Multi-Pattern Recognition Through Maximization of Signal-to-Peak-Interference Ratio With Application to Neural Spike Sorting

Jasper Wouters, Panagiotis Patrinos, Fabian Kloosterman, and Alexander Bertrand

***Abstract*—In this paper, we propose three novel linear filter design methods for use in a multi-pattern recognition task with overlapping patterns and strong peak interferers. The recognition is based on a linear filter-and-threshold approach, which is particularly interesting when the task has to be performed in a computationally constrained environment. The first method optimizes the signal-to-peak-interference ratio (SPIR) of the filter output, where the focus is on minimization of the post-filtering peak interference instead of the pre-filtering peak interference as in existing methods. The second and third method are convex approximations of the first method and are shown to be closely related to support vector machines, which establishes a natural link between SPIR-optimal filtering and the maximum margin matched filter. The proposed methods only require a template of the target patterns as prior knowledge and do not require the training data to be labelled. An extensive case study is presented in the context of neural spike sorting, in which the proposed approaches are shown to significantly outperform existing filter-and-threshold approaches for spike sorting.**

***Index Terms*—Biomedical signal processing, matched filters, pattern recognition.**

I. Introduction

A LONG-STANDING method used in pattern recognition is the matched filter [2]. The essence of the matched filter is that it maximizes the signal-to-noise ratio (SNR) at its output, possibly after a pre-whitening of the data. Qualitatively speaking, this means that a high instantaneous filter output power is desired when the pattern of interest, also referred to as the template, is present at the input of the filter, and a low output power is desired when only noise is present at the filter input. Such a filter enables the detection and localization of several occurrences of the pattern of interest in the input sequence, e.g., by comparing the filter output to a threshold. The matched filter-and-threshold approach is the optimal detector in the case of additive Gaussian noise [3]. Because of (and despite) its simplicity, the matched filter is still common practice in many application fields that require detection methods, e.g., telecommunications [4], [5], astrophysics [6], [7], and biomedical science [8], [9]. Because of its minimal computational footprint, the linear matched filter-and-threshold approach for pattern recognition is especially relevant for real-time pattern recognition in computationally constrained environments, e.g., for real-time brain activity pattern recognition in neural implants [10], [11].

In this work we focus on the recognition of multiple patterns in a signal, i.e., the detection and localization of patterns originating from multiple sources, while suppressing interfering sources and discriminating between the occurrence of patterns from different sources. A well-known example of such a multi-pattern recognition task is the neural spike sorting problem [12]–[15]. Spike sorting is the process of detecting neuronal action potentials or 'spikes' in an extracellular brain recording and assigning each detected spike to the neuron that presumably generated that spike. A naive approach for solving this problem consists of designing a matched filter for every distinct pattern source, i.e., for every detectable neuron. However, such a naive approach is known to produce filters that are incapable of properly discriminating between distinct pattern sources [16]. The lack of discriminating capability can be explained by the fact that the non-target pattern sources are usually only sparsely active in time in many applications. As such, those interfering sources are typically weakly weighted in the estimated noise statistics. To cope with this problem, signal-to-interference-plus-noise ratio (SINR) optimal filters have been proposed [16]–[18]. For such filters a trade-off can be made between the suppression of the noise floor and (known) interfering sources. This trade-off potentially gives rise to filters that have better discrimination properties. However, choosing the trade-off is non-trivial in practice and therefore there are no guarantees in terms of discrimination properties. The pattern recognition performance then strongly depends on a manual tuning by an expert during the actual filter design process.

Manuscript received March 17, 2020; revised August 30, 2020; accepted October 11, 2020. Date of publication October 28, 2020; date of current version November 10, 2020. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Vasanthan Raghavan. This work was carried out at the ESAT Laboratory of KU Leuven, in the frame of KU Leuven Special Research Fund projects C14/16/057, and the Research Foundation Flanders (FWO) project FWO G0D7516 N. This project has received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research, and innovation programme (grant agreement No 802895). The scientific responsibility is assumed by its authors. *(Corresponding author: Jasper Wouters.)*

Jasper Wouters, Panagiotis Patrinos, and Alexander Bertrand are with the Department of Electrical Engineering (ESAT), STADIUS Center for Dynamical Systems, Signal Processing and Data Analytics, KU Leuven, 3001 Leuven, Belgium (e-mail: jasper.wouters@esat.kuleuven.be; panos.patrinos@esat.kuleuven.be; alexander.bertrand@esat.kuleuven.be).

Fabian Kloosterman is with the Neuro-Electronics Research Flanders, Leuven, Belgium (e-mail: fabian.kloosterman@nerf.be).

Digital Object Identifier 10.1109/TSP.2020.3033973

1053-587X © 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on November 25,2020 at 12:58:36 UTC from IEEE Xplore. Restrictions apply.

Recently, signal-to-peak-interference ratio (SPIR) optimal filters were proposed to improve the discriminating capability of the matched filter [19]. The idea behind the SPIR-optimal filter is to only focus on suppressing inputs that cause interfering peaks in the filter output. As such, no degrees of freedom are allocated on trying to attenuate inputs that are not contributing to spurious output peaks. When training such filters from a classical matched filtering point of view, interference segments have to be identified prior to the filter design based on heuristics [19], which suffer from errors. Furthermore, even if these errors would not exist, all the degrees of freedom in such filter designs are spent on maximally suppressing the interference peaks that are observed prior to the filter design, while the resulting filtering operation itself could possibly introduce new spurious peaks at its output. Therefore, we argue that these methods actually only use a proxy for the SPIR based on statistics collected at the filter *input*, and hence do not truly optimize the SPIR at the filter *output*. A naive attempt to fix this could be to re-compute the heuristic in an iterative fashion based on knowledge of the output of the filter design in the previous step. However, there would be no convergence guarantees towards a SPIR-optimal filter.

In this work we propose three optimal filter design methods that optimize the SPIR of the filter output directly without labelling interfering segments at the pre-filtering stage. Therefore, these filters are anticipated to be more suitable for multi-pattern recognition when using a filter-and-threshold approach. The first method minimizes a power loss function, which is a generalization of the classical matched filter. The second method is a convex approximation of the former to avoid problems with local minima. The third method is an alternative convex approximation which exploits additional information that is available from the convex constraint. Besides guarantees on global optimality, the convex reformulation provides interesting parallels with support vector machines (SVM). These parallels provide additional insights into the filter design mechanics and its resulting robustness, leading to a maximum margin matched filter interpretation.

The outline of the paper is as follows. We formalize the problem statement of multi-pattern recognition in Section II and review matched filtering approaches that have been applied to this problem in Section III. In Section IV we introduce the novel non-convex SPIR-optimal filter design method and in Section V we present two convex reformulations thereof. In Section VI we review an alternative regularization approach for the proposed filter design methods, after which we benchmark the proposed methods in an extensive case study of neural spike sorting in Section VII. Finally, in Section VIII we further discuss the main results and draw conclusions.

II. Problem Statement

Consider the N-dimensional signal $\mathbf{x}[k] \in \mathbb{R}^N$ at sample time k which could contain the samples collected at N sensors at time k (spatial pattern) or a delay line of N samples of a single sensor (temporal pattern), or a combination of both (spatio-temporal pattern). Without loss of generality we assume that $E\{\mathbf{x}\} = \mathbf{0}$, i.e., the signal is zero-mean along its different dimensions. The signal $\mathbf{x}[k]$ is modelled as

$$\mathbf{x}[k] = \sum_{i=1}^{M} \mathbf{p}_i[k] + \mathbf{n}[k], \tag{1}$$

where the M different $\mathbf{p}_i$ are so-called (sparse) pattern sources which have an 'on-off' characteristic and where $\mathbf{n}$ is the noise term absorbing all sources that are not pattern sources. A pattern source $\mathbf{p}_i[k]$ is characterized by the deterministic pattern $\boldsymbol{\pi}_i \in \mathbb{R}^N$ and is defined in the following way:

$$\mathbf{p}_i[k] = \sum_{a \in \mathcal{A}_i} \delta[k-a] * \boldsymbol{\Pi}_i[k], \tag{2}$$

with $\delta[k]$ being the Kronecker delta, i.e., $\delta[k] = 1$ if $k = 0$ and $\delta[k] = 0$ for all $k \neq 0$ and $*$ the convolution operator. $\mathcal{A}_i$ is the set of sparse pattern occurrence times for the pattern source i. $\boldsymbol{\Pi}_i[k]$ is a finite support signal defined such that $\boldsymbol{\Pi}_i[0] = \boldsymbol{\pi}_i$ where $\boldsymbol{\pi}_i$ is the known deterministic pattern of interest. The content of $\boldsymbol{\Pi}_i[k]$ for $k \neq 0$ depends on the context. For purely spatial patterns, $\boldsymbol{\Pi}_i[k] = 0$ for $k \neq 0$, in which case (2) reduces to

$$\mathbf{p}_i[k] = \sum_{a \in \mathcal{A}_i} \delta[k-a]\, \boldsymbol{\pi}_i. \tag{3}$$

For cases where $\mathbf{x}[k]$ consists of a sample delay line, $\boldsymbol{\Pi}_i[k]$ will contain time-shifted versions of $\boldsymbol{\pi}_i$ at $k \neq 0$. For example, consider a saw-tooth pattern $\boldsymbol{\pi}_i = [1\ 2\ 3\ 4\ 5]^T \in \mathbb{R}^5$, where the different dimensions in the pattern represent five consecutive time samples from a single sensor. Because of the presence of the temporal delay structure in the pattern, (2) depends for its correctness on $\boldsymbol{\Pi}_i$, which can be seen as a Hankel-like expansion of $\boldsymbol{\pi}_i$:

$$\begin{aligned} &\left[\boldsymbol{\Pi}_i[-4]\ \cdots\ \boldsymbol{\Pi}_i[-1]\ \boldsymbol{\Pi}_i[0]\ \boldsymbol{\Pi}_i[1]\ \cdots\ \boldsymbol{\Pi}_i[4]\right] \\ &\quad = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 0 & 0 & 1 & 2 & 3 & 4 & 5 & 0 \\ 0 & 0 & 1 & 2 & 3 & 4 & 5 & 0 & 0 \\ 0 & 1 & 2 & 3 & 4 & 5 & 0 & 0 & 0 \\ 1 & 2 & 3 & 4 & 5 & 0 & 0 & 0 & 0 \end{bmatrix}. \end{aligned} \tag{4}$$

In this example, $\boldsymbol{\Pi}_i[k] = \mathbf{0}$ for $k < -4$ or $k > 4$.

When solving the multi-pattern recognition task, one wants to retrieve $\mathcal{A}_i$ for the pattern source of interest i, which we will refer to in this context as the *target* pattern source, given $\mathbf{x}[k]$ and the target pattern $\boldsymbol{\pi}_i$ or an estimate thereof. In this work a linear filter-and-threshold approach is taken for estimating $\mathcal{A}_i$ from the signal:

$$\hat{\mathcal{A}}_i = \left\{ k \mid \left(\mathbf{f}_i^T \mathbf{x}[k]\right)^2 > T_i \right\}, \tag{5}$$

where $\mathbf{f}_i \in \mathbb{R}^N$ denotes the linear filter, and T_i is a user-defined detection threshold. For every target pattern source i, a separate linear filter has to be designed and applied to the data in a parallel fashion where the other pattern sources are then treated as interferers, which should not cause the filter output to exceed the threshold T_i. In the remainder of this work we will focus on the design of such linear filters $\mathbf{f}_i$ that enable *filter-and-threshold-based pattern recognition* as given by (5).

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on November 25,2020 at 12:58:36 UTC from IEEE Xplore. Restrictions apply.

III. Filter Design Review

In this section we review several filter design methods that have been used in the context of filter-and-threshold-based pattern recognition. The different filter design approaches are all presented from an optimization point of view. This uniform presentation has the advantage that the different methods can be easily compared among each other.

A. Matched Filtering

A classical filter design method that is used for filter-and-threshold-based pattern recognition is the matched filter, which is also referred to as the SNR-optimal filter. The matched filter $\mathbf{f}_i^{\text{MF}}$ is given by the following optimization problem:

$$\mathbf{f}_i^{\text{MF}} \sim \underset{\mathbf{f}}{\text{argmax}} \frac{(\mathbf{f}^T\boldsymbol{\pi}_i)^2}{\frac{1}{|\mathcal{T}|}\sum_{k\in\mathcal{T}}(\mathbf{f}^T\mathbf{n}[k])^2}, \tag{6}$$

where the numerator contains the filter output power when the target pattern $\boldsymbol{\pi}_i$ is present at the input and the denominator contains the average filter output power in response to the noise term $\mathbf{n}[k]$, effectively optimizing the output SNR. The training data is represented by $\mathcal{T}$ which is the set of sample times that are available during training of the filters. Note that the denominator of (6) is defined in terms of the noise covariance estimate, i.e., $\frac{1}{|\mathcal{T}|}\sum_{k\in\mathcal{T}}(\mathbf{f}^T\mathbf{n}[k])^2 = \frac{1}{|\mathcal{T}|}\sum_{k\in\mathcal{T}}\mathbf{f}^T\mathbf{n}[k]\mathbf{n}[k]^T\mathbf{f} = \mathbf{f}^T\hat{\mathbf{R}}_{\mathbf{nn}}\mathbf{f}$. We assume that sufficient training data is available such that the covariance estimate is close to the true covariance. Also note that $\mathbf{f}_i^{\text{MF}}$ is only defined up to an arbitrary scaling, which is why we use '$\sim$' instead of '$=$' in (6).

