Multi-Resolution Autonomous Linear State Space Filters for N-Dimensional Signals

Christof Baeriswyl, *Student, IEEE*, Frédéric Waldmann, Alexander Bertrand, *Senior Member, IEEE*, and Reto A. Wildhaber

***Abstract*—Linear filtering or convolution operations can be computationally expensive, especially in low-resource environments, high data rate applications, or scenarios involving large datasets. In particular, for multi-dimensional signals, the computational cost scales polynomially with the kernel size. In this paper, we propose the use of multi-dimensional Autonomous Linear State Space Models (ALSSMs) to reduce the computational complexity of convolution and correlation operations on N-dimensional signals and extend the method to (windowed) least-squares filters. For that, we work in an ALSSM subspace and provide efficient recursive computation rules that significantly improve the efficiency of those filters. In addition, we provide various window functions and demonstrate parallelization of computations and reuse of intermediate results for multi-resolution analyses. Our method is particularly interesting for systems with limited resources, such as battery-powered and wearable devices, or when (real-time) processing of very large data sets is required, such as in the field of image processing and machine learning. We conclude with several practical examples, with ready-to-use implementations of the proposed methods as an open source Python repository.**

***Index Terms*—linear filtering, inner products, convolution, cross-correlation, N-dimensional signals, subspaces, autonomous linear state space models, multi-dimensional models, localized least squares, recursive least squares, localized windows, multi-resolution filtering.**

I. Introduction

The inner products of two signals, where one of the signals is repetitively shifted in a sliding window manner, forms the basis of many linear and non-linear filters. An immediate application for this is the convolution to apply a linear time-invariant (LTI) filter. Another relevant application is windowed or localized least squares problems, where the signal interval of interest is continuously shifted along a signal in the form of a sliding window. However, such applications are often limited by the computational complexity of their calculations, in particular for real-time or low-resource applications. This problem becomes even more critical in the case of multi-dimensional signals due to the exponential increase in the number of parameters.

In this paper, we scale down such problems by linear projection and work in a subspace of the signal vector space. We show that defining subspaces by Autonomous Linear State Space Models (ALSSMs) leads to efficient recursive computation rules for inner products applied in a sliding window manner, which considerably reduces the overall computational complexity in particular for multi-dimensional signals.

The use of ALSSMs as single-channel signal models has already been studied extensively for least squares problems in the past in [1]–[8] and has been used in many practical applications [9]–[14]. So far, multi-channel signals have commonly been processed as a series of single-channels, and the results combined in a second step, e.g. by adding up the fitting costs [6], [7], by combining per-channel fitting cost ratios [13], or by summing up per-channel filter outputs [14]. A first basic example of directly using two-dimensional ALSSMs for image processing was recently presented [15], but a generalization to arbitrary signal models and arbitrary dimensions is currently lacking.

In other applications, such as multi-resolution analyses, as required for wavelet transforms, image compression, multi-scale decomposition and similar algorithms, ALSSM, to our knowledge, have not yet been explicitly used. Today, multi-resolution algorithms are mostly based on filter banks of various window lengths [16]. However, such algorithms often scale unfavorably with the signal and filter lengths and in higher dimensions. Nevertheless, some filter bank algorithms already increase algorithmic efficiency by sharing intermediate results between the single filters, as for example the fast Fourier transform (FFT) [17] or the fast wavelet transform [18] do. However, FFT is a lossless method and is only advantageous for longer windows due to the overhead involved in its calculations [19], [20], while our proposed method, which works in subspaces with lower dimensions, benefits from highly efficient (lossy) signal approximations and can be more flexibly adapted to the actual data or problem in order to achieve further computational benefits.

Christof Baeriswyl is with KU Leuven, Department of Electrical Engineering (ESAT), STADIUS Center for Dynamical Systems, Signal Processing and Data Analytics, Leuven, Belgium, with School of Life Sciences FHNW, Institute for Medical Engineering and Medical Informatics, Muttenz, Switzerland, and also with ETH Zurich, Signal and Information Processing Laboratory, Zürich, Switzerland (e-mail: christof.baeriswyl@esat.kuleuven.be).

Frédéric Waldmann is with School of Life Sciences FHNW (e-mail: frederic.waldmann@fhnw.ch).

Alexander Betrand is with KU Leuven, ESAT-STADIUS, and also with Leuven.AI - KU Leuven institute for AI (e-mail: alexander.bertrand@esat.kuleuven.be).

Reto Wildhaber is with School of Life Sciences FHNW, and also with ETH Zurich, Zürich, Switzerland (e-mail: reto.wildhaber@fhnw.ch).

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)

Manuscript received Nov 18, 2024; revised Jun 19, 2025; accepted Oct 20, 2025.

0000–0000/00$00.00 © 2021 IEEE

A. In Summary

In this paper, we propose a framework to recursively compute inner products and related operations for one- and multi-dimensional signals based on ALSSM subspaces. We derive the computation of inner products $\langle y, y' \rangle$ for discrete-time, one- and multi-dimensional signals y and y' in an ALSSM subspace and provide recursive computation rules to evolve from k to $k+1$ such as

$$\langle y_{\bullet+k}, y' \rangle \;\rightarrow \langle y_{\bullet+k+1}, y' \rangle \;. \tag{1}$$

where $y_{\bullet+k}$ denotes the shifted version of y by k samples.

From that, we extend to related methods, such as signal convolution

$$(y * h)_k \rightarrow (y * h)_{k+1} \tag{2}$$

with impulse response h, and the least squares problem

$$\underset{x}{\operatorname{argmin}} \|\mathcal{A}x - y_{\bullet+k}\|^2 \rightarrow \underset{x}{\operatorname{argmin}} \|\mathcal{A}x - y_{\bullet+k+1}\|^2 \tag{3}$$

with design matrix $\mathcal{A}$ and coefficient vector x applied in a sliding window manner. We provide new insights into the computational efficiency of these methods, and derive computation rules allowing sharing of computations across different dimensions and/or scales in multi-resolution settings. Furthermore, we provide a new insight into the robustness of ALSSMs in inner product computation, even for cases where one of the two signals involved is poorly represented in the ALSSM subspace.

In Section II of this paper, we start with one-dimensional signals and a brief introduction to ALSSMs and ALSSM subspaces, and continue with inner products and norms in such subspaces. In Section III, we build on these inner products and subspace norms to derive recursive rules for signal convolution and for least squares problems. In Sections IV and V we generalize these concepts to multi-dimensional signals and advanced window shapes which are of particular interest in localized least-squares problems, and complete our framework in Section VI with multi-resolution analysis methods as used in many filtering applications. Finally, the paper concludes with three practical examples using real-world data from various fields including medicine in Section VII and with additional examples in the supplementary material.

An implementation of the proposed method is published online as ready-to-use examples written in Python [21].

B. Notation

The notation y_k indicates the sample at index k of a signal y. For multi-dimensional signals, we use bold index tuples such as $\boldsymbol{k} = (k_1, \ldots, k_L)$ to index an L-dimensional signal.

To indicate shifted or reversed version of discrete-time signals defined on $\mathbb{Z}$ such as $y = [\ldots, y_{-1}, y_0, y_1, y_2, \ldots]$, we subsequently use the dot-notation

$$y_{\bullet+k} \triangleq [\ldots,\; y_{k-1},\; y_k,\; y_{k+1},\; y_{k+2},\; \ldots] \;. \tag{4}$$

For example, $y_{\bullet-1}$ then indicates the signal shifted to the right by one sample and $y_{-\bullet}$ the time-reversed signal. We will also apply this notation to multi-dimensional signals where $\boldsymbol{k}$ is an index tuple, as we will see later.

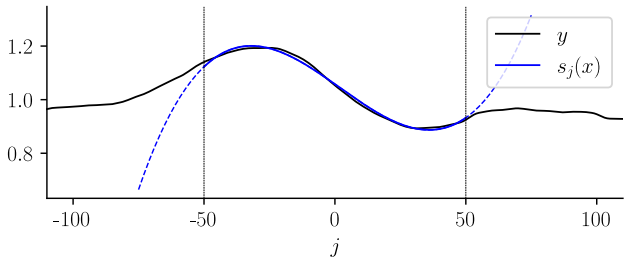


Fig. 1. Local approximation of a bioelectrical signal using ALSSMs. The arbitrary signal (black) is approximated by a 3rd order polynomial using the output of an ALSSM (blue) over the interval $j \in \{-50, \cdots, 50\}$. Therefore the model $s_j(x)$ from (9) is parametrized with matrices A and c from (10) and (11) respectively, and with (estimated) initial state $x = [5.79{\times}10^{-2},\; -6.91{\times}10^{-3},\; -1.23\times10^{-5},\; 1.19\times10^{-5}]^\mathsf{T}$. The dashed blue lines illustrate the progression of the polynomial model beyond the interval limits.

Further, we introduce the element-wise power of a signal $y = [\ldots, y_{-1}, y_0, y_1, y_2, \ldots]$ by a real number q which is

$$y^{\circ q} \triangleq [\ldots, y_{-1}^q,\; y_0^q,\; y_1^q,\; y_2^q, \ldots] \tag{5}$$

with trivial special case $y^{\circ 0} = [\ldots, 1, 1, 1, \ldots] = \mathbb{1}$. The element-wise power for multi-dimensional signals or matrices $Y^{\circ \ell}$ is analogous.

Finally, we define the Kronecker power for a vector x by a non-negative integer q as

$$x^{\otimes q} \triangleq \underbrace{x \otimes \cdots \otimes x}_{q \text{ times}}, \tag{6}$$

with special and trivial case $x^{\otimes 0} = 1$. For vectors $x \in \mathbb{R}^K$ we get $x^{\otimes \ell} \in \mathbb{R}^{K^\ell}$ and for a matrices $X \in \mathbb{R}^{K \times M}$ we get $X^{\otimes \ell} \in \mathbb{R}^{K^\ell \times M^\ell}$. Concerning the order of operations, the Kronecker power operation is to be evaluated analogously to exponentiation, i.e., before matrix multiplication.

II. Inner Products in One-Dimensional ALSSM Subspace

A. One-Dimensional ALSSM Signal Models

The output of a time-invariant Autonomous Linear State Space Model (ALSSM) of order N is defined as

$$x_{j+1} = Ax_j \tag{7}$$

$$y_j = cx_j \tag{8}$$

with transition matrix $A \in \mathbb{R}^{N \times N}$, output vector $c \in \mathbb{R}^{1 \times N}$, state vector $x_j \in \mathbb{R}^N$, and initialized at index $j = 0$ by the initial state x_0, subsequently simply denoted as x. Note that A and c are a fixed matrix and vector that together define the ALSSM for a given initial state. Substituting (7) in (8) leads to the closed form model

$$s_j(x) = cA^j x \;\in \mathbb{R} \;. \tag{9}$$

The initial state x also forms a real vector space of dimension N, and the ALSSM output (9) forms a subspace of the signal space.

The key idea is to project localized signals, i.e. signals within a fixed interval or window, into this ALSSM signal

subspace to make manipulation of signals computationally more efficient. For this, A, c are chosen such that the resulting ALSSM subspace holds a good approximation of y_j. It is known that many general signal models have an ALSSM representation, including (piecewise) polynomials, sinusoidals, exponentials, and sums and products thereof [5], [7]. (For more details on optimal A and c parameter selection or for learning of those, please refer to dedicated literature such as [3], [7], [22]). Once an ALSSM subspace is defined, the signal approximation is encoded by its initial state x which is mostly estimated e.g., using the least squares minimum (cf. later in Subsection III-C).

Figure 1 provides a trivial example, where a bioelectrical signal is locally approximated over a fixed interval by a 3rd order polynomial. For that, signal model (9) is parametrized using the Jordan form for polynomials (cf. [7, Section 3.3.1]) with

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4} , \tag{10}$$

$$c = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{1\times 4} \tag{11}$$

and with $x \in \mathbb{R}^4$.

B. Inner Product for One-Dimensional Signals

We consider the weighted inner product of two signals y and y' defined on $\mathbb{Z}$,

$$\langle y_{\bullet+k}, y' \rangle_w \triangleq \sum_{j\in\mathbb{Z}} w_j y_{j+k}\, y'_j \quad \in \mathbb{R} , \tag{12}$$

with window weight factor $w_j \in \mathbb{R}_+$. To find efficient computation rules for (12), we assume that one of the two signals can be approximated with an ALSSM, i.e. we will assume that y' is close to $\hat{y}'$, where

$$\hat{y}'_j = \begin{cases} s_j(x) & \text{for} \quad a \le j \le b , \\ 0 & \text{else} , \end{cases} \tag{13}$$

with interval borders $a, b \in \mathbb{Z}$ and with $s_j(x)$ as in (9) with fixed initial state vector x. (We will demonstrate later in Subsection III-C how to efficiently approximate y' by an ALSSM signal $\hat{y}'$ using the least squares optimum.)

We also note that it is sufficient if only one of the two signals in the inner product (12) can be approximated by an ALSSM, of which we will provide a proof in Section II-D.

In order to enable recursive calculation rules, we will select an exponentially decaying window, i.e., $w_j = \gamma^j$ with $\gamma \in \mathbb{R}_+$ (setting $\gamma = 1.0$ results in an unweighted Euclidean inner product, i.e., a rectangular window). Note that this is without loss of generality, as we will later show in Section V-D that several other popular windows can also be implemented without breaking the recursion.

Hence we get

$$\langle y_{\bullet+k}, \hat{y}' \rangle_w = \sum_{j\in\mathbb{Z}} \gamma^j y_{j+k}\, \hat{y}'_j \quad \in \mathbb{R} \tag{14}$$

$$= \sum_{j=a}^{b} \gamma^j y_{j+k}\, s_j(x)$$

$$= x^\mathsf{T} \xi(k, y) , \tag{15}$$

based on the sum term

$$\xi(k, y) \triangleq \sum_{j=a}^{b} \gamma^j y_{j+k} (cA^j)^\mathsf{T} \quad \in \mathbb{R}^N . \tag{16}$$

The recursion for $\xi(k, y) \to \xi(k+1, y)$ then follows in accordance to [4] as

$$\xi(k+1, y) = \gamma^{-1} A^{-\mathsf{T}} \xi(k, y) - \gamma^{a-1} (A^{a-1})^\mathsf{T} c^\mathsf{T} y_{k+a} + \gamma^b (A^b)^\mathsf{T} c^\mathsf{T} y_{k+b+1} . \tag{17}$$

We note that, while the computational complexity for (14) is $\mathcal{O}(M)$ with $M = b - a$ the length of the support of y', the complexity for (15) is reduced to $\mathcal{O}(N)$, with dimension N of the state vector x. For the recursive sum term (17) the complexity is $\mathcal{O}(N^2)$ when using a general matrix A, and is reduced to $\mathcal{O}(N)$ if A is (block) diagonal, which is e.g. the case for polynomial signal models [7, Section 3.3.1]. In summary, working in an ALSSM subspace renders the computational effort for an inner product *independent* of the signal length M.

