

RESEARCH

Orthogonality is Superiority in Piecewise-polynomial Signal Segmentation and Denoising

Michaela Novosadová¹, Pavel Rajmic^{1*} and Michal Šorel²

*Correspondence: rajmic@vutbr.cz

¹Signal 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

Abstract

Segmentation of signals often relies on the polynomial model which assumes that every segment is a polynomial of 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. In this work we show that using orthogonal polynomials instead of other systems in the model is beneficial when signals corrupted by noise are due to segmentation. The switch to orthogonal bases brings better resolving the breakpoints, it removes the need for including additional parameters and their tuning, and it brings numerical stability. And, it comes for free!

Keywords: signal segmentation; signal smoothing; signal approximation; denoising; piecewise polynomials; orthogonality; sparsity; proximal splitting; convex optimization

1 Introduction

Segmentation of signals is one of the most important applications in digital signal processing, while the most prominent sub-area is the segmentation of images. 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. 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 apriori, while the other class of methods aims at learning discriminative features from the training data [1, 2].

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 [3, 4, 5, 6, 7, 8, 9, 10]. In the “implicit” type of models, the signal is characterized by features that are derived from the signal by using an operator [11, 12, 13, 14, 15]. The described differences are in an analogy to the “synthesis” and “analysis” approaches, respectively, recognized in the sparse signal processing literature [16, 17]. Although the two types of models are different in their

nature, connections can be found; for example, the recent article [18] showing the relationship between splines and generalized total variation regularization.

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 [13, 14, 19, 20].

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 smoothly generalized to a higher dimension. For example, images are commonly modeled as piecewise smooth 2D-functions [21, 22, 23, 24, 25].

In [3, 9, 7], 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. [5, 26], 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 the eventual denoising performance. Worth noting that this improvement comes virtually for free.

The article is structured as follows: Sec. 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 Sec. 3. Finally, Sec. 4 provides the description of experiments and presents 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 are the expansion coefficients in such a basis. If the standard basis is used, i.e.

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

we approach a particular case when 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 $\{p_k\}$ 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:

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

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

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

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

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

Such a description of signal of dimension N is obviously overcomplete — there is $(K + 1)N$ 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.) See Fig. 1 for 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.

Due to the facts described above, the finite difference operator ∇ (properly defined later) applied to vectors \mathbf{x}_k produces sparse vectors. Actually, not only ∇ 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

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

In this optimization program, \mathbf{W} is the diagonal matrix that enables us to adjust the lengths of vectors in \mathbf{P} , operator \mathbf{L} represents the stacked differences such that

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

with $\nabla : \mathbb{R}^N \mapsto \mathbb{R}^{N-1}$, and `reshape()` takes the stacked vector $\mathbf{L}\mathbf{x}$ to the form of a matrix with disjoint columns:

$$\text{reshape}(\mathbf{L}\mathbf{x}) = \left[\nabla\mathbf{x}_0 \mid \cdots \mid \nabla\mathbf{x}_K \right]. \quad (10)$$

The first term of (8) is the penalty. Piecewise constant vectors \mathbf{x}_k suggest that these vectors are sparse under the difference operation ∇ . We use the ℓ_{21} -norm [27] that acts on a matrix \mathbf{Z} 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 ℓ_2 -norm is applied to the particular rows of \mathbf{Z} and the resulting vector is measured by the ℓ_1 -norm. The ℓ_1 -norm serves as a convex substitute of the true sparsity measure [28, 29]. The second term in (8) is the data fidelity term. The Euclidean norm reflects the fact that gaussianity of the noise is assumed and that its level should fall below δ . 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 ℓ_{21} -norm, since no polynomial should be preferred. In this regard, the polynomials should be “normalized” such that \mathbf{W} contains the reciprocal of ℓ_2 -norms of the respective polynomials. It is worth noting that in our former work, in particular in [6], 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 τ_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 [6], and we see that τ_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 [3] 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:

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

This formulation is equivalent to (8) in the sense that to a given δ , there exists λ 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 δ , in opposite to a possible shift in λ 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) [30, 31] 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 $\mathbf{L}_1, \mathbf{L}_2$ are linear operators. In our paper [6] 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 ι_C denotes the indicator function of a convex set C .

Algorithm solving (8) is described in Alg. 1. Therein, two operators are involved: Operator $\text{soft}_{\tau}^{\text{row}}(\mathbf{X})$ takes matrix \mathbf{X} and performs the row-wise group soft thresholding on it. Projector $\text{proj}_{B_2(\mathbf{y}, \delta)}(\mathbf{x})$ finds the closest point to \mathbf{x} in the ball $\{\mathbf{z} : \|\mathbf{y} - \mathbf{z}\|_2 \leq \delta\}$. Both the operations are computationally cheap, see for example [5] for details.

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

Algorithm 1: The Condat Algorithm solving (8)

Input: $\mathbf{PW}: \mathbb{R}^{N(K+1)} \rightarrow \mathbb{R}^N, \mathbf{y}, \delta$
Output: $\hat{\mathbf{x}} = \mathbf{x}^{(i+1)}$
Set parameters $\xi, \sigma > 0$ and $\rho \in (0, 2)$
Set initial primal variable $\mathbf{x}^{(0)} \in \mathbb{R}^{N(K+1)}$ and dual variables $\mathbf{u}_1^{(0)}, \mathbf{u}_2^{(0)} \in \mathbb{R}^{(N-1)(K+1)}$
for $i = 0, 1, \dots$ **do**

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

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

have the upper bound on the operator norms. The upper bound of $\|\mathbf{L}_1\|$ is:

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

and thus $\|\mathbf{L}_1\| \leq 2\sqrt{K+1}$. The operator norm of \mathbf{PW} satisfies $\|\mathbf{PW}\|^2 = \|\mathbf{P}\mathbf{W}\mathbf{W}^\top \mathbf{P}^\top\|$ 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 ℓ_1 norm [32, 33, 34] 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 [32] 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.

