

# 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

© Copyright KU Leuven

Without written permission of the thesis supervisors and the author it is forbidden to reproduce or adapt in any form or by any means any part of this publication. Requests for obtaining the right to reproduce or utilize parts of this publication should be addressed to Faculteit Ingenieurswetenschappen, Kasteelpark Arenberg 1 bus 2200, B-3001 Heverlee, +32-16-321350.

A written permission of the thesis supervisors is also required to use the methods, products, schematics and programmes described in this work for industrial or commercial use, and for submitting this publication in scientific contests.

---

## Abstract – 30 Nov

---

---

# Contents

---

<b>Abbreviations</b>	<b>6</b>
<b>Symbols</b>	<b>7</b>
<b>1 Introduction – 16 Nov [9p]</b>	<b>9</b>
1.1 Closed-loop brain-computer interfaces [1p] . . . . .	9
1.2 Sharp wave-ripples [6p] . . . . .	9
1.3 Problem statement [1p] . . . . .	9
1.4 Thesis overview [1p] . . . . .	9
<b>2 Single-channel linear filtering – 9 Nov [10p]</b>	<b>10</b>
2.1 Offline labelling of SWR segments [3p] . . . . .	10
2.2 State of the art online SWR detectors [3p] . . . . .	10
2.3 Quantifying detector performance [4p] . . . . .	10
<b>3 Multi-channel linear filtering – 26 Oct [10p]</b>	<b>11</b>
3.1 Data-driven algorithms [1p] . . . . .	11
3.2 Linear signal-to-noise maximisation [2p] . . . . .	11
3.3 Result for SWR detection . . . . .	14
3.4 Combining space and time [1p] . . . . .	17
3.5 Regularization [1p] . . . . .	21
3.6 Multiple eigenvectors [2p] . . . . .	21
3.7 Spectral preprocessing [1p] . . . . .	21
<b>4 Nonlinear signal detection – 9 Nov [9p]</b>	<b>22</b>
4.1 Recurrent neural networks [2p] . . . . .	22
4.2 Optimization [3p] . . . . .	22
4.3 Regularization [2p] . . . . .	22
4.4 Channels [1p] . . . . .	22
4.5 Network size [1p] . . . . .	22
<b>5 Discussion – 23 Nov [4p]</b>	<b>23</b>
5.1 Comparing detectors [2p] . . . . .	23
5.2 Further work [2p] . . . . .	23
<b>6 Conclusions – 30 Nov [1p]</b>	<b>24</b>

<i>CONTENTS</i>	5
<b>References</b>	<b>25</b>
<b>Appendices</b>	<b>26</b>

---

# Abbreviations

---

<b>BPF</b>	Band-pass filter. See chapter 2.
<b>CA1</b>	“Cornu Ammonis”, subregion 1. Region in the hippocampus where voltages are recorded from (see ???).
<b>CA3</b>	“Cornu Ammonis”, subregion 3. Region in the hippocampus (see ???). CA3 sends many axons (called “Schafer collaterals”) to CA1.
<b>GEVal</b>	Generalized eigenvalue. See section 3.2.
<b>GEVec</b>	Generalized eigenvector. See section 3.2.
<b>IQR</b>	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.
<b>KDE</b>	Kernel density estimate.
<b>LFP</b>	Local field potential. The extracellular electric potential (see ??).
<b>RMS</b>	Root-mean-square. $\sqrt{\langle x_t^2 \rangle}$ for a signal $x_t$ .
<b>RNN</b>	Recurrent neural network. See chapter 4.
<b>SNR</b>	Signal-to-noise ratio. See section 3.2.
<b>SOTA</b>	State of the art. The algorithm currently used for SWR detection, namely an online single channel band-pass filter.
<b>SWR</b>	Sharp wave-ripple. The pattern in the LFP that we want to detect in real-time. See ??.

---

# 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.
$\odot$	Elementwise multiplication. (“Hadamard product”).
$\sigma(\cdot)$	Sigmoid ‘squashing’ function, $\mathbb{R} \rightarrow (0, 1)$ . $\sigma(x) = \frac{1}{1+\exp(-x)}$ .
$\tanh(\cdot)$	Hyperbolic tangent, $\mathbb{R} \rightarrow (-1, 1)$ . $\tanh(x) = 2 \sigma(x) - 1$ .

## 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 =  o_t $ for online linear filters; $n_t = \sigma(o_t)$ for the RNN’s of chapter 4.
$y_t$	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 versus all detections.
- $R$  Recall. Also known as sensitivity, hit rate, or true positive rate. The fraction of detected reference SWR segments versus all reference SWR segments.
- $F_\beta$  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  $\beta$  times as much importance to recall as to precision.” [1]
- $F_1$  F-score where recall and precision are weighted equally. For the common case where the  $PR$ -curve is concave,  $F_1(T)$  is maximal when  $P = R (= F_1)$ .