We can also consider all pattern sources, except for target pattern occurrences, to be noise sources. When defining all non-target pattern sources and shifted versions of the target pattern as noise, we can adjust (6) to be

$$\mathbf{f}_i^{\text{MF}} \sim \underset{\mathbf{f}}{\text{argmax}} \frac{(\mathbf{f}^T\boldsymbol{\pi}_i)^2}{\frac{1}{|\mathcal{T}\setminus\mathcal{A}_i|}\sum_{k\in\mathcal{T}\setminus\mathcal{A}_i}(\mathbf{f}^T\mathbf{x}[k])^2}. \tag{7}$$

Note that this filter design requires complete knowledge about the target pattern source activity $\mathcal{A}_i$ in the training data, rendering it of lesser practical significance. However, it was shown in [19] that the presence of the pattern of interest in the denominator does not affect the filter solution, as such (7) can be rewritten as

$$\mathbf{f}_i^{\text{MF}} \sim \underset{\mathbf{f}}{\text{argmax}} \frac{(\mathbf{f}^T\boldsymbol{\pi}_i)^2}{\frac{1}{|\mathcal{T}|}\sum_{k\in\mathcal{T}}(\mathbf{f}^T\mathbf{x}[k])^2}, \tag{8}$$

which simplifies the filter design process greatly, by not having to compensate for the target pattern source and training the filter on the original acquired data immediately without knowledge of the activation times of the target pattern in the training data. It is well known that a closed-form solution to (8) is [20]:

$$\mathbf{f}_i^{\text{MF}} = \hat{\mathbf{R}}_{\mathbf{xx}}^{-1}\boldsymbol{\pi}_i, \tag{9}$$

where $\hat{\mathbf{R}}_{\mathbf{xx}} = \frac{1}{|\mathcal{T}|}\sum_{k\in\mathcal{T}}\mathbf{x}[k]\mathbf{x}[k]^T$ is the estimate of the signal covariance matrix. This solution is equivalent to a pre-whitening of the data followed by a matched filtering with the pre-whitened target pattern. In the special case where $\mathbf{x}$ is white over its different dimensions, i.e., $\hat{\mathbf{R}}_{\mathbf{xx}} = \mathbf{I}$ with $\mathbf{I}$ the identity matrix, the matched filter boils down to its simplest solution $\mathbf{f}_i^{\text{MF}} = \boldsymbol{\pi}_i$.

In many practical applications $\hat{\mathbf{R}}_{\mathbf{xx}}$ is ill-conditioned, or even rank deficient. Therefore, a diagonal loading can be applied to the covariance matrix estimate prior to inversion [21]:

$$\mathbf{f}_i^{\text{MF}} = \left(\hat{\mathbf{R}}_{\mathbf{xx}} + C\mathbf{I}\right)^{-1}\boldsymbol{\pi}_i, \tag{10}$$

where C is a hyper-parameter that controls the condition number of the resulting sum matrix.

We will now transform (8) in an equivalent problem that will serve as the basis for the later algorithms introduced in this paper. Note that the maximum of (8) can be found by fixing the numerator to a constant value K and minimizing the denominator (note that this also resolves the non-uniqueness problem related to optimizing the ratio (8)). When also including the diagonal loading introduced in (10), we arrive to the following equivalent constrained optimization problem

$$\begin{aligned}\mathbf{f}_i^{\text{MF}} = \underset{\mathbf{f}}{\text{argmin}}\ & \frac{1}{|\mathcal{T}|}\sum_{k\in\mathcal{T}}(\mathbf{f}^T\mathbf{x}[k])^2 + C\|\mathbf{f}\|_2^2 \\ \text{subject to}\ & (\mathbf{f}^T\boldsymbol{\pi}_i)^2 = K \qquad ,\end{aligned} \tag{11}$$

where $K \in \mathbb{R}_{>0}$ is an arbitrary strictly positive number, that controls the pattern of interest's output response power. Note that the diagonal loading in (10) is equivalent to a Tikhonov regularization in the above reformulation. Hence, the diagonal loading in (10) pushes the solution to the minimal norm solution for $C \to \infty$. This minimal norm solution is the target pattern $\boldsymbol{\pi}_i$ itself, as can be straightforwardly seen from (10). Although such matched filters have been used in the context of filter-and-threshold-based pattern recognition, they were shown to be insufficiently discriminative [16] for the target pattern, because they typically focus on minimizing the baseline noise, but fail to attenuate sparse peaks, e.g., caused by the other pattern sources or peak-interference sources, hence leading to false positive threshold crossings.

B. Matched Filtering With Source Weighting

In an attempt to improve the filter-and-threshold-based pattern recognition performance, variants of the matched filter have been proposed [16]–[18], where the different additive sources in (1) are weighted individually. By using separate weights for the different sources, the aim is to improve the discriminating capabilities of the filter. Such a weighted matched filter $\mathbf{f}_i^{\text{WMF}}$ results from the following optimization problem (cf. (11)):

$$\begin{aligned}\mathbf{f}_i^{\text{WMF}} = \underset{\mathbf{f}}{\text{argmin}}\ & \frac{1}{|\mathcal{T}|}\sum_{k\in\mathcal{T}}\Big(w_{\mathbf{n}}\cdot(\mathbf{f}^T\mathbf{n}[k])^2 \\ & + \sum_{j=1}^{M} w_{\mathbf{p}_j}\cdot(\mathbf{f}^T\mathbf{p}_j[k])^2\Big) + C\|\mathbf{f}\|_2^2 \\ \text{subject to}\ & (\mathbf{f}^T\boldsymbol{\pi}_i)^2 = K,\end{aligned} \tag{12}$$

where $w_{\mathbf{n}}$ is a single scalar weight associated with the average filter output power in response to the noise source $\mathbf{n}$. $w_{\mathbf{p}_j}$ are the single scalar weights associated with the average filter output power in response to their respective pattern sources $\mathbf{p}_j$. Note that this method assumes complete knowledge about all the pattern instances that are present in the training data and requires an estimate of the covariance matrix of the noise $\mathbf{n}$,

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on November 25,2020 at 12:58:36 UTC from IEEE Xplore. Restrictions apply.

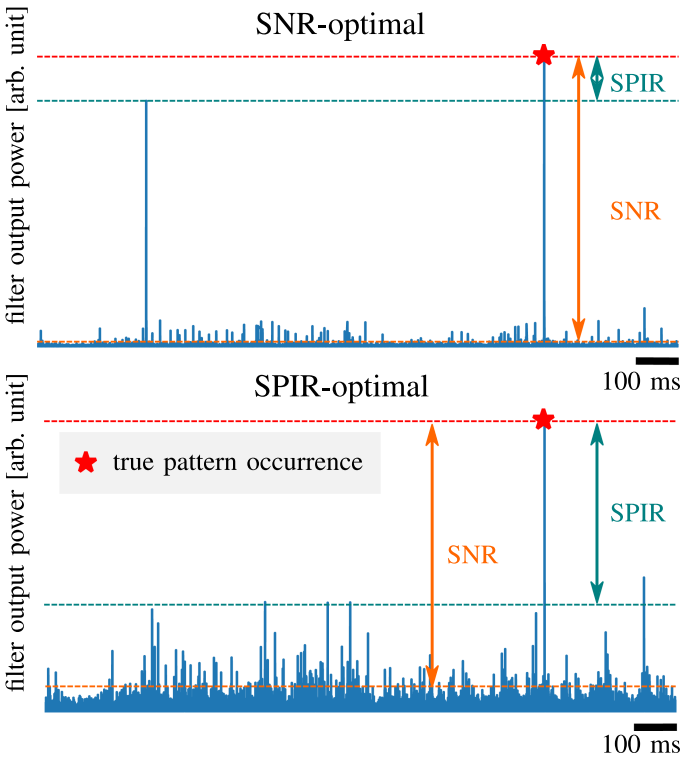


Fig. 1. Qualitative comparison between an SNR- and SPIR-optimal filter output. The SPIR-optimal filter has an increased SPIR compared to the SNR-optimal one, at the acceptable cost of a decrease in SNR, enabling a more robust filter-and-threshold-based pattern recognition for the SPIR-optimal filter output.

which can be estimated during segments where none of the pattern sources are active. Furthermore, no concrete methods are proposed for choosing the values of the different weights, as such these weights should be seen as hyper-parameters that should be optimized based on a cross-validation scheme, leading to an $(M+2)$-dimensional grid-search, which quickly becomes infeasible for increasing M. While (12) is an interesting generalization of (11), all the aforementioned issues make it hard to use it in practice.

C. SPIR-Based Filtering

The concept of SPIR-optimal filter design was proposed in [19] in an attempt to improve the discriminating capabilities of the filters $\mathbf{f}_i$ with respect to their target patterns $\boldsymbol{\pi}_i$. The main idea is to only allocate degrees of freedom in the filter design to signal segments that cause spurious peaks in the output during inactivity of the target pattern. SPIR-optimal filter design is also intended as a practical method, i.e., it does not require complete prior knowledge about different pattern sources in the mixture (as in Subsection III-B) and works with a minimal set of hyper-parameters.

The qualitative differences between the SNR-optimal and SPIR-optimal filter are highlighted in Fig. 1. The top graph shows the SNR-optimal filter output power, where two major peaks can be distinguished, of which only one peak reflects the occurrence of the target pattern. The bottom graph shows the SPIR-optimal filter output power. Here, only one dominant peak is present, while the previous spurious peak has been attenuated at the cost of an increase in overall noise power. The SPIR-optimal filter output shown in the bottom graph is suitable for filter-and-threshold-based pattern recognition, whereas the top graph does not allow for a robust filter-and-threshold-based discrimination between target and non-target pattern occurrences.

The SPIR-based filter design proposed in [19] results from solving the following optimization problem:[1]

$$\mathbf{f}_i^{\mathrm{SB}} = \underset{\mathbf{f}}{\operatorname{argmin}} \frac{1}{|\mathcal{T}|} \sum_{k \in \mathcal{T}} w_k \cdot (\mathbf{f}^T \mathbf{x}[k])^2 + C\|\mathbf{f}\|_2^2$$
$$\text{subject to} \quad (\mathbf{f}^T \boldsymbol{\pi}_i)^2 = K, \qquad , \qquad (13)$$

which is a generalization of (11) by the introduction of the binary sample weights $w_k \in \{0,1\}$, which effectively select a subset of training samples when designing the filter.

In [19], the following heuristic for the estimation of w_k was proposed (slightly simplified here for illustrative purposes):

1) Design a classical matched filter according to (11).
2) Apply the matched filter to the data and identify segments for which the output power exceeds γK with $\gamma \in [0,1]$.
3) For every sample time k for which the output power threshold is exceeded, set $w_k = 1$. Otherwise, $w_k = 0$.
4) Compute the SPIR-based filter (13).

The method was shown capable of outperforming the matched filter in a simple threshold-based pattern recognition task. However, this heuristic procedure (as well as any other alternative heuristic) may make wrong decisions in selecting the peak interference segments. Furthermore, we will argue in the next section that – even without such errors – the filter design (13) is not truly SPIR-optimal when the weights w_k are a-priori fixed.

IV. Non-Convex SPIR-Optimal Filtering

A. Definition of SPIR-Optimal Filtering

The filter design in (13) does not optimize the SPIR of the filter output directly, i.e., there is no mechanism that prevents samples $\mathbf{x}[k]$ with $w_k = 0$ from generating a peak response in the filter output. Vice versa, not every interfering peak at the input of a matched filter will automatically generate a large peak at the filter output (e.g. in the case of interferers that have a template that is near-orthogonal to the target template), such that a weight $w_k = 1$ might be unnecessary for many of the peak interferers that are visible at the filter input. In principle, the optimal binary weights in (13) should be determined based on the interference peaks in the *output* of the filter $\mathbf{f}$ itself, which creates a chicken-and-egg situation ($\mathbf{f}$ depends on the weights w_k, but the weights depend on $\mathbf{f}$). A straightforward – yet naive – fix would be to compute (13) multiple times, where the weights w_k are every time determined by the peak interference in the output of the previous filter. However, this would require a human expert to terminate this iteration, since there are no guarantees on convergence towards a workable solution of this ad-hoc process. Furthermore, none of the resulting filters would be truly SPIR-optimal as the selection of interfering segments would

[1]Note that the optimization problem (13) can be written as a maximization of a ratio, similar to (8)–(11), which explains the term SPIR.

