

RESEARCH

Open Access



# Orthogonality is superiority in piecewise-polynomial signal segmentation and denoising

Michaela Novosadová<sup>1</sup>, Pavel Rajmic<sup>1\*</sup>  and Michal Šorel<sup>2</sup>

## Abstract

Segmentation and denoising of signals often rely on the polynomial model which assumes that every segment is a polynomial of a certain degree and that the segments are modeled independently of each other. Segment borders (breakpoints) correspond to positions in the signal where the model changes its polynomial representation. Several signal denoising methods successfully combine the polynomial assumption with sparsity. In this work, we follow on this and show that using orthogonal polynomials instead of other systems in the model is beneficial when segmenting signals corrupted by noise. The switch to orthogonal bases brings better resolving of the breakpoints, removes the need for including additional parameters and their tuning, and brings numerical stability. Last but not the least, it comes for free!

**Keywords:** Signal segmentation, Signal smoothing, Signal approximation, Denoising, Piecewise polynomials, Orthogonality, Sparsity, Proximal splitting, Convex optimization

## 1 Introduction

Polynomials are an essential instrument in signal processing. They are indispensable in theory, as in the analysis of signals and systems [1] or in signal interpolation and approximation [2, 3], but they have been used also in specialized application areas such as blind source separation [4], channel modeling and equalization [5], to name a few. Orthonormal polynomials often play a special role [2, 6].

Segmentation of signals is one of the important applications in digital signal processing, while the most prominent sub-area is the segmentation of images. A plethora of methods exists which try to determine individual non-overlapping parts of the signal. The neighboring segments should be identified such that they contrast in their “character.” For digital signal processing, such a vague word has to be mathematically expressed in terms of signal features, which then differ from segment to segment. As examples, the segments could differ in their level, statistics, frequency content, texture properties, etc. In this article, we

rely on the assumption of smoothness of individual segments, which means that segments can be distinguished by their respective underlying polynomial description. The point in signal where the character changes is called a breakpoint, i.e., a breakpoint indicates the location of segment border. The features involved in the segmentation are chosen or designed a priori (i.e., model-based class), while the other class of methods aims at learning discriminative features from the training data [7, 8].

Within the first of the two classes, i.e., within approaches based on modeling, one can distinguish explicit and implicit types of models. In the “explicit” type, the signal is modeled such that it is a composition of sub-signals which often can be expressed analytically [9–16]. In the “implicit” type of models, the signal is characterized by features that are derived from the signal by using an operator [17–21]. The described differences are in an analogy to the “synthesis” and “analysis” approaches, respectively, recognized in the sparse signal processing literature [22, 23]. Although the two types of models are different in their nature, connections can be found, for example, the recent article [24] showing the relationship between splines and generalized total variation regularization or [21] discussing the relationship between “trend filtering” and spline-based smoothers.

\*Correspondence: [rajmic@vutbr.cz](mailto:rajmic@vutbr.cz)

<sup>1</sup>Signal Processing Laboratory (SPLab), Brno University of Technology, Technická 12, 616 00 Brno, Czech Republic

Full list of author information is available at the end of the article

Note that signal denoising and segmentation often rely on similar or even identical models. Indeed, when borders of segments are found, denoising can be easily done as postprocessing. Conversely, the byproducts of denoising can be used to detect segment borders. This paradigm is also true for our model, which can provide segmentation and signal denoising/approximation at the same time. As examples of other works that aim at denoising but can be used for segmentation as well, we cite [19, 20, 25, 26].

The method described in this article belongs to the “explicit” type of models. We work with noisy one-dimensional signals, and our underlying model assumes that individual segments can be well approximated by polynomials. The number of segments is supposed to be much lower than the number of signal samples—this natural assumption at the same time justifies the use of sparsity measures involved in segment identification. The model and algorithm presented for 1D in this article can be easily generalized to a higher dimension. For example, images are commonly modeled as piecewise smooth 2D-functions [27–31].

In [9, 13, 15], the authors build explicit signal segmentation/denoising models based on the standard polynomial basis  $\{1, t, t^2, \dots, t^K\}$ . In our previous articles, e.g., [11, 32], we used this basis as well. This article shows that modeling with orthonormal bases instead of the standard basis (which is clearly non-orthogonal) brings significant improvement in detection of the signal breakpoints and thus in the eventual denoising performance. It is worth noting that this improvement comes virtually for free, since the cost of generating an orthonormal basis is negligible compared to the cost of the algorithm which finds, in the iterative fashion, the numerical solution with such a basis fixed.

Worth to note that the method closest to ours is the one from [9], which was actually the initial inspiration of our work in the discussed direction. Similar to us, the authors of [9] combine sparsity, overcompleteness, and a polynomial basis; however, they approximate the solution to the model by greedy algorithms, while we rely on convex relaxation techniques. The other, above-cited methods do not exploit overcompleteness. Out of those, an interesting study [21] is similar to our model in that it allows piecewise polynomials of arbitrary (fixed) degree; however, it can be shown that their model does not allow jumps in signals, while our model does. This makes a significant difference, as will be shown later in the article.

The article is structured as follows: Section 2 introduces the mathematical model of segmentation/denoising, and it suggests the eventual optimization problem. The numerical solution to this problem by the proximal algorithm is described in Section 3. Finally, Sections 4 and 5 provide the description of experiments and analyze the results.

## 2 Problem formulation

In continuous time, a polynomial signal of degree  $K$  can be written as a linear combination of basis polynomials:

$$y(t) = x_0 p_0(t) + x_1 p_1(t) + \dots + x_K p_K(t), \quad t \in \mathbb{R}, \quad (1)$$

where  $x_k$ ,  $k = 0, \dots, K$ , are the expansion coefficients in such a basis. If the standard basis is used, i.e.,

$$p_0(t) = 1, p_1(t) = t, \dots, p_K(t) = t^K, \quad (2)$$

the respective scalars  $x_k$  correspond to the intercept, slope, etc.

Assume a discrete-time setting and limit the time instants to  $n = 1, \dots, N$ . Elements of a polynomial signal are then represented as

$$y[n] = x_0 p_0[n] + x_1 p_1[n] + \dots + x_K p_K[n], \quad n = 1, \dots, N. \quad (3)$$

In this formula, the signal is constructed by a linear combination of sampled polynomials.

Assuming the polynomials  $p_k$ ,  $k = 0, \dots, K$ , are fixed, every signal given by (3) is determined *uniquely* by the set of coefficients  $\{x_k\}$ . In contrast to this, we introduce a time index also to these coefficients, allowing them to change in time:

$$y[n] = x_0[n] p_0[n] + x_1[n] p_1[n] + \dots + x_K[n] p_K[n], \quad n = 1, \dots, N. \quad (4)$$

This may seem meaningless at this moment; however, such an excess of parameters will play a principal role shortly. It will be convenient to write this relation in a more compact form, for which we need to introduce the notation

$$\mathbf{y} = \begin{bmatrix} y[1] \\ \vdots \\ y[N] \end{bmatrix}, \quad \mathbf{x}_k = \begin{bmatrix} x_k[1] \\ \vdots \\ x_k[N] \end{bmatrix}, \quad \mathbf{P}_k = \begin{bmatrix} p_k[1] & & 0 \\ & \ddots & \\ 0 & & p_k[N] \end{bmatrix} \quad (5)$$

for  $k = 0, \dots, K$ . After this, we can write

$$\mathbf{y} = \mathbf{P}_0 \mathbf{x}_0 + \dots + \mathbf{P}_K \mathbf{x}_K \quad (6)$$

or even more shortly

$$\mathbf{y} = \mathbf{P} \mathbf{x} = [\mathbf{P}_0 | \dots | \mathbf{P}_K] \begin{bmatrix} \mathbf{x}_0 \\ \vdots \\ \mathbf{x}_K \end{bmatrix}, \quad (7)$$

where the length of the vector  $\mathbf{x}$  is  $(K + 1)$  times  $N$  and  $\mathbf{P}$  is a fat matrix of size  $N \times (K + 1)N$ .

Such a description of signal of dimension  $N$  is obviously overcomplete—there are  $(K + 1)N$  parameters to characterize it. Nevertheless, assume now that  $\mathbf{y}$  is a *piecewise* polynomial and that it consists of  $S$  independent segments. Each segment  $s \in \{1, \dots, S\}$  is then described by  $K + 1$  polynomials. In our notation, this can be achieved by letting vectors  $\mathbf{x}_k$  be *constant* within time indexes belonging to particular segments. (The polynomials in  $\mathbf{P}$  are fixed). Figure 1 shows an illustration. The reason for not using a single number describing each segment is that the positions of the segment breakpoints are unknown and will be subject to search.

Following the above argumentation, if  $\mathbf{x}_k$  are piecewise constant, the finite difference operator  $\nabla$  applied to vectors  $\mathbf{x}_k$  produces sparse vectors. Operator  $\nabla$  computes simple differences of each pair of adjacent elements in the vector, i.e.,  $\nabla : \mathbb{R}^N \mapsto \mathbb{R}^{N-1}$  such that  $\nabla \mathbf{z} = [z_2 - z_1, \dots, z_N - z_{N-1}]^T$ . Actually, not only  $\nabla$  applied to each parameterization vector produces  $S - 1$  nonzeros at maximum, but also the nonzero components of each  $\nabla \mathbf{x}_k$  occupy the same positions across  $k = 0, \dots, K$ .

