

# Robust smoothing of one-dimensional data with missing and/or outlier values

Nasser Mourad<sup>1,2</sup>

<sup>1</sup>Department of Electrical Engineering, Aswan Faculty of Engineering, Aswan University, Egypt

<sup>2</sup>Department of Electrical Engineering, College of Engineering, Shaqra University, Riyadh, Saudi Arabia

## Correspondence

Nasser Mourad, Department of Electrical Engineering, Aswan Faculty of Engineering, Aswan University, Egypt.

Email: [mouradna@aswu.edu.eg](mailto:mouradna@aswu.edu.eg)

## Abstract

Penalized least squares (PLS) is a popular data smoothing technique. However, existing PLS smoothing algorithms behave as low-pass filters (LPF), and, hence, they may introduce distortions to bandpass signals. These algorithms also have difficulties associated with the selection of the smoothing parameter and the order of the difference matrix they utilize. A new PLS smoothing algorithm is developed to overcome these limitations. In the proposed algorithm, two difference matrices and two regularisation parameters are utilized. As a result, the proposed algorithm can act as either an LPF or a bandpass filter. The particular shape of the filter is determined by the values of the regularisation parameters and the orders of the two difference matrices. The spectral characteristics of the smooth signal are utilized to derive closed-form expressions for the optimum values of these four parameters. The derived algorithm is further developed to handle the case of having missing and/or outlier samples in the measured data. The proposed algorithms are completely data driven, and they do not require any special optimisation tools. In addition, the proposed algorithms are also shown to produce significantly improved results compared to some existing smoothing techniques.

## 1 | INTRODUCTION

Signal denoising/smoothing is one of the most common problems in signal processing. Measured signals (data) are usually contaminated by different kinds of noise and/or interfering signals, for example, instrumentation noise, 50/60 Hz power line interference and baseline drift. In addition, measured signals are also susceptible to missing and/or outlier samples, for example, as a result of instrument malfunction. The case of denoising smooth signals contaminated by additive white Gaussian noise (AWGN) is considered, where a smooth signal is defined as a signal that has continuous derivatives up to some order (typically  $\geq 2$ ) over its entire domain [1]. This problem is known in the study as the signal smoothing, to distinguish it from the signal denoising problem. See [2] for the distinction between signal denoising and signal smoothing. The possibility of having missing and/or outlier samples is also considered.

Many procedures have been developed in the literature to perform signal smoothing, each of which has its own advantages and disadvantages. The simplest procedure is applying a

moving average filter to the data. However, in order to achieve a high degree of noise reduction, a large filter length  $N$  has to be utilized, which in turn could distort useful high-frequency components in the desired smooth signal [3]. Another procedure, which is a generalisation to the moving average procedure, is based on fitting each  $N$  samples of the noisy signal by a  $d$ th order polynomial, where  $N$  is odd and  $\geq d + 1$ . Using least squares fitting to calculate the coefficients of the fitting polynomial, this procedure is reduced to discrete convolution with a fixed impulse response. This procedure is known as the Savitzky–Golay (SG) filter [4]. See [3–5] and the references therein for more details about the SG filtering procedure. The SG filter has the advantage that it does not introduce delays to the smoothed signal, and it can handle short intervals of missing samples [6]. However, the SG filter may not be able to handle large intervals of missing data, and special care has to be applied to the computations at the boundary of the analyzed data [6]. In addition, the two parameters  $d$  and  $N$  control the shape of the frequency response of the SG filter [3], and hence they have to be carefully selected in order to achieve a desired performance.

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2021 The Authors. *IET Signal Processing* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology.

Another smoothing approach that has wide range of applications is called the singular spectrum analysis (SSA) [7–9]. The SSA is nonparametric, and it makes no prior assumptions about the properties (e.g. linearity, stationarity or Gaussianity) of the analysed data. Therefore, SSA can be utilized with arbitrary statistical processes [8,9]. The SSA technique is based on converting a measured time series into a matrix called the trajectory matrix, which in turn is decomposed using singular value decomposition. The separated singular vectors are then classified into signal and noise subspaces, and the smoothed signal is constructed using the signal subspace. See [7–9] for more details about the SSA technique. Two difficulties are associated with the SSA approach: choosing the appropriate dimension of the trajectory matrix and identifying the signal subspace. Some approaches have been developed to overcome the second difficulty [8–10], while the first difficulty remains problem dependent. Some guidelines about selecting the size of the trajectory matrix were given in [8].

The empirical mode decomposition (EMD) technique [11] adaptively decomposes any time series into a set of components called intrinsic mode functions (IMFs). Noise-assisted versions called ensemble EMD (EEMD) [12], complementary EEMD (CEEMD) [13] and the CEEMD with adaptive noise (CEEMDAN) [14] have been developed to further improve the decomposition capabilities of the EMD approach. The EMD-based decomposition techniques have been utilized in wide range of applications, especially in the area of signal denoising/smoothing [11–17]. In the EMD-based decomposition techniques, the IMFs are extracted in the descending order of their frequency contents. Therefore, the main idea of signal smoothing using EMD is to exclude the lower order IMFs, which most probably belong to the contaminating noise. From the filtering point of view, it was reported in [18] that EMD acts essentially as a dyadic filter bank. Therefore, excluding the lower-order IMFs is equivalent to lowpass filtering, in which the bandwidth (BW) of the filter is dynamically and automatically adjusted based on the input signal. To ensure that no signal components have erroneously eliminated with the discarded IMFs (as a result of the mode mixing phenomenon associated with the EMD), one of the noise-assisted EEMD approaches has to be utilized. However, the computation cost of EEMD-based approaches is much higher than that of the basic EMD. In addition, residual noise could be introduced in the IMFs as a side effect of the utilized noise-assisted technique [16].

Another popular smoothing approach is the penalized least squares (PLS) regression, which is often credited to a publication by Whitaker [19]. In this approach, a smooth signal is obtained by minimising a cost function that balances two objectives: the data fidelity to the measured signal and the roughness of the smoothed signal. The balance between these two objectives is controlled by a parameter  $\mu$  called the smoothing parameter, which has to be carefully selected. Based on the utilized measure of roughness, two different classes of algorithms can be distinguished. The first class is known as the smoothing spline [20,21], and in this class the roughness is measured using the square integral of the  $q$ th derivative of the

smoothed signal. Usually, the value  $q = 2$  is utilized, which results in the well-known smoothing approach called cubic spline. In the second class of algorithms, the roughness of the smoothed signal is measured using the  $q$ th order difference matrix [1,6,19,20–25]. In the case of having equally spaced samples with repeated border elements, a smooth signal in this case can be efficiently calculated using the discrete cosine transform (DCT) [1,25–27]. The optimum value of the smoothing parameter  $\mu$  used in both classes of algorithms is usually calculated using the generalized cross validation (GCV) method [28].

As will be demonstrated in Section 2.1, the existing PLS smoothing algorithms act solely as low-pass filters (LPF) of the DCT coefficients of the measured noisy data. In addition, both the BW and the transition band  $\Delta f$  (from the pass-band to the stop-band) of the LPF are controlled by the smoothing parameter  $\mu$  and the order  $q$  of the difference matrix used with these algorithms. Therefore, the following three limitations associated with the existing PLS smoothing algorithms can be recognized. First, these algorithms usually introduce distortions to bandpass signals, that is, signals their DCT coefficients are confined in the frequency band  $f_L \leq f \leq f_H$ , where  $f_L > 0$  is the lower pass-band edge frequency. In addition, the higher the value of  $f_L$ , the larger the distortion introduced to the bandpass smoothed signal. The second limitation is associated with the selection of the order  $q$  of the difference matrix. The value of  $q$  is usually selected independently from the spectral contents of the smooth signal to be restored. This may result in inappropriate values of the BW and  $\Delta f$  of the LPF, and hence a distortion could be introduced to the smoothed signal. The third limitation is associated with the selection of the value of the smoothing parameter  $\mu$ . As stated before, this is usually done using the GCV method. However, computing  $\mu$  using the GCV increases the computation cost of the smoothing algorithm. In addition, the GCV could fail to find the optimum value of  $\mu$  in cases where the GCV objective has shallow minimum [22], or when an inappropriate value of  $q$  was selected.

To overcome the above mentioned limitations, a new PLS smoothing algorithm is proposed. A new objective function is utilized in the proposed algorithm. The proposed objective function depends on four different parameters: two regularisation parameters and the orders of two difference matrices. By proper selection of the four parameters, the proposed algorithm can act as either an LPF, a high-pass filter (HPF) or a bandpass filter (BPF) to the DCT coefficients of the measured data. For automatic selection of the appropriate filter frequency response, closed-form expressions of the optimum values of the four parameters are derived based on the spectral characteristics of the smooth signal to be restored. Therefore, the proposed algorithm always attempts to minimize the distortion introduced to the smoothed signal by encouraging its band to be confined to the band of the embedded smooth signal. The proposed algorithm is referred to as the band-confined smoothing algorithm (BCSA).

The proposed BCSA algorithm is further developed to handle the case of having missing and/or outlier samples, and

the resulting algorithm is referred to as robust BCSA (RBCSA). In applying the RBCSA algorithm, the indices of the missing samples are assumed to be known a priori, while an adaptive procedure is followed to estimate the indices of the outlier samples. Therefore, both BCSA and RBCSA are completely data driven, fast and do not require any special optimisation tools to accomplish their goals.

The remaining part of this article is organized as follows. The proposed algorithms are derived in Section 2. Section 3 presents some experiments and discussions to assess the performance of the proposed algorithms. Finally, concluding remarks are given in Section 4. In addition, the following notations are used throughout the full text. A bold upper case symbol, for example,  $\mathbf{A}$  refers to a matrix; similarly  $\boldsymbol{\alpha}$  refers to a column vector; the  $k$ th element of  $\boldsymbol{\alpha}$  is referred to using either  $\alpha_k$  or  $\alpha[k]$ ;  $\boldsymbol{\alpha}[t_1:t_2]$  is the set of elements in  $\boldsymbol{\alpha}$  from index  $t_1$  to  $t_2$ ;  $\text{diag}(\boldsymbol{\alpha})$  is a diagonal matrix its main diagonal elements are the entries of  $\boldsymbol{\alpha}$ ; a discrete-time signal  $z(t)$ ,  $t = 1, \dots, T$ , is represented (using vector notations) by the vector  $\mathbf{z} \in \mathbb{R}^T$ , where  $T$  is the number of time points; and  $\|\mathbf{z}\|_p = \left(\sum_{t=1}^T |z_t|^p\right)^{\frac{1}{p}}$  is the  $\ell_p$ -norm of  $\mathbf{z}$ .

## 2 | METHODS

### 2.1 | Problem Formulation and Related Work

A noisy signal  $y(t)$  is modelled using the following equation:

$$y(t) = x(t) + v(t), \quad t = 1, \dots, T \quad (1)$$

where  $x(t)$  is a smooth signal,  $v(t)$  is a zero mean AWGN and  $T$  is the number of time points (samples). It is assumed that the samples of the noisy signal  $y(t)$  are equally spaced. This holds for most applications, because many instruments sample signals at a regular grid along time. The more general case of having arbitrarily spaced observations [22] is not considered, because this case does not result in a simple form as the one derived in this section.