The ℓ_{21} -norm cost promotes significant values in vectors $\nabla \hat{\mathbf{x}}_k$ situated at the same positions. We compute a single vector using $\nabla \hat{\mathbf{x}}_k$ s using the weighted ℓ_p -norm according to the formula

$$\mathbf{d} = \sqrt[p]{(\alpha_0 \nabla \hat{\mathbf{x}}_0)^p + \cdots + (\alpha_K \nabla \hat{\mathbf{x}}_K)^p}, \quad (14)$$

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 (14) 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 [5, 6], 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 Experiments

The experiments have 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 randomly generated piecewise quadratic signals ($K = 2$) of length $N = 300$ were generated. All signals consist of six segments of random lengths. There are noticeable jumps in value between neighboring segments. 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 $\boldsymbol{\epsilon} \sim N(0, \sigma^2)$. With these signals, we can determine the signal to noise ratio (SNR), defined as

$$SNR(\mathbf{y}_{\text{noisy}}, \mathbf{y}_{\text{clean}}) = 20 \cdot \log_{10} \frac{\|\mathbf{y}_{\text{clean}}\|_2}{\|\mathbf{y}_{\text{noisy}} - \mathbf{y}_{\text{clean}}\|_2}. \quad (15)$$

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 σ such that

$$\sigma = \frac{\|\mathbf{y}_{\text{clean}}\|_2}{\sqrt{N \cdot 10^{\frac{SNR}{10}}}}. \quad (16)$$

It is clear that the resulting σ 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 2,500 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:

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. In addition to the standard basis, we generated another 49 B-bases using formula $\mathbf{B} = \mathbf{S}\mathbf{W}_1\mathbf{A}\mathbf{W}_2$. Here, the elements of the standard basis—the columns of \mathbf{S} —are first normalized using diagonal \mathbf{W}_1 , then mixed using a random Gaussian matrix \mathbf{A} and finally dilated to different lengths using \mathbf{W}_2 having uniformly random entries at the diagonal. This way we disposed of 50 B-bases in total, 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, $\mathbf{N} = \mathbf{B}\mathbf{W}_3$. 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.

$$\mathbf{N} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top. \quad (17)$$

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 matrices \mathbf{N} and \mathbf{U} , respectively, 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 (17), now symbolized using the subscripts, $\mathbf{N} = \mathbf{U}_\mathbf{N}\mathbf{\Sigma}_\mathbf{N}\mathbf{V}_\mathbf{N}^\top$. Next, a random matrix \mathbf{A} of size 3×3 was generated, each element of whose independently follows the Gauss distribution. This matrix is then decomposed to $\mathbf{A} = \mathbf{U}_\mathbf{A}\mathbf{\Sigma}_\mathbf{A}\mathbf{V}_\mathbf{A}^\top$. The new basis \mathbf{R} is obtained as $\mathbf{R} = \mathbf{U}_\mathbf{N}\mathbf{U}_\mathbf{A}$. 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 (B, N, O, R) we disposed of 200 bases.

4.3 Experiment

The algorithm of breakpoint detection that we utilized in the experiments has been described in Sec. 3.2. We used $p = 2$ in formula (14) for computing the input vector. The Condat algorithm run for 2,000 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 δ 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 $2,500 \times 200$ experiments in total in signal breakpoints detection.

Setting parameter δ

For each of the 2,500 noisy signals, the parameter δ was calculated. Since both the noisy and clean signals are known in our simulation, δ could be determined according to

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

meaning that we allowed the model error to deviate from the ground truth by 5% at maximum. Fig. 3 shows the distribution of values of δ . For different signals, δ is concentrated around a different quantity. This effect is due to the noise generation, wherein the resulting SNR (15) was set and fixed at first, while δ is linked to the noise deviation σ that depends on the signal, cf. (16).

4.4 Evaluation

Based on vector \mathbf{d} , several values that indicate the quality of breakpoint detection process were computed. But first, for each single signal in test, two disjoint sets of indexes are 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 (14) 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

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

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 apriori knowledge that the regularizer biases the estimate. As an example, debiasing is commonly done in LASSO estimation [35, 33], 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 [28, 6, 36].

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

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 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^[1], 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 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, it means that (especially) when SNR is high, it makes sense 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 Sec. 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 experiments have been done in Matlab (2017a), and for some proximal algorithm components we benefited from using the flexible UnLocBox toolbox [37]. The code related to the experiments is available via the mentioned webpage.

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

5 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 computationally cheap to generate an orthogonal polynomial system, compared to the actual cost of iterations in the numerical algorithm. 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 estimate δ , the parameter which estimates the noise level.

Our experiment revealed that some orthonormal bases are better than others in a particular situation; we thus recommend to merge the detection results for multiple bases. Such a fusion process could be an interesting direction of future research.

We would like to extend the approach to segmentation of images as the next step.

Competing interests

The authors declare that they have no competing interests.

Author's contributions

MN performed most of the Matlab coding, experiments and plotting results. PR wrote most of the article text, both theory and description of the experiments. MŠ cooperated on the design of experiments, and critically revised the manuscript.

Acknowledgements

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

Author details

¹Signal Processing Laboratory (SPLab), Brno University of Technology, Technická 12, 616 00 Brno, Czech Republic.