---

## Introduction – 16 Nov [9p]

---

1.1 Closed-loop brain-computer interfaces [1p]

1.2 Sharp wave-ripples [6p]

Description [2p]

Scientific importance [2p]

Biophysics [1p]

Closed-loop technology [1p]

1.3 Problem statement [1p]

1.4 Thesis overview [1p]

## Single-channel linear filtering – 9 Nov [10p]

---

### 2.1 Offline labelling of SWR segments [3p]

### 2.2 State of the art online SWR detectors [3p]

### 2.3 Quantifying detector performance [4p]

We often want to summarize the accuracy of a detector in a single number. We choose the maximum  $F_1$ -score of the detector as this number. As the harmonic mean of precision and recall, it has a relatively straightforward interpretation, especially in the common case when the PR-curve is concave: the maximum  $F_1$ -score is then the point where precision equals recall.

For example, an SWR detector with a max  $F_1$ -score of 93% detects 93% of all reference SWR segments, while 93% of its detections are correct (i.e. 7% of its detections are false positives).

---

## Multi-channel linear filtering – 26 Oct [10p]

---

### 3.1 Data-driven algorithms [1p]

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  $y_t^{\text{train}}$  which marks the presence of an SWR event in  $\mathbf{z}_t^{\text{train}}$ , for every discrete time sample  $t$ . We arbitrarily define  $y_t \in \{0, 1\}$ , 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 section 2.1. 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 section 2.1 to generate target labellings  $y_t^{\text{train}}$ .

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  $y_t^{\text{train}}$  well. Sections 3.2 and 4.2 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. Some example regularization methods are discussed in sections 3.5 and 4.3.

### 3.2 Linear signal-to-noise maximisation [2p]

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

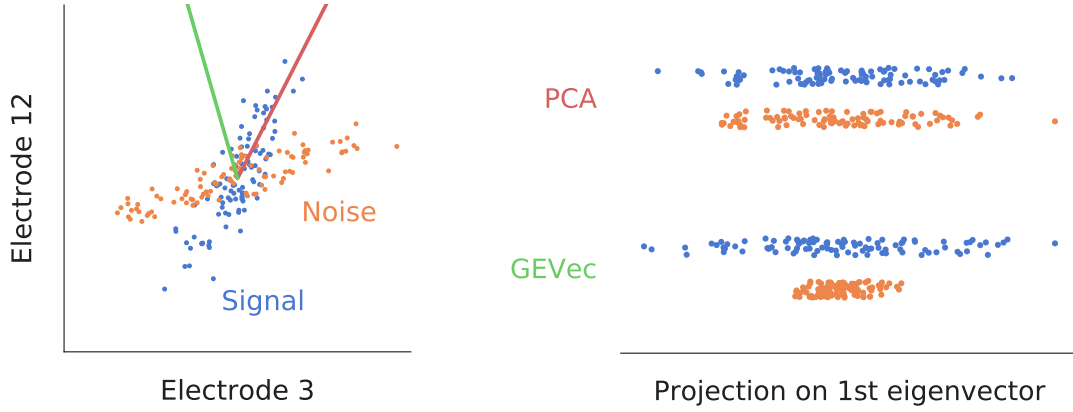


Figure 3.1: **Linear signal-to-noise maximisation.** Toy example to illustrate the generalized eigenvector approach to signal detection. *Left:* multi-channel time-series data plotted in ‘phase space’ (meaning without time axis), with blue dots representing samples where the signal was present, and orange 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.

$\mathbf{w} \in \mathbb{R}^C$  in channel (or electrode) space to project the samples  $\mathbf{z}_t \in \mathbb{R}^C$  on, so that the output signal

$$o_t = \mathbf{w}^T \mathbf{z}_t \quad (3.1)$$

has high variance (or power) during SWR events, and low variance outside them.<sup>1</sup> This principle is illustrated with a two-dimensional toy dataset in fig. 3.1. We can then detect SWR events using threshold crossings of the envelope of  $o_t$ , as discussed in section 2.2.

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  $\mathbf{z}_t^{\text{train}}$ ).

<sup>1</sup>We assume that the input signals are zero-mean, such that the power  $P$  of the output signal equals its variance:  $P_o = \langle o_t^2 \rangle = \langle (o_t - \mu_o)^2 \rangle = \text{Var}(o_t)$  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  $-0.02 \mu\text{V}$ , IQR  $0.13 \mu\text{V}$ . In comparison, the RMS-value of the recording is  $210 \mu\text{V}$ ).