C. Generalization to L_2 Norms

The concept of recursive computations of inner products in a sliding window manner as in (12) can be translated to the computation of weighted and sliding L_2 norms such as $\|y_{\bullet+k}\|_w^2 = \langle y_{\bullet+k}, y_{\bullet+k} \rangle_w$, which extends the field of applications to least squares problem solving, as discussed in [4] and later in Section III-C. For this, we generalize (16) to (element-wise) powered observations, modifying the sum to

$$\xi^{(q)}(k, y) \triangleq \sum_{i=k+a}^{k+b} y_i^{2-q} \left(c^{\otimes q} (\gamma A^{\otimes q})^{i-k}\right)^\mathsf{T} \in \mathbb{R}^{N^q} \tag{18}$$

with the recursion for $\xi^{(q)}(k, y) \to \xi^{(q)}(k+1, y)$ given in Table I and according to [5] (with slightly modified notation). Note that for $q = 1$ the sum term (18) becomes (16). Furthermore, choosing $q = 0$, 1, or 2, we gain the following useful relations to inner products and L_2 norms:

$$\langle y_{\bullet+k}, y_{\bullet+k} \rangle_w = \|y_{\bullet+k}\|_w^2 = \xi^{(0)}(k, y) \tag{19}$$

$$\langle y_{\bullet+k}, \hat{y}' \rangle_w = x^\mathsf{T} \xi^{(1)}(k, y) \tag{20}$$

$$\langle \hat{y}', \hat{y}' \rangle_w = \|\hat{y}'\|_w^2 = (x^{\otimes 2})^\mathsf{T} \xi^{(2)}(k, \mathbb{1}) . \tag{21}$$

The first equation is the trivial case where no ALSSM is used (i.e. it is the standard L_2 norm of y over a window). The second equation is equivalent to (14)–(16), while the third equation is obtained when replacing y with a copy of the ALSSM for $\hat{y}'$ in (14)–(16). The computational complexity of (19)–(21), not including (18), is $\mathcal{O}(N^q)$. However, applying an appropriate state transformation to the ALSSM first can

further reduce the computational load such as to $\mathcal{O}(N)$ for $\xi^{(2)}(k, \mathbb{1})$, cf. [6, Section 4.3.4].

Also note that (18) is a generalization of our previous work [5] with the correspondences $\kappa_k = \xi^{(0)}(k, Y)$, $\xi_k = \xi^{(1)}(k, Y)$, and $\mathrm{vec}(W_k) = \xi^{(2)}(k, \mathbb{1})$. A proof for $\mathrm{vec}(W_k) = \xi^{(2)}(k, \mathbb{1})$ is given in the supplementary material in Section A. III.A.

D. Inner Products of Subspace Signals

The inner product as in (14) depends on $\hat{y}'$ being an element of an ALSSM subspace. However, for the recursive calculations proposed in the previous section, the second signal y is also (implicitly) projected into the same subspace. We will see that, remarkably, this does not introduce any further inaccuracies to the results if the ALSSM is a proper model for only one of the two signals, i.e., if $\|y - \hat{y}\|$ or $\|y' - \hat{y}'\|$ is sufficiently small.

Theorem 1 (Inner product of subspace signals)**.** *Let y and y' be two signals in a full signal space, and P a projection matrix such that $\hat{y} = Py$ and $\hat{y}' = Py'$ with $\hat{y}$ and $\hat{y}'$ elements of an ALSSM subspace of the original signal space. Then, we get the equality*

$$\langle y, \hat{y}' \rangle = \langle \hat{y}, \hat{y}' \rangle = \langle \hat{y}, y' \rangle \ , \tag{22}$$

i.e., the inner product with only one signal projected to a subspace is equivalent to the inner product with both signals projected.

Proof. Based on the *Hilbert projection theorem* [23], if P is a projection matrix to a subspace S of a vector space V and $P^\perp$ the projection matrix to the orthogonal complement of S, we get

$$\begin{aligned} \langle y, Py' \rangle &= \langle Py + P^\perp y, Py' \rangle & (23)\\ &= \langle Py, Py' \rangle + \langle P^\perp y, Py' \rangle = \langle Py, Py' \rangle \ , & (24) \end{aligned}$$

since the inner product of orthogonal vector $\langle P^\perp y, Py' \rangle$ is zero. ■

Theorem 2 (Accuracy of inner products in subspace)**.** *Let y and y' be two signals in a full signal space and $\hat{y}$ and $\hat{y}'$ the respective projections in a common ALSSM subspace. Then, the upper boundary for the deviation of the inner product when derived in the subspace instead of the full space is*

$$|\langle y, y' \rangle - \langle \hat{y}, \hat{y}' \rangle| \leq \|y - \hat{y}\| \cdot \|y' - \hat{y}'\| \ . \tag{25}$$

As a consequence, it is sufficient to have either $\|y - \hat{y}\| \approx 0$ or $\|y' - \hat{y}'\| \approx 0$ to guarantee $\langle y, y' \rangle \approx \langle \hat{y}, \hat{y}' \rangle$ with equality for either $y = \hat{y}$ or $y' = \hat{y}'$.

Proof. Theorem 2 is a generalization of Theorem 1 with

$$\begin{aligned} |\langle y, y' \rangle - \langle \hat{y}, \hat{y}' \rangle| &= |\langle \hat{y} + (y - \hat{y}), \hat{y}' + (y' - \hat{y}') \rangle - \langle \hat{y}, \hat{y}' \rangle| \\ &= |\langle y - \hat{y}, y' - \hat{y}' \rangle| \\ &\leq \|y - \hat{y}\| \cdot \|y' - \hat{y}'\| \ , \end{aligned} \tag{26}$$

applying the orthogonality property of projection for the second line (with the consequence that $\langle \hat{y}, y' - \hat{y}' \rangle = 0$ and $\langle \hat{y}', y - \hat{y} \rangle = 0$ since $\hat{y}$ and $\hat{y}'$ are elements of the same subspace), and the *Cauchy–Schwarz inequality* for the last line. ■

An intuitive explanation for Theorem 1 and 2 is as follows: The inner product as in (14) involves two vectors, one of which lies in a lower-dimensional subspace of the vector space where the inner product is computed. However, projecting onto this subspace explicitly zeros out (after appropriate space rotation) the orthogonal complement dimensions, rendering those dimensions irrelevant for the inner product result. Or, to phrase it differently, inner products are only affected by dimensions which are non-zero in *both* vectors.

The consequences of Theorems 1 and 2 are demonstrated impressively in Fig. 2, illustrating that it is sufficient to have one of the two signals in an inner product well approximated by an ALSSM subspace.

III. Application to One-Dimensional Signals

The following applications apply linear filtering on one-dimensional signals. It is assumed that appropriate measures, such as zero padding, are taken to prevent bias towards the edges of the signal.

A. Cross-Correlation

Let y be an input signal of arbitrary length and $\hat{h}$ a signal in the ALSSM subspace with ALSSM initial state vector x, subsequently denoted as $x(\hat{h})$. The cross-correlation between these two signals at sample-shift k is then

$$\begin{aligned} (y \star \hat{h})_k = \langle y_{\bullet+k}, \hat{h} \rangle_w &= \sum_{j=a}^{b} \gamma^j y_{j+k} \hat{h}_j \\ &= x(\hat{h})^\mathsf{T} \xi(k, y) \ , \end{aligned} \tag{27}$$

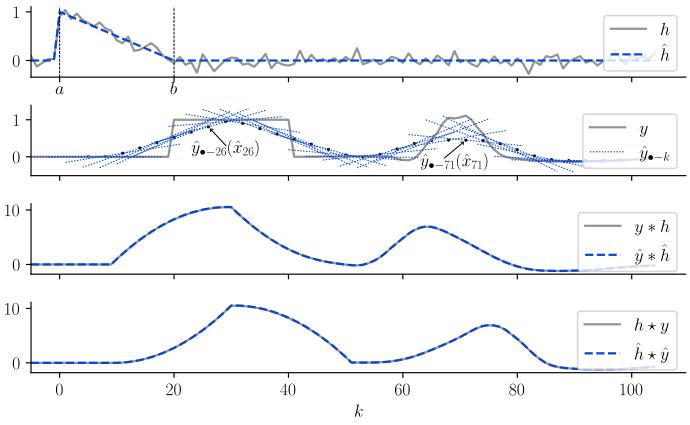
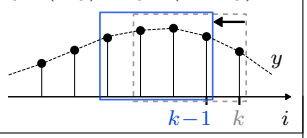
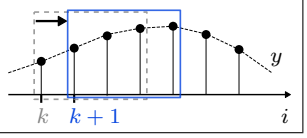


Fig. 2. Example of one-dimensional discrete-time convolutions and correlations in signal space and an ALSSM subspace spanned by a straight line model (ALSSM of system order $N = 2$ with interval borders $a = 0$ and $b = 20$. The ALSSM signal is assumed to be zero elsewhere). The top plot shows an impulse response h (solid grey line) and its projection into the ALSSM subspace (dashed blue line) at time index 0. The second plot shows the input signal y (solid grey line), and examples of the local projections to the ALSSM subspace $\hat{y}_{\bullet-k}$ shifted by k samples (dashed blue line). The third plot shows the comparison of the convolution $(y * h)$ and $(\hat{y} * \hat{h})$. The last plot shows the comparison of the cross-correlation $(h \star y)$ and $(\hat{h} \star \hat{y})$. This example shows that, if only one of the two signals is sufficiently well approximated by the ALSSM subspace (while the other one is not), the resulting convolution and correlation are still very accurate. We provide a mathematical proof for this fact in Section II-D.

TABLE I
ONE-DIMENSIONAL RECURSIVE ξ TERM COMPUTATIONS IN FORWARD AND BACKWARD DIRECTION FOR EXPONENTIAL WINDOWS

Recursions over input dimension (index k) for $q \in \{0,1,2\}$ with ALSSM parameters $A \in \mathbb{R}^{N\times N}, c \in \mathbb{R}^{1\times N}, \gamma \in \mathbb{R}_+, N \in \mathbb{N}, a, b \in \mathbb{Z}$ for $\xi^{(q)}(k,y) = \sum_{i=k+a}^{k+b} y_i^{2-q} \big(c^{\otimes q}(\gamma A^{\otimes q})^{i-k}\big)^\mathsf{T} \in \mathbb{R}^{N^q}$

Direction	Recursion	Example
"Forward" $k \to k+1$ $\gamma > 1$	$\xi^{(0)}(k+1,y) = \gamma^{-1}\xi^{(0)}(k,y) - \gamma^{a-1} y_{k+a}^2 + \gamma^b y_{k+b+1}^2$ $\xi^{(1)}(k+1,y) = \gamma^{-1}A^{-\mathsf{T}}\xi^{(1)}(k,y) - \gamma^{a-1} y_{k+a}(cA^{a-1})^\mathsf{T} + \gamma^b y_{k+b+1}(cA^b)^\mathsf{T}$ $\xi^{(2)}(k+1,\mathbb{1}) = \gamma^{-1}(A^{\otimes 2})^{-\mathsf{T}}\xi^{(2)}(k,\mathbb{1}) - \gamma^{a-1}\big(c^{\otimes 2}(A^{\otimes 2})^{a-1}\big)^\mathsf{T} + \gamma^b\big(c^{\otimes 2}(A^{\otimes 2})^b\big)^\mathsf{T}$	$\xi^{(1)}(k,y) \to \xi^{(1)}(k-1,y)$ 
"Backward" $k \to k-1$ $\gamma < 1$	$\xi^{(0)}(k-1,y) = \gamma\xi^{(0)}(k,y) + \gamma^a y_{k+a-1}^2 - \gamma^{b+1} y_{k+b}^2$ $\xi^{(1)}(k-1,y) = \gamma A^\mathsf{T}\xi^{(1)}(k,y) + \gamma^a y_{k+a-1}(cA^a)^\mathsf{T} - \gamma^{b+1} y_{k+b}(cA^{b+1})^\mathsf{T}$ $\xi^{(2)}(k-1,\mathbb{1}) = \gamma(A^{\otimes 2})^\mathsf{T}\xi^{(2)}(k,\mathbb{1}) + \gamma^a\big(c^{\otimes 2}(A^{\otimes 2})^a\big)^\mathsf{T} - \gamma^{b+1}\big(c^{\otimes 2}(A^{\otimes 2})^{b+1}\big)^\mathsf{T}$	$\xi^{(1)}(k,y) \to \xi^{(1)}(k+1,y)$ 

which for $\gamma \approx 1.0$ is the unweighted cross-correlation. An example of this is provided in Fig. 2.

B. Convolution

The convolution is in principle a cross-correlation with a time-reversed signal. Let y be an input signal of arbitrary length and $\hat{h}$ a signal in the ALSSM subspace. (For LTI filters, $\hat{h}$ corresponds to the filter impulse response). The output at index k for the convolution of these two signals is then

$$\begin{aligned}(y * \hat{h})_k &= \sum_{j=a}^{b} \gamma^j y_{k-j}\hat{h}_j \\ &= x(\hat{h})^\mathsf{T} \sum_{j=a}^{b} \gamma^j y_{-j+k}(cA^j)^\mathsf{T} \\ &= x(\hat{h})^\mathsf{T}\xi(k, y_{-\bullet}; \theta) \qquad (28) \\ &= x(\hat{h})^\mathsf{T}\xi(k, y; \theta^{-1}) \qquad (29)\end{aligned}$$

with parameters $\theta = \{A, a, b, \gamma\}$ for the first option (28) and with parameters $\theta^{-1} = \{A^{-1}, -b, -a, \gamma^{-1}\}$ for the second option (29) with the assumption that A^{-1} exists. For the sake of clarity, we here append the parameter set θ to the sum term (16) which is a collection of otherwise implicit parameters (ALSSM output vector c is not an element of the list since it remains unchanged for all sum terms). While the first option takes a time-reversed input signal, the second only inverts the ALSSM parameter set. This second option is of particular interest for causal online and low-memory convolution implementations. Figure 2 provides a convolution example.

C. Least-Squares Signal Approximation by an ALSSM

ALSSMs are mostly used for *local*, i.e., windowed signal approximations, as their outputs are in general not bounded.

We here consider the L_2 cost as in [5], which is

$$\begin{aligned}J_a^b(k,x) &= \sum_{i=k+a}^{k+b} \gamma^{i-k}\big[y_i - s_{i-k}(x)\big]^2 \\ &= \|y_{\bullet+k} - \hat{y}\|_w^2 \qquad (30)\end{aligned}$$

with signal y, signal model $\hat{y}$ with $\hat{y}_j = s_j(x) = cA^j x$ as in (13) with ALSSM initial state x, finite or infinite boundaries $a, b \in \mathbb{Z}, a < b$ and a decay factor $\gamma \in \mathbb{R}_+$. The cost (30) then expands to

$$\begin{aligned}J_a^b(k,x) &= \|y_{\bullet+k}\|_w^2 - 2\langle y_{\bullet+k}, \hat{y}\rangle_w + \|\hat{y}\|_w^2 \\ &= \xi^{(0)}(k,y) - 2x^\mathsf{T}\xi^{(1)}(k,y) + (x^{\otimes 2})^\mathsf{T}\xi^{(2)}(k,\mathbb{1})\,, \qquad (31)\end{aligned}$$

where the last step follows from (19)-(21). This cost is then minimized by the optimum initial state

$$\hat{x}_k = \operatorname*{argmin}_x J_a^b(k,x) = W_k^{-1}\,\xi^{(1)}(k,y)\,, \qquad (32)$$

with

$$\operatorname{vec}(W_k) = \xi^{(2)}(k,y) = \xi^{(2)}(k,\mathbb{1}) \qquad (33)$$

where $\operatorname{vec}(W_k)$ denotes the vectorized form of W_k obtained by stacking the columns of the matrix W_k on top of one another, and with $\mathbb{1}$ denoting a signal or vector of ones with equal dimension as y (since $y^{\circ 0} = \mathbb{1}$). We note that $\xi^{(2)}(k,\mathbb{1})$ does not dependent on the input signal and reaches a steady state. Therefore, for all experiments in this paper, $W_k = W$ is assumed to be constant, and $W_k^{-1} = W^{-1}$ is precomputed. The recursions for $\xi^{(q)}(k,y) \to \xi^{(q)}(k+1,y)$ are listed in Table I. (Despite being constant in this work, the recursion for $\xi^{(2)}(k,\mathbb{1})$ is also listed. This becomes relevant when costs with weighted samples are introduced, as proposed in [5], where W_k can no longer be assumed to be constant.)

