but – together with corroborating evidence from cognitive neuroscience[6] – point at a planning, imagination, and decision making-role for awake hippocampal replay and sharp wave-ripples.

4.7 Ripple disruption

The above evidence for the importance of SWR's is correlational / associational. Recently however, more direct evidence is being provided by studies that perform real-time SWR detection and disruption [39]–[44].

Such disruption is typically accomplished by inserting an electrode in the ventral hippocampal commisure (VHC), which is a tight bundle of axons connecting hippocampal neurons from both hemispheres. Stimulating this axon bundle by sending a current pulse through the electrode activates many hippocampal neurons, including inhibitory interneurons. These interneurons silence the hippocampus, such that none or very few spikes occur for $\pm$ 50 ms after stimulation. This means that, when the stimulation is applied during an SWR-event, the ripple oscillation disappears after the stimulation.

In 2009, two very similar studies were published that for the first time applied real-time ripple detection and disruption [39], [40]. In both studies, rats had to learn to navigate a simple maze for a chocolate reward. After each trial through the maze, the rats were allowed to sleep for one hour in a separate box. During this sleep, hippocampal ripples were detected in real time and disrupted through VHC stimulation. After sleep, the rats had to navigate the same maze. This was repeated for several days.

In both studies, ripple disruption deteriorated performance when compared with the controls:[7] more wrong turns were taken, navigating the maze took longer, and less sucessful runs were completed. Learning the maze over the course of days progressed more slowly. These results show quite directly the importance of SWR's for learning & memory – at least in the context of spatial navigation.

Other studies employing real-time ripple detection followed. As mentioned in section 3.2, SWR's also occur in the awake state. In Jadhav et al. 2012 [41], awake ripples were

[6]Cognitive neuroscience is, roughly, human neuroscience. Think psychological experiments, MRI machines, and clinical studies.

[7]The two studies used different controls: in Ego-Stengel and Wilson 2009 [39], trials on the maze where ripples were disrupted during subsequent sleep were alternated with trials on a second maze, where ripples were not disrupted during subsequent sleep. In Girardeau et al. 2009 [40], only one type of maze was used, but the rats were grouped in three types: rats without implanted electrodes, rats with ripple disruption, and rats where the VHC was stimulated, but only after $\pm$ 100 ms had passed after online detection of a ripple, meaning that the detected ripple is not disrupted. This controls for non-ripple-associated effects of VHC stimulation. The rats with delayed VHC stimulation performed equally well as the unimplaneted rats.

disrupted, while ripples during sleep were left undisturbed. This still resulted in a performance deficit for learning the maze, showing the importance of awake SWR's "for learning and memory-guided decision-making". In Girardeau et al. 2014 [42], ripple disruption was used to show that the hippocampus can generate more ripples when needed. Novitskaya et al. 2016 [45] used ripple detection to trigger stimulation not of the VHC, but of the *locus coeruleus*, a primitive and global activity regulator of the brain. In Kovács et al. 2016 [43], the hippocampus was silenced not by electrical, but by optogenetic stimulation. Finally, Talakoub et al. 2016 [44] showed that real-time ripple detection and disruption is not only feasible in rats, but also in primates.

All of these studies can benefit from accurate and fast online SWR detection, which is what is discussed in the remainder of this thesis.

Chapter 5

Offline labelling of SWR segments

To quantify the performance of a sharp wave-ripple (SWR) detection algorithm, we need an evaluation or 'test' recording, annotated with the segments of time when actual SWR events were present. This section describes how such annotations can be made, and how our data specifically was annotated.

SWR's are an empirical phenomenon of hippocampal area CA1, 'defined' by what their voltage traces look like. In other words, there is no ground truth available to know when SWR's occur. Scientists looking to annotate their LFP recordings with SWR segments therefore have to rely either on judgement calls by human labellers, or on an automated, offline SWR detection algorithm. The former could be considered more subjective, and is definitely more labour intensive than the latter – especially if multiple scientists are consulted to obtain a consensus labelling. Most studies use an automated, offline algorithm to detect SWR segments (see appendix A.1 for relevant quotes from a collection of such studies).

'Offline' here means that SWR detection happens after the recording has been completed, and that there are thus no real-time constraints on the detection algorithm. This means that 1) there are no hard bounds on algorithm execution time, and 2) that the algorithm can use information 'from the future': when deciding whether a recording sample $\mathbf{z}_t$ belongs to an SWR segment, it can consider samples $\mathbf{z}_{t_f}$ that occured after $\mathbf{z}_t$ (i.e. $t_f > t$), instead of considering only past samples $\mathbf{z}_{t_p}$ (where $t_p \leq t$).

5.1 Overview of offline ripple detection

The main steps of the offline SWR detection algorithm that most studies use – such as the ones cited in appendix A.1, and the one of this thesis – can be summarized as follows:

1. Use a single channel of input data; namely from an electrode in the pyramidal cell layer of CA1, where the ripple part of SWR's is most strongly present. (We will denote this voltage signal with z_t);
2. Band-pass filter the recording to retain only 'ripple' frequencies. (We will denote the filter output as o_t);

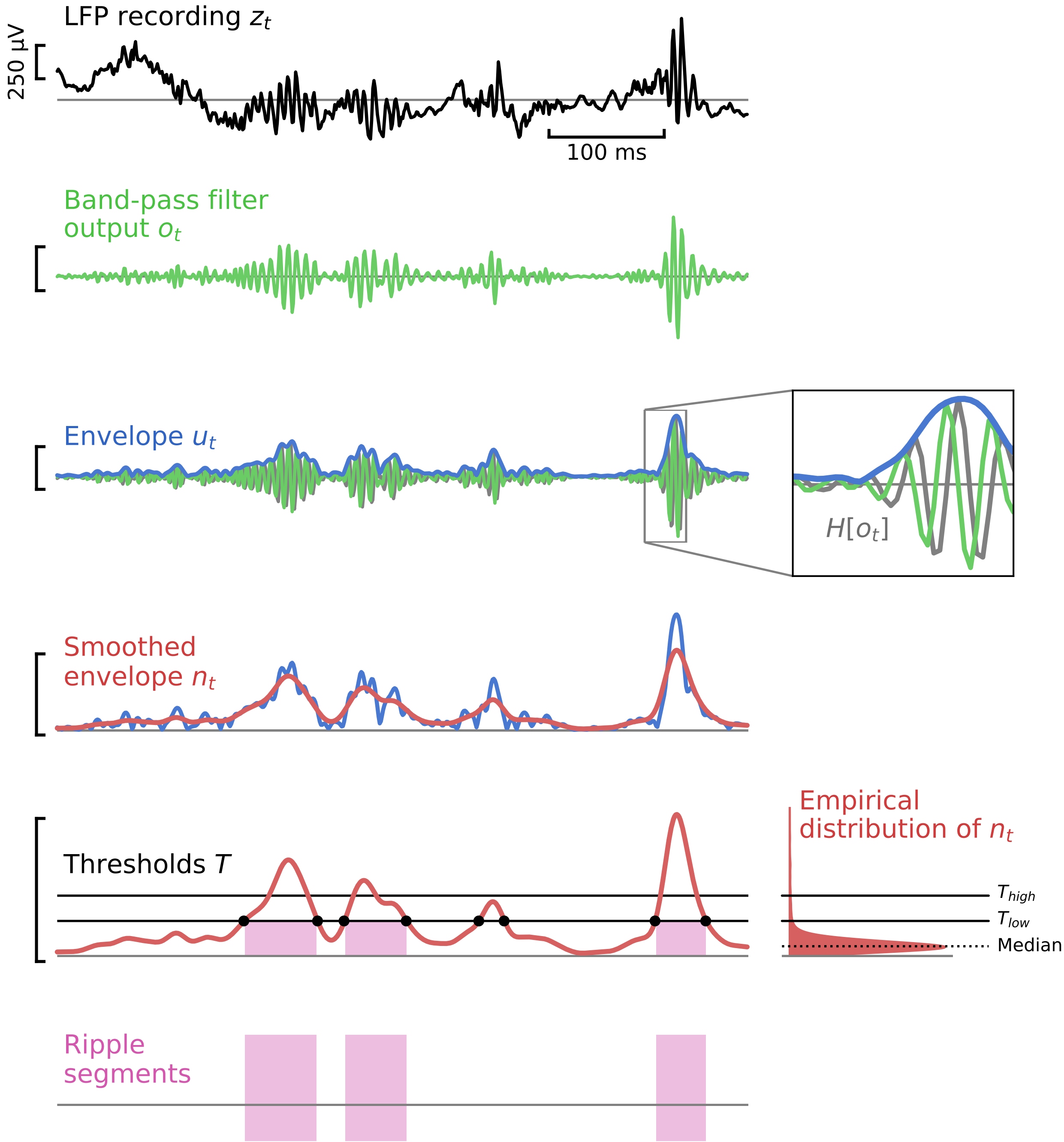


Figure 5.1: **Steps for automated, offline SWR labelling**. See text for details. Each vertical scalebar indicates the same voltage range. $H[o_t]$ denotes the Hilbert transform of o_t. Note its phase lag of 90° with respect to o_t. In the second panel from the bottom, note the two threshold crossings of T_{low} (marked with black dots) that did not result in a ripple segment, because the second threshold T_{high} was not reached. The distribution of the envelope n_t was estimated using the entire dataset (and not just the displayed fragment).

3. Obtain the envelope of the band-pass filtered signal. (We will denote this envelope with n_t);
4. Calculate a 'high' and a 'low' threshold (T_{high} and T_{low}) to apply to the envelope n_t, based on summary statistics of n_t and two custom multipliers (α_{high} and α_{low});
5. Define ripple events as times when the envelope crosses the high threshold (i.e. $n_t > T_{\text{high}}$);
6. Define the start and end time of each such ripple as the closest times where the envelope falls back below the lower threshold (i.e. $n_t < T_{\text{low}}$).

Figure 5.1 visualizes each step, as applied to a fragment of our dataset.

Note that this procedure only detects ripples, and not sharp waves.[1]

5.2 Validating parameter choices

Some of the above steps have free parameters (such as the ripple frequency band used for filtering, or the threshold multipliers α_{high} and α_{low}). If not mentioned otherwise, our parameter choices were made as follows.

In a first pass of the offline detection algorithm, ripples were detected using a very broad-band filter and a low detection threshold. This ensured that all 'true' SWR's were included in the detected events set (in addition to many spurious detections).

Next, five neuroscientists were independently asked to decide for each detected event whether it was a sharp wave-ripple or not. This was done through a custom-made web app, an example screen of which is shown in fig. 5.2. Note that the labellers could factor multiple recording channels into their decision, including stratum radiatum channels displaying sharp wave activity. Only the events that were labelled as an SWR by at least three neuroscientists were retained.

Finally, when setting the parameters of the eventual offline ripple detection algorithm, the algorithm's output was compared to the decisions made by the neuroscientists. The parameters were then adjusted until the output of the algorithm matched the neuroscientists' decisions reasonably well.

The following subsections describe the detection steps in more detail, and compare our choices of methods and parameters to those made in the literature.

5.3 Band-pass filter design

The wideband voltage signal x_t was band-pass filtered between 100 and 200 Hz, using a linear time-invariant filter with zero output lag.

[1] Although interestingly, the sharp wave part of sharp wave-ripples was discovered before the ripple part [29, p. 1].

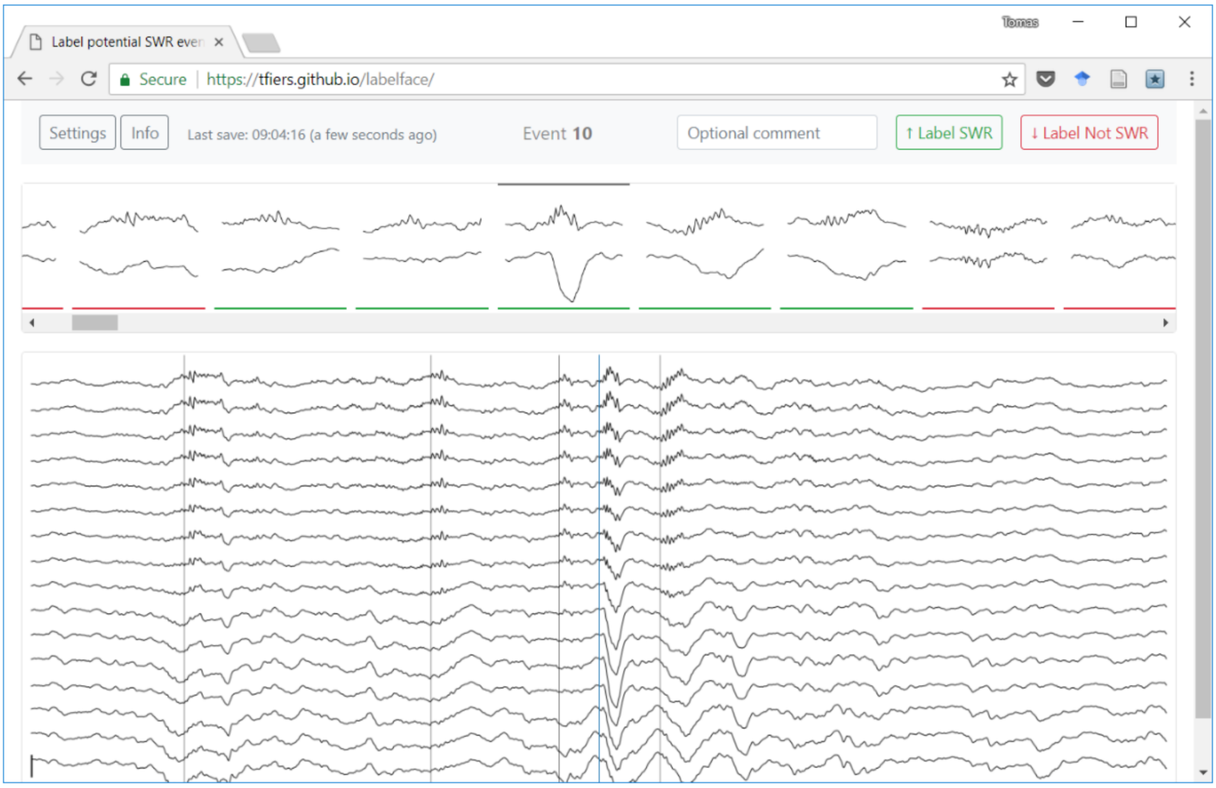


Figure 5.2: **User interface for SWR labelling**. Each event in the list at the top is represented by two voltage traces: one from an electrode in the pyramidal cell layer (the top trace), and one from an electrode in the stratum radiatum. The large plot at the bottom gives more comprehensive view of the currently active event: all 16 channels are plotted (instead of only two), and the plot ranges from 1000 ms before to 1000 ms after the event. In this large plot, the blue vertical line marks the currently active event. The grey vertical lines correspond to other detected events. (The scalebar at the beginning of this plot represents 1 mV). Users decide whether the currently active event is an SWR by clicking the buttons in the top-right corner or by using keyboard shortcuts. The source code for this labelling web app is available at `https://github.com/tfiers/labelface`.

Many other studies use a higher left bound, of about 140–150 Hz (see table A.1). Using such a high bound in our dataset however resulted in ripples that went undetected (even at low detection thresholds), although they were convincingly marked as SWR's by the neuroscientists.

The band-pass filter was designed using the windowed-sinc method, with a Kaiser window (using SciPy's `firwin` and `kaiserord` functions) [46], [47]. The transition width was chosen to be 10% of the bandwidth (i.e. 10 Hz), and the attenuation to be 40 dB, resulting in an FIR filter of order 150 (at a 1000 Hz sampling frequency).

The filter was applied bidirectionally, resulting in a zero-lag output, and a total attenuation of 80 dB. Figure 5.1 shows an example of the filter output in green.

(We cannot easily compare our filter design method with the literature, as most studies do not mention anything about filter design besides the frequency band used).

5.4 Envelope calculation

Next, the instantaneous envelope u_t of the filter output o_t was calculated using its Hilbert transform[2] $H[o_t]$:

$$u_t = \sqrt{o_t^2 + H[o_t]^2}, \tag{5.1}$$

The Hilbert transform delays each frequency component of a signal by 90° (see the gray signal in fig. 5.1) [49]. This means that the envelope of a narrowband signal such as o_t can be easily obtained as the magnitude of the so called 'analytic signal' $o_t + jH[o_t]$, as in eq. (5.1).[3]

After calculating u_t, a final, smoothed envelope n_t was obtained by convolving u_t with a Gaussian kernel (σ = 7.5 ms, support radius of 4σ). Compare the blue (u_t) and the red signal (n_t) in fig. 5.1.

Other studies (such as [50] and [51]) use a "root-mean-square" approach to calculate the envelope of the filter output o_t. In these studies, presumably, the squared signal o_t^2 is smoothed using some kernel (of unspecified type and bandwidth) to obtain the "mean-square" signal of which the square root is taken.

5.5 Threshold calculation

The two detection thresholds were calculated as follows:

$$T_{\text{high}} = \alpha_{\text{high}} \times \widetilde{n_t} \tag{5.2}$$

$$T_{\text{low}} = \alpha_{\text{low}} \times \widetilde{n_t} \tag{5.3}$$

where $\widetilde{n_t}$ denotes the median of the smoothed envelope n_t. As per the procedure described in section 5.2, we set $\alpha_{\text{high}} = 6.2$ and $\alpha_{\text{low}} = 3.6$. At a median envelope magnitude $\widetilde{n_t} = 17.0\,\mu\text{V}$, this results in thresholds $T_{\text{high}} = 105.4\,\mu\text{V}$ and $T_{\text{low}} = 61.2\,\mu\text{V}$.

Most studies calculate thresholds as follows: $T_{\text{high}} = \overline{n_t} + \beta_{\text{high}} \times \text{std}(n_t)$. Here $\overline{n_t}$ denotes the mean of the envelope, $\text{std}(n_t)$ denotes its standard deviation, and β_{high} is a custom multiplier analogous to α_{high}. For the non-negative, assymetric distributions of envelope signals, using both a measure of center and a measure of spread to define thresholds seems unnecessary (see the distribution of n_t in fig. 5.1), which is why we chose to use only one measure (namely the median).

[2] To be precise, H is a discrete approximation to the continuous Hilbert transform $\mathcal{H}$. There are multiple ways to make a discrete approximation to $\mathcal{H}$ [48]. Most of them make use of the discrete Fourier transform, which makes computing H an efficient operation. We used SciPy's `hilbert` function (which, confusingly, does not return the Hilbert transform of its input, but rather its analytical signal), and zero-padded the input signal to the nearest power of 2 or 3 to benefit from the speedup brought by the fast Fourier transform algorithm.

[3] To see why, consider a local approximation of the narrowband signal o_t by a sinusoid $a\cos\omega t$. Its Hilbert transform is then $a\sin\omega t$. From eq. (5.1), the envelope u_t will be locally approximated by the magnitude of the original signal: $u_t = \sqrt{(a\cos\omega t)^2 + (a\sin\omega t)^2} = |a|$.

The detection multiplier β_{high} varies wildly between studies: from 1, over 3, 4, and 5, up until 7 ([50]–[54], respectively).[4] It is clear that such different thresholds will give very different sensitivity-precision trade-offs for SWR detection (the lower thresholds yielding more false positive detections, and the higher thresholds yielding more missed true SWR events).

To compare our thresholds to those in the literature, we calculate the β multipliers corresponding to our chosen α multipliers. Given that our dataset has a mean envelope magnitude of 22.3 µV and a standard deviation of 22.9 µV, we find $\beta_{\text{high}} = 3.63$ and $\beta_{\text{low}} = 1.70$. These values fall near the center of those reported in the literature (see appendix A.1).

5.6 Segment post-processing

Finally, two add-hoc rules were applied to the automatically detected ripple segments. First, segments with only a small gap between them (of less than 10 ms) were joined together. Then, segments of too short a duration (less than 25 ms) were eliminated.

This step is rarely done in other studies. An exception is e.g. [52], were segments shorter than 15 ms were eliminated.

[4] Threshold multipliers cannot be compared precisely. Imagine two recordings with equally powerful ripples. Both recordings will then need an equal threshold T_{high} to detect the same types of ripples. When one of the recordings has a different background 'noise' level or a different ripple incidence rate, it will have different $\overline{n_t}$ and $\text{std}(n_t)$ values. This means that β_{high} needs to change to maintain an equal threshold T_{high}.

Chapter 6

Performance quantification of online SWR detectors

Chapter 5 described how we annotated our dataset with 'reference' SWR segments, using an offline algorithm. These reference segments can now be used to benchmark the performance of an online SWR detection algorithm. This section describes how this is done.

6.1 Generating online detections

Each online SWR detection algorithm studied in this thesis generates an output envelope n_t, where samples n_t of larger magnitude denote a higher belief that the corresponding input sample $\mathbf{z}_t$ is part of an SWR event. (See fig. 8.2A for three example output envelopes).

These output envelopes n_t can be converted to a discrete set of detection times by applying a threshold T to the envelope.[1] Additionally, we require a certain minimum duration between online detections (which we will call the "lockout time", L). The set D of detection times t_d (with $d = 0, 1, \, ..$) is then:

$$D = \{ \; t_d \mid \; n_{t_d} > T \quad \text{and} \quad t_d > t_{d-1} + L \; \}. \tag{6.1}$$

In our analysis, we chose a lockout time L based on SWR durations. Specifically, we set L to the 25-th percentile of the durations of all SWR segments detected in our dataset by the offline algorithm, resulting in a lockout time $L = 34$ ms.

The threshold T is generally set by the user of the online SWR detection algorithm (see the discussion in the next section).

[1] In general, this threshold can be adaptive (based on a moving average of signal statistics, for example), and we should therefore write "T_t" to be precise. In this thesis we only evaluate online detectors using constant thresholds T.

6.2 Quantifying accuracy

In the following, we denote reference SWR segments as the closed intervals $s_r = [t_r^{\text{start}},\ t_r^{\text{end}}]$ (with $r = 0, 1, ..$). The set of all reference segments is denoted by $S = \{s_r\}$.

To benchmark the performance of an online SWR detection algorithm (at a certain threshold T), we compare its detections D with the offline reference segments S, and divide both sets into two subsets each: namely detected versus undetected reference segments, and correct versus incorrect detections. Formally:

$$D^{\text{correct}} = \{t_d \mid \exists s_r : t_d \in s_r\} \tag{6.2}$$

$$D^{\text{incorrect}} = D \setminus D^{\text{correct}} \tag{6.3}$$

$$S^{\text{detected}} = \{s_r \mid \exists t_d : t_d \in s_r\} \tag{6.4}$$

$$S^{\text{undetected}} = S \setminus S^{\text{detected}}, \tag{6.5}$$

i.e. correct online detections are contained within a reference segment, and detected reference segments contain at least one online detection. Figure 8.2A shows some example correct and incorrect detections (green and red triangles, respectively).

Next, the number of detected reference segments is counted, and compared to the total number of reference segments. This fraction is the **recall** R of the SWR detector (also known as sensitivity, hit rate, or true positive rate):

$$R = \frac{|S^{\text{detected}}|}{|S|} = \frac{\text{\# of detected reference segments}}{\text{total \# of reference segments}} \tag{6.6}$$

Similarly, the number of correct detections is counted, and compared to the total number of detections. This fraction is the **precision** P of the SWR detector (also known as the positive predictive value):

$$P = \frac{|D^{\text{correct}}|}{|D|} = \frac{\text{\# of correct detections}}{\text{total \# of detections}} \tag{6.7}$$

Its opposite is the false discovery rate (FDR), which counts the number of so called “false positive” detections:

$$FDR = \frac{|D^{\text{incorrect}}|}{|D|} = \frac{\text{\# of incorrect detections}}{\text{total \# of detections}} = 1 - P \tag{6.8}$$

Each detection threshold yields a different (P, R)-combination. This is often a trade-off: higher thresholds yield more precise detectors (less false positive detections), at the cost of a lower sensitivity (more missed reference SWR segments). Conversely, lower thresholds generally yield more sensitive but also less precise SWR detectors.