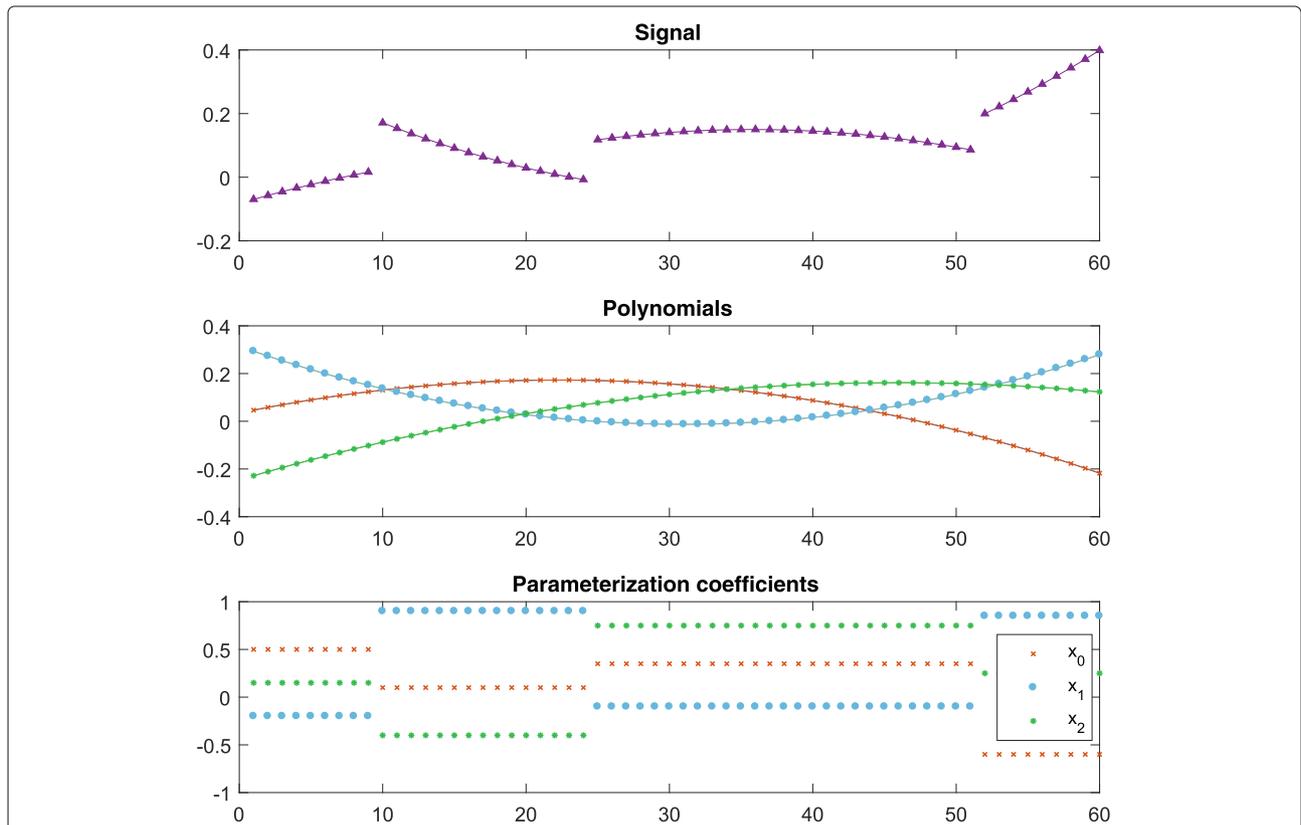
Together with the assumption that the observed signal is corrupted by an i.i.d. Gaussian noise, it motivates us to formulate the denoising/segmentation problem as finding

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \|\text{reshape}(\mathbf{L}\mathbf{x})\|_{21} \text{ s.t. } \|\mathbf{y} - \mathbf{P}\mathbf{W}\mathbf{x}\|_2 \leq \delta. \tag{8}$$

In this optimization program,  $\mathbf{W}$  is the square diagonal matrix of size  $(K + 1)N$  that enables us to adjust the lengths of vectors placed in  $\mathbf{P}$  and operator  $\mathbf{L}$  represents the stacked differences such that

$$\mathbf{L} = \begin{bmatrix} \nabla & \cdots & 0 \\ & \ddots & \\ 0 & \cdots & \nabla \end{bmatrix}, \quad \mathbf{L}\mathbf{x} = \begin{bmatrix} \nabla \mathbf{x}_0 \\ \vdots \\ \nabla \mathbf{x}_K \end{bmatrix}. \tag{9}$$

The operator  $\text{reshape}()$  takes the stacked vector  $\mathbf{L}\mathbf{x}$  to the form of a matrix with disjoint columns:



**Fig. 1** Illustration of the signal parameterization. The top plot shows four segments of a piecewise-polynomial signal (both the samples and the underlying continuous-time model); each segment is of the second order. The middle plot are the three basis polynomials, i.e., the diagonals of matrices  $\mathbf{P}_k$  (in this particular case, the respective sampled vectors are mutually orthonormal, actually). The parameterization coefficients shown in the bottom plot are vectors  $\mathbf{x}_0, \mathbf{x}_1$ , and  $\mathbf{x}_2$ . Notice that infinitely many other combinations of values in  $\mathbf{x}_0, \mathbf{x}_1$ , and  $\mathbf{x}_2$  generate the same signal, but we show the piecewise-constant case which is of the greatest interest for our study

$$\text{reshape}(\mathbf{L}\mathbf{x}) = [\nabla\mathbf{x}_0 | \dots | \nabla\mathbf{x}_K]. \quad (10)$$

It is necessary to organize the vectors in such a way for the purpose of the  $\ell_{21}$ -norm which is explained below.

The first term of (8) is the penalty. Piecewise-constant vectors  $\mathbf{x}_k$  suggest that these vectors are sparse under the difference operation  $\nabla$ . As an acknowledged substitute of the true sparsity measure, the  $\ell_1$ -norm is widely used [33, 34]. Since the vectors should be *jointly* sparse, we utilize the  $\ell_{21}$ -norm [35] that acts on a matrix  $\mathbf{Z}$  with  $p$  rows and is formally defined by

$$\begin{aligned} \|\mathbf{Z}\|_{21} &= \left\| \left[ \|\mathbf{Z}_{1,:}\|_2, \|\mathbf{Z}_{2,:}\|_2, \dots, \|\mathbf{Z}_{p,:}\|_2 \right] \right\|_1 \\ &= \|\mathbf{Z}_{1,:}\|_2 + \dots + \|\mathbf{Z}_{p,:}\|_2, \end{aligned} \quad (11)$$

i.e., the  $\ell_2$ -norm is applied to the particular rows of  $\mathbf{Z}$  and the resulting vector is measured by the  $\ell_1$ -norm. Such a penalty promotes sparsity across matrix' rows, and therefore, the  $\ell_{21}$ -norm enforces the nonzero components in the matrix to lie on the same rows.

The second term in (8) is the data fidelity term. The Euclidean norm reflects the fact that gaussianity of the noise is assumed. The level of the error is required to fall below  $\delta$ . Finally, vector  $\hat{\mathbf{x}}$  contains the achieved optimizers.

When standard polynomial basis  $\{1, t, \dots, t^K\}$  is used for the definition of  $\mathbf{P}$ , the high-order components blow up so rapidly that it brings two problems:

First, the difference vectors follow the scale of the respective polynomials. In the absence of normalization, i.e., when  $\mathbf{W}$  is identity, this is not fair with respect to the  $\ell_{21}$ -norm, since no polynomial should be preferred. In this regard, the polynomials should be "normalized" such that  $\mathbf{W}$  contains the reciprocal of  $\ell_2$ -norms of the respective polynomials. It is worth noting that in our former work, in particular in [12], we basically used model (8), but with the difference that there has been no weighting matrix and we used  $\mathbf{L} = \text{diag}(\tau_0\nabla, \dots, \tau_K\nabla)$  instead of  $\mathbf{L} = \text{diag}(\nabla, \dots, \nabla)$ , cf. (9). Finding suitable values of  $\tau_k$  has been a demanding trial-and-error process. In this perspective, simple substitution  $\mathbf{W}\mathbf{x} \rightarrow \mathbf{x}$  brings us in fact to the model from [12], and we see that  $\tau_k$  should correspond to the norms of the respective polynomial. However, it still holds true that manual adjustments of these parameters can increase the success rate of the breakpoint detection, as they depend, unfortunately, on the signal itself (recall that a part of a signal can correspond to locally high parameterization values while other part does not). This is however out of scope of this article.

Second, there is the numerics issue, meaning that the algorithms (see below) used to find the solution  $\hat{\mathbf{x}}$  failed due to the too wide range of the processed values. However, for short signals (like  $N \leq 500$ ), this problem was

solved by taking the time instants not as integers, but linearly spaced values from  $1/N$  to 1, as the authors of [9] did.

This article shows that the simple idea of shifting to orthonormal polynomials solves the two problems with no extra requirements. At the same time, orthonormal polynomials result in better detection of the breakpoints.

One may also think of an alternative, unconstrained formulation of the problem:

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \left\| \text{reshape}(\mathbf{L}\mathbf{x}) \right\|_{21} + \frac{\lambda}{2} \left\| \mathbf{y} - \mathbf{P}\mathbf{W}\mathbf{x} \right\|_2. \quad (12)$$

This formulation is equivalent to (8) in the sense that to a given  $\delta$ , there exists  $\lambda$  such that the optima are identical. However, the constrained form is preferable since changing the weight matrix  $\mathbf{W}$  does not induce any change in  $\delta$ , in opposite to a possible shift in  $\lambda$  in (12).

### 3 Algorithms

We utilize the so-called proximal splitting methodology for solving optimization problem (8). Proximal algorithms (PA) are algorithms suitable for finding minimum of a sum of convex functions. Proximal algorithms perform iterations involving simple computational tasks such as evaluation of gradient or/and proximal operators related to the individual functions.

It is proven that under mild conditions, PA provide convergence. The speed of convergence is influenced by properties of the functions involved and by the parameters used in the algorithms.

#### 3.1 Condat algorithm solving (8)

The generic Condat algorithm (CA) [36, 37] represents one possibility for solving problems of type

$$\text{minimize } h_1(\mathbf{L}_1\mathbf{x}) + h_2(\mathbf{L}_2\mathbf{x}), \quad (13)$$

over  $\mathbf{x}$ , where functions  $h_1$  and  $h_2$  are convex and  $\mathbf{L}_1$  and  $\mathbf{L}_2$  are linear operators. In our paper [12], we have compared two variants of CA; in the current work, we utilize the variant that is easier to implement—it does not require a nested iterative projector.

To connect (13) with (8), we assign  $h_1 = \|\cdot\|_{21}$ ,  $\mathbf{L}_1 = \text{reshape}(\mathbf{L}\cdot)$ ,  $h_2 = \iota_{\{\mathbf{z}: \|\mathbf{y}-\mathbf{z}\|_2 \leq \delta\}}$  and  $\mathbf{L}_2 = \mathbf{P}\mathbf{W}$ , while  $\iota_C$  denotes the indicator function of a convex set  $C$ .

Algorithm solving (8) is described in Algorithm 1. Therein, two operators are involved: Operator  $\text{soft}_{\tau}^{\text{row}}(\mathbf{Z})$  takes matrix  $\mathbf{Z}$  and performs the row-wise group soft thresholding with threshold  $\tau$  on it, i.e., it maps each element of  $\mathbf{Z}$  such that

$$z_{ij} \mapsto \frac{z_{ij}}{\|\mathbf{Z}_{i,:}\|_2} \max(\|\mathbf{Z}_{i,:}\|_2 - \tau, 0). \quad (14)$$

Projector  $\text{proj}_{B_2(\mathbf{y}, \delta)}(\mathbf{z})$  finds the closest point to  $\mathbf{z}$  in the  $\ell_2$ -ball  $\{\mathbf{z} : \|\mathbf{y} - \mathbf{z}\|_2 \leq \delta\}$ ,

$$\mathbf{z} \mapsto \frac{\delta \mathbf{z}}{\max(\|\mathbf{z}\|_2, \delta)}. \quad (15)$$

All particular operations in Algorithm 1 are quite simple, and they are obtained in  $\mathcal{O}(N)$  time. It is worth emphasizing, however, that the number of iterations necessary to achieve convergence grows with the number of time samples  $N$ . A better notion of the computational cost is provided by Table 1. It shows that both the cost per iteration and the number of necessary iterations grow linearly, resulting in an overall  $\mathcal{O}(N^2)$  complexity of the algorithm. The cost of postprocessing (described in Section 3.2) is negligible compared to such a quantity of operations.

