Efficient Denoising of Piecewise-Smooth Signals with Forward-Backward FIR Smoothers

OSBALDO VITE-CHÁVEZ, ROBERTO OLIVERA-REYNA
 Autonomous University of Zacatecas, Jalpa
 MEXICO

JORGE MUÑOZ-MINJARES, YURIY S. SHMALIY, OSCAR IBARRA-MANZANO
 Guanajuato University, Department of Electronics
 Salamanca, 36855
 MEXICO
 shmaliy@ugto.mx

LUIS MORALES-MENDOZA
 University of Veracruz, Department of Electronics
 Poza-Rica, 93390
 MEXICO

Abstract: A forward-backward (FB) unbiased finite impulse response (UFIR) smoothing algorithm is addressed to provide efficient denoising of piecewise-smooth signals with known or identifiable edge positions. Owing to the time-variant horizon, the algorithm unites advantages of linear structures with robustness of nonlinear ones. The FB UFIR smoother proposed has been examined in Gaussian and heavy-tailed noise environments and compared to the nonlinear myriad filter fitting the Cauchy statistics. Based upon, we show that the smoother is able to preserve edges without jitter and provide denoising with sufficient robustness against outliers.

Key–Words: Forward-backward smoothing, FIR structure, piecewise signal, identifiable edge

1 Introduction

Methods for piecewise signal denoising are, in general, non-real-time. In off-line processing, forward-backward (FB) filters and smoothers are also often used suggesting two rules of how to apply and combine forward and backward estimates and thus implement and connect both the causal and noncausal structures. The first way implies conducting forward and backward processing simultaneously and then combine the results at one point [1]. The method was efficiently exploited to denoise seismic traces and images [2], spectral estimation and the amplitude and phase filtering [3], etc.

Another option suggests smoothing a signal forward and then repeat it backward. Denoising is commonly more efficient here, although the procedure takes extra time. Applications can be found in state-space estimation [4] and image processing [5]. However, the approach seems to be most useful for a class of problems when positions of the breakpoints are known a priori, as in digital message transmitting/receiving with synchronized timescales, or well detectable a posteriori employing time-derivative of the origin, robust filtering, etc. The algorithm can be designed based on different kinds of smoothers, among which the UFIR one [6] demonstrates several attractive properties: it ignores noise statistics and has the bounded input/bounded output stability. Below, we examine a time-varying FB UFIR smoothing algorithm for denoising piecewise-smooth signals with known or identifiable breakpoints in Gaussian and heavy-tailed noise environments.

2 Main idea

Suppose that a discrete-time piecewise-smooth signal \( x_n \) (dashed in Fig. 1) is measured in the presence of noise as \( y_n \) and the breakpoints \( k_q, q = 0, 1, \ldots \) are known. Then:

1) Smooth measurement forward from \( k_0 + 1 \) at zero to \( n = k_1 \). If noise is white Gaussian with the variance \( \sigma^2 \), then the output noise variance at \( k_1 \) will be \( \sigma^2_{\text{forward}} = \frac{\sigma^2}{k+1} \). Next, smooth the result backward, from \( k_1 \) to \( k_0 + 1 \). The backward estimate variance can be expected to be \( \sigma^2_{\text{backward}} \cong \sigma^2_{\text{forward}} \) at each point.

2) Pass across the edge, from \( k_1 \) to \( k_1 + 1 \), with the ramp UFIR filter on a window of \( N = 2 \) points.

3) Repeat this procedure for all smooth parts, from \( k_q + 1 \) to \( k_{q+1} \), and edges, from \( k_q \) to \( k_q + 1 \).
where \( l = K - 1 \) and \( a_{jl}(N, p) \) is specified by
\[
a_{jl}(N, p) = (-1)^{j} M_{(j+1)}(N, p) / |D(N, p)| \tag{4}
\]
via a short \( l \times l \) matrix
\[
D = \begin{bmatrix}
  d_0 & d_1 & \cdots & d_l \\
  d_1 & d_2 & \cdots & d_{l+1} \\
  \vdots & \vdots & \ddots & \vdots \\
  d_l & d_{l+1} & \cdots & d_{2l}
\end{bmatrix}, \tag{5}
\]
which \( \nu \)th, \( \nu \in [0, 2l] \), component \( d_{\nu}(N, p) \) can be developed as
\[
d_{\nu}(N, p) = \sum_{i=p}^{N-1} i^\nu, \quad \nu = 0, 1, \ldots, 2l.
\]
Here, \( |D(N, p)| \) and \( M_{(j+1)}(N, p) \) are the determinant and minor of \( D(N, p) \). The low-degree, \( l \leq 2 \), \( p \)-lag responses \( \tilde{h}_{ln}(N, p) \) were found in the form of (3) and postponed to Appendices A. Note that \( \tilde{h}_{ln}(N, p) \) has been discussed in different forms in [6, 8, 9].

