Windowed State Space Filters for Peak Interference Suppression in Neural Spike Sorting

Christof Baeriswyl [1,2], Alexander Bertrand [1,3], and Reto A. Wildhaber [4]

[1]*KU Leuven, Dept. of Electrical Engineering ESAT, STADIUS Center for Dynamical Systems, Signal Processing and Data Analytics*, Leuven, Belgium
[2]*Bern University of Applied Sciences*, Biel, Switzerland
[3]*KU Leuven Institute for Artificial Intelligence (Leuven.AI)*, Leuven, Belgium
[4]*University of Applied Sciences and Arts Northwestern Switzerland & ETH Zurich*, Switzerland
{cbaerisw, abertran}@esat.kuleuven.be, reto.wildhaber@fhnw.ch

***Abstract*—Signal-to-peak-interference ratio (SPIR) optimal filters are template matching filters with peak interference suppression properties. Such max-SPIR filters are used in multi-pattern recognition problems, such as neural spike sorting in micro-electrode array probes, where cellular action potentials need to be detected and clustered according to their firing neuron cells. In high-density probes with hundreds of channels, such max-SPIR filter banks can require unacceptable high computational resources, in particular for applications with real-time demands and/or on-probe spike sorting. In this paper, we present a computationally attractive substitute for max-SPIR filters by recursively computed Autonomous Linear State Space Model (ALSSM) filters. In our approach, we approximate the impulse response of max-SPIR filters by low order ALSSMs and perform the signal convolution in the new, low-dimensional ALSSM vector space. We demonstrate our method on real neural recordings from high-density probes and show only minimal loss in detection quality while the computational complexity drops by up to a factor 10.**

***Index Terms*—multi-class pattern recognition, template matching, linear state space models, neural spike sorting**

I. Introduction

In this paper, we use Autonomous Linear State Space Models (ALSSMs) as multi-channel filters to solve a bottleneck in template matching tasks in computationally constrained environments, e.g., for real-time neural decoding in brain implants. Such applications require efficient filters due to high sampling rates, high density grids of measurement channels, and multiple patterns to be recognized. ALSSM filters are an attractive alternative since they are recursively computed linear filters working in a low-dimensional feature space.

We apply ALSSMs to the problem of template matching with peak interference suppression, which is used in, e.g., multi-class pattern recognition problems [1]. The goal is to design filters that extract a specific spatio-temporal signal template from a multi-channel signal while suppressing peak interferers that may falsely trigger a detection threshold at the filter output. This is achieved with a data-driven filter design based on maximizing the signal-to-peak-interference ratio (max-SPIR) criterion [1], [2]. In the case of multi-pattern detection, i.e., detecting and discriminating different, potentially overlapping patterns or templates, multiple such max-SPIR filters need to run in parallel, which might exceed computational resources.

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 802895), and from the Swiss National Science Foundation under Grant CR23I2_166030.

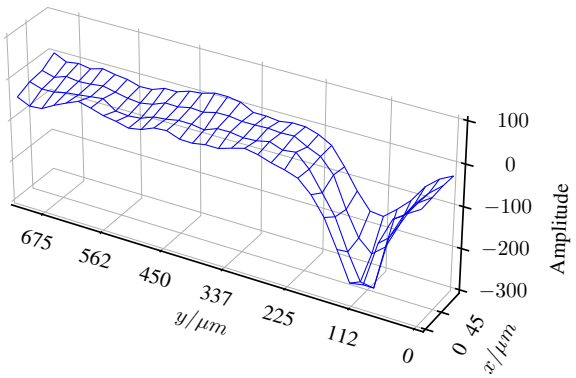

Fig. 1. Spatial plot of a single neuron spike from a micro-electrode array probe on a mouse cortex. The probe spans an area of 70 x 720 µm. Data: [3]