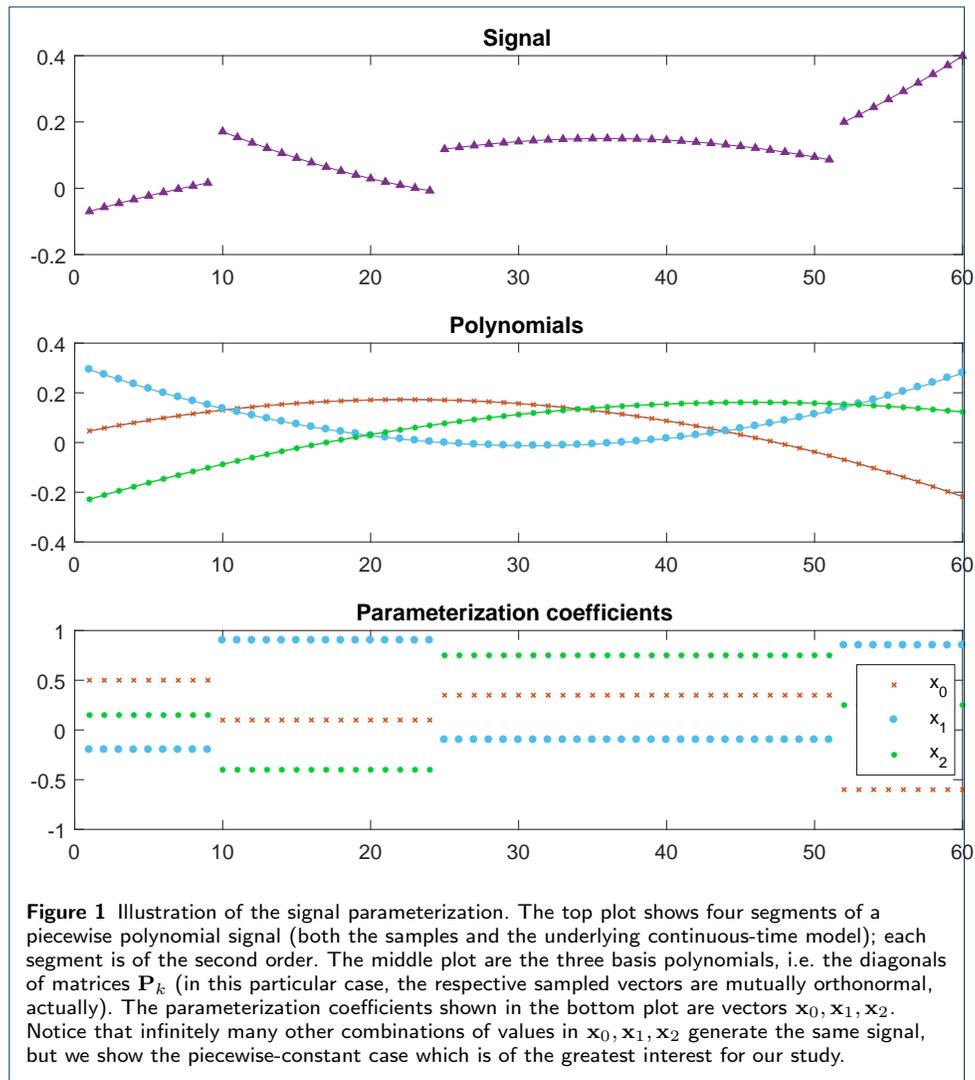
²ÚTIA AV ČR, Pod Vodárenskou věží 4, 182 08 Praha, Czech Republic.

