# PIECEWISE-POLYNOMIAL SIGNAL SEGMENTATION USING CONVEX OPTIMIZATION

Pavel Rajmic, Michaela Novosadová and Marie Daňková

A method is presented for segmenting one-dimensional signal whose independent segments are modeled as polynomials, and which is corrupted by additive noise. The method is based on sparse modeling, the main part is formulated as a convex optimization problem and is solved by a proximal splitting algorithm. We perform experiments on simulated and real data and show that the method is capable of reliably finding breakpoints in the signal, but requires careful tuning of the regularization parameters and internal parameters. Finally, potential extensions are discussed.

*Keywords:* signal segmentation, denoising, sparsity, piecewise-polynomial signal model, convex optimization

*Classification:* 46N10, 47N10, 65K10, 90C25, 90C30, 90C90

## 1. INTRODUCTION

Segmentation of a signal is one of the most important applications in digital signal processing. In practice, segmentation is largely used in image processing, but segmentation of one-, three- and even more-dimensional signals is an important task as well.

In the article, convex optimization is used to perform the segmentation of an one-dimensional signal, i. e. to automatically find the partitioning of such a signal into its segments, while the particular segments exhibit consistent character. In our case, the clean, underlying signal is assumed to break into several polynomial segments, non-overlapping and independent of one another, whose number is assumed to be considerably lower than the total number of the observed signal samples. This last fact suggests utilizing sparse signal processing techniques.

A number of real-world time series can be considered piecewise-polynomial, and thus this method is of practical interest as presented below. Nevertheless, this paper can also be seen as an intermediate step towards segmentation of images or volumes. Indeed, the methods which will be presented can be smoothly generalized to a higher dimension, where the piecewise-polynomial structure of segments is also present; for example, [33, 21] showed that images can, to a great extent, be modeled as piecewise-smooth 2D functions.

DOI: 10.14736/kyb-2017-6-1131

The recent publication [14], coping with segmentation of images, was actually the motivation for this work. The authors of [14] use greedy [36] approach enabling state-of-the-art results, and our goal was to explore to what extent the convex relaxations of sparsity measures [11] will compare with it.

Polynomial smoothing of time series has commonly been done by the so-called moving least-squares methods. Such methods consist in selecting a moving window and computing the respective windowed (i. e. local) least-squares polynomial approximations. The resulting curve is smooth, and approximates well if the underlying noiseless function is smooth [22]. However, such methods are not applicable to our setup since discontinuities are allowed.

Finding consistent segments in a signal/time series is an old problem, referred to as the change-point problem in the field of statistics. Most resources in the literature cope with finding change points in a noisy, piecewise-*constant* signal, which is either singlevariate [23] or multivariate [2, 13] (the latter is the case where most of the signals or all of them change simultaneously). In a more general context of detecting sudden changes in the underlying, latent *parameters* we mention works [37] and [1]. Article [31] utilizes the sparse techniques, but it considers the polynomial model for the *entire* signal, with additive sparse ramp discontinuities.

When the signal is assumed to lack jumps (i. e. the segments' borderpoints have to coincide), efficient methods have been presented, based on sparse modeling, see [17] or [35] in a more general setup.

Although our segmentation problem relates to the above-mentioned change-point problems, it is different, as will be revealed shortly in detail. To the best of our knowledge, there is no other work apart from [14] which would use *over-parametrization* for segmenting the signal, and apart from [32] which, however, constructs non-convex optimization problems.

**Our previous work.** We have already made some progress in the proposed direction. Our publication [30] discussed the use of convex total variation (TV) regularizers for the segmentation task. Then, a recent conference paper [24] extended the former basic approach in several directions, involing the shift to *group*-sparsity, for example. This article naturally follows up on [24] and provides, among many small improvements: (1) a deeper explanation of the model and the algorithms, (2) improved numerical stability, (3) better assesment in the changepoint detection phase, (4) an experiment on real data, (5) a software package.

**Organization of the paper.** Section 2 formulates the segmentation/denoising problem in the mathematical perspective and explains the resulting optimization program. Section 3 presents the proximal splitting algorithms and explains in detail how the Chambolle-Pock algorithm is utilized to find a solution to our problem. In addition, this section introduces post-processing steps needed to establish the segment breakpoints and to find the final signal estimate. The experiments on synthetic and real data are presented in Section 4. Section 5 describes the software used for the experiments, following the idea of reproducible research. A thorough discussion on the strengths and weaknesses of the proposed convex program is presented in Section 6. It also discusses possible modification of this problem that would render better and more stable results.

## 2. PROBLEM FORMULATION

Let the signal $\mathbf{y} = [y_1, \ldots, y_N]^\top \in \mathbb{R}^N$ be given, which is assumed to be piecewise-polynomial. The consecutive segments do not have to “meet” at the segment borders, i. e. jumps in $\mathbf{y}$ are allowed. Suppose there are $S$ segments. Within a particular segment $s \in \{1, \ldots, S\}$, elements of the vector $\mathbf{y}$ can be described by the $K+1$ parametrization coefficients $x_{s,0}, x_{s,1}, \ldots, x_{s,K}$ that represent the local intercept, the local slope etc., respectively. The $i$th element of observed $\mathbf{y}$ is

$$y_i = x_{s,0} + x_{s,1} \cdot i + x_{s,2} \cdot i^2 + \cdots + x_{s,K} \cdot i^K + e_i, \tag{1}$$

where $e_i$ stands for perturbations by uncorrelated Gaussian noise with zero mean and positive variance. In (1), we use the standard polynomial basis $\{1, i, i^2, \ldots, i^K\}$, which is shared across all segments. In vector notation, the signal model is

$$\mathbf{y} = \left[\mathbf{I}\ \mathbf{D}^1 \ldots \mathbf{D}^K\right] \begin{bmatrix} \mathbf{x}_0 \\ \vdots \\ \mathbf{x}_K \end{bmatrix} + \mathbf{e} \tag{2}$$

where $\mathbf{x}_k = [x_{1,k}, x_{2,k}, \ldots, x_{N,k}]^\top \in \mathbb{R}^N$, $k \in 0, \ldots, K$, are the parametrization vectors, $\mathbf{I} = \mathbf{I}_N$ is the identity matrix, $\mathbf{D} = \operatorname{diag}(1/N, 2/N, \ldots, 1)$ is the diagonal matrix with linearly growing elements[1], $\mathbf{D}^j$ is its power (and therefore $\mathbf{D}^0 = \mathbf{I}$), and $\mathbf{e}$ represents the noise vector. We will often use a shortened notation $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e}$ instead of (2) for simplicity.

Due to the above mentioned formulation, it is obvious that signal generation by $\mathbf{A} \cdot \mathbf{x}$ is heavily over-parametrized, meaning that there are infinitely many vectors $\mathbf{x}$ that generate identical $\mathbf{A}\mathbf{x}$. On the other hand, the parameters in $\mathbf{x}$ should be strongly related—consider that for a signal consisting of $S$ consecutive segments built from polynomials up to degree $K$, the parametrization vectors $\mathbf{x}_k$ stay piecewise-constant within each segment; in addition, the breakpoints in all $K+1$ vectors $\mathbf{x}_k$ appear at the same position. This is a crucial observation which will be used next, in composing the optimization problem.

Together with the assumption $S \ll N$, the above motivates the use of a measure of sparsity for our segmentation/denoising problem. A vector is termed $m$-sparse when it consists of no more than $m$ nonzero elements. Optimization problems involving such a vector characterization, were, however, shown to be NP-hard [3, 4]. Probably the most popular choice of a surrogate that approximates true-sparsity-inducing functionals is the $\ell_1$-norm defined by $\|\mathbf{z}\|_1 = \sum_n |z_n|$. Such a choice performs surprisingly well, while this fact has also been theoretically justified [10, 12].