Note that the optimum state estimate $\hat{x}_k$ from (32) can directly be used as the ALSSM signal approximation in (15), i.e., no additional projection step is needed from the signal space to the ALSSM space. Finally, (30) is a recursively computable form of (3) for any $\mathcal{A}$ which can be modeled

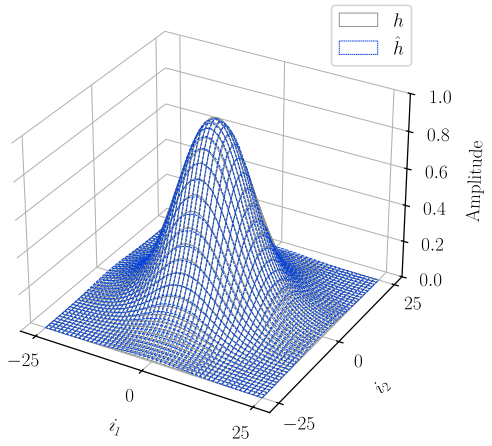


Fig. 3. Sampled Gaussian kernel h with its ALSSM approximation $\hat{h}$ as in (35) using a 2-D polynomial of degree two in each dimension. Note that the two signals are almost identical and therefore overlap in the illustration (MSE of $1.0 \cdot 10^{-3}$). The application of this approximated filter is shown in Fig. 4.

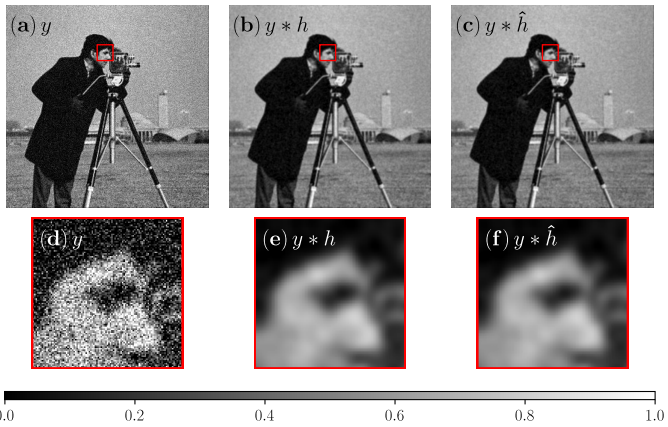


Fig. 4. Convolution of an image with a Gaussian kernel or its ALSSM approximation. Gaussian white noise was added to an image (a), which is then filtered by a sampled gaussian kernel and an ALSSM approximation of the kernel, as shown in Fig. 3 using (55). The convolution results are shown in (b) and (c), and (d)-(f) each show a zoomed section of the upper subplots. We note that the resulting error of the ALSSM convolution is very small with an MSE of $4.4 \cdot 10^{-6}$.

by an ALSSM, i.e., for any $\mathcal{A}$ with rows represented by cA^j (with j incrementing with row index).

IV. Inner Products in Multi-Dimensional ALSSM Subspace

In the following section, we extend the inner product for single-dimension signals from Section II to multi-dimensional input signals, thereby also generalizing the correlation, convolution and least-squares approximation explained in Section III. Indices are extended to tuples, i.e. $\boldsymbol{i} = (i_1, \ldots, i_L)$ and $\boldsymbol{k} = (k_1, \ldots, k_L)$, with $L \in \mathbb{N}$ the number of dimensions. We will also define a multi-dimensional version of the sum term ξ from (16), with the incorporated dimensions noted in subscript, i.e. ξ_1 would be a one-dimensional sum over the first dimension, and ξ_{12} would be a two-dimensional sum over the first and second dimension.

A. The L-Dimensional ALSSM as a Signal Model

We define an L-dimensional ALSSM by the tensor product of L ALSSMs, with each model having its own system order N_ℓ and parameters $\{A_\ell, c_\ell\}$. The output of such a model at index tuple $\boldsymbol{i} = (i_1, \ldots, i_L)$ with separated per-dimension initial states $x_1, \ldots, x_L$ is

$$\begin{aligned} s_{\boldsymbol{i}}(x) &= \big(c_1 A_1^{i_1} x_1 \otimes \cdots \otimes c_L A_L^{i_L} x_L\big) \\ &= \big(c_1 \otimes \cdots \otimes c_L\big)\big(A_1^{i_1} \otimes \cdots \otimes A_L^{i_L}\big)\big(x_1 \otimes \cdots \otimes x_L\big)\,, \end{aligned} \tag{34}$$

and with a joint initial state vector $x \in \mathbb{R}^{\prod_{\ell=1}^{L} N_\ell}$ the model is

$$s_{\boldsymbol{i}}(x) = \big(c_1 \otimes \cdots \otimes c_L\big)\big(A_1^{i_1} \otimes \cdots \otimes A_L^{i_L}\big)x \quad \in \mathbb{R}\,. \tag{35}$$

An example of a two-dimensional model is shown in Fig. 3. We note that the ALSSM can be different for each dimension, e.g., the first dimension could be modeled as a polynomial while the second dimension can be modeled as an expontially damped sinusoid. However, in many cases, the parameters A_ℓ and c_ℓ would be chosen as equal for all dimensions ℓ, which would then also simplify the equations. Nevertheless, we allow different ALSSMs for each dimension in the remainder of the paper for the sake of generality.

B. Inner Product for Two-Dimensional Signals

To derive the inner product of L-dimensional signals, we start with signals of dimension two. The inner product (12) generalizes for two-dimensional signals Y and Y', both defined on $\mathbb{Z}^2$, as

$$\begin{aligned} &\big\langle Y_{(\bullet+k_1,\bullet+k_2)}, Y' \big\rangle_w \\ &\qquad \triangleq \sum_{j_1 \in \mathbb{Z}} \sum_{j_2 \in \mathbb{Z}} w_{(j_1,j_2)} Y_{(j_1+k_1,j_2+k_2)}\, Y'_{(j_1,j_2)} \quad \in \mathbb{R}\,. \end{aligned} \tag{36}$$

To find efficient recursive computation rules for (36), we analogously restrict Y' to $\hat{Y}'$, an element of a two-dimensional ALSSM subspace (35) with $L = 2$ and $\hat{Y}'_j = \big(c_1 \otimes c_2\big)\big(A_1^{i_1} \otimes A_2^{i_2}\big)x$ and initial state x. Further, we set a finite or infinite support of boundaries $a_1, b_1 \in \mathbb{Z}$ in the first dimension, and $a_2, b_2 \in \mathbb{Z}$ in the second dimension for $\hat{Y}'$. As in Section II-A, we initially restrict ourselves to an exponentially weighted window, and later extend this to other windows. Therefore, for now, the window weights are set to $w_{(i_1,i_2)} = \gamma_1^{i_1}\gamma_2^{i_2}$ with $\gamma_1, \gamma_2 \in \mathbb{R}_+$. The inner product then generalizes to

$$\begin{aligned} &\big\langle Y_{(\bullet+k_1,\bullet+k_2)}, \hat{Y}' \big\rangle \\ &\quad \triangleq \sum_{j_1=a_1}^{b_1} \sum_{j_2=a_2}^{b_2} \gamma_1^{j_1}\gamma_2^{j_2} Y_{(j_1+k_1,j_2+k_2)}\, \hat{Y}'_{(j_1,j_2)} \\ &\quad = \sum_{j_1=a_1}^{b_1} \sum_{j_2=a_2}^{b_2} \gamma_1^{j_1}\gamma_2^{j_2} Y_{(j_1+k_1,j_2+k_2)} s_{(j_1,j_2)}(x) \\ &\quad = x^{\mathsf{T}} \sum_{j_1=a_1}^{b_1} \sum_{j_2=a_2}^{b_2} \gamma_1^{j_1}\gamma_2^{j_2} Y_{(j_1+k_1,j_2+k_2)} \big(A_1^{j_1} \otimes A_2^{j_2}\big)^{\mathsf{T}} \big(c_1 \otimes c_2\big)^{\mathsf{T}} \\ &\quad = x^{\mathsf{T}} \xi_{12}\big((k_1,k_2), Y\big) \quad \in \mathbb{R}\,. \end{aligned} \tag{37}$$

In the supplementary material in Section A.II., we prove that the double sum term $\xi_{12}\big((k_1,k_2), Y\big)$ can be split into two

successive summations, where we first sum over the first dimension with

$$\xi_1\big((k_1,k_2),Y\big)=\sum_{i_1=k_1+a_1}^{k_1+b_1} Y_{(i_1,k_2)}\big(c_1(\gamma_1 A_1)^{i_1-k_1}\big)^{\mathsf{T}} \in \mathbb{R}^{N_1}, \quad (38)$$

and use the result in the sum over the second dimension,

$$\xi_{12}((k_1,k_2),Y)=\sum_{i_2=k_2+a_2}^{k_2+b_2} \xi_1\big((k_1,i_2),Y\big)\otimes\big(c_2(\gamma_2 A_2)^{i_2-k_2}\big)^{\mathsf{T}}. \quad (39)$$

We note that (38) is equivalent to (16), which is taken as an input to (39), such that (38) and (39) are indeed found to be a generalization of the one-dimensional case (16). Again, the quantity ξ_{12} can be computed recursively over both indices k_1 and k_2, similar to the one-dimensional case, and the recursion now applies in two dimensions. The forward and backward recursions to compute (39) are a generalization of the terms from Table I and are listed in Table II, where (39) is equivalent to $\xi_{12}^{(q)}$ for $q=1$. All derivations are provided in the supplementary material in Sec. A. V. In terms of computational complexity, the effort to compute (39) for each increment of both k_1 and k_2 is $\mathcal{O}(N_1 N_2)$ with ALSSM system orders N_1 and N_2. We note that the complexity of an inner product between a sample space and an ALSSM subspace signal (37) also scales with $\mathcal{O}(N_1 N_2)$, and therefore remains independent of the actual signal length.

Providing a numerical example from image processing, convolving a signal with a 50×50 2D kernel as in Fig. 4 in the sample space requires $50\cdot 50 = 2500$ scalar multiplications and additions, i.e., multiply accumulate (MAC) operations per output pixel. By contrast, the same convolution in an ALSSM subspace of dimensions 3×3 requires $3\cdot 3 = 9$ MAC plus $3\cdot 3 + 3\cdot 9 = 36$ MAC operations for the ξ term per pixel, which is a reduction of computations by a factor ≥ 50. For separable kernels, where the 2-D convolution is substitutable by two 1-D convolutions, the speed-up factor in this example reduces to ≥ 4.

C. Inner Product for L-Dimensional Signals

The generalization of the inner product to an L-dimensional input signal Y defined on $\mathbb{Z}^L$ with indices $\boldsymbol{i}=(i_1,...,i_L)$, $\boldsymbol{k}=(k_1,\ldots,k_L)$, and using the short $y_{\bullet+\boldsymbol{k}} = y_{(\bullet+k_1,\ldots,\bullet+k_L)}$ is

$$\langle Y_{\bullet+\boldsymbol{k}}, Y'\rangle_w \triangleq \sum_{j_1=a_1}^{b_1}\cdots\sum_{j_L=a_L}^{b_L} w_{\boldsymbol{j}} Y_{\boldsymbol{j}+\boldsymbol{k}} Y'_{\boldsymbol{j}} \quad \in \mathbb{R}\ . \quad (40)$$

Restricting Y' to elements of the L-dimensional ALSSM subspace $\hat{Y}'$ with $\hat{Y}'_{\boldsymbol{j}} = \big(c_1\otimes\cdots\otimes c_L\big)\big(A_1^{j_1}\otimes\cdots\otimes A_L^{j_L}\big)x$, and, in analogy to (13), with finite or infinite support with per dimension boundaries a_ℓ and b_ℓ, leads to

$$\begin{aligned}\langle Y_{\bullet+\boldsymbol{k}}, \hat{Y}'\rangle_w &= \sum_{j_1=a_1}^{b_1}\cdots\sum_{j_L=a_L}^{b_L} \gamma_1^{j_1}\cdots\gamma_L^{j_L} Y_{\boldsymbol{j}+\boldsymbol{k}} s_{\boldsymbol{j}}(x) \ \in\mathbb{R}\\ &= x^{\mathsf{T}}\xi_{12\cdots L}(\boldsymbol{k},Y)\ , \end{aligned} \quad (41)$$

with the recursively defined sum term $\xi_{12\cdots L}$ defined as

$$\begin{aligned}&\xi_{*\ell}(\boldsymbol{k},Y)\\ &\quad\triangleq \sum_{i_\ell=k_\ell+a_\ell}^{k_\ell+b_\ell} \xi_*\big((k_1,...,i_\ell,...,k_L),Y\big)\otimes\big(c_\ell(\gamma_\ell A_\ell)^{i_\ell-k_\ell}\big)^{\mathsf{T}}, \end{aligned} \quad (42)$$

where $*$ is a substitute for the sequence of already incorporated dimensions. The first recursion (where $*=\emptyset$, an empty set) is initialized with the observations as input,

$$\xi_\emptyset(\boldsymbol{k},Y) = Y_{\boldsymbol{k}}\ . \quad (43)$$

Note that the subindices in the sum term $\xi_{12\cdots L}$ such as in (41) list the order of the incorporated dimensions.

To provide an example, the notation ξ_{123} indicates that the sum has sequentially progressed through the dimensions 1, 2, and 3. For a three-dimensional observation Y defined on $\mathbb{Z}^3$ and using the three-dimensional signal model with per-dimension ALSSM parameters $\{c_1,A_1\}$, $\{c_2,A_2\}$, and $\{c_3,A_3\}$, the recursion for ξ_{123} unrolls to the three steps

$$\xi_1(\boldsymbol{k},Y)\triangleq\sum_{i_1=k_1+a_1}^{k_1+b_1} Y_{(i_1,k_2,k_3)}\big(c_1(\gamma A_1)^{i_1-k_1}\big)^{\mathsf{T}} \quad (44)$$

$$\xi_{12}(\boldsymbol{k},Y)\triangleq\sum_{i_2=k_2+a_2}^{k_2+b_2} \xi_1\big((k_1,i_2,k_3),Y\big)\otimes\big(c_2(\gamma_2 A_2)^{i_2-k_2}\big)^{\mathsf{T}} \quad (45)$$

$$\xi_{123}(\boldsymbol{k},Y)\triangleq\sum_{i_3=k_3+a_3}^{k_3+b_3} \xi_{12}\big((k_1,k_2,i_3),Y\big)\otimes\big(c_3(\gamma_3 A_3)^{i_3-k_3}\big)^{\mathsf{T}}. \quad (46)$$

Note that, although this may be implied in this example, the dimensions can be processed in any order, leading to alternative sum terms such as $\xi_{213}(\boldsymbol{k},Y)$, etc.

