Windowed State-Space Filters for Signal Detection and Separation

Reto A. Wildhaber, Nour Zalmai, Marcel Jacomet, and Hans-Andrea Loeliger, *Fellow, IEEE*

***Abstract*—This paper introduces a toolbox for model-based detection, separation, and reconstruction of signals that is especially suited for biomedical signals, such as electrocardiograms (ECGs) or electromyograms (EMGs). The modeling is based on autonomous linear state space models (LSSMs), which are localized with flexible windows. The models are fit to observations by minimizing the squared error while the use of LSSMs leads to efficient recursive error computations and minimizations. Multisection windows enable complex models, and per-sample weights enable multistage processing or adaptive smoothing. This paper is motivated by, and intended for, practical applications, for which several examples and tabulated cost computations are given.**

***Index Terms*—Linear state space models, recursive least squares, windows, signal detection, signal interpolation, signal separation.**

I. Introduction

THE problems of detecting, separating, and reconstructing signals from noisy observations are ubiquitous in signal processing. Accordingly, a wide range of signal processing techniques has been developed and used for such tasks. In some applications (e.g., speech detection and speaker separation), frequency domain methods have been successfully used. In other applications (e.g., digital communications [2]), the emphasis is on statistical methods: beginning with a generative statistical model, practical algorithms implement (exactly or approximately) principled statistical estimation methods. However, the statistical approach is not always feasible: practical estimation algorithms exist only for specific model classes, and more fundamentally, the specification of a sufficiently expressive statistical model may be elusive.

Manuscript received June 26, 2017; revised December 21, 2017 and April 18, 2018; accepted April 18, 2018. Date of publication May 7, 2018; date of current version June 11, 2018. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Yuichi Tanaka. This work was supported by the Swiss National Science Foundation under Grant CR23I2_166030. This paper was presented in part at the 13th IASTED International Conference on Biomedical Engineering, Innsbruck, Austria, February 2017. *(Corresponding author: Reto A. Wildhaber.)*

R. A. Wildhaber is with the Department of Information Technology and Electrical Engineering, ETH Zurich, Zurich 8092, Switzerland, and also with the Institute for Human Centered Engineering, Bern University of Applied Sciences, Bern 3012, Switzerland (e-mail: reto.wildhaber@bfh.ch).

N. Zalmai and H.-A. Loeliger are with the Department of Information Technology and Electrical Engineering, ETH Zurich, Zurich 8092, Switzerland (e-mail: zalmai@isi.ee.ethz.ch; loeliger@isi.ee.ethz.ch).

M. Jacomet is with the Institute for Human Centered Engineering, Bern University of Applied Sciences, Bern 3012, Switzerland (e-mail: marcel.jacomet@bfh.ch).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2018.2833804

In this paper, we describe a new approach whose development has been driven by the needs of biomedical signals such as ECG (electrocardiogram) signals, EMG (electromyogram) signals, and many others. Such signals are often composed of short pulses with characteristic shapes on top of a wandering baseline with occasional jumps and with bursts of outliers of arbitrary nature and scale. These different components need to be detected and separated from each other. Moreover, the pulse shapes and onsets need to be recovered as accurately as possible.

An overview of signal processing techniques for analyzing ECG waveforms is given in [3], and similar techniques have been used for other biomedical signals. Most such techniques use some combination of linear filtering (including matched filters [4], wavelets [5], or filter banks [6]) with some sort of thresholding. However, all these methods have limitations, especially when noise and erratic baselines cannot be neglected.

For the kind of signals mentioned above, successful applications of principled statistical methods seem to be rare. Candidates for suitable statistical models include linear dynamical systems with occasional model switches governed by a hidden Markov model [7]–[9]. However, such models require many parameters to be estimated, and even with fixed parameters, the computations required for the actual estimation are complex. A related approach was recently proposed in [10], [11], where sparse jumps in linear state space models are handled with normal priors with unknown variance. This latter approach requires only recursions with quadratic cost functions (as in the present paper), but the algorithms are iterative and computationally more demanding than those of the present paper.

In this paper, we borrow the idea of state space models from the mentioned statistical approaches; otherwise, we remain within a least-squares framework with local templates, which is shown to be much more versatile than has been appreciated in the literature. Unlike most of the statistical literature, we use *autonomous* linear state space models (LSSMs), i.e., LSSMs without an input, which leads to efficient *recursive* least-squares algorithms. In fact, the complexity of all algorithms proposed in this paper is linear in the number of data samples to be processed. The proposed algorithms all have the same general structure: there is a single forward (in time) recursion, there is an independent single backward recursion, and then there are some additional per-sample computations that combine the results of the two recursions. (This general structure is well-known, e.g., from the Baum–Welch algorithm for hidden Markov models [12] or from the two-filter version of Kalman smoothing [13]–[15].)

Autonomous LSSMs encode sums and products of exponentials, sinusoids, and polynomials (cf. [16]), all of which generically grow to infinity, either into the future or into the past. In order for autonomous LSSMs to be useful for signal modeling, we need localization: the state space model will be used only locally, within some window around the current time index (but the current time index runs over the whole time axis).

Localization is not a new idea: rectangular windows have always been used in naive template matching, localization with a one-sided exponential window has been standard in recursive least squares [17], [18], localized polynomial fitting with a rectangular window is the very idea of the Savitzky–Golay filter [19], [20], and localizing recursive least squares with a rectangular window was considered in [21], [22], see also [23] and [24].

In this paper, we go much further: elementary rectangular and exponential windows are combined into multi-section windows, and each section of such a window can use a different LSSM. In this way, we effectively construct models composed of several different LSSMs, each with its own window. In addition, windows may have holes: for example, a window with a hole can be used to detect arbitrary pulses or bursts of outliers (cf. Section III-G2). Nonetheless, the computational complexity remains linear in the number of data samples to be processed.

In addition, we will optionally use per-sample weights that indicate the reliability of individual samples. Such reliability information may be obtained and used in multi-stage processing as follows: in the first stage, each sample is scored according to how well it fits some given model (as will be detailed in Section III-G2); these per-sample scores are then used to define per-sample weights for the second stage of processing; and so on.

Unequal per-sample weights can also be used for adaptation to varying sampling rates, which in turn may arise from varying signal dynamics as illustrated in Example V-D.

Combining these ideas—multi-section windows and models, per-sample weights, and recursive least squares—we obtain a versatile toolbox that we illustrate with several examples. Some of these examples are mere toy examples (sacrificing practical relevance for clarity of exposition), while Example V-C (ECG baseline recovery, cf. Fig. 10) and Example V-D (signal reconstruction from a quantized version of the signal, cf. Fig. 14) are derived from real applications. In fact, the proposed approach has been developed for practical needs and it has successfully been used in a number of real applications in parallel with its development.

The paper is structured as follows. Section II recalls localized model fitting by squared error minimization as well as some basics of autonomous LSSMs. In Section III, we combined the ideas of window-based localization and per-sample weights into *cost segments* and describe the recursive computation of the corresponding cost. In Section IV, we combine such *cost segments* into multi-section *composite costs* and discuss their use. In Section V, we illustrate the proposed methods with several examples:

1) pulse detection in noise (Fig. 5),
2) separation of an unknown pulse from a baseline signal (Figs. 7 and 8),
3) ECG baseline estimation (Figs. 10 and 11),
4) reconstruction of a signal from its quantized version with smoothing adapted to the varying signal dynamics (Figs. 13 and 14),
5) Morse code decoding from a received signal with noise (Fig. 16).

The notation $\mathbb{R}_+$ and $\mathbb{R}_+^*$ will be used for the non-negative real numbers and for the strictly positive real numbers, respectively.

II. A Quick Primer

This section is a quick primer for any reader, not too familiar with the topics of model fitting by error minimization and to the use of autonomous time invariant state space models.

A. Model Fitting by Error Minimization

Let $y_i \in \mathbb{R}$ with time index $i \in \{1, \ldots, K\}$ be a given (observed) signal with $K \in \mathbb{N}$, $K \gg 1$ samples. Standard model-based approaches intend to fit a desired signal model $s_i(x)$, $i \in \{1, \ldots, K\}$, with model parameter(s) x to the observations by finding $\hat{x}$ which minimizes the squared error (L2 norm), i.e.,

$$\hat{x} = \underset{x}{\operatorname{argmin}} \sum_{i=1}^{K} \left(y_i - s_i(x)\right)^2 . \tag{1}$$

Then, the signal estimate is given by $s_i(\hat{x})$. For instance, given the observations y_i plotted in black circles in Fig. 1(a), the signal estimate using a sinusoidal model

$$s_i(x) = \left[a_0 \cos(\Omega i) - a_1 \sin(\Omega i)\right] , \tag{2}$$

with known frequency $\Omega \in \mathbb{R}_+$ but unknown model parameter $x = [a_0, a_1]^{\mathsf{T}} \in \mathbb{R}^2$, is displayed with red dots.

The standard L2 norm used in (1) weights each sample squared error equally. However, in many applications, it is of great interest to introduce different weights for each sample. Think of an observed signal where one can clearly distinguish "bad" samples, e.g., due to intermittent noise. Let $w_i \in \mathbb{R}_+$, $0 \leq w_i \leq 1$, denote an associated sample weight such that w_i is close to zero for any very unreliable sample y_i and w_i is close to one for very reliable samples. Then, (1) is modified to

$$\hat{x} = \underset{x}{\operatorname{argmin}} \sum_{i=1}^{K} w_i \left(y_i - s_i(x)\right)^2 , \tag{3}$$

which takes the sample weights into account. It is often possible to associate a quality measure to each observed sample y_i. For example, in Fig. 1(a), the samples from time index 15 to 24 seem to be outliers. Neglecting those bad samples when performing signal estimation using sample weights w_i leads to Fig. 1(b). Note that the signal estimate $s_i(\hat{x})$ (in red dots) almost coincides with the observed samples y (in black circles), as opposed to Fig. 1(a). Proceeding this way, this technique can also deal with missing data or non-uniform sampling by introducing adequate sample weights.

In other applications the signal model $s_i(x)$ is only valid within a certain window and not over the all time range. This especially happens when different models switch within a signal as in Fig. 2(a) where there is a clear model switch at time index

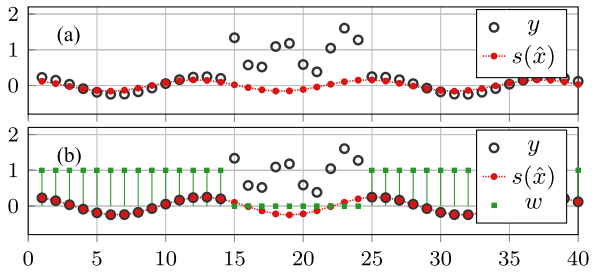


Fig. 1. Examples of fitting a model $s(x)$ to an observed signal y with or without given sample weights by squared-error minimization. (a) Using unweighted samples. (b) Using weighted samples with sample weights w.

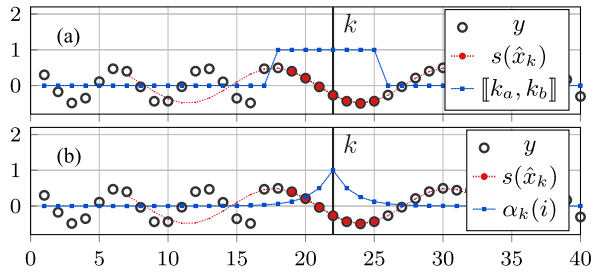


Fig. 2. Examples of fitting a model $s(x)$ to an observed signal y by squared-error minimization using a window. (a) Using a model with a rectangular window over the interval with limits $k+a$ and $k+b$, denoted as $[\![k_a\,, k_b]\!]$, highlighted at exemplary time index $k = 22$. (b) Using a model with a non-rectangular window $\alpha_k(i)$, highlighted at same exemplary time index.

17. Therefore, we would like to fit the observations with the model only locally, i.e., we limit the signal estimation in an interval with the limits $k+a$ and $k+b$ for some fixed a and $b \in \mathbb{Z}$, $a \leq b$, located around an index $k \in \{1, \ldots, K\}$, leading to minimizing the cost

$$\hat{x}_k = \underset{x}{\operatorname{argmin}} \sum_{i=k+a}^{k+b} \left(y_i - s_{i-k}(x)\right)^2 , \tag{4}$$

where the signal model is shifted at a time index k of interest. In Fig. 2(a), we display the estimated signal using this rectangular window.

If we do not want to specify sharp window bounds, we can alternatively use a two-sided exponentially-decaying window of parameter $\gamma \in \mathbb{R}_+^*$, $0 < \gamma < 1$. Then, the signal estimation boils down to the minimization of

$$\hat{x}_k = \underset{x}{\operatorname{argmin}} \sum_{i=1}^{K} \gamma^{|i-k|} \left(y_i - s_{i-k}(x)\right)^2 . \tag{5}$$

In Fig. 2(b), we display the estimated signal using this exponential window. More generally, a local fit with a window of any shape can be obtained by minimizing

$$\hat{x}_k = \underset{x}{\operatorname{argmin}} \sum_{i=1}^{K} \alpha_k(i) \left(y_i - s_{i-k}(x)\right)^2 , \tag{6}$$

where $\alpha_k(i) \in \mathbb{R}$, $0 \leq \alpha_k(i)$, is a window function located at index k.