This is why the performance of a detection algorithm is often given for a *range* of thresholds, instead of for just a single threshold: different users of the SWR detection algorithm may prefer different trade-offs between latency and precision. (One user might require a minimal amount of false detections, for example, while another may

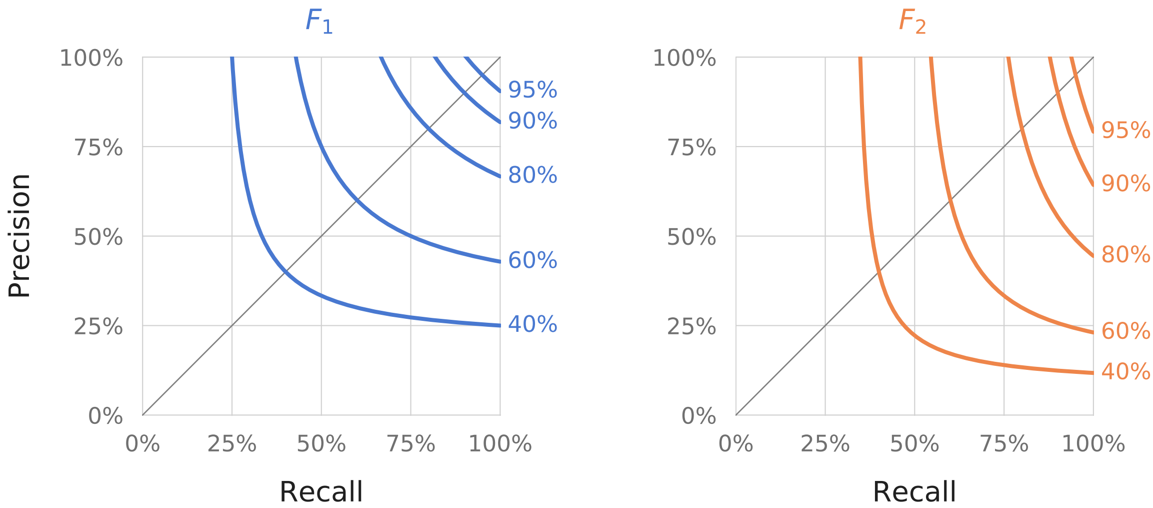


Figure 6.1: **Iso-F_β-curves**.

want to detect as many SWR events as possible, without minding false detections). When precision and recall are plotted against each other for a range of different thresholds, the result is a so called "PR-curve" (see the lower-left panel of fig. 8.3 for an example). Generally, higher quality detectors have PR-curves that draw nearer to the top-right $(1, 1)$-corner.

The precision P and recall R for a given threshold can be summarized in a single number: the so called F-score. It is defined as:

$$F_\beta = \frac{(1+\beta^2)PR}{\beta^2 P + R}. \tag{6.9}$$

It is the mean of recall and precision (more precisely: their harmonic mean, as precision and recall are ratio's), with recall weighted by β^2. It measures detection performance "for a user who attaches β times as much importance to recall as to precision." [1]

For example, the F_1-score weighs recall and precision equally. When we want to summarize the accuracy of a detector in a single number, we will often choose the *maximum* F_1-score of the detector (over the entire threshold range) as this number.

For an idea of the relationship between P, R, and F-scores, see fig. 6.1.

A note on terminology: we will refer to the precision and sensitivity performance of a detector often simply as the 'accuracy' of the detector. This is a shorthand, and does not refer to the technical definition of accuracy.[2]

[2]Technically, accuracy is the sum of true positive and true negative cases, divided by the total number of cases. This concept does does not readily apply to SWR detection, as there is no straightforward definition of "true negative" cases here. Only false positives ($D^{\text{incorrect}}$), true positives (D^{correct} and S^{detected}), and false negatives ($S^{\text{undetected}}$) are clearly defined.
Note that there are two types of "true positives": correct detections $\in D^{\text{correct}}$, and detected reference segments $\in S^{\text{detected}}$ (with $|D^{\text{correct}}| \geq |S^{\text{detected}}|$, because one reference segment may contain multiple detections).

6.3 Quantifying latency

Besides counting how many SWR segments were detected, we are also interested in how early they were detected. For each detected reference segment $s_r = [t_r^{\text{start}},\ t_r^{\text{end}}]$, we note the first online detection t_d contained within it (reference segments may contain multiple online detections).

The absolute detection latency of this SWR segment is then defined as:

$$L_r^{\text{abs}} = t_d - t_r^{\text{start}} \tag{6.10}$$

We also define a detection latency relative to the duration of the SWR segment in question:

$$L_r^{\text{rel}} = \frac{L_r^{\text{abs}}}{t_r^{\text{end}} - t_r^{\text{start}}} \tag{6.11}$$

Whereas an SWR detector yields only a single precision and a single recall value for a given threshold, it yields two entire sets of absolute and relative detection latencies (namely an absolute and a relative latency value for each detected reference segment). This is why we often summarize latency performance using the median of one of these sets. When not otherwise specified, the latency distribution (or its median) corresponding to the threshold with maximal F_1-score is reported.

Chapter 7

Current online SWR detectors

This chapter discusses the current state-of-the-art method in online SWR detection. This method is very similar to the procedure for offline ripple detection described in chapter 5: a single recording channel that contains ripples is selected; it is band-pass filtered through the ripple band; and apply a detection threshold is applied to the envelope of the filter output.

The difference is that the SWR detection must now happen under real-time constraints. No future samples can be used when deciding whether the current recording sample belongs to an SWR event. This means that the used band-pass filter (and in some studies, also the envelope estimation method) must necessarily introduce latency.

We first discuss the performance of the online band-pass filters used in the literature, and then shortly discuss online estimation of the filter output envelope.

7.1 Replicating existing filters

We selected three previously used online ripple filters and analyzed their performance. The first two filters are described in the literature. (We gathered a representative sample of original research papers that use online ripple detection – see appendix A.2. Most of these papers do not provide sufficient information to replicate the used filter. The two selected filters are from the two papers that did provide sufficient information.) The third analyzed filter is the default online ripple filter in *Falcon*. (*'Falcon'* is an open-source software package for closed-loop neuroscience, developed and used in the Kloosterman lab, and described in Ciliberti et al. 2017 [55]).

The original analyzed filters use different passbands and sampling frequencies. When we 'replicate' the original filters to apply them to our data, we use a standardized sampling frequency f_s of 1000 Hz. When designing the replica filters, we try to match the gain- and group delay-profiles of the original filters.

Ego-Stengel et al. 2009

In one of the first ripple disruption studies [39], an analog band-pass filter is used. An 8th-order Butterworth high-pass filter at 100 Hz is combined with an 8th-order

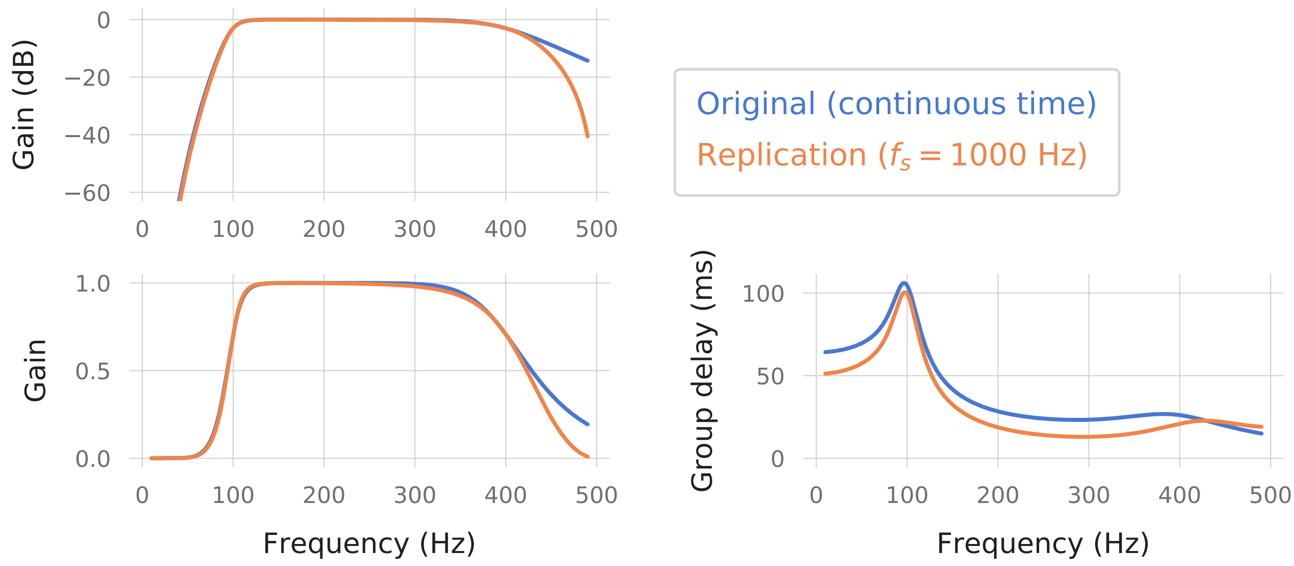


Figure 7.1: **Online ripple filter from Ego-Stengel et al. 2009**. The group delay of each filter is calculated as $-\,\mathrm{d}\phi/\mathrm{d}f$, where ϕ is the unwrapped phase response of the filter. (The derivative is calculated numerically using a Savitsky-Golay filter, with 5 samples per window, cubic polynomials, and $\Delta f \approx 0.05$ Hz). This method is also used for subsequent group delay plots.

Butterworth low-pass filter at 400 Hz. We approximate this continuous-time filter by the product of two discrete-time Butterworth filters at $f_s = 1000$ Hz: an 8th-order high-pass filter at 100 Hz, and a 2nd-order low-pass filter at 400 Hz.[1] Its frequency response is compared with that of the original filter in fig. 7.1.

Dutta et al. 2018

In this recent analysis of an online ripple detection system [52], a 30-tap FIR filter at $f_s = 3000$ Hz is used. In the open source code of this study[2], we find that the filter is designed using the windowed-sinc method, with a Hamming window. We approximate this filter at a sampling frequency of 1000 Hz with an 11-tap FIR filter, designed using the same method, window type, and passband. The correspondence is good (see fig. 7.2). Note the low attenuation outside the passband, and the constant, relatively low group-delay.

Falcon (Ciliberti et al. 2017)

The default ripple filter in *Falcon* is a Type II Chebyshev filter, with a passband of $\approx$ 130–283 Hz, edge widths of $\approx$ 10 Hz, a minimum attenuation in the stopbands of 40 dB, and

[1] Note that we do not choose the literal discretization of the continuous-time filter (namely the combination of an 8th-order high-pass and an 8th-order low-pass Butterworth filter, discretized using e.g. the zero-order-hold method or Tustin's bilinear transform). Around 400 Hz, this discretization has too high a peak in group delay and too steep a loss in gain for its frequency response profiles to match the original filter well.

[2] `https://github.com/shayokdutta/RippleDetectionAnalysis/blob/master/DataAnalysisScripts/ripple_filtering.py`

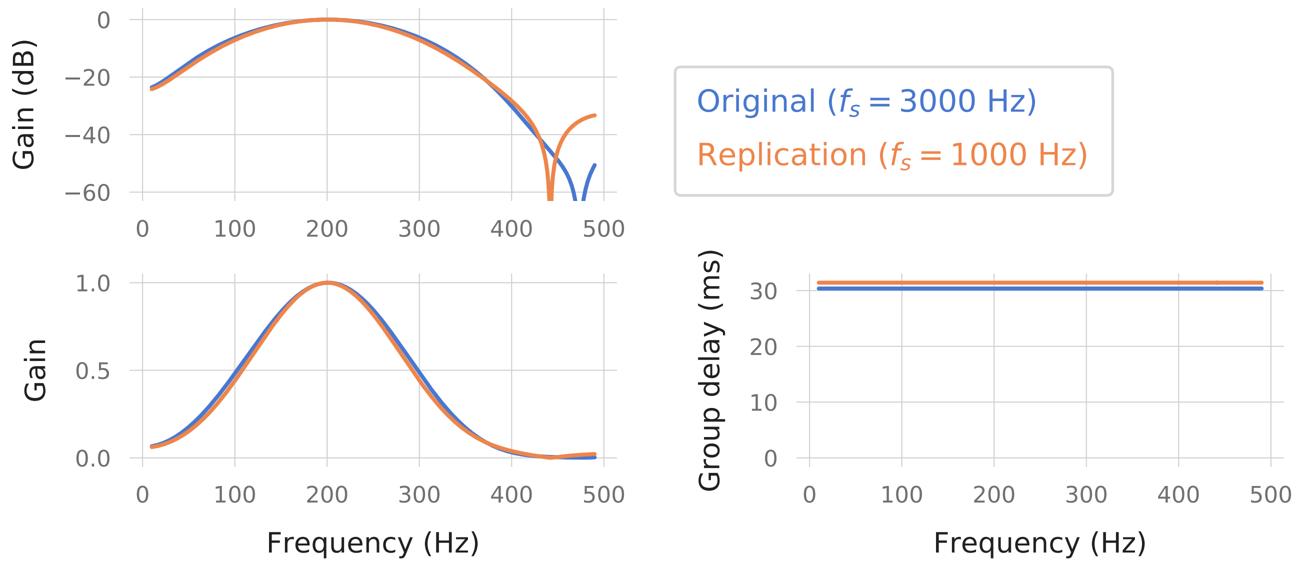


Figure 7.2: **Online ripple filter from Dutta et al. 2018**.

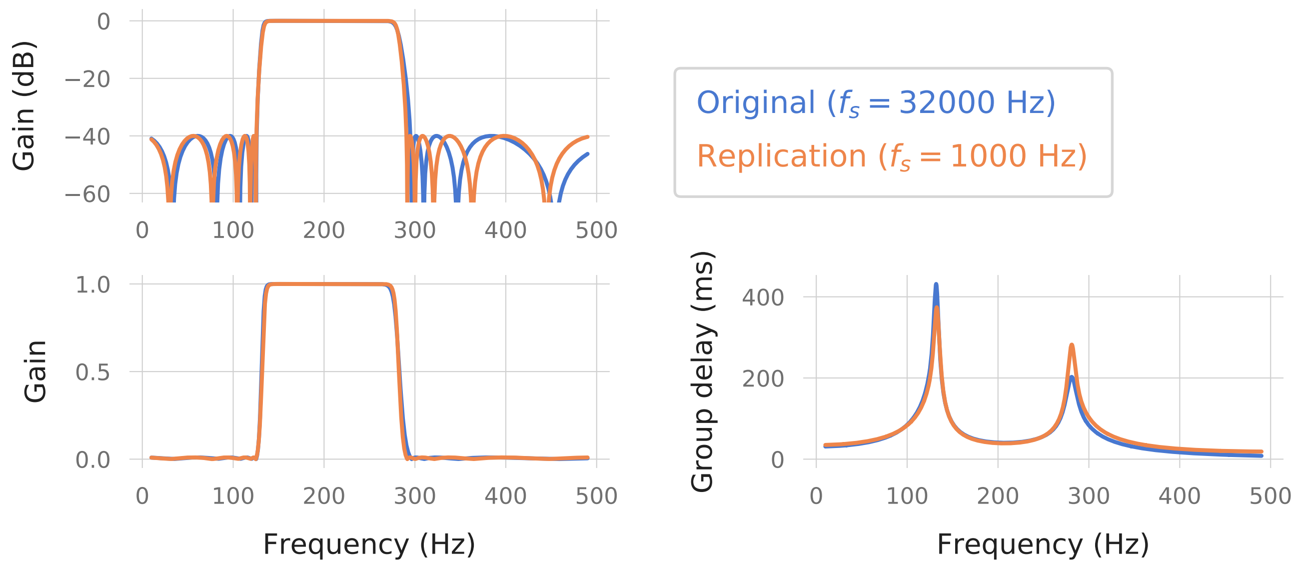


Figure 7.3: **Default online ripple filter in Falcon**.

a maximum attenuation in the passband of 1 dB. The original filter operates on data with a sampling rate of 32 kHz. We can closely approximate this filter at $f_s = 1000$ Hz using a 10th-order (21-taps) Type II Chebyshev filter, with the same design parameters. Its frequency response is shown in fig. 7.3. Note the trade-off between filter quality and group delay apparent in these and the previous plots: sharp passband edges correspond to high group delays, and vice-versa.

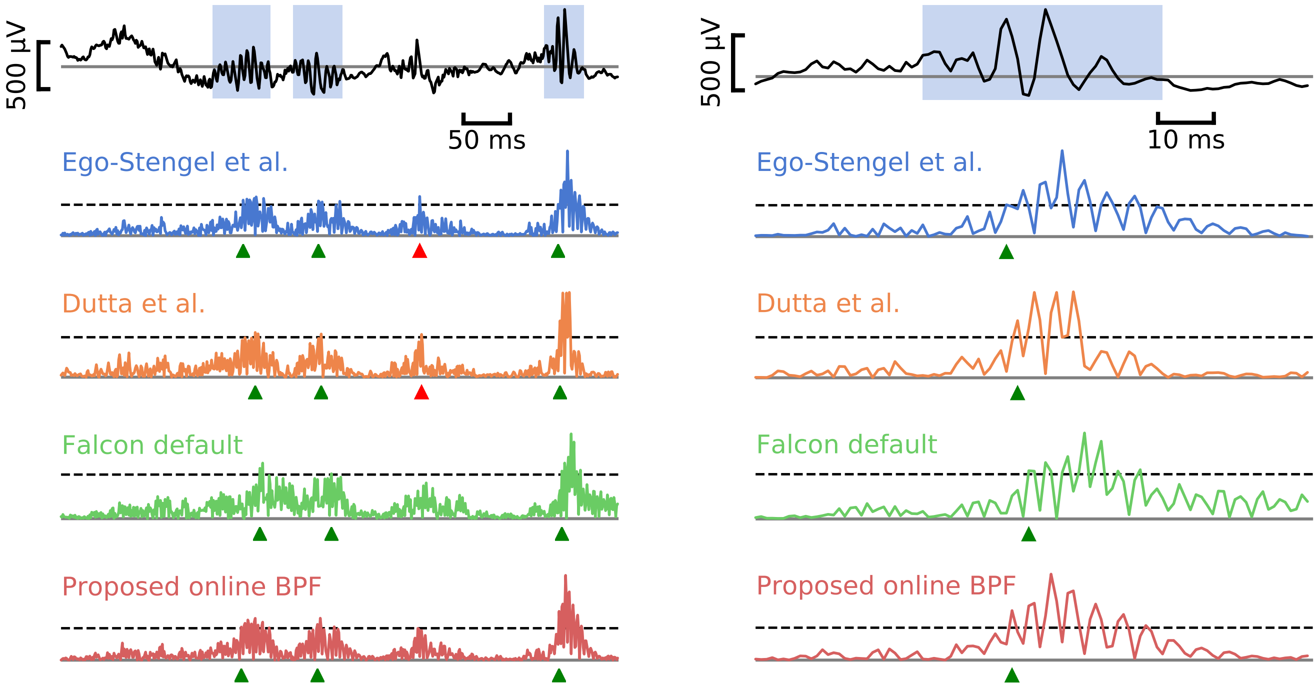


Figure 7.4: **Rectified outputs of online ripple filters**. Black signal: Selected channel from LFP recording $\mathbf{z}_t$. Light-blue vertical bands: reference SWR segments. Colored signals: output envelopes n_t of the different online ripple filters. Dashed horizontal lines: detection thresholds. (For each filter, the $\max F_1$ threshold was chosen). Green triangles: correct detections. Red triangles: incorrect detections. *Left*: 600 ms of input and output data. *Right*: zoom in on the last ripple of the left panel.

7.2 SWR detection performance

Applying these ripple filters at $f_s = 1000$ Hz to our recording yields online envelopes as in fig. 7.4. Quantifying detection performance as described in chapter 6, and using the entire evaluation data set and for a range of different thresholds, results in the PR- and latency curves of fig. 7.5.

We notice the relatively low recall and precision of the default ripple filter in *Falcon*. This is due to the presence of sharp wave-ripple events with ripple frequency < 130 Hz in our dataset (as described earlier and as confirmed by the manual labelling effort by neuroscientists). The default *Falcon* online ripple filter has a left passband edge of $\pm$ 130 Hz, and therefore does not detect these ripples.[3]

We design an online ripple filter (at $f_s = 1000$ Hz) with equal or better detection performance than the analyzed ripple filters. This filter is labelled "Proposed online BPF" in figs. 7.4 and 7.5, and is based on our replica of the Ego-Stengel et al filter. It is the product of a 6th-order high-pass Butterworth filter at 100 Hz, and a 1st-order low-pass Butterworth filter at 200 Hz. This filter will be used as the baseline SWR detection

[3]When we design a new band-pass filter based on the default *Falcon* filter (same design characteristics) but with a passband $\approx$ 100–200 Hz, we obtain a marked improvement in detection accuracy. The PR-curve (not shown) then resembles that of the Dutta et al filter. Detection latency increases even further though.

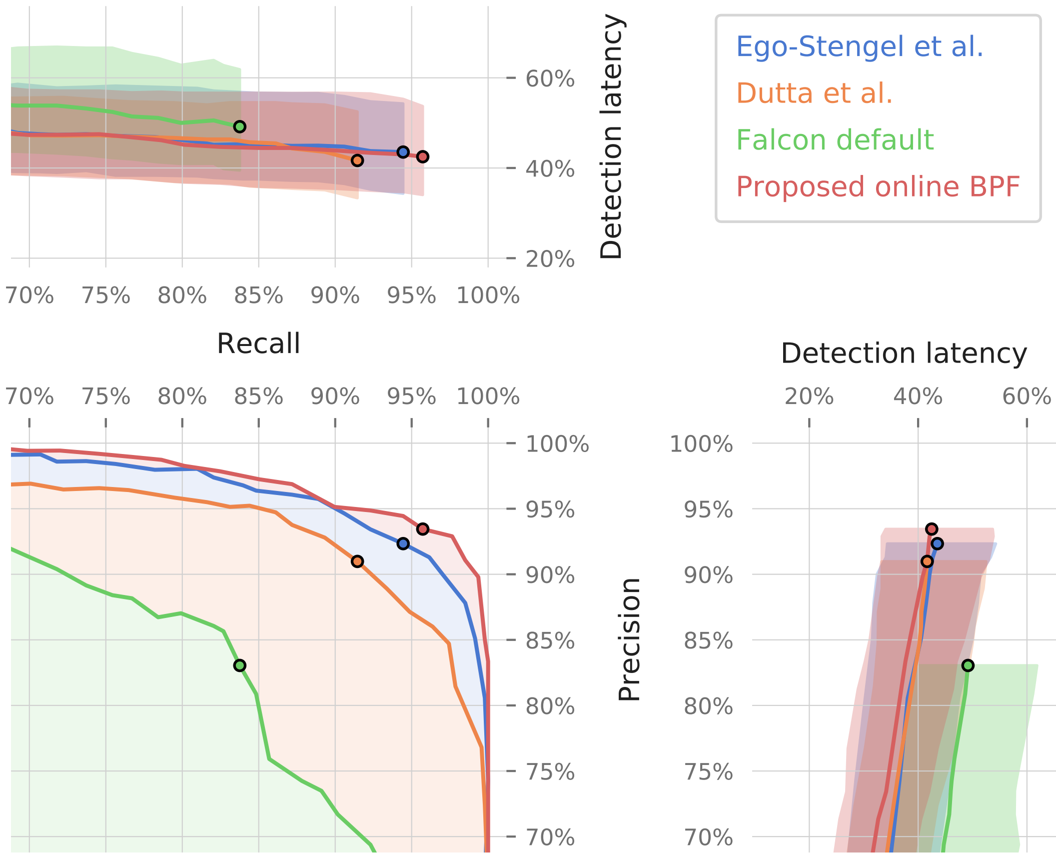


Figure 7.5: **Detection performance of different online ripple filters**. Sensitivity, precision & latency trade-offs for a range of thresholds.
Each threshold setting for a ripple filter corresponds to a point on the precision-recall curve in the bottom-left panel, and to a distribution of relative detection latencies. The median and interquartile range of this distribution are plotted in the top-left or bottom-right panel (as a point of the bold curve, and a slice of the shaded band, respectively).
These latency distribution plots are divided over two panels so that their entire range can be clearly visualized, both for the low recall – high precision regime as for the high recall – low precision regime. The cutoff is made at the $\max F_1$ point, marked with shaded black circles.

algorithm in the following chapter.