We formulate the problem as follows:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \left\{ \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \|[\tau_0 \nabla \mathbf{x}_0, \ldots, \tau_K \nabla \mathbf{x}_K]\|_{21} \right\} \tag{3}$$

[1] Note that in contrast to (1), the basis’ time vector is rescaled. We observed that using $\mathbf{D} = \operatorname{diag}(1, 2, \ldots, N)$ as in [14, 24, 30] leads to numerical problems for even a moderate $N$. Personal communication with the authors of [14] confirms this issue.

where $\boldsymbol{\tau} = [\tau_0, \ldots, \tau_K]^\top$ are $K+1$ positive weights, corresponding to individual polynomial degrees. These scalars should be set carefully according to the noise level and prior experience.

In (3), the first term is the "data fidelity" term, enforcing the estimate to approximately correspond to the observation $\mathbf{y}$. The eucleidian $\ell_2$-norm reflects the fact that gaussianity of the noise is assumed.

The second term in (3) is the "regularizer". This functional mathematically transcripts the desired properties of the estimate and assigns high values to vectors $\mathbf{x}$ that lack such properties. Piecewise-constant vectors $\mathbf{x}_k$ suggest that these vectors are sparse under the finite difference operation, which will be denoted $\nabla$, and defined by $\nabla\mathbf{z} = [z_2 - z_1, \ldots, z_N - z_{N-1}]$. We use the so-called mixed $\ell_{21}$-norm [20], which acts on a matrix $\mathbf{Z}$ of size $p \times q$ and is formally given by

$$\|\mathbf{Z}\|_{21} = \left\| \left[\|\mathbf{Z}_{1,:}\|_2, \|\mathbf{Z}_{2,:}\|_2, \ldots, \|\mathbf{Z}_{p,:}\|_2\right] \right\|_1 = \|\mathbf{Z}_{1,:}\|_2 + \ldots + \|\mathbf{Z}_{p,:}\|_2\,, \tag{4}$$

i. e. the $\ell_2$-norm is applied to the particular rows of $\mathbf{Z}$ and the resulting vector is evaluated by the $\ell_1$-norm. Therefore, the $\ell_{21}$-norm, used as a regularizer, promotes sparsity across columns of the matrix, and does not promote sparsity across the rows. In our case, matrix columns are the difference vectors $\tau_k\nabla\mathbf{x}_k$, and therefore the *joint* sparsity of differences is enforced. This means that the differences should be zero at most matrix positions, and nonzero values should be concentrated in a few rows, i. e. at the points corresponding to the changepoints in the model.

The solution, i. e. vector $\hat{\mathbf{x}} = [\hat{\mathbf{x}}_0^\top \ldots \hat{\mathbf{x}}_K^\top]^\top$ contains the obtained $N(K+1)$ optimizers, and (desirably joint) nonzero values in $\nabla\hat{\mathbf{x}}_k$s indicate the possible segment borders (changepoints).

## 3. ALGORITHMS

This section provides a description of how the optimization problem (3) is numerically solved and how the result is postprocessed to achieve the final segmented (and denoised) estimate.

### 3.1. Proximal splitting methods

Proximal splitting methodology is a tool for iterative solution of convex minimization problems [8]. Proximal algorithms are especially effective in finding the minimum of a sum of convex functions $f_i$, with mild assumptions about these functions. Based on the properties of the functions, proximal algorithms perform iterations involving an evaluation of gradients and/or proximal operators related individually to each $f_i$, which is much simpler than minimizing the composite functional by other means. Such first-order methods are in particular suitable for large-scale problems [6]. Proximal algorithms provide sequences that are guaranteed to converge to the optimal value. The speed of convergence is influenced by the properties of particular $f_i$s and by particular values of the parameters chosen in the algorithms.

For our problem (3), we use the Chambolle-Pock (CP) algorithm [6], which will be described in Section 3.3. While other algorithms could be utilized as well, we found that the CP algorithm was the fastest available. See for example our study in [24], which shows that gradient-based algorithms (like the Forward-backward based primal-dual algorithm [18]) are not suitable for our purpose, as they provide slow convergence due to the matrix $\mathbf{A}$ being ill conditioned.

### 3.2. Proximal operators

A proximal operator of a function $h$ maps $\mathbf{z} \in \mathbb{R}^N$ to another vector in $\mathbb{R}^N$, such that

$$\operatorname*{prox}_{h}(\mathbf{z}) = \operatorname*{arg\,min}_{\mathbf{x}\in\mathbb{R}^N} \left\{ h(\mathbf{x}) + \frac{1}{2}\|\mathbf{z}-\mathbf{x}\|_2^2 \right\}. \tag{5}$$

Note that below, matrices will be regarded as reshaped one-dimensional vectors, where appropriate.

This article will make use of proximal operators of two particular functions. First, we have that for real-valued matrix $\mathbf{A}$ [8, 9],

$$\operatorname*{prox}_{\frac{\tau}{2}\|\mathbf{A}\cdot-\mathbf{y}\|_2^2}(\mathbf{z}) = (\mathbf{I} + \tau\mathbf{A}^\top\mathbf{A})^{-1}(\mathbf{z} + \tau\mathbf{A}^\top\mathbf{y}). \tag{6}$$

Second, the proximal operator of the $\ell_{21}$-norm is

$$\operatorname*{prox}_{\tau\|[\cdot,\ldots,\cdot]\|_{21}}(\mathbf{Z}) = \mathrm{soft}_\tau^{\mathrm{row}}(\mathbf{Z}), \tag{7}$$

mapping matrix $\mathbf{Z} = [z_{ij}]$ to another [19]. It can be shown that it is a case of soft thresholding over groups consisting of rows of the matrix; specifically, $\mathrm{soft}_\tau^{\mathrm{row}}$ is a mapping

$$z_{ij} \mapsto \frac{z_{ij}}{\|\mathbf{Z}_{i,:}\|_2} \max(\|\mathbf{Z}_{i,:}\|_2 - \tau). \tag{8}$$

Given a convex function $f$, the proximal operator of its Fenchel–Rockafellar conjugate $f^*$ can be computed at virtually the same cost as $\operatorname{prox}_f$ owing to the Moreau identity [9, 18]:

$$\operatorname*{prox}_{\alpha f^*}(\mathbf{u}) = \mathbf{u} - \alpha \operatorname*{prox}_{f/\alpha}(\mathbf{u}/\alpha) \qquad \text{for } \alpha \in \mathbb{R}^+. \tag{9}$$

### 3.3. Chambolle-Pock splitting

The Chambolle-Pock (CP) algorithm [6] is tailored to solve problems of type

$$\text{minimize} \;\; f_1(\mathbf{x}) + f_2(L\mathbf{x}), \tag{10}$$

where both $f_1$ and $f_2$ are convex, lower semi-continuous functions with non-empty domains, mapping real vectors to $(-\infty; \infty]$. Both the functions can be non-smooth, and $L$ stands for any linear operator (representable by a matrix in our finite-dimensional setting). The CP algorithm is primal–dual and iterative, while each iteration consists of two proximal steps with respect, in turn, to $f_1$ and $f_2$, respectively, combined with the application of $L$ and $L^\top$.