### 3.2 Batch form

Utilizing \( \tilde{h}_{ln} \triangleq \tilde{h}_{ln}(N, p) \), the UFIR smoothing estimate can be provided by the discrete convolution as
\[
\tilde{x}_{n+p|n} = \sum_{i=0}^{N-1} \tilde{h}_{li} y_{n-i} - \nu, \tag{6}
\]
One can also compute (6) as \( \tilde{x}_{n+p|n} = \tilde{W}_l^T y_{n,m} = y_{n,m}^T \tilde{W}_l \), where \( m = n - N + 1, m \geq 0 \), and we have assigned \( \tilde{W}_l = [ \tilde{h}_{l0} \quad \tilde{h}_{l1} \quad \cdots \quad \tilde{h}_{l(N-1)} ]^T \) and \( y_{n,m} = [ y_n y_{n-1} \cdots y_m ]^T \). If (6) imposes the computational problem with \( N \gg 1 \), then the iterative Kalman-like algorithm [10] can be used.

### 4 FB UFIR denoising algorithms

A variety of FB UFIR denoising algorithms can be designed based on different approached. In what follows, we focus out attention on two solutions. We first design the FB UFIR smoothing algorithm aimed at maximizing the denoising effect. We then simplify it to be filtering for piecewise rectangular signals.

#### 4.1 Smoothing algorithm

An operation principle of the FB UFIR smoothing algorithm is illustrated in Fig. 2 under the supposition that the breakpoints \( k_q \) of jump discontinuities are known. Let us assume that the prior edge has occurred at \( k_0 = -1 \) and finished at zero or at \( k_0 + 1 \). The
Figure 2: Forward-backward UFIR smoothing of piecewise signals, provided location of edges: (a) algorithm, (b) time-variant averaging interval \( N(n) \), (c) time-variant degree \( l(n) \) of the impulse response \( h_{1n}(N,p) \), and (d) time-variant lag \( p(n) \).

The algorithm starts forward (FW) at \( k_0 + 1 \) (Fig. 2a) employing the \( l \)-degree (Fig. 2b) full-horizon smoother, \( N = n + 1 \) (Fig. 2c), with the time-variant lag \( p < 0 \) (Fig. 2d). The degree \( l \) is chosen such that smoother fits the signal smooth part (Fig. 2b). Because an increase in \( l \) leads to the increase in the output noise variance [6], the low-degree responses (Appendix A) can be used to cover a number of practical applications. The data thus can be smoothed with

\[
\hat{x}_{n+p|n} = \sum_{i=0}^{n} \tilde{h}_{li}(n+1,p) y_{n-i}, \quad l < n \leq k_1, \tag{7}
\]

and one must set \( \hat{x}_{n|n} = y_n \) if \( k_0 + 1 \leq n \leq l \), because \( \tilde{h}_{li}(n+1,p) \) does not exist otherwise.

To minimize errors [6], the smoother lag \( p \) is allowed to be \( p = -\left\lfloor \frac{n-k_0}{2} \right\rfloor \), where \( \lfloor x \rfloor \) means a maximum integer smaller than or equal to \( x \). That guarantees a minimum variance related to the center of the averaging interval. At a fixed \( n = k_1 \), the lag \( p \) is reduced, step by step, from \( p = -\left\lfloor \frac{k_1-k_0}{2} \right\rfloor \) to zero in order to produce estimates from \( k_1 - \left\lfloor \frac{k_1-k_0}{2} \right\rfloor + 1 \) to \( k_1 \) (Fig. 2d). So, the FW smoother output is computed starting with \( k_0 + 1 \) and finishing at \( k_1 \) as

\[
\hat{x}_{n+f(n)|n}^{FW} = \sum_{i=0}^{n-k_0-1} \tilde{h}_{li}(q(n), f(n)) y_{n-i}, \quad n > l, \tag{8}
\]

where \( q(n) = n - k_0 \) and \( f(n) = -\left\lfloor \frac{n-k_0}{2} \right\rfloor \). The output \( \hat{x}_{n|k_1}^{FW} \) provided in such way is further inverted in time as

\[
\hat{x}_{n|k_1}^{BW} = \hat{x}_{k_0+k_1+1-n|k_1}^{FW}, \quad k_0 + 1 \leq n \leq k_1. \tag{14}
\]