7.3 Online envelope estimation

Besides requiring causal band-pass filters, the real-time constraint of online SWR detection also means that we can no longer use the Hilbert transform to calculate the envelope n_t of the band-pass filter output o_t. In practice, the band-pass filter output is instead often simply rectified (i.e. $n_t = |o_t|$).

Extensions of simple rectification have also been used. In Jadhav et al. 2012 [41], the online envelope n_t is an exponentially weighted moving average (EWMA) of $|o_t|$, with a variable gain. (In their EWMA update equation, a slightly higher weight is used for $|o_t|$ whenever $n_t < |o_t|$ – see appendix A.2). In Dutta et al. 2018 [52], $|o_t|$ is smoothed with a 50 Hz low-pass FIR filter.

Both methods add latency. The EWMA method of Jadhav et al. delays the envelope by about 2 milliseconds when applied to our recording. The FIR low-pass filter of Dutta et al. has a constant group delay of about 5 milliseconds.

Smoothing $|o_t|$ yields a visually more pleasing envelope. This is not a requirement for online SWR detectors however. Smoothing may prevent spurious detections whenever the unsmoothed signal contains outliers (strong peaks in magnitude that do not correspond to a peak in the underlying 'real' signal). Our recording (and the band-pass filtered version of it) does not seem to contain such outliers however. Because smoothing adds latency, we do not smooth our online envelopes; all online envelopes in this thesis are calculated as $n_t = |o_t|$.

Chapter 8

Multi-channel linear filtering

8.1 Data-driven algorithms

In this and the following chapter, we describe *supervised*, or data-driven SWR detection algorithms: they require training data $\mathbf{z}_t^{\text{train}}$, and an associated labelling y_t^{train} which marks the presence of an SWR event in $\mathbf{z}_t^{\text{train}}$, for every discrete time sample t. For example, $y_t \in \{0, 1\}$, with $y_t = 1$ when the corresponding input sample $\mathbf{z}_t$ is part of an SWR segment, and $y_t = 0$ when it is not.

The problem of obtaining such a labelling y_t for some recording data $\mathbf{z}_t$ is the topic of chapter 5. Training labels can be obtained either by human expert labellers, or by using an automated *offline* SWR detection algorithm, where we assume that the automated labelling corresponds well to a supposed human expert labelling. In this thesis, we use the automated labelling method of chapter 5 to generate target labellings y_t^{train}.

Before a supervised algorithm can be used for real-time detection, its parameters have to be 'tuned'. This is done using a training dataset $(\mathbf{z}_t^{\text{train}}, y_t^{\text{train}})$, during the so called *training phase*. Parameters are changed such that the algorithm's output o_t for an input $\mathbf{z}_t^{\text{train}}$ matches the target labelling y_t^{train} well. Sections 8.2 and 9.3 describe how this tuning can be done for two concrete detection algorithms.

The hope is that the trained algorithm also performs well on input data $\mathbf{z}_t^{\text{test}}$ not part of the training set. That is, that the algorithm has good *generalization performance*. When this is not the case and the algorithm is tuned so that it only performs well on the training data, we say that the algorithm has been *overfit*. Often, so called *regularization* methods exist to discourage overfitting on the training data.

8.2 Linear signal-to-noise maximisation

In this chapter, we search for a linear combination of channels that yields an output signal o_t useful for sharp wave-ripple detection. More precisely, we search for a vector $\mathbf{w} \in \mathbb{R}^C$ in channel (or electrode) space to project the samples $\mathbf{z}_t \in \mathbb{R}^C$ on, so that the output signal

$$o_t = \mathbf{w}^T \mathbf{z}_t \tag{8.1}$$

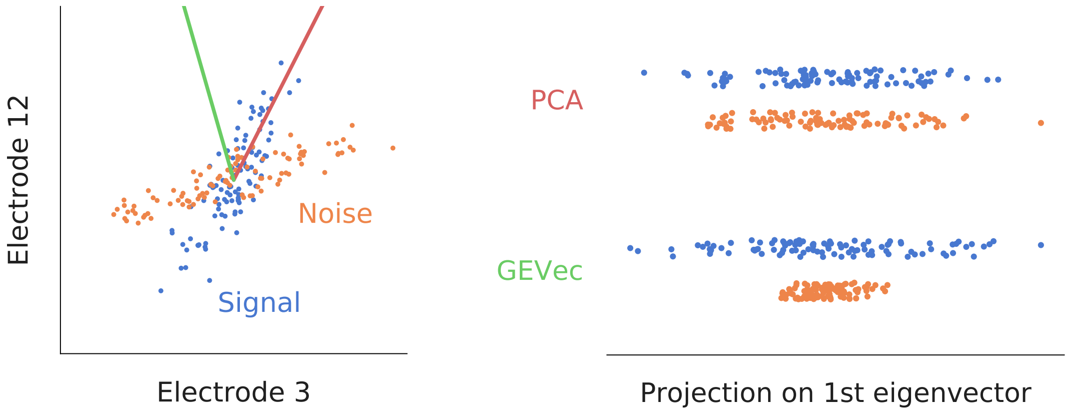


Figure 8.1: **Linear signal-to-noise maximisation**. Toy example to illustrate the generalized eigenvector approach to signal detection. *Left*: multi-channel time-series data plotted in 'phase space' (meaning without time axis), with blue dots representing samples where the signal was present, and orange dots representing samples where it was not. Actually toy data drawn from two 2-dimensional Gaussian distributions with different covariance matrices. Red vector: first eigenvector of the signal covariance matrix (also known as the first principal component). Green vector: first generalized eigenvector of the signal and noise covariance matrices. *Right*: Projection of both data sets on both the ordinary eigenvector ("PCA") and the generalized eigenvector ("GEVec"). The ratio of the projected signal data variance versus the projected noise data variance is maximised for the GEVec case.

has high variance (or power) during SWR events, and low variance outside them.[1] This principle is illustrated with a two-dimensional toy dataset in fig. 8.1. We can then detect SWR events using threshold crossings of the envelope of o_t, as discussed in chapter 6.

The next two sections describe how this vector $\mathbf{w}$ can be found.

The optimisation problem

Suppose all training samples $\mathbf{z}_t^{\text{train}}$ are gathered and divided over two data matrices $\mathbf{S} \in \mathbb{R}^{C \times N_S}$ and $\mathbf{N} \in \mathbb{R}^{C \times N_N}$, where $\mathbf{S}$ (for 'signal') contains all N_S samples of $\mathbf{z}_t^{\text{train}}$ where an SWR is present, and $\mathbf{N}$ (for 'noise') contains all N_N other samples. (These matrices can be easily constructed by concatenating segments from $\mathbf{z}_t^{\text{train}}$).

[1] We assume that the input signals are zero-mean, such that the power P of the output signal equals its variance: $P_o = \langle o_t^2 \rangle = \langle (o_t - \mu_o)^2 \rangle = \mathrm{Var}(o_t)$ when $\mu_o = 0$, which is the case for zero-mean input channels: $\mu_o = \langle o_t \rangle = \langle \mathbf{w}^T \mathbf{z}_t \rangle = \sum_i w_i \langle z_{t,i} \rangle = 0$ when $\langle z_{t,i} \rangle = 0$ for all channels i.
This zero-mean assumption is reasonably well fulfilled for the analysed recording: the sample values of a 10-second moving average of the recording are near zero (median -0.02 µV, IQR 0.13 µV. In comparison, the RMS-value of the recording is 210 µV).
Input data that is not zero-mean can be readily transformed to be so (even in an online setting), by subtracting a (moving) average from the input signal.

Equation (8.1) then becomes, in vector notation:

$$\begin{aligned}\mathbf{o}_S &= \mathbf{w}^T\mathbf{S}\\ \mathbf{o}_N &= \mathbf{w}^T\mathbf{N},\end{aligned}$$

where each element of the row vectors $\mathbf{o}_S$ and $\mathbf{o}_N$ is a filtered sample of $\mathbf{S}$ and $\mathbf{N}$, respectively. Figure 8.1 (right) shows the distribution of the values in two example data vectors $\mathbf{o}_S$ and $\mathbf{o}_N$.

We want to find the weight vector $\widehat{\mathbf{w}}$ that maximises the variance of $\mathbf{o}_S$ versus the variance of $\mathbf{o}_N$, i.e.

$$\widehat{\mathbf{w}} = \arg\max_{\mathbf{w}} \frac{\mathrm{Var}(\mathbf{o}_S)}{\mathrm{Var}(\mathbf{o}_N)} \tag{8.2}$$

$$= \arg\max_{\mathbf{w}} \frac{\frac{1}{N_S}\mathbf{o}_S\mathbf{o}_S^T}{\frac{1}{N_N}\mathbf{o}_N\mathbf{o}_N^T}$$

$$= \arg\max_{\mathbf{w}} \frac{\frac{1}{N_S}\mathbf{w}^T\mathbf{S}\mathbf{S}^T\mathbf{w}}{\frac{1}{N_N}\mathbf{w}^T\mathbf{N}\mathbf{N}^T\mathbf{w}}$$

In this last equation, we recognize the empirical covariance matrices $\mathbf{R}_{SS}$ and $\mathbf{R}_{NN}$, which are defined as:

$$\mathbf{R}_{SS} = \frac{1}{N_S}\mathbf{S}\mathbf{S}^T \tag{8.3}$$

$$\mathbf{R}_{NN} = \frac{1}{N_N}\mathbf{N}\mathbf{N}^T \tag{8.4}$$

$\mathbf{R}_{SS} \in \mathbb{R}^{C\times C}$ and $\mathbf{R}_{NN} \in \mathbb{R}^{C\times C}$ are symmetric matrices, where each diagonal element yields the variance of a channel, and each off-diagonal element yields the covariance between a pair of channels.

The condition for the optimal weight vector, eq. (8.2), is thus equivalent to:

$$\widehat{\mathbf{w}} = \arg\max_{\mathbf{w}} \frac{\mathbf{w}^T\mathbf{R}_{SS}\mathbf{w}}{\mathbf{w}^T\mathbf{R}_{NN}\mathbf{w}} \tag{8.5}$$

In appendix B, we show that the solution $\widehat{\mathbf{w}}$ to this optimisation problem is the first so called "generalized eigenvector" of $(\mathbf{R}_{SS}, \mathbf{R}_{NN})$.

The generalized eigenproblem

An arbitrarily scaled vector $\mathbf{w}_i$ is a *generalized eigenvector* (GEVec) for the ordered matrix pair $(\mathbf{R}_{SS}, \mathbf{R}_{NN})$ when the following holds:

$$\mathbf{R}_{SS}\mathbf{w}_i = \lambda_i\mathbf{R}_{NN}\mathbf{w}_i, \tag{8.6}$$

for some scalar λ_i, which is called the *generalized eigenvalue* (GEVal) corresponding to $\mathbf{w}_i$. The largest scalar λ_1 for which eq. (8.6) holds is the 'first' GEVal, and as mentioned before, the corresponding GEVec $\mathbf{w}_1$ is the solution $\widehat{\mathbf{w}}$ to the optimization problem eq. (8.5).

Since the 1960's, numerically stable algorithms exist that solve the generalized eigenproblem eq. (8.6) [56]. A specialized algorithm is applicable when the input matrices are symmetric – as is the case for $\mathbf{R}_{SS}$ and $\mathbf{R}_{NN}$. This algorithm (based on a Cholesky factorization and the classical QR-algorithm for ordinary eigenproblems) is implemented in the LAPACK software package (as `ssygv` and `dsygv`), and can be easily applied using e.g. the `eig` function from MATLAB, or the `eigh` function from SciPy's `linalg` module.

8.3 SWR detection performance

As mentioned in section 3.3, we divided the 34-minute long LFP recording into two sets. The first 60% was used as training data, to calculate the covariance matrices $\mathbf{R}_{SS}$ and $\mathbf{R}_{NN}$, and to calculate from these the optimal linear combination of channels $\widehat{\mathbf{w}}$, as described in the preceding sections. The remaining 40% was used to evaluate this filter $\widehat{\mathbf{w}}$, and to compare it to the state-of-the-art method (the single-channel online band-pass filter).

Figure 8.2A shows an excerpt of the test input signal, and the corresponding filter output envelopes (blue for state-of-the art method, orange for GEVec-based multichannel method. Filter outputs o_t are rectified to obtain envelopes $n_t = |o_t|$). Additional excerpts are shown in fig. C.4. The elements of $\widehat{\mathbf{w}}$ (i.e. the filter weights) are visualized in fig. 8.2B.

It is clear that the GEVec filter output indeed has high power during SWR events, as promised by the theoretical derivation. The filter weights and signal excerpts reveal that the GEVec output is composed mainly of a few channels in the stratum radiatum (channels 4-5-6 here), where they pick up the sharp waves. However, the filter output envelope is also high for sharp wave-like activity on these channels, without or with only very weak ripple activity in the higher channels: see figs. C.4a, C.4b and C.4d. This results in false positive detections.

We also notice some premature detections (figs. C.4a and C.4c), where the sharp wave – and thus also the GEVec filter output – already has high power before the corresponding ripple has started. The GEVec detection then happens before the start of the reference SWR-segment, which is based on ripple power. These early detections thus count (arguably unfairly so) as false positives, under the evaluation scheme that we use.

This effect, where the sharp wave is discernible before the ripple, results in faster detections when the detection *does* fall within the reference segment. When all 468 SWR events from the test set are analysed, we find a large improvement in detection latency: at thresholds where both methods detect 80% of these reference SWR segments, the median absolute detection latency drops from 24 ms for the state-of-the-art online band-pass filter to 12 ms for the GEVec-based filter. The relative detection latency drops by 32.5 percentage points, from 58.1% to 25.6%. Similar latency improvements are found for other recall and precision values: see fig. 8.3.

This strong improvement in detection latency trades off with an increase in false positives, as was already observed qualitatively. At the aforementioned sensitivity of 80%, the state-of-the-art method has a precision of 94%, whereas the GEVec-based method has a precision of only 63% (i.e. more than a third of detected events are classified as false

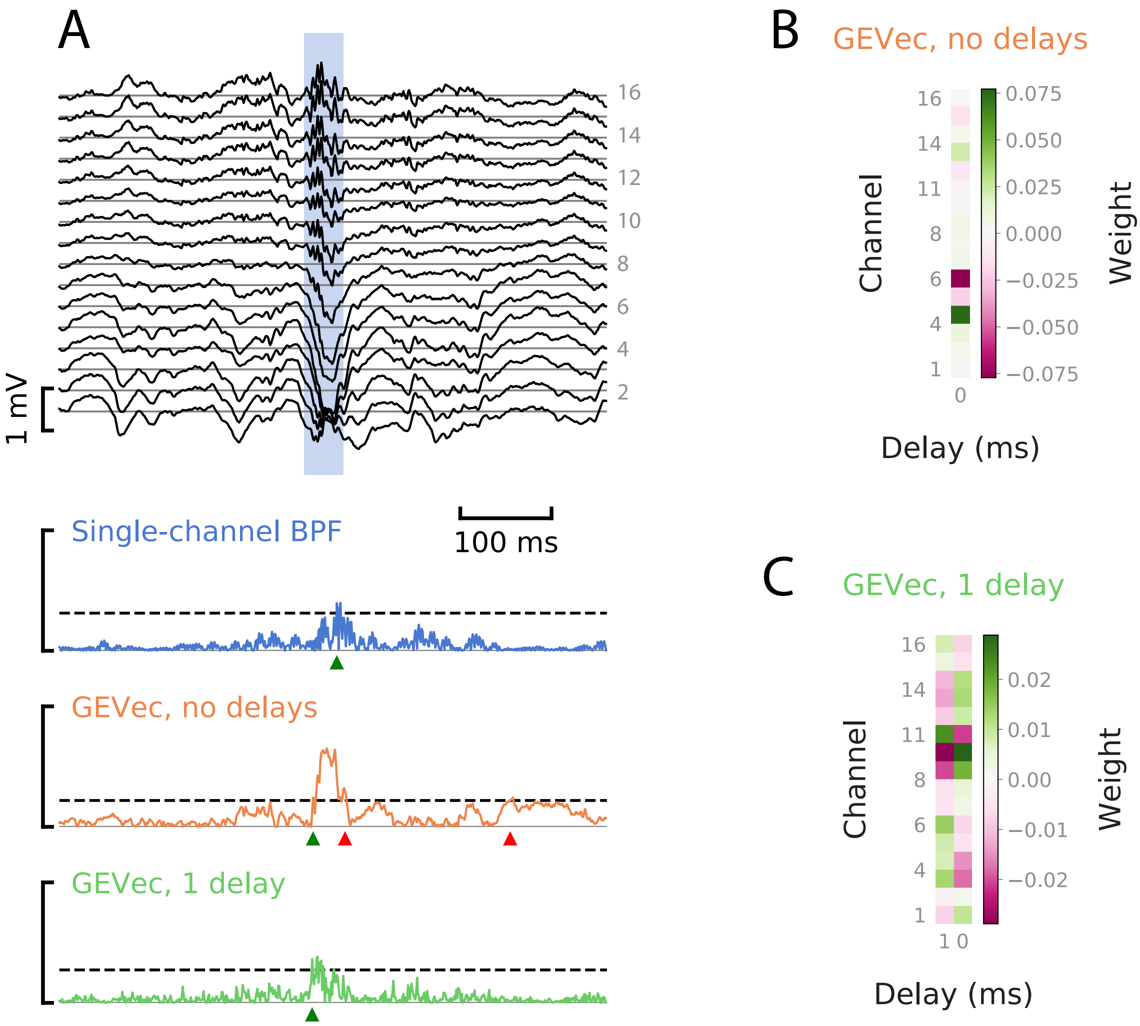


Figure 8.2: **Linear, SNR-maximising combinations of electrodes.**
A. Example input and output signals. *Top*: multi-channel LFP, $\mathbf{z}_t$. Light-blue vertical band: a reference SWR segment. *Bottom*: output envelopes n_t, for different filtering algorithms. Dashed horizontal lines: detection thresholds, chosen so that each algorithm reaches a recall value of 80%. Green triangles: correct detections. Red triangles: incorrect detections. Brackets indicate envelope range (min, max) over the entire test set.
B. Generalized eigenvector $\widehat{\mathbf{w}}$ (i.e. the weights of the multichannel filter), for a purely spatial filter.
C. Generalized eigenvector $\widehat{\mathbf{w}}$ for a spatiotemporal filter with a one-sample delay.

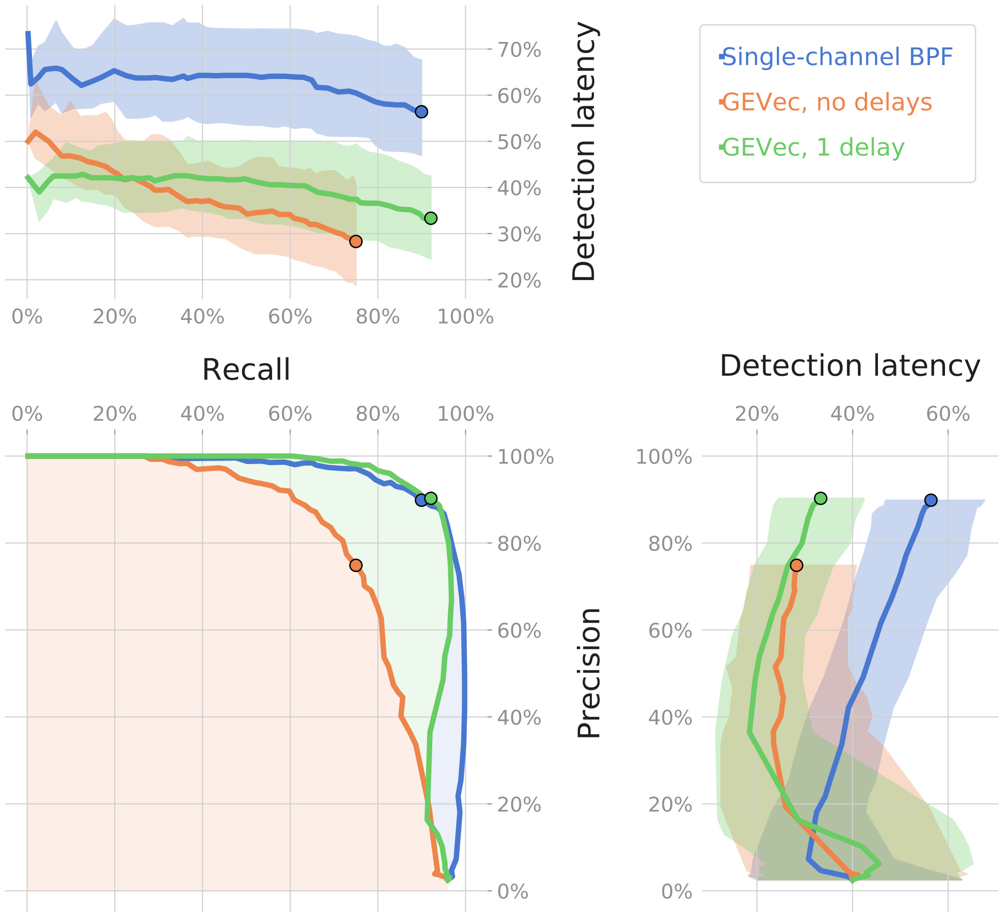


Figure 8.3: **Sensitivity, precision & latency tradeoffs,** for different linear filters & for a range of thresholds.
Each threshold setting for an algorithm corresponds to a point on the precision-recall curve in the bottom-left panel, and to a distribution of relative detection latencies. The median and interquartile range of this distribution are plotted in the top-left or bottom-right panel (as a point of the bold curve, and a slice of the shaded band, respectively).
These latency distribution plots are divided over two panels so that their entire range can be clearly visualized, both for the low recall – high precision regime as for the high recall – low precision regime. The cutoff is made at the point where recall equals precision (AKA the $\max F_1$ point), marked with shaded black circles.

positives). This strong decrease in precision is true over the entire PR-curve: see fig. 8.3.

8.4 Combining space and time

It is hardly surprising that the GEVec-based algorithm as described above cannot discern ripple activity (which is by definition a temporal pattern), as the algorithm is a purely *spatial* filter: at each timestep t, only current information from the different channels is used in calculating the output o_t, without incorporating temporal information from previous timesteps $t_p < t$.

The GEVec method can be easily adapted to also incorporate temporal information however, by defining a vector $\mathbf{z}_t^{\text{stack}} \in \mathbb{R}^{CP}$ which consists of stacked sample vectors (each consisting of C channels) from P different timesteps $t_p \leq t$. The linear weights $\mathbf{w}^{\text{stack}} \in \mathbb{R}^{CP}$ used to obtain the output signal $o_t = (\mathbf{w}^{\text{stack}})^T \mathbf{z}_t^{\text{stack}}$ are then calculated analogously to the purely spatial filter, i.e. as the first generalized eigenvector of the ordered pair $(\mathbf{R}_{SS}^{\text{stack}}, \mathbf{R}_{NN}^{\text{stack}})$, with both covariance matrices $\in \mathbb{R}^{CP \times CP}$.

Adding just one such delayed time step (i.e. $P = 2$) yields a major performance improvement (see fig. 8.3): the precision-recall curve shoots up to (and even slightly exceeds) the PR-curve of the state-of-the-art algorithm, while the latency improvements of the 'no delay' GEVec algorithm are mostly retained: at a sensitivity of 80%, the median absolute latency for the one delay GEVec filter is 15 ms, which is 9 ms faster than the state-of-the-art method (and 3 ms slower than the no delay GEVec filter). The relative latency is 36.6%: 21.5 percentage-points lower than the state-of-the-art (and 11 pp. higher than the no delay GEVec filter).

At this 80% recall mark, the one delay GEVec filter attains a precision of 97%, a 3% increase over the state-of-the-art. The full precision-recall-latency tradeoff and algorithm comparison is shown in fig. 8.3. Note that for very low thresholds, the PR-curve of the one-delay GEVec method is no longer concave: decreasing the threshold further yields more (not less) missed reference SWR segments. Figure 8.2C shows the GEVec $\mathbf{w}^{\text{stack}}$. Figure 8.2A and fig. C.4 show example output envelopes (green traces).