A particular application where such a computational bottleneck occurs is in online neural spike sorting [4]. In neural tissues, each neuron cell generates action potentials, which are observed as electric spikes on micro-electrode grids placed on the cortex. Multiple such spikes generated by a single neuron typically have a signature waveform due to neuron cell geometry and orientation [4]. The goal of neural spike sorting is to assign each detected spike to its generating neuron, which can be achieved by a max-SPIR filter bank [1], [2]. Neural signals, cf. Fig. 1, often originate from high-density, multi-channel probes with high data rates. Especially for recent neural probes with hundreds of channels [5], the computational effort of max-SPIR filters becomes substantial, as it scales linearly with the number of filter taps, channels, and patterns (i.e., observable neurons). Therefore, there is an urgent need to reduce the complexity of max-SPIR filters.

ALSSM filters share the efficiency of Infinite Impulse Response (IIR) filters, but have similar design flexibility as Finite Impulse Response (FIR) filters and are well suited to

process high-density data. In many applications, ALSSMs lead to recursive computation rules and enable efficient computation of least squares problems as proposed in [6]–[8]. In recent applications, ALSSMs were used to implement gesture recognition from magnetic fields, to estimate time delay in acoustical signals, or to perform shape discrimination in ECG signals [9]–[11]. In this paper, we use ALSSMs to reduce the computational burden in max-SPIR filter banks by showing a signal convolution in a low-dimensional ALSSM vector space.

This paper is organized as follows: we start with an introduction to ALSSM filters in Section II and an introduction to max-SPIR filters in Section III. Then, we explain how these max-SPIR filters can be modeled by more efficient ALSSM filters using local approximates in Section IV and conclude with experimental results on neural spiking data in Section V.

II. Introduction to ALSSM Filters

A. Autonomous Linear State Space Models (ALSSMs)

Autonomous Linear State Space Models (ALSSMs) are deterministic state space models without input signals. A discrete-time ALSSM of order $N \in \mathbb{N}$ is defined recursively as

$$x_{i+1} = Ax_i \ \in \mathbb{R}^N \tag{1}$$

$$s_i = cx_i \ \in \mathbb{R} \tag{2}$$

with state transition matrix $A \in \mathbb{R}^{N\times N}$, output vector $c \in \mathbb{R}^{1\times N}$, model output $s_i \in \mathbb{R}$, and state vector $x_i \in \mathbb{R}^N$ with initial state x_0 (at $i = 0$, often simply denoted as x). This model has the closed form

$$s_i(x) = cA^i x \ \in \mathbb{R} \ . \tag{3}$$

ALSSMs comprise a wide class of signal shapes including sinusoidals, exponentials, and polynomials, as well as sums and products of these [8]. In this paper, we only consider polynomial models. For example, as shown in [6], the third order polynomial in $i \in \mathbb{Z}$ with coefficients $\lambda_0, \ldots, \lambda_3 \in \mathbb{R}$ is

$$s_i(x) = \lambda_0 + \lambda_1 i + \lambda_2 i^2 + \lambda_3 i^3 \ \in \mathbb{R} \tag{4}$$

and corresponds to the output of (3) with parameters

$$\begin{aligned} A &= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\ c &= \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}, \\ x &= \begin{bmatrix} \lambda_0 & \lambda_1 & \lambda_2 & \lambda_3 \end{bmatrix}^\mathsf{T} . \end{aligned}$$

B. Localized Least Squares Approximation with ALSSMs

Now, we approximate the observed signal $y \in \mathbb{R}^K$ (where $K \in \mathbb{N}$ is the number of observed time samples) by the output sequence of an ALSSM, masked with a sliding window of fixed length and centered at time index k. The ALSSM has a fixed A and c, but an unknown x, which is to be optimized such that the least squared error between the windowed signal and the ALSSM output gets minimal. In other words, we transform the input signal y into K feature vectors $x \in \mathbb{R}^N$. In our example, we use the ALSSM model of a 3rd order polynomial (i.e., $N = 4$) and the elements of x correspond to the coefficients of the local polynomial approximation. At time k, the observed signal y is locally approximated by minimizing the squared error (over x)