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 (3.1) 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 3.1 (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  $\hat{\mathbf{w}}$  that maximises the variance of  $\mathbf{o}_S$  versus the variance of  $\mathbf{o}_N$ , i.e.

$$\begin{aligned}\hat{\mathbf{w}} &= \arg \max_{\mathbf{w}} \frac{\text{Var}(\mathbf{o}_S)}{\text{Var}(\mathbf{o}_N)} \\ &= \arg \max_{\mathbf{w}} \frac{\frac{1}{N_S} \mathbf{o}_S \mathbf{o}_S^T}{\frac{1}{N_N} \mathbf{o}_N \mathbf{o}_N^T} \\ &= \arg \max_{\mathbf{w}} \frac{\frac{1}{N_S} \mathbf{w}^T \mathbf{S} \mathbf{S}^T \mathbf{w}}{\frac{1}{N_N} \mathbf{w}^T \mathbf{N} \mathbf{N}^T \mathbf{w}}\end{aligned}\tag{3.2}$$

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

$$\mathbf{R}_{SS} = \frac{1}{N_S} \mathbf{S} \mathbf{S}^T \tag{3.3}$$

$$\mathbf{R}_{NN} = \frac{1}{N_N} \mathbf{N} \mathbf{N}^T \tag{3.4}$$

$\mathbf{R}_{SS} \in \mathbb{R}^{C \times C}$  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. (3.2), is thus equivalent to:

$$\hat{\mathbf{w}} = \arg \max_{\mathbf{w}} \frac{\mathbf{w}^T \mathbf{R}_{SS} \mathbf{w}}{\mathbf{w}^T \mathbf{R}_{NN} \mathbf{w}} \tag{3.5}$$

In appendix B, we show that the solution  $\hat{\mathbf{w}}$  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 so called *generalized eigenvector* (GEVec) for the ordered matrix pair  $(\mathbf{R}_{SS}, \mathbf{R}_{NN})$  when the following holds:

$$\mathbf{R}_{SS} \mathbf{w}_i = \lambda_i \mathbf{R}_{NN} \mathbf{w}_i, \tag{3.6}$$

for some scalar  $\lambda_i$ , which is called the *generalized eigenvalue* (GEVal) corresponding to  $\mathbf{w}_i$ . The largest scalar  $\lambda_1$  for which eq. (3.6) holds is the ‘first’ GEVal, and as mentioned before, the corresponding GEVec  $\mathbf{w}_1$  is the solution  $\hat{\mathbf{w}}$  to eq. (3.5).

Since the 1960's, numerically stable algorithms exist that solve the generalized eigenproblem eq. (3.6) [2]. 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.

### 3.3 Result for SWR detection

We divided the 34-minute long LFP recording into two datasets. 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  $\hat{\mathbf{w}}$ , as described in the preceding sections. The remaining 40% was used to evaluate this filter  $\hat{\mathbf{w}}$ , and to compare it to the state-of-the-art method (the single-channel online band-pass filter).<sup>2</sup>

Figure 3.2A 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.1. The elements of  $\hat{\mathbf{w}}$  (i.e. the filter weights) are visualized in fig. 3.2B.

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.1a, C.1b and C.1d. This results in false positive detections.

We also notice some premature detections (figs. C.1a and C.1c), 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. 3.3.

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

<sup>2</sup>The 60-40 division was chosen arbitrarily – under the constraint that there are sufficient amounts of both training and test data.