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on November 25,2020 at 12:58:36 UTC from IEEE Xplore. Restrictions apply.

always be based on a different filter than the one eventually used. To cope with these fundamental limitations we propose a truly SPIR-optimal filter design method, which inherently and automatically chooses the weights w_k based on the peak interference in the filter *output* signal. This makes the weights dependent on $\mathbf{f}$ itself, thereby creating an interaction between the filter coefficients in $\mathbf{f}$ and the weights w_k that define the filter. By including this chicken-and-egg situation in the optimization problem itself, we remove the need for an ad-hoc iterative design, while convergence towards a SPIR-optimal filter can be guaranteed.

To acquire a SPIR-optimal filter $\mathbf{f}_i^{\text{SO}}$, we replace the prior weights w_k in (13) by a weighting function g that depends directly on the filter output power:

$$\boxed{\begin{aligned}\mathbf{f}_i^{\text{SO}} = \underset{\mathbf{f}}{\text{argmin}}\ & \frac{1}{|\mathcal{T}|}\sum_{k\in\mathcal{T}} g\left((\mathbf{f}^T\mathbf{x}[k])^2\right)(\mathbf{f}^T\mathbf{x}[k])^2 + C\|\mathbf{f}\|_2^2 \\ \text{subject to}\ & (\mathbf{f}^T\boldsymbol{\pi}_i)^2 = K\end{aligned}}, \tag{14}$$

where the weighting function $g : \mathbb{R} \to \mathbb{R}$ is given by

$$g(s) = \frac{1}{1 + e^{-(s-\gamma K)}}, \tag{15}$$

where $\gamma \in [0, 1]$. This weighting function is a shifted sigmoid and is a differentiable approximation of the step function. This function approximately implements a binary weighting, which is in line with the use of binary weights in (13). As a result, samples with a power smaller than γK at the output of the filter $\mathbf{f}$ are not treated as peak interference and are ignored in the filter design of $\mathbf{f}$. Note that K in (15) is the same as in the constraint of (14), i.e., γK determines the fraction of the output power in response to the target pattern. We will explain later on (Subsection IV-C) how γ can be chosen to steer the filter design towards a predefined desired SPIR.

By making use of a weighting function g rather than prior weights w_k, the resulting filter is truly SPIR optimal. The price that one has to pay is that as opposed to (11), (12) and (13), the filter design method in (14) has no closed-form solution. Therefore, one has to use a local optimization solver, to obtain a non-convex SPIR-optimal filter. Depending on the solver that is used, convergence to a solution can be guaranteed. However, because (14) is non-convex, the resulting filter might only coincide with a local minimum of the objective function. In Section V, we will propose a convex approximation of (14) to avoid such local minima, and which allows to use efficient solvers for convex optimization.

Note that only limited prior knowledge is needed for designing a SPIR-optimal filter. Only the target pattern $\boldsymbol{\pi}_i$ is required (as opposed to the method described in Section III-B). Furthermore, the filter design is controlled by two hyper-parameters: the interpretable interference fraction γ and the regularization hyper-parameter C. In Section VI we will introduce an alternative regularization scheme, that leads to a regularization hyper-parameter that can be more easily tuned and interpreted. Note that K is not truly a hyper-parameter, but an arbitrary constant that has little influence on the SPIR itself, as long as K is chosen large enough, as explained in the next subsection.

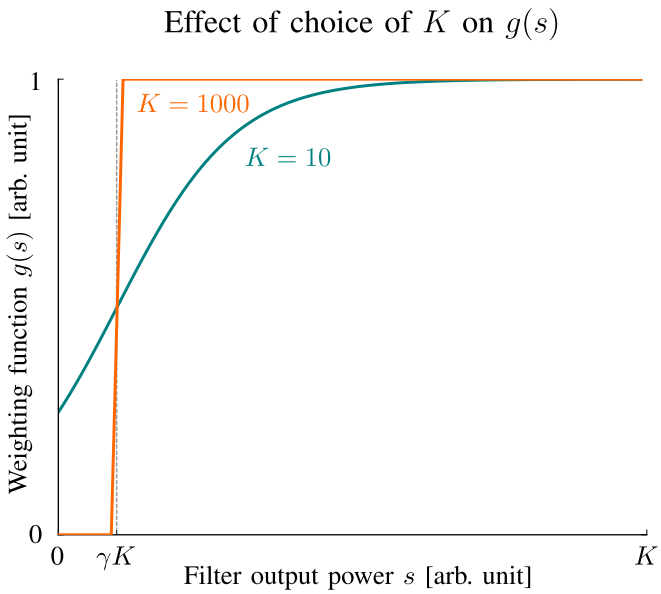


Fig. 2. The shifted sigmoid weighting function only 'behaves' as a step function for values of K that are sufficiently large. For $K = 10$ (green) and $\gamma = 0.1$, the shifted sigmoid does not reach zero for any $s \in [0, \gamma K]$. For $K = 1000$ (orange) and $\gamma = 0.1$ the shifted sigmoid is a good approximation to a step function.

B. Choice of K

We did not include an extra hyper-parameter in (15) to control the slope of the shifted sigmoid. This can be implicitly controlled through the parameter K, which is an arbitrary constant (see also (11)) defining the filter output power in response to the target pattern. By choosing a large K, the rise time of the sigmoid can be made arbitrarily short, i.e., relative[2] to the length of the interval $[0, K]$ that determines the scale of the output signal. If K is large enough, the sigmoid will sufficiently 'behave' as a step function (in practice $K = 1000$ works well, $K = 10$ is too small as the unshifted sigmoid raises from 0 to 1 in the interval $[-5, 5]$, as is shown in Fig. 2).

C. Bounds on γ

As was mentioned before, γ is bounded by nature, i.e., $\gamma \in [0, 1]$. Here, we will further narrow down this interval. In all aforementioned filter design optimization problems, we explicitly control the filter output power for the target pattern to be K. Therefore, we are able to define an explicit interference threshold γK in (15). By combining both the target power and interference threshold, we can define the desired SPIR (in dB):

$$\text{desired SPIR} = 10\log\frac{K}{\gamma K} = -10\log\gamma. \tag{16}$$

We refer to the above metric as *desired* SPIR, because there is no guarantee that all interferers are pushed below the interference threshold during the optimization process due to the finite degrees of freedom in the filter design. From Fig. 1, it is clear that the SPIR will by definition always be smaller than the SNR.

[2]Note that the rise time is fixed in absolute sense, but should be viewed relative to the interval $[0, K]$.

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on November 25,2020 at 12:58:36 UTC from IEEE Xplore. Restrictions apply.

Since the matched filter (11) is SNR-optimal, we thus require that the desired SPIR $\leq$ SNR_{MF}, where SNR_{MF} is the SNR at the output of the matched filter. From this inequality a lower bound on γ can be derived:

$$\gamma \geq \frac{1}{10^{\frac{\text{SNR}_{\text{MF}}}{10}}}, \tag{17}$$

where SNR_{MF} is in dB.

Although no theoretical narrowing down of the upper bound is provided, it goes without saying that a γ close to one is considered bad practice, because this might lead to very low SPIR as can be seen from (16). This low SPIR will likely deteriorate the filter-and-threshold-based pattern recognition performance. Therefore, without further proof, we suggest as a pragmatic upper bound $\gamma = 0.5$.

V. Convex SPIR-Optimal Filtering

In this section we propose a convex reformulation of (14). A convex reformulation has the advantage that convergence can be guaranteed to the *global* optimum of the objective function rather than to a local optimum and that efficient convex optimization solvers can be used.

A. Definition of the Convex SPIR-Optimal Filter

We define the convex SPIR-optimal filter $\mathbf{f}_i^{\text{CSO'}}$ as the solution to the following optimization problem:

$$\boxed{\begin{aligned}\mathbf{f}_i^{\text{CSO'}} = \underset{\mathbf{f}}{\text{argmin}}\ & \frac{1}{|\mathcal{T}|}\sum_{k\in\mathcal{T}} r\left(\left(\mathbf{f}^T\mathbf{x}[k]\right)^2 - \gamma K\right) + C\|\mathbf{f}\|_2^2 \\ \text{subject to}\ & \mathbf{f}^T\boldsymbol{\pi}_i = \sqrt{K}\end{aligned}}\ , \tag{18}$$

where $r(p) = \max\{0, p\}$ is the rectified linear unit (ReLU). The ReLU can be interpreted as a binary weighting of p, i.e., $r(p) = 0 \cdot p$ if $p \leq 0$, otherwise $r(p) = 1 \cdot p$. With $p \leftarrow (\mathbf{f}^T\mathbf{x})^2 - \gamma K$, a zero weight is applied to all training data points $\mathbf{x}$ for which $(\mathbf{f}^T\mathbf{x})^2 \leq \gamma K$. The ReLU $r(p)$ is a convex function and its convexity is preserved under composition with another convex function, such as $(\mathbf{f}^T\mathbf{x})^2 - \gamma K$. The nonnegative weighted sum over convex functions is also convex [22], thereby proving that the cost function in (18) is convex, given that $C \geq 0$. The set of points satisfying the power constraint in (14) do not form a convex set and can as such never lead to a convex optimization problem. Therefore, a linear constraint is used instead in (18), such that the feasible set of solutions is convex. The optimization problem in (18) is thus a convex optimization problem, because it has both a convex objective function and convex feasible set. Note that this convex reformulation is not a true power optimization any more, because the cost function sums over the binary weighted difference $(\mathbf{f}^T\mathbf{x})^2 - \gamma K$, rather than $(\mathbf{f}^T\mathbf{x})^2$ as in (14).

Because of the usage of a linear constraint in (18), rather than a quadratic constraint, the sign of the filter output response to the target pattern $\boldsymbol{\pi}_i$ is known a-priori. We can exploit this knowledge by applying an interference threshold to the filter output directly instead of the filter output power, i.e.,

$$\begin{aligned}\min_{\mathbf{f}}\ & \frac{1}{|\mathcal{T}|}\sum_{k\in\mathcal{T}} r\left(\mathbf{f}^T\mathbf{x}[k] - \sqrt{\gamma K}\right) + C\|\mathbf{f}\|_2^2 \\ \text{subject to}\ & \mathbf{f}^T\boldsymbol{\pi}_i = \sqrt{K}\end{aligned}\ , \tag{19}$$

thereby not 'wasting' degrees of freedom during the filter design on peak interferers or pattern sources that create a negative response at the filter output. In an attempt to focus more on the attenuation of larger interfering peaks compared to smaller interfering peaks, we reintroduce the squaring operator outside of the ReLU, leading to an alternative convex SPIR-optimal filter $\mathbf{f}_i^{\text{CSO"}}$ problem formulation:

$$\boxed{\begin{aligned}\mathbf{f}_i^{\text{CSO"}} = \underset{\mathbf{f}}{\text{argmin}}\ & \frac{1}{|\mathcal{T}|}\sum_{k\in\mathcal{T}} \left(r\left(\mathbf{f}^T\mathbf{x}[k] - \sqrt{\gamma K}\right)\right)^2 + C\|\mathbf{f}\|_2^2 \\ \text{subject to}\ & \mathbf{f}^T\boldsymbol{\pi}_i = \sqrt{K}\end{aligned}}\ , \tag{20}$$

Again, it can be easily shown that the alternative cost function in (20) is convex. The squared ReLU $(r(p))^2$ is a convex function. It is known that convexity is preserved under composition with an affine transformation [22], i.e., $(r(\mathbf{f}^T\mathbf{x} - \sqrt{\gamma K}))^2$ is convex if $(r(p))^2$ is convex. Again, the nonnegative weighted sum over convex functions is also convex, thereby proving that the cost function in (20) is convex, given that $C \geq 0$. Note that by introducing auxiliary variables, (20) can be expressed as a quadratic program which can be solved reliably by standard interior point methods (this is derived in Section V-B, i.e., in equation (23)). The convex reformulation in (20) is also not a true filter output power optimization, because the interference threshold $\sqrt{\gamma K}$ is subtracted from the filter output prior to squaring. It is also not a true power optimization because the filters under design do not attempt to attenuate negative peak interferers at all (as opposed to (18)). This means that more degrees of freedom can be exploited to suppress the peak interferers that really count, i.e., the ones generating positive peaks at the filter output. A logical consequence of not considering negative peak interferers during the filter design is that the filter-and-threshold-based pattern recognition using (20) has to be performed on the filter output amplitude directly:

$$\hat{\mathcal{A}}_i = \left\{k \mid \mathbf{f}_i^T\mathbf{x}[k] > T_i'\right\}, \tag{21}$$

rather than on the filter output power as originally described in (5). Note that for applications where the target pattern is often superimposed onto strong negative peak interferers, the use of (20) may result in a large amount of missed detections (false negatives), i.e., resulting in a reduction in recall.