$$J_a^b(k,x) = \sum_{i=k+a}^{k+b} \gamma^{i-k} \big(y_i - cA^{i-k}x\big)^2 \ \in \mathbb{R} \tag{5}$$

over the interval $\{k+a, k+b\}$, $a, b \in \mathbb{Z}, a < b$, with an exponential window decaying by factor $\gamma \in \mathbb{R}$. Choosing $\gamma = 1$ leads to a rectangular window, where all samples in the cost (5) are weighted equally. However, as we will see later, $\gamma \neq 1$ will guarantee numerical stability when performing recursive computations as explained below.

To simplify the computation of (14), the cost is reparametrized as

$$J_a^b(k,x) = \kappa_k - 2x^\mathsf{T}\xi_k + x^\mathsf{T} W_k x \tag{6}$$

with

$$\kappa_k = \textstyle\sum_{i=k+a}^{k+b} \gamma^{i-k}\, y_i^2 \ \in \mathbb{R} \tag{7}$$

$$\xi_k = \textstyle\sum_{i=k+a}^{k+b} \gamma^{i-k}\, y_i \left(A^{i-k}\right)^\mathsf{T} c^\mathsf{T} \ \in \mathbb{R}^N \tag{8}$$

$$W_k = \textstyle\sum_{i=k+a}^{k+b} \gamma^{i-k} \left(A^{i-k}\right)^\mathsf{T} c^\mathsf{T} c\, A^{i-k} \ \in \mathbb{R}^{N\times N} \tag{9}$$

as proposed in [7, Chapt. 4]. It follows that (6) is minimal for

$$\begin{aligned} \hat{x}_k &= \operatorname*{argmin}_x J_a^b(k,x) \\ &= W_k^{-1}\xi_k \ . \end{aligned} \tag{10}$$

Thanks to the properties of ALSSMs, we now gain recursive computation rules for W_k and ξ_k, and thus, also for (10), as shown in [6]. For the forward recursions $k \to k+1$ we get

$$\begin{aligned} \xi_{k+1} = \gamma^{-1}A^{-\mathsf{T}}\xi_k &- \gamma^{a-\delta-1}\big(A^{a-1}\big)^\mathsf{T} c^\mathsf{T} y_{k+a} \\ &+ \gamma^{b-\delta}\big(A^b\big)^\mathsf{T} c^\mathsf{T} y_{k+b+1} \end{aligned} \tag{11}$$

$$\begin{aligned} W_{k+1} = \gamma^{-1}A^{-\mathsf{T}}W_k A^{-1} &- \gamma^{a-\delta-1}\big(A^{a-1}\big)^\mathsf{T} c^\mathsf{T} c A^{a-1} \\ &+ \gamma^{b-\delta}\big(A^b\big)^\mathsf{T} c^\mathsf{T} c A^b \ , \end{aligned} \tag{12}$$

or alternatively, for the backward recursions $k \to k-1$

$$\begin{aligned} \xi_{k-1} = \gamma A^\mathsf{T}\xi_k &+ \gamma^{a-\delta}\big(A^a\big)^\mathsf{T} c^\mathsf{T} y_{k+a-1} \\ &- \gamma^{b-\delta+1}\big(A^{b+1}\big)^\mathsf{T} c^\mathsf{T} y_{k+b} \end{aligned} \tag{13}$$

$$\begin{aligned} W_{k-1} = \gamma A^\mathsf{T} W_k A &+ \gamma^{a-\delta}\big(A^a\big)^\mathsf{T} c^\mathsf{T} c A^a \\ &- \gamma^{b-\delta+1}\big(A^{b+1}\big)^\mathsf{T} c^\mathsf{T} c A^{b+1} \ . \end{aligned} \tag{14}$$

Note that W reaches a steady state for $0 \ll k \ll K$. Therefore, we set W as constant (and accept some inaccuracies towards the signal borders). In Section IV, we will exploit the recursions (11)-(14) to improve the computational efficiency of max-SPIR filters, which are introduced in the next section.