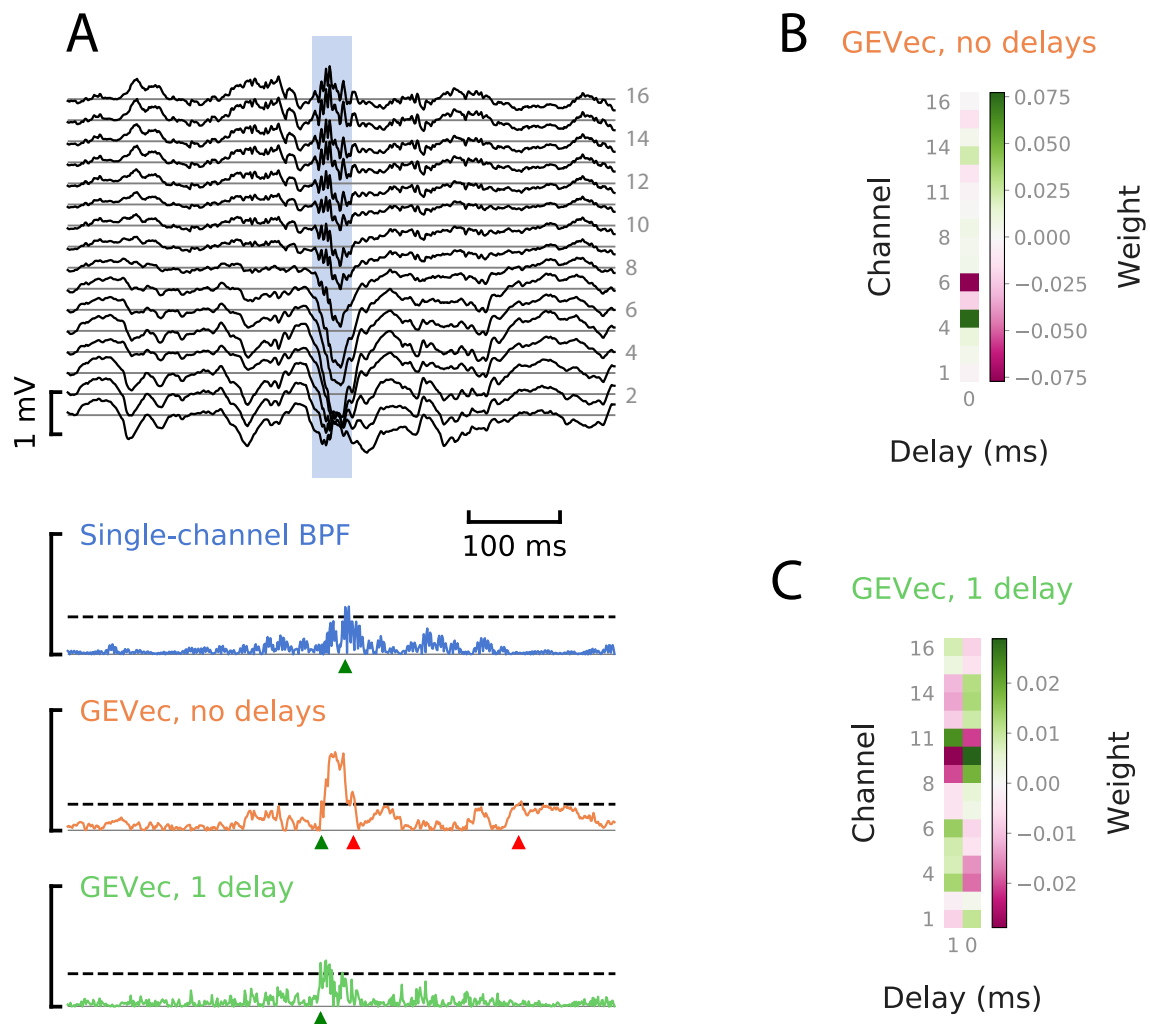


Figure 3.2: **Linear, SNR-maximising combinations of electrodes.**

**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  $\hat{\mathbf{w}}$  (i.e. the weights of the multichannel filter), for a purely spatial filter.