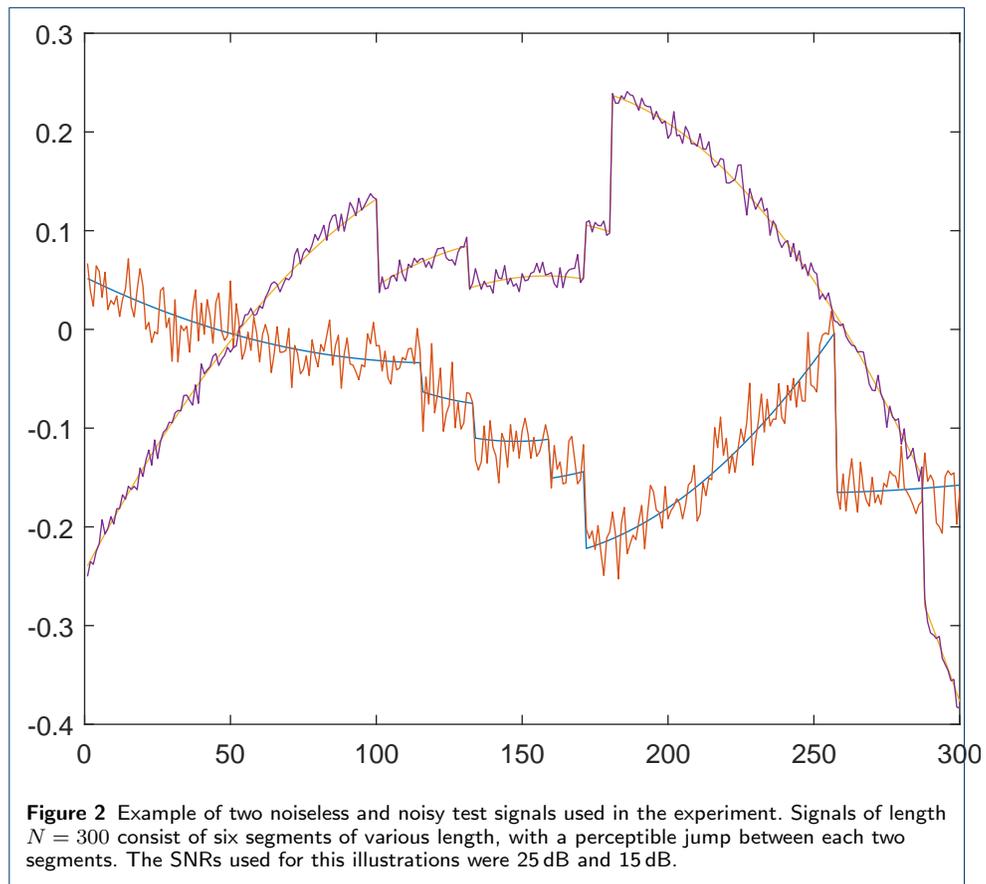
References

- Milletari, F., Navab, N., Ahmadi, S.-A.: V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 2016 Fourth International Conference on 3D Vision (3DV), pp. 565–571 (2016). doi:10.1109/3DV.2016.79
- Fritscher, K., Raudaschl, P., Zaffino, P., Spadea, M.F., Sharp, G.C., Schubert, R.: Deep neural networks for fast segmentation of 3D medical images. In: Ourselin, S., Joskowicz, L., Sabuncu, M.R., Unal, G., Wells, W. (eds.) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016, pp. 158–165. Springer, Cham (2016)
- Giryès, R., Elad, M., Bruckstein, A.M.: Sparsity based methods for overparameterized variational problems. *SIAM journal on imaging sciences* **8**(3), 2133–2159 (2015)
- Shem-Tov, S., Rosman, G., Adiv, G., Kimmel, R., Bruckstein, A.M.: On Globally Optimal Local Modeling: From Moving Least Squares to Over-parametrization. In: Breuß, M., Bruckstein, A., Maragos, P. (eds.) Innovations for Shape Analysis. Mathematics and Visualization, pp. 379–405. Springer, ??? (2012)
- Rajmic, P., Novosadová, M., Daňková, M.: Piecewise-polynomial signal segmentation using convex optimization. *Kybernetika* **53**(6), 1131–1149 (2017). doi:10.14736/kyb-2017-6-1131
- Novosadová, M., Rajmic, P.: Piecewise-polynomial signal segmentation using reweighted convex optimization. In: Proceedings of the 40th International Conference on Telecommunications and Signal Processing (TSP), Barcelona, pp. 769–774 (2017)
- Ongie, G., Jacob, M.: Recovery of discontinuous signals using group sparse higher degree total variation. *Signal Processing Letters, IEEE* **22**(9), 1414–1418 (2015). doi:10.1109/LSP.2015.2407321
- Neubauer, J., Veselý, V.: Change point detection by sparse parameter estimation. *INFORMATICA* **22**(1), 149–164 (2011)
- Selesnick, I.W., Arnold, S., Dantham, V.R.: Polynomial smoothing of time series with additive step discontinuities. *IEEE Transactions on Signal Processing* **60**(12), 6305–6318 (2012). doi:10.1109/TSP.2012.2214219
- Zhang, B., Geng, J., Lai, L.: Multiple change-points estimation in linear regression models via sparse group lasso. *IEEE Transactions on Signal Processing* **63**(9), 2209–2224 (2015). doi:10.1109/TSP.2015.2411220