An estimate of the smooth signal  $x(t)$  was obtained in [1,6,19,22–24,26,27] by solving the following penalized least squares problem:

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\text{argmin}} \frac{1}{2} \|\mathbf{y} - \mathbf{x}\|_2^2 + \frac{1}{2} \mu_L \|\mathbf{D}_q \mathbf{x}\|_2^2, \quad (2)$$

where  $\mathbf{D}_q$  is the  $q$ th order difference matrix, and  $\mu_L$  is the smoothing parameter, which controls the trade-off between the fidelity to the measured data (measured by  $\|\mathbf{y} - \mathbf{x}\|_2^2$ ) and the smoothness of the estimated signal (measured by  $\|\mathbf{D}_q \mathbf{x}\|_2^2$ ). The structure of  $\mathbf{D}_q$  for some values of  $q$  is presented shortly.

Equation (2) is a convex optimisation problem, and its minimisation results in the following linear system that allows the determination of the smoothed signal  $\hat{\mathbf{x}}$

$$\mathbf{A}_q \hat{\mathbf{x}} = \mathbf{y}, \quad (3)$$

where  $\mathbf{A}_q := (\mathbf{I} + \mu_L \mathbf{D}_q^T \mathbf{D}_q)$ ,  $\hat{\mathbf{x}}$  is an estimate of the smooth signal  $\mathbf{x}$ , and  $\mathbf{I}$  is the  $T \times T$  identity matrix. The solution vector  $\hat{\mathbf{x}}$  can be calculated directly using  $\hat{\mathbf{x}} = \mathbf{A}_q^{-1} \mathbf{y}$  as long as the data size  $T$  is small, for example, less than 1000 samples. Beyond that size, the time to calculate the inverse of  $\mathbf{A}_q$  becomes relatively large [6]. Utilising the sparsity and the symmetry of  $\mathbf{A}_q$ , more computationally efficient approaches have been proposed in [6,23]. However, these two approaches lose efficiency when an estimation of the smoothing parameter  $\mu_L$  is required [1].

Another approach to solve Equation (3) for the case  $q = 1$  was suggested in [27]. It is known that the first-order difference matrix  $\mathbf{D}_1$  is a Toeplitz banded matrix, and its first row equals  $[-1 \ 1 \ 0 \ \dots \ 0]$ . Therefore, the matrix  $\mathbf{D} := \mathbf{D}_1^T \mathbf{D}_1$  has the following structure:

$$\mathbf{D} = \begin{bmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 1 \end{bmatrix} \quad (4)$$

The author in [27] utilized the result previously reported in [29] that the matrix  $\mathbf{D}$  is diagonalized by the DCT. Therefore, the eigendecomposition of  $\mathbf{D}$  is given by:

$$\mathbf{D} = \mathbf{U} \boldsymbol{\Lambda} \mathbf{U}^T, \quad (5)$$

where  $\mathbf{U}$  is a unitary matrix ( $\mathbf{U} \mathbf{U}^T = \mathbf{I}$ ) its columns represent the basis of the DCT, and  $\boldsymbol{\Lambda} := \text{diag}(\boldsymbol{\lambda})$  is a diagonal matrix containing the eigen values of  $\mathbf{D}$ . Moreover, the  $k$ th entry of  $\boldsymbol{\lambda}$  is given by [29]:

$$\lambda_k = 2 - 2\cos((k-1)\pi/T), \quad k = 1, 2, \dots, T \quad (6)$$

For the case  $q = 2$ , the authors of [1,25–27] have assumed repeated boundary elements ( $x_0 = x_1$  and  $x_{T+1} = x_T$ ). This resulted in a second-order difference matrix  $\mathbf{D}_2 = -\mathbf{D}$ , where  $\mathbf{D}$  is as defined in Equation (4). Accordingly,  $\mathbf{D}_2^T \mathbf{D}_2 = \mathbf{D}^2$ , and its eigendecomposition is given by  $\mathbf{D}^2 = \mathbf{U} \boldsymbol{\Lambda}^2 \mathbf{U}^T$ . A generalisation of this result to the  $q$ th order difference matrix ( $q > 2$ ) was derived in [26] by expressing  $\mathbf{D}_q$  using  $\mathbf{D}_1$  and  $\mathbf{D}_2$ . For instance,  $\mathbf{D}_4 = \mathbf{D}_2 \mathbf{D}_2$  and  $\mathbf{D}_5 = \mathbf{D}_1 \mathbf{D}_2 \mathbf{D}_2$ . Following this procedure, it can be shown that  $\mathbf{D}_q^T \mathbf{D}_q = \mathbf{D}^q$ , and hence Equation (2) becomes:

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\text{argmin}} \frac{1}{2} \|\mathbf{y} - \mathbf{x}\|_2^2 + \frac{1}{2} \mu_L \mathbf{x}^T \mathbf{D}^q \mathbf{x}, \quad (7)$$

which has a solution given by Equation (3), with  $\mathbf{A}_q = (\mathbf{I} + \mu_L \mathbf{D}^q)$ . Substituting the eigen decomposition of  $\mathbf{D}^q := \mathbf{U} \boldsymbol{\Lambda}^q \mathbf{U}^T$  into the expression of  $\mathbf{A}_q$ , and applying the

unitary property of  $\mathbf{U}$ , it is straight forward to show that the solution of Equation (7) is given by:

$$\hat{\mathbf{x}} = \mathbf{U}\mathbf{\Gamma}_L\mathbf{U}^T\mathbf{y}, \quad (8)$$

where  $\mathbf{\Gamma}_L = \text{diag}(\gamma_L)$  is a diagonal matrix, and its  $k$ th diagonal entry is given by:

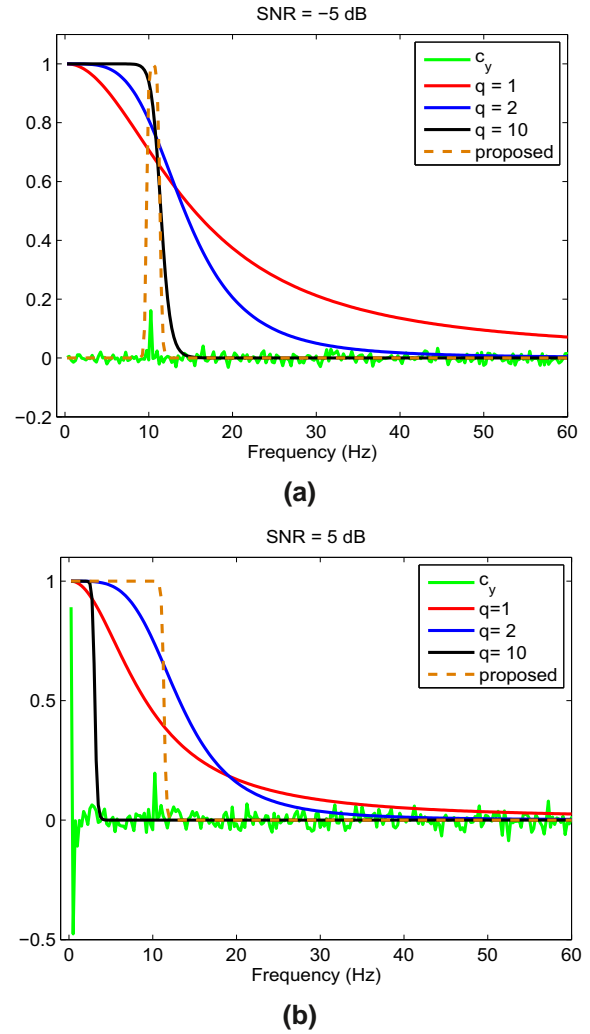
$$\gamma_L[k] = \frac{1}{1 + \mu_L \lambda_k^q}, \quad k = 1, 2, \dots, T, \quad (9)$$

Calculating the solution vector  $\hat{\mathbf{x}}$  of Equation (7) using Equation (8) can be efficiently performed using the following three steps:

- Step 1: Calculate the DCT coefficients  $\mathbf{c}_y$  (representing  $\mathbf{U}^T\mathbf{y}$ ) of the noisy input signal  $\mathbf{y}$ .
- Step 2: Construct the DCT coefficients  $\mathbf{c}_x$  of the smoothed signal  $\hat{\mathbf{x}}$  using  $\mathbf{c}_x = \gamma_L \odot \mathbf{c}_y$ , where  $\odot$  denotes the Hadamard product; that is, element-by-element multiplication, and the entries of  $\gamma$  are as defined in Equation (9).
- Step 3: Calculate the smoothed signal  $\hat{\mathbf{x}}$  as the inverse DCT of  $\mathbf{c}_x$ .

As can be observed from the three steps listed above, the solution to Equation (7) can be interpreted as a filtering (see Step 2) to the DCT coefficients  $\mathbf{c}_y$  of the noisy signal  $y(t)$ . The shape of the filter is determined by the weighting vector  $\gamma_L$  (the main diagonal of  $\mathbf{\Gamma}_L$ ), which in turn depends on the two controlling parameters  $q$  and  $\mu_L$ . As can be observed from Figure 1, the shape of  $\gamma_L$  is equivalent to an LPF. The dependency of the shape of the LPF on the two parameters  $q$  and  $\mu_L$  can be investigated by changing the value of one of the two parameters while fixing the value of the other one. Following this approach, it was observed that (this result is not presented due to space limitation) increasing either of the two parameters has the effect of reducing the transition band  $\Delta f$  from the pass-band to the stop-band of the LPF. On the other hand, it was also observed that, while increasing the value of  $q$  has the effect of increasing the BW of the LPF, increasing the value of  $\mu_L$  has the opposite effect.

Since the shape of  $\gamma_L$  is equivalent to an LPF, regardless of the spectral contents of  $x(t)$ , applying Equation (7) to smooth a bandpass signal, for example, a sinusoidal signal with frequency  $f_o > 0$ , will always result in a distorted signal due to the noise components in the frequency band  $0 \leq f < f_o$ . See Figure 1(a) for the case  $f_o = 10$  Hz. This represents one of the limitations associated with Equation (7). In addition to this limitation, existing implementations of Equation (7) suffer from the following additional two limitations. The first one is related to selecting the order  $q$  (of the difference matrix  $\mathbf{D}_q$ ) independently from the spectral contents of the smooth signal  $x(t)$ . As can be observed from Figure 1(a) for the case of smoothing a sinusoidal signal with frequency  $f_o = 10$  Hz, the value  $q = 10$  produced the best performance compared to the other two values of  $q = 1$  and  $q = 2$ . However, for the case presented in Figure 1(b), which



**FIGURE 1** Demonstration of the limitations associated with the existing PLS-based smoothing algorithms for the case of smoothing bandpass signal (a) and low-pass signal (b)

represents the same sinusoidal signal superimposed on a ramp signal, it is obvious that the case  $q = 10$  produced the worst performance compared to the other two values of  $q$ .

The second limitation associated with the existing implementation of Equation (7) is associated with the computation of the smoothing parameter  $\mu$ . Usually, the optimum value of  $\mu$  is calculated using the GCV technique [1,22,23,26]. However, the GCV technique requires a special optimisation tool to compute the optimum value of  $\mu$ . In addition, the GCV could fail to obtain the optimum value of  $\mu$  in case that inappropriate value of  $q$  was utilized. This case is presented in Figure 1. In this figure, the values of  $\mu$  associated with  $q = 1, 2$  and  $10$  were computed using the GCV approach. However, comparing the shapes of various  $\gamma$  curves in Figure 1(a) and (b) associated with each value of  $q$ , it is obvious that the presence of the low-frequency ramp signal in Figure 1(b) misleads the GCV approach. As a result, and for all values of  $q$ , large attenuations were introduced to the 10 Hz sinusoidal signal in Figure 1(b) compared to the case in Figure 1(a).