The forward or backward recursions to efficiently compute (42) are provided in Table III, which generalizes Table II. The recursive form of (42) scales with $\mathcal{O}(\prod_{\ell=1}^{L} N_\ell)$. Table IV provides a comparison of the complexity of L-dimensional convolutions in sample space versus ALSSM subspace for both separable and non-separable filters.

D. Generalization to L_2 Norms

The generalization of the sum term (42) to element-wise powered observations leads to weighted and sliding L_2 norms such as $\|Y_{\bullet+\boldsymbol{k}}\|_w^2 \triangleq \langle Y_{\bullet+\boldsymbol{k}}, Y_{\bullet+\boldsymbol{k}}\rangle_w$. For L-dimensional signals, the generalization is

$$\begin{aligned}\xi_{*\ell}^{(q)}(\boldsymbol{k},Y) \ &\triangleq \sum_{i_\ell=k_\ell+a_\ell}^{k_\ell+b_\ell} \xi_*^{(q)}\big((k_1,...,i_\ell,...,k_L),Y\big)\\ &\qquad\otimes\big(c_\ell^{\otimes q}(\gamma_\ell A_\ell^{\otimes q})^{i_\ell-k_\ell}\big)^{\mathsf{T}}, \end{aligned} \quad (47)$$

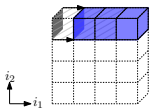
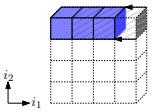
initializing the first recursion with the observation as input

$$\xi_\emptyset^{(q)}(\boldsymbol{k},Y) = Y_{\boldsymbol{k}}^{\circ(2-q)}\ . \quad (48)$$

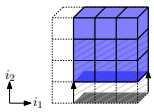
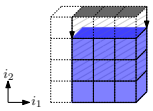
Note that for $q=1$ the sum term (47) becomes (42). This leads to the relations for the L_2 norm of multi-dimensional signals with inner product (40) for any signal Y defined on

TABLE II
TWO-DIMENSIONAL RECURSIVE ξ TERM COMPUTATIONS IN FORWARD AND BACKWARD DIRECTION FOR EXPONENTIAL WINDOWS.

Recursions over 1st input dimension (index k_1) with ALSSM parameters $A_1 \in \mathbb{R}^{N_1 \times N_1}, c_1 \in \mathbb{R}^{1 \times N_1}, \gamma_1 \in \mathbb{R}_+, N_1 \in \mathbb{N}$ for $\xi_1^{(q)}\big((k_1,k_2),Y\big) = \sum_{i_1=k_1+a_1}^{k_1+b_1} Y_{(i_1,k_2)}^{\circ(2-q)} \big(c_1^{\otimes q}(\gamma_1 A_1^{\otimes q})^{i_1-k_1}\big)^\mathsf{T}$

Direction	Recursion	Complexity[1]	Example
"Forward" $k_1 \to k_1+1$ $\gamma_1 > 1$	$\xi_1^{(q)}\big((k_1+1,k_2),Y\big) = \gamma_1^{-1}(A_1^{\otimes q})^{-\mathsf{T}}\xi_1^{(q)}\big((k_1,k_2),Y\big)$ $- \gamma_1^{a_1-1} Y_{(k_1+a_1,k_2)}^{\circ(2-q)}\big(c_1^{\otimes q}(A_1^{\otimes q})^{a_1-1}\big)^\mathsf{T}$ $+ \gamma_1^{b_1} Y_{(k_1+b_1+1,k_2)}^{\circ(2-q)}\big(c_1^{\otimes q}(A_1^{\otimes q})^{b_1}\big)^\mathsf{T}$	$\mathcal{O}(N_1^q)$ or $\mathcal{O}(N_1^{2q})$	$\xi_1^{(1)}(k_1,k_2) \to \xi_1^{(1)}(k_1+1,k_2)$ 
"Backward" $k_1 \to k_1-1$ $\gamma_1 < 1$	$\xi_1^{(q)}\big((k_1-1,k_2),Y\big) = \gamma_1(A_1^{\otimes q})^\mathsf{T}\xi_1^{(q)}\big((k_1,k_2),Y\big)$ $+ \gamma_1^{a_1} Y_{(k_1+a_1-1,k_2)}^{\circ(2-q)}\big(c_1^{\otimes q}(A_1^{\otimes q})^{a_1}\big)^\mathsf{T}$ $- \gamma_1^{b_1+1} Y_{(k_1+b_1,k_2)}^{\circ(2-q)}\big(c_1^{\otimes q}(A_1^{\otimes q})^{b_1+1}\big)^\mathsf{T}$	$\mathcal{O}(N_1^q)$ or $\mathcal{O}(N_1^{2q})$	$\xi_1^{(1)}(k_1-1,k_2) \leftarrow \xi_1^{(1)}(k_1,k_2)$ 

Recursions over 2nd input dimension (index k_2) with ALSSM parameters $A_2 \in \mathbb{R}^{N_2 \times N_2}, c_2 \in \mathbb{R}^{1 \times N_2}, \gamma_2 \in \mathbb{R}_+, N_2 \in \mathbb{N}$ for $\xi_{12}^{(q)}\big((k_1,k_2),Y\big) = \sum_{i_2=k_2+a_2}^{k_2+b_2} \xi_1^{(q)}\big((k_1,i_2),Y\big) \otimes \big(c_2^{\otimes q}(\gamma_2 A_2^{\otimes q})^{i_2-k_2}\big)^\mathsf{T}$

Direction	Recursion	Complexity[2]	Example
"Forward" $k_2 \to k_2+1$ $\gamma_2 > 1$	$\xi_{12}^{(q)}\big((k_1,k_2+1),Y\big) = \gamma_2^{-1}\big(I_{N_1}^{\otimes q} \otimes (A_2^{\otimes q})^{-1}\big)^\mathsf{T}\xi_{12}^{(q)}\big((k_1,k_2),Y\big)$ $- \gamma_2^{a_2-1}\xi_1^{(q)}\big((k_1,k_2+a_2),Y\big) \otimes \big(c_2^{\otimes q}(A_2^{\otimes q})^{a_2-1}\big)^\mathsf{T}$ $+ \gamma_2^{b_2}\xi_1^{(q)}\big((k_1,k_2+b_2+1),Y\big) \otimes \big(c_2^{\otimes q}(A_2^{\otimes q})^{b_2}\big)^\mathsf{T}$	$\mathcal{O}(N_1^q N_2^q)$	$\xi_{12}^{(1)}(k_1,k_2) \to \xi_{12}^{(1)}(k_1,k_2+1)$ 
"Backward" $k_2 \to k_2-1$ $\gamma_2 < 1$	$\xi_{12}^{(q)}\big((k_1,k_2-1),Y\big) = \gamma_2\big(I_{N_1}^{\otimes q} \otimes (A_2^{\otimes q})\big)^\mathsf{T}\xi_{12}^{(q)}\big((k_1,k_2),Y\big)$ $+ \gamma_2^{a_2}\xi_1^{(q)}\big((k_1,k_2+a_2-1),Y\big) \otimes \big(c_2^{\otimes q}(A_2^{\otimes q})^{a_2}\big)^\mathsf{T}$ $- \gamma_2^{b_2+1}\xi_1^{(q)}\big((k_1,k_2+b_2),Y\big) \otimes \big(c_2^{\otimes q}(A_2^{\otimes q})^{b_2+1}\big)^\mathsf{T}$	$\mathcal{O}(N_1^q N_2^q)$	$\xi_{12}^{(1)}(k_1,k_2-1) \leftarrow \xi_{12}^{(1)}(k_1,k_2)$ 

[1] For a (block) diagonal state transition matrix $A_1^{\otimes q}$, the complexity is $\mathcal{O}(N_1^q)$. For a general matrix $A_1^{\otimes q}$, the complexity is $\mathcal{O}(N_1^{2q})$.
[2] $\big(I_{N_1}^{\otimes q} \otimes (A_2^{\otimes q})\big)$ is always (block) diagonal.

$\mathbb{Z}^L$ and signal $\hat{Y}'$ as an element of an L-dimensional ALSSM subspace,

$$\langle Y_{\bullet+\boldsymbol{k}}, Y_{\bullet+\boldsymbol{k}}\rangle_w = \|Y_{\bullet+\boldsymbol{k}}\|_w^2 = \xi_{12\cdots L}^{(0)}(\boldsymbol{k}, Y) \tag{49}$$

$$\langle Y_{\bullet+\boldsymbol{k}}, \hat{Y}'\rangle_w = x^\mathsf{T}\xi_{12\cdots L}^{(1)}(\boldsymbol{k}, Y) \tag{50}$$

$$\langle \hat{Y}', \hat{Y}'\rangle_w = \|\hat{Y}'\|_w^2 = (x_1^{\otimes 2}\otimes\cdots\otimes x_L^{\otimes L})^\mathsf{T}\xi_{12\cdots L}^{(2)}(\boldsymbol{k}, \mathbb{1}) \tag{51}$$

$$= (R^T x^{\otimes 2})^\mathsf{T}\,\xi_{12\cdots L}^{(2)}(\boldsymbol{k}, \mathbb{1})\ , \tag{52}$$

using in the last line the permutation matrix $R = R_1 R_2 \cdots R_L$ where $R_\ell = I_{N_1\cdots N_{\ell-1}}^{\otimes 2} \otimes I_{N_\ell} \otimes K_{(N_\ell, N_{\ell+1}\cdots N_L)} \otimes I_{N_{\ell+1}\cdots N_L}$ and with commutation matrix $K_{(m,n)} \in \mathbb{R}^{mn\times mn}$ from [24].

V. APPLICATION ON MULTI-DIMENSIONAL SIGNALS

A. Cross-Correlation

Let Y be an L-dimensional input signal and $\hat{H}$ a signal represented by an L-dimensional ALSSM subspace and with associated ALSSM initial state vector x, subsequently denoted as $x(\hat{H})$. The cross-correlation between these two signals Y and $\hat{H}$ at index $\boldsymbol{k} = (k_1, \cdots, k_L)$ is then

$$\big(Y \star \hat{H}\big)_{\boldsymbol{k}} = \langle Y_{\bullet+\boldsymbol{k}}, \hat{H}\rangle_w = \sum_{j_1=a_1}^{b_1} \cdots \sum_{j_L=a_L}^{b_L} \gamma_1^{j_1}\cdots\gamma_L^{j_L} Y_{\boldsymbol{j}+\boldsymbol{k}}\hat{H}_{\boldsymbol{j}}$$

$$= x(\hat{H})^\mathsf{T}\xi_{12\cdots L}^{(1)}(\boldsymbol{k}, Y)\ . \tag{53}$$

B. Convolution

Let Y be an L-dimensional input signal and $\hat{H}$ a signal of the same dimension in the ALSSM subspace. The output at index $\boldsymbol{k} = (k_1, \ldots, k_L)$ of the convolution of these two signals Y and $\hat{H}$ is then

$$\big(Y * \hat{H}\big)_{\boldsymbol{k}} = \sum_{j_1=a_1}^{b_1} \cdots \sum_{j_L=a_L}^{b_L} \gamma_1^{j_1}\cdots\gamma_L^{j_\ell} Y_{\boldsymbol{k}-\boldsymbol{j}}\hat{H}_{\boldsymbol{j}}$$

$$= \sum_{j_1=a_1}^{b_1} \cdots \sum_{j_L=a_L}^{b_L} \gamma_1^{j_1}\cdots\gamma_L^{j_L} Y_{\boldsymbol{k}-\boldsymbol{j}}\, s_{\boldsymbol{j}}(x(\hat{H}))$$

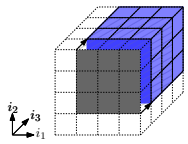
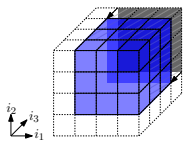
$$= x(\hat{H})^\mathsf{T}\xi_{12\cdots L}^{(1)}(\boldsymbol{k}, Y_{-\bullet}; \boldsymbol{\theta}) \tag{54}$$

$$= x(\hat{H})^\mathsf{T}\xi_{12\cdots L}^{(1)}(\boldsymbol{k}, Y; \boldsymbol{\theta}^{-1}) \tag{55}$$

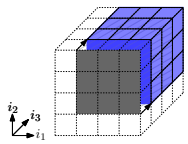
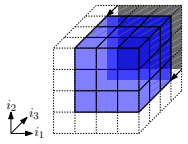
For the first option (54), we input $Y_{-\bullet}$, the signal time-reversed in all dimensions, and parameter set $\boldsymbol{\theta} = \{\boldsymbol{A}, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\gamma}\}$ with $\boldsymbol{A} = (A_1, \ldots, A_L)$, $\boldsymbol{a} = (a_1, \ldots, a_L)$, $\boldsymbol{b} = (b_1, \ldots, b_L)$, and $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_L)$. For the second option (55), we use the "inverse" parameter set $\boldsymbol{\theta}^{-1} = \{\boldsymbol{A}^{-1}, -\boldsymbol{b}, -\boldsymbol{a}, \boldsymbol{\gamma}^{-1}\}$ with $\boldsymbol{A}^{-1} = (A_1^{-1}, \ldots, A_l^{-1})$ indicating the inverse for each entry of the tuple, and likewise for $\boldsymbol{\gamma}^{-1}$. While the first option takes a time-reversed input signal, the second option only inverts the ALSSM transition matrices and time-reverses the boundaries.

TABLE III
L-DIMENSIONAL RECURSIVE ξ TERM COMPUTATIONS IN FORWARD AND BACKWARD DIRECTION FOR EXPONENTIAL (UPPER TABLE) AND GENERAL ALSSM WINDOWS (LOWER TABLE).

Recursions with exponential windows ($*$ substitutes sequence of already incorporated dimensions) for
$\xi^{(q)}_{*\ell}(\boldsymbol{k}, Y) = \sum_{i_\ell=k_\ell+a_\ell}^{k_\ell+b_\ell} \xi^{(q)}_{*}\big((k_1,...,i_\ell,...,k_L),Y\big) \otimes \big(c_\ell^{\otimes q}(\gamma_\ell A_\ell^{\otimes q})^{i_\ell-k_\ell}\big)^{\mathsf{T}}$
over the ℓ-th input dimension with index k_ℓ and ALSSM parameters $A_\ell \in \mathbb{R}^{N_\ell \times N_\ell}, c_\ell \in \mathbb{R}^{1\times N_\ell}, \gamma_\ell \in \mathbb{R}_+, N_\ell \in \mathbb{N}$.