B. SVM Interpretation

Besides guarantees on convergence to the global optimum, the convex SPIR reformulation in (20) provides an interesting SVM interpretation for SPIR-optimal filtering, leading to a maximum-margin interpretation of the SPIR criterion. Reconsider the convex reformulation (20) of the SPIR-optimal filter design problem, where the objective function is multiplied with

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on November 25,2020 at 12:58:36 UTC from IEEE Xplore. Restrictions apply.

$|\mathcal{T}|$:

$$\underset{\mathbf{f}}{\operatorname{argmin}} \sum_k \left(r\left(\mathbf{f}^T\mathbf{x}_k - \gamma\sqrt{K}\right)\right)^2 + C'\|\mathbf{f}\|_2^2$$

$$\text{subject to} \quad \mathbf{f}^T\boldsymbol{\pi}_i = \sqrt{K} \qquad , \qquad (22)$$

with $C' = C|\mathcal{T}|$ and where the sample time k is moved to the subscript to be consistent with notation in the field of SVMs. By introducing the slack variables e_k an equivalent optimization problem can be formulated:

$$\min_{\mathbf{f},\mathbf{e}} \sum_k e_k^2 + C'\|\mathbf{f}\|_2^2$$

$$\begin{aligned} \text{subject to} \quad & \mathbf{f}^T\boldsymbol{\pi}_i = \sqrt{K} \\ & \mathbf{f}^T\mathbf{x}_k - \gamma\sqrt{K} \leq e_k \quad \forall k \\ & e_k \geq 0 \quad \forall k \qquad , \end{aligned} \qquad (23)$$

where we now optimize over both $\mathbf{f} \in \mathbb{R}^N$ and $\mathbf{e} \in \mathbb{R}^{|\mathcal{T}|}$, i.e., the collection of all e_k $\forall k \in \mathcal{T}$. Next, we define the so-called threshold t and margin ρ in terms of $\sqrt{K}$ and γ:

$$\begin{aligned} \sqrt{K} &= t + \rho \\ \gamma\sqrt{K} &= t - \rho. \end{aligned} \qquad (24)$$

Reparametrizing the optimization problem in terms of t and ρ then results in:

$$\min_{\mathbf{f},\mathbf{e}} \quad \sum_k e_k^2 + C'\|\mathbf{f}\|_2^2 \qquad (25a)$$

$$\text{subject to} \quad \mathbf{f}^T\boldsymbol{\pi}_i - t = \rho \qquad (25b)$$

$$y_k\left(\mathbf{f}^T\mathbf{x}_k - t\right) \geq \rho - e_k \quad \forall k \qquad (25c)$$

$$e_k \geq 0 \quad \forall k, \qquad (25d)$$

with $y_k = -1$ $\forall k$, which in a context of SVMs introduces an implicit "ground truth" label which is in our formulation the same for each sample.

This problem is equivalent to L2-SVM [23] except for the fact that all labels are negative and that an extra linear constraint (25b) is added. The extra equality constraint leads to the loss of one degree of freedom when compared to regular L2-SVM, i.e., here we do not optimize over the threshold t (both t and ρ are fixed by fixing K and γ). This difference is essential for the SPIR-optimal filter to work. To better appreciate this statement, remember that regular SVM is a supervised learning method. However, the filter design problem addressed in this paper can be viewed as an extreme case of positive-unlabeled (PU) learning [24], i.e., we have only one labelled training sample, namely the target pattern $\boldsymbol{\pi}_i$ which represents the 'positive' class ($y = 1$). All other training samples are unknown, because they contain a mixture of noisy target and non-target pattern occurrences, and pure noise samples. In our SPIR formulation, all of those samples apparently are assigned a negative ground truth label y_k, which we know is only partially true.

So one may wonder how this method can work, given the class imbalance and erroneous training labels. This is due to the additional equality constraint compared to regular SVM. The a-priori fixed target pattern $\boldsymbol{\pi}_i$ response is key in fighting class imbalance. Understanding the robustness against erroneous (or absence of) training labels, requires some geometrical elaboration on that same equality constraint. As shown graphically in Fig. 3 for the case where $\mathbf{f} \in \mathbb{R}^2$, the equality constraint (25b) forces the pattern of interest $\boldsymbol{\pi}_i$ to be on the border of the margin ρ. To satisfy the equality constraint the separating hyperplane that is orthogonal to $\mathbf{f}_i$ has to be tangent to a half hypersphere with radius $\frac{\rho}{\|\mathbf{f}_i\|}$ and center point $\boldsymbol{\pi}_i$. The optimization problem thus consists of rotating the hyperplane on the hypersphere, trying to minimize the number of support vectors, i.e., training samples violating the predetermined margin $\frac{\rho}{\|\mathbf{f}_i\|}$, thereby driving as many e_k's close or equal to zero as possible. The support vectors are represented in Fig. 3 as orange dots. Consider now the support vectors surrounding the target pattern in Fig. 3 to represent occurrences of the target pattern in the training data. If the noise is spherically symmetric, rotating the filter hyperplane along the hypersphere, will not lead to a decrease in cost related to the target pattern occurrence support vectors. Therefore, the optimization algorithm will have to focus on the other support vectors to effectively decrease the cost, hence explaining the robustness against erroneous training labels.

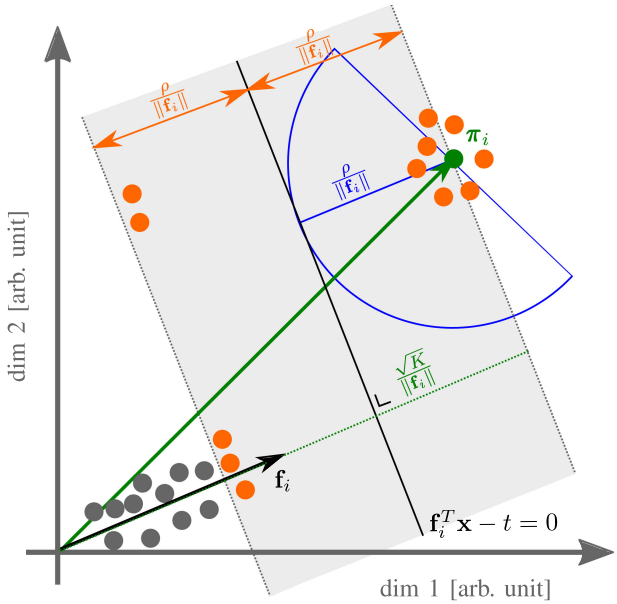


Fig. 3. A geometric interpretation of the convex SPIR-optimal optimization problem in two dimensions ($\mathbf{f} \in \mathbb{R}^2$). The shaded area represents the margin area. The green dot represents the pattern of interest $\boldsymbol{\pi}_i$. The other dots represent the training points $\mathbf{x}_k$, where the orange dots represent the support vectors. The equality constraint (25b) is satisfied if the separating line (in black) is tangent to the half ρ-circle (in blue).

Through this link between SPIR-optimal filtering and SVM, we can state that SPIR-optimal filtering can be viewed as the maximum margin counterpart of matched filtering. To the best of our knowledge, SPIR-filtering is the first true maximum margin approach to matched filtering. Similar approaches exist [25], but they have been formulated in a supervised setting, rather than the more practically relevant PU-learning setting that is more closely related to matched filtering. Another interesting result of the link between SVM and SPIR-optimal filtering, is the possibility to design a kernel variant of the SPIR-optimal filter design method similar to [26]. This kernel method has the

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on November 25,2020 at 12:58:36 UTC from IEEE Xplore. Restrictions apply.

potential of separating target pattern occurrences from the other samples in a non-linear way. Because the evaluation of kernel-SPIR-optimal filter requires for every data point the evaluation of the so-called kernel function over all its support vectors, it is not a low complexity filter and as such, it is outside of the scope of this work.

VI. Subspace Projection Regularization

In this section, we review the concept of subspace projection regularization [27] which allows to avoid a tedious tuning of the Tikhonov regularization parameter C in, e.g, (11), (14), (18) and (20). Generally speaking, the regularization of a learning problem serves to shrink the solution space compared to the unregularized solution space $\mathbb{R}^N$. The main idea here is to project the data into a lower-dimensional subspace that is orthogonal to certain undesired directions. Excluding these undesired directions altogether is a stronger way of regularizing than, e.g., Tikhonov regularization, where certain solutions are discouraged, but never explicitly excluded.

Assume the existence of a meaningful regularization subspace that is spanned by orthogonal basis vectors represented by the columns of the matrix $\mathbf{U} \in \mathbb{R}^{N \times P}$ where typically $P \ll N$. A subspace projection regularized optimal filter $\mathbf{f}$ is obtained through the following steps:

1) Construct the training data $\mathbf{x}'[k] = \mathbf{U}^T\mathbf{x}[k]$ by projecting the original training data onto the regularization subspace.
2) Design an optimal filter $\mathbf{f}' \in \mathbb{R}^P$ by training on $\mathbf{x}'$, while setting $C = 0$.
3) The optimal filter $\mathbf{f} = \mathbf{U}\mathbf{f}'$ is retrieved through the back-projection into the original data space.

Consider for example the matched filter design problem (11), which can be reformulated as a subspace projection regularized matched filter design problem through the application of the above steps, as follows:

$$\mathbf{f}_i^{\text{MF}} = \mathbf{U}\text{argmin}_{\mathbf{f}'} \frac{1}{|\mathcal{T}|} \sum_{k \in \mathcal{T}} (\mathbf{f}'^T \mathbf{U}^T \mathbf{x}[k])^2$$
$$\text{subject to} \quad (\mathbf{f}'^T \mathbf{U}^T \boldsymbol{\pi}_i)^2 = K \quad . \tag{26}$$

Besides the advantage of having explicit control over the solution subspace, another advantage of the subspace projection regularization, is that the optimization problem is formulated in a P-dimensional space, which leads to a reduction in computational effort given that $P \ll N$.

In [27], the following subspace design is proposed for the regularization of matched filters in the context of spike sorting. The first dimension in the subspace is explicitly set to the target pattern:

$$\mathbf{u}_{\boldsymbol{\pi}_i} = \boldsymbol{\pi}_i / \left\|\boldsymbol{\pi}_i\right\|_2 . \tag{27}$$

Having the target pattern explicitly in the subspace is important to assure that the resulting filter can respond to target pattern occurrences. $\mathbf{U}$ is then expanded with $P-1$ principal components of the recording data after removing the direction $\mathbf{u}_{\boldsymbol{\pi}_i}$ from the data to guarantee that $\mathbf{U}$ describes an orthogonal basis. The dimensionality of the subspace is controlled by capturing a predetermined fraction of the total signal power in the recording. Note that the use of this subspace performs an implicit denoising of the data prior to the filter design. We refer to [27] for more details.

Another advantage of this specific subspace projection regularization is that the structure of the subspace transforms the convex SPIR-optimal filter design problem into an unconstrained problem. If the filter is designed in this specific subspace, the constraint is reduced to an equation in a single unknown, and can be easily satisfied as follows:

$$\mathbf{f}'^T \mathbf{U}^T \boldsymbol{\pi}_i = f_1' \|\boldsymbol{\pi}_i\|_2 = \sqrt{K} \iff f_1' = \frac{\sqrt{K}}{\|\boldsymbol{\pi}_i\|_2}, \tag{28}$$

where f_1' represents the first element of the filter vector $\mathbf{f}'$ and is completely determined by the equality constraint. As such we have one degree of freedom less in the optimization problem. Consider $\mathbf{f}' = [f_1' \quad \mathbf{g}^T]^T$ with $\mathbf{g} \in \mathbb{R}^{P-1}$. The convex SPIR-optimal filter design problem, e.g., (20), can then be written as

$$\mathbf{f}_i^{\text{CSO"}} = f_1' \mathbf{u}_{\boldsymbol{\pi}_i} + \mathbf{U}_{-\boldsymbol{\pi}_i} \mathbf{g}_i^{\text{CSO"}}, \tag{29}$$

with

$$\mathbf{g}_i^{\text{CSO"}} = \underset{\mathbf{g}}{\text{argmin}} \sum_k \left(r \left(\mathbf{g}^T \mathbf{U}_{-\boldsymbol{\pi}_i}^T \mathbf{x}_k + f_1' \mathbf{u}_{\boldsymbol{\pi}_i}^T \mathbf{x}_k - \sqrt{\gamma K} \right) \right)^2, \tag{30}$$

Where $\mathbf{U}_{-\boldsymbol{\pi}_i} \in \mathbb{R}^{N \times (P-1)}$ is the regularization subspace minus the target pattern direction. Note that the filter design problem in (30) is a convex unconstrained smooth optimization problem.