**Algorithm 1:** The Chambolle-Pock (CP) algorithm

```
Input: Functions $f_1, f_2$, and a linear operator $L \in \mathbb{R}^{q \times p}$
Output: $\bar{\mathbf{p}}^{(n)}$
Set parameters $\zeta, \sigma > 0$ and $\theta \in [0,1]$;
Set initial primal variables $\mathbf{p}^{(0)} \in \mathbb{R}^p$ and dual variables $\mathbf{q}^{(0)} \in \mathbb{R}^q$;
Set initial output variables $\bar{\mathbf{p}}^{(0)} = \mathbf{p}^{(0)}$;
Set iteration counter $n = 0$;
repeat
    $\mathbf{q}^{(n+1)} = \mathrm{prox}_{\sigma f_2^*}\left(\mathbf{q}^{(n)} + \sigma L\,\bar{\mathbf{p}}^{(n)}\right)$;
    $\mathbf{p}^{(n+1)} = \mathrm{prox}_{\zeta f_1}\left(\mathbf{p}^{(n)} - \zeta L^\top \mathbf{q}^{(n+1)}\right)$;
    $\bar{\mathbf{p}}^{(n+1)} = \mathbf{p}^{(n+1)} + \theta(\mathbf{p}^{(n+1)} - \mathbf{p}^{(n)})$;
    $n \leftarrow n + 1$;
until convergence;
return $\bar{\mathbf{p}}^{(n)}$
```

The CP algorithm is provided in Algorithm 1. For $\theta = 1$, the authors of [6] show that the convergence is ensured when $\zeta\sigma\|L\|^2 < 1$, where $\|\cdot\|$ denotes the operator/spectral norm. Note, however, that later a more general algorithm appeared [9] that recovers the CP algorithm as its special case, while the step parameter could range to as much as $\theta \in (0, 2)$. The latter fact can significantly improve the convergence speed. Iterations are terminated if a convergence criterion is met, for example if the relative change in the primal variables is small enough: $\|\bar{\mathbf{p}}^{(n+1)} - \bar{\mathbf{p}}^{(n)}\|_2 / \|\bar{\mathbf{p}}^{(n)}\|_2 \leq \epsilon$.

Returning to (3), assign $f_1(\mathbf{x}) = \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2$. Although it is a smooth function, its gradient will not be utilized, since the model matrix $\mathbf{A}$ is ill conditioned, and therefore the iterative gradient steps are not quite effective. The proximal operator of $f_1$ is displayed in (6).

Next, assign $f_2(\mathbf{u}) = f_2(\mathbf{u}_0, \ldots, \mathbf{u}_K) = \|[\mathbf{u}_0, \ldots, \mathbf{u}_K]\|_{21}$. It is a non-smooth function and the proximal operator of $f_2$ has been established in (7) and (8); there we have $\tau = 1$. Moreover, for use in the CP algorithm, the relation (9) for the conjugate function should be used.

The linear operator $L$ and its transpose are

$$L(\mathbf{x}) = L(\mathbf{x}_0, \ldots, \mathbf{x}_K) = [\tau_0 \nabla \mathbf{x}_0, \ldots, \tau_K \nabla \mathbf{x}_K] \tag{11}$$

$$L^\top(\mathbf{u}) = L^\top(\mathbf{u}_0, \ldots, \mathbf{u}_K) = \begin{bmatrix} \tau_0 \nabla^\top \mathbf{u}_0 \\ \vdots \\ \tau_K \nabla^\top \mathbf{u}_K \end{bmatrix} \tag{12}$$

with $\nabla$ of size $(N-1) \times N$. The reader may notice that the mapping $L : \mathbb{R}^{KN} \to \mathbb{R}^{N-1 \times K}$ also involves reshaping a long vector to a matrix and back, which is necessary for the application of the $\ell_{21}$-norm as defined in (4). The analogue holds for $L^\top : \mathbb{R}^{N-1 \times K} \to$

$\mathbb{R}^{KN}$. The operator $\nabla^\top$ can be implemented as efficiently as $\nabla$ since

$$\nabla^\top \mathbf{u} = \nabla \left( \begin{bmatrix} 0 \\ -\mathbf{u} \\ 0 \end{bmatrix} \right). \tag{13}$$

An upper bound on $\|L\|$ will also be needed:

$$\begin{aligned}
\|L\|^2 &= \max_{\|\mathbf{x}\|_2=1} \|L\mathbf{x}\|_2^2 \\
&= \max_{\|\mathbf{x}\|_2=1} \|[\tau_0 \nabla \mathbf{x}_0, \ldots, \tau_K \nabla \mathbf{x}_K]\|_2^2 \\
&= \max_{\|\mathbf{x}\|_2=1} \left( \|\tau_0 \nabla \mathbf{x}_0\|_2^2 + \ldots + \|\tau_K \nabla \mathbf{x}_K\|_2^2 \right) \\
&\leq \tau_0^2 \max_{\|\mathbf{x}\|_2=1} \|\nabla \mathbf{x}_0\|_2^2 + \ldots + \tau_K^2 \max_{\|\mathbf{x}\|_2=1} \|\nabla \mathbf{x}_K\|_2^2 \\
&\leq \tau_0^2 \|\nabla\|^2 + \ldots + \tau_K^2 \|\nabla\|^2 \\
&\leq 4(\tau_0^2 + \ldots + \tau_K^2).
\end{aligned}$$

since it can be easily shown that it holds $\|\nabla\| \leq 2$. From the above it follows that $\|L\| \leq 2\sqrt{\sum_{k=0}^{K} \tau_k^2} =: 2\|\boldsymbol{\tau}\|_2$. In turn, convergence of the CP algorithm is guaranteed whenever $\zeta\sigma \leq 1/(4\|\boldsymbol{\tau}\|_2^2)$.

In the described setting, the main steps of the particular CP algorithm thus read

- $\mathbf{q}^{(n+1)} = (Id - \mathrm{soft}_{1/\sigma}^{\mathrm{row}}) \left( \mathbf{q}^{(n)}/\sigma + L\, \bar{\mathbf{p}}^{(n)} \right)$
- $\mathbf{p}^{(n+1)} = \left( \mathbf{I} + \zeta \mathbf{A}^\top \mathbf{A} \right)^{-1} \left( \mathbf{p}^{(n)} - \zeta L^\top \mathbf{q}^{(n+1)} + \tau \mathbf{A}^\top \mathbf{y} \right)$

with $Id$ standing for the identity operator. Note that the matrix $\left( \mathbf{I} + \zeta \mathbf{A}^\top \mathbf{A} \right)^{-1}$ can be precomputed, and Theorem 1 in the Appendix introduces an explicit formula for computing the inversion, which is possible thanks to the multi-diagonal structure of $\mathbf{A}^\top \mathbf{A}$. In addition, $\mathbf{A}^\top \mathbf{y}$ can also be precomputed.

### 3.4. Signal segmentation and final estimation

Vectors $\hat{\mathbf{x}}$ obtained as optimizers of our problem (3) allow simple estimation of the underlying noiseless signal according to $\hat{\mathbf{y}} = \mathbf{A}\hat{\mathbf{x}}$. This section discusses the means by which this reconstruction could be improved at a negligible computational cost.

Recall that nonzero values in $\nabla\hat{\mathbf{x}}_0, \ldots, \nabla\hat{\mathbf{x}}_K$ indicate segment borders. However, only in rare cases, can regularization weights $\tau_0, \ldots, \tau_K$ be fine-tuned to achieve truly piecewise-constant optimizers. The non-strict convexity of the total-variation-like regularizer $\|\nabla \cdot\|_{21}$ in (3) makes the problem no easier; see, for example, the discussion of this phenomenon in [28, 30]. Even when the $K + 1$ scalars $\tau_k$ are set properly, vectors $\nabla\hat{\mathbf{x}}_k$ are in practice full of small values, besides larger entries indicating possible segment borders. The underlying signal is assumed to be piecewise-polynomial, and therefore, a two-part procedure is applied to obtain the denoised signal: first, the borders of the signal segments are detected and fixed, and second, optimal parameters are found for each detected segment individually.