Direction	Recursion	Complexity $\mathcal{O}(\cdot)$	Example ($L=3$)
"Forward" $k_\ell \to k_\ell+1$ $\gamma_{k_\ell} > 1$	$\xi^{(q)}_{*\ell}\big((...,k_\ell+1,...),Y\big)$ $= \gamma_\ell^{-1}\big(I_{N_1}^{\otimes q} \otimes \cdots \otimes I_{N_{\ell-1}}^{\otimes q} \otimes (A_\ell^{\otimes q})^{-1}\big)^{\mathsf{T}} \xi^{(q)}_{*\ell}\big((...,k_\ell,...),Y\big)$ $- \gamma_\ell^{a_\ell-1}\xi^{(q)}_{*}\big((...,k_\ell+a_\ell,...),Y\big) \otimes \big(c_\ell^{\otimes q}(A_\ell^{\otimes q})^{a_\ell-1}\big)^{\mathsf{T}}$ $+ \gamma_\ell^{b_\ell}\xi^{(q)}_{*}\big((...,k_\ell+b_\ell+1,...),Y\big) \otimes \big(c_\ell^{\otimes q}(A_\ell^{\otimes q})^{b_\ell}\big)^{\mathsf{T}}$	$\prod_{\ell=1}^{L} N_\ell^{\otimes q}$	$\xi^{(1)}_{123}(k_1,k_2,k_3) \to \xi^{(1)}_{123}(k_1,k_2,k_3+1)$ 
"Backward" $k_\ell \to k_\ell-1$ $\gamma_{k_\ell} < 1$	$\xi^{(q)}_{*\ell}\big((...,k_\ell-1,...),Y\big)$ $= \gamma_\ell\big(I_{N_1}^{\otimes q} \otimes \cdots \otimes I_{N_{\ell-1}}^{\otimes q} \otimes A_\ell^{\otimes q}\big)^{\mathsf{T}} \xi^{(q)}_{*\ell}\big((...,k_\ell,...),Y\big)$ $+ \gamma_\ell^{a_\ell}\xi^{(q)}_{*}\big((...,k_\ell+a_\ell-1,...),Y\big) \otimes \big(c_\ell^{\otimes q}(A_\ell^{\otimes q})^{a_\ell}\big)^{\mathsf{T}}$ $- \gamma_\ell^{b_\ell+1}\xi^{(q)}_{*}\big((...,k_\ell+b_\ell,...),Y\big) \otimes \big(c_\ell^{\otimes q}(A_\ell^{\otimes q})^{b_\ell+1}\big)^{\mathsf{T}}$	$\prod_{\ell=1}^{L} N_\ell^{\otimes q}$	$\xi^{(1)}_{123}(k_1,k_2,k_3-1) \leftarrow \xi^{(1)}_{123}(k_1,k_2,k_3)$ 

Recursions with windows generated by a window ALSSM with parameters $\check{A}_\ell$ and $\check{c}_\ell$ ($*$ substitutes sequence of already incorporated dimensions) for
$\check{\xi}^{(q)}_{*\ell}(\boldsymbol{k}, Y) = \sum_{i_\ell=k_\ell+a_\ell}^{k_\ell+b_\ell} \check{\xi}^{(q)}_{*}\big((k_1,...,i_\ell,...,k_L),Y\big) \otimes \big((\check{c}_\ell \otimes c_\ell^{\otimes q})(\check{A}_\ell \otimes A_\ell^{\otimes q})^{i_\ell-k_\ell}\big)^{\mathsf{T}}$
over ℓ-th input dimension with index k_ℓ and ALSSM parameters $A_\ell \in \mathbb{R}^{N_\ell \times N_\ell}, c_\ell \in \mathbb{R}^{1\times N_\ell}, \check{A}_\ell \in \mathbb{R}^{\check{N}_\ell \times \check{N}_\ell}, \check{c}_\ell \in \mathbb{R}^{1\times \check{N}_\ell}, N_\ell, \check{N}_\ell \in \mathbb{N}$.

Direction	Recursion	Complexity $\mathcal{O}(\cdot)$	Example ($L=3$)
"Forward" $k_\ell \to k_\ell+1$ $\gamma_{k_\ell} > 1$	$\check{\xi}^{(q)}_{*\ell}\big((...,k_\ell+1,...),Y\big)$ $= \Big(I_{\check{N}_1} \otimes I_{N_1}^{\otimes q} \otimes \cdots \otimes I_{\check{N}_{\ell-1}} \otimes I_{N_{\ell-1}}^{\otimes q} \otimes (\check{A}_\ell \otimes A_\ell^{\otimes q})^{-1}\Big)^{\mathsf{T}} \check{\xi}^{(q)}_{*\ell}\big((...,k_\ell,...),Y\big)$ $- \check{\xi}^{(q)}_{*}\big((...,k_\ell+a_\ell,...),Y\big) \otimes \big((\check{c}_\ell \otimes c_\ell^{\otimes q})(\check{A}_\ell \otimes A_\ell^{\otimes q})^{a_\ell-1}\big)^{\mathsf{T}}$ $+ \check{\xi}^{(q)}_{*}\big((...,k_\ell+b_\ell+1,...),Y\big) \otimes \big((\check{c}_\ell \otimes c_\ell^{\otimes q})(\check{A}_\ell \otimes A_\ell^{\otimes q})^{b_\ell}\big)^{\mathsf{T}}$	$\prod_{\ell=1}^{L} \check{N}_\ell N_\ell^{\otimes q}$	$\xi^{(1)}_{123}(k_1,k_2,k_3) \to \xi^{(1)}_{123}(k_1,k_2,k_3+1)$ 
"Backward" $k_\ell \to k_\ell-1$ $\gamma_{k_\ell} < 1$	$\check{\xi}^{(q)}_{*\ell}\big((...,k_\ell-1,...),Y\big)$ $= \Big(I_{\check{N}_1} \otimes I_{N_1}^{\otimes q} \otimes \cdots \otimes I_{\check{N}_{\ell-1}} \otimes I_{N_{\ell-1}}^{\otimes q} \otimes \check{A}_\ell \otimes A_\ell^{\otimes q}\Big)^{\mathsf{T}} \check{\xi}^{(q)}_{*\ell}\big((...,k_\ell,...),Y\big)$ $+ \check{\xi}^{(q)}_{*}\big((...,k_\ell+a_\ell-1,...),Y\big) \otimes \big((\check{c}_\ell \otimes c_\ell^{\otimes q})(\check{A}_\ell \otimes A_\ell^{\otimes q})^{a_\ell}\big)^{\mathsf{T}}$ $- \check{\xi}^{(q)}_{*}\big((...,k_\ell+b_\ell,...),Y\big) \otimes \big((\check{c}_\ell \otimes c_\ell^{\otimes q})(\check{A}_\ell \otimes A_\ell^{\otimes q})^{b_\ell+1}\big)^{\mathsf{T}}$	$\prod_{\ell=1}^{L} \check{N}_\ell N_\ell^{\otimes q}$	$\xi^{(1)}_{123}(k_1,k_2,k_3-1) \leftarrow \xi^{(1)}_{123}(k_1,k_2,k_3)$ 

Thus, the second option is of particular interest for causal online and low-memory implementations. Both options are sumarized in Table IV.

C. Multi-Dimensional Least-Squares with ALSSMs

We consider an L_2 cost in analogy to (30) with L-dimensional signal Y, sample indices $\boldsymbol{i} = (i_1,\dots,i_L)$ and $\boldsymbol{k} = (k_1,\dots,k_L)$, finite boundaries $\boldsymbol{a},\boldsymbol{b}$, with $\boldsymbol{a} = (a_1,\dots,a_L)$ and for $\boldsymbol{b}$ analogously, and per-dimension decay factors $\gamma_1,\dots,\gamma_L \in \mathbb{R}_+$, and joint ALSSM initial state x. The cost (30) then generalizes to L dimensions with

$$J^{\boldsymbol{b}}_{\boldsymbol{a}}(\boldsymbol{k},x) = \sum_{i_1=k_1+a_1}^{k_1+b_1} \cdots \sum_{i_L=k_L+a_L}^{k_L+b_L} \gamma_1^{i_1-k_1}\cdots\gamma_L^{i_L-k_L}\big[Y_{\boldsymbol{i}} - s_{\boldsymbol{i}-\boldsymbol{k}}(x)\big]^2 \quad (56)$$
$$= \|Y_{\bullet+\boldsymbol{k}}\|_w^2 - 2\langle Y_{\bullet+\boldsymbol{k}}, \hat{Y}\rangle_w + \|\hat{Y}\|_w^2$$
$$= \xi^{(0)}_{12\cdots L}(\boldsymbol{k},Y) - 2x^{\mathsf{T}}\xi^{(1)}_{12\cdots L}(\boldsymbol{k},Y) + (R^{\mathsf{T}}x^{\otimes 2})^{\mathsf{T}}\xi^{(2)}_{12\cdots L}(\boldsymbol{k},\mathbb{1}). \quad (57)$$

Minimizing (57) over x provides a local state estimate at L-dimensional index $\boldsymbol{k}$, which is

$$\hat{x}_{\boldsymbol{k}} = \operatorname*{argmin}_x J^b_a(\boldsymbol{k},x) = W_{\boldsymbol{k}}^{-1}\,\xi^{(1)}_{12\cdots L}(\boldsymbol{k},Y)\ , \quad (58)$$

with

$$\operatorname{vec}(W_{\boldsymbol{k}}) = R\,\xi^{(2)}_{12\cdots L}(\boldsymbol{k},\mathbb{1}) \quad (59)$$

with $\operatorname{vec}(W_{\boldsymbol{k}})$ as in (33).

A practical example of 2D signal approximation and detection by least squares is provided in Fig. 8 and later in Section VII-A.

D. Advanced ALSSM Windows

Upto this point, the quadratic costs such as in (56) were limited to exponentially decaying windows. However, it is possible to introduce more sophisticated windows without breaking the recursive computation rules by using an additional ALSSM [6], [7], denoted as the window ALSSM.

TABLE IV
COMPARISON OF L-DIMENSIONAL CONVOLUTIONS IN SAMPLE SPACE AND CONVOLUTIONS IN ALSSM VECTOR SPACE.

L-dimensional convolution at index $\boldsymbol{k} = (k_1, k_2, ..., k_L)$ of observation $Y \in \mathbb{R}^{K_1 \times K_2 \times \cdots \times K_L}$ and filter $H \in \mathbb{R}^{M_1 \times M_2 \times \cdots \times M_L}$, $M_\ell = b_\ell - a_\ell + 1$ in sample space (first row) and with an L-dimensional ALSSM of dimensions $1, 2, \ldots, L$ in the ALSSM subspace (second row). $N_1 \ldots N_L$ indicate the ALSSM system order per dimension.

Type[1]	Convolution	Computational Complexity $\mathcal{O}(\cdot)$	
		$\xi(\cdot)$	Convolution
Inseparable	$(Y * H)_{\boldsymbol{k}} = \sum_{j_L=a_L}^{b_L} \cdots \sum_{j_1=a_1}^{b_1} Y_{(\boldsymbol{k}-\boldsymbol{j})} H_{\boldsymbol{j}}$		$\prod_{\ell=1}^{L} K_\ell M_\ell$
	$(Y * \hat{H})_{\boldsymbol{k}} = x(H)^\mathsf{T} \xi^{(1)}_{12\cdots L}(\boldsymbol{k}, Y_{-\bullet}; \boldsymbol{\theta})$ $= x(H)^\mathsf{T} \xi^{(1)}_{12\cdots L}(\boldsymbol{k}, Y; \boldsymbol{\theta}^{-1})$	$\prod_{\ell=1}^{L} K_\ell N_\ell$	$\prod_{\ell=1}^{L} K_\ell N_\ell$

L-dimensional convolution at index $\boldsymbol{k} = (k_1, k_2, ..., k_L)$ of observation $Y \in \mathbb{R}^{K_1 \times K_2 \times \cdots \times K_L}$ and separable filter $H = h^{(1)} * \cdots * h^{(L)}$ with $h^{(\ell)} \in \mathbb{R}^{M_\ell}$, $M_\ell = b_\ell - a_\ell + 1$ in sample space (first row) and with an L-dimensional ALSSM of dimensions $1, 2, \ldots, L$ in the ALSSM subspace (second row). $N_1 \ldots N_L$ indicate the ALSSM system order per dimension, θ_ℓ the ℓ-th ALSSMs parameters.

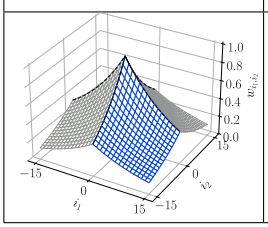
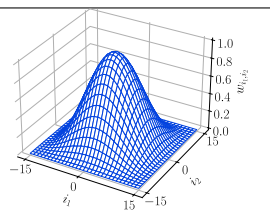
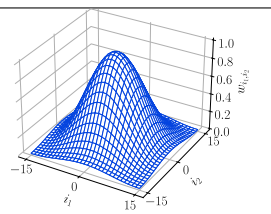
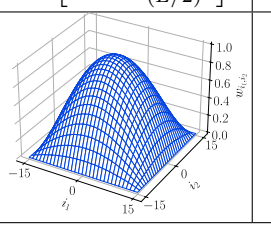
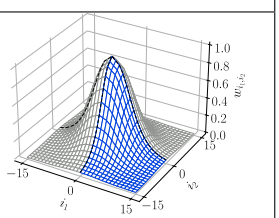
Type	Convolution	Computational Complexity[2] $\mathcal{O}(\cdot)$	
		$\xi(\cdot)$	Convolution
Separable	$\left(Y * h^{(1)} * \cdots * h^{(L)}\right)_{\boldsymbol{k}} = \sum_{j_L=a_L}^{b_L} \cdots \sum_{j_1=a_1}^{b_1} Y_{(\boldsymbol{k}-\boldsymbol{j})} h^{(1)}_{j_1} \cdots h^{(L)}_{j_L}$		$\prod_{\ell=1}^{L} K_\ell \left(\sum_{\ell=1}^{L} M_\ell\right)$
	$\left(Y * \hat{h}^{(1)} * \cdots * \hat{h}^{(L)}\right)_{\boldsymbol{k}}$ $= x(h^{(L)})^\mathsf{T} \xi^{(1)}_L\left(\boldsymbol{k}, x(h^{(L-1)})^\mathsf{T} \xi^{(1)}_{L-1}(\boldsymbol{k}, \cdots; \theta_{L-1}); \theta_L\right)$ with $\xi^{(1)}_1(\boldsymbol{k}, Y_{-\bullet}; \theta_1)$ $= x(h^{(L)})^\mathsf{T} \xi^{(1)}_L\left(\boldsymbol{k}, x(h^{(L-1)})^\mathsf{T} \xi^{(1)}_{L-1}(\boldsymbol{k}, \cdots; \theta^{-1}_{L-1}); \theta^{-1}_L\right)$ with $\xi^{(1)}_1(\boldsymbol{k}, Y; \theta^{-1}_1)$	$\prod_{\ell=1}^{L} (K_\ell) \left(\sum_{\ell=1}^{L} N_\ell\right)$ or $\prod_{\ell=1}^{L} (K_\ell) \left(\sum_{\ell=1}^{L} {N_\ell}^2\right)$	$\prod_{\ell=1}^{L} K_\ell \left(\sum_{\ell=1}^{L} N_\ell\right)$

[1] A separable L-D filter can be written as a product of L 1-D filters, and the L-D convolution is then equivalent to L 1-D convolutions.

[2] The complexity of $\xi(\cdot)$ scales linearly with N_ℓ for a (block) diagonal matrix A_ℓ, and scales with ${N_\ell}^2$ for a general matrix A_ℓ.