**C.** Generalized eigenvector  $\hat{\mathbf{w}}$  for a spatiotemporal filter with a one-sample delay.

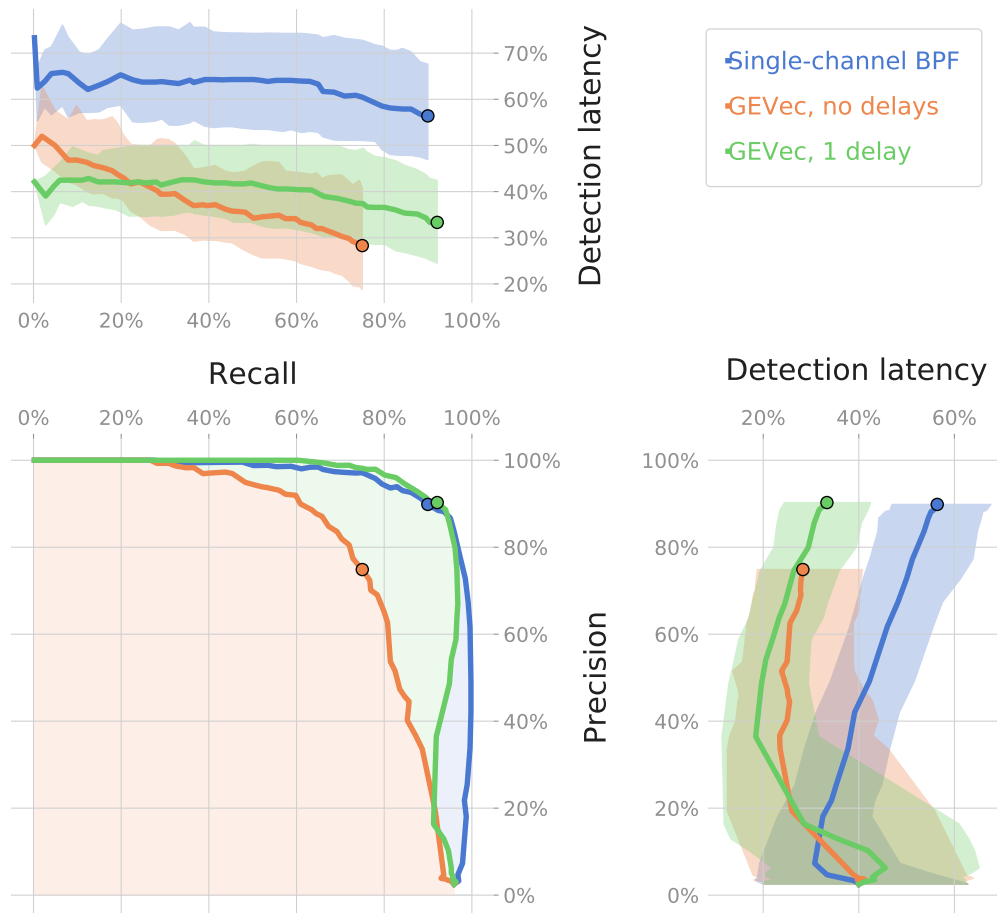


Figure 3.3: **Sensitivity, precision & latency tradeoffs**, for different linear filters & for a range of thresholds.

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.



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 positives). This strong decrease in precision is true over the entire  $PR$ -curve: see fig. 3.3.

### 3.4 Combining space and time [1p]

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. 3.3): 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. 3.3. 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 3.2C shows the GEVec  $\mathbf{w}^{\text{stack}}$ . Figure 3.2A and fig. C.1 show example output envelopes (green traces).

These results, particularly the visualized weights in fig. 3.2C, 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. 3.4) 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.

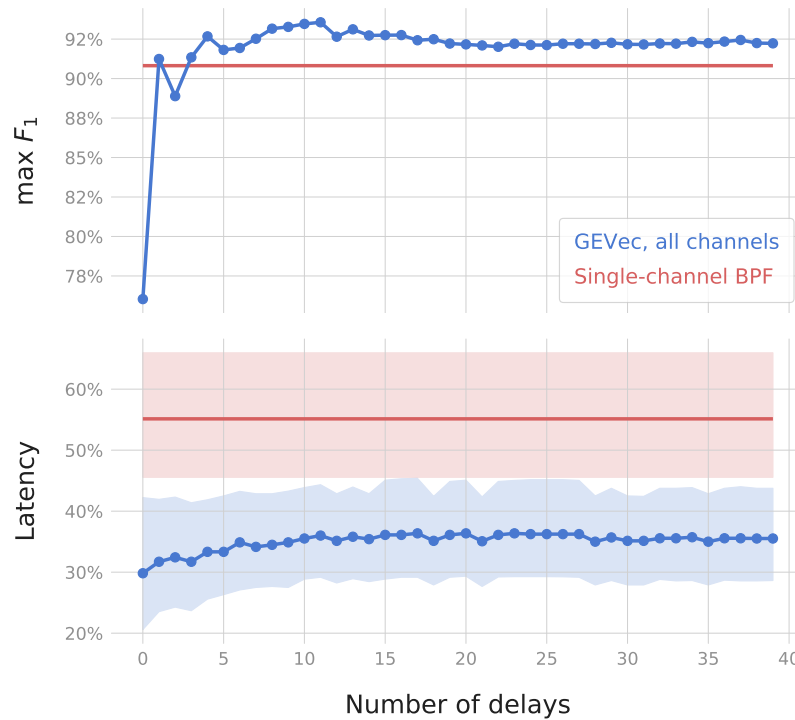


Figure 3.4: **Performance of the GEVec-based SWR detector, for different delay line lengths.** At the chosen 1000 Hz sampling rate, each delay corresponds to 1 ms. The red baseline is the state-of-the-art SWR detector. Detection latency is specified as a fraction of the duration of the corresponding 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.

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 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. 3.5) and latency (fig. 3.6).

We note some general trends. Using no extra delays results in low accuracy detectors, no matter which channels are included (top row of fig. 3.5). For most input channel