11. Bleakley, K., Vert, J.-P.: The group fused Lasso for multiple change-point detection. Technical report (2011). <https://hal.archives-ouvertes.fr/hal-00602121>
12. Kim, S.-J., Koh, K., Boyd, S., Gorinevsky, D.: ℓ_1 trend filtering. *SIAM Review* **51**(2), 339–360 (2009). doi:10.1137/070690274
13. Selesnick, I.W.: In: Balan, R., Begué, M., Benedetto, J.J., Czaja, W., Okoudjou, K.A. (eds.) *Sparsity-Assisted Signal Smoothing*, pp. 149–176. Springer, Cham (2015). doi:10.1007/978-3-319-20188-7
14. Selesnick, I.: Sparsity-assisted signal smoothing (revisited). In: *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference On*, pp. 4546–4550. IEEE, ??? (2017). doi:10.1109/ICASSP.2017.7953017
15. Tibshirani, R.J.: Adaptive piecewise polynomial estimation via trend filtering. *Annals of Statistics* **42**(1), 285–323 (2014). doi:10.1214/13-AOS1189
16. Elad, M., Milanfar, P., Rubinstein, R.: Analysis versus synthesis in signal priors. In: *Inverse Problems* **23** (200), pp. 947–968 (2005)
17. Nam, S., Davies, M., Elad, M., Gribonval, R.: The cosparsity analysis model and algorithms. *Applied and Computational Harmonic Analysis* **34**(1), 30–56 (2013). doi:10.1016/j.acha.2012.03.006
18. Unser, M., Fageot, J., Ward, J.P.: Splines are universal solutions of linear inverse problems with generalized TV regularization. *SIAM Review* **59**(4), 769–793 (2017)
19. Condat, L.: A direct algorithm for 1-D total variation denoising. *Signal Processing Letters, IEEE* **20**(11), 1054–1057 (2013). doi:10.1109/LSP.2013.2278339
20. Selesnick, I.W., Parekh, A., Bayram, I.: Convex 1-D total variation denoising with non-convex regularization. *IEEE Signal Processing Letters* **22**(2), 141–144 (2015). doi:10.1109/LSP.2014.2349356
21. Elad, M., Starck, J., Querre, P., Donoho, D.: Simultaneous cartoon and texture image inpainting using morphological component analysis (mca). *Applied and Computational Harmonic Analysis* **19**(3), 340–358 (2005)
22. Bredies, K., Holler, M.: A TGV-based framework for variational image decompression, zooming, and reconstruction. part I. *Siam Journal On Imaging Sciences* **8**(4), 2814–2850 (2015). doi:10.1137/15M1023865
23. Holler, M., Kunisch, K.: On infimal convolution of tv-type functionals and applications to video and image reconstruction. *SIAM Journal on Imaging Sciences* **7**(4), 2258–2300 (2014). doi:10.1137/130948793
24. Knoll, F., Bredies, K., Pock, T., Stollberger, R.: Second order total generalized variation (tgv) for mri. *Magnetic Resonance in Medicine* **65**(2), 480–491 (2011). doi:10.1002/mrm.22595
25. Kutyniok, G., Lim, W.-Q.: Compactly supported shearlets are optimally sparse. *Journal of Approximation Theory* **163**(11), 1564–1589 (2011). doi:10.1016/j.jat.2011.06.005
26. Novosadová, M., Rajmic, P.: Piecewise-polynomial curve fitting using group sparsity. In: *Proceedings of the 8th International Congress on Ultra Modern Telecommunications and Control Systems, Lisbon*, pp. 317–322 (2016)
27. Kowalski, M., Torrèsani, B.: Structured Sparsity: from Mixed Norms to Structured Shrinkage. In: Gribonval, R. (ed.) *SPARS'09 – Signal Processing with Adaptive Sparse Structured Representations*, pp. 1–6 (2009). Inria Rennes – Bretagne Atlantique. <http://hal.inria.fr/inria-00369577/en/>
28. Candes, E.J., Wakin, M.B., Boyd, S.P.: Enhancing sparsity by reweighted ℓ_1 minimization. *Journal of Fourier Analysis and Applications* **14**, 877–905 (2008)
29. Donoho, D.L., Elad, M.: Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization. *Proceedings of The National Academy of Sciences* **100**(5), 2197–2202 (2003)
30. Condat, L.: A generic proximal algorithm for convex optimization—application to total variation minimization. *Signal Processing Letters, IEEE* **21**(8), 985–989 (2014). doi:10.1109/LSP.2014.2322123
31. Condat, L.: A primal-dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms. *Journal of Optimization Theory and Applications* **158**(2), 460–479 (2013). doi:10.1007/s10957-012-0245-9
32. Rajmic, P., Novosadová, M.: On the limitation of convex optimization for sparse signal segmentation. In: *Proceedings of the 9th International Conference on Telecommunications and Signal Processing*, pp. 550–554. Brno University of Technology, ??? (2016)
33. Hastie, T., Tibshirani, R., Wainwright, M.: *Statistical Learning with Sparsity*. CRC Press, Boca Raton (2015)
34. Rajmic, P.: Exact risk analysis of wavelet spectrum thresholding rules. In: *Electronics, Circuits and Systems, 2003. ICECS 2003. Proceedings of the 2003 10th IEEE International Conference On*, vol. 2, pp. 455–4582 (2003). doi:10.1109/ICECS.2003.1301820
35. Tibshirani, R.: Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society. Series B (Methodological)* **58**(1), 267–288 (1996)
36. Daňková, M., Rajmic, P.: Low-rank model for dynamic MRI: joint solving and debiasing. In: *ESMRMB 2016, 33rd Annual Scientific Meeting, Vienna, AT, September 29–October 1: Abstracts, Friday. Magnetic Resonance Materials in Physics Biology and Medicine*, pp. 200–201. Berlin: Springer, ??? (2016)
37. Perraudin, N., Shuman, D.I., Puy, G., Vandergheynst, P.: Unlocbox A Matlab convex optimization toolbox using proximal splitting methods (2014)

Figures





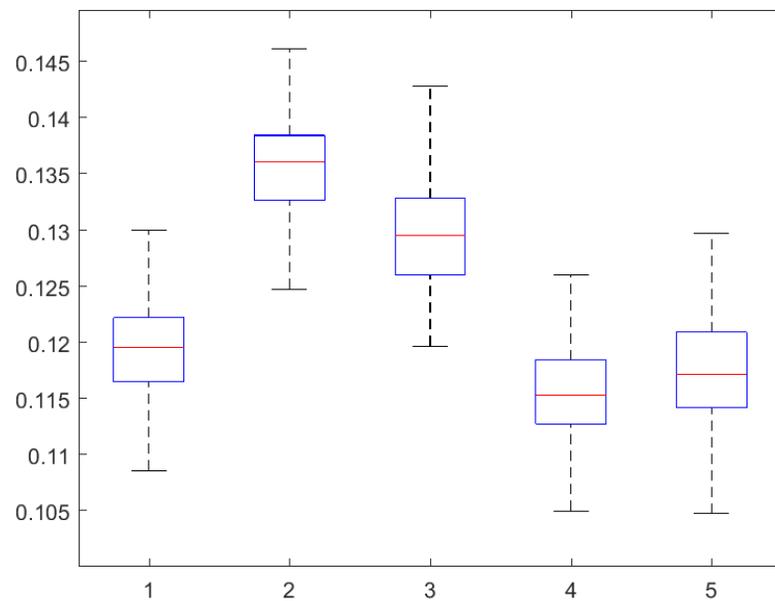


Figure 3 Distribution of δ parameter across the five groups of test signals. The SNR is 25 dB in this example. The box plots show the maximum and minimum, first quartile and the third quartile forming the edges of the rectangle, and the median value within the box. Values of δ vary within the signal (which is given by particular realizations of the noise) and between the signals (which is due to fixing the SNR rather than the noise power).