In the remaining part of this section, new objective function is suggested to overcome the first limitation associated with smoothing bandpass signals. In addition, closed-form expressions for the optimum values of  $q$  and  $\mu$  are derived to overcome the limitations associated with the existing implementation of Equation (7).

## 2.2 | Proposed objective function

In this subsection, a new objective function is proposed to overcome the limitation associated with Equation (7) in smoothing bandpass signals. As will be shown later, the proposed objective function will result in a weighting vector  $\gamma_B$  that can be adjusted to either have the characteristics of an LPF (similar to  $\gamma_L$  defined in Equation (9)) or a BPF. Recall that a BPF can be interpreted as a cascade of an LPF and a HPF with their cutoff frequencies properly selected. Therefore, for the objective function Equation (7) to result in a weighting vector  $\gamma_B$  having the characteristics of a BPF, a modification to Equation (7) is necessary. This modification can be performed by incorporating another penalty term that has the capability of producing a weighting vector  $\gamma_H$  having the characteristics of an HPF. This can be accomplished by observing that replacing the penalty term  $(\mu_L \mathbf{x}^T \mathbf{D}^q \mathbf{x})$  in Equation (7) by the following penalty term  $(1/\mu_H \mathbf{x}^T \mathbf{D}^{-p} \mathbf{x})$  and solving the resulting optimisation problem, the solution vector in this case will have the following expression  $\hat{\mathbf{x}} = \mathbf{U} \mathbf{\Gamma}_H \mathbf{U}^T \mathbf{y}$ , which is similar to Equation (8) with the exception that the diagonal entries  $\gamma_H$  of the diagonal matrix  $\mathbf{\Gamma}_H$  are given by:

$$\gamma_H[k] = \frac{1}{1 + \frac{1}{\mu_H} \lambda_k^{-p}} = 1 - \frac{1}{1 + \mu_H \lambda_k^p}, k = 1, \dots, T \quad (10)$$

It is obvious from Equation (10) that the weighting vector  $\gamma_H$  has the characteristics of an HPF, and the effect of the two parameters  $\mu_H$  and  $p$  on the shape of  $\gamma_H$  is similar to the effects of  $\mu_L$  and  $q$  on the shape of  $\gamma_L$ .

Based on the above discussion, the following objective function is proposed to get a waiting vector  $\gamma_B$  having the characteristics of a BPF:

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{y} - \mathbf{x}\|_2^2 + \frac{1}{2} \mathbf{x}^T \left( \mu_L \mathbf{D}^q + \frac{1}{\mu_H} \mathbf{D}^{-p} + \frac{\mu_L}{\mu_H} \mathbf{D}^{q-p} \right) \mathbf{x}, \quad (11)$$

where  $\mu_L$  and  $\mu_H$  are two regularisation parameters,  $q$  and  $p$  are two integer numbers representing the orders of the difference matrices  $\mathbf{D}_q$  and  $\mathbf{D}_p$ , the matrix  $\mathbf{D}^q := \mathbf{D}_q^T \mathbf{D}_q$ , and  $\mathbf{D}^{-p}$  is the inverse of the matrix  $\mathbf{D}^p := \mathbf{D}_p^T \mathbf{D}_p$ . The objective function Equation (7) is a special case of the proposed objective function Equation (11) for the case  $\mu_H = \infty$ . Even though the first two penalty terms in Equation (11) can result in a weighting

vector having the characteristics of a BPF, the incorporation of the third penalty term  $\frac{\mu_L}{\mu_H} \mathbf{x}^T \mathbf{D}^{q-p} \mathbf{x}$  simplifies the expression of the resulting weighting vector, and makes the calculation of the four controlling parameters ( $p$ ,  $q$ ,  $\mu_L$  and  $\mu_H$ ) straight forward, as will be shown later.

The minimisation of Equation (11) results in the following linear system of equations:

$$\left( \mathbf{I} + \mu_L \mathbf{D}^q + \frac{1}{\mu_H} \mathbf{D}^{-p} + \frac{\mu_L}{\mu_H} \mathbf{D}^{q-p} \right) \hat{\mathbf{x}} = \mathbf{y} \quad (12)$$

Applying the eigendecomposition Equation (5) of the matrix  $\mathbf{D}$  we get:

$$\mathbf{U} \left( \mathbf{I} + \mu_L \mathbf{\Lambda}^q + \frac{1}{\mu_H} \mathbf{\Lambda}^{-p} + \frac{\mu_L}{\mu_H} \mathbf{\Lambda}^{q-p} \right) \mathbf{U}^T \hat{\mathbf{x}} = \mathbf{y} \quad (13)$$

where  $\mathbf{\Lambda} := \text{diag}(\boldsymbol{\lambda})$  is a diagonal matrix containing the eigenvalues of  $\mathbf{D}$ , and the  $k$ th entry of  $\boldsymbol{\lambda}$  is defined in Equation (6). Since  $\mathbf{U}$  is a unitary matrix, and the overall matrix between the two brackets in the LHS of Equation (13) is diagonal, the solution vector  $\hat{\mathbf{x}}$  of Equation (13) can be efficiently calculated as:

$$\hat{\mathbf{x}} = \mathbf{U} \mathbf{\Gamma}_B \mathbf{U}^T \mathbf{y}, \quad (14)$$

where  $\mathbf{\Gamma}_B := \text{diag}(\gamma_B)$  is a diagonal matrix, and its  $k$ th diagonal entry is given by:

$$\begin{aligned} \gamma_B[k] &= \frac{1}{1 + \mu_L \lambda_k^q + \frac{1}{\mu_H} \lambda_k^{-p} + \frac{\mu_L}{\mu_H} \lambda_k^{q-p}} \\ &= \frac{1}{(1 + \mu_L \lambda_k^q)} \frac{1}{\left(1 + \frac{1}{\mu_H} \lambda_k^{-p}\right)} \\ &= \gamma_L[k] \gamma_H[k], \end{aligned} \quad (15)$$

where  $\gamma_L[k]$  and  $\gamma_H[k]$  are as defined in Equations (9) and (10), and they, respectively, represent the  $k$ th entries of the diagonal matrices  $\mathbf{\Gamma}_L := \text{diag}(\gamma_L)$  and  $\mathbf{\Gamma}_H := \text{diag}(\gamma_H)$ . The weighting vectors  $\gamma_L$  and  $\gamma_H$  are presented in Figure 2.

From Equation (15), it is obvious that the diagonal matrix  $\mathbf{\Gamma}_B$  can be expressed as  $\mathbf{\Gamma}_B = \mathbf{\Gamma}_L \mathbf{\Gamma}_H$ , and, hence Equation (14) can be expressed as:

$$\hat{\mathbf{x}} = \mathbf{U} \mathbf{\Gamma}_L \mathbf{\Gamma}_H \mathbf{U}^T \mathbf{y}, \quad (16)$$

For the case  $\mu_H = \infty$  we have  $\mathbf{\Gamma}_H = \mathbf{I}$ , and, hence, Equation (16) is reduced to the low-pass filtering Equation (8). In addition, setting  $\mu_L = 0$  results in  $\mathbf{\Gamma}_L = \mathbf{I}$ , and  $\hat{\mathbf{x}}$  calculated



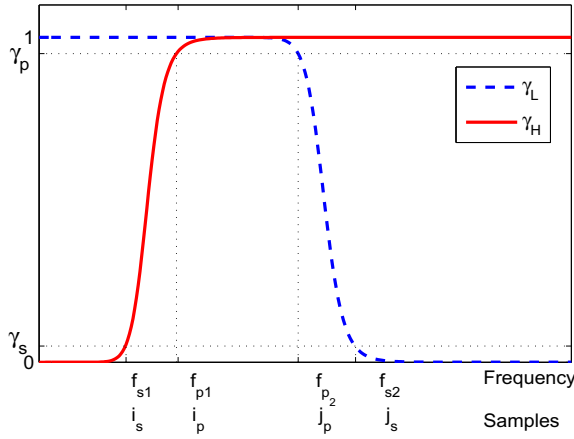


FIGURE 2 Characteristics and specifications of  $\gamma_L$  and  $\gamma_H$

using Equation (16) in this case is a high-pass filtered version of the input signal  $\mathbf{y}$ . Moreover, and as can be observed from Equation (15), the two parameters  $\mu_L$  and  $q$  control the shape of  $\gamma_L$ , while  $\mu_H$  and  $p$  control the shape of  $\gamma_H$ . Therefore, if these four parameters are selected such that  $\gamma_L$  and  $\gamma_H$  overlap as in Figure 2, then the weighting vector  $\gamma_B := \gamma_L \odot \gamma_H$  will have the characteristic of a BPF. In the next section, the spectral information of  $x(t)$  is utilized to derive closed-form expressions for the optimum values of  $q$ ,  $p$ ,  $\mu_L$  and  $\mu_H$ .

### 2.3 | Calculating the optimum values of $q$ , $p$ , $\mu_L$ and $\mu_H$

In this subsection, the spectral characteristics of  $\gamma_L$  are utilized first to find the closed-form expressions for the optimum values of  $q$  and  $\mu_L$ . Similar procedure will then be followed to find the closed-form expressions of  $p$  and  $\mu_H$  using the characteristics of  $\gamma_H$ . The optimum values of these four parameters are derived for the objective of confining the band of the estimated signal to the band of the imbedded smooth signal, and, hence, minimising the distortion introduced to the estimated signal.

Referring to Figure 2, it is assumed that the nonzero DCT coefficients of the smooth bandpass signal  $x(t)$  are confined in the frequency band  $f_{p1} \leq f \leq f_{p2}$ . In addition, it is also assumed that  $\gamma_L[j_p] = \gamma_p$ , where  $j_p = \lceil 2Tf_{p2}/F_s \rceil$  is the index associated with  $f_{p2}$ ,  $F_s$  is the sampling frequency,  $T$  is the total number of samples,  $\lceil a \rceil$  is the smallest integer number greater than  $a$  and  $0 \ll \gamma_p < 1$ . In addition, it is assumed that the upper stop-band edge frequency of the BPF is given by  $f_{s2} = f_{p2} + \Delta f$ , where  $\Delta f$  is a desired transition band, which usually has a small value, for example,  $\Delta f = 1$  Hz. Furthermore, it is assumed that  $\gamma_L[j_s] = \gamma_s$ , where  $j_s = \lceil 2Tf_{s2}/F_s \rceil$  is the index associated with  $f_{s2}$ , and  $0 < \gamma_s \ll 1$ . The following values  $\gamma_p = 0.99$  and  $\gamma_s = 0.01$  are utilized.