VII. Case Study: Max-SPIR Filters for Neural Spike Sorting

In this section we will compare the novel SPIR-optimal filter design methods proposed in Section IV and V and compare them to the matched filter in a filter-and-threshold-based multi-pattern recognition task, namely neural spike sorting [14]. Spike sorting is the process of detecting action potentials or 'spikes' generated by neurons in an extracellular multi-channel brain recording and assigning each spike to its putative source neuron. Closely related to the neural spike sorting problem is the surface electromyography (sEMG) decomposition problem [28]–[30]. In the field of sEMG decomposition, the goal is to assign recorded motor-unit action potentials to their putative motor units. In a typical spike sorting pipeline [31], the algorithm can be divided in two stages: first a clustering algorithm is applied to all detected spike waveforms, where the cluster centroids serve as templates. In the second stage those templates are used to design matched filters to detect spikes from single neurons while resolving spike overlap.

Spike overlap occurs when two or more neurons with spatially overlapping spike waveforms fire near-simultaneously in time. Overlapping spike waveforms can not be resolved in the clustering stage, because overlapping spike waveforms typically do not form coherent clusters, nor do feature space projections of overlapping spike waveforms show a clear relationship with the single-unit spike waveforms they are composed of [32]. However, spike overlap is known to behave as a linear superposition at the level of the extracellular recording [33]. Therefore, given a set of single-unit spike templates from the clustering stage, a matched filtering approach can be taken for resolving spike overlap. However, as mentioned earlier, the SNR-optimal matched filter as used in many state-of-the-art spike sorting algorithms

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on November 25,2020 at 12:58:36 UTC from IEEE Xplore. Restrictions apply.

and software packages is insufficiently discriminative [16] for resolving spike overlap in a filter-and-threshold-based fashion. Therefore, heuristic iterative schemes are in use, leading to an increased complexity and new failure mechanisms for such heuristic matched filtering stages. The availability of proper discriminative template matching filters would eliminate the need for a heuristic iteration when resolving spike overlap, because discriminative filter-and-threshold based spike sorting has spike resolving capabilities by definition. Such discriminative filters can thus decrease the complexity of existing spike sorting algorithms. It is expected that the proposed SPIR-optimal filters are better at discriminating between spikes from their target neuron and spikes from other neurons than the SNR-optimal matched filters that are currently in use in spike sorting pipelines. Furthermore, filter-and-threshold-based sorting was proposed in [19] as a candidate for on-probe spike sorting, because of its low computational complexity.

In this analysis we will only focus on the filter-and-threshold-based spike sorting stage, i.e., we assume that we have already identified a spike template of interest $\boldsymbol{\pi}_i$ for which we will design a filter-and-threshold-based pattern recognition filter. We use an extracellular ground truth data set, i.e., a neural recording for which the spike times of a fraction of the neurons are known. Because of this knowledge, the spike templates of the ground truth neurons can be trivially retrieved, by averaging over all the spikes of a specific neuron (in practice such ground truth is not available, in which case $\boldsymbol{\pi}_i$ should be computed as a cluster centroid during the initial spike clustering phase). The ground truth recording is generated by the SHYBRID open-source tool [34] in combination with SpikeInterface [35] providing access to a Neuropixel probe [36] dataset with 374 recording channels that is publicly available [37]. Note that the public dataset contains an extracellular recording that has been already preprocessed. The preprocessing consists of a common average referencing and high-pass filtering with a cut-off frequency of 300 Hz. These preprocessing steps are a standard practice in spike sorting.

The subset of channels that are relevant to neuron i are collected in the S_i-channel signal $\mathbf{r}_i[k] \in \mathbb{R}^{S_i}$ with S_i the number of relevant electrodes for sorting the neuron i. Note that the spike pattern is spatially sparse, such that we can ignore most of the 374 recording channels. Here we define the spatial region of interest by considering all electrodes within a 100 μm radius of the electrode with the largest absolute peak in the template $\boldsymbol{\pi}_i$. Because the spike template is a spatio-temporal pattern, rather than a spatial pattern, we define a spatio-temporal delay line $\mathbf{x}[k] = [\mathbf{r}[k]^T\ \mathbf{r}[k-1]^T\ \ldots\ \mathbf{r}[k-L+1]^T]^T$ such that $N = S_i L$ with $L = 30$ corresponding to a 1 ms temporal window, given that the sampling frequency $f_s = 30$ kHz. For every neuron we consider one minute of recordings.

The ground truth recording that is used in this work contains the full ground truth spike information of 50 neurons. For each of these 50 neurons, we design four filters (the matched filter $\mathbf{f}_i^{\text{MF}}$ (11), the non-convex SPIR-optimal filter $\mathbf{f}_i^{\text{SO}}$ (14) and the convex SPIR-optimal filters $\mathbf{f}_i^{\text{CSO'}}$ (18) and $\mathbf{f}_i^{\text{CSO''}}$ (20)), and compare their filter-and-threshold-based spike sorting performance.

We use the regularization subspace design from [27], with a subspace that contains the template $\boldsymbol{\pi}_i$ and that captures 90% of the total signal power (further on we will analyse the influence of this factor). The initial value for γ is chosen such that the initial desired SPIR of the filters corresponds to 10 dB. In order to ensure a sufficient amount of effective training samples, γ is dynamically changed during the filter design procedure. If less than 5000 interference threshold crossings are attained at the output of a filter, the filter design is iteratively repeated with a decreased γ. This practice is meant to ensure a well-conditioned training for cases in which there is little or no interference. Note that by decreasing γ, the SPIR-optimal filter converges towards the matched filter in case of little or no interference. The number of threshold crossings of 5000 corresponds to an average minimum of 25 effective training samples per degree of freedom in the filter design.

We quantify the filter-and-threshold-based spike sorting performance in terms of both precision and recall, which are defined in the following way:

$$\text{precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}, \tag{31}$$

and

$$\text{recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}. \tag{32}$$

For every neuron i and for each filter individually, we select the threshold T_i that leads to the maximal F_1-score [38], where F_1 is defined as follows:

$$\text{F}_1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}. \tag{33}$$

In other words we are comparing the best possible performances that can be obtained for every filter in terms of F_1. Prior to comparing the different filter design methods, we split the neurons in two different groups based on the precision of the matched filter solution. Neurons with a precision > 0.9 are assumed to not be affected by peak interferers (29 neurons), whereas those with a precision ≤ 0.9 are expected to be more affected (21 neurons). We thus expect SPIR filters to show a larger benefit in the second ('interfering') group, while achieving similar performance as a matched filter in the first ('non-interfering' group).

The threshold-based spike sorting performances for the non-interfering and interfering group are summarized in Table I and II, respectively. We do not consider the heuristic SPIR-based filter design results in this comparison, because it performs poorly on the complex spike sorting data considered here with an overall average recall of 53% and a precision of only 40% (without ad-hoc iteration). The SPIR-based filtering fails due to the fundamental limitations discussed in Section III-C. We performed Wilcoxon signed-rank tests with $\alpha = 0.05$ to test for statistically significant performance differences between the different methods. A Bonferroni correction is applied to correct both for the six pairwise comparisons (between the different filter types) and two metrics (recall and precision), resulting in a significance level of $\text{p} < \frac{\alpha}{6 \cdot 2} = 0.00417$.

For the non-interfering neurons, we do not expect a benefit from using SPIR-optimal filters as the non-interfering neurons are selected as neurons on which the matched filter performs well. A statistical analysis shows that the SPIR-optimal filters can also provide satisfying results for such neurons. Indeed, comparing the matched filter $\mathbf{f}_i^{\text{MF}}$ recall (0.981 ± 0.043 (average $\pm$ standard deviation)) with non-convex SPIR-optimal

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on November 25,2020 at 12:58:36 UTC from IEEE Xplore. Restrictions apply.

TABLE I

FILTER-AND-THRESHOLD-BASED SPIKE SORTING PERFORMANCE IN TERMS OF PRECISION AND RECALL FOR THE GROUND-TRUTH NEURONS IN THE NON-INTERFERING GROUP. EVERY ROW CORRESPONDS TO A NEURON, WHEREAS THE GROUPED PAIRS OF COLUMNS CORRESPOND TO THE DIFFERENT FILTER DESIGN METHODS PRESENTED IN THIS PAPER

$\mathbf{f}_i^{MF}$ as in (11)		$\mathbf{f}_i^{SO}$ as in (14)		$\mathbf{f}_i^{CSO'}$ as in (18)		$\mathbf{f}_i^{CSO''}$ as in (20)	
precision	recall	precision	recall	precision	recall	precision	recall
0.909	1.000	1.000	1.000	1.000	1.000	1.000	1.000
0.910	0.862	0.928	0.882	0.876	0.932	0.899	0.821
0.912	0.974	0.951	0.991	0.966	0.983	0.935	0.983
0.937	0.982	0.949	0.990	0.968	0.982	0.975	0.886
0.950	0.987	0.975	0.983	0.966	0.990	0.885	0.907
0.954	0.994	0.954	0.995	0.958	0.994	0.911	0.944
0.955	0.967	0.948	0.985	0.953	0.982	0.920	0.928
0.958	1.000	0.958	1.000	0.958	1.000	1.000	1.000
0.967	0.974	0.982	0.984	0.989	0.981	0.929	0.916
0.976	0.804	1.000	0.941	1.000	0.980	1.000	1.000
0.977	1.000	0.989	1.000	0.989	1.000	0.977	1.000
0.979	0.954	0.979	0.981	0.992	0.989	0.981	0.971
0.982	1.000	0.982	1.000	0.982	1.000	0.972	0.995
0.988	0.994	0.994	1.000	0.994	1.000	0.994	1.000
0.995	0.991	0.987	1.000	0.987	1.000	1.000	0.986
0.998	0.996	0.993	0.996	0.991	0.996	0.969	0.971
0.998	1.000	1.000	1.000	1.000	1.000	0.988	0.992
0.999	0.987	1.000	0.987	1.000	0.986	0.959	0.912
0.999	0.989	1.000	0.991	0.999	0.992	0.992	0.985
0.999	0.996	1.000	0.996	1.000	0.995	0.925	0.851
1.000	0.989	1.000	0.989	1.000	0.989	1.000	0.989
1.000	0.997	1.000	0.994	1.000	0.988	1.000	0.982
1.000	0.998	1.000	0.998	1.000	0.995	0.990	0.954
1.000	1.000	1.000	1.000	0.996	1.000	0.968	0.996
1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.993
1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
0.977 ± 0.03	**0.981 ± 0.043**	**0.985 ± 0.021**	**0.989 ± 0.024**	**0.985 ± 0.026**	**0.992 ± 0.013**	**0.971 ± 0.035**	**0.964 ± 0.049**

TABLE II

FILTER-AND-THRESHOLD-BASED SPIKE SORTING PERFORMANCE IN TERMS OF PRECISION AND RECALL FOR THE GROUND-TRUTH NEURONS IN THE INTERFERING GROUP. EVERY ROW CORRESPONDS TO A NEURON, WHEREAS THE GROUPED PAIRS OF COLUMNS CORRESPOND TO THE DIFFERENT FILTER DESIGN METHODS PRESENTED IN THIS PAPER. THE PERFORMANCE METRIC VALUES SHOWN IN RED CORRESPOND TO VALUES LOWER THAN 65%, WHEREAS THE VALUES SHOWN IN GREEN CORRESPOND TO VALUES HIGHER THAN 80%