III. Introduction to max-SPIR Filters

Signal-to-peak-interference ratio (SPIR) optimal filters are discriminative template matching filters are used in pattern recognition tasks with overlapping patterns and/or strong peak interferers. A SPIR filter is a spatio-temporal FIR filter that is

designed to peak at signal components that resemble a known spatio-temporal template π, while suppressing noise and peak interferers.

There exist different approaches to learn coefficients of max-SPIR filters. Without loss of generality, here we use the full convex SPIR-optimal learning rules from [1] to derive the coefficients of a M-channel filter $h \in \mathbb{R}^{ML}$, $M, L \in \mathbb{N}$, (in which the L per-channels filter coefficients of all M channels are all stacked in a single vector), although the approach explained in the remaining of this paper applies to any other max-SPIR designs as well. The filter h is then found as

$$h = \underset{h}{\operatorname{argmin}} \frac{1}{Q} \sum_{k \in \tau} r\Big((h^\mathsf{T} y_k)^2 - \beta\Omega\Big) + C\|h\|_2^2$$
$$\text{subject to } h^\mathsf{T}\pi = \sqrt{\Omega}\ , \tag{15}$$

where the inner product $h^\mathsf{T} y_k$ defines the filter output at time k between the spatio-temporal filter h and a M-channel signal y (of which the L most recent samples of all M channels are stacked in y_k). Furthermore, $\pi \in \mathbb{R}^{ML}$ is the (known) target pattern (a classic matched filtering approach would set $h = \pi$) and $\tau \in \mathbb{N}^Q$ is the set of training samples with length $Q \in \mathbb{N}$. $C \in \mathbb{R}_{>0}$ is a hyper-parameter that controls the amount of regularization, $\Omega \in \mathbb{R}_{>0}$ is an arbitrary strictly positive number that controls the output response to the pattern, $\beta \in \mathbb{R}_{>0}$ is chosen such that the degrees of freedom in the filter design are fully spent on suppressing peak interferers rather than on minimizing the noise floor, and finally we use the function $r(x) = \max\{0, x\}$. With this Rectified Linear Unit (ReLU), a zero weight is applied to training data points that do not generate a strong output response, i.e., for which $(h^\mathsf{T} y_k)^2 \leq \beta\Omega$. We use a SLSQP optimization algorithm for the minimization of (15) and refer to [1] for more details.

Once the coefficients h are successfully learned (this learning process typically happens offline based on relevant training data), target patterns are detected putting a tunable threshold $T \in \mathbb{R}_{>0}$ on the filter output power $\hat{y}_k^2 > T$ with

$$\hat{y}_k = h^\mathsf{T} y_k \ \in \mathbb{R}\ . \tag{16}$$

IV. Max-SPIR Filters with ALSSMs for Neural Spike Sorting

In this section, we approximate h of max-SPIR filters with ALSSMs to simplify the convolution (16), which is the main computational burden during online operation of max-SPIR filters.

We apply max-SPIR filters to the specific problem of neural spike sorting, which is an instance of a multi-pattern recognition task. The goal of neural spike sorting is to link each detected spike to its putative neuron. In [1], it was shown that this can be achieved based on a bank of max-SPIR filters, where each filter is sensitive to the spikes of one particular neuron while suppressing spikes of other neurons (which can potentially overlap with a target spike).

A. Impulse Response Approximation with ALSSMs

We first approximate the impulse response h of length L of a single channel and generalize to multiple channels after.