These results, particularly the visualized weights in fig. 8.2C, indicate that this one-delay GEVec filter utilizes spatiotemporal information about both the sharp wave and the ripple.

Choosing the number of delays

Adding more delays (fig. 8.4) improves detection accuracy even further – up to a peak *max* F_1 of 93% at about eleven delays. (This corresponds to eleven milliseconds, or approximately half a ripple phase). Detection latency also increases with increasing number of delays, but only slightly, always staying well below the state-of-the-art latency. Like the *max* F_1 score, latency stagnates after about eleven delays. Further, we note that the latency distributions of the GEVec-based detectors have a lower spread than those of the state-of-the-art detector.

There is thus a slight tradeoff to be made when choosing the number of delays for a GEVec-based detector: using more delays yields detectors that are more accurate, but

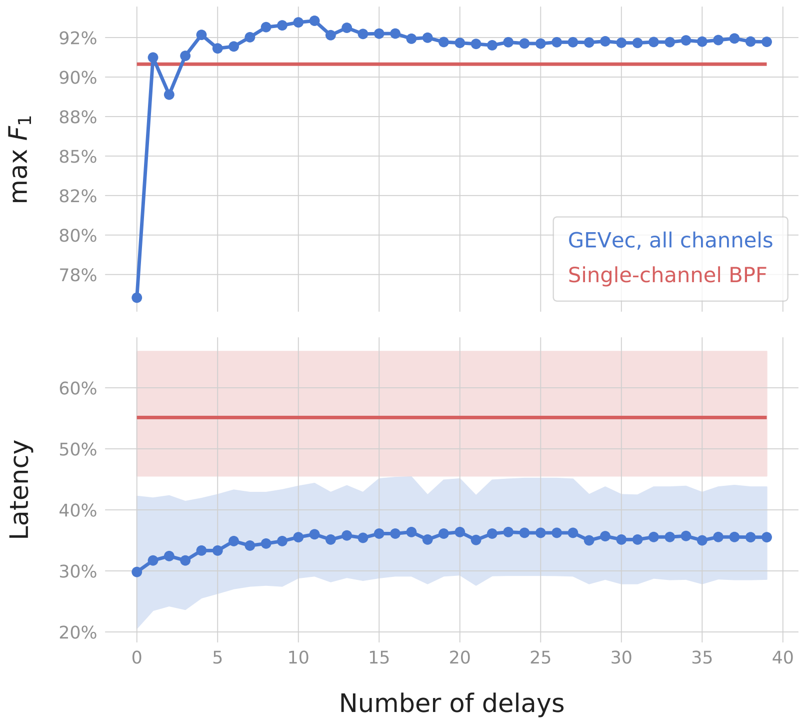


Figure 8.4: **Performance of the GEVec-based SWR detector, for different delay line lengths**. At the chosen 1000 Hz sampling rate, each delay corresponds to 1 ms. The red baseline is the state-of-the-art SWR detector. Detection latency is specified as a fraction of the duration of the corresponding SWR event, and is evaluated at the threshold where each detector reaches its maximum F_1-score. In the latency panel, bold lines and shaded areas indicate the median and the interquartile range of the latency distributions, respectively.

also slightly slower. Given that the decrease in speed is minor (about five percentage-points), it is reasonable to choose the amount of delays that maximizes detection accuracy. In this analysis, this optimal point is reached at an eleven milliseconds-long delay line.

Selecting channels

Many CA1 LFP recordings are not made with multichannel probes, but rather with one or more tetrodes. It is therefore relevant to ask how GEVec-based detectors perform on these types of recordings. We can approximate this setting with our current multichannel probe recording, by only using one or a few channels.

Each such selection of input channels, in combination with a certain number of delays, yields a different GEVec-based SWR detector. We evaluate and compare these detectors using the test data set, analyzing both accuracy (fig. 8.5) and latency (fig. 8.6).

We note some general trends. Using no extra delays results in low accuracy detectors, no matter which channels are included (top row of fig. 8.5). For most input channel combinations, the tradeoff in number of delays observed in the previous section is preserved: using more delays increases accuracy (up to a certain point), while also

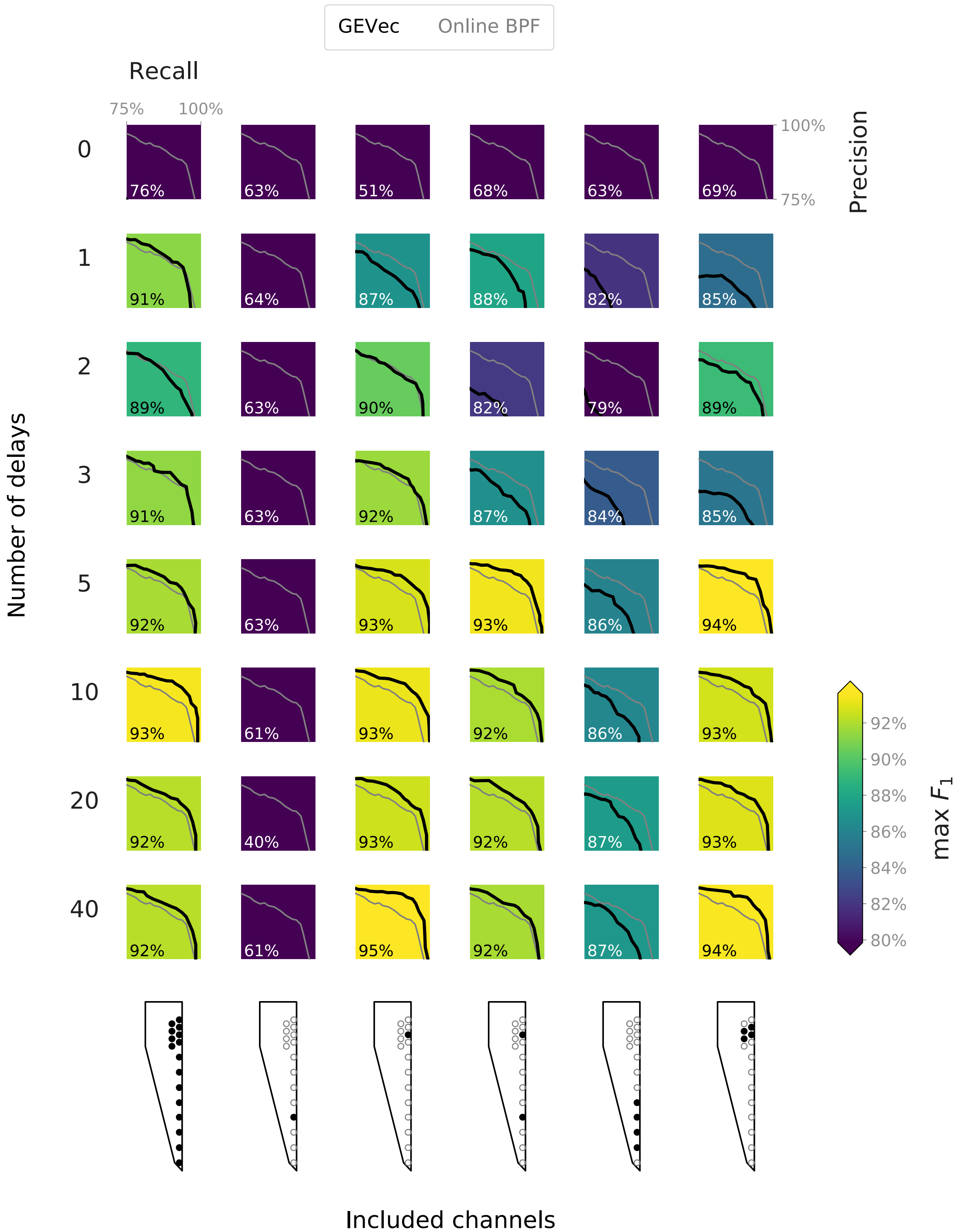


Figure 8.5: **Accuracy of GEVec-based SWR detectors,** for different combinations of active input channels and number of delays. Active input channels are marked in solid black on a schematic of the probe tip. (Height of schematic: 687 μm). The percentage and background color of each panel indicate the maximum F_1 value obtained for that GEVec-based SWR detector. Each panel includes the same baseline from the state-of-the-art SWR detector, in gray.

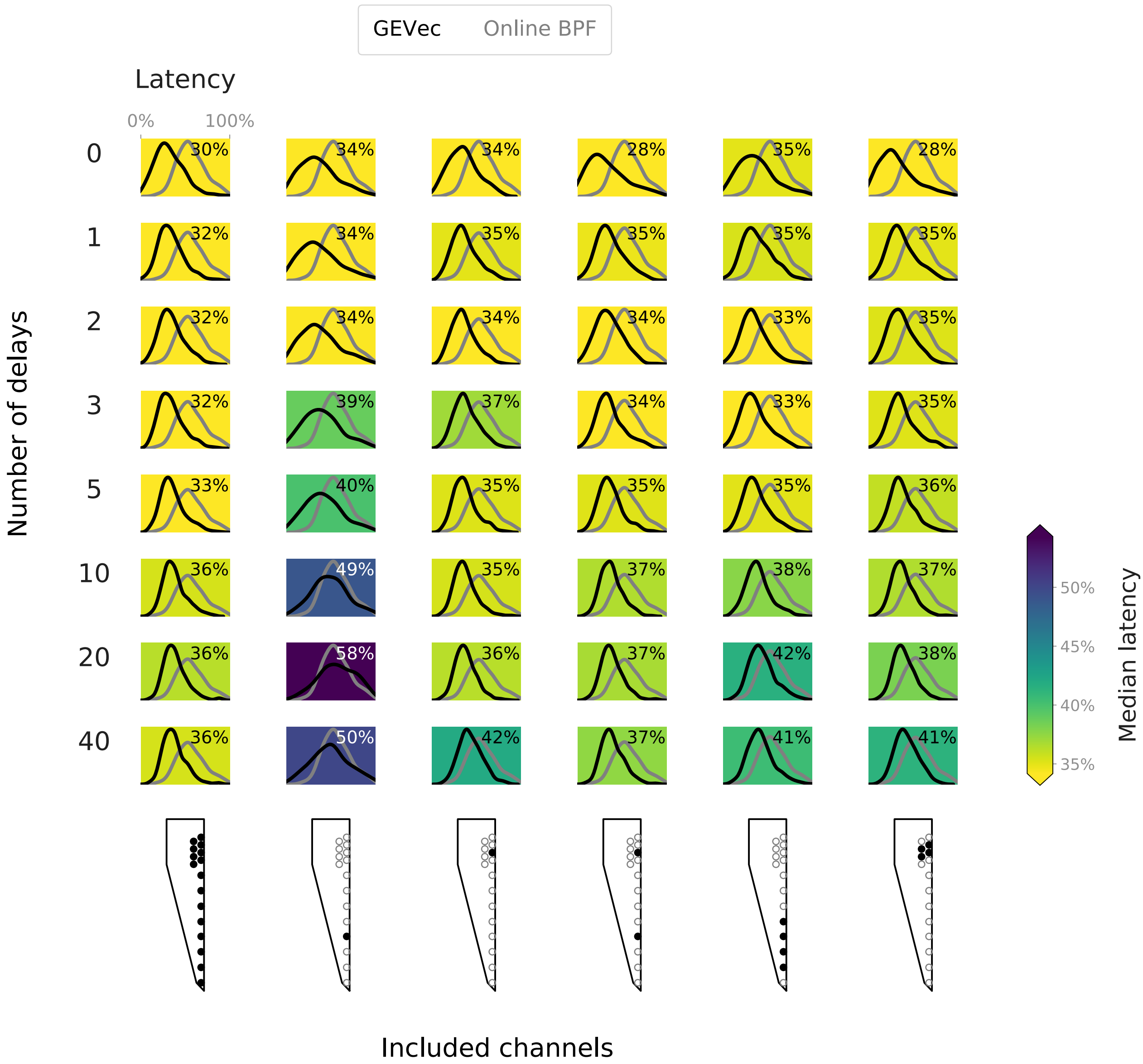


Figure 8.6: **Latency of GEVec-based SWR detectors**. See fig. 8.5 for legend.

increasing detection latency.

The weight visualizations of fig. 8.2 show a relatively high weight for the stratum radiatum / sharp wave channels. Using only one such channel as input results in low accuracy detectors however, no matter how many delays are used (second column of figs. 8.5 and 8.6). Using four such channels (second to last column) results in detectors with an accuracy approaching, but not reaching, that of the state-of-the art ripple-based detector when a sufficient number of delays is used. Interestingly, this decent accuracy is reached despite none of these four channels (numbers 2 to 5) displaying any noticeable ripple activity (fig. 8.2A).

Conversely, using one channel from the pyramidal cell layer (where ripples are observed) yields GEVec-based detectors on par with those that use all channels as input, both regarding accuracy and latency (third and first column of figs. 8.5 and 8.6).

This one channel GEVec-based detector thus also outperforms the "state-of-the-art" detector based on a band-pass filter. The state-of-the-art detector also uses only one channel as input, and is also a linear filter. This suggests that either 1) finding linear filter weights through 'machine learning' (the GEVec technique) yields better results than manual filter design; or 2) that the band-pass filter used as "state-of-the-art" detector was not designed optimally in the first place.

Adding a stratum radiatum / sharp wave channel to the single pyramidal cell layer channel (fourth column of figs. 8.5 and 8.6), or using a cluster of channels in the pyramidal cell layer (last column) does not improve performance over the one channel case.

Chapter 9

Nonlinear filtering

The SWR detectors discussed up to this point – both the band-pass and the GEVec-based filters – calculate their output as a linear combination of input samples. This chapter explores whether a nonlinear method to calculate the output signal can improve SWR detection performance.

9.1 Recurrent neural networks

The nonlinear filtering algorithm that we choose to explore is the recurrent neural network (RNN). RNN's have advanced the state-of-the-art in many sequence and time-series processing tasks [57]. (Examples include speech and language modelling [58], [59]; protein structure prediction [60]; and diagnosis prediction from intensive care unit time series [61]).[1] Relevant for SWR detection, RNN's can process multi-channel time-series, and calculating a new output sample has a relatively low cost[2], which is beneficial for real-time operation.

A recurrent neural network is a nonlinear dynamical system, driven by an input signal $\mathbf{z}_t \in \mathbb{R}^C$. In the case of SWR detection, $\mathbf{z}_t$ is the multichannel LFP recording, with C the number of channels. The RNN maintains an internal state vector $\mathbf{h}_t \in \mathbb{R}^M$, which is updated at each discrete time step t as:

$$\mathbf{h}_t = f(\mathbf{h}_{t-1}, \mathbf{z}_t). \tag{9.1}$$

f is a nonlinear function, parametrised by the coefficients of several affine transformations[3] (see section 9.2 for the complete update equations). The state vector $\mathbf{h}_t$ allows the RNN to maintain an efficient and task-relevant memory of past input samples [58].[4] We

[2]In contrast with very deep fully connected or convolutional neural networks.

[2]For the reader with an interest in neuroscience: RNNs have recently been used in various computational studies. Examples include [62] and [63], where an RNN was trained on a navigational task, after which the entries of the internal state vector of the RNN exhibited grid-cell-like activation patterns; [64], who combined reinforcement learning theory with RNNs to model reward seeking and value-based computations; [65], where RNNs were used to quantify and classify behaviour of the *C. Elegans* model organism; and [66], who modelled the fMRI hemodynamic response using an RNN. For an overview of artifical neural networks in general as models in neuroscience, see [67].

[3]Linear transformations (rotate, shear, scale), plus translations.

can read out the state vector of the RNN to obtain an output time series $n_t \in \mathbb{R}$, with

$$n_t = g(\mathbf{h}_t), \tag{9.2}$$

where g is again a parametrized nonlinear function.

By tuning the coefficients of the affine transformations in f and g in an offline training phase, we can change the behaviour of the RNN so that its output n_t for a given input signal $\mathbf{z}_t$ approaches a desired output y_t. RNN's are thus also a data-driven algorithm. (See section 9.3 for a description of this training phase, which involves the so called 'backpropagation through time' method). In the case of SWR detection, we can design a training signal y_t that is for examples 1 during SWR events in $\mathbf{z}_t$, and 0 otherwise (see section 8.1). If we avoid overfitting to the training data, we can use the trained RNN to detect SWR's in unseen signals $\mathbf{z}_t$. This is, in short, the proposed method of SWR detection in this chapter.

9.2 Update equations

The RNN type that we choose for SWR detection is the so called *gated recurrent units* RNN, or *GRU* [68]. It is closely related to the popular "long short-term memory" (LSTM) RNN [57], [69]. The update equations for the GRU are simpler than those of the LSTM however, while achieving comparable performance [70].

The state-update function f and the readout function g are defined as follows for a GRU RNN:

$$f: \ \mathbf{h}_t = \mathbf{u}_t \odot \mathbf{h}_{t-1} + (1 - \mathbf{u}_t) \odot \widetilde{\mathbf{h}}_t, \tag{9.3}$$

$$g: \ n_t = \sigma(\mathbf{w}_{hn}^T \mathbf{h}_t + b_n) \tag{9.4}$$

with

$$\widetilde{\mathbf{h}}_t = \tanh(\mathbf{r}_t \odot \mathbf{W}_{hh}\mathbf{h}_{t-1} + \mathbf{W}_{zh}\mathbf{z}_t + \mathbf{b}_h), \tag{9.5}$$

$$\mathbf{u}_t = \sigma(\mathbf{W}_{hu}\mathbf{h}_{t-1} + \mathbf{W}_{zu}\mathbf{z}_t + \mathbf{b}_u), \tag{9.6}$$

$$\mathbf{r}_t = \sigma(\mathbf{W}_{hr}\mathbf{h}_{t-1} + \mathbf{W}_{zr}\mathbf{z}_t + \mathbf{b}_r). \tag{9.7}$$

$\tanh$ is the hyperbolic tangent, and σ is the logistic sigmoid, defined as $\sigma(x) = 1/(1 + \exp(-x))$. They are applied point-wise, i.e. separately to each element of their argument vector. Similarly, "$\odot$" denotes point-wise multiplication. σ compresses the real number line to $(0, 1)$, while $\tanh(x) = 2\,\sigma(x) - 1$ maps it to $(-1, 1)$. Therefore, for this RNN, $\mathbf{h}_t \in (-1, 1)^M$ and the output $n_t \in (0, 1)$.

The coefficients in the matrices $\mathbf{W}\_$ and in the vectors $\mathbf{b}\_$ are called the *weights* and *biases* of the RNN, respectively.[5] They are parameters of the algorithm, determined a priori (i.e. before online SWR detection), through an optimisation procedure described in section 9.3.

[4] The vector $\mathbf{h}_t$ is also called the 'hidden state', and its M entries are sometimes called 'hidden units', 'neurons', or 'memory cells'.

[5] Often collectively referred to simply as the weights. The vector $\mathbf{w}_{hn}$ and the scalar b_n are also part of the weights and biases. They may be regarded as a one-row matrix and a one-dimensional vector, respectively.

From eq. (9.3), each element of the hidden state $\mathbf{h}_t$ is thus a weighted average of its existing value in $\mathbf{h}_{t-1}$, and a candidate new value in $\widetilde{\mathbf{h}}_t$. The weighting, in $\mathbf{u}_t$, depends on both the current input $\mathbf{z}_t$, and the full previous hidden state $\mathbf{h}_{t-1}$ (eq. (9.6)).

Similarly, each element of the candidate new hidden state $\widetilde{\mathbf{h}}_t$ is a function of both the current input $\mathbf{z}_t$, and all elements of the previous hidden state $\mathbf{h}_{t-1}$ (eq. (9.5)). A so called "reset" multiplier, in $\mathbf{r}_t$ (eq. (9.7)), determines how much the existing memory values in $\mathbf{h}_{t-1}$ influence the candidate new memory value, compared to the influence of the current input $\mathbf{z}_t$. (When an element of $\mathbf{r}_t$ is 0, the corresponding hidden unit in $\widetilde{\mathbf{h}}_t$ is only a function of the current input, and is thus decoupled from its past. The hidden unit in $\mathbf{h}_t$ can therefore be 'reset').

Finally, the output signal n_t is a linear projection of the hidden state, translated and squashed to be in $(0, 1)$ (eq. (9.4)).

The above equations (9.3) to (9.7) for the GRU RNN define a so called one-layer network.[6] An RNN with L layers has L corresponding hidden unit vectors $\mathbf{h}^{(1)}, \mathbf{h}^{(2)}, .., \mathbf{h}^{(L)}$, that may each have a different dimension. The update functions for these hidden unit vectors can be formulated as follows. If $f^{(1)}(\mathbf{h}^{(1)}_{t-1}, \mathbf{z}_t)$ is the function defined by eqs. (9.3) and (9.5) to (9.7), we can define similar functions $f^{(2)}(\mathbf{h}^{(2)}_{t-1}, \mathbf{h}^{(1)}_t)$, $f^{(3)}(\mathbf{h}^{(3)}_{t-1}, \mathbf{h}^{(2)}_t)$, and so forth. These functions all have the same form, but do not share weights and biases (i.e. each layer has separately learnable weights and biases). The output is then: $n_t = \sigma(\mathbf{w}^T_{hn}\mathbf{h}^{(L)}_t + b_n)$; i.e. a readout of the last layer.

9.3 Optimization

As described previously, we make use of a reference offline labelling of the training data, marking the segments where an SWR is present. This reference labelling can be used to create a training signal y_t, that we want the RNN output n_t to approach. As $n_t \in (0, 1)$ (see eq. (9.4)), we can define for example $y_t = 1$ during SWR events, and $y_t = 0$ outside them. Alternatively, we can construct y_t to be 1 only around the start of each reference SWR segment.

Given a target signal y_t, and the actual RNN output n_t, we can compare them to quantify how close the RNN output matches the target signal. We choose cross-entropy for this comparison. The so called *loss function* ℓ_t is then defined as:

$$\ell(n_t, y_t) = -y_t \log(n_t) - (1 - y_t) \log(1 - n_t) \tag{9.8}$$

The loss ℓ_t is lower whenever the RNN output is more similar to the target signal. The mean loss $\langle \ell_t \rangle$ over some dataset can thus be used as a proxy for how useful the RNN output n_t is for SWR detection. Just as in chapter 8, we want to find the weights w_i (the elements of the matrices $\mathbf{W}\_$ and the vectors $\mathbf{b}\_$ from eqs. (9.3) and (9.4)) that minimize this expected loss $\langle \ell_t \rangle$. Unlike in chapter 8, there is no known way to find the global minimum of this function.

[6] That is, one layer per timestep. A recurrent neural network can also be regarded as a very deep feedforward neural network in which all the layers share the same weights.

Instead, a form of gradient descent is used to iteratively and stochastically approach a local minimum of $\langle \ell_t \rangle$. We divided the training data into 300 ms long chunks. For each such chunk, the total loss $L = \sum_{300\text{ms}} \ell_t$ is calculated. Then, the partial derivatives $\partial L / \partial w_{i,t}$ of this loss are calculated, for each parameter w_i in the RNN, and for each timestep t in the chunk. This calculation of the gradient of L is done through the so called *backpropagation* algorithm [71], which is an application of the chain rule from calculus. In this case we apply so called "backpropagation through time" (BPTT) – a common way to train RNN's [59]. Such derivative calculations are often done through an automatic differentiation program. In our case, we used the `PyTorch` library [72].

For each parameter w_i in the RNN, the BPTT algorithm yields a set of partial derivatives $\{ \partial L / \partial w_{i,t_1}, \partial L / \partial w_{i,t_2}, .. \}$ that each tell how much and in which direction the chunk loss L changes when the parameter w_i is increased at timestep t. Taking the mean of this set of partial derivatives over all timesteps in the chunk yields a value $\partial L / \partial w_i$ that can be used to tell how the parameter w_i should be adapted to decrease the loss L:

$$w_i \leftarrow w_i - \eta \frac{\partial L}{\partial w_i} \tag{9.9}$$

(i.e. if $\partial L / \partial w_i > 0$ then the loss would increase by increasing w_i; so decrease w_i). Doing this for all parameters w_i of the RNN results in a complete so called gradient step. η determines the step size, and is called the *learning rate*. Having a separate and adaptive learning rate $\eta_{i,t}$ for each parameter has been shown to greatly increase convergence speed of stochastic gradient descent. We used the AdaMax optimization algorithm [73] to update our RNN parameters with such adaptive learning rates, using the default hyperparameters as suggested in the paper.