$\mathbf{f}_i^{MF}$ as in (11)		$\mathbf{f}_i^{SO}$ as in (14)		$\mathbf{f}_i^{CSO'}$ as in (18)		$\mathbf{f}_i^{CSO''}$ as in (20)	
precision	recall	precision	recall	precision	recall	precision	recall
0.022	1.000	**0.023**	1.000	**0.050**	1.000	1.000	1.000
0.180	0.821	**0.321**	**0.607**	0.852	0.821	1.000	1.000
0.258	0.948	**0.376**	0.914	0.949	0.966	1.000	1.000
0.399	0.821	**0.455**	0.911	**0.557**	0.842	0.832	0.781
0.406	**0.457**	**0.451**	**0.455**	**0.430**	**0.553**	0.699	**0.501**
0.488	**0.494**	**0.532**	**0.527**	**0.615**	**0.527**	0.770	**0.531**
0.522	1.000	**0.522**	1.000	**0.522**	1.000	**0.522**	1.000
0.529	**0.558**	0.671	0.777	0.767	0.770	0.751	0.867
0.556	**0.536**	0.889	0.857	0.958	0.821	0.893	0.893
0.564	0.663	0.762	**0.560**	0.926	**0.602**	0.812	0.651
0.593	0.981	0.731	0.823	0.801	0.919	0.912	0.848
0.595	0.995	**0.596**	0.992	0.812	0.785	0.856	0.847
0.667	0.923	0.732	0.888	0.827	0.853	0.832	0.785
0.677	**0.608**	0.696	0.663	0.735	0.660	0.802	0.733
0.684	0.720	0.797	0.733	0.814	0.933	0.934	0.947
0.711	0.873	0.988	1.000	1.000	1.000	0.975	1.000
0.778	1.000	0.778	1.000	0.778	1.000	1.000	1.000
0.788	0.766	0.893	0.771	0.823	0.843	0.854	0.848
0.819	0.984	0.829	0.969	0.828	0.966	0.763	0.831
0.833	0.992	0.833	0.992	0.833	0.992	0.977	0.992
0.848	0.921	0.935	0.946	0.966	0.943	0.873	0.728
0.567 ± 0.220	**0.812 ± 0.190**	**0.658 ± 0.238**	**0.828 ± 0.176**	**0.754 ± 0.220**	**0.847 ± 0.152**	**0.860 ± 0.121**	**0.847 ± 0.153**

filter $\mathbf{f}_i^{SO}$ recall (0.989 ± 0.024) results in $p = 0.00763$, with convex SPIR-optimal filter $\mathbf{f}_i^{CSO'}$ recall (0.992 ± 0.013) results in $p = 0.0219$, and with convex SPIR-optimal filter $\mathbf{f}_i^{CSO''}$ recall (0.964 ± 0.049) results in $p = 0.0129$, such that we can state that no statistically significant differences are found in terms of recall for the above comparisons. When comparing the recall between the different proposed SPIR-optimal filter design methods, i.e., $\mathbf{f}_i^{SO}$ and $\mathbf{f}_i^{CSO'}$ with $p = 0.875$, $\mathbf{f}_i^{SO}$ and $\mathbf{f}_i^{CSO''}$ with $p = 0.000681$, and, $\mathbf{f}_i^{CSO'}$ and $\mathbf{f}_i^{CSO''}$ with $p = 0.000538$, a significant difference in recall is identified between the usage of $\mathbf{f}_i^{SO}$ and $\mathbf{f}_i^{CSO''}$ on the one hand, and $\mathbf{f}_i^{CSO'}$ and $\mathbf{f}_i^{CSO''}$ on the other hand, twice in disadvantage of $\mathbf{f}_i^{CSO''}$. In terms of precision no significant

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on November 25,2020 at 12:58:36 UTC from IEEE Xplore. Restrictions apply.

differences are found between the pairwise combinations of filter design methods. For reasons of clarity, we will not elaborate on the quantitative details of the precision analysis for the non-interfering group. The interested reader can find the performance metrics and their averages in Table I. For the non-interfering examples we can thus conclude that there is no significant difference between the methods, except for a significant difference between the recall when comparing between the SPIR-optimal filter design methods. We argue that because there is no harmful interference in these examples, $\mathbf{f}_i^{\text{CSO''}}$ can not benefit from neglecting negative sources during the filter design, such that it can only suffer from potential destructive interference leading to the observed significant decrease in recall. Note that the practical implication of this significance is limited. The average difference in recall when comparing those methods is only around 2.5%. Based on the results presented in this paragraph, we can conclude that the use of SPIR-optimal filters is a viable alternative to the classical matched filter for the group of non-interfering neurons.

We now turn our attention to the interfering neurons, where by design we expect to see an improvement between the precision of the matched filter and of the SPIR-optimal filters. Comparing the matched filter $\mathbf{f}_i^{\text{MF}}$ precision (0.567 ± 0.220), with non-convex SPIR-optimal filter $\mathbf{f}_i^{\text{SO}}$ precision (0.658 ± 0.238) results in $p = 0.000196$, with convex SPIR-optimal filter $\mathbf{f}_i^{\text{CSO'}}$ precision (0.754 ± 0.220) results in $p = 0.000196$, and with convex SPIR-optimal filter $\mathbf{f}_i^{\text{CSO''}}$ precision (0.860 ± 0.121) results in $p = 0.000120$, such that we see that the anticipated effect is statistically backed up. When comparing between the SPIR-optimal filter design methods, $\mathbf{f}_i^{\text{SO}}$ and $\mathbf{f}_i^{\text{CSO'}}$ with $p = 0.00185$, $\mathbf{f}_i^{\text{SO}}$ and $\mathbf{f}_i^{\text{CSO''}}$ with $p = 0.000892$, and, $\mathbf{f}_i^{\text{CSO'}}$ and $\mathbf{f}_i^{\text{CSO''}}$ with $p = 0.0187$, the convex SPIR filters are shown to outperform the non-convex SPIR solution. When performing the pairwise comparisons for the recall, no significant effects are noted (again we omit the quantitative details, and refer to Table II). Hence, the SPIR-optimal filters can obtain a significantly higher precision, while preserving the recall. The average improvement in precision between the different filter design methods in the order that they were presented in this work, is around 10%, as can be seen from Table II. When comparing the use of matched filtering in threshold-based spike sorting with the use of convex SPIR-optimal filtering $\mathbf{f}_i^{\text{CSO''}}$, an improvement of almost 30% (in absolute sense) in average precision is noted. Besides the statistically significant results, the convex SPIR-optimal filters $\mathbf{f}_i^{\text{CSO''}}$ "transform" the threshold-based spike trains for 8 interfering neurons into single-unit spike trains, according to our arbitrary bound of precision >0.9. This is an increase of 16% in detected single-unit spike trains for the entire data.

In Figs. 4 and 5 we turn our attention to two example neurons from Table II. Fig. 4 corresponds to the red highlighted neuron in Table II. Given a precision of 0.406, 0.451, 0.430, and, 0.699 for $\mathbf{f}^{\text{MF}}$, $\mathbf{f}^{\text{SO}}$, $\mathbf{f}^{\text{CSO'}}$, and, $\mathbf{f}^{\text{CSO''}}$, respectively, the precision does not improve much for the SPIR-optimal filter when compared to $\mathbf{f}^{\text{MF}}$, except for a minor improvement in precision for $\mathbf{f}^{\text{CSO''}}$. All of these precisions are obtained at a similar recall. The top part of Fig. 4 shows the different filter outputs for three different extracellular recording snippets that are shown at the bottom of this figure. The leftmost recording snippet corresponds to a spike waveform occurrence of the target neuron. This target neuron

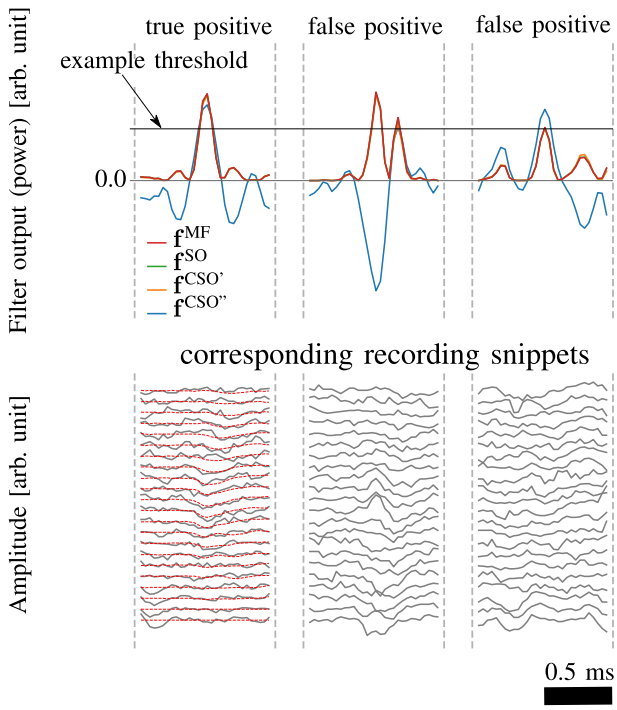


Fig. 4. **Top**: The different filter outputs for three example recording snippets associated with the red highlighted target neuron from Table II are shown. For this specific target neuron the output from $\mathbf{f}^{\text{MF}}$, $\mathbf{f}^{\text{SO}}$ and $\mathbf{f}^{\text{CSO'}}$ largely coincide. Note that the output from $\mathbf{f}^{\text{CSO''}}$ is not squared, because the filter-and-threshold-based pattern recognition has to be performed on the filter output amplitude directly (see (21)). **Bottom**: The filter input multi-channel recording snippets are shown that correspond to the filter outputs from the top half of this figure. A recording snippet that contains a true spike of the target neuron is overlaid with the spatio-temporal spike template of the target neuron (dashed red).

spike occurrence snippet contains a spike template overlay. A closer look at the different filter outputs reveals that the outputs of $\mathbf{f}^{\text{MF}}$, $\mathbf{f}^{\text{SO}}$, and, $\mathbf{f}^{\text{CSO'}}$ largely coincide for this neuron. To better understand this behaviour, consider the target neuron spike snippet and its template overlay. From this snippet and its template overlay, one can see that the peak-signal-to-noise ratio (PSNR) of the target neuron spike template is rather poor. More precisely the PSNR equals 2 dB when measured across all spike instances of this neuron. Given the linear nature of the filters, target neurons with low PSNR are expected to suffer from many interfering segments in their filter output. Based on this observation, the noise covariance structure of $\mathbf{f}^{\text{MF}}$ is likely to be very similar to the implicit interference covariance structure that lead to $\mathbf{f}^{\text{SO}}$ and $\mathbf{f}^{\text{CSO'}}$ due to the many interfering segments that appear during filter design, hence, leading to very similar filters. The second example filter output in Fig. 4 illustrates why $\mathbf{f}^{\text{CSO''}}$ has a better precision, i.e., the filter is capable of ignoring negative interferers. However, the rightmost example shows that also $\mathbf{f}^{\text{CSO''}}$ is ultimately bounded by the PSNR, which is shown by the peak that results from random noise fluctuations.

On the other hand, Fig. 5 which corresponds to the green highlighted neuron in Table II, shows a continuously improving precision of 0.180, 0.321, 0.852, and, 1.000 for $\mathbf{f}^{\text{MF}}$, $\mathbf{f}^{\text{SO}}$, $\mathbf{f}^{\text{CSO'}}$, and, $\mathbf{f}^{\text{CSO''}}$, respectively. This figure contains two examples that correspond to spike occurrences of the target neuron, which are indicated by the spike template overlay. It is apparent that the

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on November 25,2020 at 12:58:36 UTC from IEEE Xplore. Restrictions apply.

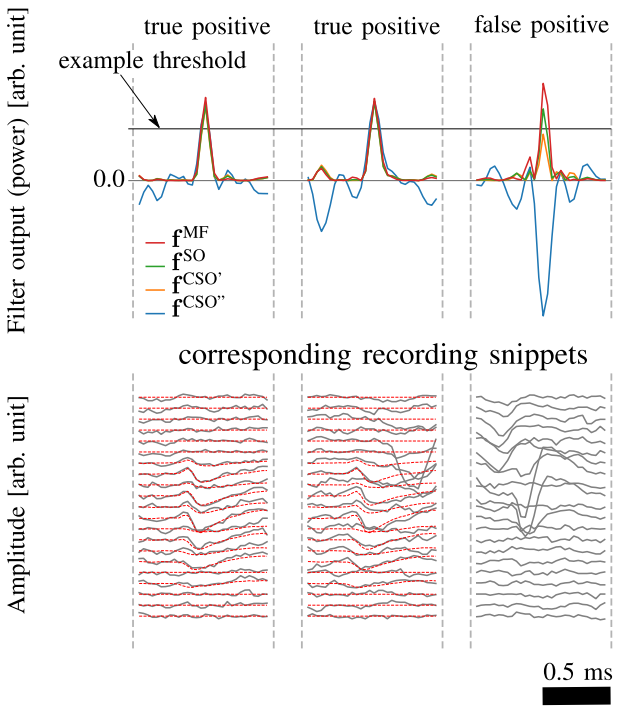


Fig. 5. **Top**: The different filter outputs for three example recording snippets associated with the green highlighted target neuron from Table II are shown. **Bottom**: The filter input multi-channel recording snippets are shown that correspond to the filter outputs from the top half of this figure. A recording snippet that contains a true spike of the target neuron is overlaid with the spatio-temporal spike template of the target neuron (dashed red).

PSNR of the target neuron spike template is more favourable when compared to the previous figure. The PSNR equals 12 dB when measured across all spike instances of this neuron. Therefore, during filter design, the implicit interference covariance structure is likely to be more distinct from the noise covariance, leading to more discriminative filters as is indicated by the increasing filter precision. The rightmost example in Fig. 5 illustrates the continuously improving interference suppression capabilities of the different filters in response to a spike from a non-target neuron.