Substituting the values of  $j_p$  and  $j_s$  into the expression of  $\gamma_L$  we get:

$$\gamma_p = \frac{1}{1 + \mu_L \lambda_{j_p}^q} \quad (17)$$

$$\gamma_s = \frac{1}{1 + \mu_L \lambda_{j_s}^q} \quad (18)$$

where  $\lambda_{j_p}$  and  $\lambda_{j_s}$  are calculated using Equation (6) by substituting  $k = j_p$  and  $k = j_s$ , respectively. Rearranging Equations (17) and (18) to eliminate  $\mu_L$ , it is straight forward to show that:

$$\frac{1 - \gamma_p}{\gamma_p \lambda_{j_p}^q} = \frac{1 - \gamma_s}{\gamma_s \lambda_{j_s}^q}$$

which can be written in the following form:

$$\left( \frac{\lambda_{j_p}}{\lambda_{j_s}} \right)^q = \frac{(1 - \gamma_p)\gamma_s}{(1 - \gamma_s)\gamma_p}$$

Therefore, the optimum value of  $q$  can be calculated using the following expression:

$$q = \frac{\log\left(\frac{(1 - \gamma_p)\gamma_s}{(1 - \gamma_s)\gamma_p}\right)}{\log\left(\frac{\lambda_{j_p}}{\lambda_{j_s}}\right)} \quad (19)$$

This expression can be further simplified by applying the following two constraints on the desired shape of  $\gamma_L$ : (1)  $\gamma_s \ll 1$  and (2)  $\gamma_p = (1 - \gamma_s)$ . In this case, Equation (19) is reduced to the following simplified expression:

$$q = \frac{2\log(\gamma_s)}{\log\left(\frac{\lambda_{j_p}}{\lambda_{j_s}}\right)} \quad (20)$$

Substituting the expressions of  $\lambda_{j_p}$  and  $\lambda_{j_s}$  (defined in Equation (6)) into Equation (20), and using the properties of the trigonometric functions, the value of  $q$  can be expressed in terms of the design parameters  $\Delta f$  and  $f_{p2}$  as:

$$q = \frac{\log(\gamma_s)}{-\log\left(1 + \frac{\pi \Delta f}{F_s \tan\left[(f_{p2} - 1)\pi/F_s\right]}\right)} \quad (21)$$

Note that, since the quantity  $\pi \Delta f / F_s \ll 1$ , the following approximations were applied in deriving the expression in Equation (21):  $\cos(\pi \Delta f / F_s) \simeq 1$  and  $\sin(\pi \Delta f / F_s) \simeq \pi \Delta f / F_s$ .

The optimum value of  $\mu_L$  can then be obtained by substituting the calculated value of  $q$  into Equation (18) (or equivalently Equation (17)), and rearranging the equation to get the following expression:

$$\mu_L = \frac{\gamma_s}{(1 - \gamma_s)\lambda_{j_p}^q} \quad (22)$$

$$= \frac{\gamma_s}{(1 - \gamma_s) \left( 2 - 2\cos\left(2\pi(f_{p_2} - 1)/F_s\right) \right)^q}$$

A similar procedure can be followed to determine the optimum values of  $p$  and  $\mu_H$  using the expression of  $\gamma_H$ . In this case, the following two constraints are applied:  $\gamma_H[i_p] = \gamma_p$  and  $\gamma_H[i_s] = \gamma_s$ , where  $i_p$  and  $i_s$  are respectively, the indices associated with the lower pass-band edge frequency  $f_{p_1}$  and the lower stop-band edge frequency  $f_{s_1}$  of the desired BPF. See Figure 2. The derived expressions are:

$$p = \frac{2\log(\gamma_s)}{\log\left(\frac{\lambda_{i_s}}{\lambda_{i_p}}\right)} \quad (23)$$

$$= \frac{\log(\gamma_s)}{\log\left(1 - \frac{\pi \Delta f}{F_s \tan\left[(f_{p_1} - 1)\pi/F_s\right]}\right)}$$

$$\mu_H = \frac{(1 - \gamma_s)}{\gamma_s \lambda_{i_p}^p} \quad (24)$$

$$= \frac{(1 - \gamma_s)}{\gamma_s \left( 2 - 2\cos\left(2\pi(f_{p_1} - 1)/F_s\right) \right)^p}$$

Note that, following the proposed procedure, no special optimisation tools are required to compute the regularisation parameters (as in the GCV approach). In addition, the calculated values of  $q$ ,  $p$ ,  $\mu_L$  and  $\mu_H$  are optimum in the sense that they produce a desired response based on the spectral contents of the measured data.

## 2.4 | Estimating the edge frequencies $f_{p_1}$ and $f_{p_2}$

It was assumed in the previous subsection that the edge frequencies  $f_{p_1}$  and  $f_{p_2}$  are known a priori. However, in real practice, the values of these two frequencies are generally unknown. In this subsection, an iterative procedure based on applying a statistical test is suggested to estimate the edge frequencies  $f_{p_1}$  and  $f_{p_2}$ , or equivalently their indices  $i_p$  and  $j_p$ , respectively. The statistical test applied here is the F-test, which is used for testing the equality of two variances, and it can be implemented using the Matlab function 'varTest2.m'. See [30] for more details about the F-test.

Let's start first with the procedure of estimating the value of  $j_p$ . It is obvious that all the DCT coefficients  $\mathbf{c}_y[j_p : T]$  are noise coefficients. Therefore, in the proposed

**TABLE 1** The proposed BCSA algorithm

$[\hat{\mathbf{x}}] = \text{BCSA}(\mathbf{y}, \alpha_0, \gamma_s, \Delta f, F_s)$

1. Calculate  $\mathbf{c}_y = \text{DCT}(\mathbf{y})$ , and  $\gamma_p = 1 - \gamma_s$ .
2. Set  $\hat{j}_p = \lceil 0.8T \rceil$ ,  $a = \lceil (1 + \hat{j}_p)/2 \rceil$ ,  $\mathbf{c}_j = \mathbf{c}_y[\hat{j}_p : T]$ ,  $\mathbf{c}_a = \mathbf{c}_y[a : \hat{j}_p]$
3. Calculate  $\widehat{SNR}$  using Equation (25), and set  $\alpha = \alpha_0/\widehat{SNR}$ .
4. while  $\hat{j}_p \geq a + 2$  (% updating the index  $\hat{j}_p$  associated with  $f_{p_2}$ )
  - (a) Apply the F-test using the Matlab command:  $b = \text{varTest2}(\mathbf{c}_a, \mathbf{c}_j, \text{'tail', 'right', 'alpha'}, \alpha)$ ;
  - (b) Check the result of the test:
    - if  $b = 1$  (i.e.  $\mathbf{c}_a$  and  $\mathbf{c}_j$  are significantly different)
      - Update:  $a \leftarrow \lceil (a + \hat{j}_p)/2 \rceil$ ,  $\mathbf{c}_a = \mathbf{c}_y[a : \hat{j}_p]$
      - elseif  $b = 0$  (i.e.  $\mathbf{c}_a$  and  $\mathbf{c}_j$  are not significantly different)
        - Update:  $\hat{j}_p \leftarrow a$ ,  $a \leftarrow \lceil (1 + \hat{j}_p)/2 \rceil$ ,  $\mathbf{c}_j = \mathbf{c}_y[\hat{j}_p : T]$ ,
    - $\mathbf{c}_a = \mathbf{c}_y[a : \hat{j}_p]$ ,  $\widehat{SNR}$  using Equation (25),  $\alpha \leftarrow \alpha_0/\widehat{SNR}$
5. Set  $\hat{i}_p = 1$  and  $b = \lceil (1 + \hat{j}_p)/2 \rceil$ .
6. while  $b \geq \hat{i}_p + 2$  (% updating the index  $\hat{i}_p$  associated with  $f_{p_1}$ )
  - (a) Update:  $\mathbf{c}_b = \mathbf{c}_y[1 : b]$
  - (b) Apply the F-test using the Matlab command:  $b = \text{varTest2}(\mathbf{c}_b, \mathbf{c}_j, \text{'tail', 'right', 'alpha'}, \alpha)$ ;
  - (c) Check the result of the test:
    - if  $b = 1$  : Update  $b \leftarrow \lceil (b + \hat{i}_p)/2 \rceil$
    - elseif  $b = 0$  : Update  $\hat{i}_p \leftarrow b$  and  $b \leftarrow \lceil (\hat{i}_p + \hat{j}_p)/2 \rceil$
7. Calculate  $f_{p_1} = \lceil F_s \hat{i}_p / 2T \rceil$  and  $f_{p_2} = \lceil F_s \hat{j}_p / 2T \rceil$ .
8. Utilize  $f_{p_2}$  to calculate  $q$  and  $\mu_L$  using Equations (21) and (22), respectively.
9. If  $f_{p_1} > \Delta f$ , calculate  $p$  and  $\mu_H$  using Equations (23) and (24), respectively. Otherwise, set  $\mu_H = \infty$ .
10. Utilize  $q$ ,  $\mu_L$ ,  $p$  and  $\mu_H$  to calculate  $\gamma_B$  using Equation (15).
11. Calculate:  $\hat{\mathbf{x}} = \text{IDCT}(\gamma_B \odot \mathbf{c}_y)$ .

procedure, a relatively large value  $1 \ll \hat{j}_p < T$  is first used as an initial guess of the value of  $j_p$  to ensure that all the coefficients  $\mathbf{c}_j := \mathbf{c}_y[\hat{j}_p : T]$  are noise coefficients. For all the examples presented, the value of  $\hat{j}_p$  was always initiated to  $\hat{j}_p = \lceil 0.8T \rceil$ . The value of  $\hat{j}_p$  is then iteratively decreased using the iterative procedure presented by Step 4 in Table 1.

A similar procedure has been followed to estimate the lower index  $i_p$ . Since, the coefficients  $\mathbf{c}_j[1 : i_p]$  are noise coefficients, an initial guess  $\hat{i}_p = 1$  is first applied, and then the value of  $\hat{i}_p$  is iteratively increased using the iterative procedure presented by Step 6 in Table 1. Note that, in applying this iterative procedure, the previously detected noise coefficients  $\mathbf{c}_j$  are utilized as reference noise coefficients.

In applying the proposed algorithm, it was found that the optimum value of the significance level  $\alpha$  (used with the F-test) has to be selected inversely proportional to the input  $SNR$ . Therefore, it is suggested to select the value of  $\alpha$  using the following formula  $\alpha = \alpha_0/\widehat{SNR}$ , where  $\widehat{SNR}$  is an estimate of the input  $SNR$ , and  $\alpha_0$  is a small positive number, for example, 0.01. It is suggested to calculate  $\widehat{SNR}$  from the DCT coefficients  $\mathbf{c}_y$  of the measured data  $y(t)$  using the following equation:

$$\widehat{SNR} = \frac{\frac{1}{T}\|\mathbf{c}_y\|_2^2 - \frac{1}{T - \hat{j}_p + 1}\|\mathbf{c}_j\|_2^2}{\frac{1}{T - \hat{j}_p + 1}\|\mathbf{c}_j\|_2^2}, \quad (25)$$

$$= \left(1 - \frac{\hat{j}_p - 1}{T}\right) \frac{\|\mathbf{c}_y\|_2^2}{\|\mathbf{c}_j\|_2^2} - 1$$

where  $\mathbf{c}_j = \mathbf{c}_y[\hat{j}_p : T]$ . Note that, for better estimate of the  $\widehat{SNR}$ , the value of the  $\widehat{SNR}$  has to be updated each time a new estimate of  $\hat{j}_p$  is obtained.

The proposed BCSA is summarized in Table 1. The algorithm accepts five inputs and produces the smoothed signal  $\hat{\mathbf{x}}$  as an output. Note that, in Step 9 in Table 1, the input signal  $\mathbf{y}$  is considered as a bandpass signal only if the estimated value of the lower pass-band edge frequency  $f_{p_1}$  satisfies  $f_{p_1} > \Delta f$ .