9.4 Regularization

Repeating this procedure for all chunks, and for multiple passes over the training data, yields a trained RNN. The danger then exists that the network is *overfit* to the training data; i.e. it achieves a low loss on the training data, but it performs badly on unseen data. We avoided this with so called *early stopping* on a validation set.

As described in section 3.3, the first 20 minutes of the 34 minutes-long recording were used as training data for data-driven online algorithms. For the RNN, these 20 minutes were further split into a part for training proper (the first 15.6 minutes), and a part for validation (the final 4.4 minutes). The RNN was trained for 50 passes over the proper training part. After each training pass (or *epoch*), the loss on the validation set was calculated, as an estimate of how the RNN would perform on unseen data. The RNN where the validation loss was lowest was then chosen for the final evaluation of online SWR detection performance on the held-out test data. See fig. 9.1 for an example validation loss curve over training time.

9.5 Results for SWR detection

The first GRU network that we analyzed used all 16 input LFP channels, had 2 layers of 40 hidden units each, and was trained on a target function y_t that is 1 during the entire duration of each reference SWR segment, and 0 elsewhere. Figure 9.1 shows the

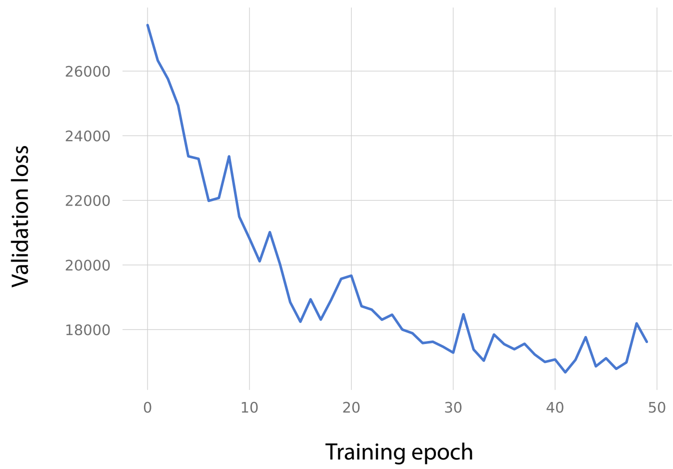


Figure 9.1: **Regularization by early stopping**. Evolution of total loss on the validation data during training of an RNN.

evolution of the validation loss while training this RNN. Figure 9.2 shows example output envelopes n_t.

We notice that the RNN output matches the target block-shaped function y_t relatively well. Together with the validation loss curve, it seems that the target function was relatively well learnable.

Secondly, the RNN clearly uses more information than the ripple oscillation only to generate its output. In the right panel of fig. 9.2, the RNN outputs a positive with high confidence. The low frequency LFP over all channels indeed has the profile of an SWR; but the ripple is almost non-existent, as can be seen in the band-pass and GEVec filter outputs. This detection of the RNN is thus classified as a false positive. Additional examples of this phenomenon are shown in fig. C.5.

This phenomenon, where the RNN seemingly uses the low frequency spatiotemporal pattern of SWR's but does not require clear ripples, leads to a low precision for the RNN detector, under our evaluation scheme. Figure 9.3 compares the precision-recall curve of this RNN detector with those of previous online SWR detectors, as well as comparing their detection latencies.

The detection thresholds in fig. 9.2 are chosen at the maximal F_1 point for each output envelope. Because of the relatively low precision (high false-positive rate) of the RNN, this threshold is set high for the RNN envelope. This leads often to relatively late detections; see e.g. the rightmost detection in fig. 9.2.

To investigate the influence of input channels, we trained a GRU-RNN that only used a single input channel (namely the pyramidal cell layer channel that is also used as input for the band-pass filters). The resulting RNN behaves very much like the band-pass filter (fig. C.8). It does not show the non-ripple false positive detections of the all-channel RNN. Its detection performance approaches, but is not as good as the baseline online band-pass filter (fig. C.7).

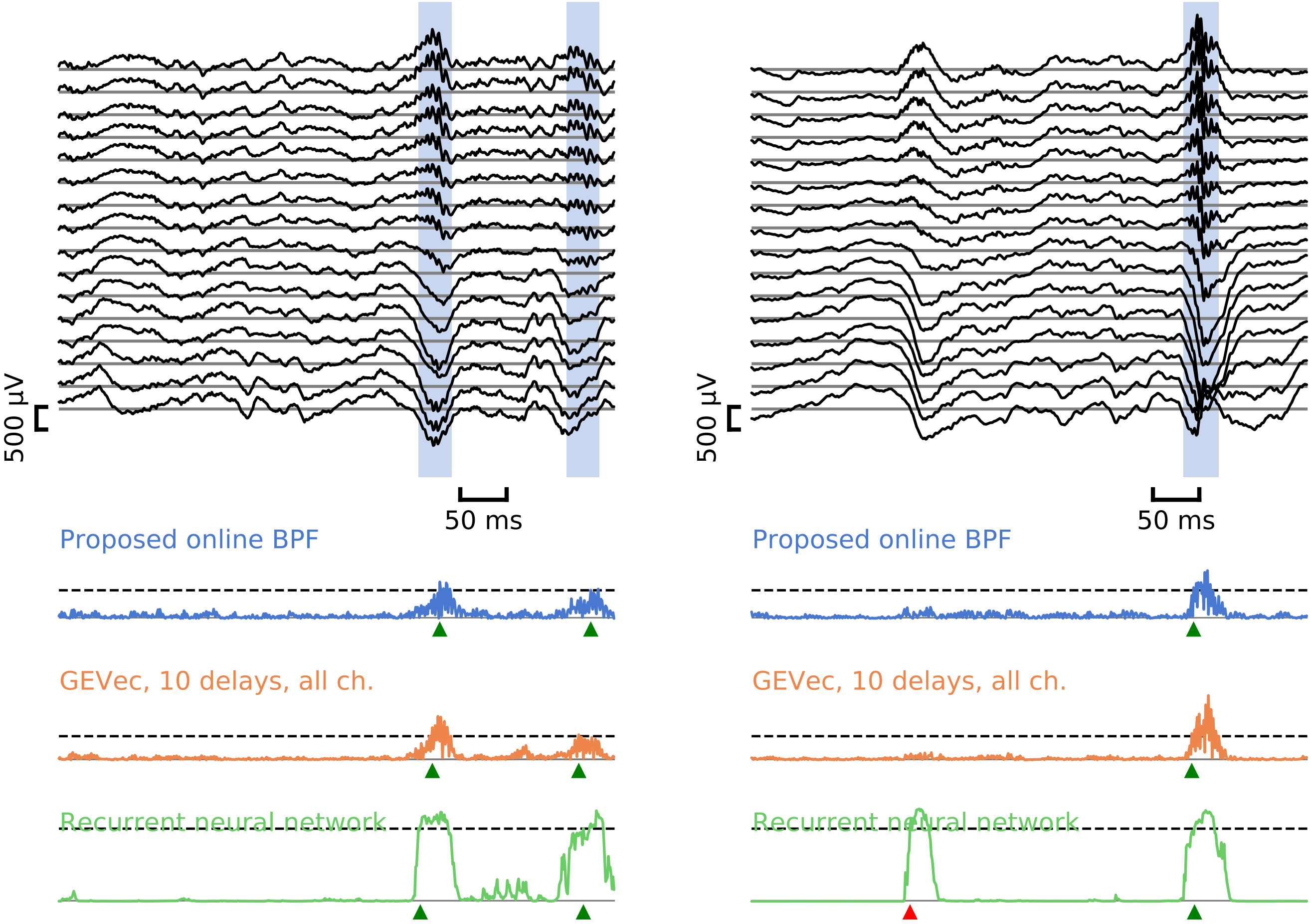


Figure 9.2: **RNN output envelopes**. See fig. 7.4 for legend.

A second net was trained on this single channel, this time with only one hidden layer, of 25 hidden units. Its performance was slightly worse than the two-layer, 40 hidden units per layer RNN (fig. C.7).

Finally, we experimented with a different target function; Namely a y_t that is 1 only around the start of reference SWR segments. This did not result in good performance for all-channel, two-layer, 40 hidden unit RNN's (fig. C.6).

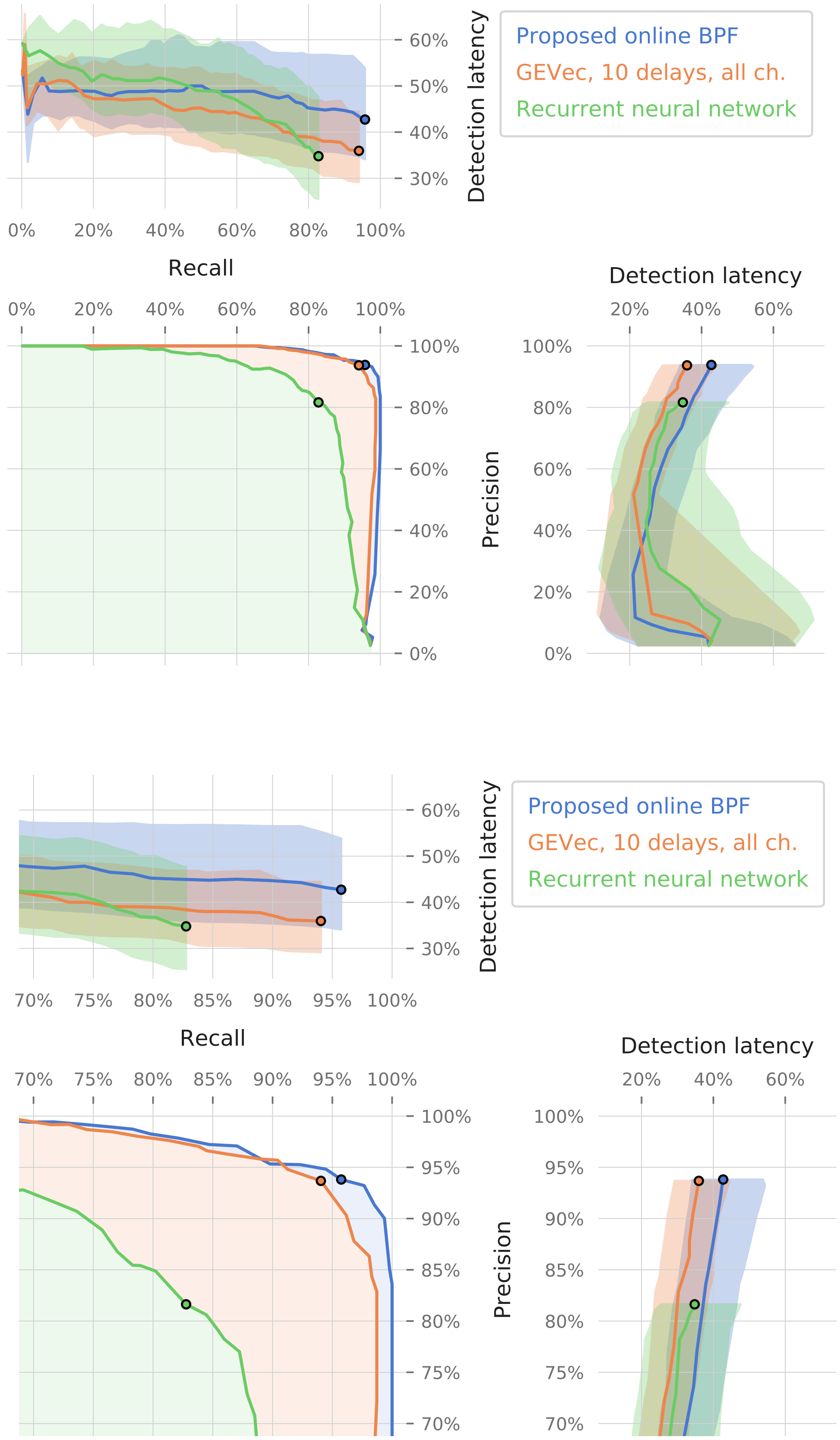


Figure 9.3: **Online SWR detection performance of an RNN**. See fig. 7.5 for legend. *Top*: Full PR-curve. *Bottom*: zoom-in on high-recall/high-precision region.

Chapter 10

Conclusions

Recall the problem statement from the introduction: Can we find an algorithm that detects sharp wave-ripples with less latency than the existing online algorithms, while being at least equally sensitive and precise? And, prompted by the development of neural probes: Are spatially distributed multichannel recordings advantageous for this task?

For the new SWR detection algorithm that we investigated, the answer to both questions seems to be negative; see fig. 10.1. We applied for the first time a data-driven, multichannel algorithm to SWR detection. This algorithm, based on the generalized eigenvector decomposition, achieves performance roughly on par with the best state-of-the-art algorithm; but not surpassing it. Using the new algorithm with multiple spatially distributed input channels yields similar performance as when it is used with only a single channel.

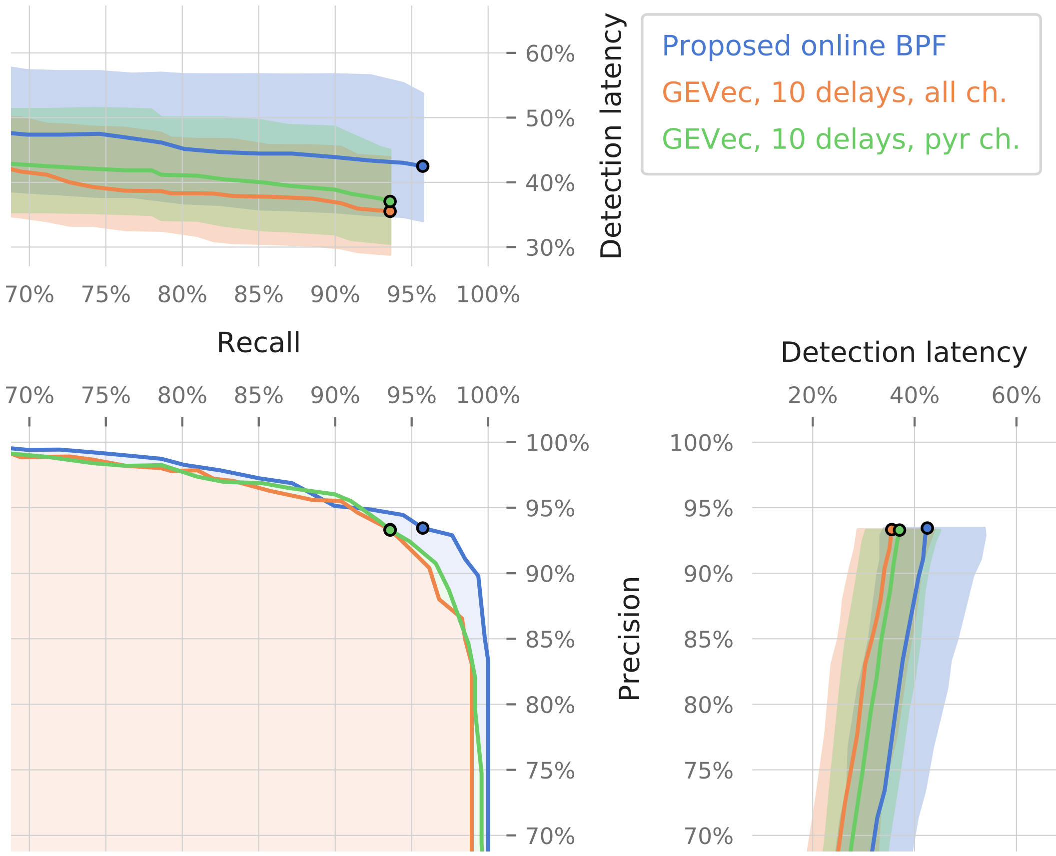


Figure 10.1: **Performance comparison of the best SWR detectors**. Legend as in fig. 7.5. "Proposed online BPF" is the best representative of the current generation of SWR detectors (band-pass filters on a pyramidal cell layer channel; see chapter 7). "GEVec" is the newly investigated, data-driven algorithm. Using this algorithm with about ten delay samples (corresponding to 10 ms at $f_s = 1000$ Hz) yielded optimal performance (see chapter 8). Both a version of the GEVec algorithm using multiple, spatially distributed input channels ("all ch."), and a version using the same, single input channel as the band-pass filter ("pyr ch.") are presented here.
The GEVec-based detectors and the band-pass filter have very similar performance. The GEVec-based detectors have very slightly lower latency (Difference in medians of relative latency is $\pm$ 6 percentage-points. Difference in medians of absolute latency is 2 milliseconds). In the high recall regime (at $R > 93\%$), the band-pass filter is slightly more precise (less false positive detections). The single-channel and the multichannel GEVec-based detectors show near identical performance.

Appendices

Appendix A

SWR detection in the literature

We gathered a representative sample of research papers that discuss sharp wave-ripples (and/or other hippocampal oscillations). Different studies define these oscillations with different frequency bands – these differences are listed in table A.1. Appendices A.1 and A.2 quote papers that use SWR detection; specifically their description of the SWR detection procedure. Emphasis is added.

Source	Theta (Hz)	High gamma (Hz)	Ripple (Hz)
Nádasdy et al. 1999 [50]			150 – 250
Csicsvari et al. 2000 [51]			80 – 250
Behrens et al. 2005 [53]			40 – 400
O'Keefe 2007 [74]	6 – 12	30 – 100	100 – 200
Girardeau et al. 2009 [40]			100 – 200
Ego-Stengel et al. 2009 [39]			100 – 400
Jadhav et al. 2012 [41], online			100 – 400
Jadhav et al. 2012 [41], offline			150 – 250
Buzsáki 2015 [29]	6 – 10	100+	110 – 200
Colgin 2016 [75]	6 – 12	60 – 100	150 – 200
Sadowski et al. 2016 [54]			120 – 250
Talakoub et al. 2016 [44]			80 – 150
Eichenbaum 2017 [76]	4 – 12	80 – 140	
Dutta et al. 2018 [52]			150 – 250
Ólafsdóttir et al. 2018 [31]	6 – 12		140 – 250
`fklab`	6 – 12	60 – 140	140 – 225
L2 recording	5 – 10		100 – 200

Table A.1: **Frequency bands of hippocampal LFP events**, according to different sources (both original research papers and literature reviews). When the source does not make a distinction between high and low gamma, the full gamma range is given. Note that Behrens et al. considered artificially induced SWR's in ex-vivo hippocampus slices. Talakoub et al. studied macaque monkeys; other primary research papers studied rats. '`fklab`' refers to the default frequency bands used in the data-analysis software used in the Kloosterman lab. The last row refers to the dataset analysed in this thesis.

A.1 Offline detection algorithms

Nádasdy et al. 1999

> "For the extraction of sharp-wave (SPW) ripple events during sleep, the wide-band recorded data were bandpass filtered digitally (**150–250 Hz**). The power (**root mean square**) of the filtered signal was calculated, and the beginning, peak, and end of individual ripple episodes were determined. The threshold for ripple detection was set to **7 SDs above the background mean** power (Csicsvari et al., 1999 [77])." [50]

Csicsvari et al. 2000

> "*Detection of SPW-Associated Fast Ripples*: The procedures described here were identical to those described earlier (Csicsvari et al., 1999b [78]). The wide-band (1–5 kHz) recorded data was digitally band-pass filtered (**80–250 Hz**), and the power (**root-mean-square**) of the filtered signal was calculated for each electrode. The mean and standard deviation (SD) of the power signal were calculated to determine the detection threshold. Oscillatory epochs with a power of **one** or more **SD above the mean** were detected. The beginning and the end of oscillatory epochs were marked at points where the power fell below **0.5 SD**. Theta periods, detected by using the theta-delta power ratio (Csicsvari et al., 1999a [77]), were excluded from the analysis." [51]

Behrens et al. 2005

> "For ripple detection, raw data were filtered with a Spike 2 software band-pass filter of **40–400 Hz** (threshold: **4–6** times the s.d. of **eventless baseline noise**). For sharp wave detection, recordings were low-pass filtered at 20 Hz." [53]

Girardeau et al. 2009

> "Offline ripple detection was performed by band-pass filtering (**100–200 Hz**), **squaring and normalizing**, then thresholding the field potential recorded in CA1 pyramidal layer. Ripples were defined as events peaking at **>5 standard deviations** and lasting **<100 ms**." [40]

Jadhav et al. 2012

> "SWRs were detected during post-hoc analysis as described previously (11, 23). Raw LFPs recorded from the tetrodes used for online SWR detection were filtered between **150 – 250 Hz** and the SWR envelope was determined using a **Hilbert transform**. The envelope was **smoothed with a Gaussian with a s.d. of 4 ms and a width of 32 ms**. SWRs were defined as contiguous periods when the smoothed SWR envelope stayed **above 3 s.d.** of the mean [sic] for **at least 15 ms** on at least one tetrode." [41]

Sadowski et al. 2016

> "Ripples were detected offline in the LFP recorded on one CA1 channel. Raw LFP signal was filtered between **120 and 250 Hz**, and deflections in the ripple **power envelope** greater than **5 SDs from the mean** were classified as ripple events. Ripple start times were defined locally as when ripple power exceeded **2 SDs**. Samples of raw LFP and detected ripple times were compared manually to verify detection fidelity." [54]

Dutta et al. 2018

> "Post-recording, ripple events were defined on tetrodes that displayed characteristics of the CA1 area of the hippocampus. Specifically, the recorded LFP in one of the channels of the selected tetrode (same one subject to online detection for our realtime analysis) first had a digital reference subtracted away. This signal was then LFP band filtered with a 400 Hz low-pass infinite impulse response (IIR) filter (from Trodes). Afterwards the signal was decimated and ripple band filtered (**150–250 Hz**) with a **25 tap** finite impulse response (**FIR**) filter. Ripple band filtering was done using a forward and a time-reversed path, resulting in a net **zero group delay** (time shift from filtering). The instantaneous power of the ripple band filtered signal was then calculated via a **Hilbert Transform** and further **smoothened with a Gaussian kernel** with a **4 ms standard deviation**. Ripple events were detected as times when z-score of the smoothened power signal signal exceeded a threshold of **3 z-units** for **at least 15 ms**. The canonical ripple epochs were defined as the time points from which the processed signal **returned down to the mean before and after threshold crossings** [79], [80].
>
> In cases when multiple electrodes (typically channels on different tetrodes) are available for ripple detection, a different canonical definition is required. Ripples were initially defined using as above for each electrode. A canonical multichannel ripple was defined as one which is simultaneously detected on each electrode (two in our analysis). The multichannel ripple epoch is defined as the union of the detected single-channel ripple epochs, i.e., the start of the earliest ripple detected and to end with bound of last ripple detected. As such, we obtain a conservative ripple detection latency estimate while covering the entire span of the time the LFP is in a high ripple band power state. We reanalyzed our data with the canonical ripples being defined on different channels and tetrodes with a 300 tap bandpass FIR filter allowing 1% "ripple" in the passband with -30 dB suppression in the stopband but our results and subsequent conclusions remained consistent." [52]

A.2 Online detection algorithms

Girardeau et al. 2009

> "The onset of SPW-Rs was detected online by filtering the signal in the ripple-band and thresholding it. [..] Brain signals were preamplified (..), acquired and **digitized** using two synchronized Power1401 systems (CED, Cambridge, UK). [..] In both cases (test and control) the number of stimulations was **limited to 5 per second**." [40]

Ego-Stengel et al. 2009

> "We selected one tetrode in CA1, for which the LFP signal exhibited ripple events of large amplitude, for online ripple detection. The LFP was amplified and filtered online in the ripple band by an **8th-order Butterworth lowpass filter at 400 Hz** followed by an **8th-order Butterworth highpass filter at 100 Hz** (KrohnHite 3384 analog filters, total gain 10,000). A threshold-crossing detector (FHC Window Discriminator) was used to generate TTL pulses when the ripple amplitude exceeded a value adjusted manually by the experimenter on the first experimental day for each rat (0.1 ± 0.02 mV). These pulses triggered isolated stimulation units via a computer- controlled burst generator with a preset 1-ms delay [..] A **2-s recovery period** was forced after any stimulation burst before the next stimulation could be triggered." [39]