Finally, combining the local fit idea via a window function with the use of sample weights, the general minimizing problem of interest for signal estimation writes as

$$\hat{x}_k = \underset{x}{\operatorname{argmin}} \sum_{i=k+a}^{k+b} \alpha_k(i) w_i \left(y_i - s_{i-k}(x)\right)^2 . \tag{7}$$

All the different cases (1), (3), (4), (5) and (6) are special cases of (7) and can be seen as time-dependent norms.

B. Autonomous State Space Models

The computation and minimization of all these proposed cost functions, using a sliding window, are in general computationally exhaustive. The use of state space models help to reduce the complexity by having efficient recursive cost computations.

We recall that the output s_k of an autonomous (i.e., input free), time-invariant state space system of order N given by

$$x_k = Ax_{k-1} + Bu_k \tag{8}$$

$$s_k = cx_k \tag{9}$$

with *state-transition matrix* $A \in \mathbb{R}^{N\times N}$, *output vector* $c \in \mathbb{R}^{1\times N}$, *state vector* $x_k \in \mathbb{R}^{N\times 1}$, *input vector* $Bu_k = 0$ can generate a wide class of discrete-time signals, parameterized by an *initial state vector* x_0. Such a signal can also be written as

$$s_k(x_0) = cA^k x_0 . \tag{10}$$

For our example in (2), we use a system of order $N = 2$ and assign

$$A = \begin{bmatrix} \cos\Omega & -\sin\Omega \\ \sin\Omega & \cos\Omega \end{bmatrix}, \quad c = [1, 0] , \quad B = 0 , \tag{11}$$

which leads, with $x = [a_0, a_1]^\mathsf{T} \in \mathbb{R}^2$, to

$$s_{i-k}(x) = cA^{i-k}x \tag{12}$$

$$= \left[a_0 \cos(\Omega(i-k)) - a_1 \sin(\Omega(i-k))\right] . \tag{13}$$

Further examples of autonomous state space systems are listed in Table I.

In the following, we combine the ideas sketched in this section. We use state space models along with time-dependent weighted norms in order to enable recursive and efficient error computation and minimization of cost functions of the general form (7), with a complexity of $\mathcal{O}(N^2 f)$, with $N \in \mathbb{N}$ the state space system order and $f \in \mathbb{R}_+$ the time resolution or sampling frequency.

III. Cost Segment: Definition and Computation

A. General Form and Properties

Given the observed signal $y_i \in \mathbb{R}$ and associated quality measure *sample weights* $w_i \in \mathbb{R}_+$, for $i \in \mathbb{Z}$, we look at the cost

$$\sum_{i=k+a}^{k+b} \alpha_{k+\delta}(i) w_i \left(y_i - s_{i-k}(x)\right)^2 \tag{14}$$

for $k \in \mathbb{Z}$, with bounds $a, b \in \mathbb{Z} \cup \{\pm\infty\}$ and $a \leq b$, the *window weights* $\alpha_k(i) \in \mathbb{R}_+^*$, $(i,k) \in \mathbb{Z}^2$ with shift $\delta \in \mathbb{Z}$, and the estimated signal $s_j(x) \in \mathbb{R}$, $j \in \mathbb{Z}$, parameterized by a *state vector* x.

In order to recursively and efficiently compute (14) the signal estimate $s_j(x)$ and the *window weight* $\alpha_k(i)$ need to be carefully selected. For that purpose we model $s_j(x)$ with a deterministic linear state space model of order $N \in \mathbb{N}$, i.e.,

$$s_j(x) = cA^j x, \quad j \in \mathbb{Z} \tag{15}$$

with *output vector* $c \in \mathbb{R}^{1\times N}$, *state-transition matrix* $A \in \mathbb{R}^{N\times N}$, *initial state vector* $x \in \mathbb{R}^{N\times 1}$ at $j = 0$. A list of common and often used models is given in Table I.

Concerning the *window weights* we focus on sequences satisfying the properties

$$\begin{cases} \alpha_k(k) = 1 \\ \alpha_{k+1}(i) = \gamma_{k+1}^{-1}\alpha_k(i), \end{cases} \tag{16}$$

for all $(k,i) \in \mathbb{Z}^2$ and for some fixed coefficients $\gamma_j \in \mathbb{R}_+^*$, $j \in \mathbb{Z}$. By induction, we can prove that for any $k \in \mathbb{Z}$ and $i \in \mathbb{Z}$, we have

$$\alpha_k(i) = \frac{\prod_{j=k+1}^{+\infty}\gamma_j}{\prod_{j=i+1}^{+\infty}\gamma_j} = \begin{cases} \prod_{j=i+1}^{k}\gamma_j^{-1} & \text{if } i < k, \\ \prod_{j=k+1}^{i}\gamma_j & \text{if } i > k, \\ 1 & \text{if } i = k. \end{cases} \tag{17}$$

Note that for any $k \in \mathbb{Z}$, the ratio

$$\frac{\alpha_k(i+1)}{\alpha_k(i)} = \frac{\alpha_0(i+1)}{\alpha_0(i)} = \gamma_{i+1} \tag{18}$$

between the *window weight* at $i+1$ and i is independent of k, which implies that the ratio between any *window weights* is independent of k. In the particular case where $\gamma_j = \gamma$, for all $j \in \mathbb{Z}$, we have

$$\alpha_k(i) = \gamma^{k-i}\ . \tag{19}$$

Using the estimate $s_j(x)$ as in (15) and the *window weights* (16), we get the parameterized *cost segment*

$$J_a^b(k,x,\theta) = \sum_{i=k+a}^{k+b}\alpha_{k+\delta}(i)w_i\left(y_i - cA^{i-k}x\right)^2 \tag{20}$$

with $x \in \mathbb{R}^{N\times 1}$ and parameter set $\theta = (c, \delta, \boldsymbol{\alpha})$ where $c \in \mathbb{R}^{1\times N}$, $\delta \in \mathbb{Z}$ and $\boldsymbol{\alpha} = [\alpha_k(i)]_{(k,i)\in\mathbb{Z}^2}$. This cost can be recursively and efficiently computed as shown in the next section. Note that the shift δ introduced in (20) leads to the normalization $\alpha_{k+\delta}(k+\delta) = 1$ for all $(k,\delta) \in \mathbb{Z}^2$. This normalization helps in designing suitable windows.

B. Recursive Cost Computation

Expanding (20), the *cost segment* can be parameterized as

$$J_a^b(k,x,\theta) = x^{\mathsf{T}}W_k x - 2x^{\mathsf{T}}\xi_k + \kappa_k \tag{21}$$

with

$$W_k = \sum_{i=k+a}^{k+b}\alpha_{k+\delta}(i)w_i\left(A^{i-k}\right)^{\mathsf{T}}c^{\mathsf{T}}c\left(A^{i-k}\right) \in \mathbb{R}^{N\times N} \tag{22}$$

$$\xi_k = \sum_{i=k+a}^{k+b}\alpha_{k+\delta}(i)w_i y_i\left(A^{i-k}\right)^{\mathsf{T}}c^{\mathsf{T}} \in \mathbb{R}^{N\times 1} \tag{23}$$

$$\kappa_k = \sum_{i=k+a}^{k+b}\alpha_{k+\delta}(i)w_i y_i^2 \in \mathbb{R}\ . \tag{24}$$

Additionally, we here introduce (for later use)

$$\nu_k = \sum_{i=k+a}^{k+b}\alpha_{k+\delta}(i)w_i \in \mathbb{R} \tag{25}$$

$$\varphi_k = \sum_{i=k+a}^{k+b}\alpha_{k+\delta}(i) \in \mathbb{R} \tag{26}$$

with ν_k the number of weighted samples in the window and φ_k the area of the window.

The *window weight* properties in (16) lead to two recursive ways to compute W_k, ξ_k, κ_k and ν_k: either as a forward (from k to $k+1$) or as a backward (from k to $k-1$) recursion. Let use the short notation $k_a = k+a$, $k_b = k+b$ and $k_\delta = k+\delta$. Then, the forward recursion is

$$\begin{aligned} W_{k+1} &= \gamma_{k_\delta+1}^{-1}A^{-\mathsf{T}}W_k A^{-1} \\ &\quad - \alpha_{k_\delta+1}(k_a)w_{k_a}\left(A^{a-1}\right)^{\mathsf{T}}c^{\mathsf{T}}c\left(A^{a-1}\right) \\ &\quad + \alpha_{k_\delta+1}(k_b+1)w_{k_b+1}\left(A^{b}\right)^{\mathsf{T}}c^{\mathsf{T}}c\left(A^{b}\right) \end{aligned} \tag{27}$$

$$\begin{aligned} \xi_{k+1} &= \gamma_{k_\delta+1}^{-1}A^{-\mathsf{T}}\xi_k \\ &\quad - \alpha_{k_\delta+1}(k_a)w_{k_a}\left(A^{a-1}\right)^{\mathsf{T}}c^{\mathsf{T}}y_{k_a} \\ &\quad + \alpha_{k_\delta+1}(k_b+1)w_{k_b+1}(A^{b})^{\mathsf{T}}c^{\mathsf{T}}y_{k_b+1} \end{aligned} \tag{28}$$

$$\begin{aligned} \kappa_{k+1} &= \gamma_{k_\delta+1}^{-1}\kappa_k - \alpha_{k_\delta+1}(k_a)w_{k_a}y_{k_a}^2 \\ &\quad + \alpha_{k_\delta+1}(k_b+1)w_{k_b+1}y_{k_b+1}^2 \end{aligned} \tag{29}$$

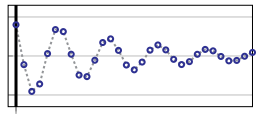
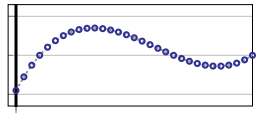
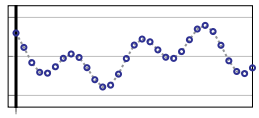
$$\begin{aligned} \nu_{k+1} &= \gamma_{k_\delta+1}^{-1}\nu_k - \alpha_{k_\delta+1}(k_a)w_{k_a} \\ &\quad + \alpha_{k_\delta+1}(k_b+1)w_{k_b+1}\ . \end{aligned} \tag{30}$$

The backward recursion is

$$\begin{aligned} W_{k-1} &= \gamma_{k_\delta}A^{\mathsf{T}}W_k A \\ &\quad + \alpha_{k_\delta-1}(k_a-1)w_{k_a-1}\left(A^{a}\right)^{\mathsf{T}}c^{\mathsf{T}}c\left(A^{a}\right) \\ &\quad - \alpha_{k_\delta-1}(k_b)w_{k_b}\left(A^{b+1}\right)^{\mathsf{T}}c^{\mathsf{T}}c\left(A^{b+1}\right) \end{aligned} \tag{31}$$

$$\begin{aligned} \xi_{k-1} &= \gamma_{k_\delta}A^{\mathsf{T}}\xi_k \\ &\quad + \alpha_{k_\delta-1}(k_a-1)w_{k_a-1}\left(A^{a}\right)^{\mathsf{T}}c^{\mathsf{T}}y_{k_a-1} \\ &\quad - \alpha_{k_\delta-1}(k_b)w_{k_b}(A^{b+1})^{\mathsf{T}}c^{\mathsf{T}}y_{k_b} \end{aligned} \tag{32}$$

TABLE I
SOME STATE SPACE MODELS

Signal			*State Space Model Parameters*	*Example*
Name	$s_i(x)$	*Parameters*	cA^ix	$s_i(x) = cA^ix$
Decaying (co-)sine	$\rho^i[a_0\cos(\Omega i) - a_1\sin(\Omega i)]$	$\rho \in \mathbb{R}_+,\ \Omega \in \mathbb{R}_+$	$A = \rho\begin{bmatrix}\cos\Omega & -\sin\Omega \\ \sin\Omega & \cos\Omega\end{bmatrix}$ $c = [1, 0]$ $x = [a_0, a_1]^\mathsf{T} \in \mathbb{R}^{2\times 1}$	
Line	$a_0 + a_1 i$	$(a_0, a_1) \in \mathbb{R}^2$	$A = \begin{bmatrix}1 & 1 \\ 0 & 1\end{bmatrix}$ $c = [1, 0]$ $x = [a_0, a_1]^\mathsf{T} \in \mathbb{R}^{2\times 1}$	
Polynomial of order $N-1$	$a_0+a_1 i+\ldots+a_{N-1}i^{N-1}$	$(a_0, a_1, \ldots, a_{N-1}) \in \mathbb{R}^N$	$A = \begin{bmatrix}A_{1,1} & \cdots & A_{1,N} \\ \vdots & A_{n,n'} & \vdots \\ A_{N,1} & \cdots & A_{N,N}\end{bmatrix}$ $A_{n,n'} = \begin{cases}\binom{n'}{n} & \text{if } n \le n' \\ 0 & \text{if } n > n'\end{cases}$ $c = [1, 0, \ldots, 0] \in \mathbb{R}^{1\times N}$ $x = [a_0, \ldots, a_{N-1}]^\mathsf{T} \in \mathbb{R}^{N\times 1}$	
Linear combination of M systems	$s_i^{(1)}(x^{(1)})+\ldots+s_i^{(M)}(x^{(M)})$	$s_i^{(m)}(x^{(m)}) \in \mathbb{R}$ $m \in \{1, \ldots, M\}$	$A = \operatorname{diag}(A_1, \ldots, A_M)$ $c = [c_1, \ldots, c_M]$ $x = \left[\left(x^{(1)}\right)^\mathsf{T}, \ldots, \left(x^{(M)}\right)^\mathsf{T}\right]^\mathsf{T}$	

$$\kappa_{k-1} = \gamma_{k_\delta}\kappa_k + \alpha_{k_\delta-1}(k_a-1)w_{k_a-1}y^2_{k_a-1} - \alpha_{k_\delta-1}(k_b)w_{k_b}y^2_{k_b} \tag{33}$$

$$\nu_{k-1} = \gamma_{k_\delta}\nu_k + \alpha_{k_\delta-1}(k_a-1)w_{k_a-1} - \alpha_{k_\delta-1}(k_b)w_{k_b}\ . \tag{34}$$

The recursions for φ_k are easily obtained from the recursion of ν_k by considering $w_k = 1$ for $k \in \mathbb{Z}$, and thus, are not explicitly stated.