### 3.4.1. Changepoints detection

The process of changepoints detection attempts avoiding false alarms caused by small values in the difference vectors. Thanks to the use of the mixed norm, significant values in $\nabla\hat{\mathbf{x}}_k$s tend to be situated at identical indexes, and our detection procedure can benefit from this property. Thus, we start by forming a single vector out of all achieved $\nabla\hat{\mathbf{x}}_k$s, which is done by computing the euclidean distance according to the formula

$$\mathbf{d} = \sqrt{(\alpha_0 \nabla\hat{\mathbf{x}}_0)^2 + \cdots + (\alpha_K \nabla\hat{\mathbf{x}}_K)^2}, \tag{14}$$

where all the operations are considered as acting elementwise on the vectors. The scalars $\alpha_0, \ldots, \alpha_K$ are positive weights used as balancers because the range of values in the parametrization vectors differs with the order of the polynomial they correspond to.[2] The weights are computed as $\alpha_k = 1/\max(|\nabla\hat{\mathbf{x}}_k|)$. Then, a moving median filter is applied to $\mathbf{d}$ and the filtered signal obtained is subtracted from $\mathbf{d}$, yielding $\hat{\mathbf{d}}$. This approach leaves the significant breakpoints and filters the segments between breakpoints, such that it provides a signal with more zero (or close to zero) values in $\hat{\mathbf{d}}$. Finally, the indexes in $\hat{\mathbf{d}}$ satisfying $\left|\hat{\mathbf{d}}\right| > \lambda$ with a properly set threshold $\lambda$ constitute the detected breakpoints.

Occasionally, two or even more breakpoints could be detected that are adjacent to each other. In ideal conditions, this would indicate segments of length one. We observed that in practice, however, such a situation can appear for long segments as well. Therefore, anytime such a situation is detected, only the absolute largest index is chosen.

### 3.4.2. Obtaining final optimal parameters

As the final step, denoising/smoothing is applied to each detected segment *separately.* This is done simply by forming a regression matrix consisting of the polynomial basis up to order $K$ as its columns, and applying an ordinary least-squares method to data $\mathbf{y}$ restricted to the segment range. By doing this, we obtain a new parametrization coefficient set $\dot{\mathbf{x}}$, which is constant on the segment-by-segment basis. The complete denoised signal $\dot{\mathbf{y}}$ is then reconstructed according to $\dot{\mathbf{y}} = \mathbf{A}\dot{\mathbf{x}}$. The least-squares refit could be seen as a procedure usually termed "debiasing", commonly used in LASSO-type estimation [5, 34], which is inherently biased by the use of the $\ell_1$-norm [29].

## 4. EXPERIMENTS

### 4.1. Experiments on synthetic data

**Quadratic signal with low noise.** A randomly generated piecewise-polynomial signal of degree 2 and of total length $N = 300$ was used. The signal contained six independent polynomial segments. The parametrization vectors $\mathbf{x}_0$, $\mathbf{x}_1$ and $\mathbf{x}_2$ in $\mathbb{R}^N$ used for the generation of the signal are piecewise-constant, as had been assumed. The signal $\mathbf{y}$ is created by computing $[\mathbf{I}\ \mathbf{D}\,\mathbf{D}^2][\mathbf{x}_0^\top\ \mathbf{x}_1^\top\ \mathbf{x}_2^\top]^\top$ and the independent and identically distributed (iid) Gaussian noise is added to it, such that the signal-to-noise ratio (SNR) is 30.15 dB.

[2] Alternatively, other measures can be used, for example $\ell_p$-norms for different $p \in [1, \infty]$. In our experience, however, the $\ell_2$-distance as in (14) was the best one for the practical experiments.

Figure 1 shows an approximation of this signal achieved with $\tau_0 = 0.1153$, $\tau_1 = 0.0807$, $\tau_2 = 0.0576$ (these regularization parameters were tuned to obtain the parametrization coefficients as close to piecewise-constant as possible). The CP algorithm was stopped after 2,000 iterations. The median filter window of length 5 and the threshold $\lambda = 0.1$ were used, see Sec. 3.4. While the SNR of the input signal was 30.15 dB, the SNR of the signal $\mathbf{A}\hat{\mathbf{x}}$ was 34.57 dB and the SNR of the final estimate after the least-squares method ($\mathbf{A}\dot{\mathbf{x}}$, shown in the upper figure in green) was 43.61 dB. Observe the perfect detection of the changepoints and the great fit of the polynomial segments.

**Quadratic signal with high noise.** The same as above, a signal has been generated, but with the parametrization vectors $\mathbf{x}_0$, $\mathbf{x}_1$ and $\mathbf{x}_2$ taken differently, including different positions of the changepoints. The iid Gaussian noise for this case had higher variance, the SNR is 19.77 dB.

Figure 2 shows the results that have been achieved using $\tau_0 = 0.8439$, $\tau_1 = 0.5907$, $\tau_2 = 0.4219$ after 1,000 iterations of the CP algorithm, again followed by the least-squares refitting. A median filter window of length 5 and the threshold $\lambda = 0.22$ were used. The SNR of the signal $\mathbf{A}\hat{\mathbf{x}}$ was 29.85 dB and the SNR of the final estimate $\mathbf{A}\dot{\mathbf{x}}$ was 33.66 dB. Note that the model fit is worse due to stronger noise but, at the same time, the changepoints have again been detected perfectly, including the very short segment around time index 130.

### 4.2. Experiment on real data

Our model has been tested on a signal acquired by an OTDR (Optical Time Domain Reflectometer) instrument.[3] Optical Time Domain Reflectometry [15] is a basic method used for evaluating the quality of optical routes in optical networks. A series of subsequent optical pulses with a duration in the order of tens or hunderds of nanoseconds is launched into the fiber and the amount of back-scattered light is measured.

Since the reflections follow an exponential model, the acquired data are subject to logarithmization, which makes the measurement linearized. Our linearized signal is depicted in red in Fig. 3. Such data can be used to characterize the fiber along its length. Separate sections of the cable can be detected, since the welding points are visible in the graph as jumps. We can also characterize the attenuation factor of each cable segment, as it is represented by the local slope of the signal.

The signal was 446 samples long, which corresponds to ca 55 kilometers of cable. The signal was measured with a wavelength of 1310 nm. The cable consists of five segments of almost identical length. We run our optimization program on this data with $\tau_0 = 1.6$ and $\tau_1 = 1.4$. After 1500 iterations of the CP algorithm and subsequent post-processing (with median filter length 25 and $\lambda = 0.11$), we got perfect detection of the changepoints.

The time-domain result of the OTDR segmentation/denoising is depicted in green in Fig. 3. Fig. 4 shows the intermediate steps in terms of the parametrization vectors. We observe that $\hat{\mathbf{x}}_0, \hat{\mathbf{x}}_1$ are far from having stepwise character, in contrast to what is desirable. On the other hand, $\nabla\hat{\mathbf{x}}_0$ and $\nabla\hat{\mathbf{x}}_1$ still carry significant values that lead to the detection of segments.

[3] Our signal is one of the example signals from the OTDR FTB-7000 software by EXFO Inc., Canada, ver. 6.14.72.280.