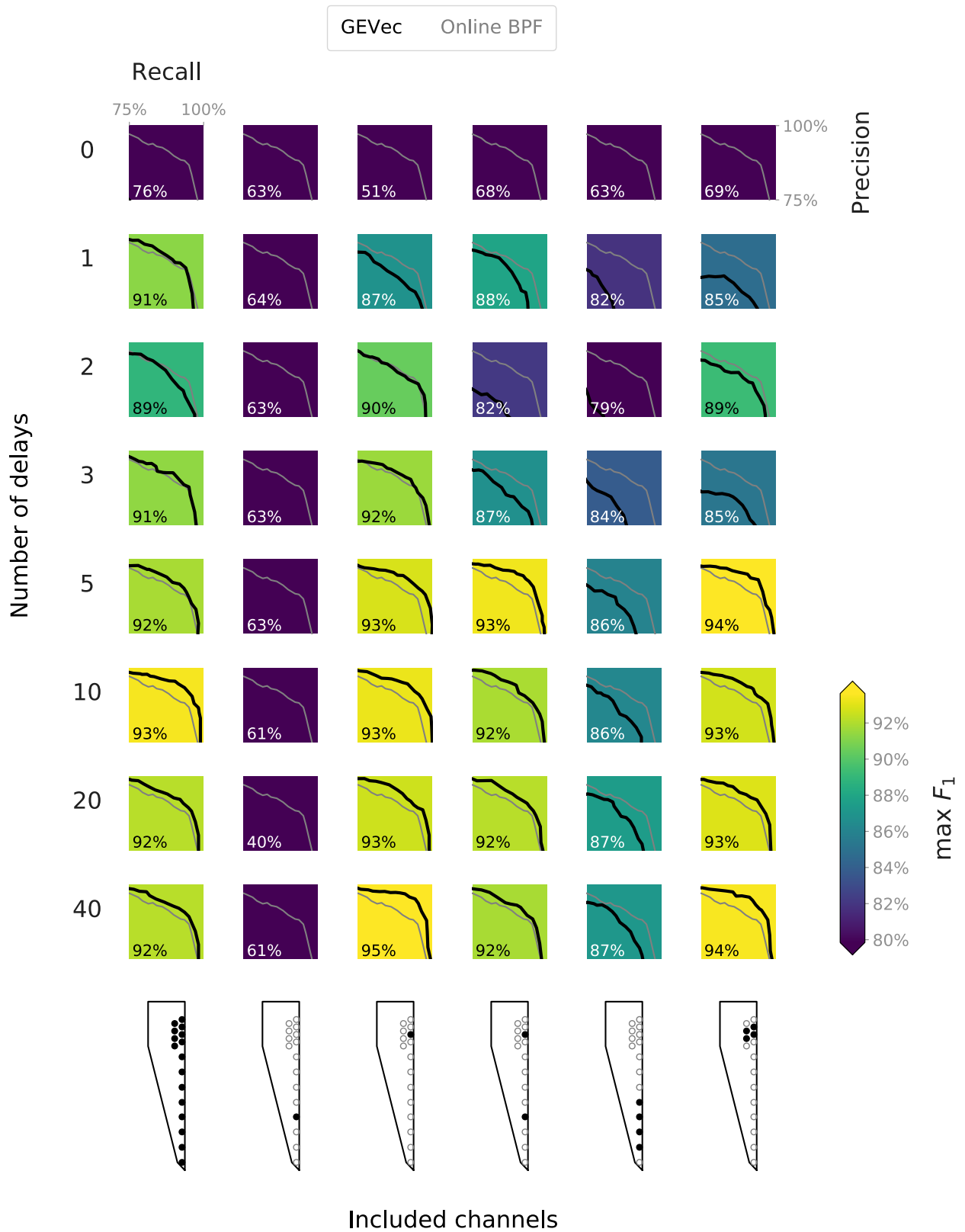


Figure 3.5: **Accuracy of GEVec-based SWR detectors**, for different combinations of active input channels and number of delays. Active input channels are marked in solid black on a schematic of the probe tip. (Height of schematic: 687  $\mu\text{m}$ ). The percentage and background color of each panel 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 3.6: **Latency of GEVec-based SWR detectors.** See fig. 3.5 for legend.

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 increasing detection latency.

The weight visualizations of fig. 3.2 show a relatively high weight for the stratum radiatum / sharp wave channels. Using only one such channel as input results in low accuracy detectors though, no matter how many delays are used (second column of figs. 3.5 and 3.6). Using four such channels (second to last row) 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, none of these four channels (numbers 2 to 5) display any noticeable ripple activity (fig. 3.2A).

**3.5 Regularization [1p]**

**3.6 Multiple eigenvectors [2p]**

**3.7 Spectral preprocessing [1p]**

## Nonlinear signal detection – 9 Nov [9p]

---

- 4.1 Recurrent neural networks [2p]
- 4.2 Optimization [3p]
- 4.3 Regularization [2p]
- 4.4 Channels [1p]
- 4.5 Network size [1p]

---

## Discussion – 23 Nov [4p]

---

5.1 Comparing detectors [2p]

5.2 Further work [2p]

---

## Conclusions – 30 Nov [1p]

---



---

# 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] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Fourth edition. The Johns Hopkins University Press, 2013.
- [3] Lloyd N. Trefethen and David III Bau. *Numerical Linear Algebra*. Siam, 1997.

---

# Appendices

---

<b>A</b>	<b>Data description</b>	<b>27</b>
<b>B</b>	<b>Generalized eigenvectors maximize signal-to-noise</b>	<b>28</b>
	Theorem . . . . .	28
	Proof . . . . .	28
<b>C</b>	<b>Supplemental figures</b>	<b>31</b>

## *Appendix A*

---

# Data description

---

## 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  $\hat{\mathbf{w}}$  to eq. (3.5)) 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 3.2).