In order to compute the cost (20) for all $k \in \{k_0, \ldots, K\}$, $k_0 \in \mathbb{Z}$, $K \in \mathbb{Z}$, the forward recursion begins at $k = k' = \min(k_0 - b, k_0) - 1$, and the backward at $k = k' = \max(K - a, K) + 1$, with initial values $W_{k'} = 0$, $\xi_{k'} = 0$, $\kappa_{k'} = 0$, $\nu_{k'} = 0$, $\varphi_{k'} = 0$. Wherever y_k is an observation out of range, e.g., $k < k_0$ or $k > K$, we set $w_k = 0$ (and thus $y_k w_k = 0$). The recursion and initialization of $\alpha_k(\cdot)$ are already implied in (16) or (18).

C. Some Choices of Window Functions

For now, we assume that the *sample weights* w_i, $i \in \mathbb{Z}$ are given a priori. Then, it remains to select the parameters γ_j of the *window weights* (see (16)), $j \in \mathbb{Z}$, which is actually not convenient.

Therefore, we introduce the *effective sample number* as

$$\overleftarrow{g}_k = \sum_{i=-\infty}^{k+\delta} \alpha_{k+\delta}(i)w_i \tag{59}$$

or

$$\overrightarrow{g}_k = \sum_{i=k+\delta}^{+\infty} \alpha_{k+\delta}(i)w_i\ , \tag{60}$$

(if any is bounded) that corresponds to the weighted number of samples in the window to the left or right of $k + \delta$. The choice of the bound $k + \delta$ in these two sums usually coincides with $k + a$ for right-sided windows or $k + b$ for left-sided windows. However, we intentionally do not restrict it to these two values.

Note that $\overleftarrow{g}_k$ for the left-sided window satisfies the following recursive relation

$$\overleftarrow{g}_{k+1} = \gamma^{-1}_{k+\delta+1}\overleftarrow{g}_k + w_{k+\delta+1} \tag{61}$$

and $\overrightarrow{g}_k$ for the right-sided window

$$\overrightarrow{g}_{k-1} = \gamma_{k+\delta}\overrightarrow{g}_k + w_{k+\delta-1}\ . \tag{62}$$

In fact, we are interested in a fixed *effective sample number*, i.e., $\overleftarrow{g}_k = g$ or $\overrightarrow{g}_k = g$ independent of k. We prefer to select g instead of γ_k. For weighted samples $(w_i \in \mathbb{R}_+, i \in \mathbb{Z})$, g is the fixed point of (61) or (62), and thus, γ_k is function of w_k.

For unweighted samples $(w_i = 1, i \in \mathbb{Z})$, an exponential window is generated with $\gamma_k = \gamma \neq 1$, for all $k \in \mathbb{Z}$. Then,

TABLE II
SELECTION OF COST SEGMENTS – USING CONSTANT WINDOW DECAY

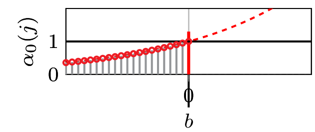
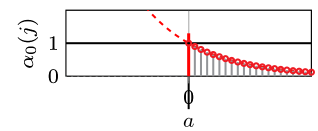
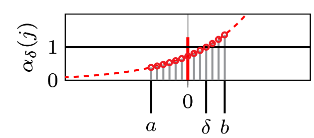
Segment	*Recursions*	*Sample* $(k=0)$
Cost Segment $J_a^b(k,x,\delta) = \sum_{i=k+a}^{k+b} \gamma^{i-k-\delta} w_i \left(y_i - s_{i-k}(x)\right)^2$, $k \in \{k_0,\dots,K\}$		
Left-sided ($a{=}-\infty$, $b{=}0$, $\gamma{<}1$, $\delta=0$)	$W_{k+1} = \gamma^{-1}A^{-\mathsf{T}}W_kA^{-1} + w_{k+1}c^\mathsf{T}c$ (35) $\xi_{k+1} = \gamma^{-1}A^{-\mathsf{T}}\xi_k + w_{k+1}c^\mathsf{T}y_{k+1}$ (36) $\kappa_{k+1} = \gamma^{-1}\kappa_k + w_{k+1}y_{k+1}^2$ (37) $\nu_{k+1} = \gamma^{-1}\nu_k + w_{k+1}$ (38)	
Right-sided ($a{=}0$, $b{=}\infty$, $\gamma{>}1$, $\delta=0$)	$W_{k-1} = \gamma A^\mathsf{T}W_kA + w_{k-1}c^\mathsf{T}c$ (39) $\xi_{k-1} = \gamma A^\mathsf{T}\xi_k + w_{k-1}c^\mathsf{T}y_{k-1}$ (40) $\kappa_{k-1} = \gamma\kappa_k + w_{k-1}y_{k-1}^2$ (41) $\nu_{k-1} = \gamma\nu_k + w_{k-1}$ (42)	
Finite support ($-\infty{<}a{\le}b{<}\infty$, $\delta\in\mathbb{Z}$)	$W_{k+1} = \gamma^{-1}A^{-\mathsf{T}}W_kA^{-1} - \gamma^{a-\delta-1}w_{k+a}\left(A^{a-1}\right)^\mathsf{T}c^\mathsf{T}c\left(A^{a-1}\right) + \gamma^{b-\delta}w_{k+b+1}\left(A^b\right)^\mathsf{T}c^\mathsf{T}cA^b$ (43) $\xi_{k+1} = \gamma^{-1}A^{-\mathsf{T}}\xi_k - \gamma^{a-\delta-1}w_{k+a}\left(A^{a-1}\right)^\mathsf{T}c^\mathsf{T}y_{k+a} + \gamma^{b-\delta}w_{k+b+1}\left(A^b\right)^\mathsf{T}c^\mathsf{T}y_{k+b+1}$ (44) $\kappa_{k+1} = \gamma^{-1}\kappa_k - \gamma^{a-\delta-1}w_{k+a}y_{k+a}^2 + \gamma^{b-\delta}w_{k+b+1}y_{k+b+1}^2$ (45) $\nu_{k+1} = \gamma^{-1}\nu_k - \gamma^{a-\delta-1}w_{k+a} + \gamma^{b-\delta}w_{k+b+1}$ (46) - - - - - - - - - - - - - - - - $W_{k-1} = \gamma A^\mathsf{T}W_kA + \gamma^{a-\delta}w_{k+a-1}\left(A^a\right)^\mathsf{T}c^\mathsf{T}cA^a - \gamma^{b-\delta+1}w_{k+b}\left(A^{b+1}\right)^\mathsf{T}c^\mathsf{T}cA^{b+1}$ (47) $\xi_{k-1} = \gamma A^\mathsf{T}\xi_k + \gamma^{a-\delta}w_{k+a-1}\left(A^a\right)^\mathsf{T}c^\mathsf{T}y_{k+a-1} - \gamma^{b-\delta+1}w_{k+b}\left(A^{b+1}\right)^\mathsf{T}c^\mathsf{T}y_{k+b}$ (48) $\kappa_{k-1} = \gamma\kappa_k + \gamma^{a-\delta}w_{k+a-1}y_{k+a-1}^2 - \gamma^{b-\delta+1}w_{k+b}y_{k+b}^2$ (49) $\nu_{k-1} = \gamma\nu_k + \gamma^{a-\delta}w_{k+a-1} - \gamma^{b-\delta+1}w_{k+b}$ (50)	
Forward Recursions $(k\to k{+}1)$, $k' = \min(k_0-b, k_0)-1$ $\mid$ *Backward Recursions* $(k \to k{-}1)$, $k' = \max(K-a, K)+1$. *Initialization* $W_{k'}=0$, $\xi_{k'}=0$, $\kappa_{k'}=0$, $\nu_{k'}=0$. For $k<k_0$ or $k>K$ set $w_k=0$ (and thus $y_kw_k=0$).		

TABLE III
SELECTION OF COST SEGMENTS – USING ADAPTIVE WINDOW DECAY

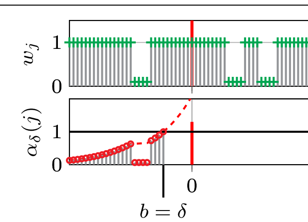
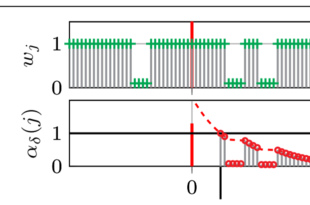
Segment	*Recursions*	*Sample* $(k=0)$
Cost Segment $J_a^b(k,x,\delta) = \sum_{i=k+a}^{k+b} \alpha_{k+\delta}(i) w_i \left(y_i - s_{i-k}(x)\right)^2$, $k \in \{k_0,\dots,K\}$		
Left-sided ($a{=}-\infty$, $b\in\mathbb{Z}$, $\delta=b$, $\overrightarrow{\gamma}_k{<}1$)	$W_{k+1} = \overrightarrow{\gamma}_{k+\delta+1}^{-1}A^{-\mathsf{T}}W_kA^{-1} + w_{k+b+1}c^\mathsf{T}c$ (51) $\xi_{k+1} = \overrightarrow{\gamma}_{k+\delta+1}^{-1}A^{-\mathsf{T}}\xi_k + w_{k+b+1}c^\mathsf{T}y_{k+b+1}$ (52) $\kappa_{k+1} = \overrightarrow{\gamma}_{k+\delta+1}^{-1}\kappa_k + w_{k+b+1}y_{k+b+1}^2$ (53) $\nu_{k+1} = \overrightarrow{\gamma}_{k+\delta+1}^{-1}\nu_k + w_{k+b+1}$ (54)	
Right-sided ($a\in\mathbb{Z}$, $b{=}\infty$, $\delta=a$, $\overleftarrow{\gamma}_k{>}1$)	$W_{k-1} = \overleftarrow{\gamma}_{k+\delta}A^\mathsf{T}W_kA + w_{k+a-1}c^\mathsf{T}c$ (55) $\xi_{k-1} = \overleftarrow{\gamma}_{k+\delta}A^\mathsf{T}\xi_k + w_{k+a-1}c^\mathsf{T}y_{k+a-1}$ (56) $\kappa_{k-1} = \overleftarrow{\gamma}_{k+\delta}\kappa_k + w_{k+a-1}y_{k+a-1}^2$ (57) $\nu_{k-1} = \overleftarrow{\gamma}_{k+\delta}\nu_k + w_{k+a-1}$ (58)	
Forward Recursions $(k\to k{+}1)$, $k' = \min(k_0-b, k_0)-1$ $\mid$ *Backward Recursions* $(k \to k{-}1)$, $k' = \max(K-a, K)+1$. *Initialization* $W_{k'}=0$, $\xi_{k'}=0$, $\kappa_{k'}=0$, $\nu_{k'}=0$. For $k<k_0$ or $k>K$ set $w_k=0$ (and thus $y_kw_k=0$).		