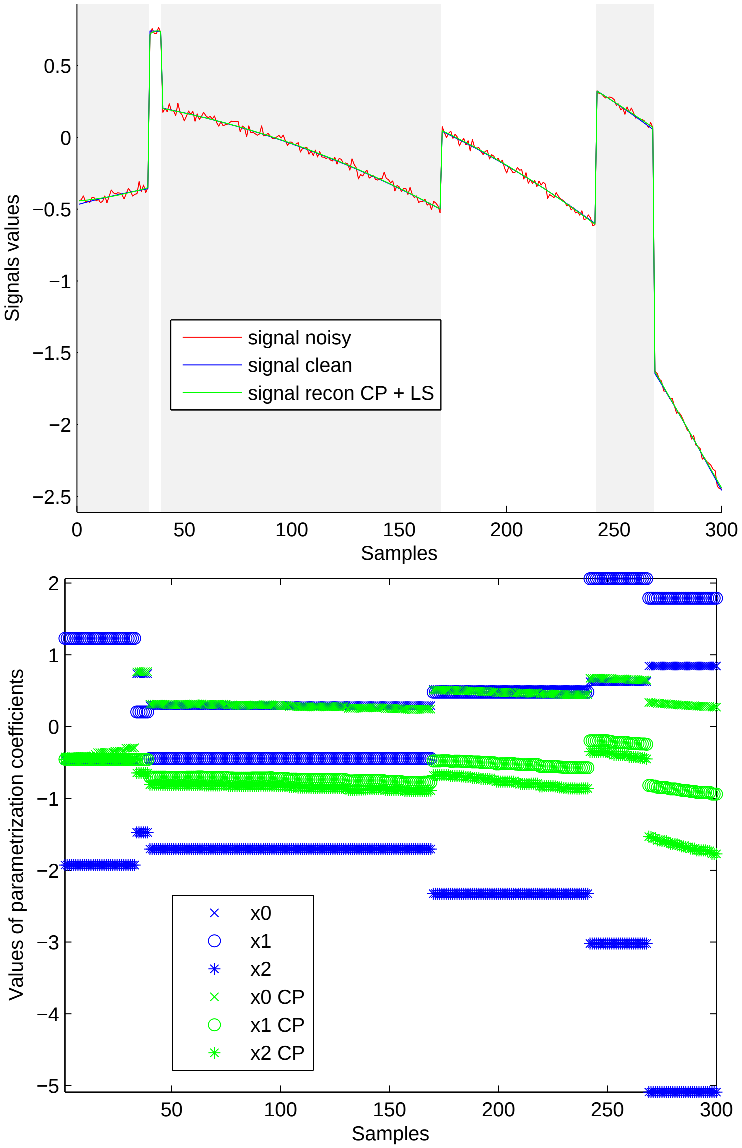

**Fig. 1.** Approximation of a random piecewise-polynomial signal in the low-noise case. In the upper figure, the clean signal and its noisy version $\mathbf{y}$ are depicted, as well as the final approximation achieved by our algorithm. In the lower figure, the respective parametrization coefficients are presented (i. e. the vectors $\hat{\mathbf{x}}_k$ subsequently used for the changepoint detection).

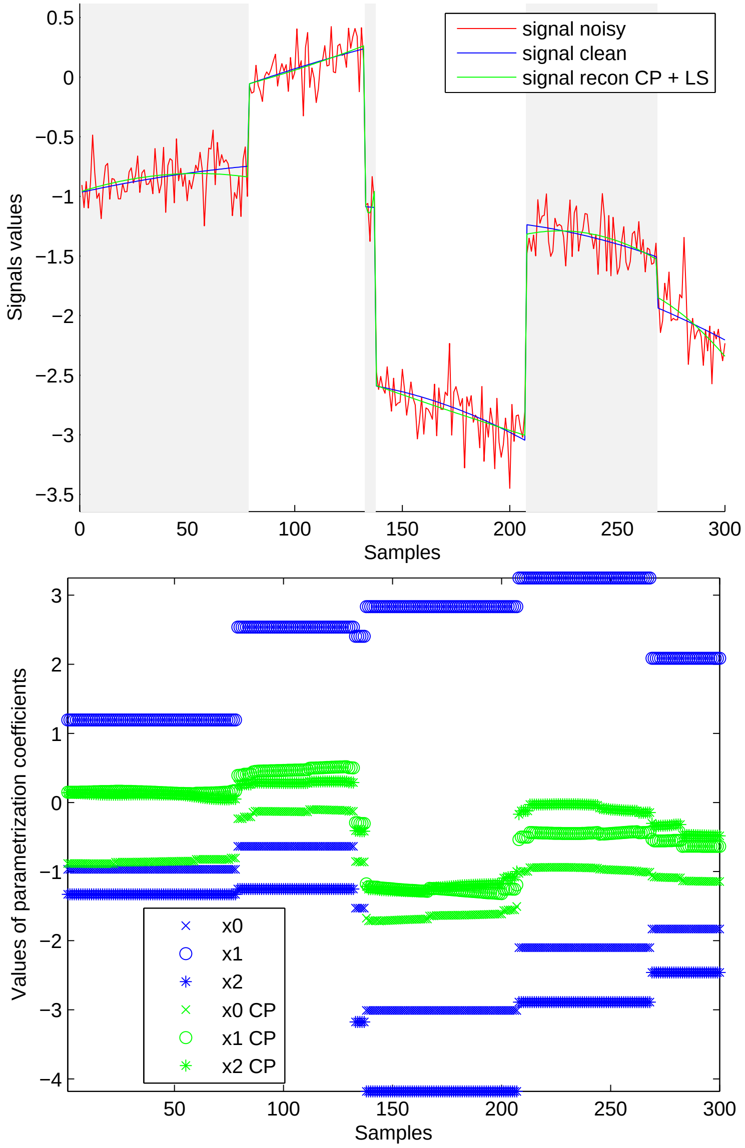

**Fig. 2.** Similarly to Fig. 1, an approximation of a random piecewise-polynomial signal is presented, this time in the high-noise regime. The bottom plot shows the parametrization coefficients before they are passed on to the changepoint detection.

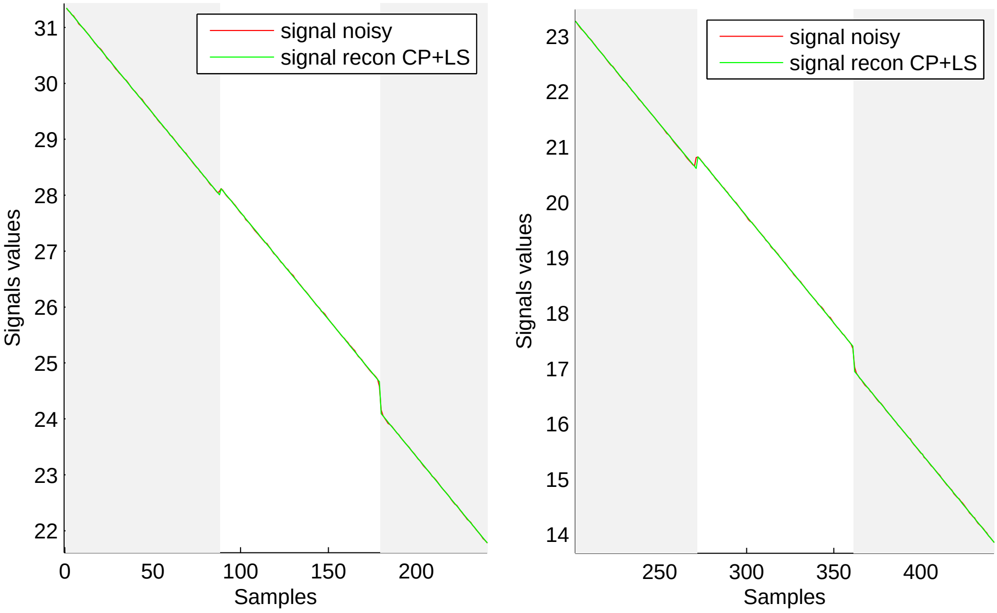

**Fig. 3.** Segmentation/denoising of real OTDR data. The optical cable consists of five segments. We got perfect detection of the changepoints.