This ratio of variances in eq. (3.5) 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:

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

We must then prove the following:

### Theorem

The generalized eigenvector  $\mathbf{w}_1$  corresponding to the largest generalized eigenvalue  $\lambda_1$  of  $(\mathbf{A}, \mathbf{B})$ , is also the vector  $\hat{\mathbf{w}}$  that maximises the generalized Rayleigh quotient  $r(\mathbf{w})$  of  $(\mathbf{A}, \mathbf{B})$ .

### Proof

As a first step, we will show that if  $\hat{\mathbf{w}}$  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  $\lambda_1$  of  $(\mathbf{A}, \mathbf{B})$  corresponds to the maximum of  $r(\mathbf{w})$ .

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

$$\nabla r(\hat{\mathbf{w}}) = \mathbf{0}. \quad (\text{B.2})$$

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

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

With eq. (B.2), we then have the following condition for our maximising vector  $\hat{\mathbf{w}}$ :

$$2\mathbf{A}\hat{\mathbf{w}}(\hat{\mathbf{w}}^T\mathbf{B}\hat{\mathbf{w}}) = 2\mathbf{B}\hat{\mathbf{w}}(\hat{\mathbf{w}}^T\mathbf{A}\hat{\mathbf{w}})$$

or

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

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

This is the generalized eigenvalue/eigenvector definition (eq. (3.6)) for  $\mathbf{w}_i = \hat{\mathbf{w}}$  and  $\lambda_i = r(\hat{\mathbf{w}})$ .

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

As the second step, we now show that  $r(\hat{\mathbf{w}})$  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 [3, 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  $\lambda_i$ . The maximum of a convex combination of one-dimensional points is obtained in the largest of these points. If  $\lambda_1$  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  $(\mathbf{A}, \mathbf{B})$ .

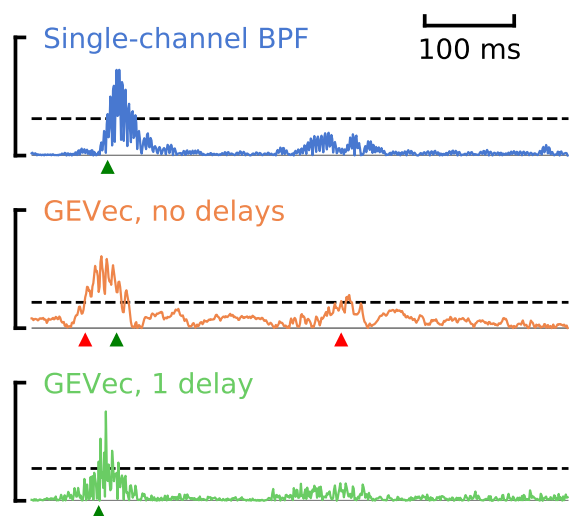
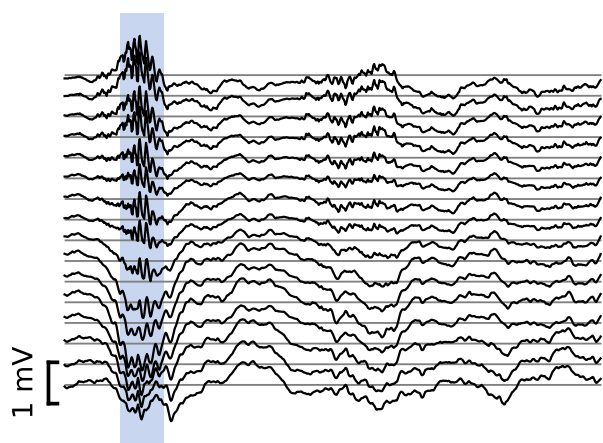
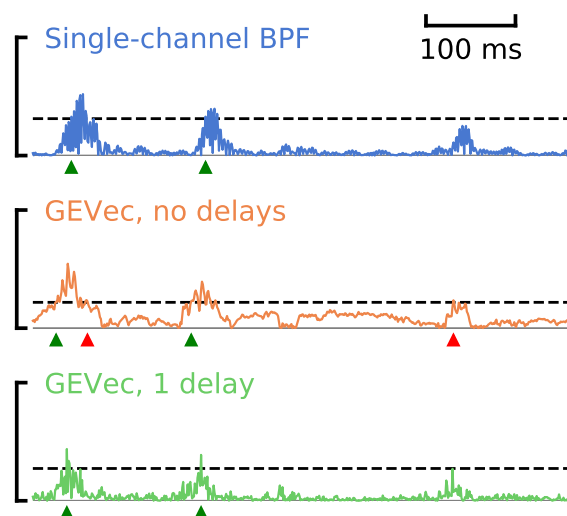
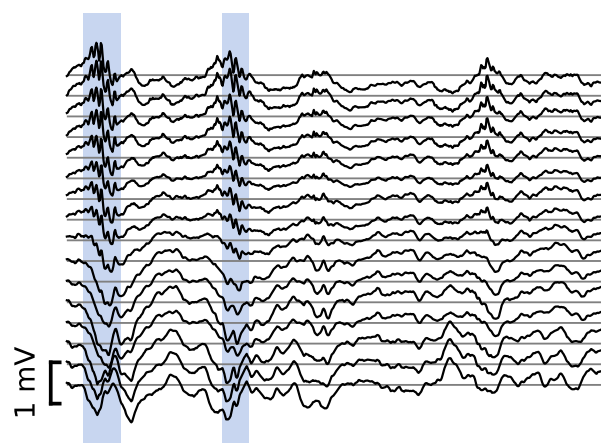
□