TABLE IV
SOME WINDOW FUNCTIONS

Window	*Left-Sided*	*Right-Sided*
Rectangular limit case $g \to \infty$	$\overrightarrow{\gamma} = \overleftarrow{\gamma} = 1$	
Constant Decay (independent of w_k)	$\overleftarrow{\gamma}_k = \overleftarrow{\gamma} = \dfrac{g}{g-1} = \overleftarrow{G}(g)$	$\overrightarrow{\gamma}_k = \overrightarrow{\gamma} = \left(\dfrac{g}{g-1}\right)^{-1} = \overrightarrow{G}(g)$
Adaptive Decay (depending on w_k)	$\overleftarrow{\gamma}_k = \dfrac{g}{g-w_k} = \overleftarrow{G}(g, w_k)$	$\overrightarrow{\gamma}_k = \left(\dfrac{g}{g-w_{k-1}}\right)^{-1} = \overrightarrow{G}(g, w_{k-1})$
With $g > 1$ and $0 \leq w_k \leq 1$ for $k \in \{1, \ldots, K\}$		

TABLE V
SOME COMMON COST MINIMIZATIONS

Constraint	*Known*	*Minimization*
Unconstrained $x \in \mathbb{R}^N$	-	$\hat{x}_k = W_k^{-1}\xi_k$ (70)
Linear Scalar $x = Hv,\ v \in \mathbb{R}$	$H \in \mathbb{R}^{N\times 1}$	$\hat{v}_k = \dfrac{\xi_k^{\mathsf{T}} H}{H^{\mathsf{T}} W_k H}$ (71)
Linear Combination With Offset $x = Hv + h,$ $v \in \mathbb{R}^M$	$H \in \mathbb{R}^{N\times M}$ $h \in \mathbb{R}^N$	$\hat{v}_k = (H^{\mathsf{T}} W_k H)^{-1} H^{\mathsf{T}}(\xi_k - W_k h)$ (72)
$N \in \mathbb{N}$: system order; $M \in \mathbb{N}$: constraint subspace dimension, $M \leq N$.		

$\alpha_k(i) = \gamma^{k-i}$ and

$$\overleftarrow{g} = \sum_{i=-\infty}^{k+\delta} \alpha_{k+\delta}(i) = \frac{1}{1-\gamma^{-1}} \tag{63}$$

$$\overrightarrow{g} = \sum_{i=k+\delta}^{+\infty} \alpha_{k+\delta}(i) = \frac{1}{1-\gamma}\ . \tag{64}$$

Again, we are interested in a fixed *effective sample number* and thus, assign $\overleftarrow{g} = g$ or $\overrightarrow{g} = g$. The relation to γ for any selected $g > 1$ immediately follows. A summary of choices for γ_k and γ, respectively, by selecting g is given in Table IV.

The use of a rectangular window (see Table IV) might lead to numerical instabilities. Thus, we recommend to use a window of constant decay with a fairly large g instead. Furthermore, note that for numerical stability the selection of a left- or right-sided window area already implies the direction of the recursions, i.e., a left-sided window leads to a forward recursion and vice versa.

In both cases, using weighted or unweighted samples, we are guaranteed that the cost function takes g effective samples into account. In particular, for weighted samples, in regions where w_i is small, the *window weights* $\alpha_{k+\delta}(i)$ stretches to cover more past samples whereas in regions where w_i is big enough, the *window weights* $\alpha_{k+\delta}(i)$ shrinks and covers only recent samples.

An alternative justification for this approach is given by the statistical modeling theory. Assigning the sample variance $\sigma_i^2 = w_i^{-1} \in \mathbb{R}_+^*$ to the sample y_i, fixing the *effective sample number* is equivalent to fixing the variance of the associated *cost segment*.

In practice, the *cost segment* bounds a or b are often set to 0 or $\pm\infty$, leading to simplifications in the recursions (27)–(33). A list of recursions for common bounds is given in Table II with constant window decay ($\gamma_k = \gamma$, independent of k) and in Table III with adaptive window decay (γ_k, depending on k).

D. Relative Sample Density

We define the *relative sample density* as

$$\phi_k = \frac{\nu_k}{\varphi_k} \in \mathbb{R}\ . \tag{65}$$

The *relative sample density* is a local measure of the relative number of weighted samples considered in the current window and depends on the *sample weights* w_i. Note that since $0 \leq w_j \leq 1$, $j \in \mathbb{Z}$, we have $\phi_k \in [0, 1]$. (Within this context, we could also denote ν_k as the *sample density*.)

E. Model Fitting by Cost Minimization

For most of the signal separation or detection tasks, the cost $J_a^b(k, x, \theta)$ in (21) needs to be minimized with respect to the constrained or unconstrained state vector x for a fixed index k. Here we focus on linear constraints, i.e.,

$$x = Hv + h \tag{66}$$

with fixed $H \in \mathbb{R}^{N\times M}$ and $h \in \mathbb{R}^{N\times 1}$ and with unknown $v \in \mathbb{R}^{M\times 1}$. Thus, using (21) we have

$$\hat{v}_k = \underset{v\in\mathbb{R}^M}{\operatorname{argmin}} J_a^b(k, Hv + h, \theta) \tag{67}$$

$$= \underset{v\in\mathbb{R}^M}{\operatorname{argmin}} \left(v^{\mathsf{T}} H^{\mathsf{T}} W_k H v - 2v^{\mathsf{T}} H^{\mathsf{T}}(\xi_k - W_k h)\right) \tag{68}$$

$$= (H^{\mathsf{T}} W_k H)^{-1} H^{\mathsf{T}}(\xi_k - W_k h)\ , \tag{69}$$

where for the last step from (68) to (69) the derivative with respect to v is computed and set to zero. A summary of common optimization problems derived from (69) is given in Table V. Additional pertinent optimization problems are summarized in [16, Chapter 6]. As is obvious e.g., from (71) in Table V, the actual computation of (69) does not necessarily require a matrix inversion. Some important cases where the matrix inversion in (69) can be simplified are discussed in Section III-H.

F. Detection Using Error Ratios, Cost Ratios

We want to robustly detect the occurrence of certain events (e.g., pulses with some characteristic shapes or model switches) characterized by the set $\mathcal{X}_1$ of allowed states, in contrast to the alternative set $\mathcal{X}_2$, in a single-channel signal. For that purpose, we introduce error ratios.

The per-sample error ratio between a first *cost segment*, J_a^b with model parameters θ_1 and a second, alternative *cost segment*,

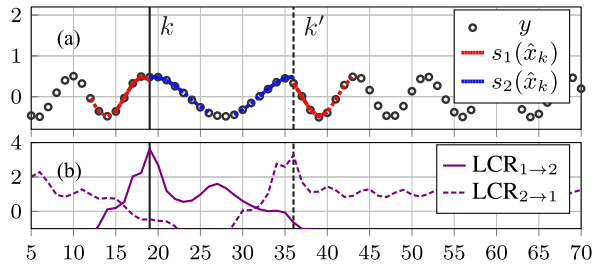


Fig. 3. Toy example of model switch detection using *log-cost ratios*. (a) Observations y (black circles) generated by two alternating sinusoidal models of different frequencies; trajectories at detected model switches from model 1, $s_1(\hat{x}_k)$ (red line), to model 2, $s_2(\hat{x}_k)$ (blue line), at index $k = 19$, and from model 2 to model 1 at index $k' = 36$. (b) *Log-cost ratio* $\mathrm{LCR}_{1\to2}$ (solid line) detecting switches from model 1 to model 2, and $\mathrm{LCR}_{2\to1}$ (dashed line) detecting switches from model 2 to model 1.

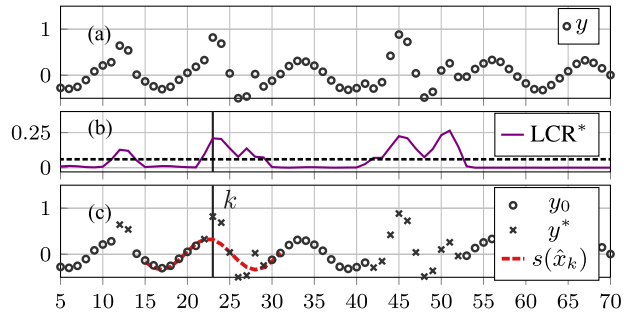


Fig. 4. Toy example of outlier detection. (a) Observations y with outliers. (b) *Log-cost ratio* LCR^* (solid line) with threshold for outlier detection (dashed line). (c) Detected outliers y^*, remaining observations y_0, and local model fit of model $s(\hat{x}_k)$ with state estimate $\hat{x}_k$ around index $k = 23$ (red dashed line). This example uses a model of a decaying sinusoidal with *cost segment* parameters: $a = -\infty$, $\bar{a} = -3$, $\bar{b} = 3$, and $b = +\infty$.

$\bar{J}_a^b$ with model parameters θ_2 is

$$r_k = \frac{\min\limits_{x_1\in\mathcal{X}_1} \frac{1}{\nu_k^{(1)}} J_a^b(k, x_1, \theta_1)}{\min\limits_{x_2\in\mathcal{X}_2} \frac{1}{\nu_k^{(2)}} \bar{J}_a^b(k, x_2, \theta_2)} \in \mathbb{R}_+ \,, \tag{73}$$

with $\nu_k^{(1)}$ (resp. $\nu_k^{(2)}$) the number of weighted samples in the first (resp. second) *cost segment*. Wherever the two *cost segments* have the same and non-adaptive windows, (73) simplifies into the error ratio

$$q_k = \frac{\min\limits_{x_1\in\mathcal{X}_1} J_a^b(k, x_1, \theta_1)}{\min\limits_{x_2\in\mathcal{X}_2} \bar{J}_a^b(k, x_2, \theta_2)} \in \mathbb{R}_+ \,. \tag{74}$$

In statistical theory, this ratio, up to irrelevant constants, is actually interpreted as a likelyhood ratio between two hypotheses, cf. [16]. In analogy, we introduce the *log-cost ratio* (LCR). The LCR between two models is

$$\mathrm{LCR}_k = -\frac{1}{2}\log \frac{\min\limits_{x_1\in\mathcal{X}_1} \frac{1}{\nu_k^{(1)}} J_a^b(k, x_1, \theta_1)}{\min\limits_{x_2\in\mathcal{X}_2} \frac{1}{\nu_k^{(2)}} \bar{J}_a^b(k, x_2, \theta_2)} \,, \tag{75}$$

and using (73)

$$\mathrm{LCR}_k = -\frac{1}{2}\log(r_k) \,, \tag{76}$$

or using (74)

$$\mathrm{LCR}_k = -\frac{1}{2}\log(q_k) \,. \tag{77}$$

We note, that the local minima of the error ratio coincides with the local maxima of the LCR. Since the error ratios are better displayed in a logarithmic scale, we subsequently use LCR's instead of error ratios. As a side remark, the normalized error ratio as in (73) is also known as the sample entropy in statistical theory.

G. Model Switches and Outliers

1) Model Switch Detection: Let $y \in \mathbb{R}^K$, $K \in \mathbb{N}$ be the observations of a signal generator switching between two different models. Furthermore, let $J_a^0(k, x^{(1)}, \theta_1)$ be the *cost segment*, fitting the first model to the left of index k, and $\bar{J}_1^b(k, x^{(2)}, \theta_2)$ the *cost segment*, fitting the second model to the right of index k; both with bounds $a, b \in \mathbb{Z} \cup \{\pm\infty\}$, $a \le 0 < b$, but individual *state vectors* $x_k^{(1)} \in \mathcal{X}_1$ and $x_k^{(2)} \in \mathcal{X}_2$, and with individual model parameters θ_1 and θ_2. To detect switches from model 1 to model 2, we compute the *log-cost ratio*

$$\overset{1\to2}{\mathrm{LCR}}_k = -\frac{1}{2}\log\left(\frac{\min\limits_{x\in\mathcal{X}_1} J_a^0(k, x, \theta_1) + \min\limits_{x\in\mathcal{X}_2} \bar{J}_1^b(k, x, \theta_2)}{\min\limits_{x\in\mathcal{X}_1} J_a^b(k, x, \theta_1)}\right) \tag{78}$$

for all indices k. At abrupt model changes, this LCR should rise. Thus, we extract the local maxima from the LCR using any common peak detector algorithm. An illustrative example with two sinusoidal signals of different frequencies is given in Fig. 3.

2) Outlier Detection: Let $y \in \mathbb{R}^K$, $K \in N$ be the observations and $J_a^b(k, x, \theta)$ the *cost segment* fitting the signal around index k with bounds $a, b \in \mathbb{Z} \cup \{\pm\infty\}$, $a \le b$, with *state vector* $x_k \in \mathcal{X}$, and model parameters θ. We want to identify all samples in y not following that model (and to exclude them from further processing steps, i.e., to treat them as outliers). To do so, we compare at any index k the fitting error of the *cost segment* with bounds a and b to the fitting error of a *cost segment* with a gap spanning the subinterval $\{\bar{a}+1, \ldots, \bar{b}-1\}$, $\bar{a}, \bar{b} \in \mathbb{Z}$, $a \le \bar{a} \le \bar{b} \le b$, i.e., we consider the *log-cost ratio*

$$\mathrm{LCR}_k^* = -\frac{1}{2}\log\left(\frac{\frac{1}{\nu_{\bar{a}}+\nu_{\bar{b}}}\left(J_a^{\bar{a}}(k, \hat{x}_k, \theta) + J_{\bar{b}}^b(k, \hat{x}_k, \theta)\right)}{\frac{1}{\nu_0} J_a^b(k, \hat{x}_k, \theta)}\right) \tag{79}$$

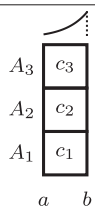
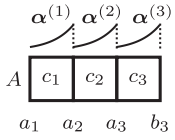
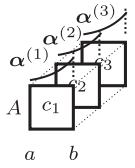
using the cost (sub)segments $J_a^{\bar{a}}$, $J_{\bar{b}}^b$, and J_a^b with corresponding numbers of weighted samples $\nu_{\bar{a}}$, $\nu_{\bar{b}}$, and ν_0 (cf. (25)) and with the localized state estimate

$$\hat{x}_k = \min_{x\in\mathcal{X}}\left(J_a^{\bar{a}}(k, x, \theta) + J_{\bar{b}}^b(k, x, \theta)\right) \,. \tag{80}$$

Finally, to identify the outliers, we apply a threshold on the LCR_k^* in (79).