Jadhav et al. 2012

> "We disrupted awake hippocampal SWRs [..] with the use of an online feedback system similar to that used in previous studies that disrupted SWRs during post-behavior sleep [39], [40]. SWRs in CA1 were detected by monitoring power in the ripple band simultaneously across multiple tetrodes. [..] This [detection-triggered stimulation] terminated the ripple oscillation **within 25 ms of SWR onset** and transiently inhibited CA1 spiking [..].
> We recorded continuous local field potentials (LFP, filtered 0.5-400 Hz and sampled at 1500 Hz) from all tetrodes (one channel was chosen from each tetrode for LFP recording). [..]
> *Real-time detection algorithm*. Field potential signals from the 5-6 tetrodes chosen for online detection were broadly filtered in the ripple band (**20 tap band-pass IIR filter, 100-400 Hz**). In order to establish a disruption threshold, we calculated smoothed values of the mean and s.d. of the absolute value of the filtered LFP signal on each tetrode being used for detection using an iterative procedure:[1]
>
> $$\mu_t^{\text{est}} = \mu_{t-1}^{\text{est}} \frac{N^{\text{smooth}} - 1}{N^{\text{smooth}}} + \frac{|o_t|}{N^{\text{smooth}}}$$
> $$\sigma_t^{\text{est}} = \sigma_{t-1}^{\text{est}} \frac{N^{\text{smooth}} - 1}{N^{\text{smooth}}} + \frac{\left||o_t| - \mu_{t-1}^{\text{est}}\right|}{N^{\text{smooth}}}$$
>
> Here μ^{est} and σ^{est} are the estimated mean and s.d. of the absolute value of the filtered LFP, o, and N^{smooth} is the number of samples for smoothing (typically 10000). We allowed these estimates to stabilize before each run session. To generate a smoothened

estimate of the envelope (n^{est}) of the filtered LFP, we used the following iterative estimator:[2]

$$n_t^{\text{est}} = (1 - g_{t-1})n_{t-1}^{\text{est}} + g_{t-1}|o_t|$$

To allow for rapid detection of increases in power, we used a larger gain g for periods when the envelope was increasing: when the envelope was decreasing ($|o| \leq n^{\text{est}}$), $g = 0.2$ when the envelope was increasing, we used a moving average of the last 19 values of g and 1.2:

$$g_t = \begin{cases} 0.2, & \text{if} \quad n_{t-1}^{\text{est}} \geq |o| \\ \langle g_{t-20}, g_{t-19}, \ldots, g_{t-1}, 1.2 \rangle, & \text{otherwise} \end{cases}$$

The threshold for disruption was set to **4-6 s.d. above the mean**. To prevent false-positives, vHC stimulation was triggered only when the smoothed LFP envelope exceeded threshold on at least 2 tetrodes. Stimulation rate was limited to a maximum of 4 Hz by enforcing a **lock-out period of 250 ms** after each stimulation event." [41]

Talakoub et al. 2016

"Local field potentials were [..] sampled at **32 kHz**. [..] Selecting the electrode channel with the highest-amplitude ripple activity [..]. In this study, we set the threshold to **6-sd** of the band activity, a value similar to rodent interruption studies. The 6-sd threshold was estimated based on the average and variance of the ripple band from earlier recordings which were consistent over days in these experiments (9.5 ± 0.1 µV mean ± SEM).
[..] the signals were bandpass filtered using a custom-designed finite impulse response (**FIR**) filter with **512 taps** (**16 ms delay**, which is about one cycle of ripple activity).[3] Activities slower than **80 Hz** or faster than **150 Hz** are suppressed more than 20 dB (Fig. [..]). " [44]

Dutta et al. 2018

"Like our canonical ripple detections, the realtime or online detection algorithm is comprised of single channel and multichannel modalities. The single channel case performs realtime reference subtraction from the LFP (filtered in the same way as the offline case) and **decimation from 30 kHz to 3 kHz** as in the canonical detection case. However, the difference between online and canonical detections begins from ripple band filtering. The decimated signal is **filtered to the ripple band with a 30 tap FIR filter**. In order to perform a realtime instantaneous power estimation and smoothing,

[1] Equations rearranged for readability, and variable names substituted to match those used in this thesis. Note that μ^{est} and σ^{est} are exponentially weighted moving averages of respectively $|o|$ and $\big||o| - \mu^{\text{est}}\big|$.

[2] Like μ^{est}, n^{est} is also an exponentially weighted moving average of $|o|$, but with a much larger weight for the most recent value of $|o|$. Testing the estimator of Jadhav et al. on our data, we find a weight g that oscilates between 0.2 and values between 0.27 and 0.30.

[3] An FIR filter has a constant group delay of (# taps $-$ 1)$/(2f_s)$. A 512 tap FIR filter at $f_s = 32$ kHz should therefore have a group delay of 8 ms (and not the reported 16 ms). Or, conversely, an FIR filter with a group delay of 16 ms should have $\approx$ 1024 taps at $f_s = 32$ kHz (and not the reported 512 taps).

the realtime algorithm computes the **absolute value of the ripple band filtered signal** and **further filters it by a 33 tap 50 Hz low-pass FIR** (instead of a Hilbert transform followed by Gaussian kernel smoothing). These filters cause an intrinsic sample delay from the offline case (≈10.167 ms in our case). It is worth noting that the number of filter taps as well as filter types were determined by analyzing algorithmic delay and detection accuracy based on metrics described in the Data Analysis subsection. To normalize detection thresholds in the realtime case, the mean and standard deviation of the smoothed envelope are estimated over a 20 minute training period. In two ≈90 minute sleep box recording sessions, when we sampled 20 minute time intervals at random (N=1000), the resulting mean and standard deviation were within 5% of the values for the entire sessions. This length of time and subsequent error in parameter estimation likely depends on the behavioral and/or sleep state of the animal — in our experimental recordings, animals were contained in a sleep box. realtime detections are then triggered when the envelope crosses a threshold defined as α standard deviations above the mean (threshold = $\alpha \cdot \sigma + \mu$) or α z-units. Following a detection, there is a **200 ms lockout period** where we ignore any further threshold crossings (i.e., to avoid stimulation artifacts). **Additionally**, we impose a hard **limit on the number of detections per second** (set to **three** during the experiments in this work)." [52]

Appendix B

Generalized eigenvectors maximize signal-to-noise

In this appendix, we show that the weight vector that maximizes the ratio of signal to noise variance (i.e. the solution $\widehat{\mathbf{w}}$ to eq. (8.5)) is equivalent to the first generalized eigenvector $\mathbf{w}_1$ of the ordered pair of covariance matrices $(\mathbf{R}_{SS}, \mathbf{R}_{NN})$ (as defined in section 8.2).

This ratio of variances in eq. (8.5) is a quotient of quadratic forms, namely the so called "generalized Rayleigh quotient" of $(\mathbf{R}_{SS}, \mathbf{R}_{NN})$.

Formally, the generalized Rayleigh quotient of a non-zero vector $\mathbf{w} \in \mathbb{R}^N$ and the ordered, symmetric matrix pair $(\mathbf{A}, \mathbf{B})$ is the scalar $r(\mathbf{w})$ defined as:

$$r(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{A} \mathbf{w}}{\mathbf{w}^T \mathbf{B} \mathbf{w}} \tag{B.1}$$

Recall that a vector $\mathbf{w}_i$ is a generalized eigenvector of $(\mathbf{A}, \mathbf{B})$, with corresponding generalized eigenvalue λ_i, if

$$\mathbf{A}\mathbf{w}_i = \lambda_i \mathbf{B}\mathbf{w}_i \tag{B.2}$$

We must then prove the following:

B.1 Theorem

The generalized eigenvector $\mathbf{w}_1$ corresponding to the largest generalized eigenvalue λ_1 of $(\mathbf{A}, \mathbf{B})$, is also the vector $\widehat{\mathbf{w}}$ that maximises the generalized Rayleigh quotient $r(\mathbf{w})$ of $(\mathbf{A}, \mathbf{B})$.

B.2 Proof

As a first step, we will show that if $\widehat{\mathbf{w}}$ is the maximum of $r(\mathbf{w})$, that it is indeed an eigenvector of $(\mathbf{A}, \mathbf{B})$. In the second step, we will show that the largest eigenvalue λ_1 of $(\mathbf{A}, \mathbf{B})$ corresponds to the maximum of $r(\mathbf{w})$.

If $\widehat{\mathbf{w}}$ is a maximum of $r(\mathbf{w})$, then

$$\boldsymbol{\nabla} r(\widehat{\mathbf{w}}) = \mathbf{0}. \tag{B.3}$$

Working out the partial derivatives that comprise the gradient of $r(\mathbf{w})$, we find:

$$\boldsymbol{\nabla} r(\mathbf{w}) = \frac{2\mathbf{A}\mathbf{w}\left(\mathbf{w}^T\mathbf{B}\mathbf{w}\right) - 2\mathbf{B}\mathbf{w}\left(\mathbf{w}^T\mathbf{A}\mathbf{w}\right)}{\left(\mathbf{w}^T\mathbf{B}\mathbf{w}\right)^2}$$

With eq. (B.3), we then have the following condition for our maximising vector $\widehat{\mathbf{w}}$:

$$2\mathbf{A}\widehat{\mathbf{w}}\left(\widehat{\mathbf{w}}^T\mathbf{B}\widehat{\mathbf{w}}\right) = 2\mathbf{B}\widehat{\mathbf{w}}\left(\widehat{\mathbf{w}}^T\mathbf{A}\widehat{\mathbf{w}}\right)$$

or

$$\mathbf{A}\widehat{\mathbf{w}} = \frac{\widehat{\mathbf{w}}^T\mathbf{A}\widehat{\mathbf{w}}}{\widehat{\mathbf{w}}^T\mathbf{B}\widehat{\mathbf{w}}}\,\mathbf{B}\widehat{\mathbf{w}}$$

$$\mathbf{A}\widehat{\mathbf{w}} = r(\widehat{\mathbf{w}})\,\mathbf{B}\widehat{\mathbf{w}}$$

This is the generalized eigenvalue/eigenvector definition (eq. (B.2)) for $\mathbf{w}_i = \widehat{\mathbf{w}}$ and $\lambda_i = r(\widehat{\mathbf{w}})$.

We have thus shown that if $\widehat{\mathbf{w}}$ is a maximum of $r(\mathbf{w})$, that it is an eigenvector of $(\mathbf{A}, \mathbf{B})$, with $r(\widehat{\mathbf{w}})$ its corresponding eigenvalue.

As the second step, we now show that $r(\widehat{\mathbf{w}})$ is the *largest* eigenvalue of $(\mathbf{A}, \mathbf{B})$. We follow the reasoning of Trefethen and Bau, who prove a related result for the ordinary Rayleigh quotient [81, p. 204].

We will rewrite the generalized Rayleigh quotient $r(\mathbf{w})$ by writing the arbitrary vector $\mathbf{w}$ as a linear combination of the generalized eigenvectors $\mathbf{w}_i$ of $(\mathbf{A}, \mathbf{B})$: $\mathbf{w} = \sum_i c_i\mathbf{w}_i$. Then:

$$\begin{aligned} r(\mathbf{w}) &= \frac{(\sum_i c_i\mathbf{w}_i)^T\mathbf{A}(\sum_i c_i\mathbf{w}_i)}{(\sum_i c_i\mathbf{w}_i)^T\mathbf{B}(\sum_i c_i\mathbf{w}_i)} \\ &= \frac{\sum_i c_i^2\mathbf{w}_i^T\mathbf{A}\mathbf{w}_i}{\sum_i c_i^2\mathbf{w}_i^T\mathbf{B}\mathbf{w}_i} \\ &= \frac{\sum_i c_i^2\lambda_i\mathbf{w}_i^T\mathbf{B}\mathbf{w}_i}{\sum_i c_i^2\mathbf{w}_i^T\mathbf{B}\mathbf{w}_i}. \end{aligned}$$

Generalized eigenvectors are defined up to a scaling factor. We may therefore define our $\mathbf{w}_i$ to be scaled such that $\mathbf{w}_i^T\mathbf{B}\mathbf{w}_i = 1$. We then have:

$$r(\mathbf{w}) = \frac{\sum_i c_i^2\lambda_i}{\sum_i c_i^2}.$$

Each generalized Rayleigh quotient is thus a convex combination of generalized eigenvalues λ_i. The maximum of a convex combination of one-dimensional points is obtained in the largest of these points. If λ_1 is thus the largest generalized eigenvalue of $(\mathbf{A}, \mathbf{B})$, then $\max r(\mathbf{w}) = \lambda_1$.

We have thus shown that $\arg\max\ r(\mathbf{w}) = \mathbf{w}_1$, where $\mathbf{w}_1$ is an eigenvector of $(\mathbf{A}, \mathbf{B})$, and that its corresponding eigenvalue $\lambda_1 = \max r(\mathbf{w})$ is the largest of the eigenvalues of $(\mathbf{A}, \mathbf{B})$.

$\square$

Appendix C

Supplemental figures

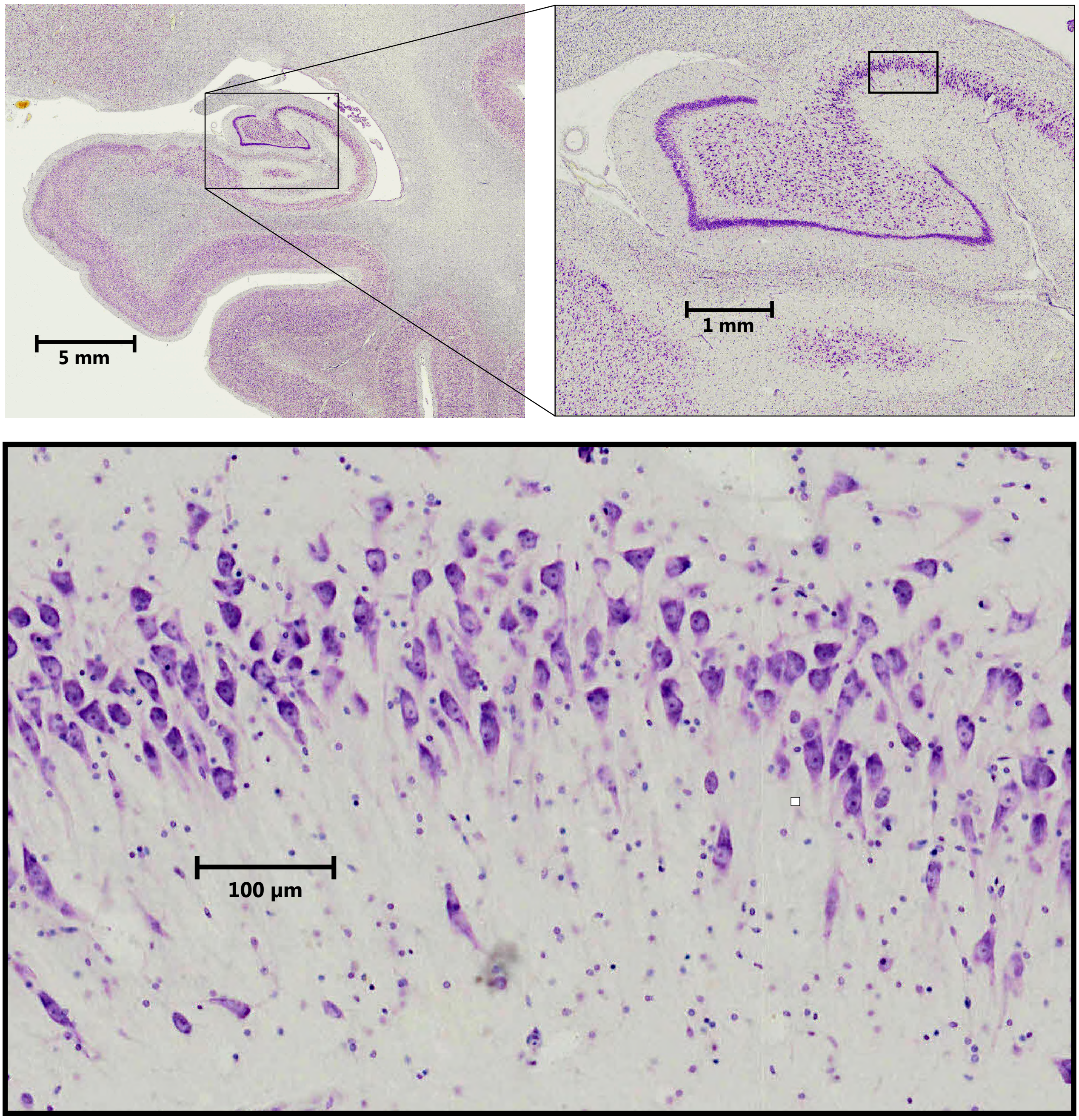


Figure C.1: **Individual neurons in the human hippocampus**. Zoom-in of the top-right panel of fig. 3.1. Black box in the top right panel marks location of the bottom panel. Small white box in bottom panel hints at the size of fig. C.3.

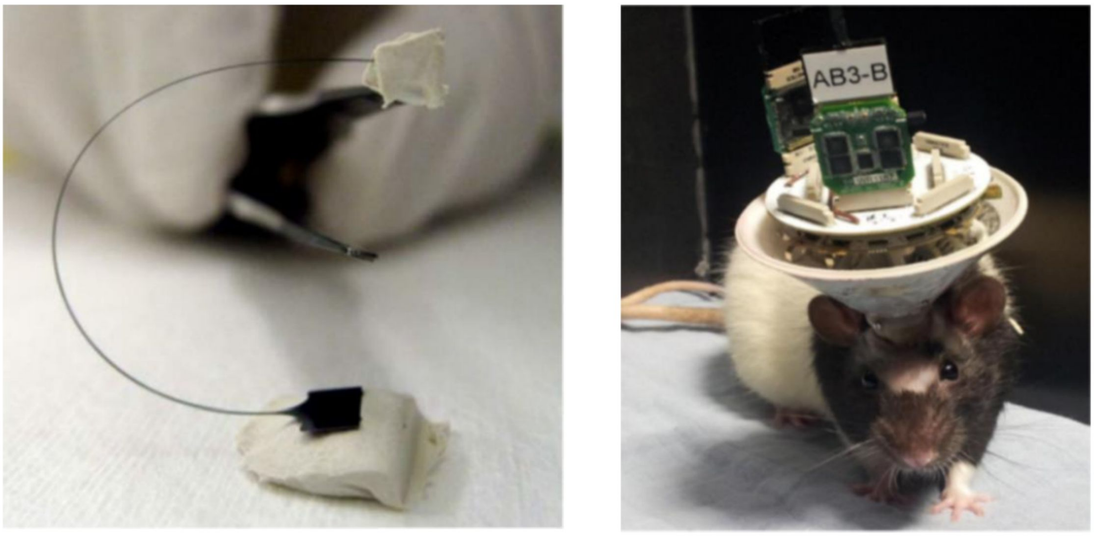


Figure C.2: **Probe shank & micro-drive array**. Reproduced from [3]. *Left*: flexible, 40 mm long probe shank. *Right*: test subject implanted with so called 'micro-drive array'. This is an electrode positioning device surgically attached to the rat's skull. The rat is anesthesised during surgery and is administered pain-killing drugs during the recovery period. The micro-drive array guides multiple electrodes into the brain (such as "tetrodes", bendable probes, and reference or stimulation electrodes), and connects them to the recording and stimulation hardware. The depth of the inserted electrodes can be precisely adjusted. This device allows to make voltage recordings while the animal can move around (the so called "freely behaving" setup, which allows for more natural behaviour than immobilised recording setups).

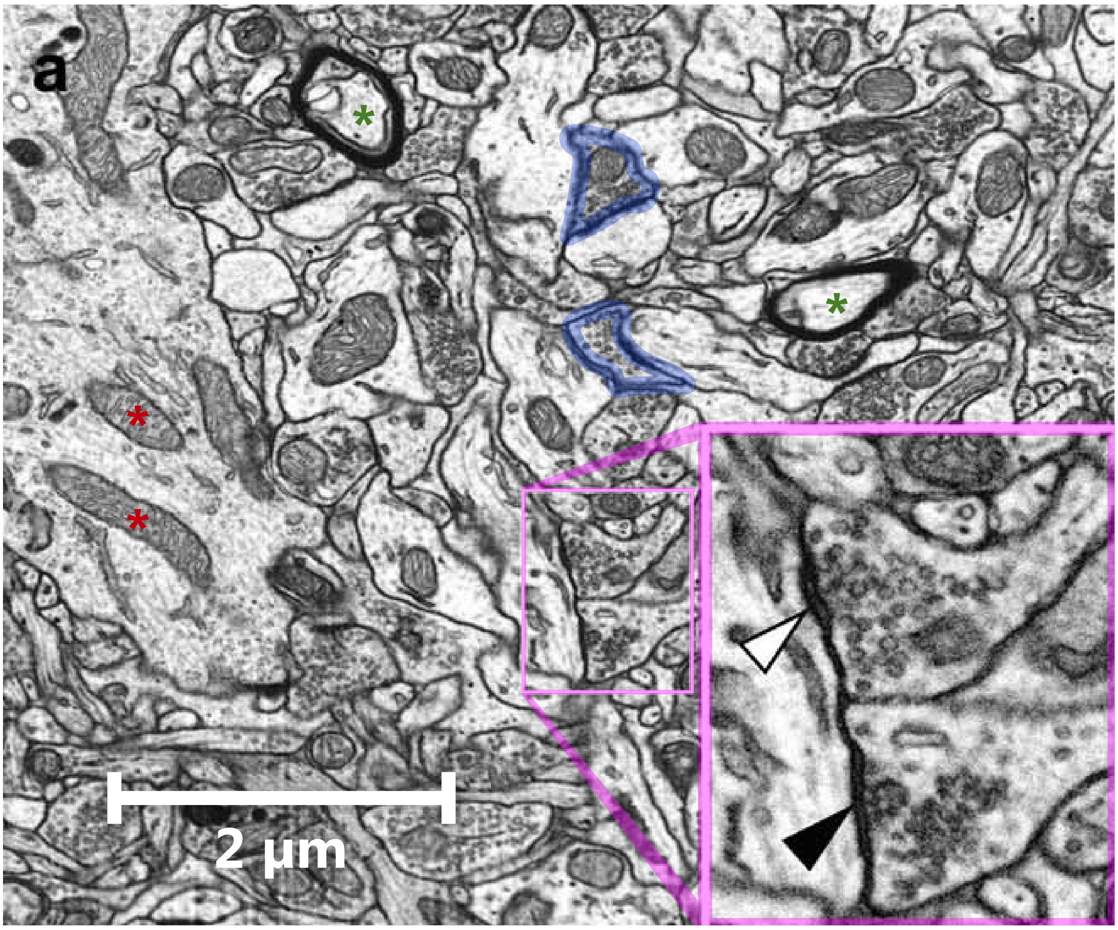
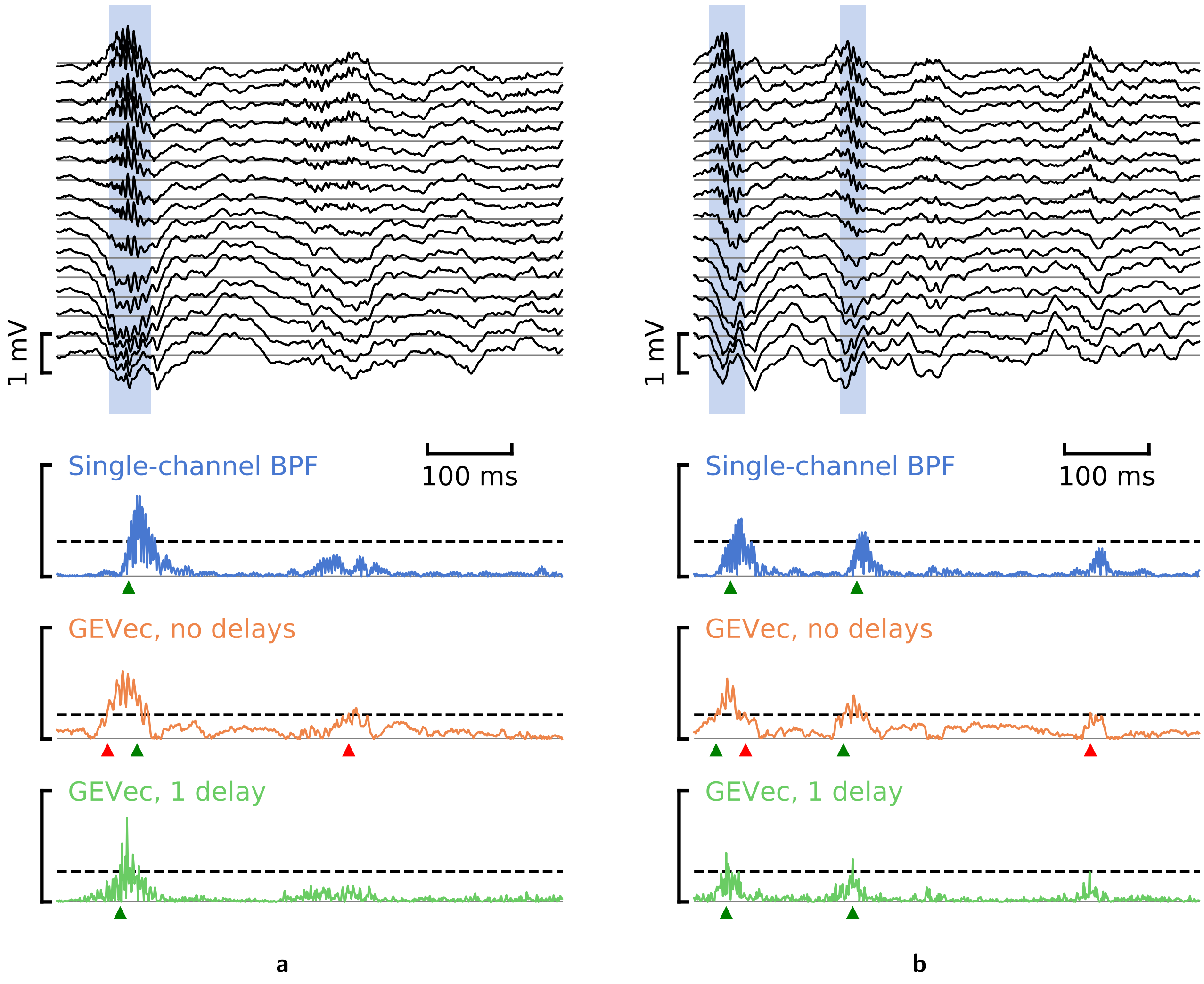