## 2.5 | Dealing with missing and outlier samples

In addition to the contaminating noise, measured data are also vulnerable to missing and/or outlier samples. Missing samples can be resulted, for instance, from instrument malfunction or cease of measurements during maintenance periods. Missing samples can be also introduced intentionally, for example, in the process of over-sampling a data series [6]. On the other hand, outlier samples are defined as extreme values that deviate from other sample values in the measured data. They may indicate experimental errors or variability in the measurements.

Before explaining the procedure to handle missing and outlier samples, let's first define a diagonal matrix  $\mathbf{S} \in \mathbb{R}^{T \times T}$ , its diagonal entries equal one except at the indices corresponding to the missing and the outlier samples, at which the diagonal entries equal zero. The complement of  $\mathbf{S}$ , denoted as  $\bar{\mathbf{S}}$ , satisfies the relation  $\mathbf{S} + \bar{\mathbf{S}} = \mathbf{I}$ , where  $\mathbf{I} \in \mathbb{R}^{T \times T}$  is the identity matrix. Then, to handle the case of having missing and/or outlier samples, it is suggested to encourage the optimisation problem in Equation (11) to focus only on the most reliable samples in the measured data vector  $\mathbf{y}$ . This can be achieved by incorporating  $\mathbf{S}$  into Equation (11) as follows:

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{S}(\mathbf{y} - \mathbf{x})\|_2^2 + \frac{1}{2} \mathbf{x}^T \left( \mu_L \mathbf{D}^q + \frac{1}{\mu_H} \mathbf{D}^{-p} + \frac{\mu_L}{\mu_H} \mathbf{D}^{q-p} \right) \mathbf{x}, \quad (26)$$

The minimisation of Equation (26) results in the following linear system:

$$\left( \mathbf{S} + \mu_L \mathbf{D}^q + \frac{1}{\mu_H} \mathbf{D}^{-p} + \frac{\mu_L}{\mu_H} \mathbf{D}^{q-p} \right) \hat{\mathbf{x}} = \mathbf{S} \mathbf{y} \quad (27)$$

Substituting  $\mathbf{S} = \mathbf{I} - \bar{\mathbf{S}}$  into the LHS of Equation (27), and rearranging the resulting equation we get:

$$\left( \mathbf{I} + \mu_L \mathbf{D}^q + \frac{1}{\mu_H} \mathbf{D}^{-p} + \frac{\mu_L}{\mu_H} \mathbf{D}^{q-p} \right) \hat{\mathbf{x}} = \mathbf{S} \mathbf{y} + \bar{\mathbf{S}} \hat{\mathbf{x}} \quad (28)$$

Comparing Equation (28) with Equation (12) we find that, even though the LHS of both equations is identical, the solution vector  $\hat{\mathbf{x}}$  appears on both sides of Equation (28). Therefore, Equation (28) can be solved iteratively as follows: 1) starting with an initial solution vector  $\hat{\mathbf{x}}_0 = \mathbf{0}$ , then the solution vector can be updated using the following equation, which has a form similar to Equation (14):

$$\hat{\mathbf{x}}_{k+1} = \mathbf{U} \mathbf{T}_{B,k} \mathbf{U}^T \mathbf{y}_k, \quad k = 0, 1, \dots \quad (29)$$

where  $\mathbf{y}_k = \mathbf{S} \mathbf{y} + \bar{\mathbf{S}} \hat{\mathbf{x}}_k$ , and the parameters used for constructing  $\mathbf{T}_{B,k}$  are estimated from the DCT of  $\mathbf{y}_k$ . Note that, the entries of  $\mathbf{y}_k$  corresponding to the reliable samples are given by  $\mathbf{S} \mathbf{y}$ , and, hence, they equal the corresponding entries in  $\mathbf{y}$ . On the other hand,  $\bar{\mathbf{S}} \hat{\mathbf{x}}_k$  represents the entries of  $\mathbf{y}_k$  corresponding to the missing and/or outlier samples, which are updated in each iteration by the corresponding entries in  $\hat{\mathbf{x}}_k$ .

For the case of missing samples, the indices  $\mathcal{J}_m$  of the missing samples are usually known a priori. However, the indices  $\mathcal{J}_o$  of the outlier samples are generally unknown, and, hence, they have to be estimated. Starting with an initial estimate  $\mathcal{J}_o = []$ , and running Equation (29), it was observed that most of the outlier samples appear in the residual signal  $\mathbf{v}_k := \mathbf{y} - \hat{\mathbf{x}}_{k+1}$ . Therefore, it is suggested to estimate  $\mathcal{J}_o$  from the residual signal  $\mathbf{v}_k$  using the procedure presented in Step 2 (d-g) in Table 2, which is inspired by the procedure followed in [31]. The success of the proposed algorithm in correcting outlier samples depends crucially on the success of this iterative procedure in correctly estimating the standard deviation of  $\mathbf{v}$  from the residual signal  $\mathbf{v}_k$ , and hence estimating the optimum value of the threshold parameter  $\xi_k$ . Therefore, it is necessary to identify the breakdown point of this iterative procedure, that is, the maximum number of outlier samples beyond which the procedure fails.

Unfortunately, it is not easy to derive mathematical expression for the breakdown point. Therefore, it is suggested

**TABLE 2** The proposed RBCSA algorithm

$$[\hat{\mathbf{x}}] = \text{RBCSA}(\mathbf{y}, \alpha_0, \gamma_s, \Delta f, F_s, \mathcal{J}_m)$$

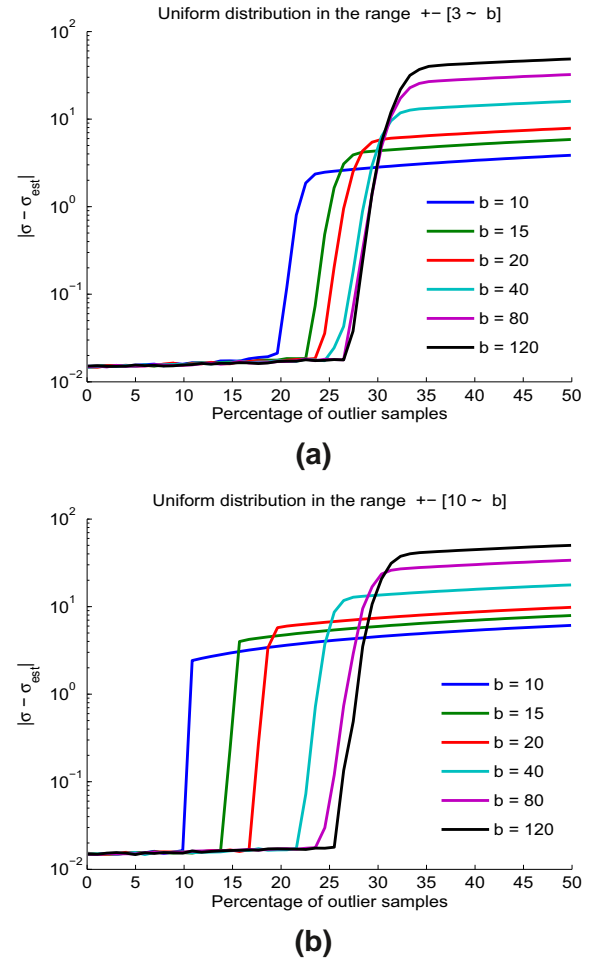
1. Initialisation:  $\hat{\mathbf{x}}_0 = \mathbf{0}$ , and  $\mathcal{J}_o = []$ .
2. for  $k = 0 : K_{\max}$ 
  - (a) Construct  $\mathbf{y}_k$ :  $\mathbf{y}[\mathcal{J}_m] = \hat{\mathbf{x}}_k[\mathcal{J}_m]$ ,  $\mathbf{y}_k \leftarrow \mathbf{y}$ ,  $\mathbf{y}_k[\mathcal{J}_o] \leftarrow \hat{\mathbf{x}}_k[\mathcal{J}_o]$ .
  - (b) Calculate:  $[\hat{\mathbf{x}}_{k+1}] = \text{BCSA}(\mathbf{y}_k, \alpha_0, \gamma_s, \Delta f, F_s)$ .
  - (c) Terminate if  $\|\hat{\mathbf{x}}_{k+1} - \hat{\mathbf{x}}_k\|_2 / \|\hat{\mathbf{x}}_{k+1}\|_2 \leq \epsilon$ .
  - (d) Calculate:  $\mathbf{v}_k = \mathbf{y} - \hat{\mathbf{x}}_{k+1}$ .
  - (e) Set  $\bar{\mathbf{v}}_k = \mathbf{v}_k$ , and calculate  $\xi_k = 3std(\bar{\mathbf{v}}_k)$ .
  - (f) while  $\max(|\bar{\mathbf{v}}_k|) > \xi_k$
- Exclude outlier samples:  $\bar{\mathbf{v}}_k(|\bar{\mathbf{v}}_k| > \xi_k) = []$ ,  
Update:  $\xi_k \leftarrow 3std(\bar{\mathbf{v}}_k)$ .
- (g) Identify the outlier indices  $\mathcal{J}_o$  as the indices of  $\mathbf{v}_k$  its magnitude values exceed  $\xi_k$ .
3.  $\hat{\mathbf{x}} = \hat{\mathbf{x}}_{k+1}$ .



to follow the following Monte Carlo simulations to investigate the behaviour of the iterative procedure, and hence approximately identifying its breakdown point. In this simulation, a random noise vector  $\mathbf{v} \in \mathbb{R}^{1000}$  is first generated from a Gaussian distribution with zero mean and unit variance. Then,  $\delta\%$  of the samples of  $\mathbf{v}$  is replaced by random outlier values uniformly distributed in the range  $\pm[\alpha \sim b]$ . The values of  $\delta$  considered in this simulation increases from 0.1 to 50 in steps of 1. The value of  $\alpha$  represents the smallest artefact value, and hence, it has to be selected  $\geq 3\sigma_v$ , where  $\sigma_v = 1$  is the standard deviation of the samples in  $\mathbf{v}$ . The values of  $\alpha$  considered in this simulation are  $\alpha = 3$  and  $\alpha = 10$ , and, with each value of  $\alpha$ ,  $b$  takes one of the following values [10, 15, 20, 40, 80, 120] to investigate the impact of the largest artefact value on the performance of the iterative procedure. For each combination of  $\delta$ ,  $\alpha$  and  $b$ , new artefacts are generated and their locations are randomly selected. The iterative procedure is then applied on the constructed vector to estimate the value of its standard deviation. The absolute difference between  $\sigma_v$  and its estimate is then calculated. This procedure is repeated 2000 different times for each combination of  $\delta$ ,  $\alpha$  and  $b$ , and the average value of the calculated difference is computed. The result of this simulation is shown in Figure 3.