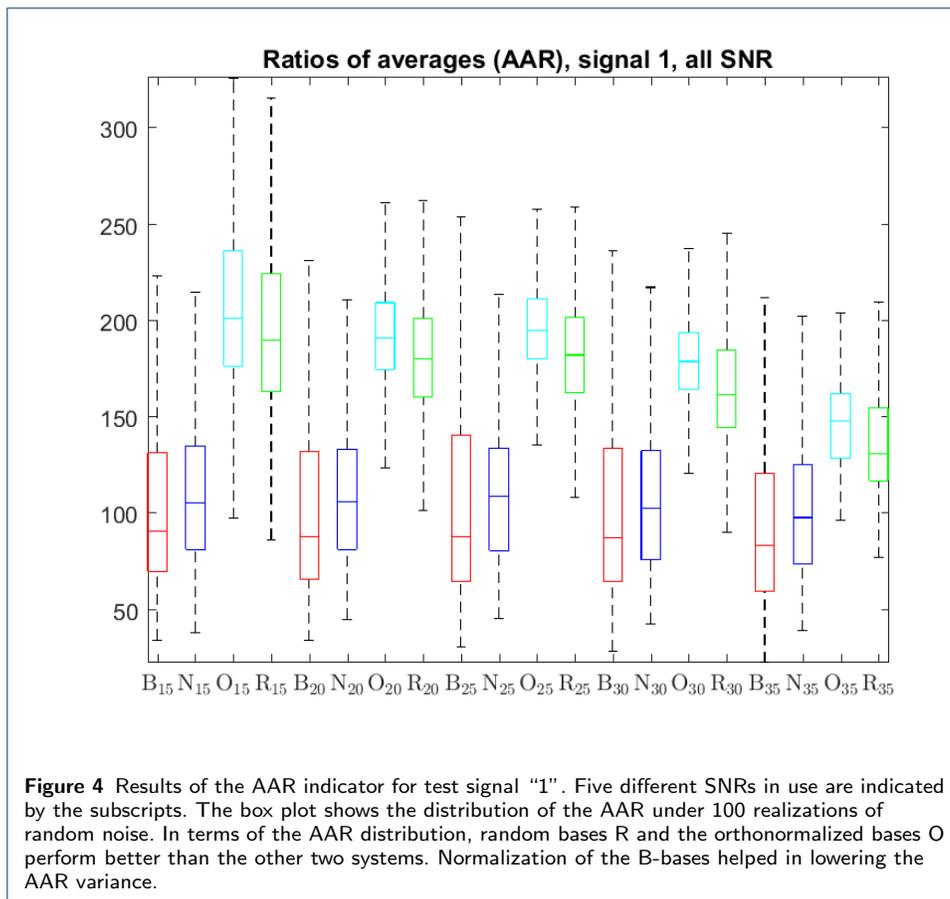
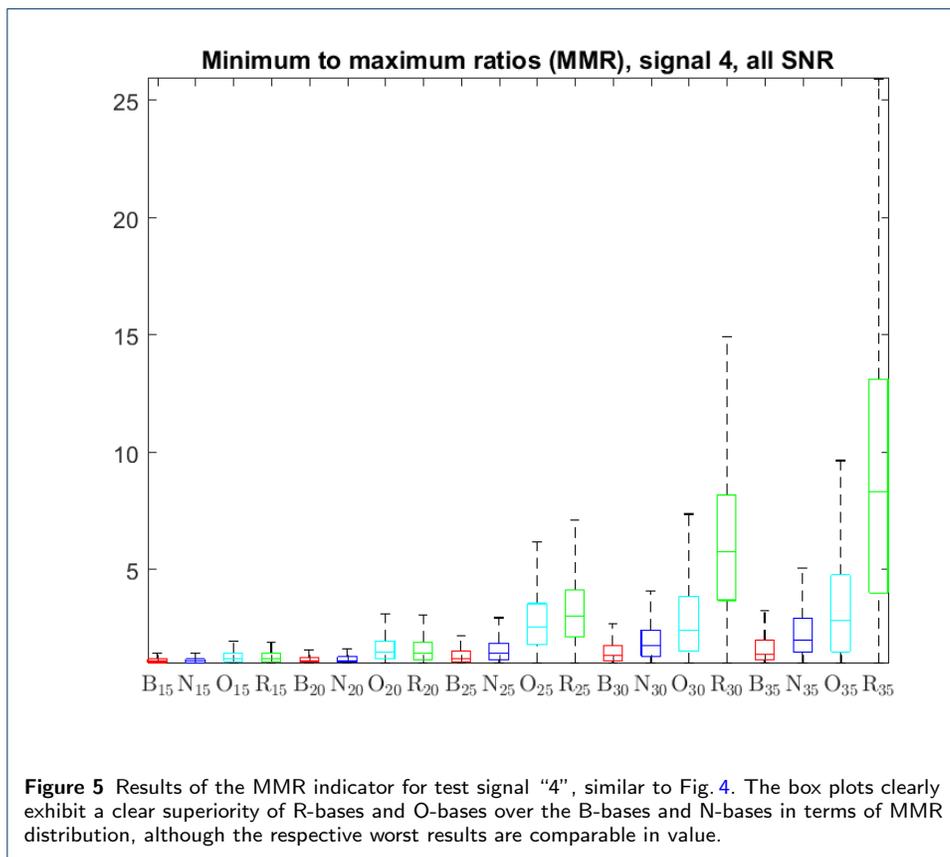
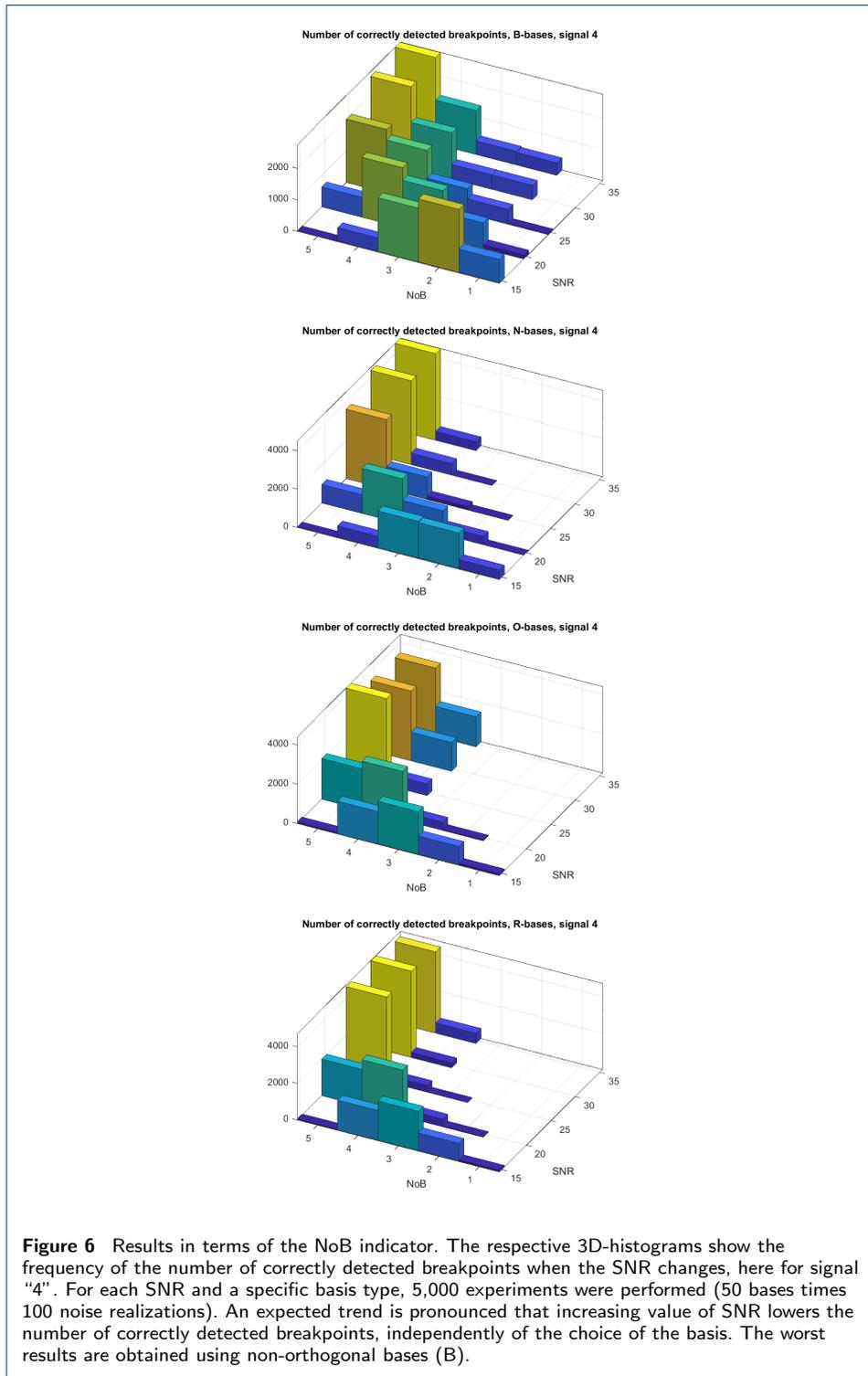