Figure C.3: **The environment in which recordings are made**. *Figure adapted from [82]*. Scanning electron micrograph of a slice of mouse cortex. (Rat hippocampal tissue looks similar, see e.g. [83]). This type of tissue is called "neuropil". It consists of the long and narrow excrescences of neurons (dendrites and axons) and of glial cells, seen here in cross and through section. Note that cell bodies are much larger than the cross sections of dendrites and axons as seen here; a typical neural cell body ("soma") of 15 µm wide would be larger than this image, which is 7 µm wide. (See also the small white box in fig. C.1, which is about the same size as this electron micrograph). Some landmarks are the many mitochondria (two of which are annotated with red *); two myelinated axons (green *); and all the lipid bilayers separating cells from their environment (thin black lines, two of which are highlighted in blue). The pink inset (1 µm by 1.24 µm) shows two "boutons" (axon terminals) synapsing onto a dendrite. Note the synaptic vesicles (the many small dark circles), which encapsulate neurotransmitter molecules, ready for release in the synaptic clefts (black and white arrows).


1 mV
Single-channel BPF
100 ms
GEVec, no delays
GEVec, 1 delay
a
1 mV
Single-channel BPF
100 ms
GEVec, no delays
GEVec, 1 delay
b

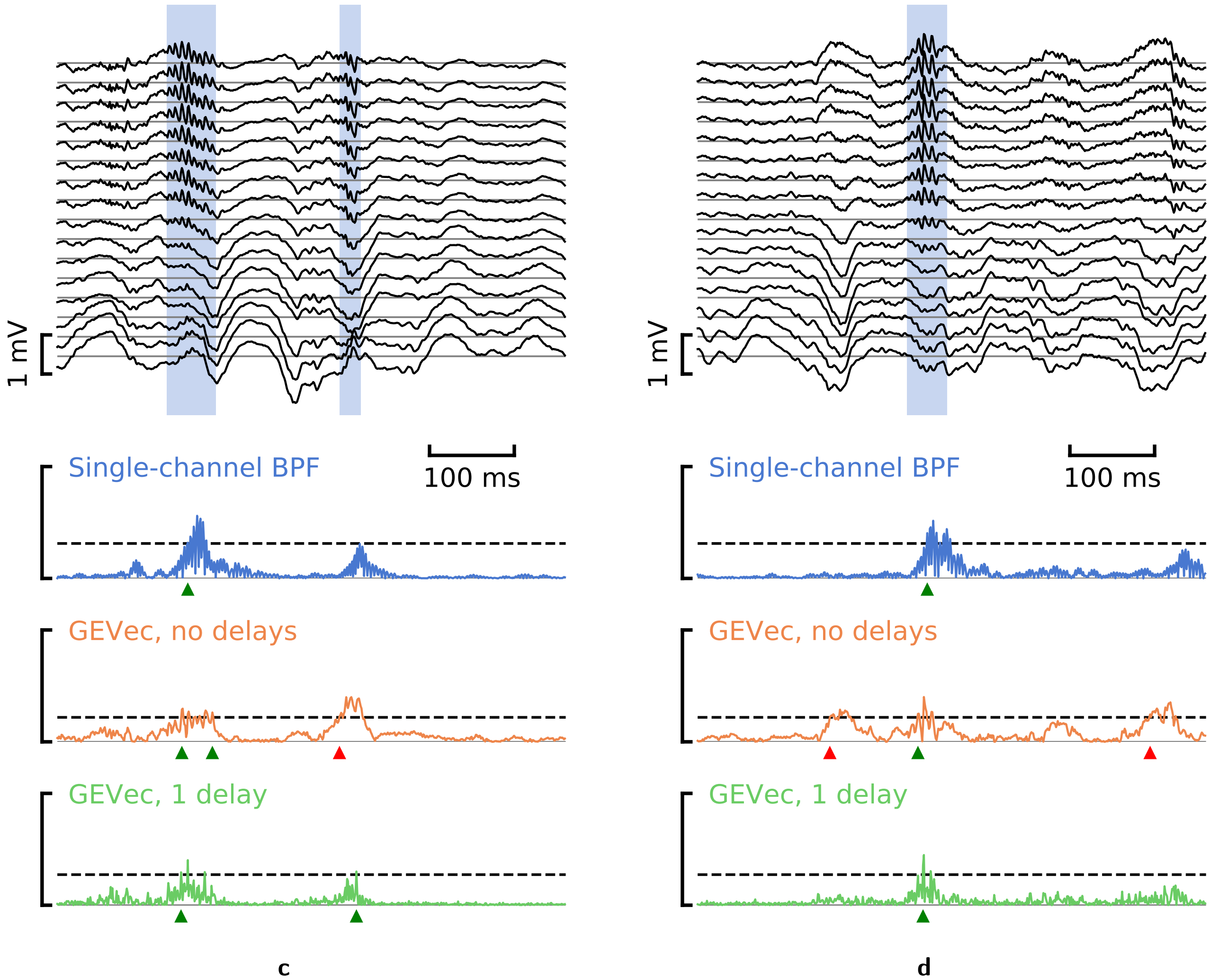


Figure C.4: **Extracts from input data and corresponding linear filter output envelopes**. See fig. 8.2 for legend.

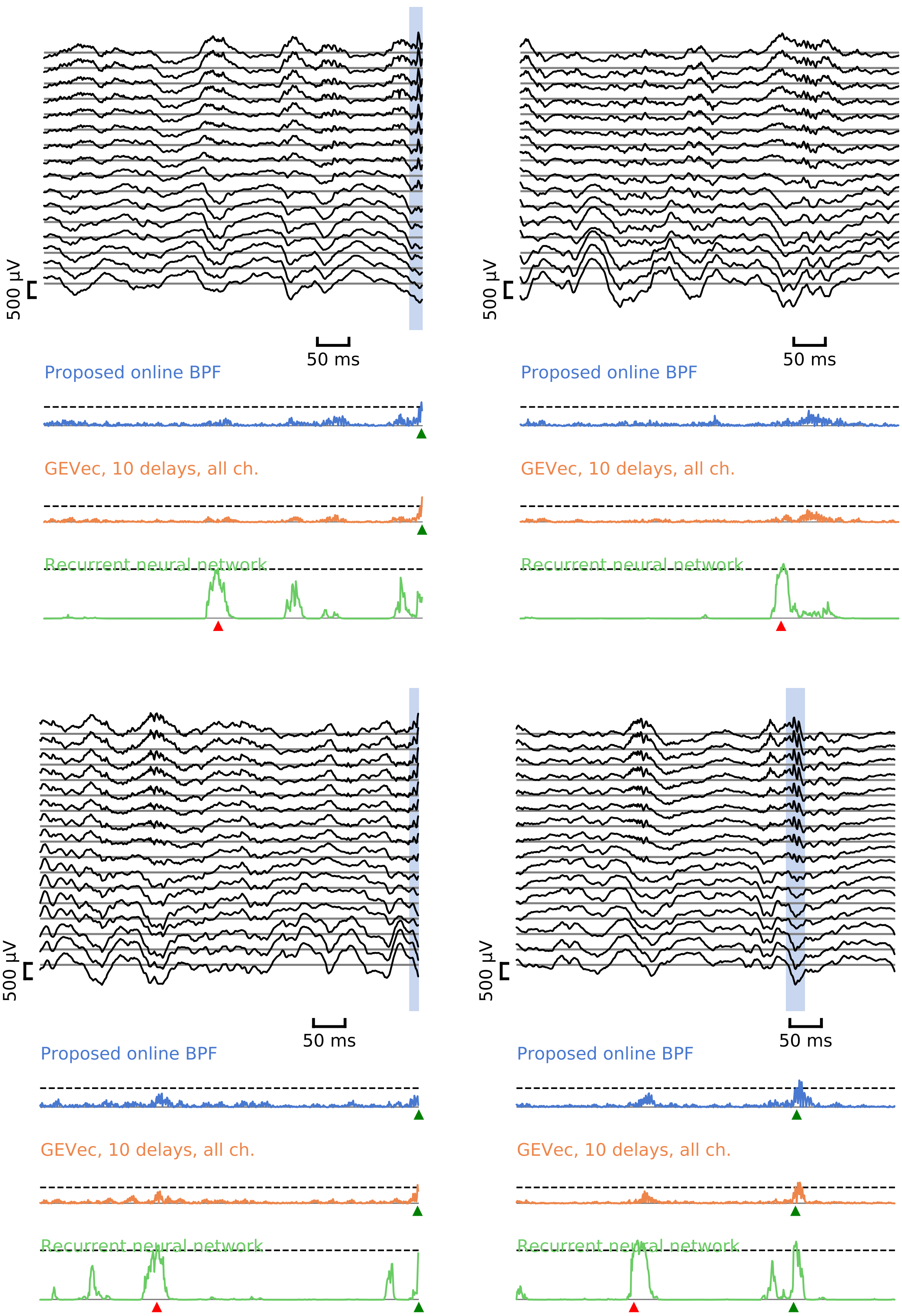


Figure C.5: **False positive RNN detections**, caused by SWR-like spatiotemporal LFP profiles without well-developed ripples.

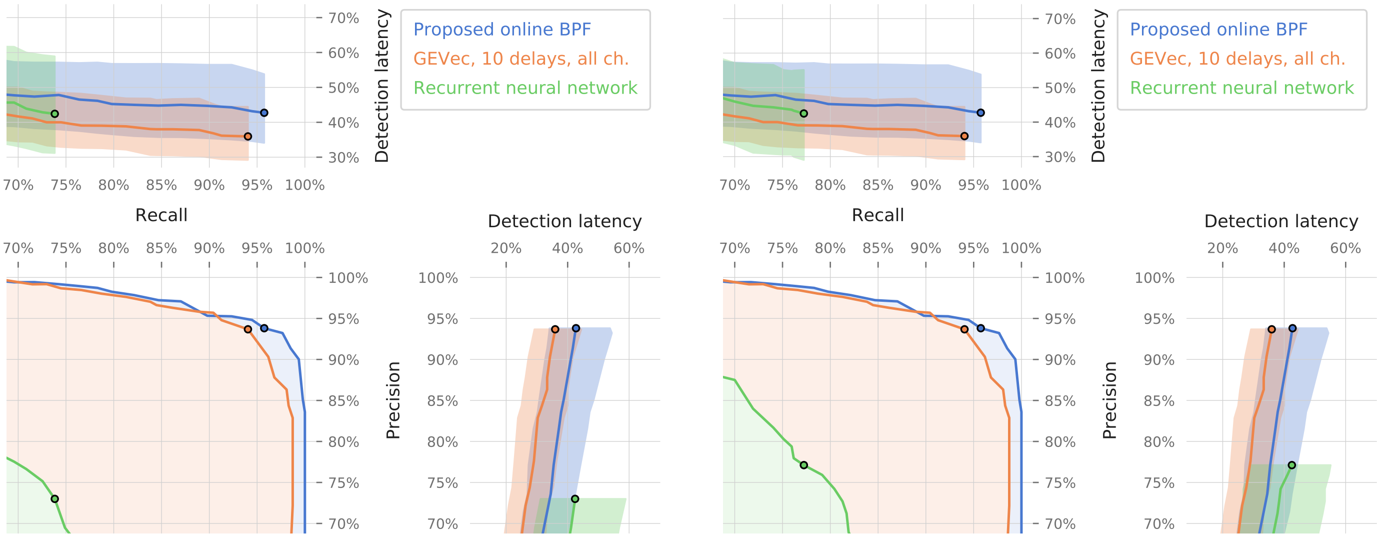


Figure C.6: **Start-block RNN's**. Online SWR detection performance of GRU-RNN's using block functions at the start of reference SWR segments as target function. *Left*: Target is 1 for 35 ms before to 55 ms after reference SWR start. *Right*: Target is 1 for 15 ms before to 50 ms after reference SWR start. See fig. 7.5 for legend.

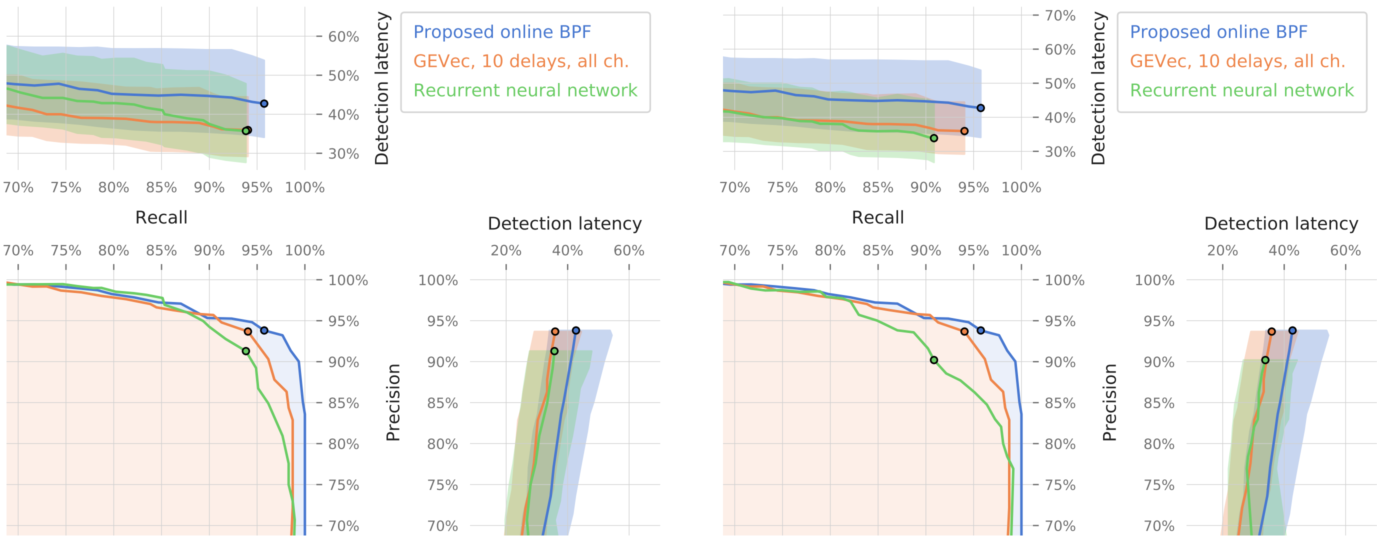


Figure C.7: **Single-input-channel RNN's**. Online SWR detection performance of GRU-RNN's using only a pyramidal cell layer LFP channel as input. *Left*: two-layer RNN, with 40 hidden units per layer. *Right*: one-layer RNN, with 25 hidden units. See fig. 7.5 for legend.

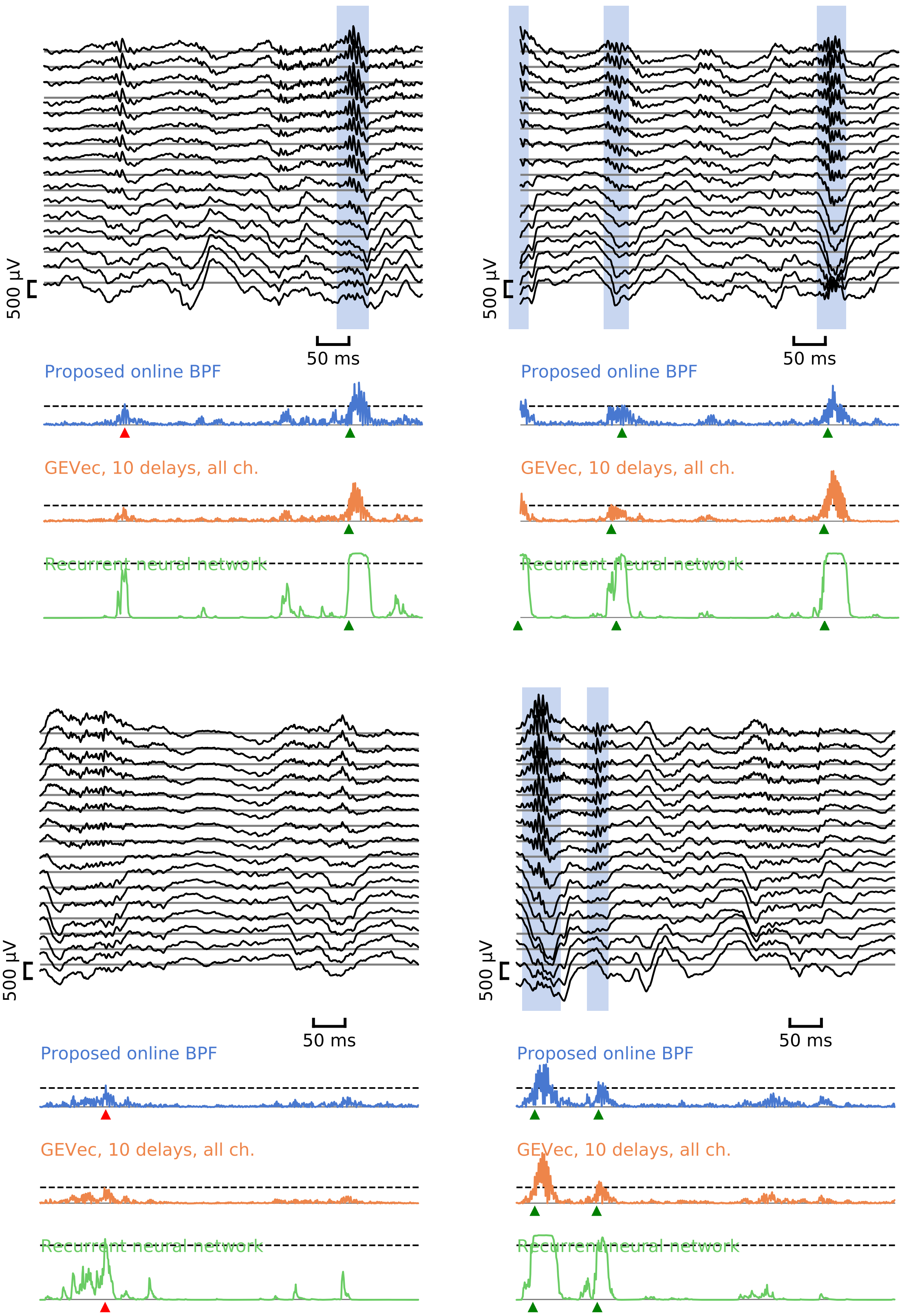


Figure C.8: **Single-input-channel RNN envelopes.**

References

[1] C. J. Van Rijsbergen. *Information Retrieval*. 2nd edition. Newton, MA, USA: Butterworth-Heinemann, 1979. URL: http://www.dcs.gla.ac.uk/Keith/Preface.html.

[2] Max O. Krucoff, Shervin Rahimpour, Marc W. Slutzky, V. Reggie Edgerton, and Dennis A. Turner. "Enhancing Nervous System Recovery through Neurobiologics, Neural Interface Training, and Neurorehabilitation". In: *Frontiers in Neuroscience* 10 (2016). ISSN: 1662-453X. DOI: 10.3389/fnins.2016.00584. URL: https://www.frontiersin.org/articles/10.3389/fnins.2016.00584/full (visited on 12/12/2018).

[3] Frédéric Michon, Arno Aarts, Tobias Holzhammer, Patrick Ruther, Gustaaf Borghs, Bruce McNaughton, and Fabian Kloosterman. "Integration of Silicon-Based Neural Probes and Micro-Drive Arrays for Chronic Recording of Large Populations of Neurons in Behaving Animals". In: *Journal of Neural Engineering* 13.4 (2016), p. 046018. ISSN: 1741-2552. DOI: 10.1088/1741-2560/13/4/046018. URL: http://stacks.iop.org/1741-2552/13/i=4/a=046018 (visited on 05/31/2018).

[4] James J. Jun, Nicholas A. Steinmetz, Joshua H. Siegle, et al. "Fully Integrated Silicon Probes for High-Density Recording of Neural Activity". In: *Nature* 551.7679 (Nov. 2017), pp. 232–236. ISSN: 1476-4687. DOI: 10.1038/nature24636. URL: https://www.nature.com/articles/nature24636 (visited on 08/09/2018).

[5] Paul L. Nunez and Ramesh Srinivasan. *Electric Fields of the Brain: The Neurophysics of EEG*. Oxford University Press, USA, 2006.

[6] Robert Plonsey and Roger C. Barr. *Bioelectricity: A Quantitative Approach*. 3rd ed. New York, NY: Springer, 2007. 528 pp. ISBN: 978-0-387-48864-6.

[7] Orjan G. Martinsen and Sverre Grimnes. "Electrolytics". In: *Bioimpedance and Bioelectricity Basics*. 3rd ed. Academic Press, 2015.

[8] Richard Feynman, Robert Leighton, and Michael Sands. *The Feynman Lectures on Physics, Vol II: Mainly Electromagnetism and Matter*. Vol. 2. 3 vols. The California Institute of Technology, 2013. URL: http://www.feynmanlectures.caltech.edu/.

[9] P Marszalek, J J Zielinsky, M Fikus, and T Y Tsong. "Determination of Electric Parameters of Cell Membranes by a Dielectrophoresis Method." In: *Biophysical Journal* 59.5 (May 1991), pp. 982–987. ISSN: 0006-3495. pmid: 1831052. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1281332/ (visited on 08/12/2018).

[10] J. C. Weaver and K. H. Schoenbach. "Biodielectrics". In: *IEEE Transactions on Dielectrics and Electrical Insulation* 10.5 (Oct. 2003), pp. 715–716. ISSN: 1070-9878. DOI: 10.1109/TDEI.2003.1237322.