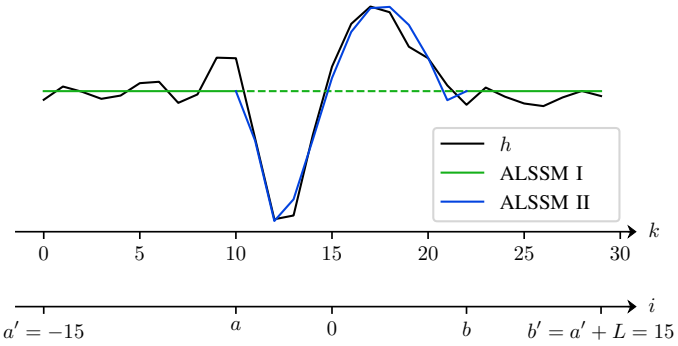

Fig. 2. Max-SPIR filter impulse response h (black line) approximated with a superposition of two ALSSMs; ALSSM I (green line) of order 0, i.e., a constant offset, spans from index a' to b', and ALSSM II (blue line) of a 3rd order polynomial spans from index a to b. ALSSM II is shifted such that its model energy becomes maximum.

Figure 2 represents one channel of the impulse response of a max-SPIR filter obtained from (15) applied to neural spiking data. To optimally approximate this impulse response, we use a superposition of two ALSSMs as arranged in Fig. 2: ALSSM I is a polynomial of order 0, modeling a constant offset spanning the full impulse response of length L from a' to b' (i.e., $L = b' - a'$). ALSSM II, a 3rd order polynomial, covers the middle section of h spanning from a to b; the position of this middle section is chosen such that the energy covered by ALSSM II is maximum, which coincides with the main waveform in the impulse response.

The superposition of ALSSMs leads to a joint ALSSM with joint parameters and a joint state vector. However, such composed models remain recursively computable and comparable in their computational complexity, cf. [6]. The impulse response h is then represented in the ALSSM domain via the feature vector $\hat{x}_h$, which is found by applying (10) to the impulse response h.

B. Convolution in the ALSSM Feature Space

Using ALSSMs, the convolution (16) becomes substantially cheaper when performed in the low-dimensional ALSSM feature space of the state vectors x. We start with single ALSSMs and extend to ALSSM superpositions afterwards.

Let $s_i(\hat{x}_h)$ be the ALSSM signal model (3) and $\hat{h} = [s_a(\hat{x}_h), s_{a+1}(\hat{x}_h), \ldots, s_b(\hat{x}_h)]^\mathsf{T}$ the approximated impulse response of h. Further, we have the reversed single channel observation signal $\overleftarrow{y}$ with $\overleftarrow{y}_k = y_{-k}$.

Then, (16) modifies to

$$\begin{aligned}\big(y * \hat{h}\big)_k &= \sum_{i=a}^{b} y_{k-i}\, s_i(\hat{x}_h) = \sum_{i=k+a}^{k+b} \overleftarrow{y}_i\, c A^{i-k} \hat{x}_h \\ &= \hat{x}_h^\mathsf{T} \underbrace{\sum_{i=k+a}^{k+b} \overleftarrow{y}_i (A^{i-k})^\mathsf{T} c^\mathsf{T}}_{\overleftarrow{\xi}_k} \\ &= \hat{x}_h^\mathsf{T} \overleftarrow{\xi}_k \end{aligned} \tag{17}$$

with $\overleftarrow{\xi}_k$ as in (8) ,$\overleftarrow{y}$ the input signal in (8), and $\gamma \approx 1$. $\overleftarrow{\xi}_k$ is computed applying the recursion (11) or (13). Note that the initial state vectors $\hat{x}$ in (10) representing y are never explicitly computed.

In summary, the convolution (16) is substituted by the low-dimensional convolution (17) with

- (17) is an approximation of (16),
- (16) is a vector inner product of size L, (17) is a vector inner product of size N with $N \ll L$,
- N is independent of the window length, and
- $\overleftarrow{\xi}_k$ is independent of h and only computed once for multiple max-SPIR filters (each extracting different patterns).