TABLE V
TWO-DIMENSIONAL WINDOWS WITH ALSSMS

Two-dimensional windows generated by ALSSMs with $w_{i_1,i_2} = w_{i_1} \cdot w_{i_2} = \breve{c}\breve{A}^{i_1}\breve{x} \cdot \breve{c}\breve{A}^{i_2}\breve{x} = \breve{c}^{\otimes 2}(\breve{A}^{i_1-k_1} \otimes \breve{A}^{i_2-k_2})\breve{x}^{\otimes 2}$. The windows are of length $\breve{M} \in \mathbb{N}$ in both dimensions.

Exponential Window (or rectangular for $\gamma = 1$)	Hann Window	Hamming Window	Welch Window	Gaussian Window (approximated)
$w_{i_1} = \gamma^{i_1}$ $w_{i_2} = \gamma^{i_2}$	$w_{i_1} = 0.5 + 0.5\cos(\Omega i_1)$ $w_{i_2} = 0.5 + 0.5\cos(\Omega i_2)$ $\Omega = 2\pi/\breve{M}$	$w_{i_1} = 0.54 + 0.46\cos(\Omega i_1)$ $w_{i_2} = 0.54 + 0.46\cos(\Omega i_2)$ $\Omega = 2\pi/\breve{M}$	$w_{i_1} = 1 - \left(\frac{i_1}{\breve{M}/2}\right)^2$ $w_{i_2} = 1 - \left(\frac{i_2}{\breve{M}/2}\right)^2$	$w_{i_1} = e^{-\frac{1}{2}\left(\frac{i_1}{\sigma}\right)^2}$ $w_{i_2} = e^{-\frac{1}{2}\left(\frac{i_2}{\sigma}\right)^2}$ e.g. for $\sigma = 5$: $\rho = 0.6,\ \breve{x} = \begin{bmatrix}1.0 & 0.42 & 0.18\end{bmatrix}^\mathsf{T}$
$\breve{A} = \begin{bmatrix}\gamma\end{bmatrix}$ $\breve{c} = [1]$ $\breve{x} = [1]$	$\breve{A} = \begin{bmatrix}1 & 0 & 0\\ 0 & \cos(\Omega) & -\sin(\Omega)\\ 0 & -\sin(\Omega) & \cos(\Omega)\end{bmatrix}$ $\breve{c} = \begin{bmatrix}1 & 1 & 0\end{bmatrix}$ $\breve{x} = \begin{bmatrix}0.5 & 0.5 & 0\end{bmatrix}^\mathsf{T}$	$\breve{x} = \begin{bmatrix}0.54 & 0.46 & 0\end{bmatrix}^\mathsf{T}$	$\breve{A} = \begin{bmatrix}1 & 1 & 1\\ 0 & 1 & 2\\ 0 & 0 & 1\end{bmatrix}$ $\breve{c} = \begin{bmatrix}1 & 0 & 0\end{bmatrix}$ $\breve{x} = \begin{bmatrix}1 & 0 & \frac{-1}{(\breve{L}/2)^2}\end{bmatrix}^\mathsf{T}$	$\breve{A} = \rho\begin{bmatrix}1 & 1 & 1\\ 0 & 1 & 2\\ 0 & 0 & 1\end{bmatrix}$ $\breve{c} = \begin{bmatrix}1 & 0 & 0\end{bmatrix}$ $\breve{x} = \begin{bmatrix}x_1 & x_2 & x_3\end{bmatrix}^\mathsf{T}$
				

In the following, we generalize the one-dimensional window ALSSMs from [6], [7] to the multi-dimension data.

Let the window weights in (40) be provided by the ALSSM

$$\begin{aligned} w_{\boldsymbol{j}} &= \breve{s}_{\boldsymbol{j}}(\breve{x}_1, \dots, \breve{x}_L) \\ &= \left(\breve{c}_1 \breve{A}_1^{j_1} \breve{x}_1 \otimes \cdots \otimes \breve{c}_L \breve{A}_L^{i_L} \breve{x}_L\right) , \end{aligned} \tag{60}$$

i.e., the output of an L-dimensional window ALSSM consisting of L ALSSMs, each with its own parameters $\breve{A}_\ell$, $\breve{c}_\ell$ and initial state $\breve{x}_\ell$. Then, the windowed squared error at window location $\boldsymbol{k}$ when using model (34) with per-dimension separated states is

$$\begin{aligned} \breve{J}_{\boldsymbol{a}}^{\boldsymbol{b}}(\boldsymbol{k}, x) = & \sum_{i_1=k_1+a_1}^{k_1+b_1} \cdots \sum_{i_L=k_L+a_L}^{k_L+b_L} \breve{s}_{\boldsymbol{i}-\boldsymbol{k}}(\breve{x}) \left[Y_{\boldsymbol{i}} - s_{\boldsymbol{i}-\boldsymbol{k}}(x)\right]^2 \\ = & (\breve{x}_1 \otimes \cdots \otimes \breve{x}_L)^{\mathsf{T}} \breve{\xi}^{(0)}_{12\cdots L}(\boldsymbol{k}, Y) \\ & -2(\breve{x}_1 \otimes x_1 \otimes \cdots \otimes \breve{x}_L \otimes x_L)^{\mathsf{T}} \breve{\xi}^{(1)}_{12\cdots L}(\boldsymbol{k}, Y) \\ & +(\breve{x}_1 \otimes x_1^{\otimes 2} \otimes \cdots \otimes \breve{x}_L \otimes x_L^{\otimes 2})^{\mathsf{T}} \breve{\xi}^{(2)}_{12\cdots L}(\boldsymbol{k}, \mathbb{1}) \end{aligned} \tag{61}$$

with

$$\begin{aligned} \breve{\xi}^{(q)}_{*\ell}(\boldsymbol{k}, Y) \triangleq & \sum_{i_\ell=k_\ell+a_\ell}^{k_\ell+b_\ell} \breve{\xi}^{(q)}_{*}\big((k_1, ..., i_\ell, ..., k_L), Y\big) \\ & \otimes \big((\breve{c}_\ell \otimes c_\ell^{\otimes q})(\breve{A}_\ell \otimes A_\ell^{\otimes q})^{i_\ell - k_\ell}\big)^{\mathsf{T}} \end{aligned} \tag{62}$$

initialized as in (48) and with its own recursion, see Table III.

Alternatively, using the more general signal model (35) with joint state vector x, cost computation (63) modifies to

$$\begin{aligned} & \breve{J}_{\boldsymbol{a}}^{\boldsymbol{b}}(\boldsymbol{k}, x) \\ & = (\breve{x}_1 \otimes \cdots \otimes \breve{x}_L)^{\mathsf{T}} \breve{\xi}^{(0)}_{12\cdots L}(\boldsymbol{k}, Y) \\ & \quad -2x^{\mathsf{T}}(\breve{x}_1 \otimes I_{N_1} \otimes \cdots \otimes \breve{x}_L \otimes I_{N_L})^{\mathsf{T}} \breve{\xi}^{(1)}_{12\cdots L}(\boldsymbol{k}, Y) \\ & \quad +(R^{\mathsf{T}} x^{\otimes 2})^{\mathsf{T}} (\breve{x}_1 \otimes I_{N_1}^{\otimes 2} \otimes \cdots \otimes \breve{x}_L \otimes I_{N_L}^{\otimes 2})^{\mathsf{T}} \breve{\xi}^{(2)}_{12\cdots L}(\boldsymbol{k}, \mathbb{1}) \end{aligned} \tag{63}$$

and gets minimized by

$$\begin{aligned} \hat{x}_{\boldsymbol{k}} &= \operatorname*{argmin}_x \breve{J}_{\boldsymbol{a}}^{\boldsymbol{b}}(\boldsymbol{k}, x) \\ &= \breve{W}_{\boldsymbol{k}}^{-1} (\breve{x}_1 \otimes I_{N_1} \otimes \cdots \otimes \breve{x}_L \otimes I_{N_L})^{\mathsf{T}} \breve{\xi}^{(1)}_{12\cdots L}(\boldsymbol{k}, Y) , \end{aligned} \tag{64}$$

using, in analogy to (33),

$$\operatorname{vec}(\breve{W}_{\boldsymbol{k}}) = R\, (\breve{x}_1 \otimes I_{N_1}^{\otimes 2} \otimes \cdots \otimes \breve{x}_L \otimes I_{N_L}^{\otimes 2})^{\mathsf{T}} \breve{\xi}^{(2)}_{12\cdots L}(\boldsymbol{k}, \mathbb{1}) . \tag{65}$$

A proof for (63) is provided in the supplementary material in Section A.III.

Table V presents a selection of practical two-dimensional windows. It is important to note that when employing advanced windows, all derived recursion formulas are still applicable. However, the use of advanced windows increases the overall ALSSM system order, which in turn leads to a higher computational load.

VI. Shared and Parallelized Computations

The linearity of the ξ terms allows calculations to be divided into multiple fractions. This has several applications, such as reusing intermediate results to obtain results for multiple window sizes and resolutions conveniently, e.g. in multi-resolution analysis, or to perform interleaved computations enabling parallelized processing and taking full advantage of dedicated hardware platforms such as Graphics Processing Units (GPUs) and the like.

A. Multi-Resolution Least-Squares in ALSSM Subspace

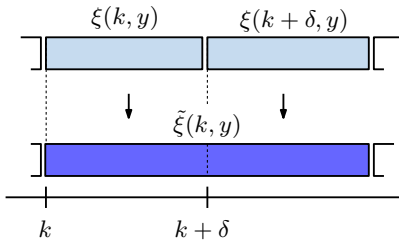


Fig. 5. Reuse of ξ terms for efficient multi-resolution processing as applied in (66). The figure illustrates the merging of two ξ terms of smaller window sizes (light blue) to a new sum term $\tilde{\xi}$ of double the window size (dark blue).

For many applications we need the inner product for different window lengths, such as for filter banks, wavelet transforms, and other types of multi-resolution analysis. Interestingly, the various sum terms ξ presented in this paper can be merged to form sum terms for larger windows without needing to recompute them. We apply this technique in multi-resolution analysis by summing shifted versions of ξ terms, resulting in a multiple of the window size.

In (16) we provided $\xi(k, y)$ for a 1-dimensional inner product as in (14) using an ALSSM signal approximation in an interval of fixed length. It is straightforward to show that the sum term spanning two intervals of equal length, $\{a, \dots, b\}$ and $\{a+\delta, \dots, b+\delta\}$ with $\delta \in \mathbb{Z}$ is

$$\begin{aligned} \tilde{\xi}(k, y) &= \sum_{j=a}^{b} \gamma^j y_{j+k} (cA^j)^{\mathsf{T}} + \sum_{j=a+\delta}^{b+\delta} \gamma^j y_{j+k} (cA^j)^{\mathsf{T}} \\ &= \xi(k, y) + (\gamma A^{\mathsf{T}})^\delta \; \xi(k+\delta, y) . \end{aligned} \tag{66}$$

This allows to compute an ALSSM fitted over a longer window (i.e. of lower resolution), thereby re-using results from ALSSMs that were fitted over two shorter windows (i.e. of higher resolution). The mechanism of (66) is also illustrated in Fig. 5.

Analogously, the sum term for L-dimensional observations (47) spanning two (or more) intervals of equal size, where the second interval is shifted by $\boldsymbol{\delta} = (\delta_1, \dots, \delta_L)$ samples in each dimension, is

$$\tilde{\xi}^{(q)}_{*\ell}(\boldsymbol{k}, Y) = \xi^{(q)}_{*\ell}(\boldsymbol{k}, Y) + (D^{(q)}_{*\ell})^{\mathsf{T}} \xi^{(q)}_{*\ell}(\boldsymbol{k}+\boldsymbol{\delta}, Y) , \tag{67}$$

with recursively defined but constant shift matrix

$$D^{(q)}_{*\ell} = D^{(q)}_{*} \otimes (\gamma_\ell A_\ell^{\otimes q})^{\delta_\ell} , \tag{68}$$

starting with $D^{(q)} = 1$. The shift matrix $D^{(q)}_{*\ell}$ shifts the sum term $\xi^{(q)}_{*\ell}$ in (67) in all dimensions at once. While the proof for (66) is straightforward, the proof for (67) is rather tedious and provided in the supplementary material in Section A.IV.

The method presented in (67) prevents the recalculation of the sum term ξ for each window size in multi-resolution

applications and instead reuses the results of the smallest window to calculate the wider ones. This is in contrast to the standard implementation of ALSSMs, where recalculation for each scaling is required.

Figure 6 shows an immediate application of this multi-scale approach, where the output of polynomial filters of different window lengths is illustrated. For the calculations of these filter outputs, sum terms ξ of smaller window sizes were reused to generate filter output of larger window sizes according to (66). This application is a multi-resolution version of the well-known Savitzky-Golay filter.

A second application of the multi-resolution approach is provided in Section VII-B when shapes of different length need to be detected in a bioelectrical signal. A third example of multi-resolution analysis using a stationary wavelet transform with a Haar wavelet is provided in the supplementary material in Section A.I.

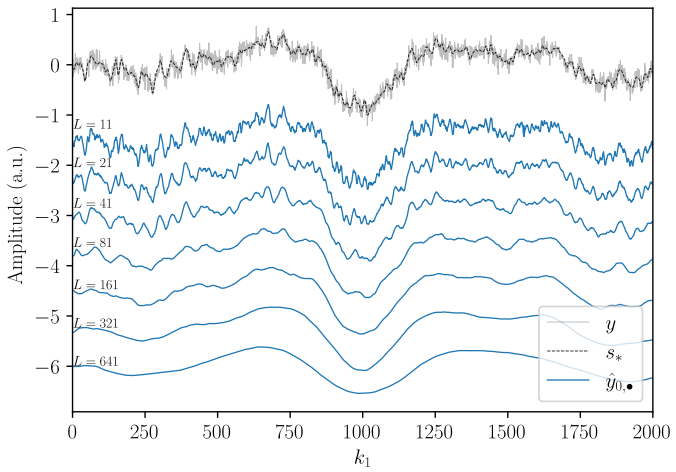


Fig. 6. Multi-resolution Savitzky-Golay filter using local polynomials of degree three and with seven different window sizes $L \in \{11, 21, 41, 81, 161, 321, 641\}$. The observed signal y (solid grey line) is a noisy version of s_* (dashed black line), to which Gaussian white noise is added. $\hat{y}_{0,\bullet}$ (solid blue lines) are the filter output of the ALSSM model $\hat{y}_{0,k} = s_0(x_k)$ according to (9) for each window length, efficiently computed as proposed in Section VI-A. The segment boundaries for the finite windows are set to $a = \lceil -L/2 \rceil$ and $b = \lfloor L/2 \rfloor$ and the window decay factor to $\gamma^{-1} = 0.999$. Data source of s_*: [25].

B. Interleaved Computation (Parallelization)

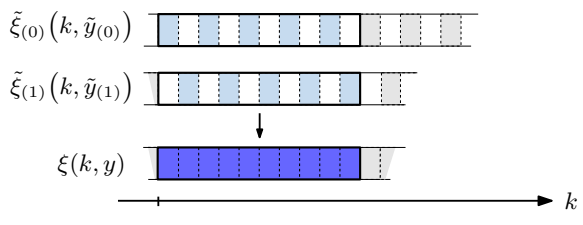


Fig. 7. Use of sample interleaving to parallelize ξ-term computations as proposed in (69) and (70). The two $\tilde{\xi}$ terms are only calculated on every second sample (light blue), while ξ is the aggregation of both (dark blue).