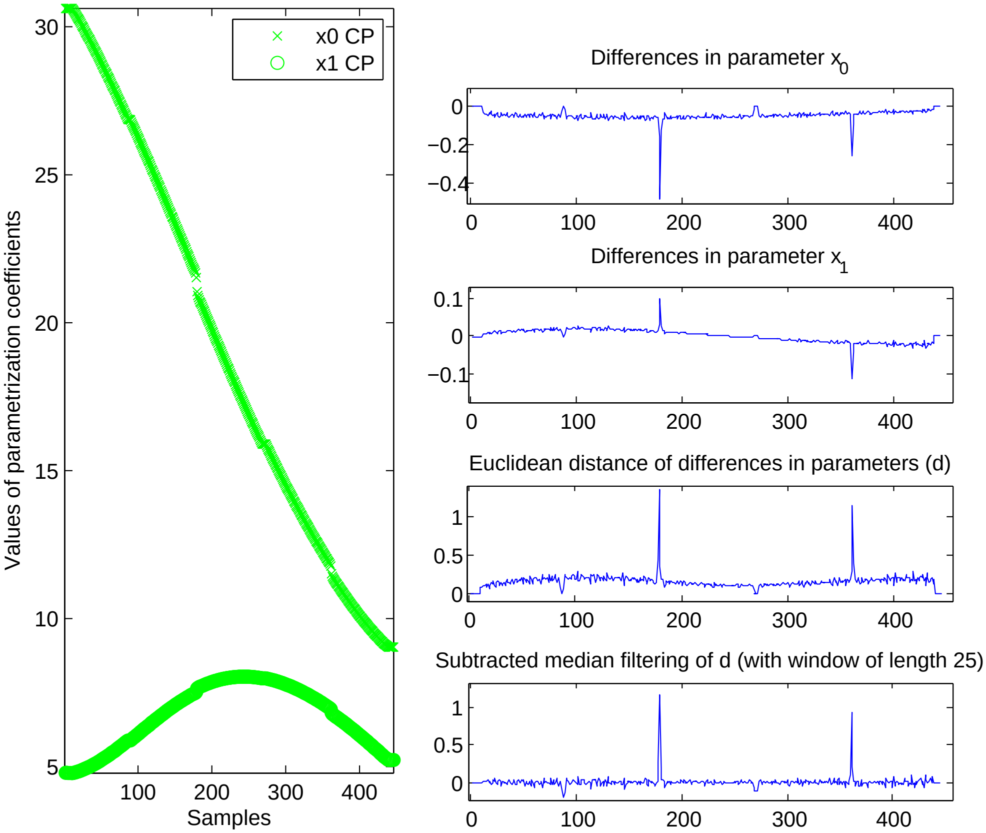

**Fig. 4.** The intermediate results related to the OTDR experiment. Left: parametrization vectors $\hat{\mathbf{x}}_0, \hat{\mathbf{x}}_1$ as obtained by the CP algorithm. Right: post-processing of $\hat{\mathbf{x}}_0, \hat{\mathbf{x}}_1$ for the detection of segment borders.

## 5. SOFTWARE

The experiments have been done in Matlab (R2010a) and for the proximal algorithms we benefited from using the flexible UnLocBox toolbox[4] [27] in version 1.7.2. In the spirit of reproducible research, we make the experiments-related codes available. The reader can download the archive at
`http://www.utko.feec.vutbr.cz/~rajmic/software/sparse_signal_segment.zip`

A brief description of the main `m`-files follows:

`DEMO_1` .................... script that performs the synthetic experiment with low noise (datafile `DEMO_1_data.mat`), see Fig. 1

`DEMO_2` .................... script that performs the synthetic experiment with higher noise (file `DEMO_2_data.mat`), see Fig. 2

`DEMO_3` .................... script that performs the segmentation experiment on real data (datafile `DEMO_3_data.mat`), see Fig. 3 and Fig. 4

`lib_CP` .................... function which computes the parameters $\hat{\mathbf{x}}$ using the CP algorithm; this function internally uses Unlocbox routines

`lib_recon_from_params` ... function that reconstructs the signal from the (estimated) parameters

`lib_least_squares_fit` ... function refitting a given segment of the signal by the ordinary least-squares method, producing the parameters $\dot{\mathbf{x}}$

`matrix_inversion` ........ function that computes the inversion of $\mathbf{I} + \lambda\mathbf{A}^\top\mathbf{A}$ according to Theorem 1

`lib_generate_signal` ..... function that randomly generates a piecewise-polynomial signal of selected degree, number of segments and length

`lib_signal_denoising` .... function which takes $\hat{\mathbf{x}}$, performs changepoints detection, least-squares refitting of each detected segment, and the final signal reconstruction $\dot{\mathbf{y}}$

`lib_chanpoint_detect` .... function that detects borders of the segments.

## 6. DISCUSSION AND CONCLUSION

The experiments show that the proposed methodology of signal segmentation/denoising is very promising. However, careful tuning of the parameters must be carried out to obtain proper segmentation.

More specifically, the less a changepoint is barely recognizable (in our case with the naked eye in a 1D signal), the more tuning is required, and there are signals whose segmentation is very user-demanding. We experienced such a case with OTDR data (not published here), where, despite much effort, we were not able to force a breakpoint belonging to a tiny jump in the time series. However, we notice that this behaviour is in fact in agreement with the analysis in [30], where we argued that an easy (or easier) achievement of good results is in fact complicated by the TV(-like) regularizer. In this article, the $\ell_{21}$-norm is used, which actually exhibits the properties of the total variation regularizer. In brief, the main problem of the functionals containing the difference operator and the $\ell_1$-norm is their non-strict convexity.

[4] `https://lts2.epfl.ch/unlocbox`

One could therefore hope for better results via using more robust recovery programs, developed by introducing non-convex regularizers. For example, using $\ell_p$-"norms" with $p < 1$ [7] would be a natural step further.

One could also think about imitating non-convexity via a series of convex programs. Such an approach is not new in general, it has been both theoretically and practically justified in [5], for example. To be more specific, a series of convex problems is formulated where the parameters of the currently solved problem depend on the latest solution. In our case, this approach would suggest augmenting problem (3) by changing the regularizer to elementwise weighted differences, instead of applying a common weight $\tau_k$ to the vector $\nabla \mathbf{x}_k$ in (3). Such weights for all difference terms would change adaptively after a suitable number of iterations of the CP algorithm, based on the latest solution. The points which are most probably the breakpoints would gradually be assigned lower weight, thus less penalized. The procedure would stop when the breakpoint locations become stable. In fact, problem (3) would change merely by introducing a modified linear operator

$$\bar{L}^{(j)}(\mathbf{x}) = \begin{bmatrix} \operatorname{diag}\left(\mathbf{w}_1^{(j)}\right) \cdot \nabla & \cdots & 0 \\ & \ddots & \\ 0 & \cdots & \operatorname{diag}\left(\mathbf{w}_K^{(j)}\right) \cdot \nabla \end{bmatrix} \cdot \mathbf{x},$$

where $j$ represents the number of problem repetitions.

Despite its theoretical attractivity, we report that we were not successful with such an approach. The explanation is that adaptively changing the weights $\mathbf{w}_k^{(j)}$ causes disbalance between the data fidelity term and the regularizer. Due to the complex structure of the regularizer, it is definitively not clear how one should "normalize" the weights in each repetition to keep a desirable balance.