Note that the selection of the subinterval does not restrict the outlier detection to a particular outlier position or outlier burst length, but rather controls the time scale of the detection method: short intervals preferably detect single outlier samples while wider intervals have their preference on outlier bursts.

TABLE VI
INSTANCES OF COMPOSITE COSTS

	Symbol	$\tilde{J}(k,x,\Theta)$ General Form	$\tilde{J}(k,x,\Theta)$ $\sum$ Cost Segments	Parameters
Window Weights Superposition		$\sum_{i=k+a}^{k+b}\Big[\sum_{q=1}^{Q} v_q \alpha_{k+\delta_q}^{(q)}(i)\Big] w_i \big(y_i - cA^{i-k}x\big)^2$	$\sum_{q=1}^{Q} v_q J_a^b(k,x,\theta_q)$	$\theta_q = (c, \delta_q, \boldsymbol{\alpha}^{(q)})$ $q \in \{1,\dots,Q\}$
Model Superposition		$\sum_{i=k+a}^{k+b} \alpha_{k+\delta}(i) w_i \Big(y_i - \sum_{m=1}^{M} c_m A_m^{i-k} x^{(m)}\Big)^2$	$J_a^b(k,x,\theta)$	$A = \mathrm{diag}(A_1,\dots,A_M)$ $x^{\mathsf{T}} = [(x^{(1)})^{\mathsf{T}},\dots,(x^{(M)})^{\mathsf{T}}]$ $\theta = ([c_1,\dots,c_M], \delta, \boldsymbol{\alpha})$
Non-Overlapping Segments Chaining		$\sum_{p=1}^{P}\Big[\sum_{i=k+a_p}^{k+b_p} \alpha_{k+\delta}^{(p)}(i) w_i \big(y_i - c_p A^{i-k}x\big)^2\Big]$	$\sum_{p=1}^{P} J_{a_p}^{b_p}(k,x,\theta_p)$	$a_p \le b_p$ and $b_p < a_{p+1}$ $\theta_p = (c_p, \delta_p, \boldsymbol{\alpha}^{(p)})$ $p \in \{1,\dots,P\}$
Concurrent Segments		$\sum_{i=k+a}^{k+b} w_i\Big[\sum_{r=1}^{R} \alpha_k^{(r)}(i)\big(y_i - c_r A^{i-k}x\big)^2\Big]$	$\sum_{r=1}^{R} J_a^b(k,x,\theta_r)$	$\theta_r = (c_r, \delta_r, \boldsymbol{\alpha}^{(r)})$ $r \in \{1,\dots,R\}$

Thus, the interval borders $\bar{a}$ and $\bar{b}$ are tuning parameters and its selection is not too critical to obtain valid result. A simple outlier detection example detecting outliers of different burst length with using a constant subinterval length is given in Fig. 4.

3) Sample Weights for Subsequent Processing: So far, we have considered the *sample weights* w_i to be given along with the observations, e.g., as the output of an initial processing step. While there exist many methods and strategies to obtain meaningful *sample weights*, we have discussed in Section III-G1 and Section III-G2 two methods based on our approach, suitable to empirically obtain such *sample weights*. The immediate application of these two approaches is given in Example V-B and in Example V-C.

H. Remarks on the Cost Computation

The computational complexity of FIR filters scales linearly with the filter length $L \in \mathbb{N}$ and linearly with the observation signal length K, leading to the complexity $\mathcal{O}(KL)$. Thanks to the use of autonomous linear state space systems of order $N \in \mathbb{N}$, the effort to compute the proposed time-dependent weighted L2 norms in its general form as in (14) reduces to the complexity $\mathcal{O}(KN^2)$ (cf. also [25]). Note that in any practical application $N \ll L$ and that the complexity of the proposed method is independent of the window parameters; for windows of finite support, it is, thus, *independent* of the window length.

I. Remarks on the Cost Minimization

The minimization of (69) may require to solve a system of linear equations. As this minimization simplifies in many important special cases of (14), we provide next a (non-exhaustive) list of such simplifications:

1) Recursive Inversions of W_k: To efficiently compute the inverse of W_{k+1} as in (51) (and correspondingly the inverse of W_{k-1} as in (55)), we apply the matrix inversion lemma, leading to a recursive form for the inverse of W_k avoiding any matrix inversions. For the forward recursion $k \in \{1,\dots,K\}$, we get

$$W_{k+1}^{-1} = G_k - \frac{w_{k_b+1} G_k c^{\mathsf{T}} c G_k}{1 + w_{k_b+1} c G_k c^{\mathsf{T}}} \tag{81}$$

with

$$G_k = \overrightarrow{\gamma}_{k_\delta+1} A W_k^{-1} A^{\mathsf{T}}. \tag{82}$$

A corresponding formula holds for the inverse of W_{k-1} using backward recursions.

2) Precomputation of W_k (and Its Inverse): If all parameters are independent of k, i.e., $w_k = w$ and $\gamma_k = \gamma$ (cf. Table II), then the recursions for W_k as in (27) and (31) lead to a steady state. As such, when $W_{k+1} = W_k$, equations (27) and (31) result in a Lyapunov equations [14, Appendix D]. As an immediate consequence, W_k converges as $k \to \infty$ and is easily precomputed offline, along with the matrix inversion in (69).

3) Independence of ξ_k and W_k (and Its Inverse) From the State Vector: The values ξ_k and W_k are independent of the initial *state vector* x, as obvious from the expanded *cost segment* (21). This leads to reduced computational burden, whenever we need to compute multiple *cost segments* sharing a common LSSM and common window parameters: All these *cost segments* then also share the quantities ξ_k and W_k, and, depending on the minimization constraints, also the inverse in (69). An example

TABLE VII
SELECTION OF WINDOW WEIGHTS SUPERPOSITIONS

Window	*Symbol*	*Design Parameters*	*Figure (left-sided)*
$\infty \geq g_1 > g_2 \geq 1$ | left-sided: $\gamma_1 = \overleftarrow{G}(g_1)$, $\gamma_2 = \overleftarrow{G}(g_2)$ | right-sided: $\gamma_1 = \vec{G}(g_1)$, $\gamma_2 = \vec{G}(g_2)$			
Exponential $\tilde{\alpha}_0(i) = \gamma_1^i$	$\boldsymbol{\alpha}$		1, 0, 0
Pulse $\tilde{\alpha}_0(i) = \beta(\gamma_1^i - \gamma_2^i)$ $\beta \in \mathbb{R}_+$	$\tilde{\boldsymbol{\alpha}}$	$\mu_0 = \frac{\log\left(\frac{\log \gamma_2}{\log \gamma_1}\right)}{\log\left(\frac{\gamma_2}{\gamma_1}\right)} \in \mathbb{R}$ $\beta = (\gamma_1^{\mu_0} - \gamma_2^{\mu_0})^{-1} \in \mathbb{R}_+$	1, 0, μ_0, 0
Delayed Decay $\tilde{\alpha}_0(i) = (\beta+1)\gamma_1^i - \beta\gamma_2^i$ $\beta \in \mathbb{R}_+$	$\tilde{\boldsymbol{\alpha}}$	$\beta = \frac{\log(\gamma_1)}{\log(\gamma_2) - \log(\gamma_1)}$ for $\beta = 1$ $\begin{cases} \gamma_2 = \gamma_1^2 \\ \mu_0 = \frac{\log(1 \pm 2^{-0.5})}{\log(\gamma_1)} \end{cases}$	1, 0.5, 0, μ_0, 0

of this kind is the detection of multiple ECG shapes as in [1, Section 4.2].

IV. COMPOSITE COSTS

The *cost segment* $J_a^b(k, x, \theta)$ in (20) is nicely computable, but only allows the use of a single model (c, A, x) and a single window satisfying (16). Thus, to circumvent those restrictions, we join several such *cost segments* together and create a more versatile cost, called *composite cost*.

We define $P \in \mathbb{N}$ *cost segments* of respective parameters $\theta_p = \left(c_p, \delta^{(p)}, \boldsymbol{\alpha}^{(p)}\right)$ and boundaries $a_p \in \mathbb{Z} \cup \{-\infty\}$, $b_p \in \mathbb{Z} \cup \{+\infty\}$, $a_p \leq b_p$, $p \in \{1, \ldots, P\}$ leading to the general form of a *composite cost*

$$\tilde{J}(k, x, \boldsymbol{\Theta}) = \sum_{p=1}^{P} v_p J_{a_p}^{b_p}(k, x, \theta_p) \,, \tag{83}$$

with $\boldsymbol{\Theta} = (\theta_1, \theta_2, \ldots, \theta_P)$ and the *segment scalars* $v_p \in \mathbb{R}$. Since each *cost segment* $J_{a_p}^{b_p}(k, x, \theta_p)$ has its recursive computation as in Section III-B with quantities $W_k^{(p)}$, $\xi_k^{(p)}$ and $\kappa_k^{(p)}$, the *composite cost* $\tilde{J}(k, x, \boldsymbol{\Theta})$ is obtained according to (21), using the substitutes

$$W_k = \sum_{p=1}^{P} v_p W_k^{(p)} \tag{84}$$

$$\xi_k = \sum_{p=1}^{P} v_p \xi_k^{(p)} \tag{85}$$

$$\kappa_k = \sum_{p=1}^{P} v_p \kappa_k^{(p)} \,. \tag{86}$$

In addition, we have

$$\nu_k = \sum_{p=1}^{P} v_p \nu_k^{(p)} \tag{87}$$

$$\varphi_k = \sum_{p=1}^{P} v_p \varphi_k^{(p)} \,. \tag{88}$$

Note that definition of the *composite cost* is carefully chosen: while each *cost segment* has its individual observation vector c_p and window $\boldsymbol{\alpha}^{(p)}$, the *state-transition matrix* A, the *state vector* x, and the time index k are in common for all *cost segments*. (This is not a loss in generality, as multiple models are easily stacked (cf. *Model Superposition* in Table VI), and using the introduced index shift δ accordingly shifts each *cost segment* to a common index k.) As an immediate consequence, Table V also minimizes the *composite cost* as in (83). This is proven as follows: we write all *cost segments* in (83) in their parameterized forms as in (21); then we apply the substitutes (84)–(86), leading again to a form as in (21) which is minimized by (69).

A. Instances of Composite Costs

Instances of *composite costs* are applications of the general form (83). A summary of such instances is given in Table VI. Since a *composite cost* is a linear combination of *cost segments*, all linear combinations of *composite costs* are also *composite costs*, which allows us recursive use.

Table VII further highlights particular applications of the instance *window weights superposition* form Table VI, denoting

the linear combination of *window weights* as

$$\tilde{\alpha}_k(i) = \sum_{q=1}^{Q} v_q \alpha^{(q)}_{k+\delta_q}(i)\,. \tag{89}$$

As a side remark, the linear combinations of *window weights* is also valid for windows with adaptive decay if we substitute $\overleftarrow{G}(g)$ by $\overleftarrow{G}(g, w_k)$ and $\overrightarrow{G}(g)$ by $\overrightarrow{G}(g, w_k)$ (cf. Table IV).

Two more involved examples of *composite costs* with direct applications are given in the following subsection.

B. Multi-Time-Scale

In many applications, sensor signals are a superposition of multiple source signals, including interference signals. Most likely, not all sources share the same time scale. Therefore, separation methods should exploit this feature and deal with superimposed models of variable time scale.

Assume we have $M \in \mathbb{N}$ source signal and each source is described by its parameters $(A_m, c_m, x^{(m)})$, $m \in \{1, \dots, M\}$. As we can adjust the models' time scale by adjusting the segment bounds, as well as the *window weights*, we propose two approaches: the first one assigns each model to specific intervals, the second one assigns each model to a specific *window weight* function.

Subsequently, we use the block diagonal matrices $C = \operatorname{diag}(c_1, \dots, c_M)$ and $A = \operatorname{diag}(A_1, \dots, A_M)$, and the vector $x^{\mathsf{T}} = \left[(x^{(1)})^{\mathsf{T}} \quad \cdots \quad (x^{(M)})^{\mathsf{T}} \right]$.

1) Time-Scaled by Interval Limits: Each model being active in specific intervals, we rearrange the time axis into $P \in \mathbb{N}$ non-overlapping segments, such that the assignment matrix $\Lambda \in \{0,1\}^{P \times M}$ is well defined and $\Lambda_{m,p} = 1$, indicates that model "m" is active in segment "p".