To enable parallelized computation, we subdivide the samples into S subsets applying an interleaving pattern, similar to a polyphase decomposition in linear filtering [26], and process the sets independently. The final ξ-term is then retrieved from intermediate results of the S subsets, which are computable in parallel.

Let $\tilde{y}_{(s)}$ be one out of S interleaved signals with only the samples y_{nS+s} for any valid $n \in \mathbb{Z}$ and $s \in \{0, \ldots, S-1\}$. For example, for $S = 2$ we have the two interleaved signals $\tilde{y}_{(0)} = [\ldots, y_0, y_2, y_4, \ldots]$ and $\tilde{y}_{(1)} = [\ldots, y_1, y_3, y_5, \ldots]$. Then, using the integer modulo operator $\%$ and integer division operator $/\!/$, the ξ-term composes as

$$\xi(k, y) = \sum_{s=0}^{S-1} (\gamma A^{\mathsf{T}})^{(k+s)\%S} \tilde{\xi}_{(s)}\big(k, \tilde{y}_{(s)}\big) \tag{69}$$

with

$$\tilde{\xi}_{(s)}(k, \tilde{y}) \triangleq \xi\big((k+s)/\!/S, \tilde{y}; \theta_s\big), \tag{70}$$

using ξ with a parameter set, as introduced earlier in (29), $\theta_s = \{A^S, a/\!/S, b/\!/S, \gamma^S\}$, with the selection of a and b restricted to a multiple of S. The mechanism of (69) is also illustrated in Fig. 7. Note that the ξ terms in (70) working on interleaved signals again follow the proven recursion rules and are fully independent from each other, i.e. are fully parallelized.

VII. Practical Examples

In this section, we provide several practical examples to illustrate and appreciate the use of ALSSMs in multi-dimensional and/or multi-resolution settings. Table VI shows a summary comparison of the computational complexity and of the actual effort in terms of scalar multiplications (MAC) for the presented ALSSM methods and corresponding sample space methods (using standard matched filters).

A. Character Detection as a 2-D Least Squares Problem

Images are probably the most common two-dimensional signals. In this toy example, we demonstrate pattern matching by detecting characters in a grayscale image, as shown in Fig. 8. To detect characters, we first transform the full image into an ALSSM subspace spanned by a 2D polynomial model of degree 3 in both dimensions, parametrizing the ALSSM from (35) as

$$s_{\boldsymbol{j}}(x) = \big((c_1 A_1^{j_1}) \otimes (c_2 A_2^{j_2})\big) x \tag{71}$$

with $\boldsymbol{j} = (j_1, j_2)$, and with A_1 and A_2 as in (10), c_1 and c_2 as in (11), and $x \in \mathbb{R}^{16}$.

Then, we proceed as follows:

1. Define the size of a rectangular window, appropriate to detect the characters and use this window size in (56), e.g., set $a_1 = a_2 = -35$ and $b_1 = b_2 = 35$. This is the size of the dashed box in Fig. 8.
2. Compute the sum terms $\xi_{12}^{(0)}(\boldsymbol{k}, Y)$, $\xi_{12}^{(1)}(\boldsymbol{k}, Y)$, and $\xi_{12}^{(2)}(\boldsymbol{k}, Y)$ applying the recursions provided in Table II with the image as the input Y and for every valid index $\boldsymbol{k} = (k_1, k_2)$. (decay factors γ_1, γ_2 are set to close to 1.0 to get a rectangular-like yet computationally stable window.)

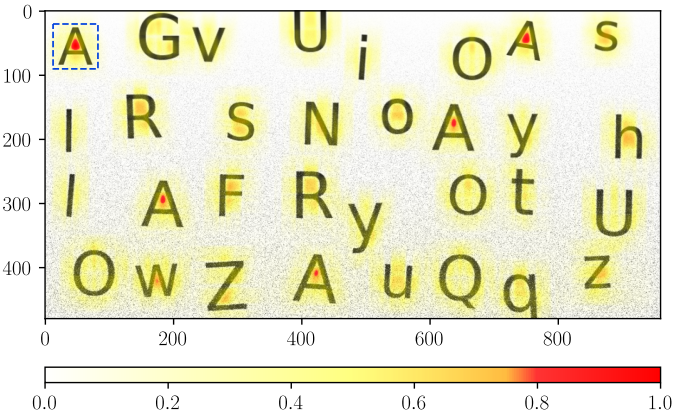


Fig. 8. Pattern detection in a two-dimensional image using ALSSM subspace projection and a cost ratio in a sliding window manner as discussed in Section VII-A. The raw image is in grayscale and shown in the background and shows additive Gaussian noise of increasing intensity from top to bottom. For the pattern detection, the cost ratio (72) is computed for every coordinate using the state vector of the reference window (marked by the dashed box) in the ALSSM subspace $\hat{x}_{\mathrm{ref}}$. This cost ratio is a measure of likelihood for detecting the reference pattern and is displayed as a yellowish overlay on top of the raw image, peaking in red for very high likelihood.

3. Compute all state vectors $\hat{x}_{\boldsymbol{k}}$ applying (58) for all pixels based on the precomputed ξ terms.
4. Select $\hat{x}_{\mathrm{ref}}$ as the $\hat{x}_{\boldsymbol{k}}$ corresponding to the reference character position (marked by the dashed box in Fig. 8).
5. Compute for every pixel index $\boldsymbol{k}$ the cost ratio

$$\mathrm{CR}_{\boldsymbol{k}} = \frac{J_a^b(\boldsymbol{k}, \hat{x}_{\boldsymbol{k}})}{J_a^b(\boldsymbol{k}, \hat{x}_{\mathrm{ref}})} \tag{72}$$

as a likelihood measure for a character detection. Characters are most likely located where the CR value peaks.

This likelihood measure $\mathrm{CR}_{\boldsymbol{k}}$ is displayed in Fig. 8 (as a yellowish overlay, peaking in red for very maximum likelihood).

A comparison of the computational effort of our approach and the standard matched filters can be found in Table VI.

TABLE VI
COMPARISON OF THE COMPUTATIONAL COMPLEXITY OF SAMPLE SPACE AND OUR ALSSM SUBSPACE METHOD ON PRACTICAL EXAMPLES, WITH ALSSMS OF ORDERS N_ℓ, WINDOW LENGTHS M_ℓ, AND n SHAPES.

Example	Sample Space		ALSSM Space	
	$\mathcal{O}(\cdot)$	MAC	$\mathcal{O}(\cdot)$	MAC
Character Detection (Sec. VII-A) $M_1 = M_2 = 35; N_1 = N_2 = 4$	$M_1 M_2$	1225	$N_1 N_2$	139 (11%)
ECG Segmentation (Sec. VII-B) $M_1 = \{40, 20, 20, 20\}$; $M_2 = 6$; $N_1 = 3$; $N_2 = 4$; $n = 4$	$M_1 M_2 n$	600	$N_1 N_2 n$	110 (18%)
Water Droplet Counting (Sec. VII-C) $M_1 = M_2 = \{40, 20, 14, 10\}$; $N_1 = N_2 = 6$; $n = 4$	$M_1 M_2 n$	2296	$N_1 N_2 n$	512 (22%)
3D Edge Detection (Sec. VII-D) $M_1 = M_2 = M_3 = 15$; $N_1 = N_2 = N_3 = 2$	$\prod_{\ell=1}^{3} M_\ell$	72	$\prod_{\ell=1}^{3} N_\ell$	44 (61%)

B. ECG Segmentation by Pattern Matching

For the second example, we consider an electrocardiography (ECG) recording, which was recorded with an electrode array consisting of 6 electrodes, applied from left to right on a subject's chest. Such a signal can be considered as a two-dimensional recording with the two dimensions position and time. In that signal, we are interested in efficiently detecting the reoccurrence of certain waveforms. The ECG signal provided includes 6 chest leads, denoted as V1 to V6, see Fig. 9. The waves of interest are *normal* and *abnormal* QRS complexes, as well as variations of T waves.

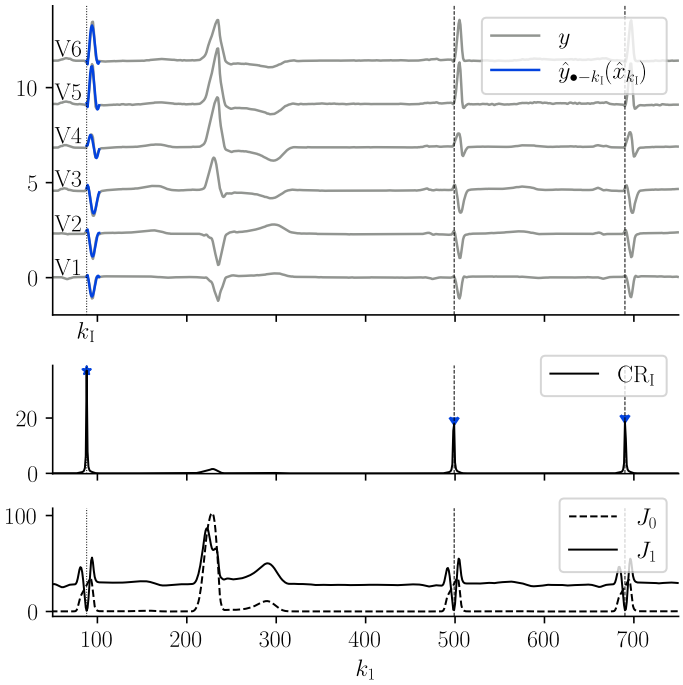


Fig. 9. Template matching on multi-channel signal. The upper plot shows the 6-lead ECG signal (grey lines), with the projection to the ALSSM subspace of the reference wave (blue line) at time index k_{I}. The center plot shows the CR (72), peaking at the reference template and at later pattern occurrences (around $k_1 = 500$ and $k_1 = 680$), but not at other wave forms of similar shapes. The lower plot shows the two cost remainders going into (72), $J_0 = J_a^b(k, \mathbb{0})$ and $J_1 = J_a^b(k, \hat{x}_{\mathrm{ref}})$. Data source: [27]. Figure 10 shows a zoomed out version of this plot with detection of multiple shapes of different lengths.

To model these shapes, we use a polynomial of degree 2 in the first dimension and a polynomial of degree 3 in the second dimension, and parametrize the ALSSM from (35) as

$$s_{\boldsymbol{j}}(x) = \big((c_1 A_1^{j_1}) \otimes (c_2 A_2^{j_2})\big) x \tag{73}$$

with $\boldsymbol{j} = (j_1, j_2)$, c_2 and A_2 as in Example VII-A, and with

$$c_1 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \quad A_1 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix},$$

and $x \in \mathbb{R}^{12}$.

To detect these types of waves efficiently, we proceed according to Steps 1 to 6 from the previous Example VII-A with the following modifications: since some waveforms are longer than others, we compute the ξ terms only for the shortest length waveform with the recursions provided in Table II. The ξ terms for the longer waveforms are then efficiently calculated with the multi-resolution computation (66). In addition, the

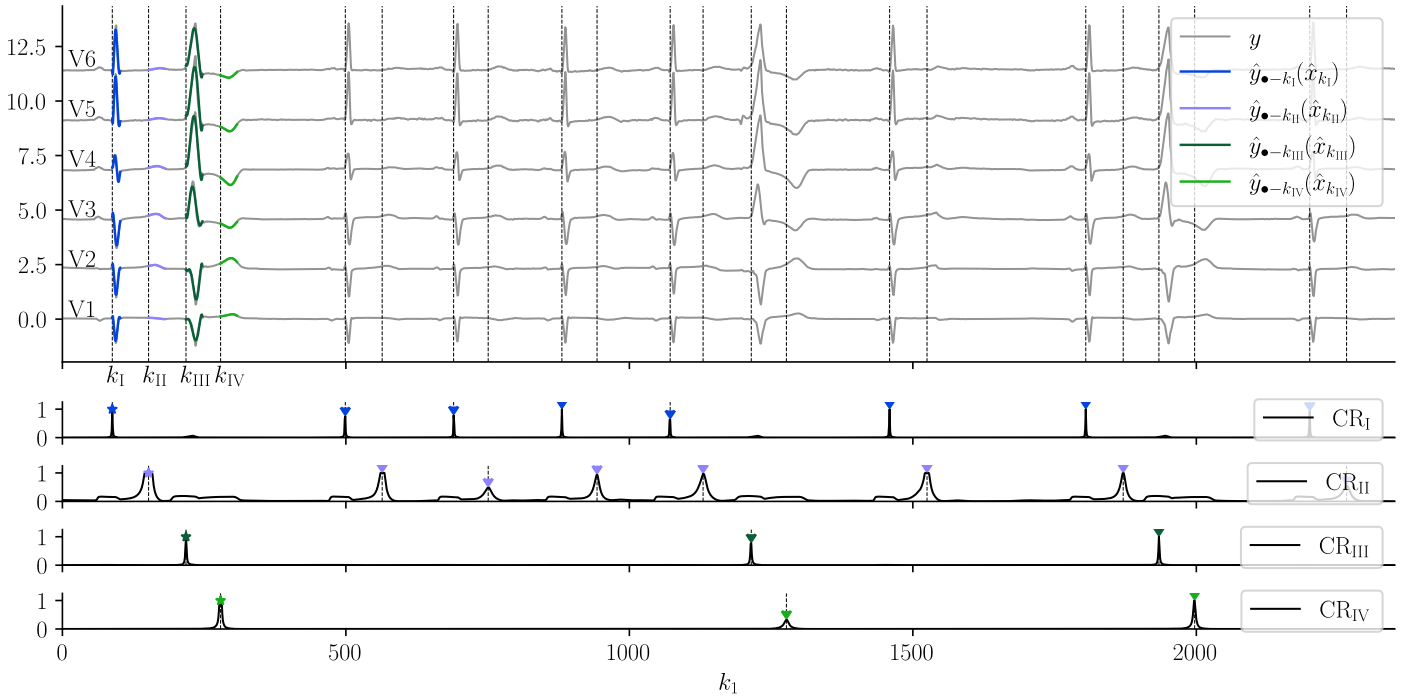


Fig. 10. Zoomed out version of Fig. 9 with detection of four different shapes, each of individual length. The upper plot shows the 6-lead ECG signal (grey lines), with the four reference shapes projection to the ALSSM subspace. The lower plots show the four per-shape CR values, each peaking only on its referenced shape and normalized for plotting reasons. Data source: [27].

cost ratio in step 5 changes to

$$\mathrm{CR}(\boldsymbol{k}) = \frac{J_a^b(\boldsymbol{k}, 0)}{J_a^b(\boldsymbol{k}, \hat{x}_{\mathrm{ref}})} \tag{74}$$

using a noise-only cost term in the numerator. This a non-linear approach closely related to likelihood based signal detection. In contrast to matched filters, this non-linear approach is robust against abrupt changes in input amplitudes, which often leads to false detection in linear filtering. This behaviour is observable in Fig. 9 for example, where detection of small amplitude signals such as the p waves (k_{II}) are not interfered with by high amplitude pulses such as the QRS waves (k_{I} and k_{III}).

Figures 9 and 10 show exemplary results for the detection of single or multiple waveforms respectively in an ECG signal.

This example shows that our method is a computationally attractive approach for shape detection and discrimination, especially in environments with limited computational resources.