To generalize to our superimposed ALSSMs from Section IV-A, $\overleftarrow{\xi}_k$ in (17) gets replaced by the composed equivalent. All the rest, including the computational complexity of the inner product in (17), remains the same. Figure 3 shows in the upper plot an example of an observed signal with target spikes and interfering spikes with higher amplitude. The filter output power of a max-SPIR filter and of an ALSSM filter with approximated impulse response $\hat{h}$ and with convolution in the ALSSM feature space is shown in the lower plot. It is observed that in both cases, the interfering spikes are fully suppressed while the target spikes are enhanced.

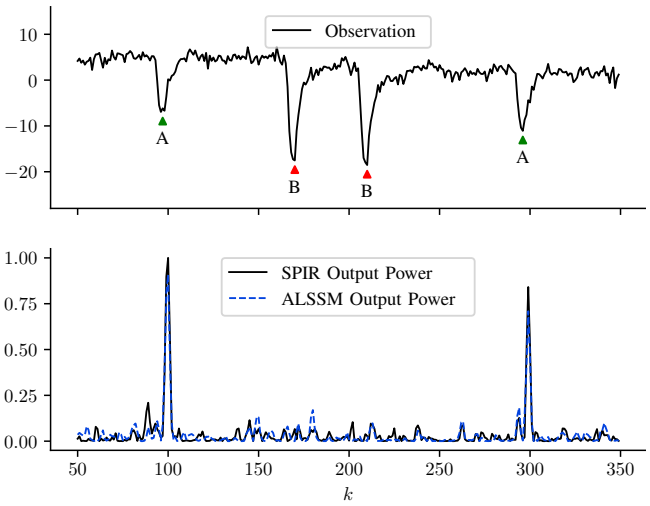

Fig. 3. The upper plot shows the observed signal sampled at 30 kHz with target spikes (A) at $k = 100$ and $k = 300$ and interfering spikes (B). The lower plot shows the filter output power of a max-SPIR filter (black line) and its ALSSM approximation filter (blue line). Both filters peak at the target spikes and show distinct suppression of the interfering spikes.

C. Extension to Multiple Patterns and Multiple Channels

For the extension to J spike templates (for J different neurons), every template j gets its own max-SPIR filter coefficients $h^{(j)}$ and a different output of the filter convolution (16).

For the extension to M-channel recordings, the approximation from Section IV-A is applied channel-wise, leading to M impulse responses $\hat{h}_1^{(j)}, \ldots, \hat{h}_M^{(j)}$ (which are stacked as in the vector h found from (15)) and the filter output is the sum of the per-channel convolutions computed from (17).

Figure 4 shows the ALSSM approximation of a max-SPIR filter impulse response for multiple channels for a given target spike template $\pi^{(j)}$.

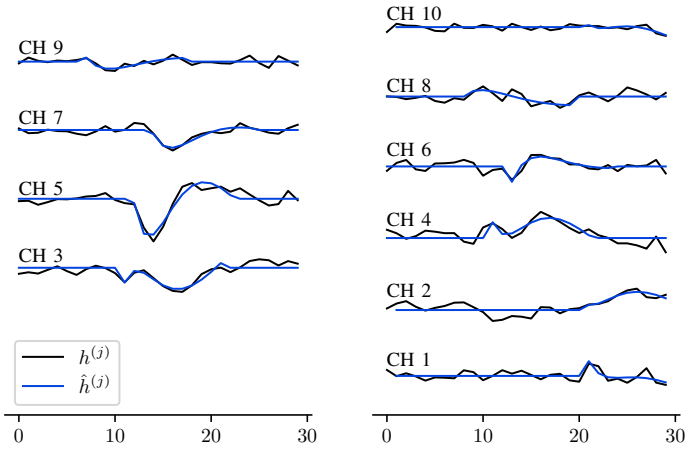

Fig. 4. The plots show the impulse responses $h^{(j)}$ for a max-SPIR filter with 11 channels (black lines) with the corresponding approximations $\hat{h}^{(j)}$ of the superimposed ALSSMs (blue lines).