Each *cost segment* is characterized by its lower and upper bounds $a_p \in \mathbb{Z}$ and $b_p \in \mathbb{Z}$, $a_p \le b_p$, $b_p < a_{p+1}$, and with the associated *window weights* $\boldsymbol{\alpha}^{(p)}$. We define $\theta_p = \left(\Lambda_p C, \delta_p, \boldsymbol{\alpha}^{(p)}\right)$ with $\Lambda_p \in \mathbb{R}^{1 \times M}$ the pth row of Λ, such that all models active in the pth segment are selected. Hence, we get a *composite cost*

$$\tilde{J}(k, x, \boldsymbol{\Theta}) = \sum_{p=1}^{P} J^{b_p}_{a_p}(k, x, \theta_p)\,, \tag{90}$$

as an instance of (83) with $\boldsymbol{\Theta} = (\theta_1, \dots, \theta_P)$. Note that the very left and the very right segment might be of infinite length ($a_1 = -\infty$ or $b_P = \infty$).

2) Time-Scaled by Window Weights: For each linear state space model of parameters $(c_m, A_m, x^{(m)})$ we associate its own *window weights* $\alpha^{(m)}_k(i)$, for $m \in \{1, \dots, M\}$. We are interested in computing and minimizing the following cost function

$$J^b_a(k, x, \delta) = \sum_{i=k+a}^{k+b} w_i \sum_{m=1}^{M} \left[\left(\alpha^{(m)}_{k+\delta}(i) - \alpha^{(m-1)}_{k+\delta}(i) \right) \cdot \left(y_i - \sum_{j=m}^{M} c_m (A_m)^{i-k} x^{(m)} \right)^2 \right], \tag{91}$$

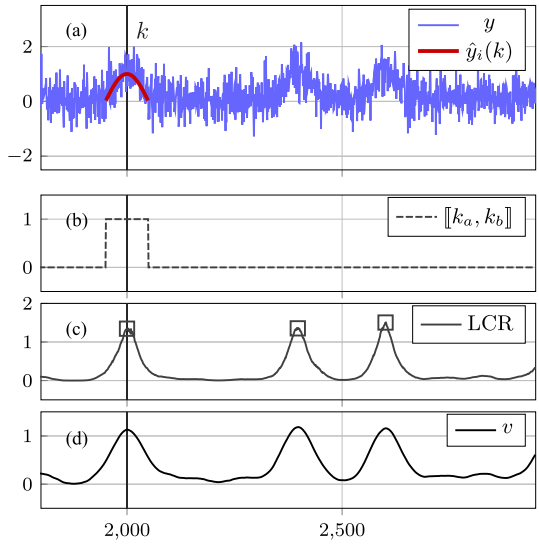


Fig. 5. Detection of cosine pulses in noisy sensor signal as in Example V-A. (a) Sensor signal y with example cosine pulse estimate $\hat{y}_i(k)$ as in (95) evaluated for highlighted time index $k = 2000$. (b) Rectangular window with interval limits $k + a$ and $k + b$, denoted as $[\![k_a, k_b]\!]$, also evaluated for highlighted time index k. (c) LCR as in (93) with detected peaks (rectangular marks). (d) Estimate of scaling factor v_k as in (94).

with $\alpha^{(0)}_{k+\delta}(i) = 0$, for all k and i. This cost form is inspired by thinking of $\alpha^{(m)}_k(i)$ as rectangular windows, revealing the analogy to Section IV-B1 above. Then, defining $T_m = [\underbrace{1 \cdots 1}_{m}\ 0 \cdots 0]$, the cost (91) can also be written as

$$J^b_a(k, x, \delta) = \sum_{i=k+a}^{k+b} w_i \sum_{m=1}^{M} \left[\left(\alpha^{(m)}_{k+\delta}(i) - \alpha^{(m-1)}_{k+\delta}(i) \right) \cdot \left(y_i - T_m C A^{i-k} x \right)^2 \right]. \tag{92}$$

Note that all the models are active over the same (finite or infinite) interval, but with different *window weight* sequences. As a second side remark, defining T to successively activate the models, i.e., to an upper triangular matrices as proposed, is a reasonable selection for many applications but not compulsory.

With $\theta_{m,\psi} = \left(T_m C,\ \delta, \boldsymbol{\alpha}^{(m-\psi)}\right)$, $\psi \in \{0,1\}$, we get a *composite cost* as in (83) with $\boldsymbol{\Theta} = (\theta_{1,0}, \theta_{1,1}, \theta_{2,0}, \theta_{2,1}, \dots, \theta_{M,0}, \theta_{M,1})$. Finally, the *composite cost* $\tilde{J}^b_a(k, x, \boldsymbol{\Theta})$ is easily computed and minimized as in Section IV. In practice, a common shift δ for all *window weights* $m \in \{1, \dots, M\}$, as proposed in this approach, is sufficient. Nevertheless, the generalization from δ to δ_m is straightforward.

V. Examples

A. Pulse Detection

Problem: In this example, we observe, after sampling at sampling rate f_s, a noisy sensor signal $y_k \in \mathbb{R}$ for $k \ge 1$, containing non-overlapping finite-duration cosine pulses $u_j = \cos(\Omega j)$, $j \in \mathbb{Z}$, $-\frac{\pi}{\Omega} \le j \le \frac{\pi}{\Omega}$, $\Omega = 2\pi \frac{f_0}{f_s}$, of pulse frequency f_0. We want do detect these non-overlapping, repetitive pulses and estimate their amplitude (cf. Fig. 5).

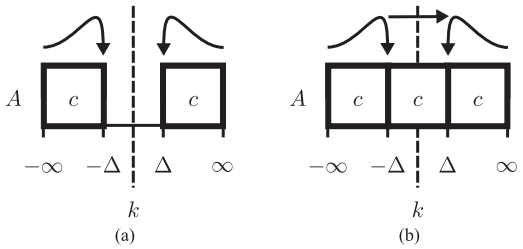


Fig. 6. Two *composite costs* providing the LCR (96) in Example V-B. (a) *Composite cost* $\tilde{J}_0$ consisting of two *cost segments* fitting to the baseline signal to the left or the right of an assumed interjacent pulse of unknown shape. (b) *Composite cost* $\tilde{J}_1$ consisting of three consecutive *cost segments* fitting to a continuous baseline signal (i.e., no interjacent pulse assumed).

Solution: We use a *cosine model* with A and c according to Table I with $\rho = 1$ and a single *finite support cost segment* $J_a^b(x, k, \delta)$ with $a = -\lceil \pi/\Omega \rceil, b = \lfloor \pi/\Omega \rfloor$ and $\delta = 0$ according to Table II, and an exponential *window weight* $\alpha_k(i)$ with a sufficiently large g, leading to nearly rectangular window shape with guaranteed computational stability. We use forward computation starting according to Section III-B at $k = -b$ and proceed using (43)–(46). Furthermore, we localize the pulses by performing peak detection on the *log-cost ratio*

$$\mathrm{LCR}_k = -\frac{1}{2} \log \frac{\min\limits_v J_a^b(k, Hv, \theta)}{J_a^b(k, 0, \theta)} , \tag{93}$$

using the shape amplitude estimate $v \in \mathbb{R}$ (linear scaling)

$$v_k = \operatorname*{argmin}_v J_a^b(k, Hv, \theta) , \tag{94}$$

with $H = [1, 0]^{\mathsf{T}}$. The signal estimate obtained at index k is then given by

$$\hat{y}_i(k) = s_{i-k}(Hv_k) = cA^{i-k} Hv_k . \tag{95}$$

B. Separation of Unknown Pulses From Baseline Signal

Problem: We want to estimate pulses of known width 2Δ, $\Delta \in \mathbb{N}_+$ but unknown pulse shape from noisy observations $y_i = \overline{y}_i + z_i$, $i \geq 1$ (with noise-free signal $\overline{y}_i \in \mathbb{R}$ and additive white Gaussian noise $z_i \in \mathbb{R}$).

Solution: We choose an approach with 2 processing steps. In a first step (I), we estimate the positions of the pulses. In the second step (II), we extract the interference signal.

We use a *composite cost* of *Non-Overlapping Segments* (cf. Section IV-A) with a single model and three segments. The model of parameters (A, c) is a cubic polynomial for fitting the interferences (cf. Table II). We define the segments as:

segm.	a	$b+1$	δ	*window* (cf. Table VII)
1	$-\infty$	$-\Delta$	0	*Pulse* $\begin{cases} \gamma_{1,i} = \overleftarrow{G}(0.7 f_s, w_i) \\ \gamma_{2,i} = \overleftarrow{G}(0.1 f_s, w_i) \end{cases}$
2	$-\Delta$	$+\Delta$	0	*Rectangular* $\gamma_1 = 1$
3	$+\Delta$	$+\infty$	0	*Pulse* $\begin{cases} \gamma_{1,i} = \overrightarrow{G}(0.7 f_s, w_i) \\ \gamma_{2,i} = \overrightarrow{G}(0.1 f_s, w_i) \end{cases}$

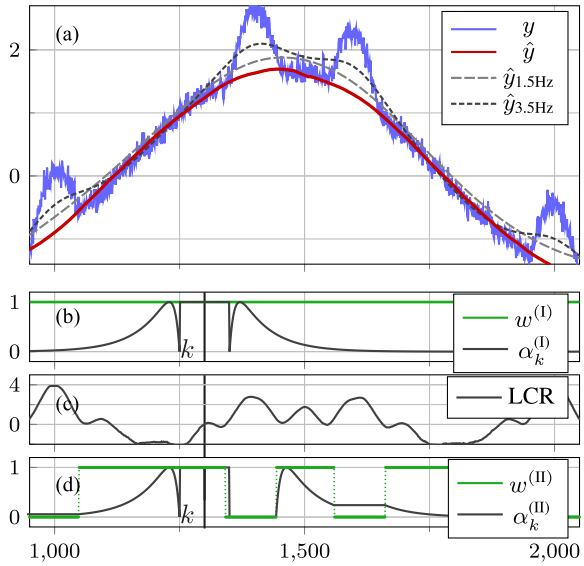


Fig. 7. Estimation of baseline signal $\hat{y}$ superimposed on a noisy signal of non-symmetric pulses as in Example V-B. (a) Observed signal y and estimated interference after applying 2nd processing step ($\hat{y}_i = \hat{y}_i^{(\mathrm{II})} = cx_i^{(\mathrm{II})}$). For comparison, baseline estimates $\hat{y}_{1.5\,\mathrm{Hz}}$ and $\hat{y}_{3.5\,\mathrm{Hz}}$ using an IIR filter (2nd order high-pass butterworth filter, run forward and backward with corner frequencies 1.5 Hz and 3.5 Hz, respectively) are additionally displayed. (b) *Sample weights* $w^{(\mathrm{I})}$ and *window weights* $\alpha_k^{(\mathrm{I})}$ (at time index $k = 1300$) of first processing step. (c) LCR as in (96) used for pulse detection in the 1th processing step. (d) *Sample weights* $w^{(\mathrm{II})}$ and adaptive *window weights* $\alpha_k^{(\mathrm{II})}$ (at time index $k = 1300$) of second step.

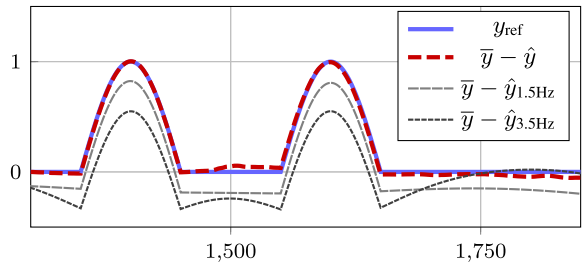


Fig. 8. Pulses extracted using the method as proposed in Example V-B working on weighted samples, in comparison to a high-pass IIR filter (2nd order high-pass butterworth filter, run forward and backward with corner frequencies 1.5 Hz and 3.5 Hz, respectively). Signal y_{ref} is the noise-free and interference-free reference pulse signal, $\hat{y}_i = \hat{y}_i^{(\mathrm{II})} = cx_i^{(\mathrm{II})}$ the estimate from our method using a polynomial interference model, and $\hat{y}_{1.5\,\mathrm{Hz}}$ and $\hat{y}_{3.5\,\mathrm{Hz}}$ the IIR filtered signals displayed for comparison.

Segment 1 and 2 are computed forward and segment 3 backward. Note that the *cost segments* 1 and 3 use a window of adaptive decay (Table IV).

For processing step (I), we set the *sample weight* of that step to 1, i.e., $w_i^{(\mathrm{I})} = 1$, $i \geq 1$. Furthermore, we denote θ_p, $p \in \{1, 2, 3\}$, the parameters of the pth *cost segment*, and define the parameter sets $\boldsymbol{\Theta}_0 = (\theta_1, \theta_3)$ for the *composite cost* $\tilde{J}_0^{(\mathrm{I})}$ and $\boldsymbol{\Theta}_1 = (\theta_1, \theta_2, \theta_3)$ for $\tilde{J}_1^{(\mathrm{I})}$ (cf. Fig. 6(a) and Fig. 6(b)). Then, we get the *state vector* $x^{(\mathrm{I})}$ by minimizing $\tilde{J}_0^{(\mathrm{I})}(k, x, \boldsymbol{\Theta}_0)$ over x unconstrained.