C. Water Droplet Counting with Sinusoidal Fitting

The third example is from the field of image analysis, where we need to classify and count different types of water drops on a hydrophobic surface. A sample image is provided in Fig. 11. The droplets are either dome-shaped or toroidal.

For both shapes, we use a signal model consisting of a constant offset with two additive cosine waves of distinct frequencies ω and 2ω,

$$\begin{aligned} s_{\boldsymbol{j}}(x) &= x^{\mathsf{T}} \left(\begin{bmatrix} 1 \\ \cos(\omega j_1) \\ \cos(2\omega j_1) \end{bmatrix} \otimes \begin{bmatrix} 1 \\ \cos(\omega j_2) \\ \cos(2\omega j_2) \end{bmatrix} \right), \\ &= \big((c_1 A_1^{j_1}) \otimes (c_2 A_2^{j_2}) \big) x \end{aligned} \tag{75}$$

with $\boldsymbol{j} = (j_1, j_2)$ and

$$c_1 = c_2 = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

and with block diagonals

$$A_1 = A_2 = \mathrm{diag}\left(1, R(\omega), R(2\omega)\right)$$

where

$$R(\omega) = \begin{bmatrix} \cos(\omega) & -\sin(\omega) \\ \sin(\omega) & \cos(\omega) \end{bmatrix}.$$

To detect the droplets robustly, we also need an alternative model (the null-hypothesis) to adapt the inhomogenous gray scale in the image backgrounds,

$$\mathring{s}_{\boldsymbol{j}}(\mathring{x}) = \lambda \tag{76}$$

with $\lambda \in \mathbb{R}$, which is a simple constant offset model. The input signal Y is the two-dimensional image with Y_k indexing the pixel grayscale intensity value at coordinates $\boldsymbol{k} = (k_1, k_2)$.

Then, for both models s and $\mathring{s}$, we compute the necessary ξ terms according to steps 1 to 4 from Example VII-A. The cost ratio in step 5 then changes to

$$\mathrm{CR}(\boldsymbol{k}) = \frac{\mathring{J}_a^b(\boldsymbol{k}, \hat{\lambda}_{\boldsymbol{k}})}{J_a^b(\boldsymbol{k}, \hat{x}_{\boldsymbol{k}})} \tag{77}$$

using the respective model in the numerator and denominator. Finally, droplets are located at local peaks of the CR value.

Figure 11 shows an example of detected droplets and Fig. 12 an example of the toroidal shape fit applying the proposed model. In terms of computational effort, a comparison against four matched filters (one filter for each window size) in the sample space is shown in Table VI.

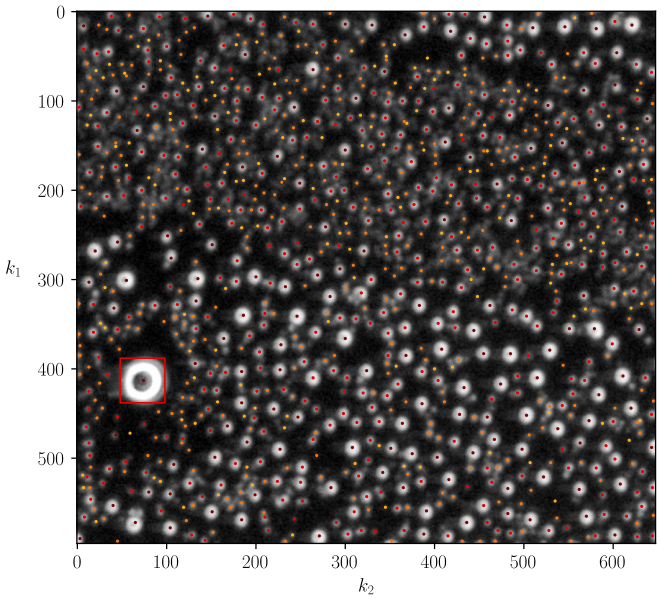


Fig. 11. Water droplets detected on a grayscale image applying a least-squares fit of a two-dimensional ALSSM model. Droplets were detected wherever the LCR (77) peaked locally. To detect and cluster droplets of various sizes, the model (75) was applied repetitively with window sizes of 40 samples (red rectangle), 20 samples (dark red dots), 14 samples (red dots), and 10 samples (orange dots).

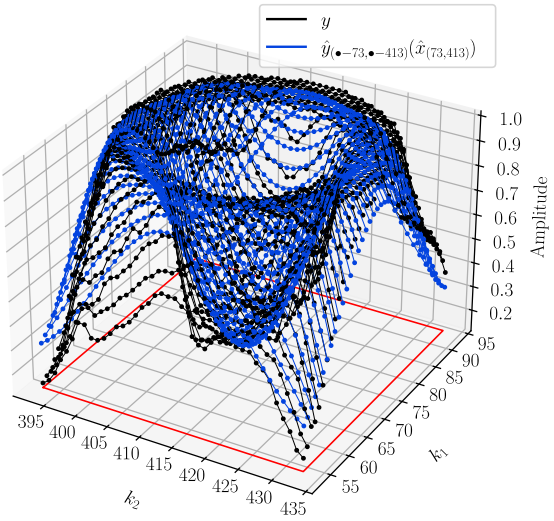


Fig. 12. Least-squares fit of the model (blue dots) proposed in (75) to a grayscale intensities of a water droplet image (black dots). This droplet was generated using window interval borders $a_1 = a_2 = -20$, $b_1 = b_2 = 20$, and cosine model frequency $\omega = 2\pi/40$.

D. Edge Detection Using Normal Vectors of 3D Planes

The fourth example implements an edge detection method in three-dimensional data sets such as from magnetic resonance imaging (MRI) or computed tomography (CT) scans. For that, three-dimensional planes are locally fit to the image voxel intensities; the ALSSM representation of such planes is

$$s_{\boldsymbol{j}}(x) = x^{\mathsf{T}} \left(\begin{bmatrix} 1 \\ j_1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ j_2 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ j_3 \end{bmatrix} \right), \tag{78}$$

with index tuple $\boldsymbol{j} = (j_1, j_2, j_3)$ and gradient

$$\operatorname{grad}(s_{\boldsymbol{j}}) \triangleq \frac{\partial s_{\boldsymbol{j}}(x)}{\partial \boldsymbol{j}} = \big[[x]_5, [x]_3, [x]_1\big]^{\mathsf{T}} \tag{79}$$

with $[x]_n$ here indicating the n-th element of state vector x. Finally, the Euclidean norm $\|\operatorname{grad}(s_{\boldsymbol{j}})\|$ is considered as a local measure of the edge intensity at index $\boldsymbol{j}$.

Figure 13 shows the implementation of a common gradient-based edge detection method, where a standard canny edge algorithm with non-maximum suppression, thinned edges, and double thresholding and hysteresis-edge-tracking is applied [28].

VIII. Conclusion

In this paper, we have generalized the application of ALSSMs to linear filters of multi-dimensional input signals and have derived recursive computation rules for those. This is of particular interest for applications with limited resources such as energy or memory, or for applications with real-time constraints. The method is also a linking element between very large data sets and machine learning algorithms, where data compression, feature detection, and selection is essential.

Acknowledgments

We thank Andrew Brown from FHNW for proofreading, Lydia Feller from FHNW for providing the water droplets image, and Hugo Aguettaz from ETH Zürich for the inspiring discussions about notation and other topics.

References

[1] H.-A. Loeliger, S. Neff, and C. Reller, "Self-synchronizing signal parsing with spiking feature-detection filters," in *2014 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2014, pp. 123–128.

[2] L. Bruderer, H.-A. Loeliger, and N. Zalmai, "Local statistical models from deterministic state space models, likelihood filtering, and local typicality," in *2014 IEEE International Symposium on Information Theory*. IEEE, 2014, pp. 1106–1110.

[3] C. Reller, *State-space methods in statistical signal processing: New ideas and applications*. ETH Zurich, 2013, vol. 23.

[4] N. Zalmai, R. A. Wildhaber, and H.-A. Loeliger, "Autonomous state space models for recursive signal estimation beyond least squares," in *2017 25th European Signal Processing Conference (EUSIPCO)*. IEEE, 2017, pp. 341–345.

[5] 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.

[6] R. A. Wildhaber, "Localized state space and polynomial filters with applications in electrocardiography," Ph.D. dissertation, ETH Zurich, 2019.

[7] N. Zalmai, "A state space world for detecting and estimating events and learning sparse signal decompositions," Ph.D. dissertation, ETH Zurich, 2017.

[8] E. Ren, "Using local state space model approximation for fundamental signal analysis tasks," Ph.D. dissertation, ETH Zurich, 2023.

[9] N. Zalmai, R. A. Wildhaber, D. Clausen, and H.-A. Loeliger, "Inferring depolarization of cells from 3d-electrode measurements using a bank of linear state space models," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 3331–3335.

[10] 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.

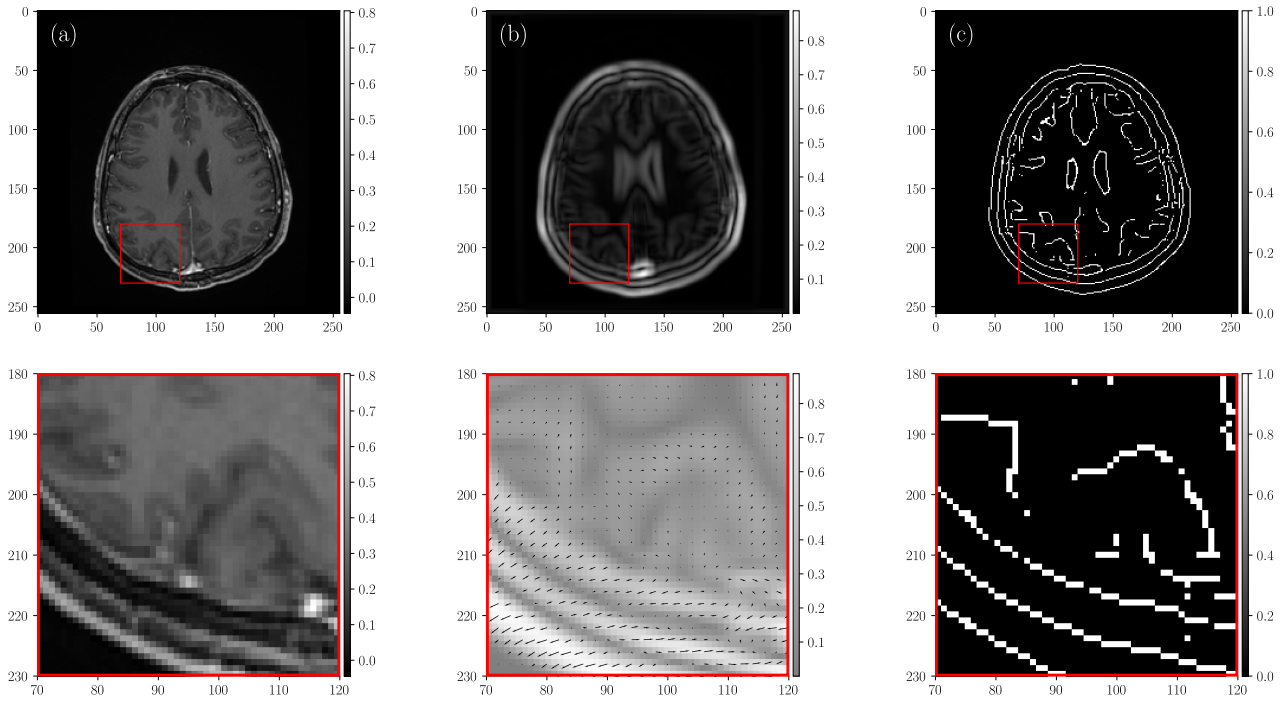


Fig. 13. 3-D edge detection on hippocampal MRI with a kernel (window) size of $11 \times 11 \times 11$ pixels. Subfigure (a) shows slice 200 of the original MRI image, Subfigure (b) the magnitude of the gradients resulting from (79), and Subfigure (c) shows the thinned edges postprocessed with the standard canny edge algorithm [28]. Note that the figure displays only one slice out of many while our detection algorithm uses all slices. The lower figures show zoomed versions of the upper figures, indicated by the red rectangle, revealing the gradient vectors (black arrows) in the center plot. Data source: [29]

[11] F. Waldmann, C. Baeriswyl, R. Andonie, and R. A. Wildhaber, "Onset detection of pulse-shaped bioelectrical signals using linear state space models," in *Current Directions in Biomedical Engineering*, vol. 8, no. 2. De Gruyter, 2022, pp. 101–104.

[12] 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.

[13] E. Ren, G. C. Ornelas, and H.-A. Loeliger, "Real-time interaural time delay estimation via onset detection," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 4555–4559.

[14] C. Baeriswyl, A. Bertrand, and R. A. Wildhaber, "Windowed state space filters for peak interference suppression in neural spike sorting," in *2022 30th European Signal Processing Conference (EUSIPCO)*. IEEE, 2022, pp. 1208–1212.

[15] B. Ma, "Smoothed-nuv priors for imaging and beyond," Ph.D. dissertation, ETH Zurich, 2021.

[16] P. Vaidyanathan, "Multirate systems and filter banks," *Prentice Hall Signal Processing Series*, 1993.

[17] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex fourier series," *Mathematics of computation*, vol. 19, no. 90, pp. 297–301, 1965.

[18] S. G. Mallat, "Multifrequency channel decompositions of images and wavelet models," *IEEE Transactions on Acoustics, speech, and signal processing*, vol. 37, no. 12, pp. 2091–2110, 1989.

[19] S. W. Smith *et al.*, "The scientist and engineer's guide to digital signal processing," 1997.

[20] R. D. Strum and D. E. Kirk, *First principles of discrete systems and digital signal processing*. Reading, Massachusetts: Addison-Wesley, 1988.

[21] R. A. Wildhaber, F. Waldmann, L. Fleischmann, and C. Baeriswyl, "lmlib - an open source model-based signal processing library," August 2024. [Online]. Available: http://lmlib.ch

[22] N. Zalmai, H. Malmberg, and H.-A. Loeliger, "Blind deconvolution of sparse but filtered pulses with linear state space models," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4194–4198.

[23] M. Vetterli, J. Kovačević, and V. K. Goyal, *Foundations of signal processing*. Cambridge University Press, 2014.

[24] J. R. Magnus and H. Neudecker, "The commutation matrix: some properties and applications," *The annals of statistics*, vol. 7, no. 2, pp. 381–394, 1979.

[25] I. Zyma, S. Tukaev, I. Seleznov, K. Kiyono, A. Popov, M. Chernykh, and O. Shpenkov, "Electroencephalograms during mental arithmetic task performance," *Data*, vol. 4, no. 1, 2019. [Online]. Available: https://www.mdpi.com/2306-5729/4/1/14

[26] K. K. Parhi, *VLSI digital signal processing systems: design and implementation*. John Wiley & Sons, 2007.

[27] V. Tihonenko, A. Khaustov, S. Ivanov, A. Rivin, and E. Yakushenko, "St petersburg incart 12-lead arrhythmia database," *PhysioBank PhysioToolkit and PhysioNet*, 2008.

[28] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[29] A. Shah, "Hippocampal contouring dataset for radiotherapy," *kaggle*, 2008. [Online]. Available: https://www.kaggle.com/datasets/aryashah2k/hippocampal-sparing-dataset/data