A way to avoid the described complication is to solve a series of *constrained* convex problems,

$$\arg\min_{\mathbf{x}} \left\| \bar{L}^{(j)} \mathbf{x} \right\|_{21} \text{ subject to } \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2 \leq \delta, \tag{15}$$

where $\delta$ is the estimate of the noise level. Constructing an efficient algorithm to solve (15) is however outside the scope of this article.[5]

[5] In the course of the review process, our study addressing this issue has been published, see [25]. The paper compares different numerical approaches to solve (15) and at the same time it shows that although reweighting brings some improvement in detecting breakpoints, careful tuning of parameters has still to be carried out.

## 7. APPENDIX

**Theorem 1.** Let $\mathbf{A} = \left[\mathbf{I}\ \mathbf{D}\ \mathbf{D}^2 \dots \mathbf{D}^k\right]$ be a matrix with $\mathbf{D} = \operatorname{diag}\left(d_1, \dots, d_N\right)$ square and diagonal. Then for the inversion of the $N(k+1) \times N(k+1)$ matrix $\mathbf{I} + \lambda \mathbf{A}^\top \mathbf{A}$ with $\lambda \in \mathbb{R}^+$ it holds

$$\left(\mathbf{I} + \lambda \mathbf{A}^\top \mathbf{A}\right)^{-1} = \mathbf{I} - \lambda \left(\mathbf{I} \otimes \mathbf{C}^{-1}\right) \mathbf{A}^\top \mathbf{A},$$

where the identity matrices $\mathbf{I}$ are of appropriate sizes, the symbol $\otimes$ denotes the Kronecker product and the $N \times N$ matrix $\mathbf{C}$ is defined by

$$\mathbf{C} = \mathbf{I} + \lambda \sum_{i=0}^{k} \mathbf{D}^{2i} = \mathbf{I} + \lambda \mathbf{A}\mathbf{A}^\top.$$

Proof. In the definition of matrix $\mathbf{C}$, we utilize the fact that $\mathbf{A}\mathbf{A}^\top = \left[\mathbf{I} \dots \mathbf{D}^k\right] \cdot \left[\mathbf{I} \dots \mathbf{D}^k\right]^\top = \sum_{i=0}^{k} \mathbf{D}^{2i}$. We have $\left(\mathbf{I} \otimes \mathbf{C}\right)^{-1} = \left(\mathbf{I} \otimes \mathbf{C}^{-1}\right)$ and we use this fact to find the product:

$$\begin{aligned}
&\left(\mathbf{I} + \lambda \mathbf{A}^\top \mathbf{A}\right)^{-1} \cdot \left(\mathbf{I} + \lambda \mathbf{A}^\top \mathbf{A}\right) \\
&= \left[\mathbf{I} - \lambda \left(\mathbf{I} \otimes \mathbf{C}^{-1}\right) \mathbf{A}^\top \mathbf{A}\right] \cdot \left(\mathbf{I} + \lambda \mathbf{A}^\top \mathbf{A}\right) \\
&= \mathbf{I} - \lambda \left(\mathbf{I} \otimes \mathbf{C}^{-1}\right) \mathbf{A}^\top \mathbf{A} + \lambda \mathbf{A}^\top \mathbf{A} - \lambda^2 \left(\mathbf{I} \otimes \mathbf{C}^{-1}\right) \mathbf{A}^\top \mathbf{A} \mathbf{A}^\top \mathbf{A} \\
&= \mathbf{I} + \left(\mathbf{I} \otimes \mathbf{C}^{-1}\right) \left[-\mathbf{I} + \left(\mathbf{I} \otimes \mathbf{C}\right) - \lambda \mathbf{A}^\top \mathbf{A}\right] \lambda \mathbf{A}^\top \mathbf{A}.
\end{aligned}$$

Now, since

$$\mathbf{I} \otimes \mathbf{C} = \begin{bmatrix} \mathbf{I} + \lambda \sum_{i=0}^{k} \mathbf{D}^{2i} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{I} + \lambda \sum_{i=0}^{k} \mathbf{D}^{2i} \end{bmatrix}$$

and

$$\mathbf{A}^\top \mathbf{A} = \begin{bmatrix} \mathbf{I} & \mathbf{D} & \mathbf{D}^2 & \dots & \mathbf{D}^k \\ \mathbf{D} & \mathbf{D}^2 & \mathbf{D}^3 & \dots & \mathbf{D}^{k+1} \\ \mathbf{D}^2 & \mathbf{D}^3 & \mathbf{D}^4 & \dots & \mathbf{D}^{k+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{D}^k & \mathbf{D}^{k+1} & \mathbf{D}^{k+2} & \dots & \mathbf{D}^{2k} \end{bmatrix},$$

we see that

$$-\mathbf{I} + \left(\mathbf{I} \otimes \mathbf{C}\right) - \lambda \mathbf{A}^\top \mathbf{A} = \begin{bmatrix} \lambda \sum_{i \neq 0} \mathbf{D}^{2i} & -\lambda \mathbf{D} & -\lambda \mathbf{D}^2 & \cdots & -\lambda \mathbf{D}^k \\ -\lambda \mathbf{D} & \lambda \sum_{i \neq 1} \mathbf{D}^{2i} & -\lambda \mathbf{D}^3 & \cdots & -\lambda \mathbf{D}^{k+1} \\ -\lambda \mathbf{D}^2 & -\lambda \mathbf{D}^3 & \lambda \sum_{i \neq 2} \mathbf{D}^{2i} & \cdots & -\lambda \mathbf{D}^{k+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\lambda \mathbf{D}^k & -\lambda \mathbf{D}^{k+1} & -\lambda \mathbf{D}^{k+2} & \cdots & \lambda \sum_{i \neq k} \mathbf{D}^{2i} \end{bmatrix}.$$

Multiplying this matrix from right by $\lambda \mathbf{A}^\top \mathbf{A}$ results in the zero matrix.

Since $\mathbf{I} + \lambda \mathbf{A}^\top \mathbf{A}$ is symmetric, so is its inverse, and it is not necessary to prove the multiplication in the opposite order.

We note that the theorem could be proven with the help of the (Sherman–Morrison–Woodbury) inversion lemma [16, 26], but we prefer the presented approach for clarity. □

## ACKNOWLEDGEMENT

Research described in this paper was financed by the National Sustainability Program under grant LO1401 and by the Czech Science Foundation under grant GA16-13830S. For the research, the infrastructure of the SIX Center was used.

The authors are thankful to Nathanaël Perraudin, Zdeněk Průša and Michal Šorel for valuable discussion and comments. Thanks are also due to Petr Münster for discussing the OTDR technology.

(Received January 19, 2017)

## REFERENCES

[1] D. Angelosante and G. B. Giannakis: Group Lassoing change-points in piecewise-constant AR processes. EURASIP J. Advances Signal Process. *2012* (2012), 1, 1–16. DOI:10.1186/1687-6180-2012-70

[2] K. Bleakley and J. P. Vert: The group fused Lasso for multiple change-point detection. Technical Report.