Next we will study the effect of the choice of regularization subspace power fraction on the spike sorting performance. The power fraction is the fraction of the total signal power that is accounted for in the regularization subspace, and as such, it serves as the regularization hyper parameter in this work. In the previous analysis we chose a power fraction of 0.9 (i.e., 90%) because the resulting subspace is believed to provide a good trade-off between preserving and emphasizing the important signal characteristics, while providing a major reduction in dimensionality and computational complexity of the filter design. We can indeed verify from Fig. 6 that the average spike sorting performance (both in terms of recall and precision) in the non-interfering group for a power fraction of 0.9 is near perfect, and approximately coincides with the spike sorting performance when using 0.95 or 0.97 as a regularization power fraction. When considering the interfering group one can see that for the recall a similar conclusion can be drawn, i.e., the difference in average recall between 0.9, 0.95 and 0.97 is only marginal. However, for the precision we see that by making use of a power fraction of 0.95 compared to 0.9 a couple of percent points could have been won. Also in terms of precision, we see that for a power fraction of 0.97 the regularizing effect starts breaking down again. This is to be expected because a higher power fraction can be interpreted as a lesser regularization, due to the fact that the solution subspace is less constrained. From Fig. 6 it also becomes clear that the usage of a power fraction of 0.7 or 0.8 leads to reduced spike sorting performance in all cases. Purely based on spike sorting performance, 0.95 would be the better choice for use as a regularization power fraction. However, by also taking into account the relative dimensionality, i.e., the fraction between the dimensionality of the regularization subspace and the original problem subspace, we argue that the use of 0.9 is a reasonable practical choice. From Fig. 7 one can see that a reduction of almost 70% is obtained in terms of the dimensionality of the filter design subspace. By using a power fraction of 0.95 a reduction of only 45% is obtained. This difference in relative dimensionality will have a big practical impact in terms of memory usage and computation time.

When comparing between the different proposed filter design methods, the application of the different filters has the exact same computational complexity because they all share the same multiply-accumulate structure. However, differences in computational complexity are expected when comparing the different filter design methods during *training*. Fig. 8 reports on the proposed filter design training computation times for the target neurons from this case study (both interfering and non-interfering). The data points that are shown in this figure are normalized with respect to the computation times of the non-convex SPIR-optimal filter design of the corresponding target neurons. For every target neuron, a fixed random initialization point is used for the design of the three different proposed filters. Furthermore, a general-purpose optimization solver is used that is based on sequential quadratic programming [39] using default parameter settings [40] for the design of the different filters. The advantage of using a single optimization framework is that function and gradient evaluation (i.e., the main computational burden during the optimization) happens in a uniform way between the different proposed design methods, such that the computation times can be compared in a meaningful way. Note that the use of specialized frameworks, e.g., targeted at convex optimization, as well as non-default parameter settings, might further improve the convergence speed. However, these practical considerations are outside the scope of this work.

From Fig. 8, it can be seen that the median relative computation time for $\mathbf{f}^{CSO'}$ is equal to 0.98, with outliers ranging from 0.26 to 3.33. From these data we conclude that the computational work required for the design of $\mathbf{f}^{SO}$ and $\mathbf{f}^{CSO'}$ is similar. However, $\mathbf{f}^{CSO'}$ is guaranteed to converge to a global optimum, whereas this is not the case for $\mathbf{f}^{SO}$. This difference in convergence behaviour was also seen in the spike sorting performance analysis that was conducted earlier. The median relative computation time for $\mathbf{f}^{CSO''}$ is equal to 0.62, with outliers ranging from 0.14 to 1.60. Given that the 95th percentile is equal to 0.89, we conclude that $\mathbf{f}^{CSO''}$ requires less computational work when compared to $\mathbf{f}^{SO}$. Again, $\mathbf{f}^{CSO''}$ is guaranteed to converge to a global optimum, and is now also shown to be superior in terms of the filter design computation time. A possible explanation

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on November 25,2020 at 12:58:36 UTC from IEEE Xplore. Restrictions apply.

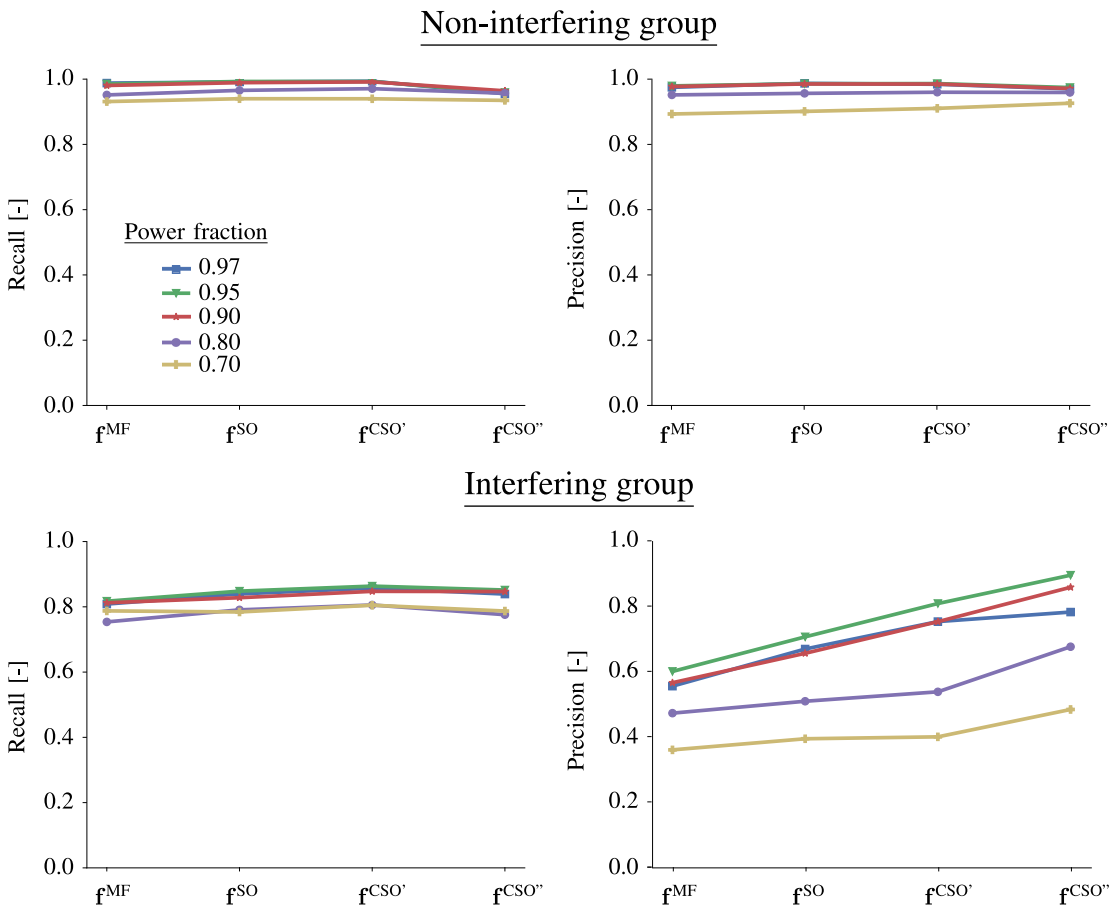


Fig. 6. The effect on the spike sorting performance of the choice of regularization subspace power fraction is studied. The average precision and recall for all analysed filter design methods are shown for both the non-interfering (top) and interfering (bottom) group.

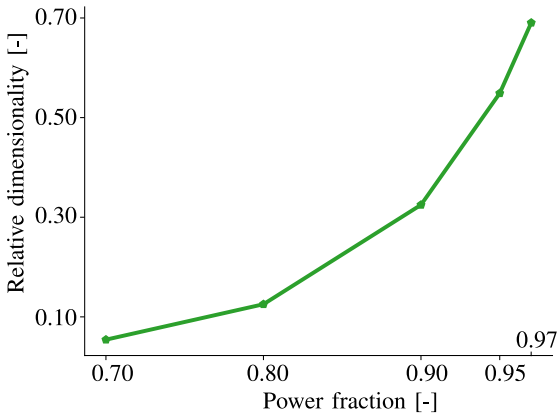


Fig. 7. The relative dimensionality of the filter design subspace is shown in function of the regularization subspace power fraction.

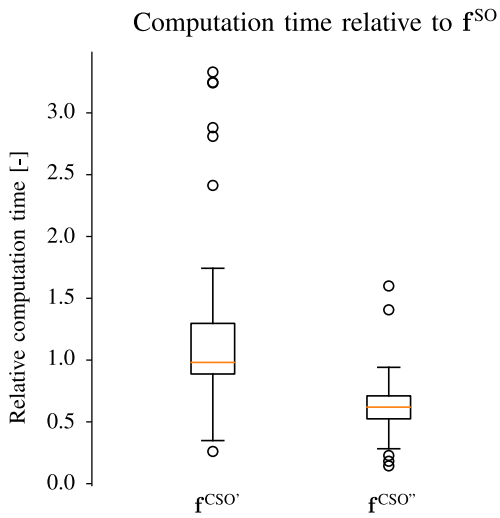


Fig. 8. Boxplots are shown for the filter design computation times of the convex filters $\mathbf{f}^{\mathrm{CSO'}}$ and $\mathbf{f}^{\mathrm{CSO''}}$. The computation times have been normalized with respect to the computation times of the non-convex filters $\mathbf{f}^{\mathrm{SO}}$ of the corresponding target neurons.

for the superior convergence behaviour can be found in the fact that negative interferers exist in the design of $\mathbf{f}^{\mathrm{CSO''}}$. As such, less abrupt changes in peak interference structure are expected during the optimization, which could potentially explain the observed improved convergence behaviour.

VIII. Conclusion

We have presented three novel filter design methods for use in filter-and-threshold-based multi-pattern recognition. The first SPIR-optimal filter design method was proposed to cope with the limitations of heuristic SPIR-optimal filter design methods. Two convex approximations of the first method were then proposed, which come with explicit guarantees on global optimality. The

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on November 25,2020 at 12:58:36 UTC from IEEE Xplore. Restrictions apply.

second convex reformulation was shown to be closely related to the theory of SVMs, and can be viewed as an extreme case of PU learning. Through this relationship with SVMs we concluded that this convex SPIR-optimal filter can be viewed as a maximum margin reformulation of the matched filter. Besides this conclusion, the relationship with SVMs provided a geometrical insight into the mechanics of the proposed filter design methods. Also, the proposed methods require no prior knowledge (except for the template of the target pattern), which make them particularly interesting for practical use, in which complete knowledge about the signal statistics is rarely available.

We have benchmarked the proposed filter design methods against the use of classical matched filters in the context of filter-and-threshold-based spike sorting. We were able to show that the proposed methods all outperform the matched filtering approach. The difference in precision between non-convex SPIR and convex SPIR-optimal filtering might be attributed to the discovery of a local rather than global optimal solution in the non-convex case. Another important difference is that convex SPIR-optimal filters $\mathbf{f}_i^{\text{CSO"}}$ can also make use of the sign in the filter output, i.e., interferers that lead to a negative output response are harmless, whereas for a power optimization, these interferers might become dominant again. Whether or not the sign of the filter output is exploited could explain the difference in average precision between $\mathbf{f}_i^{\text{CSO'}}$ and $\mathbf{f}_i^{\text{CSO"}}$. The results from these experiments indicate that the convex SPIR-optimal filters are promising candidates for replacing the matched filter in filter-and-threshold-based pattern recognition tasks. When comparing non-convex SPIR optimization to the optimization of $\mathbf{f}_i^{\text{CSO"}}$, the optimization of of $\mathbf{f}_i^{\text{CSO"}}$ showed considerably faster convergence. During this case study we have also investigated the use of a subspace projection regularization approach, which has the ability to transform the convex SPIR-optimal filter design problem into an unconstrained optimization problem. Furthermore, because of the use of this regularization approach, both proposed filter design methods had a minimal set of interpretable hyper-parameters. Interpretable hyper-parameters are interesting in general, because they can be chosen in an informed manner, rather than a brute-force grid search type of experiment.

Despite the fact that the proposed filters were shown to outperform the classical matched filters, both SPIR-optimal and classical matched filters fail under certain conditions. First, they both require the availability of the target pattern prior to the filter design. In the case of spike sorting, it is not always possible to obtain a pattern or template estimate. This can happen because a certain neuron is inactive in the training data or because the neuron is consistently active together with another neuron, such that only overlapping spike waveforms for the intended target neuron are available from clustering. Furthermore, neither the SPIR-optimal filter, nor the classical matched filter can suppress noise segments that contain scaled version of the target pattern. This is due to the linear nature of the filters under study. Furthermore, when the underlying signal statistics are non-stationary, the filter solutions will be rendered useless as the statistics evolve throughout time. A common source of non-stationarity in spike sorting is probe drift, which means that the recording probe moves relative to the neurons it records from. Future research should focus on the development of adaptive SPIR-optimal filters that accommodate for the changing signal statistics. Finally, as indicated in the case study, the filter-and-threshold pattern recognition performance is ultimately bound by the (P)SNR of the target patterns embedded in the signal as illustrated in the example in Fig. 4.