## *Appendix C*

---

# Supplemental figures

---

**a****b**



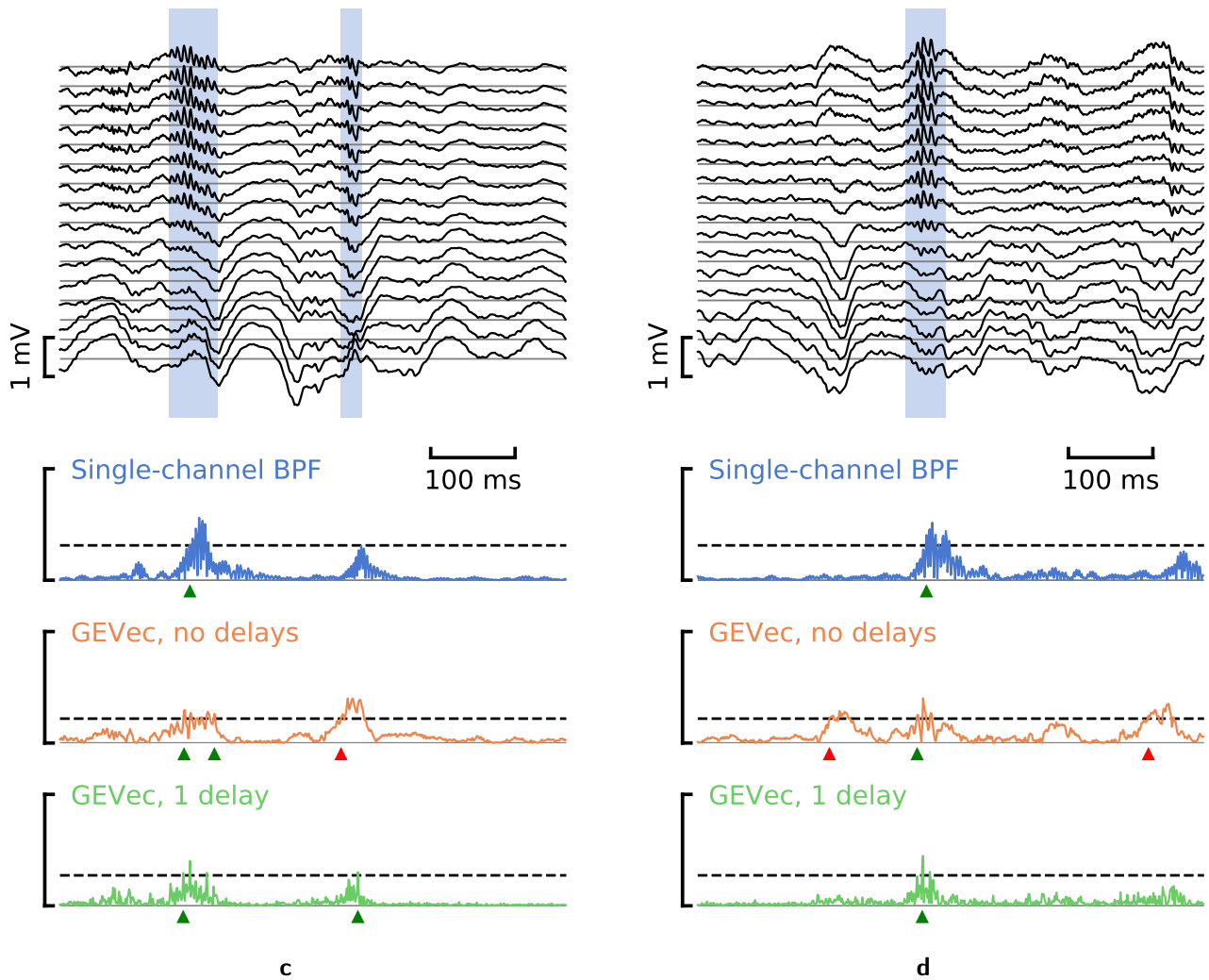


Figure C.1: Extracts from input data and corresponding linear filter output envelopes. See fig. 3.2 for legend.