---

**Algorithm 1:** The Condat Algorithm solving (8)

---

**Input:**  $\mathbf{PW} : \mathbb{R}^{N(K+1)} \rightarrow \mathbb{R}^N$ ,  $\mathbf{y} \in \mathbb{R}^N$ ,  $\delta > 0$

**Output:**  $\hat{\mathbf{x}} = \mathbf{x}^{(i+1)}$

Set parameters  $\xi, \sigma > 0$  and  $\rho \in (0, 2)$ . Set initial primal variable  $\mathbf{x}^{(0)} \in \mathbb{R}^{N(K+1)}$ , set dual variables  $\mathbf{u}_1^{(0)} \in \mathbb{R}^{(N-1)(K+1)}$ ,  $\mathbf{u}_2^{(0)} \in \mathbb{R}^N$ .

**for**  $i = 0, 1, \dots$  **do**

$$\begin{cases} \bar{\mathbf{x}}^{(i+1)} = \\ \mathbf{x}^{(i)} - \xi \left( \mathbf{L}^\top \text{reshape}^\top \left( \mathbf{u}_1^{(i)} \right) + (\mathbf{PW})^\top \left( \mathbf{u}_2^{(i)} \right) \right) \\ \mathbf{x}^{(i+1)} = \rho \bar{\mathbf{x}}^{(i+1)} + (1 - \rho) \mathbf{x}^{(i)} \\ \mathbf{p}_1^{(i)} = \mathbf{u}_1^{(i)} + \sigma \text{reshape} \left( \mathbf{L} \left( 2\bar{\mathbf{x}}^{(i+1)} - \mathbf{x}^{(i)} \right) \right) \\ \bar{\mathbf{u}}_1^{(i+1)} = \mathbf{p}_1^{(i)} - \sigma \text{soft}_{1/\sigma}^{\text{row}} \left( \mathbf{p}_1^{(i)} / \sigma \right) \\ \mathbf{u}_1^{(i+1)} = \rho \bar{\mathbf{u}}_1^{(i+1)} + (1 - \rho) \mathbf{u}_1^{(i)} \\ \mathbf{p}_2^{(i)} = \mathbf{u}_2^{(i)} + \sigma \mathbf{PW} \left( 2\bar{\mathbf{x}}^{(i+1)} - \mathbf{x}^{(i)} \right) \\ \bar{\mathbf{u}}_2^{(i+1)} = \mathbf{p}_2^{(i)} - \sigma \text{proj}_{B_2(\mathbf{y}, \delta)} \left( \mathbf{p}_2^{(i)} / \sigma \right) \\ \mathbf{u}_2^{(i+1)} = \rho \bar{\mathbf{u}}_2^{(i+1)} + (1 - \rho) \mathbf{u}_2^{(i)} \end{cases}$$

**return**  $\mathbf{x}^{(i+1)}$

---

Convergence of the algorithm is guaranteed when it holds  $\xi \sigma \|\mathbf{L}_1^\top \mathbf{L}_1 + \mathbf{L}_2^\top \mathbf{L}_2\| \leq 1$ . For the use of the inequality  $\|\mathbf{L}_1^\top \mathbf{L}_1 + \mathbf{L}_2^\top \mathbf{L}_2\| \leq \|\mathbf{L}_1\|^2 + \|\mathbf{L}_2\|^2$ , it is necessary to have the upper bound on the operator norms. The upper bound of  $\|\mathbf{L}_1\|$  is:

$$\begin{aligned} \|\mathbf{L}_1\|^2 = \|\mathbf{L}\|^2 &= \max_{\|\mathbf{x}\|_2=1} \|\mathbf{L}\mathbf{x}\|_2^2 = \max_{\|\mathbf{x}\|_2=1} \left\| \begin{bmatrix} \nabla \mathbf{x}_0 \\ \vdots \\ \nabla \mathbf{x}_K \end{bmatrix} \right\|_2^2 \\ &= \max_{\|\mathbf{x}\|_2=1} \left( \sum_{k=0}^K \|\nabla \mathbf{x}_k\|_2^2 \right) \\ &\leq \sum_{k=0}^K \left( \max_{\|\mathbf{x}\|_2=1} \|\nabla \mathbf{x}_k\|_2^2 \right) \\ &\leq \sum_{k=0}^K \|\nabla\|^2 \leq 4(K+1) \end{aligned} \quad (16)$$

and thus  $\|\mathbf{L}_1\| \leq 2\sqrt{K+1}$ . The operator norm of  $\mathbf{PW}$  satisfies  $\|\mathbf{PW}\|^2 = \|\mathbf{PW}\mathbf{W}^\top \mathbf{P}^\top\|$ , and thus, it suffices to find the maximum eigenvalue of  $\mathbf{PW}^2 \mathbf{P}^\top$ . Since  $\mathbf{PW}$  has the multi-diagonal structure (cf. relation (7)),  $\mathbf{PW}^2 \mathbf{P}^\top$  is diagonal, and in effect, it is enough to find the maximum on its diagonal. Altogether, the convergence is guaranteed when  $\xi \sigma (\max \text{diag}(\mathbf{PW}^2 \mathbf{P}^\top) + 4(K+1)) \leq 1$ .

### 3.2 Signal segmentation/denoising

Vectors  $\hat{\mathbf{x}}$  as the optimizers of problem (8) allow a means to estimate the underlying signal; it can be done simply by  $\hat{\mathbf{y}} = \mathbf{PW}\hat{\mathbf{x}}$ . However, this way we do not obtain the segment ranges. Second disadvantage of this approach is that the jumps are typically underestimated in size, which comes from the bias inherent to the  $\ell_1$  norm [38–40] as the part of the optimization problem.

The nonzero values in  $\nabla \hat{\mathbf{x}}_0, \dots, \nabla \hat{\mathbf{x}}_K$  indicate segment borders. In practice, it is almost impossible to achieve truly piecewise-constant optimizers [38] as in the model case in Fig. 1, and vectors  $\nabla \hat{\mathbf{x}}_k$  are crowded by small elements, besides larger values indicating possible segment borders. We apply a two-part procedure to obtain the segmented and denoised signal: the breakpoints are detected first, and then, each detected segment is denoised individually.

Recall that the  $\ell_{21}$ -norm cost promotes significant values in vectors  $\nabla \hat{\mathbf{x}}_k$  situated at the same positions. As a base for breakpoint detection, we gather  $\nabla \hat{\mathbf{x}}_k$ s to a single vector using the weighted  $\ell_2$ -norm according to the formula

$$\mathbf{d} = \sqrt{(\alpha_0 \nabla \hat{\mathbf{x}}_0)^2 + \dots + (\alpha_K \nabla \hat{\mathbf{x}}_K)^2}, \quad (17)$$

**Table 1** Time spent per iteration (in seconds) and the total number of iterations until convergence with respect to  $N$ , for an orthonormal polynomial base, fixed  $K = 2$

$N$	300	503	705	904	1106	1308	1509	1711	1913
100 iter. [s]	.05272	.05295	.06518	.07274	.08840	.09585	.10539	.11914	.12295
conver. [iter]	690	790	960	1090	1170	1290	1370	1470	1530

where  $\alpha_k = 1/\max(|\nabla\hat{\mathbf{x}}_k|)$  are positive factors serving to normalize the range of values in the parameterization vectors differences. The computations in (17) are elementwise.

The comparisons presented in this article will be concerned only with the detection of breakpoints, and thus, in our further analysis, we process no more than the vector  $\mathbf{d}$ . However, in case we would like to recover the denoised signal, we would proceed as in our former works [11, 12], where first a moving median filter is applied to  $\mathbf{d}$  and subtracted from  $\mathbf{d}$ , allowing to keep the significant values and at the same time to push small ones toward zero. Put simply, values larger than a selected threshold then indicate the breakpoints positions. The second step is denoising itself, which is done by least squares on each segment separately, using (any) polynomial basis of degree  $K$ .

#### 4 Experiment—does orthogonality help in signals with jumps?

The experiment has been designed to find out whether substituting non-orthogonal bases with the orthogonal ones reflects in emphasizing the positions of breakpoints when exploring the vector  $\mathbf{d}$ .

##### 4.1 Signals

As test signals, five piecewise quadratic signals ( $K = 2$ ) of length  $N = 300$  were randomly generated. They are generated such that they contain polynomial segments similar to the 1D test signals presented in [9]. All signals consist of six segments of random lengths. There are noticeable jumps in value between neighboring segments, which is the difference to the test signals in [9]. The noiseless

signals are denoted by  $\mathbf{y}_{\text{clean}}$  and examples are depicted in Fig. 2.

The signals have been corrupted by the Gaussian i.i.d. noise, resulting in signals  $\mathbf{y}_{\text{noisy}} = \mathbf{y}_{\text{clean}} + \boldsymbol{\epsilon}$  with  $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2)$ . With these signals, we can determine the signal-to-noise ratio (SNR), defined as

$$\text{SNR}(\mathbf{y}_{\text{noisy}}, \mathbf{y}_{\text{clean}}) = 20 \cdot \log_{10} \frac{\|\mathbf{y}_{\text{clean}}\|_2}{\|\mathbf{y}_{\text{noisy}} - \mathbf{y}_{\text{clean}}\|_2}. \quad (18)$$

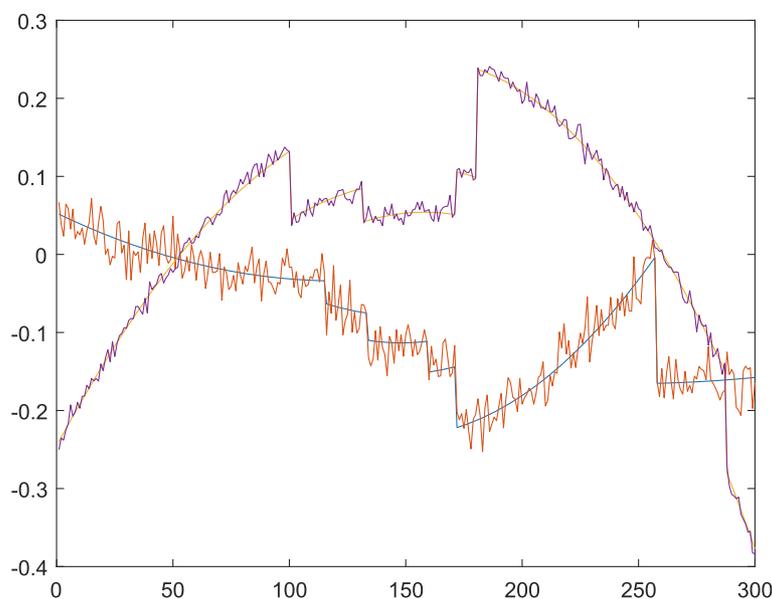
Five SNR values were prescribed for the experiment: 15, 20, 25, 30, and 35 dB. These numbers entered into the calculation of the respective noise standard deviation  $\sigma$  such that

$$\sigma = \frac{\|\mathbf{y}_{\text{clean}}\|_2}{\sqrt{N \cdot 10^{\frac{\text{SNR}}{10}}}}. \quad (19)$$

It is clear that the resulting  $\sigma$  is influenced by energy of the clean signal as well. For each signal and each SNR, 100 realizations of noise were generated, making a set of 2500 noisy signals in total.