Although the proposed methods were shown to significantly improve the filter-and-threshold-based pattern recognition performance for spike sorting, the results showed that there is still room for improvement for many of the so-called interfering neurons. From an SVM point of view, we might conclude that for many of the interfering neurons, the target spike pattern and non-target spike patterns are not linearly separable. As already mentioned, the convex SPIR-optimal filter design method can be straightforwardly converted into a non-linear kernel method. Such a kernel SPIR-optimal filter has the potential to separate non-linearly separable data at the cost of an increase in computational complexity.

IX. Acknowledgment

The authors would like to thank Prof. Johan Suykens for the constructive and insightful discussions related to the material that is presented in this manuscript. They would also like to thank Simon Geirnaert for his helpful suggestions towards the use of kernel methods, even though the results were not included in the paper due to space constraints.

References

[1] J. Wouters, F. Kloosterman, and A. Bertrand, "Signal-to-peak-interference ratio maximization with automatic interference weighting for threshold-based spike sorting of high-density neural probe data," in *Proc. 9th Int. IEEE/EMBS Conf. Neural Eng. (NER)*, 2019, pp. 247–250.

[2] G. Turin, "An introduction to matched filters," *IRE Trans. Inf. Theory*, vol. 6, no. 3, pp. 311–329, 1960.

[3] R. D. Hippenstiel, *Detection Theory: Applications and Digital Signal Processing*. Boca Raton, FL, USA: CRC Press, 2001.

[4] F. Salahdine, H. El Ghazi, N. Kaabouch, and W. F. Fihri, "Matched filter detection with dynamic threshold for cognitive radio networks," in *Proc. Int. Conf. Wireless Netw. Mobile Commun. (WINCOM)*, 2015, pp. 1–6.

[5] U. Salama, P. L. Sarker, and A. Chakrabarty, "Enhanced energy detection using matched filter for spectrum sensing in cognitive radio networks," in *Proc. Joint 7th Int. Conf. Informat., Electron. Vis. 2nd Int. Conf. Imag., Vis. Pattern Recognit.*, 2018, pp. 185–190.

[6] S. Nadathur and R. Crittenden, "A detection of the integrated sachs–wolfe imprint of cosmic superstructures using a matched-filter approach," *Astrophysical J. Lett.*, vol. 830, no. 1, 2016, Art. no. L19.

[7] M. A. B. Othman, J. Belz, and B. Farhang-Boroujeny, "Performance analysis of matched filter bank for detection of linear frequency modulated chirp signals," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 53, no. 1, pp. 41–54, Feb. 2017.

[8] T. Chanwimalueang, W. von Rosenberg, and D. P. Mandic, "Enabling r-peak detection in wearable ECG: Combining matched filtering and hilbert transform," in *Proc. IEEE Int. Conf. Digit. Signal Process.*, 2015, pp. 134–138.

[9] X. Wu, H. Chen, T. Gan, J. Chen, C.-W. Ngo, and Q. Peng, "Automatic hookworm detection in wireless capsule endoscopy images," *IEEE Trans. Med. Imag*, vol. 35, no. 7, pp. 1741–1752, Jul. 2016.

[10] Y. Yang, S. Boling, and A. J. Mason, "A hardware-efficient scalable spike sorting neural signal processor module for implantable high-channel-count brain machine interfaces," *IEEE Trans. Biomed. Circuits Syst.*, vol. 11, no. 4, pp. 743–754, Aug. 2017.

[11] D. Valencia and A. Alimohammad, "An efficient hardware architecture for template matching-based spike sorting," *IEEE Trans. Biomed. Circuits Syst.*, vol. 13, no. 3, pp. 481–492, Jun. 2019.

[12] M. S. Lewicki, "A review of methods for spike sorting: The detection and classification of neural action potentials," *Netw.: Comput. Neural Syst.*, vol. 9, no. 4, pp. R53–R78, 1998.

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on November 25,2020 at 12:58:36 UTC from IEEE Xplore. Restrictions apply.

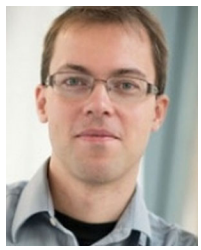
[13] R. Q. Quiroga, Z. Nadasdy, and Y. Ben-Shaul, "Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering," *Neural Comput.*, vol. 16, no. 8, pp. 1661–1687, 2004.
[14] S. Gibson, J. W. Judy, and D. Marković, "Spike sorting: The first step in decoding the brain: The first step in decoding the brain," *IEEE Signal Process. Mag.*, vol. 29, no. 1, pp. 124–143, Jan. 2011.
[15] C. Rossant *et al.*, "Spike sorting for large, dense electrode arrays," *Nat. Neurosci.*, vol. 19, no. 4, pp. 634–641, 2016.
[16] F. Franke, M. Natora, C. Boucsein, M. H. Munk, and K. Obermayer, "An online spike detection and spike classification algorithm capable of instantaneous resolution of overlapping spikes," *J. Comput. Neurosci.*, vol. 29, nos. 1/2, pp. 127–148, 2010.
[17] W. M. Roberts and D. K. Hartline, "Separation of multi-unit nerve impulse trains by a multi-channel linear filter algorithm," *Brain Res.*, vol. 94, pp. 141–149, Aug. 1975.
[18] R. Vollgraf and K. Obermayer, "Improved optimal linear filters for the discrimination of multichannel waveform templates for spike-sorting applications," *IEEE Signal Process. Lett.*, vol. 13, no. 3, pp. 121–124, Mar. 2006.
[19] J. Wouters, F. Kloosterman, and A. Bertrand, "Towards online spike sorting for high-density neural probes using discriminative template matching with suppression of interfering spikes," *J. Neural Eng.*, vol. 15, no. 5, 2018, Art. no. 056005.
[20] B. Picinbono, "Generalized matched filters," in *Proc. 5th ASSP Workshop Spectr. Estimation Model.*, 1990, pp. 182–186.
[21] N. M. Nasrabadi, "Regularized spectral matched filter for target recognition in hyperspectral imagery," *IEEE Signal Process. Lett.*, vol. 15, pp. 317–320, Mar. 2008.
[22] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[23] Y. Koshiba and S. Abe, "Comparison of l1 and l2 support vector machines," in *Proc. Int. Joint Conf. Neural Netw.*, 2003, vol. 3, pp. 2054–2059.
[24] M. Claesen, F. De Smet, J. A. Suykens, and B. De Moor, "A robust ensemble approach to learn from positive and unlabeled data using SVM base models," *Neurocomputing*, vol. 160, pp. 73–84, 2015.
[25] A. Rodriguez, V. N. Boddeti, B. V. Kumar, and A. Mahalanobis, "Maximum margin correlation filter: A new approach for localization and classification," *IEEE Trans. Image Process.*, vol. 22, no. 2, pp. 631–643, Feb. 2012.
[26] X. Huang, L. Shi, and J. A. Suykens, "Asymmetric least squares support vector machine classifiers," *Comput. Statist. Data Anal.*, vol. 70, pp. 395–405, 2014.
[27] J. Wouters, F. Kloosterman, and A. Bertrand, "A data-driven regularization approach for template matching in spike sorting with high-density neural probes," in *Proc. 41st Annu. Int. Conf. IEEE Eng. Med. and Biol. Soc.*, 2019, pp. 4376–4379.
[28] R. Merletti and D. Farina, *Surface Electromyography: Physiology, Engineering, and Applications*. Hoboken, NJ, USA: Wiley, 2016.
[29] A. Holobar and D. Zazula, "Multichannel blind source separation using convolution kernel compensation," *IEEE Trans. Signal Process.*, vol. 55, no. 9, pp. 4487–4496, Sep. 2007.
[30] I. Gligorijević, J. P. van Dijk, B. Mijović, S. Van Huffel, J. H. Blok, and M. De Vos, "A new and fast approach towards sEMG decomposition," *Med. Biol. Eng. Comput.*, vol. 51, no. 5, pp. 593–605, 2013.
[31] P. Yger *et al.*, "A spike sorting toolbox for up to thousands of electrodes validated with ground truth recordings in vitro and in vivo," *Elife*, vol. 7, 2018, Art. no. e34518.
[32] J. Wouters, F. Kloosterman, and A. Bertrand, "A neural network-based spike sorting feature map that resolves spike overlap in the feature space," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2020, pp. 1175–1179.
[33] C. Ekanadham, D. Tranchina, and E. P. Simoncelli, "A unified framework and method for automatic neural spike identification," *J. Neurosci. Methods*, vol. 222, pp. 47–55, 2014.
[34] J. Wouters, F. Kloosterman, and A. Bertrand, "SHYBRID: A graphical tool for generating hybrid ground-truth spiking data for evaluating spike sorting performance," *Neuroinformatics*, to be published, doi: 10.1007/s12021-020-09474-8.
[35] A. P. Buccino *et al.*, "Spikeinterface, a unified framework for spike sorting," *BioRxiv*, 2019, p. 796599.
[36] J. J. Jun *et al.*, "Fully integrated silicon probes for high-density recording of neural activity," *Nature*, vol. 551, no. 7679, pp. 232–236, 2017.
[37] N. Steinmetz, M. Carandini, and K. D. Harris, ""Single Phase3" and "Dual Phase3" Neuropixels Datasets," Mar. 2019. [Online]. Available: https://figshare.com/articles/_Single_Phase3_Neuropixels_Dataset/7666892
[38] A. Tharwat, "Classification assessment methods," *Appl. Comput. Informat.*, 2018.
[39] D. Kraft, "A software package for sequential quadratic programming," *Forschungsbericht- Deutsche Forschungs- und Versuchsanstalt fur Luft- und Raumfahrt*, 1988.
[40] P. Virtanen *et al.*, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nat. Methods*, vol. 17, pp. 261–272, 2020.



Jasper Wouters received the Ph.D. degree in electrical engineering from KU Leuven, in 2020. He is a Postdoctoral Researcher with the STADIUS Center for Dynamical Systems, Signal Processing and Data Analytics of the Electrical Engineering Department (ESAT), KU Leuven. His research concerns the design of signal processing and machine learning methods for neural interfaces, in particular for spike detection and spike sorting algorithms in next-generation high density neural probes.



Panagiotis Patrinos received the Ph.D. degree in control and optimization, the M.S. degree in applied mathematics and the M.Eng. from the National Technical University of Athens, in 2010, 2005, and 2003, respectively. He is associate professor at the Department of Electrical Engineering (ESAT) of KU Leuven, Belgium. In 2014 he was a visiting professor at Stanford University. After his Ph.D. he held postdoc positions at the University of Trento and IMT Lucca, Italy, where he became an assistant professor in 2012. His current research interests lie in the intersection of optimization control and learning. In particular he is interested in the theory and algorithms for structured nonconvex optimization as well as learning-based, model predictive control with a wide range of applications including autonomous vehicles, machine learning and signal processing.



Fabian Kloosterman received the Ph.D. degree in neurobiology from the University of Amsterdam, The Netherlands, in 2003. He is a Principal Investigator at the Neuro-Electronics Research Flanders (NERF) and an Associate Professor in the Faculty of Psychology and Educational Sciences of KU Leuven, Belgium. Until 2011, he worked as a Postdoctoral Fellow and Senior Researcher at the Massachusetts Institute of Technology (MIT) in Cambridge, MA, USA. His research explores the link between brain, cognition and behavior. In particular, he studies the neural mechanisms of spatial navigation, learning and memory using a combination of large-scale measurement and closed-loop manipulation of brain activity. He is further interested in the development and application of neurotechnology and computational tools to understand brain function.



Alexander Bertrand received the Ph.D. degree in electrical engineering from KU Leuven, in 2011. He is currently Associate Professor at the Electrical Engineering Department (ESAT) of KU Leuven, Belgium. He was a Visiting Researcher at the University of California, Los Angeles (in 2010) and at the University of California, Berkeley (in 2013). His research involves multi-channel signal processing algorithm design with a focus on biomedical sensor arrays, such as electroencephalography (EEG), neural implants, and distributed body- and neuro-sensor networks, with applications in, a.o., brain-computer interfaces and hearing prostheses. Dr. Bertrand is recipient of an ERC starting grant (2018), the 2012 FWO/IBM Award, the 2013 KU Leuven Research Council Award (Science & Technology), and was awarded several research fellowships and grants. He is currently Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING (2nd term), and a recurrent Associate Editor for the yearly international conference of the IEEE Engineering in Medicine and Biology Society (IEEE EMBC 2016–2020). He is currently a member of the Sensor Array & Multi-channel (SAM) technical committee and a board member of the Benelux chapter of the IEEE Signal Processing Society. He has served as Lead Guest Editor for *Signal Processing* (Elsevier), and as Area Chair and Technical Program Committee Member for several international signal processing conferences and workshops.

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on November 25,2020 at 12:58:36 UTC from IEEE Xplore. Restrictions apply.