To finally estimate the positions of the qth pulse, denoted as i_q, $q \in \mathbb{N}$, in the first step, we use a peak detection on the

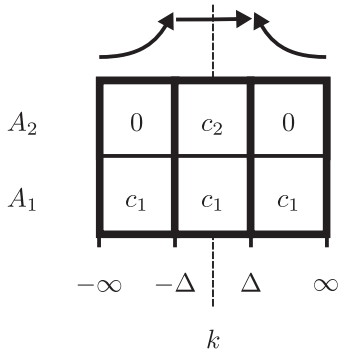


Fig. 9. Single *composite cost* $\tilde{J}^{(\mathrm{II})}$ consisting of three consecutive *cost segments* of baseline model (A_1, c_1) in the lower row and a single *cost segment* of a signal model (A_2, c_2) in the upper row, as in Example V-C.

log-cost ratio

$$\mathrm{LCR}_k = -\frac{1}{2}\log \frac{\frac{1}{\nu_0}\tilde{J}_0^{(\mathrm{I})}(k, x^{(\mathrm{I})}, \boldsymbol{\Theta}_0)}{\frac{1}{\nu_1}\tilde{J}_1^{(\mathrm{I})}(k, x^{(\mathrm{I})}, \boldsymbol{\Theta}_1)}\,. \tag{96}$$

For processing step (II), we set the *sample weight* of that step such that $w_i^{(\mathrm{II})} = 0$ for $|i - i_q| \le \Delta$ and $w_i^{(\mathrm{II})} = 1$ otherwise. Then, we compute $\tilde{J}_1^{(\mathrm{II})}(k, x, \boldsymbol{\Theta}_1)$ and get $x^{(\mathrm{II})}$ by unconstrained minimization. The interference estimate is then $\hat{y}_i = \hat{y}_i^{(\mathrm{II})} = cx_i^{(\mathrm{II})}$ and thus, the pulse signals (with Gaussian noise) is $y_i - \hat{y}_i$.

To measure the performance, we compare our method with an IIR filter (2nd order high-pass butterworth filter, run forward and backward). Selecting corner frequencies of IIR filters always leads to a trade-off between interference fit and signal deformation. Fig. 7 shows an overview of the two processing steps of our algorithm, and Fig. 8 a detailed view comparing our method to standard IIR filters.

C. ECG Signal Baseline Recovery

Problem: ECG signals are commonly superimposed with multiple artifacts, e.g., caused by motion and muscular artifacts. However, for many medical applications and diagnosis, a proper recovery of the baseline of an electrocardiogram (ECG) signal is essential. For example, the elevation of the ST-segment (signal level in between the QRS and the T-wave) is an important criteria for a myocardial infarction, so called ST elevation myocardial infarction (STEMI). To remove the baseline signal with only minimal alterations of the ECG signal, high-pass filters with a corner frequency of 0.05 Hz are commonly used [26]. But ambulatory ECG devices often show signals with strong baseline artifacts, requiring high-pass filters with corner frequency 0.5 Hz, which, in turn, prohibits a valid ST-segment evaluation. In the following, we want to extract the baseline of such an ambulatory ECG signal using our proposed method, while preserving the ECG signal shape as good as possible. We compare the result with the output of zero-phase IIR filters, i.e., IIR filters applied in forward and reverse direction.

Solution: Let y_k, $k \ge 1$ be a sequence of an ECG signal superimposed with strong baseline interferences. We again chose

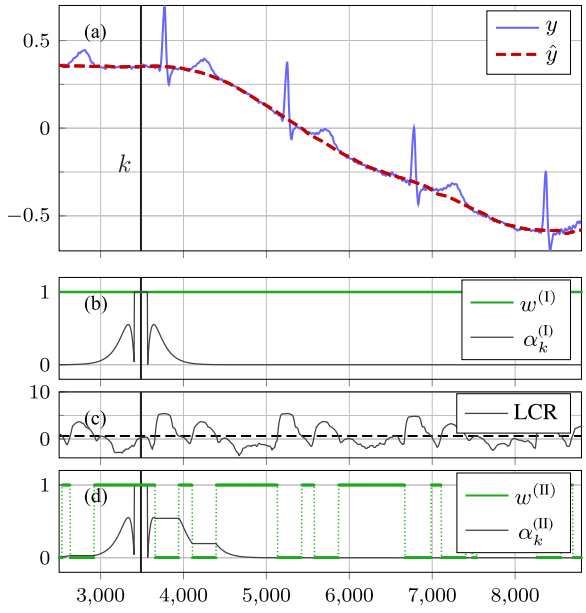


Fig. 10. Baseline extraction of ECG signal y as in Example V-C. (a) Observed ECG signal y and baseline estimate of the second processing step $\hat{y}_i = \hat{y}_i^{(\mathrm{II})} = cx_i^{(\mathrm{II})}$. (b) *Sample weights* $w^{(\mathrm{I})}$ and *window weights* $\alpha_k^{(\mathrm{I})}$ of first step. (c) LCR (solid line) with detection threshold (dashed line) of the first step. (d) *Sample weights* $w^{(\mathrm{II})}$ and adaptive *window weights* $\alpha_k^{(\mathrm{II})}$ of second step.

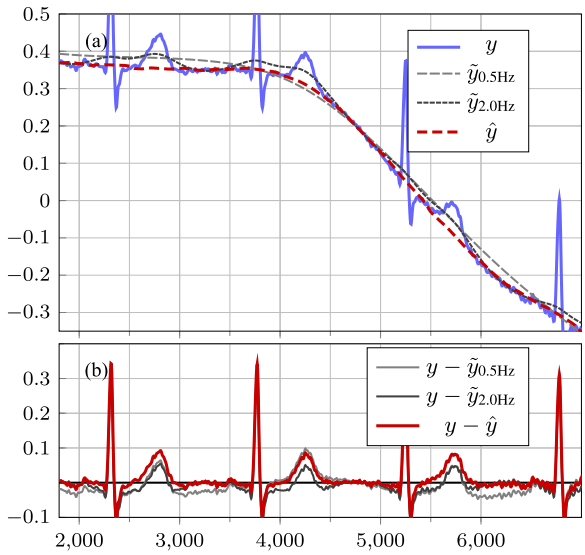


Fig. 11. Comparison of our method with a two-sided high-pass butterworth IIR filter of 2nd order, as in Example V-C. (a) IIR filtered signal $\tilde{y}_{0.5\,\mathrm{Hz}}$ and $\tilde{y}_{2.0\,\mathrm{Hz}}$ with filter corner frequencies at 0.5 Hz and 2.0 Hz, signal estimates from the polynomial interference model $\hat{y}_i = \hat{y}_i^{(\mathrm{II})} = cx_i^{(\mathrm{II})}$. (b) Comparison of the baseline-reduced ECG signals.

an approach with two processing steps. In the first step (I), we extract the positions of cardiac activity using a cubic polynomial model (A_1, c_1) with three segments and parameter Δ as in Example V-B. We again compute the LCR and set the *sample weights* $w_i^{(I)} = 0$ for any LCR value exceeding a fixed threshold. To improve the detection of close-by cardiac activities, we repeat the process of step (I) once more, using an updated $w_i^{(I)}$.

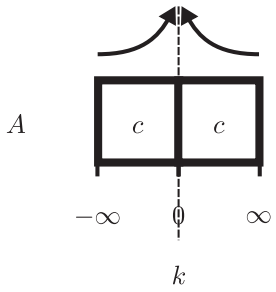


Fig. 12. Single *composite cost* $\tilde{J}$ consisting of two consecutive *cost segments* of single model (A, c), forming a continuous polynomial model with a two-sided exponentially decaying window, as in Example V-D.

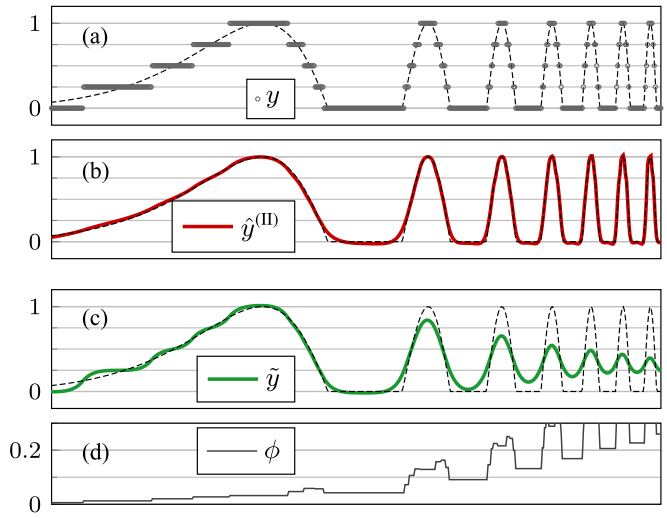


Fig. 13. Reconstruction of a test signal from its quantized version controlled by the speed of changes in the signal as in Example V-D. (a) (Non-quantized) Reference signal (dashed line) and corresponding quantized signal y. (b) Adaptive signal estimate $\hat{y}_k = \hat{y}_k^{(\mathrm{II})} = c\hat{x}_k^{(\mathrm{II})}$ reconstructed from quantized signal y, using adaptive window decay controlled by the *relative sample density* ϕ. (c) Non-adaptive signal estimate $\tilde{y}$ reconstructed from quantized signal y, using a constant window decay. (d) *Relative sample density* ϕ.

For the second step (II), we use two models in three segments, as shown in Fig. 9, resulting in a *composite cost* $\tilde{J}^{(\mathrm{II})}$. While model (A_1, c_1) still corresponds to the interference signal, model (A_2, c_2), also a cubic polynomial, smooths the signal locally over a rectangular window of 2Δ samples.

The resulting interference estimate $\hat{y}_i = \hat{y}_i^{(\mathrm{II})} = cx_i^{(\mathrm{II})}$ shows a clear improvement of the fit to the signal baseline, in comparison to IIR filters. Selecting the corner frequencies for IIR filters leads to a trade-off between baseline fit and signal deformation. A comparison is given in Fig. 10 with detailed view in Fig. 11.

D. Reconstruction of a Signal From Its Quantized Version

Problem: Let $y_k \in \{\ldots, -2\Delta, -\Delta, 0, \Delta, 2\Delta, \ldots\}$ be a given quantized signal for $k \geq 1$, quantized with the step size $\Delta \in \mathbb{R}_+^*$. We wish to reconstruct the unquantized signal, as demonstrated in Fig. 13 for a test signal, and in Fig. 14 for a real ECG signal.

Note that this problem setup is directly related to the interpolation of non-uniformly sampled data.

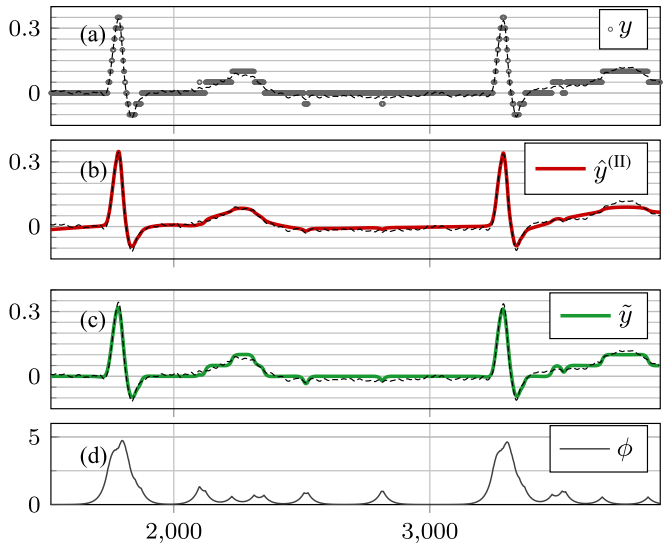


Fig. 14. Reconstruction of an ECG signal from its quantized version controlled by the speed of changes in the signal as in Example V-D. The horizontal grid corresponds to quantization levels. (a) (Non-quantized) Reference ECG signal (dashed line) and corresponding quantized signal y. (b) Adaptive signal estimate $\hat{y}_k = \hat{y}_k^{(\mathrm{II})} = c\hat{x}_k^{(\mathrm{II})}$ reconstructed from quantized signal y, using adaptive window decay controlled by the *relative sample density* ϕ. (c) Non-adaptive signal estimate $\tilde{y}$ reconstructed from quantized signal y, using a constant window decay. (d) Used *relative sample density* ϕ.

Solution: We process the signal in two steps. In step (I), we only estimate a measure for the speed of changes in the observed signal. Therefor we compute the *sample density* ϕ_k for all k as in Section III-D using the auxiliary observation signal $y'_k = y_k - y_{k-1}$ and applying a two-sided exponential window with a *effective sample number* $g^{(\mathrm{I})}$.

In step (II), we assign the *sample density* to the *sample weights*, i.e., $w_k^{(\mathrm{II})} = \phi_k$ and introduce a *composite cost* using a polynomial model of order 3 with a two-sided exponential window (Fig. 12) with $g^{(\mathrm{II})}$. Finally, we do an unconstrained minimization (70) on this *composite cost* and get the desired signal estimate $\hat{y}_k = \hat{y}_k^{(\mathrm{II})} = c\hat{x}_k^{(\mathrm{II})}$. Note, that our adaptive estimate $\hat{y}_k$, controlled by the *sample density*, fits equally over all the different sections of the signal. By contrast, an estimate with fixed smoothing, labeled $\tilde{y}$, only fits nicely at sections of unique time scale, but fails otherwise (resulting in a "quantization pattern" on lower frequencies and "over-smoothing" on higher frequencies) (Fig. 13(c)). In particular for the ECG signal, $\hat{y}$ differs from $\tilde{y}$ mainly in two ways: on one hand $\tilde{y}$ does not reach the full peak amplitude at the sharp peaks (QRS wave) and, on the other hand, $\tilde{y}$ shows the "quantization pattern" in slow parts, such as e.g., the T waves (Fig. 14). Further note, that the proposed method is not sensitive to any amplitude scaling.