V. Experiments

We validate the proposed algorithm on the public Neuropixel dataset with 374 extracellular channels placed on a mouse cortex spanning an area of 0.27 mm^2 [3]. The dataset was recorded with a sampling rate of 30 kHz and preprocessed with common average referencing and high-pass filtering with cut-off frequency at 300 Hz, which is a standard procedure in spike sorting.

To generate ground truth data from real recordings, the software package SHYBRID [12] was used, an open source tool to remove spike trains from a region of the probe and inject them at another region (details in [12]). In total, we injected 30 spike trains from 30 different neurons. From these 30 neurons, 15 contain spikes that overlap with at least one other spike train (due to being injected on the same probe location). The other 15 neurons have no overlap with other injected spike trains, but might overlap with spikes from the neurons at the probe region where the spike train was injected.

For the analysis, the template $\pi^{(j)}$ for each neuron of interest is assumed to be known and extracted in a prior spike clustering phase in the spike sorting pipeline. Furthermore, when analyzing the data, we reduced the channels to the subset of channels that are located closer than 100 µm to the channel where the spike reaches its highest absolute amplitude; in average, 21 channels were selected. All test signals have a length of two minutes, whereof one minute is used as the SPIR filter training set.

A. Spike Sorting Performance

Spikes are detected by setting a threshold on the filter output (16) (here approximated as (17)). The threshold is chosen to maximize the F1 score

$$F_1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{18}$$

TABLE I
AVERAGE SPIKE SORTING PERFORMANCE AND COMPUTATIONAL EFFORT PER TIME INDEX k OF THE CONVOLUTION OF SNR OPTIMAL, MAX-SPIR, AND SPIR ALSSM APPROXIMATION FILTERS.

Filter	Precision	Recall	F1 Score	Scalar Additions	Scalar Multiplications
SNR optimal matched filter	0.794 ± 0.185	0.902 ± 0.086	0.836 ± 0.131	157 250 (100%)	157 500 (100%)
Full convex max-SPIR filter	0.894 ± 0.103	0.929 ± 0.070	0.910 ± 0.084	157 250 (100%)	157 500 (100%)
SPIR ALSSM approximation filter	0.845 ± 0.130	0.909 ± 0.086	0.873 ± 0.105	31 224 (19.8%)	22 622 (14.3%)

with

$$\text{precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}} \tag{19}$$

and

$$\text{recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}} \,. \tag{20}$$

Table I shows the average spike sorting performance for an SNR optimal filter [13], the original max-SPIR filter, and the ALSSM approximated max-SPIR filter. The SNR optimal filter is added as a benchmark as this is the most commonly used template matching filter in state-of-the-art spike sorting pipelines [14]. The ALSSM method shows overall minimal loss in detection quality compared to the max-SPIR filter and it still outperforms the SNR optimal matched filter.

B. Computational Effort Comparison

SPIR-ALSSM filtering includes four steps: filter training, ALSSM approximation of the filter, ALSSM transformation of the observations, and convolution in the ALSSM feature space. The computational load of step 1 and 2 are irrelevant in this context since they are performed offline. Step 3, the ALSSM signal transformation, involves the computation of ξ_k (8). One recursion step of ξ_k comes down to 3 scalar multiplications using the sparse Jordan canonical form as proposed in [8, Chap. 3], and the signal approximation is independent of the filter $h^{(j)}$ and thus of the number of templates J. Step 4, the convolution, now reduces from L (template length) scalar multiplications, which is in our example 30 to 100, to N (ALSSM system order) scalar multiplications, which is 5 in our example (1 coefficient for the constant baseline and 4 coefficients for the 3rd order polynomial).

Table I gives a comparison of the overall computational load per time sample for the SNR optimal and the full convex max-SPIR filter compared to our ALSSM approximation. For our results, we used $L = 30$ filter taps, $M = 374$ channels, $J = 250$ target neurons, and 21 channels per neuron (100 µm radius). The ALSSM method has about factor 5 less additions and factor 7 less multiplications. Since the ALSSM is independent of the template length, the computational superiority increases further if longer templates are used.