As can be observed from Figure 3, the largest value of  $\delta$  at which the iterative procedure fails (i.e. the breakdown point) equals 26.5% and 25.5% for the cases  $\alpha = 3$  and  $\alpha = 10$ , respectively. However, this large value occurs only when the dynamic range of the artefact is large, that is, when  $b \gtrsim 100$ . In addition, there is a direct relation between the breakdown point and the value of  $b$ , and the worst case occurs when  $b = \alpha = 10$ . In this case, the iterative procedure can correctly estimate the standard deviation of the noise as long as the percentage number of outlier samples is lower than 10%. This experiment was repeated for a wide range of values of  $\alpha$  and  $b$  and it was found that the breakdown point is always less than 30%. In addition, for all the considered cases for which  $b = \alpha \geq 10$ , the breakdown point was found to be  $\approx 10\%$ . The breakdown point decreased to 5% when the following values  $b = \alpha = 4$  were selected. However, in this last case, the values of the added samples are too small to be considered as artefacts. From the above discussion, the following conclusion can be drawn. The breakdown point of the iterative procedure is approximately confined to the range 10%~25%, where the upper value can be obtained only if the artefacts have wide range of values, and the lower value is obtained when all the artefact values  $= \pm\alpha$ , where  $\alpha \geq 10$ .

The overall algorithm, which is referred to as RBCSA, is summarised in Table 2. As shown in Table 2, the inputs to the RBCSA algorithm are identical to those of the BCSA, in addition to the indices  $\mathcal{J}_m$  of the missing samples. If the input vector  $\mathbf{y}$  does not have missing samples, then the input  $\mathcal{J}_m$  is simply entered as an empty set  $\mathcal{J}_m = []$ . The algorithm terminates if either a maximum number of iterations  $K_{\max}$  is reached (e.g.  $K_{\max} = 100$ ), or the condition in Step 2(c) is satisfied, where  $\epsilon$  is a small number (e.g.  $\epsilon = 10^{-4}$ ). It is worth mentioning that the developed RBCSA algorithm cannot be



**FIGURE 3** Determining the breakdown point associated with the iterative procedure used for identifying the outlier samples. The outliers in this simulation have uniform distribution in the range  $\pm[a \sim b]$ , where  $b \in [10, 15, 20, 40, 80, 120]$ , and  $a$  either equals 3 (a) or 10 (b)

used for outlier correction in case that the original signal itself contains high-amplitude samples, for example, the R-peaks in ECG signals. In such cases, a variant of the iterative procedure presented in Step 2(d–g) in Table 2 is needed to distinguish the high-amplitude signal values from the contaminating outlier samples. This will be considered as a topic for future research. The Matlab codes of the two algorithms BCSA and RBCSA are available upon request.

### 3 | SIMULATIONS AND RESULTS

In this section, five different examples are presented to examine the performance of the proposed algorithms. In the first four examples, simulated signals are generated and AWGN is added at specified values of the SNR. The performance of the various algorithms considered in this section is quantified using the signal-to-noise ratio improvement ( $SNR_{\text{imp}}$ ) in dB. The  $SNR_{\text{imp}}$  is computed using the following equation [26]:

$$SNR_{imp}[dB] = 10 \log_{10} \frac{\sum_{t=1}^T (y(t) - x(t))^2}{\sum_{t=1}^T (x(t) - \hat{x}(t))^2}, \quad (30)$$

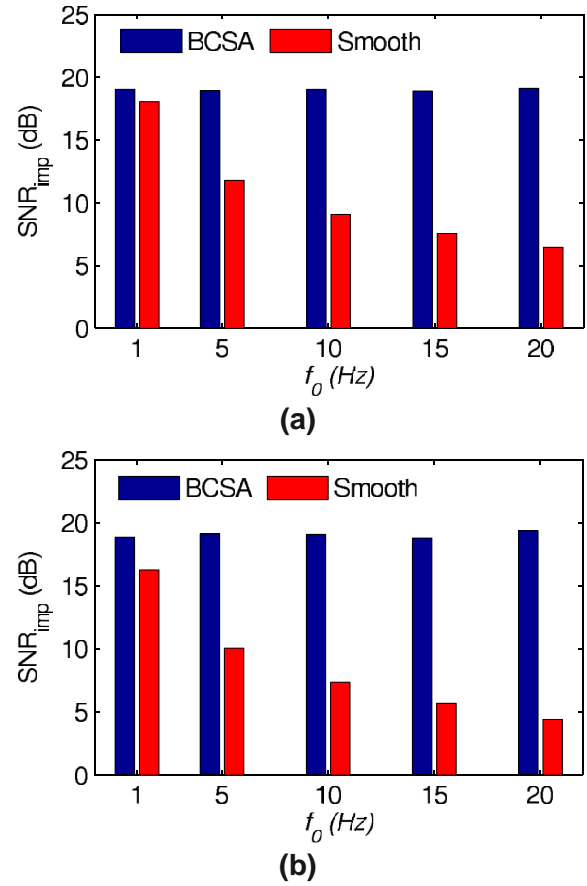
where  $y(t)$  is the measured signal,  $T$  is the number of samples,  $x(t)$  is the smooth signal and  $\hat{x}(t)$  is its estimate. In addition, the following parameters  $\gamma_s = 0.01$ ,  $\Delta f = 1$  Hz and  $\alpha_0 = 0.01$  are used with the proposed algorithms.

### 3.1 | Example 1: dependency of the performance on the lower edge frequency $f_{p_1}$

As stated before, all the PLS smoothing algorithms based on Equation (2) behave as LPFs, while the proposed BCSA algorithm has the capability of switching between LPF and BPF based on the significant DCT coefficients of  $x(t)$ . Therefore, for smoothing bandpass signals, it is expected that the performance of the existing PLS smoothing algorithms (e.g. the smooth algorithm developed in [1]) degrades as the value of the lower edge frequency  $f_{p_1}$  of  $x(t)$  increases. The effect of the value of  $f_{p_1}$  on the performance of the smooth algorithm [1] and the proposed BCSA is investigated in this example. In this comparison, a noisy bandpass signal  $y(t)$  is constructed by first generating a sinusoidal signal with frequency  $f_0$  Hz, and then adding AWGN at a specified value of the SNR. The following five values of  $f_0 = 1, 5, 10, 15$  and  $20$  Hz are considered, while only the following two extreme values of  $SNR = 0$  and  $15$  dB are considered. The constructed noisy signal  $y(t)$  is applied to the two algorithms, and the  $SNR_{imp}$  associated with each algorithm is then calculated using the obtained smooth signals. This procedure is repeated 200 different times for each combination of  $f_0$  and  $SNR$ , where in each time a new AWGN is generated. The value of the  $SNR_{imp}$  associated with each algorithm is then calculated as the average over the 200 runs. The results are shown in Figure 4. As can be observed from Figure 4, and for the two considered values of the SNR, the  $SNR_{imp}$  associated with the proposed algorithm is almost independent on the value of  $f_0$ , while the  $SNR_{imp}$  associated with the smooth algorithm decreases exponentially as the value of  $f_0$  increases. Thanks to the bandpass switching capability associated with the proposed BCSA algorithm.

### 3.2 | Example 2: qualitative comparison

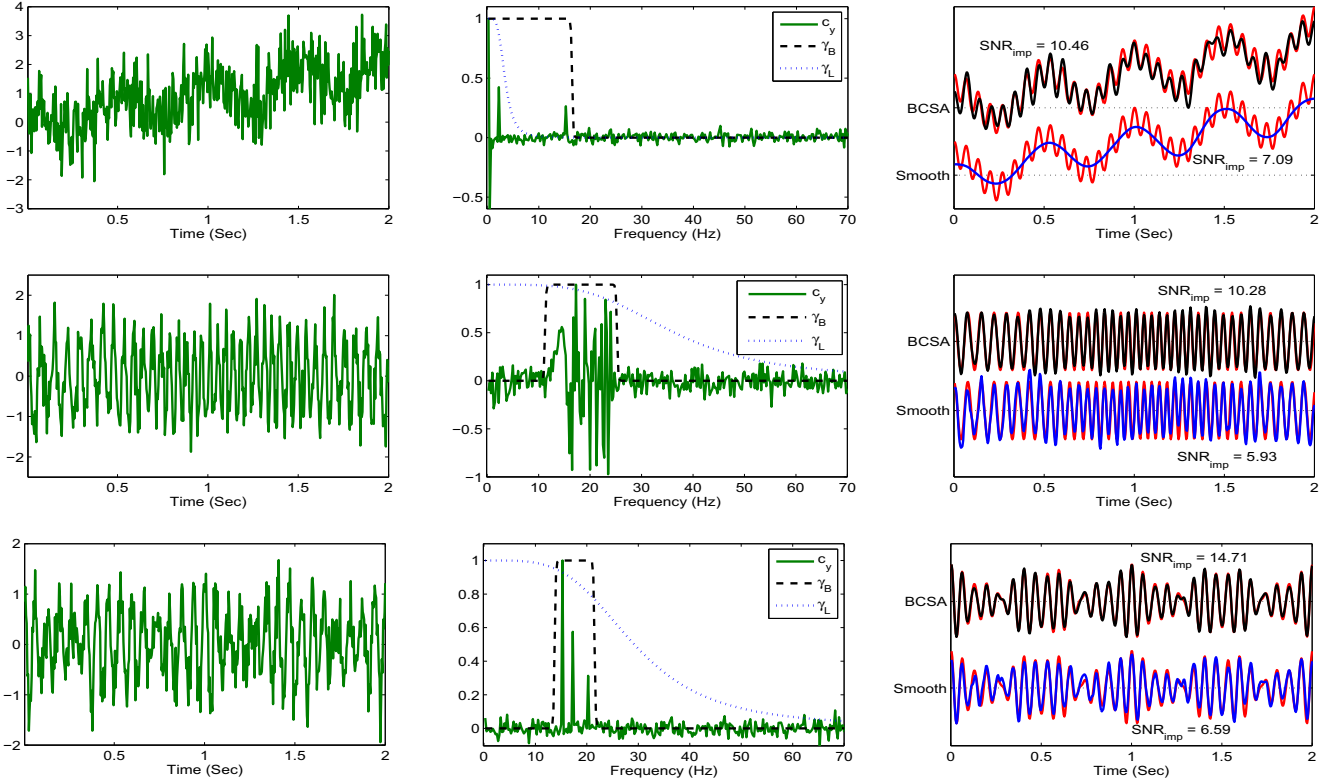
In this example, the performance of the proposed BCSA algorithm is qualitatively compared with that of the smooth algorithm [1], which belongs to the family of PLS smoothing algorithms. Three different signals are utilised in this comparison. The duration of each signal equals 2 s, while the signals are sampled at 256 Hz. Therefore, each signal consists of  $T = 512$  samples. The first signal consists of a ramp signal in addition to two sinusoidal signals having frequencies 2 and 15 Hz. Due to the presence of the ramp signal, the DCT coefficients of this composite signal start from the zero frequency, and, hence, it represents a low-pass signal. The second



**FIGURE 4** Comparison between the proposed BCSA algorithm and the smooth algorithm for the case of smoothing a sinusoidal signal having frequency  $f_0$  and contaminated with AWGN at 0 dB (a) and 15 dB (b)

signal is a frequency modulated signal, while the third signal consists of three sinusoidal signals having frequencies 15, 17 and 20 Hz. None of these two signals has DCT coefficients near the zero frequency, and, hence, they can be considered as bandpass signals. For each signal, AWGN is added at 5 dB SNR, and the noisy signal is applied to both algorithms. The results are presented in Figure 5.