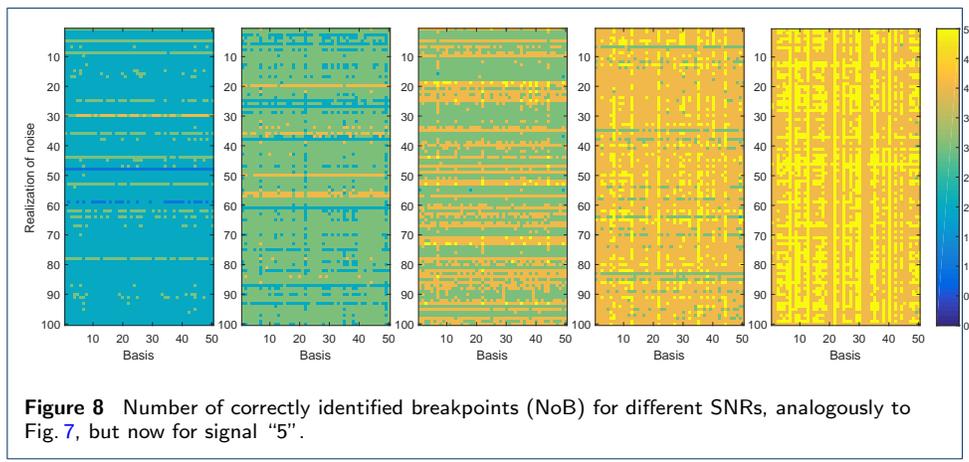
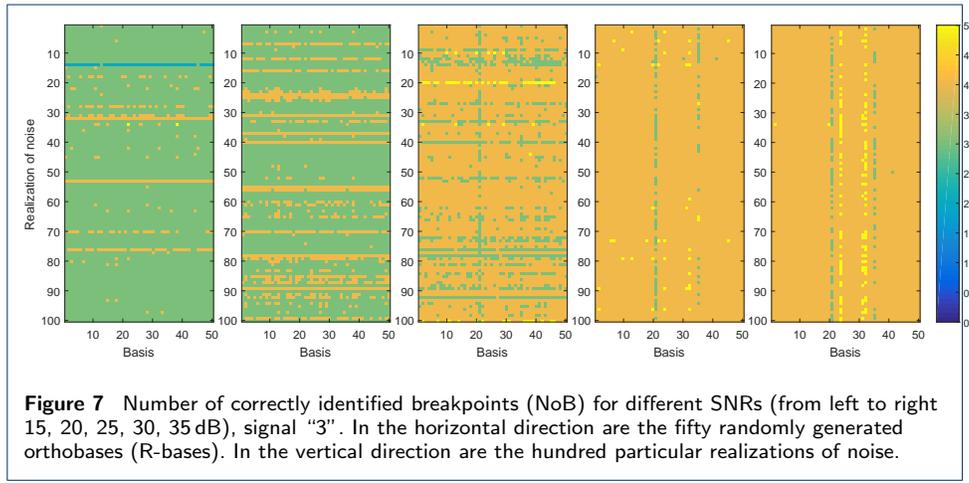


Figure 4 Results of the AAR indicator for test signal "1". Five different SNRs in use are indicated by the subscripts. The box plot shows the distribution of the AAR under 100 realizations of random noise. In terms of the AAR distribution, random bases R and the orthonormalized bases O perform better than the other two systems. Normalization of the B-bases helped in lowering the AAR variance.







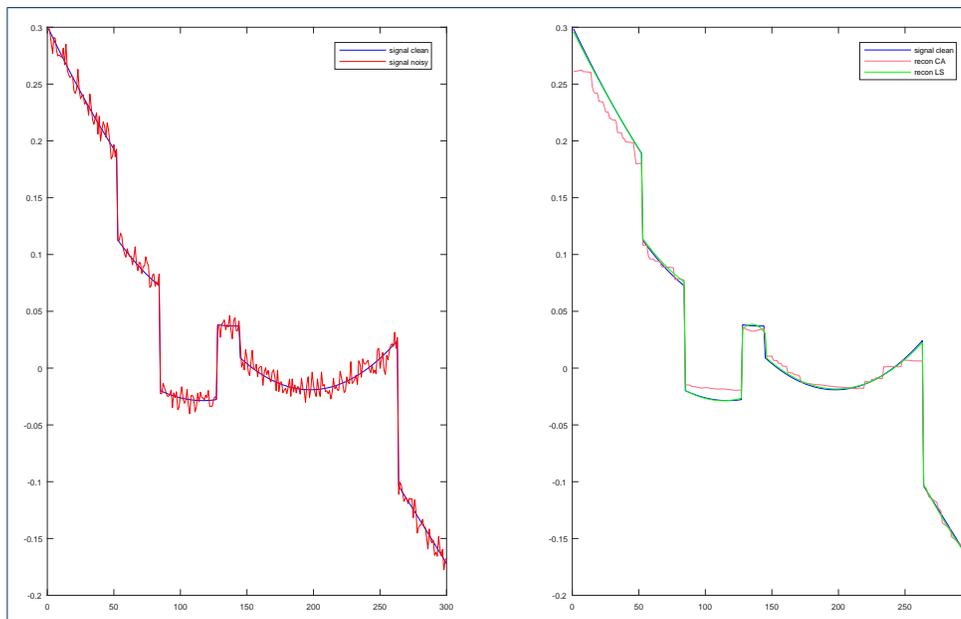


Figure 9 An example of time-domain reconstruction, test signal “1”. Left side shows the noiseless and noisy signals, the plot on the right hand presents the direct signal estimate $\hat{y} = PW\hat{x}$ (CA), and the respective least squares refit (LS), on top of the noiseless signal. Clearly, LS radically improves the adherence to the data (and thus improves the MSE). The bias of the CA is explained in Sec. 3.2.

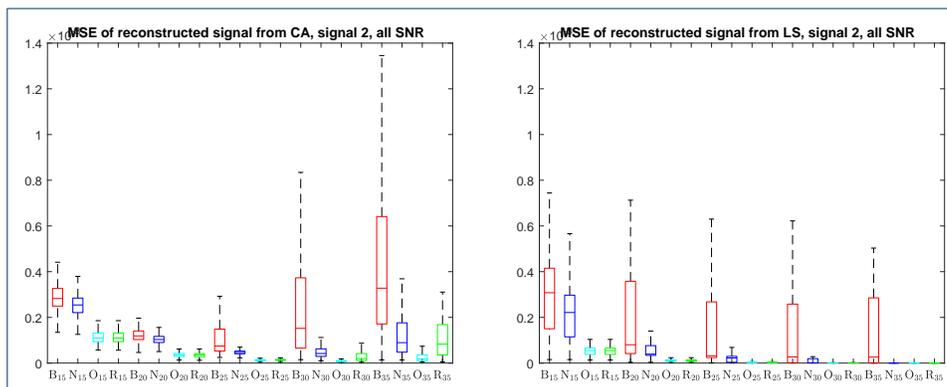


Figure 10 Results in terms of MSE for test signal “2”. Left plot shows the case of direct signal estimates (CA), right plot shows the MSE for the least squares (LS). The plots have same scale. While simple normalization (N-bases) helps reducing the MSE, orthobases clearly bring an extra improvement.