E. Morse Code Detection

Problem: We want do detect the symbol 'L' from the Morse alphabet, i.e., $(\bullet - \bullet\bullet)$, from a single channel sensor data $y_i \in \mathbb{R}$ for $i \geq 1$. For the transmission, we assume an on-off modulated oscillator running at constant frequency f_0, generating a pulse-to-pause ratio of 1:2 for a dot, and a ratio of 1:4 for a dash (cf. Fig. 16).

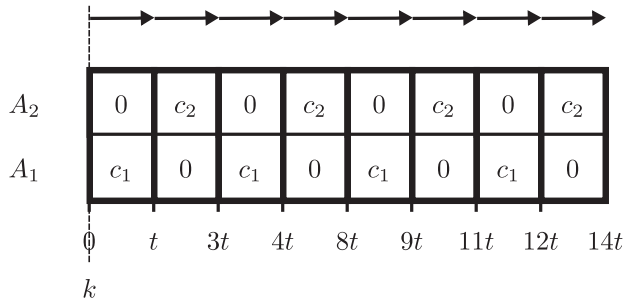


Fig. 15. Single *composite cost* $\tilde{J}$ consisting of a total of eight *cost segments* assigned to the pulse model (A_1, c_1) in the lower row or to the pause model (A_2, c_2) in the upper row, as in Example V-E.

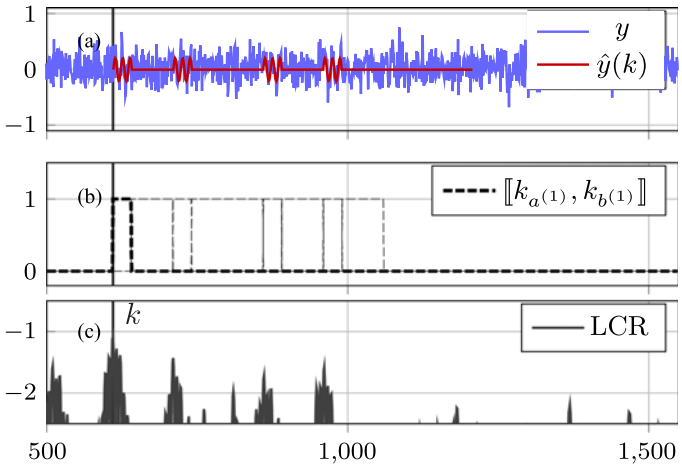


Fig. 16. Detection of Morse code symbol with additive white Gaussian noise as in Example V-E. (a) Sensor signal with highlighted estimation of found Morse code trajectory $\hat{y}_i(k) = s_{i-k}(Hv_k)$ at index k. (b) Highlighted interval limits of first *cost segment*, denoted as $[\![k_{a^{(1)}}, k_{b^{(1)}}]\!]$ (black dashed line) and outline of remaining *cost segments* intervals limits (gray dashed lines). (c) *Log-cost ratio*.

Solution: Let $Q = 4$ be the number of Morse dots or dashes in the symbol to detect and let $t \in \mathbb{N}$ be the discrete duration of a dot. We use a *composite cost* of *non-overlapping segments* (cf. Table VI) with two models and $2Q$ segments (Fig. 15). For the oscillator, we use a cosine model with A_1 and c_1 (cf. Table VI). For the pauses, we use a constant model of parameters $A_2 = [1]$ and $c_2 = [1]$. Then, the *composite cost* is parametrized as $A = \mathrm{diag}\,(A_1, A_2)$, $C = \mathrm{diag}\,(c_1, c_2)$, $\Lambda = [I_2, \ldots, I_2] \in \mathbb{R}^{2\times 2Q}$ and the segments are such that:

Segm.	a	$b+1$	δ	*Window* (cf. Table VII)
1	0	t	0	
2	t	$3t$	0	
3	$3t$	$4t$	0	
4	$4t$	$8t$	0	*Exponential*
5	$8t$	$9t$	0	$\gamma_1 = \overleftarrow{G}(10^5)$
6	$9t$	$11t$	0	
7	$11t$	$12t$	0	
8	$12t$	$14t$	0	

We compute all segments in forward direction. Finally, assuming a fixed and known phase of the oscillator when the code start, we constrain the state such that $x = Hv$ with $H = [1, 0, 0]^{\mathsf{T}}$ and minimize over $v \in \mathbb{R}$ using (71). (In case of an unknown initial phase, we use (70) with $H = [I_2, 0_{2\times 1}]^{\mathsf{T}}$ and optimize over $v \in \mathbb{R}^2$ instead).

To decide the most likely symbol among multiple, we compute the LCR for each symbol.

VI. Conclusion

We have proposed a versatile toolbox for detection, separation, and reconstruction of signals that was developed especially for the analysis of ECGs and similar signals. Using autonomous linear state space models leads to efficient recursive computations. Multi-section windows enable complex models, and per-sample weights enable multi-stage processing and adaptive smoothing. The proposed methods have been successfully applied to analyze the data of a clinical trial in human cardiology, and they have been used over a long period of time to process ECGs and other biomedical signals.

The main limitations of the proposed method are as follows. First, complex models may need large matrices A, leading to high computational costs (as discussed in Section III-H). Second, the proposed method cannot handle significant superpositions of independent pulses. This latter limitation is overcome by the (more complex) method of [27] and [16, Part III].

We also mention that a generalization of the proposed approach from squared errors to polynomial cost functions was proposed in [25].

Acknowledgment

The authors would like to thank Prof. Dr. H. Tanner and Dr. A. Haeberlin of the Department of Cardiology at the Bern University Hospital for the valuable collaboration.

References

[1] R. A. Wildhaber, N. Zalmai, M. Jacomet, and H. A. Loeliger, "Signal detection and discrimination for medical devices using windowed state space filters," in *Proc. 13th IASTED Int. Conf. Biomed. Eng.*, Feb. 2017, pp. 125–133.

[2] J. G. Proakis, *Digital Communications*, 5th ed. New York, NY, USA: McGraw-Hill, 2008.

[3] B. U. Kohler, C. Hennig, and R. Orglmeister, "The principles of software QRS detection," *IEEE Eng. Med. Biol. Mag.*, vol. 21, no. 1, pp. 42–57, Jan. 2002.

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

[5] S. Mallat and W. L. Hwang, "Singularity detection and processing with wavelets," *IEEE Trans. Inf. Theory*, vol. 38, no. 2, pp. 617–643, Mar. 1992.

[6] V. X. Afonso, W. J. Tompkins, T. Q. Nguyen, and S. Luo, "ECG beat detection using filter banks," *IEEE Trans. Biomed. Eng.*, vol. 46, no. 2, pp. 192–202, Feb. 1999.

[7] A. Logothetis and V. Krishnamurthy, "Expectation maximization algorithms for MAP estimation of jump Markov linear systems," *IEEE Trans. Signal Process.*, vol. 47, no. 8, pp. 2139–2156, Aug. 1999.

[8] A. Doucet and C. Andrieu, "Iterative algorithms for state estimation of jump Markov linear systems," *IEEE Trans. Signal Process.*, vol. 49, no. 6, pp. 1216–1227, Jun. 2001.

[9] E. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky, "Bayesian nonparametric inference of switching dynamic linear models," *IEEE Trans. Signal Process.*, vol. 59, no. 4, pp. 1569–1585, Apr. 2011.
[10] H. A. Loeliger, L. Bruderer, H. Malmberg, F. Wadehn, and N. Zalmai, "On sparsity by NUV-EM, Gaussian message passing, and Kalman smoothing," in *Proc. Inf. Theory Appl. Workshop*, Jan. 2016, pp. 1–10.
[11] 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 *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2016, pp. 3331–3335.
[12] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.
[13] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. Basic Eng.*, vol. 82, no. 1, pp. 35–45, Mar. 1960.
[14] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*. Englewood Cliffs, NJ, USA: Prentice-Hall, 2000.
[15] H.-A. Loeliger, J. Dauwels, J. Hu, S. Korl, L. Ping, and F. R. Kschischang, "The factor graph approach to model-based signal processing," *Proc. IEEE*, vol. 95, no. 6, pp. 1295–1322, Jun. 2007.
[16] N. Zalmai, "A state space world for detecting and estimating events and learning sparse signal decompositions," Ph.D. dissertation, Dept. Inf. Technol. Elect. Eng., ETH Zurich, Zurich, Switzerland, 2017.
[17] D. Lee, M. Morf, and B. Friedlander, "Recursive least squares ladder estimation algorithms," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 29, no. 3, pp. 627–641, Jun. 1981.
[18] S. S. Haykin, *Adaptive Filter Theory*, 5th ed. London, U.K.: Pearson, 2014.
[19] R. W. Schafer, "What is a Savitzky–Golay filter?" *IEEE Signal Process. Mag.*, vol. 28, no. 4, pp. 111–117, Jul. 2011.
[20] A. Savitzky and M. J. E. Golay, "Smoothing and differentiation of data by simplified least squares procedures," *Anal. Chem.*, vol. 36, no. 8, pp. 1627–1639, 1964.
[21] P. Buxbaum, "Fixed-memory recursive filters," *IEEE Trans. Inf. Theory*, vol. IT-20, no. 1, pp. 113–115, Jan. 1974.
[22] G. Bierman, "Fixed memory least squares filtering," *IEEE Trans. Inf. Theory*, vol. IT-21, no. 6, pp. 690–692, Nov. 1975.
[23] A. Bruckstein and T. Kailath, "Recursive limited memory filtering and scattering theory," *IEEE Trans. Inf. Theory*, vol. IT-31, no. 3, pp. 440–443, May 1985.
[24] L. Bruderer, H.-A. Loeliger, and N. Zalmai, "Local statistical models from deterministic state space models, likelihood filtering, and local typicality," in *Proc. IEEE Int. Symp. Inf. Theory*, 2014, pp. 1106–1110.
[25] N. Zalmai, R. A. Wildhaber, and H. A. Loeliger, "Autonomous state space models for recursive signal estimation beyond least squares," in *Proc. 25th Eur. Signal Process. Conf.*, Aug. 2017, pp. 341–345.
[26] R. Abächerli and H.-J. Schmid, "Meet the challenge of high-pass filter and st-segment requirements with a dc-coupled digital electrocardiogram amplifier," *J. Electrocardiol.*, vol. 42, no. 6, pp. 574–579, 2009.
[27] N. Zalmai, R. Keusch, H. Malmberg, and H.-A. Loeliger, "Unsupervised feature extraction, signal labeling, and blind signal separation in a state space world," in *Proc. 25th Eur. Signal Process. Conf.*, 2017, pp. 838–842.

Reto A. Wildhaber received the degree in electrical engineering from the University of Applied Science, Rapperswil, Switzerland, in 2003, the M.Sc. degree in medicine and the M.D. degree both from the University of Zurich, Zurich, Switzerland, in 2013 and 2014, respectively. He is currently working toward the Ph.D. degree at the Department of Information Technology and Electrical Engineering, ETH Zurich, Zurich, Switzerland. He is employed with Bern University of Applied Science and collaborates with the University Hospital Bern.

Nour Zalmai received the Diploma in electrical engineering from Supélec, Gif-sur-Yvette, France, in 2013, the M.Sc. and Ph.D. degrees in electrical engineering both from ETH Zurich, Zurich, Switzerland, in 2013 and 2017, respectively.

His research interests include signal processing and graphical models for learning sparse signal decompositions.

Marcel Jacomet received the M.Sc. and Dr. sc. techn. degrees in electronic engineering from the Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, in 1983 and 1990, respectively. Before joining the BFH in 1992, he was responsible for the introduction of new technologies with Muller-Martini Electronics Inc, Zofingen, Switzerland.

He is a Professor for microelectronics with the Bern University of Applied Sciences (BFH), Bern, Switzerland and currently the Head of the BFH Research Institute Human Centered Engineering HuCE. His research interests include numerous projects in the fields of Fuzzy Logic, RFID, biometric recognition algorithms, chip design, and hardware algorithms with focus on medical applications. He and his team were the recipient of the ETH/McKinsey price as well as twice with the Swiss Technology Award.

Hans-Andrea Loeliger (S'85–M'92–SM'03–F'04) received a Diploma in electrical engineering and the Ph.D. degree from ETH Zurich, Zurich, Switzerland, in 1992. He was a Professor with the Department of Information Technology and Electrical Engineering of ETH Zurich. From 1992 to 1995, he was with Linköping University, Linköping, Sweden. From 1995 to 2000, he was a full-time Technical Consultant and Co-owner of a consulting company, from where he returned to ETH in 2000. His research interests include the broad areas of signal processing, machine learning, information theory, error correcting codes, communications, and electronic circuits.