The middle column in Figure 5 presents the DCT coefficients of the three signals, as well as the weighting vectors  $\gamma_L$  and  $\gamma_B$  estimated by the smooth algorithm and the BCSA algorithm, respectively. As can be observed from the three subfigures in the middle column, the nonzero entries of the weighting vector  $\gamma_B$  estimated by the proposed BCSA algorithm are always confined to the BW of the embedded smooth signal  $x(t)$ . Therefore, the amount of distortion introduced to the estimated smooth signal by the out-of-band noise components is minimised by the proposed algorithm. No such performance can be observed in the weighting vector  $\gamma_L$  estimated by the smooth algorithm, even in the case of the low-pass signal presented in the upper subfigures. The main reason behind the limited performance of the smooth algorithm, compared to the proposed BCSA algorithm, resides in the way of selecting the parameters  $q$  and  $\mu$ . In the smooth algorithm, the value of  $q$  is set equal to two, while the



**FIGURE 5** Smoothing three different signals. **Left:** the measured noisy signals. **Middle:** the DCT coefficients of the measured signals (solid green curves), and the weighting vectors  $\gamma_B$  and  $\gamma_L$  estimated, respectively, by the BCSA and the smooth algorithms. **Right:** the smooth signals estimated by the BCSA and the smooth algorithm (represented by the black and the blue waveforms, respectively) superimposed on the original smooth signal  $x(t)$  represented by the red waveforms

smoothing parameter  $\mu$  is selected using the GCV approach. For this low value of  $q$ , the GCV alone cannot confine the nonzero entries of  $\gamma_L$  to the band of  $x(t)$ . As a result, and as can be observed from the right column in Figure 5, the proposed BCSA algorithm outperforms the smooth algorithm for the three signals considered in this example. The values presented on the subfigures on the right column represent the  $SNR_{imp}$  in dB associated with each algorithm. It is evident that the proposed algorithm has estimated the imbedded smooth signals with relatively high values of the  $SNR_{imp}$  compared to the smooth algorithm. More qualitative and quantitative comparisons between the two algorithms are presented in the next two examples.

### 3.3 | Example 3: comparison with other smoothing techniques

In this example, the performance of the proposed BCSA algorithm is compared with the performance of the following algorithms: smooth [1], the cubic spline *CSpline* (implemented using the Matlab function '*csapsGCV.m*' [32]), the SSA [10], the SG filter [4] (implemented using the Matlab function '*sgolayfilt.m*'), the discrete wavelet transform coefficient shrinkage (implemented using the Matlab function '*wden.m*') and the

complementary EEMD with adaptive noise (CEEMDAN) [14], which, for simplicity, is abbreviated as EEMD. Recall that the EEMD algorithm decomposes a signal into a set of IMFs in decreasing order of their frequency contents. Therefore, to automatically identify the IMFs related to the noise, it is suggested to calculate the autocorrelation parameter  $c_i := \sum_{t=2}^T u_i(t)u_i(t-1)$ , where  $u_i(t)$  is the normalised  $i$ th IMF. For the IMFs associated with the noise, the value of  $c_i$  will be small compared to the other IMFs associated with the smooth signal. Therefore, for smoothing a signal, the first  $i_v$  IMFs are first discarded, and the smoothed signal is then constructed by summing the remaining IMFs, where  $i_v$  is identified as the index at which the coefficients  $c_i$  have abrupt increase in its value.

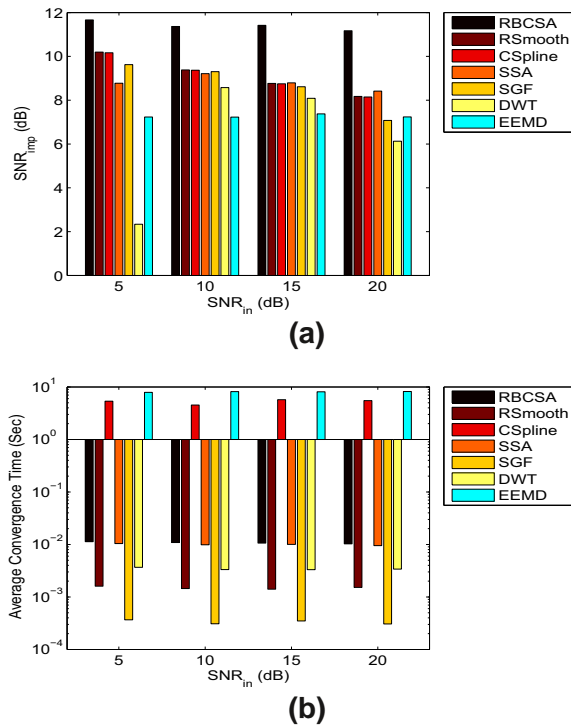
In this example, a noisy signal  $y(t)$  is constructed by first generating a smooth signal  $x(t)$ , and then adding AWGN at a specified value of the SNR. The duration of  $x(t)$  equals 4 s, and it was sampled at 256 Hz. Therefore,  $x(t)$  consists of  $T = 1024$  samples. In addition,  $x(t)$  is composed of three different sinusoidal signals having frequencies 2.5, 5 and 10 Hz, and the considered values of the SNR are 5, 10, 15 and 20 dB. For each value of the SNR, the constructed noisy signal is applied to the competing algorithms, and a smooth signal is obtained from each algorithm. The  $SNR_{imp}$  associated with each algorithm is then calculated. This procedure is repeated 200 different times for each value of the SNR, where in each time a new AWGN is

generated. The  $SNR_{imp}$  associated with each algorithm is then calculated as the average over the 200 runs. The results are shown in Figure 6(a). As can be observed from this subfigure, the proposed algorithm outperforms all the competing algorithms for all the considered values of the SNR.

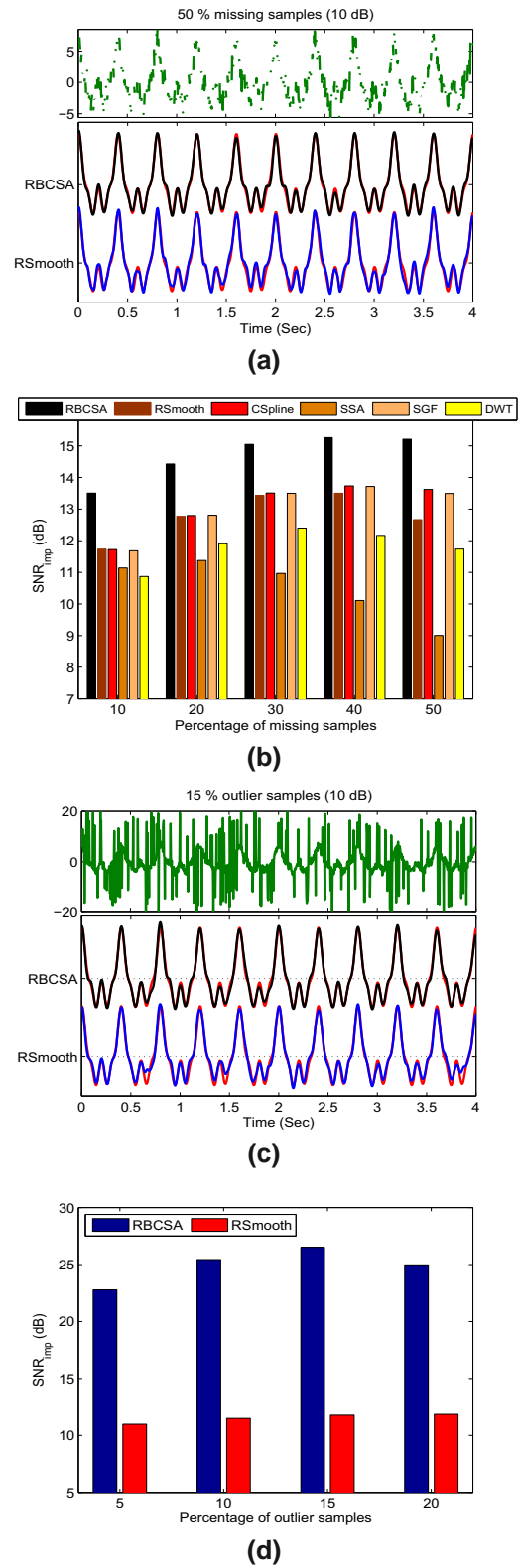
On the other hand, the average convergence time of the seven algorithms considered in this comparison is presented in Figure 6(b). As can be observed from this subfigure, the competing algorithms can be classified in terms of the average convergence time into two classes. The first class contains the two algorithms (CEEMDAN and the cubic spline) that have much higher execution time compared to the remaining algorithms, which belong to the second class. In addition, it can be also observed from Figure 6(b) that both the BCSA and the SSA algorithms have the highest average convergence time among all the algorithms in the second class. However, the average convergence time of these two algorithms is about 10 ms, which is much smaller than the signal duration. Therefore, all algorithms in the second class can be adapted to be used in real-time applications.

### 3.4 | Example 4: missing and outlier samples

In this example, the performance of the proposed RBCSA algorithm in smoothing signals associated with missing or outlier samples is examined. For the case of missing samples,



**FIGURE 6** Comparison between the proposed BCSA and six other techniques in terms of the  $SNR_{imp}$  in dB (a) and the average convergence time in seconds (b) as functions of the input SNR in dB



**FIGURE 7** (a) Signals from top to bottom are: a noisy signal with 50% missing samples, and the estimated signals using RBCSA and RSmooth, (b) the average  $SNR_{imp}$  for the case of missing samples, (c) signals from top to bottom are: a noisy signal with 15% outlier samples, and the estimated signals using RBCSA and RSmooth and (d) the average  $SNR_{imp}$  for the case of outlier samples

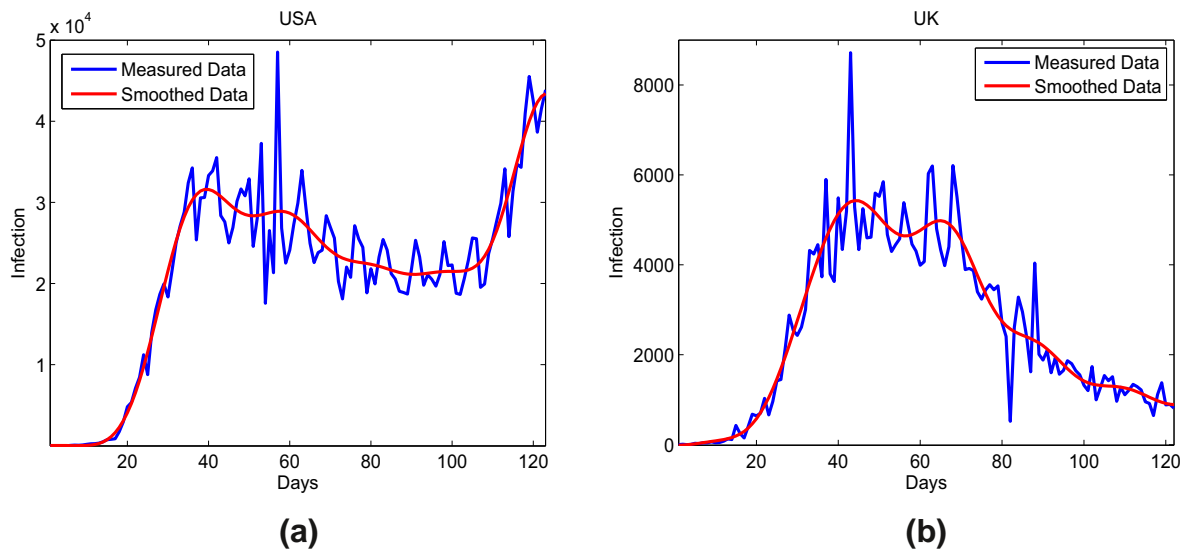
the algorithms considered in Example 3 are iteratively applied to the constructed signal in a manner similar to the RBCSA in Table 2. The EEMD is not considered in this comparison due to its high computation cost. For the case of outlier samples, the proposed RBCSA is compared only with the robust smooth algorithm 'RSmooth' developed in [1], because only these two algorithms are capable of identifying the indices of the outlier samples. In both cases, the noisy signal  $y(t)$  is first constructed by adding AWGN at 10 dB to the smooth signal  $x(t)$  utilized in Example 3. Then, for the case of missing samples,  $\delta\%$  of the constructed noisy signal  $y(t)$  is excluded, where the indices of these samples are randomly selected. The values of  $\delta$  considered in this example are 10, 20, ..., 50%. A sample of the constructed noisy signal for the case  $\delta = 50\%$  is shown in the upper panel in Figure 7(a). The smoothed signals obtained by the proposed RBCSA and the RSmooth algorithms are shown in the lower panel in Figure 7(a). To quantitatively assess the performance of the various algorithms, and for each value of  $\delta$ , the  $SNR_{imp}$  associated with each algorithm is calculated as the average over 200 different runs, where in each run new AWGN is added at 10 dB and new  $\delta\%$  indices of the constructed noisy signal are discarded. The average  $SNR_{imp}$  associated with all algorithms is shown in Figure 7(b). As shown in this figure, and for all the considered values of  $\delta$ , the proposed RBCSA outperforms all the competing algorithms.