[11] Orjan G. Martinsen and Sverre Grimnes. *Bioimpedance and Bioelectricity Basics*. 3rd ed. Academic Press, 2015.

[12] Peter Dayan and L. F. Abbott. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. 1st ed. Computational neuroscience. Cambridge, Mass: Massachusetts Institute of Technology Press, 2001. 460 pp. ISBN: 978-0-262-04199-7.

[13] Ruth W. Chabay and Bruce A. Sherwood. *Matter and Interactions*. 4th edition. Wiley, 2015.

[14] E.R. Kandel, T.M. Jessell, J.H. Schwartz, S.A. Siegelbaum, and A.J. Hudspeth. "Membrane Potential and the Passive Electrical Properties of the Neuron". In: *Principles of Neural Science, Fifth Edition*. 5th ed. Principles of Neural Science. McGraw-Hill Education, 2013. ISBN: 978-0-07-139011-8. URL: https://books.google.be/books?id=s64z-LdAIsEC.

[15] Claude Bedard and Alain Destexhe. "A Generalized Theory for Current-Source Density Analysis in Brain Tissue". In: *Physical Review E* 84.4 (Oct. 10, 2011). ISSN: 1539-3755, 1550-2376. DOI: 10.1103/PhysRevE.84.041909. arXiv: 1101.1094. URL: http://arxiv.org/abs/1101.1094 (visited on 12/19/2018).

[16] Ulla Mitzdorf. "Current Source-Density Method and Application in Cat Cerebral Cortex: Investigation of Evoked Potentials and EEG Phenomena". In: *Physiological reviews* 65.1 (1985), pp. 37–100.

[17] C. Nicholson and J. A. Freeman. "Theory of Current Source-Density Analysis and Determination of Conductivity Tensor for Anuran Cerebellum". In: *Journal of Neurophysiology* (Mar. 1, 1975). DOI: 10.1152/jn.1975.38.2.356. URL: http://www.physiology.org/doi/abs/10.1152/jn.1975.38.2.356 (visited on 12/19/2018).

[18] Henrik Lindén, Espen Hagen, Szymon Leski, Eivind S. Norheim, Klas H. Pettersen, and Gaute T. Einevoll. "LFPy: A Tool for Biophysical Simulation of Extracellular Potentials Generated by Detailed Model Neurons". In: *Frontiers in Neuroinformatics* 7 (2014). ISSN: 1662-5196. DOI: 10.3389/fninf.2013.00041. URL: https://www.frontiersin.org/articles/10.3389/fninf.2013.00041/full#h3 (visited on 07/26/2018).

[19] György Buzsáki, Costas A. Anastassiou, and Christof Koch. "The Origin of Extracellular Fields and Currents — EEG, ECoG, LFP and Spikes". In: *Nature Reviews Neuroscience* 13.6 (June 2012), pp. 407–420. ISSN: 1471-0048. DOI: 10.1038/nrn3241. URL: https://www.nature.com/articles/nrn3241 (visited on 07/26/2018).

[20] Eric Michel, Daniel Hernandez, and Soo Yeol Lee. "Electrical Conductivity and Permittivity Maps of Brain Tissues Derived from Water Content Based on T1-Weighted Acquisition". In: *Magnetic Resonance in Medicine* 77.3 (Mar. 1, 2017), pp. 1094–1103. ISSN: 1522-2594. DOI: 10.1002/mrm.26193. URL: http://onlinelibrary.wiley.com/doi/abs/10.1002/mrm.26193 (visited on 08/12/2018).

[21] Walter Pitts. "Investigations on Synaptic Transmission". In: *Trans. 9 Th Conf.* Cybernetics. 1952, pp 159-166.

[22] B. Narasinga Rao, K. P. Rao, and R. Ramana Rao. "Morphometric Study of Hippocampus in Adult Human Brains". In: *Int J Appl Basic Med Res* 2 (2012), pp. 139–143.

[23] Jürgen K. Mai, Milan Majtanik, and George Paxinos. *Atlas of the Human Brain*. Fourth. Academic Press, 2015.

[24] Jürgen K. Mai, Thomas Voss, and George Paxinos. *Atlas of the Human Brain – Hi-Resolution Sections · Cells (Nissl Staining) · Virtual Microscopy*. 2008. URL: http://www.thehumanbrain.info/brain/sections/microslices/ (visited on 12/16/2018).

[25] George Paxinos and Charles Watson. *The Rat Brain in Stereotaxic Coordinates*. 6th edition. Academic Press, 2007.

[26] Lisa J. Kjonigsen, Trygve B. Leergaard, Menno P. Witter, and Jan G. Bjaalie. "Digital Atlas of Anatomical Subdivisions and Boundaries of the Rat Hippocampal Region". In: *Frontiers in Neuroinformatics* 5 (2011), p. 2. ISSN: 1662-5196. DOI: 10.3389/fninf.2011.00002. pmid: 21519393.

[27] David Amaral and Pierre Lavenex. "Hippocampal Neuroanatomy". In: *The Hippocampus Book*. Ed. by Per Andersen, Richard Morris, David Amaral, and John O'Keefe. OCLC: ocm64444087. Oxford ; New York: Oxford University Press, 2007. ISBN: 978-0-19-510027-3.

[28] Gabrielle Girardeau and Michaël Zugaro. "Hippocampal Ripples and Memory Consolidation". In: *Current Opinion in Neurobiology*. Behavioural and cognitive neuroscience 21.3 (June 1, 2011), pp. 452–459. ISSN: 0959-4388. DOI: 10.1016/j.conb.2011.02.005. URL: http://www.sciencedirect.com/science/article/pii/S0959438811000316 (visited on 06/25/2018).

[29] György Buzsáki. "Hippocampal Sharp Wave-ripple: A Cognitive Biomarker for Episodic Memory and Planning". In: *Hippocampus* 25.10 (Sept. 26, 2015), pp. 1073–1188. ISSN: 1050-9631. DOI: 10.1002/hipo.22488. URL: http://onlinelibrary.wiley.com/doi/full/10.1002/hipo.22488 (visited on 04/23/2018).

[30] György Buzsáki. "Hippocampal Sharp Waves: Their Origin and Significance". In: *Brain Research* 398.2 (Nov. 29, 1986), pp. 242–252. ISSN: 0006-8993. DOI: 10.1016/0006-8993(86)91483-6. URL: http://www.sciencedirect.com/science/article/pii/0006899386914836 (visited on 01/04/2019).

[31] H Freyja Ólafsdóttir, Daniel Bush, and Caswell Barry. "The Role of Hippocampal Replay in Memory and Planning". In: *Current Biology* 28.1 (2018), R37–R50.

[32] Dale Purves, George J. Augustine, David Fitzpatrick, William C. Hall, Anthony-Samuel LaMantia, Richard D. Mooney, Michael L. Platt, and Leonard E. White, eds. *Neuroscience*. 6th. New York: Sinauer Associates is an imprint of Oxford University Press, Oct. 12, 2017. 960 pp. ISBN: 978-1-60535-380-7.

[33] E.R. Kandel, T.M. Jessell, J.H. Schwartz, S.A. Siegelbaum, and A.J. Hudspeth. *Principles of Neural Science*. Fifth edition. McGraw-Hill Education, 2013. ISBN: 978-0-07-139011-8.

[34] Mark Bear, Barry Connors, and Mike Paradiso. *Neuroscience: Exploring the Brain*. Fourth, International edition. Philadelphia Baltimore New York London: LWW, Mar. 31, 2015. 1008 pp. ISBN: 978-1-4511-0954-2.

[35] Casimir M. Wierzynski, Evgueniy V. Lubenov, Ming Gu, and Athanassios G. Siapas. "State-Dependent Spike-Timing Relationships between Hippocampal and Prefrontal Circuits during Sleep". In: *Neuron* 61.4 (Feb. 26, 2009), pp. 587–596. ISSN: 1097-4199. DOI: 10.1016/j.neuron.2009.01.011. pmid: 19249278.

[36] Adrien Peyrache, Mehdi Khamassi, Karim Benchenane, Sidney I. Wiener, and Francesco P. Battaglia. "Replay of Rule-Learning Related Neural Patterns in the Prefrontal Cortex during Sleep". In: *Nature Neuroscience* 12.7 (July 2009), pp. 919–926. ISSN: 1546-1726. DOI: 10.1038/nn.2337. URL: https://www.nature.com/articles/nn.2337 (visited on 08/02/2018).

[37] Fabian Kloosterman. "Analysis of Hippocampal Memory Replay Using Neural Population Decoding". In: *Neuronal Network Analysis: Concepts and Experimental Approaches*. Ed. by Tommaso Fellin and Michael Halassa. Neuromethods. Totowa, NJ: Humana Press, 2012, pp. 259–282. ISBN: 978-1-61779-633-3. DOI: 10.1007/7657_2011_8. URL: https://doi.org/10.1007/7657_2011_8 (visited on 01/04/2019).

[38] Natalia Caporale and Yang Dan. "Spike Timing–Dependent Plasticity: A Hebbian Learning Rule". In: *Annual Review of Neuroscience* 31.1 (June 17, 2008), pp. 25–46. ISSN: 0147-006X. DOI: 10.1146/annurev.neuro.31.060407.125639. URL: https://www.annualreviews.org/doi/10.1146/annurev.neuro.31.060407.125639 (visited on 01/04/2019).

[39] Valérie Ego-Stengel and Matthew A Wilson. "Disruption of Ripple-Associated Hippocampal Activity during Rest Impairs Spatial Learning in the Rat". In: *Hippocampus* 20.1 (2009), pp. 1–10. URL: http://onlinelibrary.wiley.com.kuleuven.ezproxy.kuleuven.be/doi/10.1002/hipo.20707/full.

[40] Gabrielle Girardeau, Karim Benchenane, Sidney I Wiener, György Buzsáki, and Michaël B Zugaro. "Selective Suppression of Hippocampal Ripples Impairs Spatial Memory". In: *Nature Neuroscience* 12.10 (Oct. 2009), pp. 1222–1223. ISSN: 1097-6256, 1546-1726. DOI: 10.1038/nn.2384. URL: http://www.nature.com/articles/nn.2384 (visited on 04/16/2018).

[41] Shantanu P. Jadhav, Caleb Kemere, P. Walter German, and Loren M. Frank. "Awake Hippocampal Sharp-Wave Ripples Support Spatial Memory". In: *Science* 336.6087 (June 15, 2012), pp. 1454–1458. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.1217230. pmid: 22555434. URL: http://science.sciencemag.org/content/336/6087/1454 (visited on 04/19/2018).

[42] Gabrielle Girardeau, Anne Cei, and Michaël Zugaro. "Learning-Induced Plasticity Regulates Hippocampal Sharp Wave-Ripple Drive". In: *Journal of Neuroscience* 34.15 (Apr. 9, 2014), pp. 5176–5183. ISSN: 0270-6474, 1529-2401. DOI: 10.1523/JNEUROSCI.4288-13.2014. pmid: 24719097. URL: http://www.jneurosci.org/content/34/15/5176 (visited on 04/16/2018).

[43] Krisztián A. Kovács, Joseph O'Neill, Philipp Schoenenberger, Markku Penttonen, Damaris K. Ranguel Guerrero, and Jozsef Csicsvari. "Optogenetically Blocking Sharp Wave Ripple Events in Sleep Does Not Interfere with the Formation of Stable Spatial Representation in the CA1 Area of the Hippocampus". In: *PLOS ONE* 11.10 (2016), e0164675. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0164675. URL: http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0164675 (visited on 06/06/2018).

[44] Omid Talakoub, Andrea Gomez Palacio Schjetnan, Taufik A. Valiante, Milos R. Popovic, and Kari L. Hoffman. "Closed-Loop Interruption of Hippocampal Ripples through Fornix Stimulation in the Non-Human Primate". In: *Brain Stimulation: Basic, Translational, and Clinical Research in Neuromodulation* 9.6 (Nov. 1, 2016), pp. 911–918. ISSN: 1935-861X, 1876-4754. DOI: 10.1016/j.brs.2016.07.010. pmid: 27576185. URL: http://www.brainstimjrnl.com/article/S1935-861X(16)30202-9/abstract (visited on 04/16/2018).

[45] Yulia Novitskaya, Susan J. Sara, Nikos K. Logothetis, and Oxana Eschenko. "Ripple-Triggered Stimulation of the Locus Coeruleus during Post-Learning Sleep Disrupts Ripple/Spindle Coupling and Impairs Memory Consolidation". In: *Learning & Memory* 23.5 (Jan. 5, 2016), pp. 238–248. ISSN: 1072-0502, 1549-5485. DOI: 10.1101/lm.040923.115. pmid: 27084931. URL: http://learnmem.cshlp.org/content/23/5/238 (visited on 01/06/2019).

[46] Tom Roelandts. *How to Create a Configurable Filter Using a Kaiser Window*. Dec. 26, 2016. URL: https://tomroelandts.com/articles/how-to-create-a-configurable-filter-using-a-kaiser-window (visited on 11/25/2018).

[47] Eric Jones, Travis Oliphant, Pearu Peterson, et al. "SciPy: Open Source Scientific Tools for Python". In: (2018). URL: http://www.scipy.org/.

[48] Alan V. Oppenheim and Ronald W. Schafer. *Discrete-Time Signal Processing*. 3rd. Upper Saddle River, NJ, USA: Prentice Hall Press, 2009. ISBN: 978-0-13-198842-2.

[49] Richard G. Lyons. *Understanding Digital Signal Processing*. Pearson Education, Nov. 1, 2010. 1166 pp. ISBN: 978-0-13-702852-8.

[50] Zoltán Nádasdy, Hajime Hirase, András Czurkó, Jozsef Csicsvari, and György Buzsáki. "Replay and Time Compression of Recurring Spike Sequences in the Hippocampus". In: *Journal of Neuroscience* 19.21 (Nov. 1, 1999), pp. 9497–9507. ISSN: 0270-6474, 1529-2401. DOI: 10.1523/JNEUROSCI.19-21-09497.1999. pmid: 10531452. URL: http://www.jneurosci.org/content/19/21/9497 (visited on 08/02/2018).

[51] Jozsef Csicsvari, Hajime Hirase, Akira Mamiya, and György Buzsáki. "Ensemble Patterns of Hippocampal CA3-CA1 Neurons during Sharp Wave–Associated Population Events". In: *Neuron* 28.2 (Nov. 1, 2000), pp. 585–594. ISSN: 0896-6273. DOI: 10.1016/S0896-6273(00)00135-5. URL: http://www.sciencedirect.com/science/article/pii/S0896627300001355 (visited on 08/01/2018).

[52] Shayok Dutta, Etienne Ackermann, and Caleb Kemere. "Analysis of an Open Source, Closed-Loop, Realtime System for Hippocampal Sharp-Wave Ripple Disruption". In: *bioRxiv* (Apr. 11, 2018). DOI: 10.1101/298661. URL: https://www.biorxiv.org/content/early/2018/04/11/298661 (visited on 05/30/2018).

[53] Christoph J. Behrens, Leander P. van den Boom, Livia de Hoz, Alon Friedman, and Uwe Heinemann. "Induction of Sharp Wave–Ripple Complexes *in Vitro* and Reorganization of Hippocampal Networks". In: *Nature Neuroscience* 8.11 (Nov. 2005), pp. 1560–1567. ISSN: 1546-1726. DOI: `10.1038/nn1571`. URL: `https://www.nature.com/articles/nn1571` (visited on 07/31/2018).

[54] Josef H. L. P. Sadowski, Matthew W. Jones, and Jack R. Mellor. "Sharp-Wave Ripples Orchestrate the Induction of Synaptic Plasticity during Reactivation of Place Cell Firing Patterns in the Hippocampus". In: *Cell Reports* 14.8 (Mar. 1, 2016), pp. 1916–1929. ISSN: 2211-1247. DOI: `10.1016/j.celrep.2016.01.061`. URL: `http://www.sciencedirect.com/science/article/pii/S2211124716300390` (visited on 06/17/2018).

[55] Davide Ciliberti and Fabian Kloosterman. "Falcon: A Highly Flexible Open-Source Software for Closed-Loop Neuroscience". In: *Journal of neural engineering* 14.4 (2017), p. 045004.

[56] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Fourth edition. The Johns Hopkins University Press, 2013.

[57] Klaus Greff, Rupesh Kumar Srivastava, Jan Koutník, Bas R. Steunebrink, and Jürgen Schmidhuber. "LSTM: A Search Space Odyssey". In: *IEEE Transactions on Neural Networks and Learning Systems* 28.10 (Oct. 2017), pp. 2222–2232. ISSN: 2162-237X, 2162-2388. DOI: `10.1109/TNNLS.2016.2582924`. arXiv: `1503.04069`. URL: `http://arxiv.org/abs/1503.04069` (visited on 06/10/2018).

[58] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep Learning". In: *Nature* 521.7553 (May 2015), pp. 436–444. ISSN: 1476-4687. DOI: `10.1038/nature14539`. URL: `https://www.nature.com/articles/nature14539` (visited on 06/08/2018).

[59] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. Vol. 1. MIT press Cambridge, 2016.

[60] Søren Kaae Sønderby and Ole Winther. "Protein Secondary Structure Prediction with Long Short Term Memory Networks". In: (Dec. 25, 2014). arXiv: `1412.7828 [cs, q-bio]`. URL: `http://arxiv.org/abs/1412.7828` (visited on 06/10/2018).

[61] Zachary C. Lipton, David C. Kale, Charles Elkan, and Randall Wetzel. "Learning to Diagnose with LSTM Recurrent Neural Networks". In: (Nov. 11, 2015). arXiv: `1511.03677 [cs]`. URL: `http://arxiv.org/abs/1511.03677` (visited on 06/10/2018).

[62] Christopher J. Cueva and Xue-Xin Wei. "Emergence of Grid-like Representations by Training Recurrent Neural Networks to Perform Spatial Localization". In: (Feb. 15, 2018). URL: `https://openreview.net/forum?id=B17JTOe0-` (visited on 06/07/2018).

[63] Andrea Banino, Caswell Barry, Benigno Uria, et al. "Vector-Based Navigation Using Grid-like Representations in Artificial Agents". In: *Nature* 557.7705 (May 2018), pp. 429–433. ISSN: 1476-4687. DOI: `10.1038/s41586-018-0102-6`. URL: `https://www.nature.com/articles/s41586-018-0102-6` (visited on 06/07/2018).

[64] H. Francis Song, Guangyu R. Yang, and Xiao-Jing Wang. "Reward-Based Training of Recurrent Neural Networks for Cognitive and Value-Based Tasks". In: *eLife* 6 (Jan. 13, 2017), e21492. ISSN: 2050-084X. DOI: 10.7554/eLife.21492. URL: https://elifesciences.org/articles/21492 (visited on 06/07/2018).

[65] Kezhi Li, Avelino Javer, Eric E. Keaveny, and Andre E.X. Brown. "Recurrent Neural Networks with Interpretable Cells Predict and Classify Worm Behaviour". In: (Nov. 20, 2017). DOI: 10.1101/222208. URL: http://biorxiv.org/lookup/doi/10.1101/222208 (visited on 06/07/2018).

[66] Umut Güçlü, Van Gerven, and Marcel A. J. "Modeling the Dynamics of Human Brain Activity with Recurrent Neural Networks". In: *Frontiers in Computational Neuroscience* 11 (2017). ISSN: 1662-5188. DOI: 10.3389/fncom.2017.00007. URL: https://www.frontiersin.org/articles/10.3389/fncom.2017.00007/full (visited on 06/07/2018).

[67] Nikolaus Kriegeskorte. "Deep Neural Networks: A New Framework for Modeling Biological Vision and Brain Information Processing". In: *Annual Review of Vision Science* 1.1 (Nov. 18, 2015), pp. 417–446. ISSN: 2374-4642. DOI: 10.1146/annurev-vision-082114-035447. URL: http://www.annualreviews.org/doi/10.1146/annurev-vision-082114-035447 (visited on 06/07/2018).

[68] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. "Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation". In: *arXiv preprint arXiv:1406.1078* (2014).

[69] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[70] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling". In: (Dec. 11, 2014). arXiv: 1412.3555 [cs]. URL: http://arxiv.org/abs/1412.3555 (visited on 06/10/2018).

[71] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. "Learning Representations by Back-Propagating Errors". In: *Nature* 323.6088 (Oct. 1986), pp. 533–536. ISSN: 1476-4687. DOI: 10.1038/323533a0. URL: https://www.nature.com/articles/323533a0 (visited on 01/10/2019).

[72] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. "Automatic Differentiation in PyTorch". In: *NIPS-W*. 2017.

[73] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: (Dec. 22, 2014). arXiv: 1412.6980 [cs]. URL: http://arxiv.org/abs/1412.6980 (visited on 01/10/2019).

[74] John O'Keefe. "Hippocampal Neurophysiology in the Behaving Animal". In: *The Hippocampus Book*. Oxford University Press, 2007.

[75] Laura Lee Colgin. "Rhythms of the Hippocampal Network". In: *Nature Reviews Neuroscience* 17.4 (2016), p. 239.

[76] Howard Eichenbaum. "Prefrontal–Hippocampal Interactions in Episodic Memory". In: *Nature Reviews Neuroscience* 18.9 (2017), p. 547.

[77] Jozsef Csicsvari, Hajime Hirase, András Czurkó, Akira Mamiya, and György Buzsáki. "Oscillatory Coupling of Hippocampal Pyramidal Cells and Interneurons in the Behaving Rat". In: *Journal of Neuroscience* 19.1 (Jan. 1, 1999), pp. 274–287. ISSN: 0270-6474, 1529-2401. DOI: 10.1523/JNEUROSCI.19-01-00274.1999. pmid: 9870957. URL: http://www.jneurosci.org/content/19/1/274 (visited on 11/13/2018).

[78] Jozsef Csicsvari, Hajime Hirase, András Czurkó, Akira Mamiya, and György Buzsáki. "Fast Network Oscillations in the Hippocampal CA1 Region of the Behaving Rat". In: *The Journal of Neuroscience* 19.16 (Aug. 15, 1999), RC20–RC20. ISSN: 0270-6474, 1529-2401. DOI: 10.1523/JNEUROSCI.19-16-j0001.1999. URL: http://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.19-16-j0001.1999 (visited on 11/13/2018).

[79] Sen Cheng and Loren M. Frank. "New Experiences Enhance Coordinated Neural Activity in the Hippocampus". In: *Neuron* 57.2 (Jan. 24, 2008), pp. 303–313. ISSN: 0896-6273. DOI: 10.1016/j.neuron.2007.11.035. pmid: 18215626. URL: https://www.cell.com/neuron/abstract/S0896-6273(07)01030-6 (visited on 11/13/2018).

[80] Caleb Kemere, Margaret F. Carr, Mattias P. Karlsson, and Loren M. Frank. "Rapid and Continuous Modulation of Hippocampal Network State during Exploration of New Places". In: *PLOS ONE* 8.9 (2-Sep-2013), e73114. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0073114. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0073114 (visited on 11/13/2018).

[81] Lloyd N. Trefethen and David III Bau. *Numerical Linear Algebra*. Siam, 1997.

[82] G. Knott, H. Marchman, D. Wall, and B. Lich. "Serial Section Scanning Electron Microscopy of Adult Brain Tissue Using Focused Ion Beam Milling". In: *Journal of Neuroscience* 28.12 (Mar. 19, 2008), pp. 2959–2964. ISSN: 0270-6474, 1529-2401. DOI: 10.1523/JNEUROSCI.3189-07.2008. URL: http://www.jneurosci.org/cgi/doi/10.1523/JNEUROSCI.3189-07.2008 (visited on 07/29/2018).

[83] E. Anne Martin, Derek Woodruff, Randi L. Rawson, and Megan E. Williams. "Examining Hippocampal Mossy Fiber Synapses by 3D Electron Microscopy in Wildtype and Kirrel3 Knockout Mice". In: *eNeuro* 4.3 (May 1, 2017), ENEURO.0088–17.2017. ISSN: 2373-2822. DOI: 10.1523/ENEURO.0088-17.2017. URL: http://www.eneuro.org/content/4/3/ENEURO.0088-17.2017 (visited on 07/29/2018).