[3] A. M. Bruckstein, D. L. Donoho, and M. Elad: From sparse solutions of systems of equations to sparse modeling of signals and images. SIAM Rev. *51* (2009), 1, 34–81. DOI:10.1137/060657704

[4] E. J. Candes and M. B. Wakin: An introduction to compressive sampling. IEEE Signal Process. Magazine *25* (2008), 2, 21–30. DOI:10.1109/msp.2007.914731

[5] E. J. Candes, M. B. Wakin, and S. P. Boyd: Enhancing sparsity by reweighted $\ell_1$ minimization. J. Fourier Analysis Appl. *14* (2008), 877–905. DOI:10.1007/s00041-008-9045-x

[6] A. Chambolle and T. Pock: A first-order primal-dual algorithm for convex problems with applications to imaging. J. Math. Imaging Vision *40* (2011), 1, 120–145. DOI:10.1007/s10851-010-0251-1

[7] R. Chartrand: Shrinkage mappings and their induced penalty functions. In: IEEE International Conference on Acoustics, Speech, and Signal Processing 2014. DOI:10.1109/icassp.2014.6853752

[8] P. Combettes and J. Pesquet: Proximal splitting methods in signal processing. In: Fixed-Point Algorithms for Inverse Problems in Science and Engineering 2011, pp. 185–212. DOI:10.1007/978-1-4419-9569-8_10

[9] L. Condat: A generic proximal algorithm for convex optimization – application to total variation minimization. Signal Process. Lett., IEEE *21* (2014), 8, 985–989. DOI:10.1109/lsp.2014.2322123

[10] D. L. Donoho and M. Elad: Optimally sparse representation in general (nonorthogonal) dictionaries via $\ell_1$ minimization. Proc. National Acad. Sci. *100* (2003), 5, 2197–2202. DOI:10.1073/pnas.0437847100

[11] M. Elad: Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing. Springer 2010.

[12] M. Fornasier, ed.: Theoretical Foundations and Numerical Methods for Sparse Recovery. De Gruyter, Berlin, Boston 2010. DOI:10.1515/9783110226157

[13] J. Frecon, N. Pustelnik, P. Abry, and L. Condat: On-the-fly approximation of multivariate total variation minimization. IEEE Trans. Signal Process. *64* (2016), 9, 2355–2364. DOI:10.1109/tsp.2016.2516962

[14] R. Giryes, M. Elad, and A. M. Bruckstein: Sparsity based methods for overparameterized variational problems. SIAM J. Imaging Sci. *8* (2015), 3, 2133–2159. DOI:10.1137/140998585

[15] GN Nettest: Understanding OTDR. GN Nettest (2000).

[16] G. H. Golub, C. F. V. Loan: Matrix Computations. Third edition. Johns Hopkins University Press 1996.

[17] S. J. Kim, K. Koh, S. Boyd, and D. Gorinevsky: $\ell_1$ trend filtering. SIAM Rev. *51* (2009), 2, 339–360. DOI:10.1137/070690274

[18] N. Komodakis and J. Pesquet: Playing with duality: An overview of recent primal-dual approaches for solving large-scale optimization problems. IEEE Signal Process. Magazine *32* (2015), 6, 31–54. DOI:10.1109/msp.2014.2377273

[19] M. Kowalski: Sparse regression using mixed norms. Applied Comput. Harmonic Analysis *27* (2009), 3, 303–324. DOI:10.1016/j.acha.2009.05.006

[20] M. Kowalski and B. Torrésani: Structured Sparsity: from Mixed Norms to Structured Shrinkage. In: SPARS'09 – Signal Processing with Adaptive Sparse Structured Representations (R. Gribonval, ed.), Inria Rennes – Bretagne Atlantique 2009, pp. 1–6.

[21] G. Kutyniok and W. Q. Lim: Compactly supported shearlets are optimally sparse. J. Approx. Theory *163* (2011), 11, 1564–1589.

[22] D. Levin: Between moving least-squares and moving least-$\ell_1$. BIT Numerical Math. *55* (2015), 3, 781–796. DOI:10.1007/s10543-014-0522-0

[23] J. Neubauer and V. Veselý: Change point detection by sparse parameter estimation. Informatica *22* (2011), 1, 149–164. DOI:10.1049/pbce065e_ch7

[24] M. Novosadová and P. Rajmic: Piecewise-polynomial curve fitting using group sparsity. In: Proc. 8th International Congress on Ultra Modern Telecommunications and Control Systems, Lisabon 2016, pp. 317–322. DOI:10.1109/icumt.2016.7765379

[25] M. Novosadová and P. Rajmic: Piecewise-polynomial signal segmentation using reweighted convex optimization. In: Proc. 40th International Conference on Telecommunications and Signal Processing (TSP), Barcelona 2017, pp. 769–774. DOI:10.1109/tsp.2017.8076092

[26] M. Šorel and F. Šroubek: Fast convolutional sparse coding using matrix inversion lemma. Digital Signal Process. *55* (2016), 44–51. DOI:10.1016/j.dsp.2016.04.012

[27] N. Perraudin, D. I. Shuman, G. Puy, and P. Vandergheynst: Unlocbox A Matlab convex optimization toolbox using proximal splitting methods (2014).

[28] T. Pock: Fast Total Variation for Computer Vision. Dissertation Thesis, Graz University of Technology 2008.

[29] P. Rajmic: Exact risk analysis of wavelet spectrum thresholding rules. In: Electronics, Circuits and Systems, 2003. ICECS 2003. Proc. 10th IEEE International Conference *2* (2003), pp. 455–458. DOI:10.1109/icecs.2003.1301820

[30] P. Rajmic and M. Novosadová: On the limitation of convex optimization for sparse signal segmentation. In: Proc. 39th International Conference on Telecommunications and Signal Processing, Brno University of Technology 2016, pp. 550–554. DOI:10.1109/tsp.2016.7760941

[31] I. W. Selesnick, S. Arnold, and V. R. Dantham: Polynomial smoothing of time series with additive step discontinuities. IEEE Trans. Signal Process. *60* (2012), 12, 6305–6318. DOI:10.1109/tsp.2012.2214219

[32] S. Shem-Tov, G. Rosman, G. Adiv, R. Kimmel, and A. M. Bruckstein: Innovations for Shape Analysis. Chap. On Globally Optimal Local Modeling: From Moving Least Squares to Over-parametrization. In: Mathematics and Visualization, Springer 2012, pp. 379–405. DOI:10.1007/978-3-642-34141-0_17

[33] J. L. Starck, E. J. Candes, and D. L. Donoho: The curvelet transform for image denoising. IEEE Trans. Image Process. *11* (2002), 6, 670–684. DOI:10.1109/tip.2002.1014998

[34] R. Tibshirani: Regression shrinkage and selection via the LASSO. J. Royal Statist. Soc. Ser. B (Methodological) *58* (1996), 1, 267–288.

[35] R. J. Tibshirani: Adaptive piecewise polynomial estimation via trend filtering. Ann. Statist. *42* (2014), 1, 285–323. DOI:10.1214/13-aos1189

[36] J. Tropp: Greed is good: Algorithmic results for sparse approximation. IEEE Trans. Inform. Theory *50* (2004), 2231–2242. DOI:10.1109/tit.2004.834793

[37] B. Zhang, J. Geng, and L. Lai: Multiple change-points estimation in linear regression models via sparse group lasso. IEEE Trans. Signal Process. *63* (2015), 9, 2209–2224. DOI:10.1109/tsp.2015.2411220

*Pavel Rajmic, Signal Processing Laboratory (SPLab), Dept. of Telecommunications, Faculty of Electrical Engineering and Communication, Brno University of Technology, Technická 12, 616 00 Brno. Czech Republic.*
*e-mail: rajmic@feec.vutbr.cz*

*Michaela Novosadová, Signal Processing Laboratory (SPLab), Dept. of Telecommunications, Faculty of Electrical Engineering and Communication, Brno University of Technology, Technická 12, 616 00 Brno. Czech Republic.*
*e-mail: novosadova@phd.feec.vutbr.cz*

*Marie Daňková, Signal Processing Laboratory (SPLab), Dept. of Telecommunications, Faculty of Electrical Engineering and Communication, Brno University of Technology, Technická 12, 616 00 Brno. Czech Republic.*
*e-mail: m.dankova@phd.feec.vutbr.cz*