For the case of outlier samples, the signals are constructed as before, with the exception that, instead of discarding  $\delta\%$  of the samples, these samples are replaced by random outlier values uniformly distributed in the range  $\pm[10\sim 20]$ . The values of  $\delta$  considered in this example are 5%, 10%, 15% and 20%. A sample of the constructed noisy signal for the case  $\delta = 15\%$  is shown in the upper panel in Figure 7(c). The smoothed signals obtained by the proposed RBCSA and the RSmooth algorithms are shown in the lower panel in

Figure 7(c). To quantitatively assess the performance of the two algorithms, and for each value of  $\delta$ , the  $SNR_{imp}$  associated with each algorithm is calculated as the average over 200 different runs, where in each run new AWGN is added at 10 dB and new  $\delta\%$  outlier samples are generated, where the indices of the outlier samples are randomly selected. The average  $SNR_{imp}$  associated with the two algorithms are shown in Figure 7(d). As shown in this figure, and for all the considered values of  $\delta$ , the proposed RBCSA outperforms the RSmooth algorithm.

### 3.5 | Example 5: smoothing data of the daily infection by coronavirus disease 2019 (CoViD-19)

The proposed BCSA algorithm is now tested using the real data of the daily infection by the CoViD-19. Two different datasets are considered in this example. The datasets represent the daily infection by the coronavirus in the United States and the United Kingdom in the interval from the first of March 2020 to the end of June 2020. The data is publicly published by the European Centre for Disease Prevention and Control at [33]. The recorded data are presented by the *blue* waveforms in Figure 8, while the smoothed data are presented by the *red* waveforms. As can be observed from Figure 8, the two datasets have been accurately smoothed by the proposed BCSA algorithm. It can be also observed from the smoothed curves that the behaviour of daily infections in the two countries was almost similar (however with different scales) in the first 70 days. However, in the following 50 days, it is evident that the rate of infection in the United Kingdom has decreased significantly, while for the United States, the rate of infection has increased significantly in the last 20 days of the considered interval.



**FIGURE 8** Smoothing the data of daily infection by the CoViD-19 in the United States (a) and the United Kingdom (b) for the interval from the first of March 2020 to the end of June 2020



## 4 | CONCLUSION

A new smoothing algorithm was developed to overcome the limitations associated with the existing PLS smoothing algorithms based on Equation (2). A new objective function was utilised in the developed algorithm. The proposed objective function depends on four parameters: two regularisation parameters and the orders of two difference matrices. By properly selecting the values of the four parameters, the proposed algorithm can act either as an LPF or a BPF. To minimise the distortion introduced to the estimated signal, the spectral characteristics of the imbedded smooth signal were utilised to derive closed-form expressions for the optimum values of these four parameters. As a result, the distortion introduced by the proposed algorithm to the smoothed signal is always smaller than the distortion introduced by the existing PLS smoothing algorithms based on Equation (2), which act solely as LPFs. In addition, and in contrast to the GCV method used with the existing PLS algorithms, no special optimisation tools are required to estimate the optimum value of the regularisation parameters in the proposed algorithm. The derived algorithm was further developed to handle the case of having missing and/or outlier samples in the measured data. An adaptive procedure was suggested to estimate the indices of the outlier samples. Therefore, the proposed algorithms are completely data driven, and they do not require any user intervention. In addition, the proposed algorithms were also shown to produce significantly improved results compared to some existing smoothing techniques.

## REFERENCES

- Garcia, D.: Robust smoothing of gridded data in one and higher dimensions with missing values. *Comput. Stat. Data Anal.* 54, 1167–1178 (2010)
- Barclay, V.J., Bonner, R.F.: Application of wavelet transforms to experimental spectra: smoothing, denoising, and data set compression. *Anal. Chem.* 69, 78–90 (1997)
- Orfanidis: Introduction to Signal Processing (1995–2009). <http://www.ece.rutgers.edu/~orfanidis/intro2sp> Accessed 17 June 2020.
- Savitzky, A., Golay, M.J.E.: Smoothing and differentiation of data by simplified least squares procedures. *Anal. Chem.* 36, 1627–1639 (1964)
- Huang, H., Hu, S., Sun, Y.: A discrete curvature estimation based low-distortion adaptive Savitzky–Golay filter for ECG denoising. *Sensors* 19, 1617 (2019)
- Eilers, P.H.C.: A perfect smoother. *Anal. Chem.* 75, 3631–3636 (2003)
- Golyandina, N., Zhigljavsky, A.: Singular Spectrum Analysis for Time Series. SpringerBriefs in Statistics (2013). <https://doi.org/10.1007/978-3-642-34913-3>
- Hassani, H., Mahmoudvand, R.: Singular Spectrum Analysis Using R. Palgrave Advanced Texts in Econometrics (2018). <https://doi.org/10.1057/978-1-137-40950-8>
- Harmouche, J., et al.: The sliding singular spectrum analysis: a data-driven nonstationary signal decomposition tool. *IEEE Trans. Signal Process.* 66, 251–263 (2018)
- Mourad, N.: ECG denoising algorithm based on group sparsity and singular spectrum analysis. *Biomed. Signal Process. Control.* 50, 62–71 (2019)
- Huang, N.E., et al.: The empirical mode decomposition and Hilbert spectrum for nonlinear and nonstationary time series analysis. *Proc. R. Soc. London Ser. A.* 454, 903–995 (1998)
- Wu, Z., Huang, N.E.: Ensemble empirical mode decomposition: a noise-assisted data analysis method. *Adv. Adapt. Data Anal.* 1, 1–41 (2009)
- Yeh, J.R., Shieh, J.S., Huang, N.E.: Complementary ensemble empirical mode decomposition: a novel noise enhanced data analysis method. *Adv. Adapt. Data Anal.* 2, 135–156 (2010)
- Colominas, M.A., Schlotthauer, G., Torres, M.E.: Improved complete ensemble EMD: a suitable tool for biomedical signal processing. *Biomed. Signal Process. Control.* 14, 19–29 (2014)
- Zuo, L., et al.: Noise suppression method of microseismic signal based on complementary ensemble empirical mode decomposition and wavelet packet threshold. *IEEE Access.* 7, 176504–176513 (2019)
- Moradi, N., Dousty, M., Sotero, R.C.: Spatiotemporal empirical mode decomposition of resting-state fMRI signals: application to global signal regression. *Front. Neurosci.* 13, 1–14 (2019)
- Xi, X., et al.: Denoising of surface electromyogram based on complementary ensemble empirical mode decomposition and improved interval thresholding. *Rev. Sci. Instrum.* 90, 1–9 (2019)
- Flandrin, P., Rilling, G., Gonçalves, P.: Empirical mode decomposition as a filter bank. *IEEE Signal Process. Lett.* 11, 112–114 (2004)
- Whittaker, E.T.: On a new method of graduation. *Proc. Edinburgh Math. Soc.* 41, 63–75 (1923)
- Wahba, G.: Spline Models for Observational Data. Society for Industrial Mathematics, Philadelphia (1990). <https://epubs.siam.org/doi/book/10.1137/1.9781611970128?mobileUi=0>
- Kitahara, D., Condat, L., Hirabayashi, A.: One-dimensional edge-preserving spline smoothing for estimation of piecewise smooth functions, pp. 5611–5615. *IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP, Brighton, (2019)*. <https://doi.org/10.1109/ICASSP2019.8683260>
- Stickel, J.J.: Data smoothing and numerical differentiation by a regularization method. *Comput. Chem. Eng.* 34, 467–475 (2010)
- Weinert, H.L.: Efficient computation for Whittaker–Henderson smoothing. *Comput. Stat. Data Anal.* 52, 959–974 (2007)
- Zhang, F., et al.: Baseline correction for infrared spectra using adaptive smoothness parameter penalized least squares method. *Spectrosc. Lett.* 1–12 (2020)
- Garcia, D.: A fast all-in-one method for automated post-processing of PIV data. *Exp. Fluids.* 50, 1247–1259 (2011)
- Mourad, N.: New two-stage approach to ECG denoising. *IET Signal Process.* 13, 596–605 (2019)
- Buckley, M.J.: Fast computation of a discretized thin-plate smoothing spline for image data. *Biometrika.* 81, 247–258 (1994)
- Craven, P., Wahba, G.: Smoothing noisy data with spline functions. Estimating the correct degree of smoothing by the method of generalized crossvalidation. *Numer. Math.* 31, 377–403 (1978)
- Jain, A.K.: Fundamentals of Digital Image Processing. Prentice-Hall, Englewood Cliffs (1989). <https://www.amazon.com/Fundamentals-Digital-Image-Processing-Anil/dp/0133361659>
- Snedecor, G.W., Cochran W.G.: Statistical Methods, 8th ed. Iowa State University Press (1989). <https://www.amazon.com/Statistical-Methods-George-W-Snedecor/dp/0813815614>
- Bayer, F.M., Kozakevicius, A.J., Cintra, R.J.: An iterative wavelet threshold for signal denoising. *Signal Process.* 162, 10–20 (2019)
- mathworks.com [homepage on the internet]. MATLAB Central File Exchange. Matthew Taliaferro (2021) [Cited 2021 Feb 01] <https://www.mathworks.com/matlabcentral/fileexchange/63168-csapsgcv-x-y-p-xx-w> Accessed 1 Feb 2021
- data.europa.eu [homepage on the internet]. EU Open Data Portal [Cited 2021 Feb 01] <https://data.europa.eu/euodp/data/dataset/covid-19-coronavirus-data> Accessed 1 Feb 2021

**How to cite this article:** Mourad N. Robust smoothing of one-dimensional data with missing and/or outlier values. *IET Signal Process.* 2021;15:323–336. <https://doi.org/10.1049/sil2.12033>