The signal edge following after \( k_1 \) is viewed as a linear function, from \( k_1 \) to \( k_1 + 1 \). Thus, the UFIR filter passes across the edge with the ramp, \( l = 1 \), impulse response \( h_{1n}(N,p) \) having \( N = 2 \) and \( p = 0 \) as in Fig. 2. By (A.1), the response becomes \( \tilde{h}_{1n}(2,0) = 1 - n \), we infer that \( \hat{x}_{k_1+1|k_1+1} = y_{k_1+1} \), and this filtering procedure can hence be avoided.

Denoising of the subsequent smooth parts can be accomplished similarly. Substitute in (8)–(14) \( k_0 \) with \( k_q \) and \( k_1 \) with \( k_{q+1} \). Accordingly, the FB UFIR estimate of the piecewise signal will be specialized with

\[
\hat{x}_{n|k_q+1}^{FB} = \hat{x}_{k_q+k_{q+1}+1-n|k_{q+1}}^{BW}, \quad k_q + 1 \leq n \leq k_{q+1} + 1. \tag{15}
\]

4.2 Filtering algorithm

If a signal is composed of rectangular pulses, the FB UFIR smoothing algorithm can be simplified to the filtering one, provided \( p = 0 \),

\[
\hat{x}_{n|n}^{FW} = \begin{cases} \displaystyle\sum_{i=0}^{q(n)-1} \tilde{h}_{li}(q(n), 0)y_{n-i}, & n > l, \\ y_n, & n \leq l, \end{cases} \tag{16}
\]
For an arbitrary number \( q \) of breakpoints, the FB UFIR filter output can be found similarly to (15) by

\[
x_{n|k+1}^{\text{FB}} = x_{k+q+1,n|k+q+1}, \quad k + 1 \leq n \leq k_q + 1.
\]

Errors in FB denoising can be limited in the three-sigma sense [11] with the error bound (EB) \( \beta_l(N) \) as

\[
\beta_l(N) = 3\sigma g_l^{1/2}(N, 0),
\]

where the noise power gain (NPG) \( g_l(N, 0) \) is determined by \( g_l(N, 0) = \sum_{n=0}^{N-1} h_{0n}(N, 0) = a_{0l}(N, 0) \).

Figure 3: FB UFIR denoising of a set of digital numbers contaminated by white Gaussian noise: (a) signal \( x_n \), measurement \( y_n \), and estimate \( \hat{x}_n \), (b) error \( \epsilon_n \), and (c) error probability for \( |\epsilon_n| > 0.5 \).

5 Applications

Below, we give several applications for FB UFIR denoising of piecewise-smooth signals in Gaussian and Cauchy noise environments.

5.1 Denoising in Gaussian noise

In the first experiment, we suppose that a set of transmitted digital numbers \( x_n \) appears at the receiver detector with AWGN as \( y_n \) (Fig. 3a). Denoising is required such that the absolute error does not exceed 0.5. Provided the GPS-based locking of timescales with a negligible error of \( < 100 \) ns, time positions of all numbers are supposed to be known exactly. Because the message is stepwise, the FB UFIR filtering algorithm (17)–(21) is applied with \( h_{0n}(N, p) = 1/N \) to produce \( \hat{x}_n \) (Fig. 3a).

The difference \( \epsilon_n \) between \( x_n \) and \( \hat{x}_n \) is shown in Fig. 3b. It can be seen that \( \epsilon_n \) falls well within a gap between 0.5 and –0.5. It can also be shown that the FB UFIR filtering algorithm provides a reliable denoising with the error probability of about 10\(^{-4} \) for \( \gamma = S_{\text{max}}^2/2\sigma^2 = 0 \text{ dB} \) . Note that the output appears here with a delay on a half bit-length.

5.2 Denoising in Cauchy noise

In industrial applications, noise often exists with outliers or becomes heavy tailed [7]. To study the effect of noise tails on the FB UFIR estimate, we consider a stepwise message \( x_n \) (Fig. 4a) contaminated with the heavy-tailed noise having the Cauchy probability density function (pdf) \( p(x) = \frac{1}{\pi \alpha^2 + (x - \beta)^2} \), where \( \alpha \) is the scale parameter also known as the noise dispersion responsible for the tail length and \( \beta \) is the location akin to the mean value in the normal law. Because the noise tails can be extremely large, we saturate them with 7 and –2, imitating an electronic channel, and run the FB UFIR filtering algorithm.