##### 4.2 Bases

Since the test signals are piecewise quadratic, the bases subject to testing all consist of three linearly independent discrete-time polynomials. For the sake of this section, the three basis vectors can be viewed as the columns of the  $N \times 3$  matrix. The connection to problem (8) is that these vectors form the diagonals of the system matrix  $\mathbf{PW}$ . In the following, the  $N \times 3$  basis matrices will be distinguished by the letter indicating the means of their generation:



**Fig. 2** Example of two noiseless and noisy test signals used in the experiment. Signals of length  $N = 300$  consist of six segments of various length, with a perceptible jump between each two segments. The SNRs used for this illustrations were 25 and 15 dB

#### 4.2.1 Non-orthogonal bases (B)

Most of the papers that explicitly model the polynomials utilize directly the standard basis (2), which is clearly not orthogonal either in continuous nor discrete setting. The norms of such polynomials differ significantly as well. We generated 50 B bases using formula  $\mathbf{B} = \mathbf{SD}_1\mathbf{AD}_2$ . Here, the elements of the standard basis—the columns of  $\mathbf{S}$ —are first normalized using a diagonal matrix  $\mathbf{D}_1$ , then mixed using a random Gaussian matrix  $\mathbf{A}$  and finally dilated to different lengths using  $\mathbf{D}_2$  having uniformly random entries at the diagonal. This way we acquired 50 bases, which are non-orthogonal and non-normalized at the same time.

#### 4.2.2 Normalized bases (N)

Another set of 50 bases, the N bases, were obtained by simply normalizing the length of the B basis polynomials,  $\mathbf{N} = \mathbf{BD}_3$ . We want to find out whether this simple step helps in detecting the breakpoints.

#### 4.2.3 Orthogonal bases (O)

Orthogonal bases were obtained by orthogonalization of N bases. The process was as follows: A matrix  $\mathbf{N}$  was decomposed by the SVD, i.e.,

$$\mathbf{N} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top. \quad (20)$$

Matrix  $\mathbf{U}$  consists of three orthonormal columns of length  $N$ . The new orthonormal system is simply the matrix  $\mathbf{O} = \mathbf{U}$ .

One could doubt whether the new basis  $\mathbf{O}$  spans the same space as  $\mathbf{N}$  does. Since  $\mathbf{N}$  has full rank,  $\mathbf{\Sigma}$  contains three positive values on its diagonal. Because  $\mathbf{V}$  is also orthogonal, the answer to the above question is positive. A second question could be whether the new system is still consistent with any polynomial basis on  $\mathbb{R}$ . The answer is yes again, since both matrices  $\mathbf{N}$  and  $\mathbf{U}$  can be substituted by their continuous-time counterparts, thus generating the identical polynomial.

#### 4.2.4 Random orthogonal bases (R)

The last class consists of random orthogonal polynomial bases. The R bases were generated as follows: First, the SVD has been applied to the matrix  $\mathbf{N}$  as in (20), now symbolized using the subscripts,  $\mathbf{N} = \mathbf{U}_N\mathbf{\Sigma}_N\mathbf{V}_N^\top$ . Next, a random matrix  $\mathbf{A}$  of size  $3 \times 3$  was generated, each element of whose independently follows the Gauss distribution. This matrix is then decomposed to  $\mathbf{A} = \mathbf{U}_A\mathbf{\Sigma}_A\mathbf{V}_A^\top$ . The new basis  $\mathbf{R}$  is obtained as  $\mathbf{R} = \mathbf{U}_N\mathbf{U}_A$ . Note that since both matrices on the right hand side are orthonormal, the columns of  $\mathbf{R}$  form an orthonormal basis spanning the desired space. Elements of  $\mathbf{U}_A$  determine the linear combinations used in forming  $\mathbf{R}$ .

We generated 50 such random bases, meaning that in total 200 bases (B, N, O, R) were ready for the experiment.

#### 4.2.5 A note on other polynomial bases

One could think of using predefined polynomial bases as Chebychev or Legendre bases, for example. Note that such bases are defined in continuous time and are therefore orthogonal with respect to an integral scalar product [6]. Sampling such systems at equidistant time-points does not lead to orthogonal bases; actually when preparing this article, we found out that their orthogonalization via the SVD (as done above) significantly changes the course of the basis vectors. As far as we know, there are no predefined discrete-time orthogonal polynomial systems. In combination with the observation that neither the sampled nor the orthogonalized systems perform better than other non-ortho- or orthosystems, respectively, we did not include any such system in our experiments.

### 4.3 Experiment

The algorithm of breakpoint detection that we utilized in the experiments has been described in Section 3.2. We used formula (17) for computing the input vector. The Condat algorithm run for 2000 iterations which was sufficient in all cases. Three items were subject to vary within the experiments, configuring the problem (8):

- The input signal  $\mathbf{y}$ ,
- parameter  $\delta$  controlling the modeling error,
- the basis of polynomials  $\mathbf{PW}$   
(induced from the columns of matrices  $\mathbf{B}$ ,  $\mathbf{N}$ ,  $\mathbf{O}$ , or  $\mathbf{R}$ ).

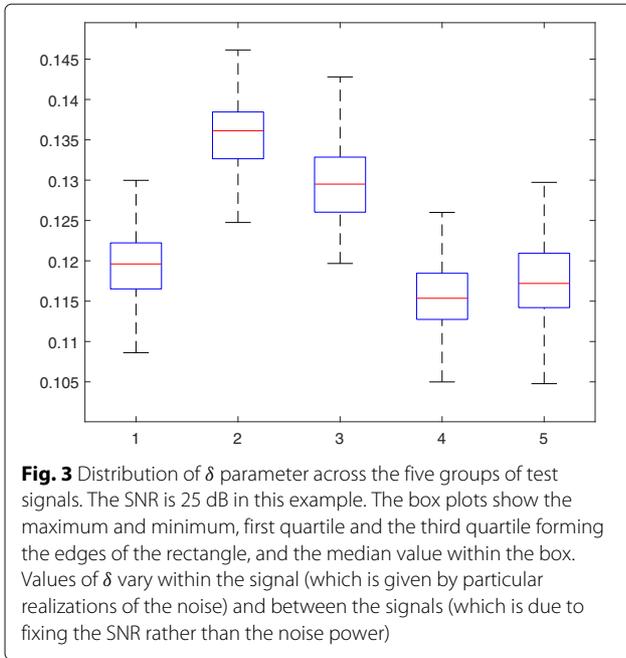
Each signal entered into calculation with each of the bases, making  $2500 \times 200$  experiments in total in signal breakpoints detection.

#### 4.3.1 Setting parameter $\delta$

For each of the 2500 noisy signals, the parameter  $\delta$  was calculated. Since both the noisy and clean signals are known in our simulation,  $\delta$  should be close to the actual  $\ell_2$  error caused by the noise. We found out that particular  $\delta$  leading to best breakpoint detection varies around the  $\ell_2$  error, however. For the purpose of our comparison, we fixed a universal value of  $\delta$  determined according to

$$\delta = \|\mathbf{y}_{\text{noisy}} - \mathbf{y}_{\text{clean}}\|_2 \cdot 1.05 \quad (21)$$

meaning that we allowed the model error to deviate from the ground truth by 5% at maximum. Figure 3 shows the distribution of values of  $\delta$ . For different signals,  $\delta$  is concentrated around a different quantity. This effect is due to the noise generation, wherein the resulting SNR (18) was set and fixed at first, while  $\delta$  is linked to the noise deviation  $\sigma$  that depends on the signal, cf. (19).



Note that in practice, however,  $\delta$  would have to take into account not only the (even unknown) noise level, but also the modeling error, since real signals do not follow the polynomial model exactly. A good choice of  $\delta$  unfortunately requires a trial process.

#### 4.4 Evaluation

The focus of the article is to study whether orthogonal polynomials lead to better breakpoint detection than the non-orthogonal polynomials. To evaluate this, several values that indicate the quality of breakpoint detection process were computed. These markers are based on vector  $\mathbf{d}$ .

But first, for each single signal in test, define two disjoint sets of indexes, chosen out of  $\{1, \dots, N\}$ :

*Highest values (HV):* Recall that each of the clean test signals contains five breakpoints. Note also that  $\mathbf{d}$  defined by (17) is nonnegative. The HV group thus gathers the indexes of the five values in  $\mathbf{d}$  that are likely to represent breakpoints. These five indexes are selected iteratively: At first, the largest value is chosen to belong to HV. Then, since it can happen that multiple high values sit next to each other, the two neighboring indexes to the left and two to the right are omitted from further consideration. The remaining four steps select the rest of the HV members in the same manner.

*Other values (OV):* The second group consists of the remaining indexes in  $\mathbf{d}$ . The indexes excluded during the HV selection are not considered in OV. This way, the number of elements in OV is 274 at least and 289 at most,

depending on the particular course of the HV selection process.

For each signal, the ratio of the averages of the values belonging to HV versus the average of the values in OV is computed; we denote this ratio AAR. We also computed the MMR indicator, which we define as the ratio of the minimum of values of HV to the maximum of the OV values. Both these indicators, and especially the MMR, should be as high as possible to enable safe recognition of the breakpoints.

The next parameter in evaluation was the number of correctly detected breakpoints (NoB). We are able to introduce NoB in our report since the true positions of the breakpoints are known. The breakpoint positions are not always found exactly, especially due to the influence of the noise (will be discussed later), and therefore, we consider the breakpoint as detected correctly if the indicated position lies within an interval of  $\pm$  two indexes from the ground truth.

In addition, classical mean square error (MSE) has been involved to complete the analysis. The MSE measures the average distance of the denoised signal from the noiseless original and is defined as

$$\text{MSE}(\mathbf{y}_{\text{denoised}}, \mathbf{y}_{\text{clean}}) = \frac{1}{N} \|\mathbf{y}_{\text{denoised}} - \mathbf{y}_{\text{clean}}\|_2^2. \quad (22)$$

As  $\mathbf{y}_{\text{denoised}}$ , two variants were considered: (a) the direct signal estimate computed as  $\hat{\mathbf{y}} = \mathbf{P}\mathbf{W}\hat{\mathbf{x}}$ , where  $\hat{\mathbf{x}}$  is the solution to (8) and (b) the estimate where the ordinary least squares have been used separately on each of the detected segments with a polynomial of degree two.

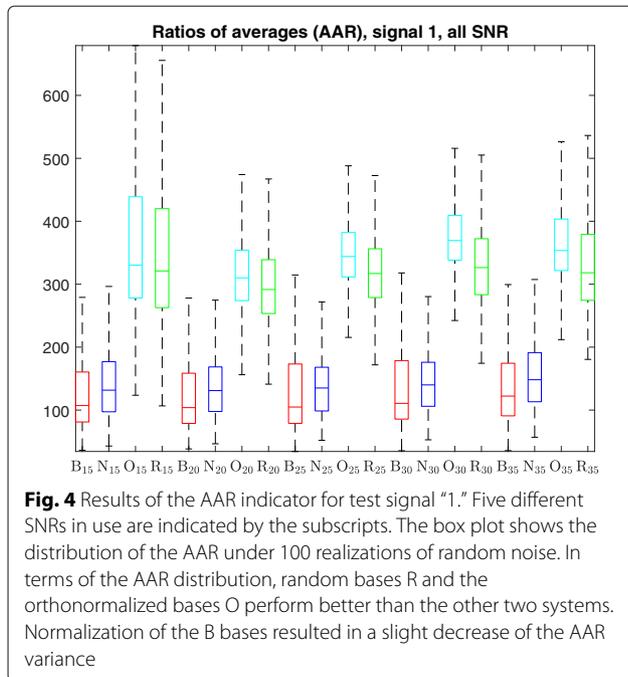
Note that approach (b) is an instance of the so-called debiasing methods, which is sometimes done in regularized regression, based on the a priori knowledge that the regularizer biases the estimate. As an example, debiasing is commonly done in LASSO estimation [39, 41], where the biased solution is used only to fix the sparse vector support and least squares are then used to make a better fit on the reduced subset of regressors, see also related works [12, 33, 42].

The results from approach a will be abbreviated “CA” in the following, meaning “Condat Algorithm”, and the results from the least squares adjustment by “LS”

#### 4.5 Results and discussion

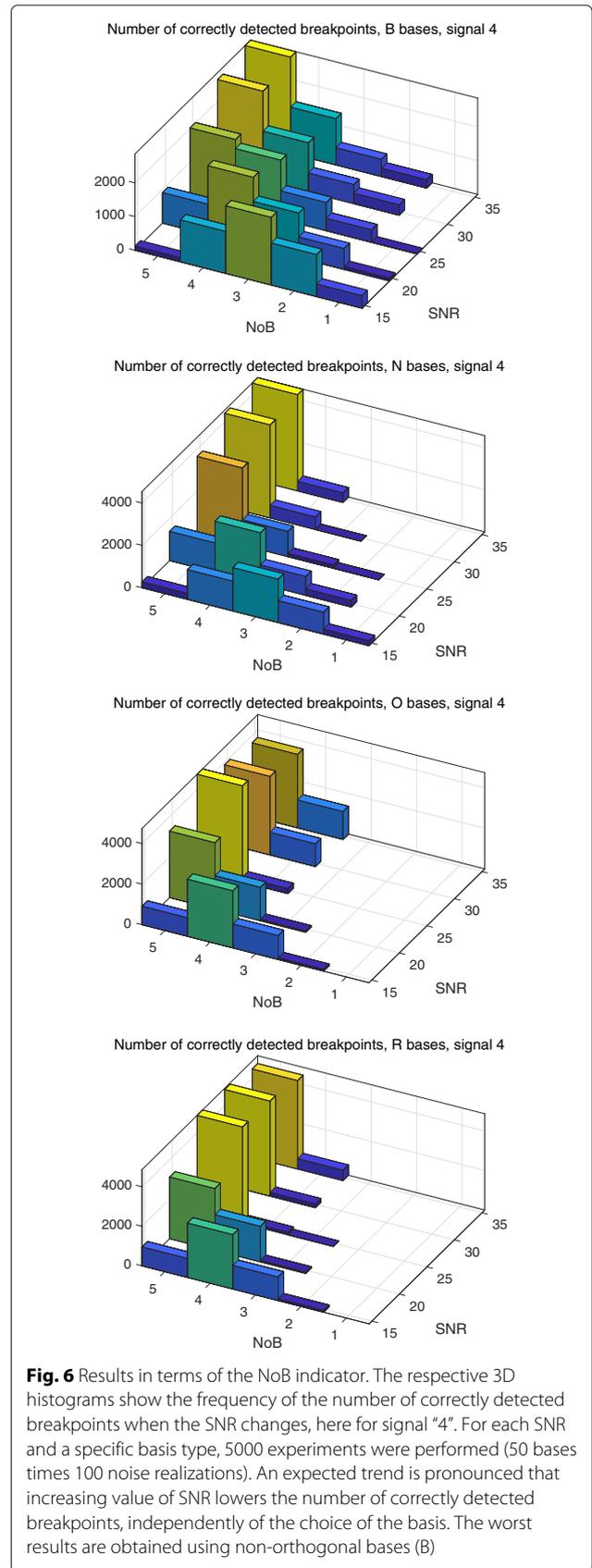
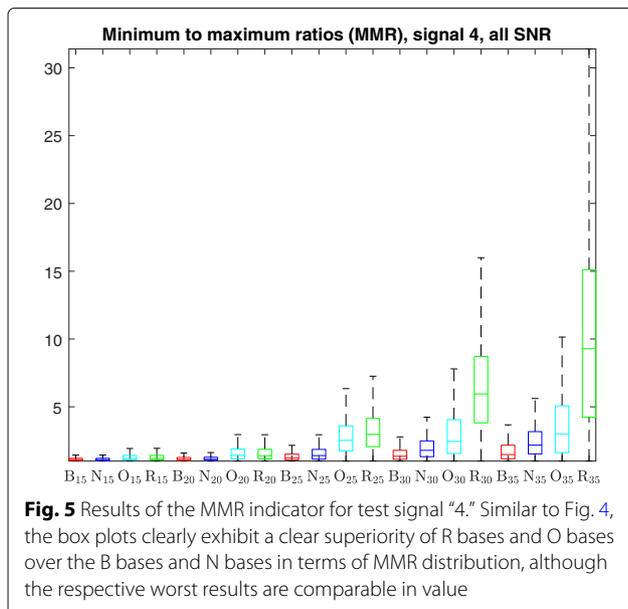
Using orthogonal bases reflects in significantly better results than working with non-orthogonal bases. The improvement can be observed in all parameters in consideration. The AAR, MMR, and NoB indicators increase with orthogonal bases and the MSE decreases.

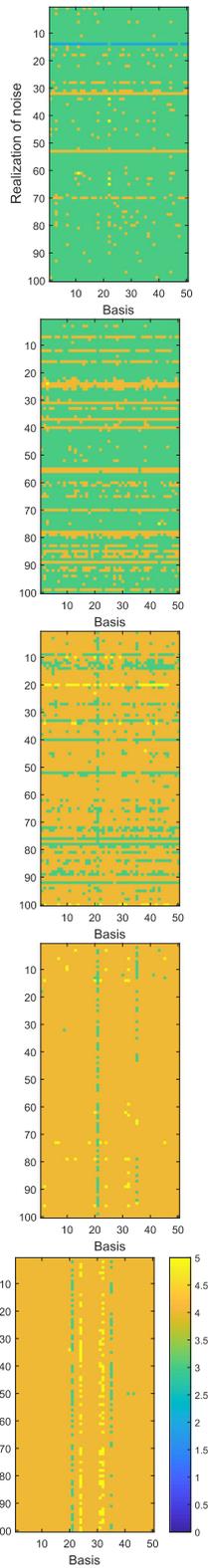
An example comparison of the three types of bases in terms of the AAR is depicted in Fig. 4. A larger AAR



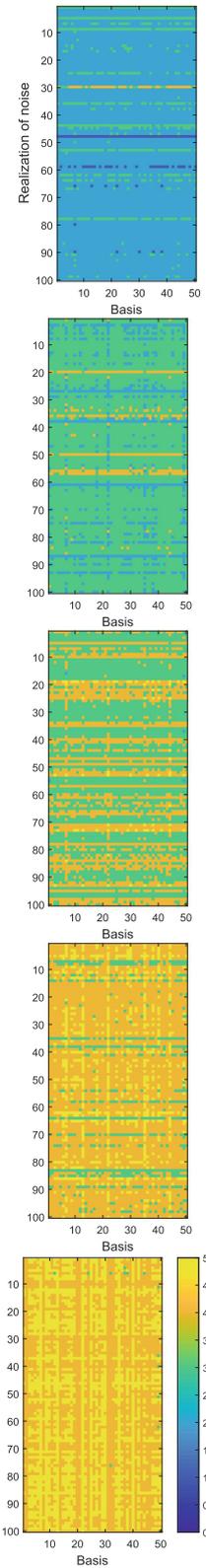
means that the averages of the HV and OV values, respectively, are more apart. Analogously, Fig. 5 shows an illustration of the performance in terms of the MMR. The MMR gets greater when the smallest value from HV is better separated from the greatest value from OV. This creates a means for correct detection of the breakpoints. From both figures, it is clear that R and O bases are preferable over N bases.

The reader has noticed that Figs. 4 and 5 do not show the comparison across all the test signals. The reason is





**Fig. 7** Number of correctly identified breakpoints (NoB) for different SNRs. From left to right 15, 20, 25, 30, 35 dB, signal “3.” In the horizontal direction are the fifty randomly generated orthobases (R bases). In the vertical direction are the hundred particular realizations of noise



**Fig. 8** Number of correctly identified breakpoints (NoB) for different SNRs. Analogously to Fig. 7, but now for signal “5”

that it is not possible to fairly fuse results for different signals, since the signal shape and size of the jumps influence the values of the considered parameters. Another reason is that the energy of the noise differs across signals, even when the SNR is fixed (see discussion of this effect above). However, looking at the complete list of figures which are available at the accompanying webpage<sup>1</sup>, the same trend is observed in all of the figures: the orthogonal(ized) bases perform better than the non-orthogonal bases. At the same time, there is no clear suggestion whether R bases are better than O bases; while Fig. 5 shows superiority of R bases, other plots at the website contain various results.

The NoB is naturally the ultimate criterion for measuring the quality of segmentation. Histograms of the NoB parameter for one particular signal are shown in Fig. 6. From this figure, as well as from the supplementary material at the webpage, we can conclude that B bases are beaten by N bases. Most importantly, the two orthogonal classes of bases (O, R) perform better than the N bases in a majority of cases (although one finds situation when the systems perform on par). Looking closer to obtain a final statement whether O bases or R bases are preferable, we can see that R bases usually provide better detection of breakpoints; however, the difference is very small. This might be the result of the test database being too small.

Does the distribution of NoB in Fig. 6 also suggest that some of the bases may perform better than others within the same class, when the signal and the SNR are fixed? It is not fair to make such a conclusion based on the histograms; histograms cannot reveal whether the effect on NoB is due to the particular realization of noise or it is due to differences between the bases, regardless of noise. Let us take some effort to find the answer to the question. Figures 7 and 8 show selected maps of NoB. It is clearly visible that for mild noise levels, there are bases that perform better than the others and that also a few bases perform significantly worse—in a uniform manner. In the low SNR regime, on contrary, the horizontal structures in the images prevail, meaning that specific noise shape takes over. This effect can be explained easily: the greater is the amplitude of the noise, the greater is the probability that an “undesirable” noise sample in the nearness of the breakpoint spoils its correct identification.

In practice, nevertheless, the signal to be denoised/segmented is given including the noise. In light of the presented NoB analysis (Figs. 7 and 8 in particular), it means that (especially) when SNR is high, it may be beneficial to run the optimization task (8) multiple times, i.e., with different bases, fusing the particular results for a final decision.

The last measure of performance is the MSE. First, Fig. 9 presents an example of denoising using the direct and least squares approach (those are described in Section 4.4).

Figures 10 and 11 show successful and distracting results in terms of MSE, respectively. While with signals “1” to “4,” orthobases improve MSE, it is not the case of signal “5.” It is interesting to note that signal “5” does not exhibit great performance in terms of the other indicators (AAR, MMR, NoB) neither.

#### 4.6 Software

The experiment has been done in MATLAB (2017a) on a PC with Intel i7 processor and with 16 GB of RAM. For some proximal algorithm components we benefited from using the flexible UnLocBox toolbox [43]. The code related to the experiments is available via the mentioned webpage.

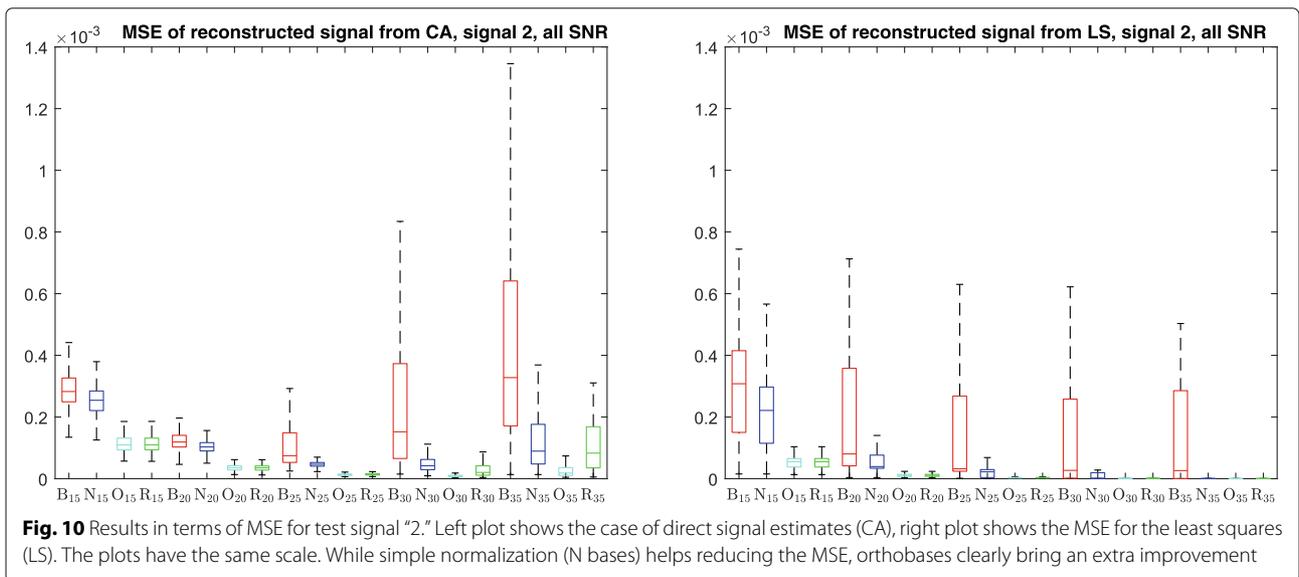
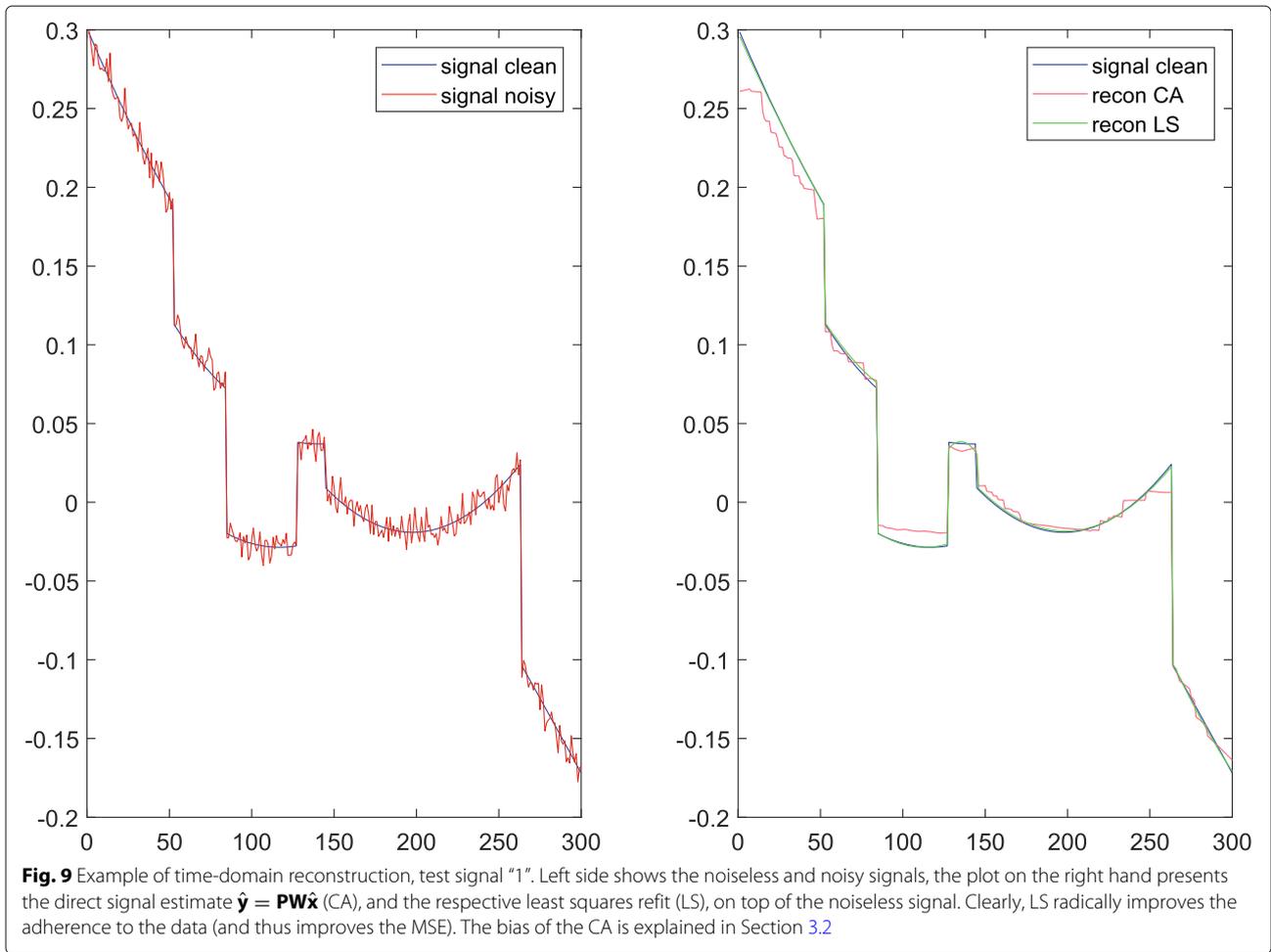
It is computationally cheap to generate an orthogonal polynomial system, compared to the actual cost of iterations in the numerical algorithm. For  $N = 300$ , convergence has been achieved after performing 2000 iterations of Algorithm 1. While one iteration takes about 0.5 ms, generation of one orthonormal basis (dominated by the SVD) takes up to 1 ms.

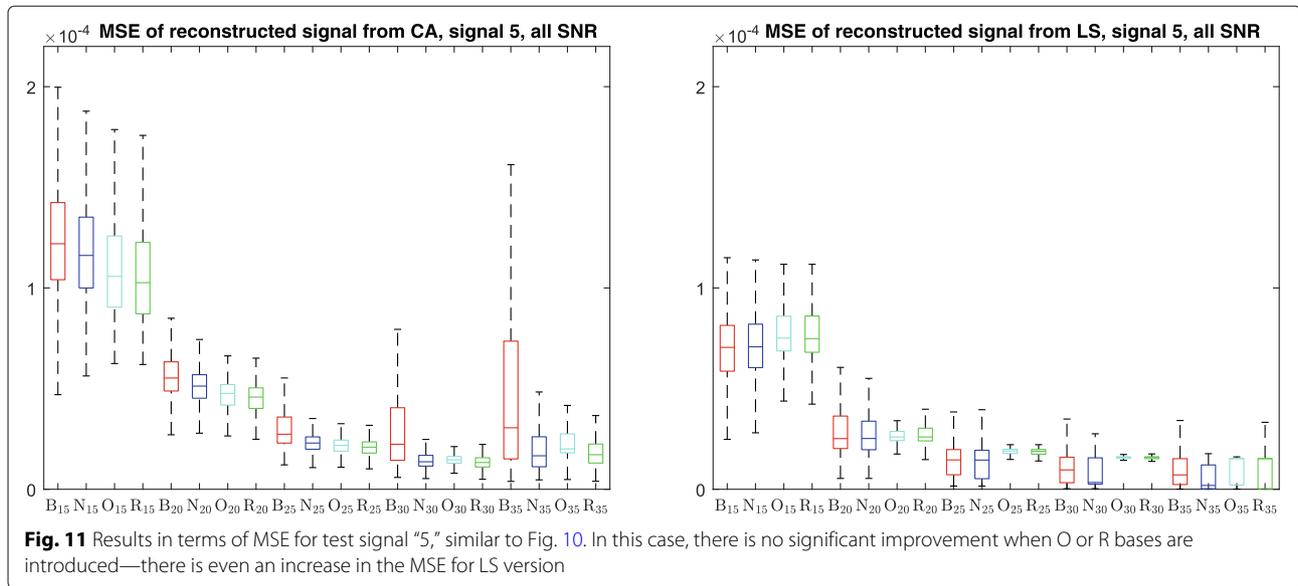
### 5 Experiment—the effect of jumps

Another experiment has been performed focusing on the sensitivity of the breakpoint detection in relation to the size of the jumps in signal. For this study, we utilized a single signal, depicted in blue in Fig. 12; the signal length was again of length  $N = 300$ . It contains five segments of similar length, and quadratic polynomials are used, similar to test signals in [9]. The signal is designed such that there are no jumps on the segment borders. Nine new signals were generated from this signal in a way that segments two and four were raised up by a constant value; nine constants uniformly ranging from 5 to 45 were applied. Each signal was corrupted by gaussian noise 100 times independently, with 10 different variances. This way, 10 000 signals were available in this study in total.

As the polynomial systems, three O bases and three B bases were randomly chosen out of the set of 50 of the same kind from the experiment above. We ran the optimization problem (8) on the signals with  $\delta$  set according to (21). Each solution was then transformed to the vector  $\mathbf{d}$  (see formula (17)). Four largest elements (since there are five true segments) in  $\mathbf{d}$  were selected and their positions were considered the estimated breakpoints. Evaluation of correctly detected breakpoints (NoB) was performed as in the above experiment, with the difference that  $\pm 4$  samples from the true position were accepted as a successful detection.

Figure 13 shows the average results. It is clear that the presence of even small jumps prioritize the use of O bases, while, interestingly, in case of little or no jumps, B bases perform slightly better (note, however, that both



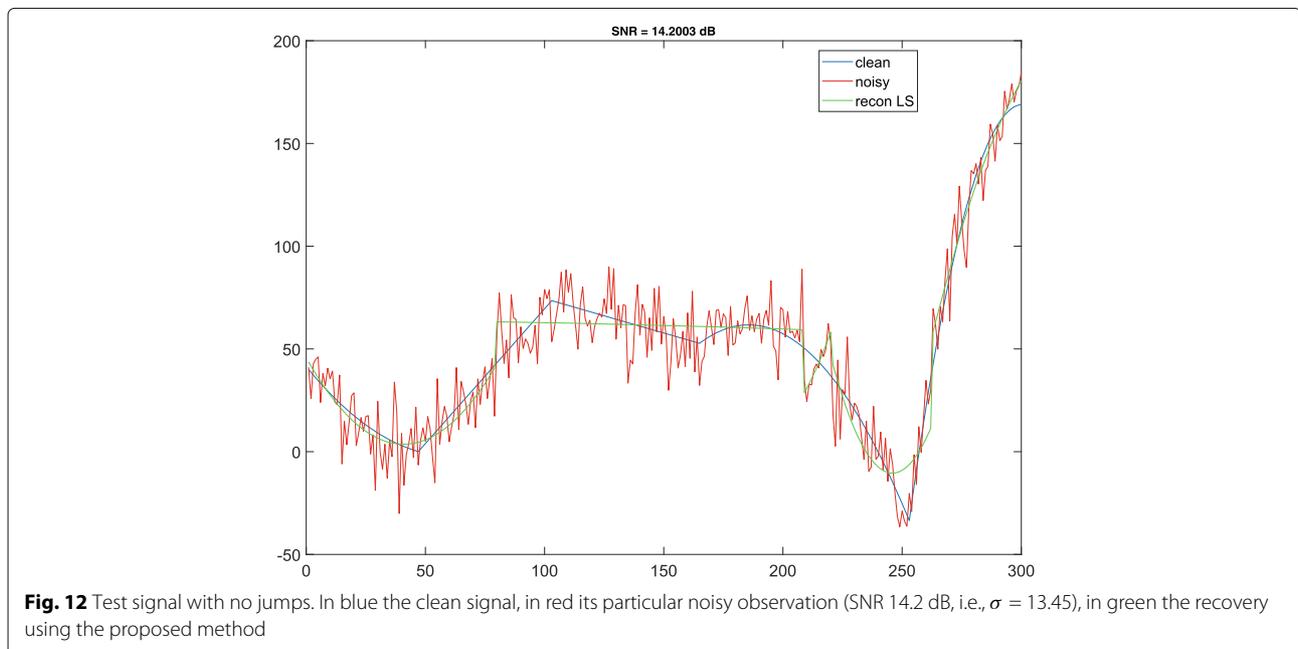


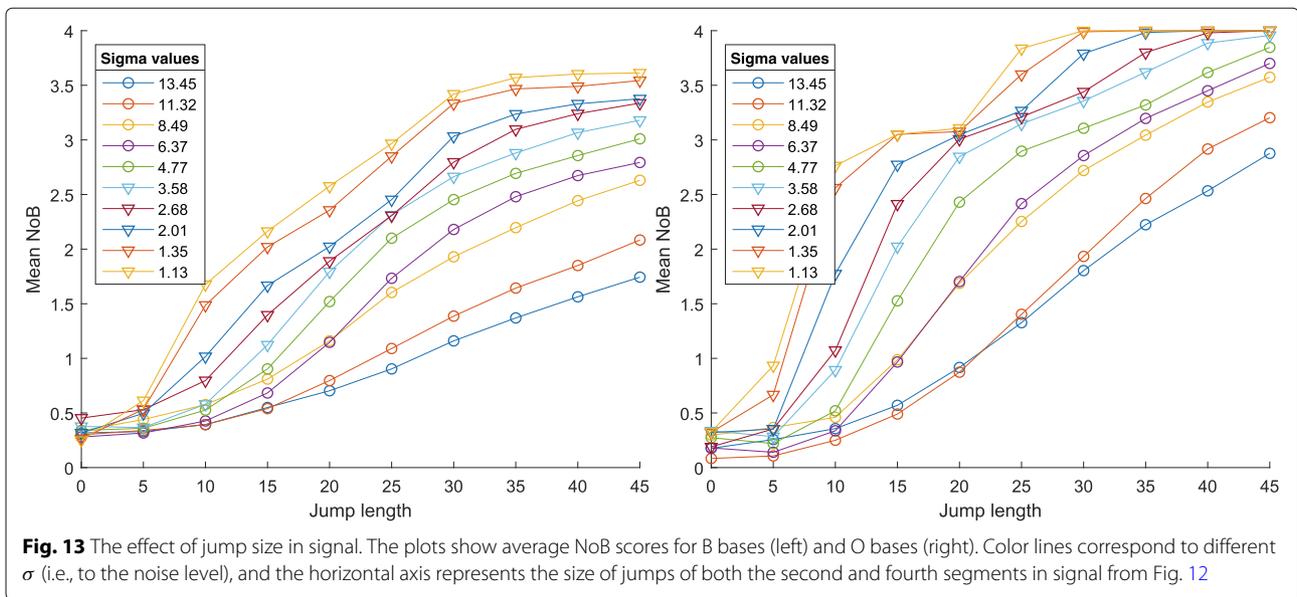
systems perform bad in terms of NoB for such small jump levels).

We comment the results such that although our model includes cases when the signal does not contain jumps, such cases could benefit from extending the model by the additional condition that the segments have to tie up at the breakpoints. For small jumps, our model does not resolve the breakpoints correctly, independent of the choice of the basis.

### 6 Conclusion

The experiment confirmed that using orthonormal bases is highly preferable over the non-orthogonal bases when solving the piecewise-polynomial signal segmentation/denoising problem. It has been shown that the number of correctly detected breakpoints is increased when orthobases are used. Also other performance indicators are improved on average with orthobases, and the plots show that the improvement is the more pronounced





the higher is the noise level. The effect comes almost for free, since it is cheap to generate an orthogonal system, relative to the cost of the numerical algorithm that utilizes the system. In addition, the new approach avoids demanding hands-on setting of “normalization” weights that has been done both by us and by other researchers previously. The user still has to choose  $\delta$ , the quantity which includes the noise level and the model error.

Our experiment revealed that some orthonormal bases are better than others in a particular situation; our results indicate that it could be beneficial to merge detection results of multiple runs with different bases. Such a fusion process could be an interesting direction of future research.

During the revision process of this article, our paper that generalizes the model (8) to two dimensions has been published, see [44]. It shows that it is possible to detect edges in images using this approach; however, it does not aim at comparing different polynomial bases.

## Endnote

<sup>1</sup><http://www.utko.feec.vutbr.cz/~rajmic/sparsegment>

## Abbreviations

AAR: Average to average ratio, Section 4; B bases: Non-orthogonal polynomial bases, Section 4; CA: The Condat Algorithm, Sections 3 and 4; LS: (Ordinary) Least squares, Section 4; MMR: Maximum to minimum ratio, Section 4; MSE: Mean square error, Section 4; N bases: Normalized (nonorthogonal) polynomial bases, Section 4; NoB: Number of correctly detected breakpoints, Section 4; O bases: Orthonormalized N bases, Section 4; PA: Proximal algorithms, Section 3; R bases: Random orthogonal polynomial bases, Section 4; SVD: Singular value decomposition, Section 4

## Acknowledgements

The authors want to thank Vítězslav Veselý, Zdeněk Průša, Michal Fusek, and Nathanaël Perraudin for valuable discussion and comments and to the

reviewers for their careful reading, their comments, and ideas that improved the article. The authors thank the anonymous reviewers for their suggestions that raised the level of both the theoretic and experimental parts.

## Funding

Research described in this paper was financed by the National Sustainability Program under grant LO1401 and by the Czech Science Foundation under grant no. GA16-13830S. For the research, infrastructure of the SIX Center was used.

## Availability of data and materials

The accompanying webpage <http://www.utko.feec.vutbr.cz/~rajmic/sparsegment> contains Matlab code, input data and the full listing of figures. The Matlab code relies on a few routines from the UnlocBox, available at <https://epfl-lts2.github.io/unlocbox-html/>.

## Authors' contributions

MN performed most of the MATLAB coding, experiments, and plotting results. PR wrote most of the article text and both theory and description of the experiments. MS cooperated on the design of experiments and critically revised the manuscript. All authors read and approved the final manuscript.

## Ethics approval and consent to participate

Not applicable.

## Consent for publication

Not applicable.

## Competing interests

The authors declare that they have no competing interests.

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## Author details

<sup>1</sup>Signal Processing Laboratory (SPLab), Brno University of Technology, Technická 12, 616 00 Brno, Czech Republic. <sup>2</sup>The Czech Academy of Sciences, Institute of Information Theory and Automation, Pod Vodárenskou věží 4, 18208 Prague, Czech Republic.

Received: 25 June 2018 Accepted: 13 December 2018

Published online: 25 January 2019

## References

1. P. Prandoni, M. Vetterli, *Signal Processing for Communications, 1st ed. Communication and information sciences.* (CRC Press; EPFL Press, Boca Raton, 2008)
2. M. V. Wickerhauser, *Mathematics for Multimedia.* (Birkhäuser, Basel, Birkhäuser, Boston, 2009)
3. M. Unser, Splines: a perfect fit for signal and image processing. *IEEE Signal Process. Mag.* **16**(6), 22–38 (1999). <https://doi.org/10.1109/79.799930>
4. S. Redif, S. Weiss, J. G. McWhirter, Relevance of polynomial matrix decompositions to broadband blind signal separation. *Signal Process.* **134**(C), 76–86 (2017)
5. J. Foster, J. McWhirter, S. Lambrothar, I. Proudler, M. Davies, J. Chambers, Polynomial matrix qr decomposition for the decoding of frequency selective multiple-input multiple-output communication channels. *IET Signal Process.* **6**(7), 704–712 (2012)
6. G. G. Walter, X. Shen, *Wavelets and Other Orthogonal Systems, Second Edition. Studies Adv. Math.* (Taylor & Francis, CRC Press, Boca Raton, 2000)
7. F. Milletari, N. Navab, S.-A. Ahmadi, in *2016 Fourth International Conference on 3D Vision (3DV)*. V-net: fully convolutional neural networks for volumetric medical image segmentation, (2016), pp. 565–571. <https://doi.org/10.1109/3DV.2016.79>
8. K. Fritscher, P. Raudaschl, P. Zaffino, M. F. Spadea, G. C. Sharp, R. Schubert, ed. by S. Ourselin, L. Joskowicz, M. R. Sabuncu, G. Unal, and W. Wells. *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016* (Springer, Cham, 2016), pp. 158–165
9. R. Giryes, M. Elad, A. M. Bruckstein, Sparsity based methods for overparameterized variational problems. *SIAM J. Imaging Sci.* **8**(3), 2133–2159 (2015)
10. S. Shem-Tov, G. Rosman, G. Adiv, R. Kimmel, A. M. Bruckstein, in *Innovations for Shape Analysis. Mathematics and Visualization*, ed. by M. Breuß, A. Bruckstein, and P. Maragos. *On Globally Optimal Local Modeling: From Moving Least Squares to Over-parametrization* (Springer, Berlin/New York, 2012), pp. 379–405
11. P. Rajmich, M. Novosadová, M. Daňková, Piecewise-polynomial signal segmentation using convex optimization. *Kybernetika.* **53**(6), 1131–1149 (2017). <https://doi.org/10.14736/kyb-2017-6-1131>
12. M. Novosadová, P. Rajmich, in *Proceedings of the 40th International Conference on Telecommunications and Signal Processing (TSP)*. Piecewise-polynomial signal segmentation using reweighted convex optimization (Brno University of Technology, Brno, Barcelona, 2017), pp. 769–774
13. G. Ongie, M. Jacob, Recovery of discontinuous signals using group sparse higher degree total variation. *Signal Process. Lett. IEEE.* **22**(9), 1414–1418 (2015). <https://doi.org/10.1109/LSP.2015.2407321>
14. J. Neubauer, V. Veselý, Change point detection by sparse parameter estimation. *INFORMATICA.* **22**(1), 149–164 (2011)
15. I. W. Selesnick, S. Arnold, V. R. Dantham, Polynomial smoothing of time series with additive step discontinuities. *IEEE Trans. Signal Process.* **60**(12), 6305–6318 (2012). <https://doi.org/10.1109/TSP.2012.2214219>
16. B. Zhang, J. Geng, L. Lai, Multiple change-points estimation in linear regression models via sparse group lasso. *IEEE Trans. Signal Process.* **63**(9), 2209–2224 (2015). <https://doi.org/10.1109/TSP.2015.2411220>
17. K. Bleakley, J.-P. Vert, The group fused Lasso for multiple change-point detection. Technical report (2011). <https://hal.archives-ouvertes.fr/hal-00602121>
18. S.-J. Kim, K. Koh, S. Boyd, D. Gorinevsky,  $\ell_1$  trend filtering. *SIAM Rev.* **51**(2), 339–360 (2009). <https://doi.org/10.1137/070690274>
19. I. W. Selesnick, *Sparsity-Assisted Signal Smoothing.* (R. Balan, M. Begué, J. J. Benedetto, W. Czaja, K. A. Okoudjou, eds.) (Springer, Cham, 2015), pp. 149–176. <https://doi.org/10.1007/978-3-319-20188-76>
20. I. Selesnick, in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference On. Sparsity-assisted signal smoothing (revisited)* (IEEE, 2017), pp. 4546–4550. <https://doi.org/10.1109/ICASSP.2017.7953017>
21. R. J. Tibshirani, Adaptive piecewise polynomial estimation via trend filtering. *Annals Stat.* **42**(1), 285–323 (2014). <https://doi.org/10.1214/13-AOS1189>
22. M. Elad, P. Milanfar, R. Rubinstein, in *Inverse Problems 23 (2007)*. Analysis versus synthesis in signal priors (IOP Publishing Ltd., 2005), pp. 947–968
23. S. Nam, M. Davies, M. Elad, R. Gribonval, The cosparsity analysis model and algorithms. *Appl. Comput. Harmon. Anal.* **34**(1), 30–56 (2013). <https://doi.org/10.1016/j.acha.2012.03.006>
24. M. Unser, J. Fageot, J. P. Ward, Splines are universal solutions of linear inverse problems with generalized TV regularization. *SIAM Rev.* **59**(4), 769–793 (2017)
25. L. Condat, A direct algorithm for 1-D total variation denoising. *Signal Process. Lett. IEEE.* **20**(11), 1054–1057 (2013). <https://doi.org/10.1109/LSP.2013.2278339>
26. I. W. Selesnick, A. Parekh, I. Bayram, Convex 1-D total variation denoising with non-convex regularization. *IEEE Signal Process. Lett.* **22**(2), 141–144 (2015). <https://doi.org/10.1109/LSP.2014.2349356>
27. M. Elad, J. Starck, P. Querre, D. Donoho, Simultaneous cartoon and texture image inpainting using morphological component analysis (mca). *Appl. Comput. Harmon. Anal.* **19**(3), 340–358 (2005)
28. K. Bredies, M. Holler, A TGV-based framework for variational image decomposition, zooming, and reconstruction. part I. *Siam J. Imaging Sci.* **8**(4), 2814–2850 (2015). <https://doi.org/10.1137/15M1023865>
29. M. Holler, K. Kunisch, On infimal convolution of TV-type functionals and applications to video and image reconstruction. *SIAM J. Imaging Sci.* **7**(4), 2258–2300 (2014). <https://doi.org/10.1137/130948793>
30. F. Knoll, K. Bredies, T. Pock, R. Stollberger, Second order total generalized variation (TGV) for MRI. *Magn. Reson. Med.* **65**(2), 480–491 (2011). <https://doi.org/10.1002/mrm.22595>
31. G. Kutyniok, W.-Q. Lim, Compactly supported shearlets are optimally sparse. *J. Approximation Theory.* **163**(11), 1564–1589 (2011). <https://doi.org/10.1016/j.jat.2011.06.005>
32. M. Novosadová, P. Rajmich, in *Proceedings of the 8th International Congress on Ultra Modern Telecommunications and Control Systems. Piecewise-polynomial curve fitting using group sparsity* (IEEE, Lisbon, 2016), pp. 317–322
33. E. J. Candes, M. B. Wakin, S. P. Boyd, Enhancing sparsity by reweighted  $\ell_1$  minimization. *J. Fourier Anal. Appl.* **14**, 877–905 (2008)
34. D. L. Donoho, M. Elad, Optimally sparse representation in general (nonorthogonal) dictionaries via  $\ell_1$  minimization. *Proc. Natl. Acad. Sci.* **100**(5), 2197–2202 (2003)
35. M. Kowalski, B. Torrèsani, in *SPARS'09 – Signal Processing with Adaptive Sparse Structured Representations*, ed. by R. Gribonval. *Structured Sparsity: from Mixed Norms to Structured Shrinkage*, (2009), pp. 1–6. Inria Rennes – Bretagne Atlantique. <http://hal.inria.fr/inria-00369577/en/>. Accessed 2 Jan 2018
36. L. Condat, A generic proximal algorithm for convex optimization—application to total variation minimization. *Signal Process. Lett. IEEE.* **21**(8), 985–989 (2014). <https://doi.org/10.1109/LSP.2014.2322123>
37. L. Condat, A primal-dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms. *J. Optim. Theory Appl.* **158**(2), 460–479 (2013). <https://doi.org/10.1007/s10957-012-0245-9>
38. P. Rajmich, M. Novosadová, in *Proceedings of the 9th International Conference on Telecommunications and Signal Processing. On the limitation of convex optimization for sparse signal segmentation* (Brno University of Technology, Vienna, 2016), pp. 550–554
39. T. Hastie, R. Tibshirani, M. Wainwright, *Statistical Learning with Sparsity.* (CRC Press, Boca Raton, 2015)
40. P. Rajmich, in *Electronics, Circuits and Systems, 2003. ICECS 2003. Proceedings of the 2003 10th IEEE International Conference On, vol. 2. Exact risk analysis of wavelet spectrum thresholding rules*, (2003), pp. 455–458. <https://doi.org/10.1109/ICECS.2003.1301820>
41. R. Tibshirani, Regression shrinkage and selection via the LASSO. *J. R. Stat. Soc. Ser. B Methodol.* **58**(1), 267–288 (1996)
42. M. Daňková, P. Rajmich, in *ESMRMB 2016, 33rd Annual Scientific Meeting, Vienna, AT, September 29–October 1: Abstracts, Friday. Magnetic Resonance Materials in Physics Biology and Medicine. Low-rank model for dynamic MRI: joint solving and debiasing* (Springer, Berlin, 2016), pp. 200–201
43. N. Perraudin, D. I. Shuman, G. Puy, P. Vandergheynst, Unlocbox A Matlab convex optimization toolbox using proximal splitting methods (2014). <https://epfl-lts2.github.io/unlocbox-html/>
44. M. Novosadová, P. Rajmich, in *Proceedings of the 12th International Conference on Signal Processing and Communication Systems (ICSPCS)*. Image edges resolved well when using an overcomplete piecewise-polynomial model, (2018). <https://arxiv.org/abs/1810.06469>