VI. CONCLUSION

We proposed Autonomous Linear State Space Models (ALSSMs) to approximate computationally expensive signal-to-peak-interference ratio (SPIR) optimal filters used in threshold-based neural spike sorting.

The presented approach has been successfully applied on real neural data from high-density, multi-channel probes to extract spiking information in extracellular recordings with overlapping neurons. It was shown that our approximation outperforms the detection rate of SNR optimal matched filters and has a substantially lower computational cost than SNR optimal or max-SPIR filters.

This paper showed an example in the neuroscience field, however the method is directly adaptable to other fields.

REFERENCES

[1] J. Wouters, P. Patrinos, F. Kloosterman, and A. Bertrand, "Multi-pattern recognition through maximization of signal-to-peak-interference ratio with application to neural spike sorting," *IEEE Transactions on Signal Processing*, vol. 68, pp. 6240–6254, 2020.

[2] 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," *Journal of neural engineering*, vol. 15, no. 5, p. 056005, 2018.

[3] N. Steinmetz, M. Carandini, and K. D. Harris, ""Single Phase3" and "Dual Phase3" Neuropixels Datasets," 3 2019. [Online]. Available: https://figshare.com/articles/_Single_Phase3_Neuropixels_Dataset/7666892

[4] 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 Processing Magazine*, vol. 29, no. 1, pp. 124–143, 2012.

[5] C. M. Lopez, S. Mitra, J. Putzeys, B. Raducanu, M. Ballini, A. Andrei, S. Severi, M. Welkenhuysen, C. Van Hoof, S. Musa, and R. F. Yazicioglu, "22.7 a 966-electrode neural probe with 384 configurable channels in 0.13µm soi cmos," in *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, 2016, pp. 392–393.

[6] R. A. Wildhaber, N. Zalmai, M. Jacomet, and H.-A. Loeliger, "Windowed state-space filters for signal detection and separation," *IEEE Transactions on Signal Processing*, vol. 66, no. 14, pp. 3768–3783, 2018.

[7] R. A. Wildhaber, "Localized state space and polynomial filters with applications in electrocardiography," Ph.D. dissertation, ETH Zurich.

[8] N. Zalmaï, "A state space world for detecting and estimating events and learning sparse signal decompositions," Ph.D. dissertation, ETH Zurich.

[9] N. Zalmai, C. Kaeslin, L. Bruderer, S. Neff, and H.-A. Loeliger, "Gesture recognition from magnetic field measurements using a bank of linear state space models and local likelihood filtering," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 2569–2573.

[10] R. A. Wildhaber, E. Ren, F. Waldmann, and H.-A. Loeliger, "Signal analysis using local polynomial approximations," in *2020 28th European Signal Processing Conference (EUSIPCO)*. IEEE, 2021, pp. 2239–2243.

[11] R. A. Wildhaber, N. Zalmai, M. Jacomet, and H.-A. Loeliger, "Signal detection and discrimination for medical devices using windowed state space filters," in *2017 13th IASTED International Conference on Biomedical Engineering (BioMed)*. IEEE, 2017, pp. 125–133.

[12] J. Wouters, F. Kloosterman, and A. Bertrand, "Shybrid: A graphical tool for generating hybrid ground-truth spiking data for evaluating spike sorting performance," *Neuroinformatics*, vol. 19, no. 1, pp. 141–158, 2021.

[13] F. Franke, R. Quian Quiroga, A. Hierlemann, and K. Obermayer, "Bayes optimal template matching for spike sorting–combining fisher discriminant analysis with optimal filtering," *Journal of computational neuroscience*, vol. 38, no. 3, pp. 439–459, 2015.

[14] P. Yger, G. L. Spampinato, E. Esposito, B. Lefebvre, S. Deny, C. Gardella, M. Stimberg, F. Jetter, G. Zeck, S. Picaud *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, p. e34518, 2018.