Linear methods are not optimal for heavy-tailed noise sources. Therefore, as a reference, we also run the maximum likelihood myriad filter \( \hat{\beta} = \arg \min_{\beta} N_{m-1} \prod_{i=0}^{N_m-1} \left[ \alpha^2 + (y_{n-i} - \beta)^2 \right] \) designed in [12] under the Cauchy noise statistics, where \( n \geq N_m - 1 \), \( N_m \) is even and chosen such that the MSE is minimal, and \( y_n \) is the measurement. The myriad algorithm is smoothing. therefore \( \hat{\beta} \) is related to the center \( n - N_m - 1/2 \) of the observation interval.

Fig. 4a sketches a typical output \( \hat{x}_n \) of the FB UFIR filter for a moderate dispersion \( \alpha = 0.05 \), for which we have found \( N_m = 61 \). In turn, Fig. 4b and Fig. 4c give us typical errors. Inherently, the myriad filter provides better denoising in Cauchy noise, but preserves edges with jitter keeping EP close to unity. Although this performance can be improved in weighted myriad smoothing, we were not able to
Figure 4: Stepwise message denoising in heavy-tailed Cauchy noise, $\alpha = 0.05$: (a) signal $x_n$, measurement $y_n$, and the FB UFIR filter output $\tilde{x}_n$, (b) FB UFIR denoising error $\varepsilon_n$, and (c) Myriad filter denoising error.

Figure 5: Simulated signal power at the receiver antenna: (a) piecewise signal power and (b) identified edges.

6b and Fig. 6d for the FB UFIR smoother and in Fig. 6f for the myriad filter. Inherently, the myriad filter produces here lower errors, except for jitter. But the linear FB UFIR smoother also demonstrates a sufficiently good robustness against noise tails, similarly to Fig. 4b.

6 Concluding remarks

Although the FB UFIR smoother examined in this paper is not universal, it has certain applications, especially when denoising of digital messages with known edge positions is required in harsh environment. The algorithm can also be useful when edges are well detectable, as in contrast images, or their positions are allowed to be adjusted in postprocessing. If so, then the result comes up without jitter.

5.3 Denoising of a piecewise signal power

We finally provide denoising of a piecewise variations in the electromagnetic wave power at the antenna. An example of such variations provided each minute during 10 hours in the presence of white Gaussian noise is shown in Fig. 5a. Edge positions were identified here by the time derivative as shown in Fig. 5b. The FB UFIR smoother was employed with $l = 1$ and $l = 2$ to produce errors shown in Fig. 6a and Fig. 6c, respectively. In turn, Fig. 6e sketches the errors produced by the myriad filter with $N_m = 61$. It follows that in the Gaussian noise environment the linear structure work better than the robust one. In the Cauchy noise environment, $\alpha = 0.05$, the results are shown in Fig.

TBD

A Low-degree FIRs for $p$-lag unbiased smoothers

$$\tilde{h}_{1n}(N, p) = \frac{2(2N - 1) + 6p}{N(N + 1)} - \frac{6(N - 1 + 2p)}{N(N^2 - 1)}n,$$  \hspace{1cm} (A.1)

$$\tilde{h}_{2n}(N, p) = \tilde{a}_{i02} + \tilde{a}_{12}n + \tilde{a}_{22}n^2,$$  \hspace{1cm} (A.2)

where

$$\tilde{a}_{i02} = 3 \frac{3N(N - 1) + 2 + 2p[3(2N - 1) + 5p]}{N(N + 1)(N + 2)},$$  \hspace{1cm} (A.3)
Figure 6: Denoising errors for a signal shown in Fig. 6. In white Gaussian noise: (a) FB UFIR smoother, \( l = 1 \), (c) FB UFIR smoother, \( l = 2 \), and (e) Myriad filter, \( N_m = 61 \). In Cauchy noise, \( \alpha = 0.05 \): (b) FB UFIR smoother, \( l = 1 \), (d) FB UFIR smoother, \( l = 2 \), and (f) Myriad filter, \( N_m = 33 \).

\[
\tilde{a}_{12} = -6 \frac{3(N-1)(N-2) + 2p(8N - 11)}{(2N-1) + 30p^2(N-1)} N(N^4 - 5N^2 + 4), \\
\tilde{a}_{22} = 30 \frac{N^2 - 3N + 2 + 6p(N-1) + 6p^2}{N(N^4 - 5N^2 + 4)},
\]

(A.4)

(A.5)

References:


