Recursive Weighted Median Filters Admitting Negative Weights and Their Optimization

Gonzalo R. Arce, *Fellow, IEEE,* and José L. Paredes

***Abstract*—A *recursive weighted median* (RWM) filter structure admitting negative weights is introduced. Much like the sample median is analogous to the sample mean, the proposed class of RWM filters is analogous to the class of infinite impulse response (IIR) linear filters. RWM filters provide advantages over linear IIR filters, offering near perfect "stopband" characteristics and robustness against noise. Unlike linear IIR filters, RWM filters are always stable under the bounded-input bounded-output criterion, regardless of the values taken by the feedback filter weights. RWM filters also offer a number of advantages over their nonrecursive counterparts, including a significant reduction in computational complexity, increased robustness to noise, and the ability to model "resonant" or vibratory behavior. A novel "recursive decoupling" adaptive optimization algorithm for the design of this class of recursive WM filters is also introduced. Several properties of RWM filters are presented, and a number of simulations are included to illustrate the advantages of RWM filters over their nonrecursive counterparts and IIR linear filters.**

***Index Terms*—Adaptive filter, median filters, nonlinear signal processing, recursive median filter, robustness.**

I. Introduction

WEIGHTED median (WM) smoothers[1] have received considerable attention in signal processing research over the last two decades [1]–[3]. It is often stated that there are many analogies between weighted median smoothers and linear FIR filters. Recently, however, it was shown that WM smoothers are highly constrained, having significantly less powerful characteristics than linear FIR filters. In fact, WM smoothers are equivalent to normalized weighted mean filters admitting only positive weights—a severely constrained subset of linear FIR filters. Admitting only positive weights, weighted median smoothers are, in essence, restricted to "lowpass" type filtering characteristics. In a large number of engineering applications where "bandpass" or "highpass" type filtering characteristics are required, weighted median smoothers are inadequate.

To overcome these limitations, a generalized weighted median filtering structure admitting positive and negative weights has been recently introduced [4]. The generalization follows naturally, is surprisingly simple, and leads to a significantly richer class of weighted median filters. In fact, much like the sample median can be thought of as being analogous to the sample mean, the generalized WM filter structure is analogous to linear FIR filters. Weighted median filters, admitting negative weights, have been shown to be capable of effectively addressing a number of fundamental problems in signal processing that could not adequately be addressed by prior weighted median smoother structures [4].

Having the framework for weighted median filters, it is natural to extend it to other more general signal processing structures. This paper focuses on precisely this goal. In particular, we introduce a class of recursive weighted median filters, admitting real-valued weights. These filters are analogous to the class of infinite impulse response (IIR) linear filters. Recursive filter structures are particularly important because they can be used to model "resonances" that appear in many natural phenomena such as in speech. In fact, in the linear filtering framework, a large number of systems can be better characterized/modeled by a pole-zero transfer function than by a transfer function containing only zeros. In addition, IIR linear filters often lead to reduced computational complexity reduction. Much like IIR linear filters provide these advantages over linear FIR filters, recursive WM filters also exhibit superior characteristics than nonrecursive WM filters. For instance, an infinitely iterated use of a weighted median filter can often be synthesized by a single pass of a properly designed recursive weighted median filter [2]. Indeed, recursive WM filters can synthesize nonrecursive WM filters of much larger window sizes. In terms of noise attenuation, recursive median smoothers have far superior characteristics than their nonrecursive counterparts [5], [6]. A disadvantage of recursive median smoothers is that they can exhibit some blurring artifacts on the output; nonetheless, these artifacts have been observed when no optimization of the weights is performed. Having the optimization tool introduced in this paper, we believe that many of those artifacts common to recursive median smoothers can be minimized.

It will also be shown in this paper that RWM filters can provide advantages over linear IIR filters. Notably, "bandpass" and "highpass" RWM filters exhibit perfect "stopband" characteristics not attainable with linear IIR filters. Moreover, unlike their

Manuscript received April 27, 1999; revised August 13, 1999. This work was supported in part by the National Science Foundation under Grants MIP-9530923 and CDA-9703088 and in part by the ATIRP Consortium sponsored by the U.S. Army Research Laboratory under the Federated Laboratory Program, Cooperative Agreement DAAL01-96-2-0002. The associate editor coordinating the review of this paper and approving it for publication was Prof. Chin-Liang Wang.

G. R. Arce is with the Department of Electrical and Computer Engineering, University of Delaware, Newark, DE 19716 USA (e-mail: arce@ee.udel.edu).

J. L. Paredes is with the Department of Electrical and Computer Engineering, University of Delaware, Newark, DE 19716 USA. He is also with the Electrical Engineering Department, University of Los Andes, Mérida, Venezuela (e-mail: paredesj@ee.udel.edu).

Publisher Item Identifier S 1053-587X(00)01540-3.

[1] Weighted median smoothers admitting only positive weights have traditionally been referred to in the literature as WM filters, although, as detailed in this paper, they are limited to lowpass operators. In this paper, we denote these structures as "smoothers" to differentiate them from the more powerful WM filter structures that admit negative weights that can synthesize general frequency selecting filtering.

IIR filter counterparts, RWM filters are always stable under the bounded-input bounded-output criterion, regardless of the values taken by the feedback filter weights. In the presence of noise, the advantages of RWM filters over IIR filters are even more overwhelming, offering robustness to noise levels that are unacceptable with traditional IIR filters. In fact, the performance of IIR linear filters is strongly degraded if the input signal is contaminated with impulsive noise.

In practice, the (real-valued) filter coefficients of the proposed RWM filter structures must be determined in some fashion. This paper presents the first optimization method for the design of recursive WM filters and smoothers. A novel "recursive decoupling" adaptive optimization algorithm of RWM filter weights is developed under the mean absolute error (MAE) criterion. In this framework, the previous outputs used to compute the recursive WM filter output are replaced by desired previous outputs. Thus, the recursive WM filter becomes a two-input, single-output filter that depends on the input samples and on delayed samples of the desired response. This structure avoids the feedback inherent in the recursive operation and, therefore, leads to a much simpler derivation of the gradient in the steepest descent algorithm used to update the filter coefficients. The adaptive RWM filter algorithm, which is referred to as the *fast least mean absolute (LMA)* recursive WM algorithm, has an update complexity comparable with that of the LMS algorithm.

The organization of the paper is as follows. In Section II, the new recursive weighted median filtering structure admitting real-valued coefficients is introduced. In Section III, the threshold decomposition property is adapted to this problem and used, subsequently, to derive the adaptive algorithms for the optimization of the recursive WM filter coefficients. In Section IV, several examples of recursive WM filters are shown, and their performance is compared with that of nonrecursive WM filters and to linear IIR filters. Finally, some conclusions are drawn in Section V.

II. Recursive WM Filters Admitting Real-Valued Weights

In order to define the class of RWM filters, it is best to first recast the similarities between linear FIR filters and weighted median filters. Given an observation set $X_1, X_2, \ldots, X_N$, the sample mean $\bar{\beta} = \text{MEAN}(X_1, X_2, \ldots, X_N)$ can be generalized to linear FIR filters as

$$\beta = \text{MEAN}(W_1 \cdot X_1, W_2 \cdot X_2, \ldots, W_N \cdot X_N) \qquad (1)$$

where $W_i \in R$. It will be seen shortly that it is useful to rewrite (1) as

$$\begin{aligned}\bar{\beta} = \text{MEAN}(&|W_1| \cdot \text{sgn}(W_1)X_1\\ &|W_2| \cdot \text{sgn}(W_2)X_2, \ldots, |W_N| \cdot \text{sgn}(W_N)X_N)\end{aligned} \qquad (2)$$

where sgn denotes the sign function defined as

$$\text{sgn}(X) = \begin{cases} +1, & \text{if } X \geq 0 \\ -1, & \text{if } X < 0. \end{cases} \qquad (3)$$

Note, in (2), that the sign of the weight affects the corresponding input sample, and the weighting is constrained to be non-negative.

It was shown in [4] that the sample median $\hat{\beta} = \text{MEDIAN}(X_1, X_2, \ldots, X_N)$, which plays an analogous role to the sample mean in location estimation, can be extended to the general weighted median filter structure admitting positive and negative weights as

$$\begin{aligned}\tilde{\beta} = \text{MEDIAN}(&|W_1| \diamond \text{sgn}(W_1)X_1\\ &|W_2| \diamond \text{sgn}(W_2)X_2, \ldots, |W_N| \diamond \text{sgn}(W_N)X_N)\end{aligned} \qquad (4)$$

with $W_i \in R$ for $i = 1, 2, \ldots, N$, and where $\diamond$ is the replication operator defined as $W_i \diamond X_i = \overbrace{X_i, X_i, \ldots, X_i}^{W_i \text{ times}}$. Again, the weight signs are uncoupled from the weight magnitude values and are merged with the observation samples. The weight magnitudes play the equivalent role of positive weights in the framework of weighted median smoothers [3]. Although the weights in (4) may seem restricted to integer values, a more general interpretation of the $\diamond$ operator will be presented shortly.

The filters in (2) and (4) can be thought of as nonrecursive filter duals. This duality is next extended to their recursive forms. The general structure of linear IIR filters is defined by the difference equation

$$Y(n) = \sum_{\ell=1}^{N} A_\ell Y(n-\ell) + \sum_{k=-M_1}^{M_2} B_k X(n-k) \qquad (5)$$

where the output is formed not only from the input but also from previously computed outputs. The filter weights consist of two sets: the feedback coefficients $\{A_\ell\}$ and the feedforward coefficients $\{B_k\}$. In all, $N + M_1 + M_2 + 1$ coefficients are needed to define the recursive difference equation in (5). Often, M_1 is set to zero for a causal IIR filter implementation.

The generalization of (5) to an RWM filter structure is straightforward. Following a similar approach to that introduced in [4], the summation operation is replaced with the *median* operation, and the *multiplication* weighting is replaced by weighting through *signed replication*:

$$\begin{aligned}Y(n) = \text{MEDIAN}\Big(&|A_\ell| \diamond \text{sgn}(A_\ell)Y(n-\ell)|_{\ell=1}^{N}\\ &|B_k| \diamond \text{sgn}(B_k)X(n-k)|_{k=-M_1}^{M_2}\Big).\end{aligned} \qquad (6)$$

A noncausal implementation is assumed from now on, where $M_2 = 0$ and $M_1 = M$, leading to

$$\begin{aligned}Y(n) = \text{MEDIAN}(&|A_N| \diamond \text{sgn}(A_N)Y(n-N), \ldots\\ &|A_1| \diamond \text{sgn}(A_1)Y(n-1), |B_0| \diamond \text{sgn}(B_0)X(n), \ldots\\ &|B_M| \diamond \text{sgn}(B_M)X(n+M)).\end{aligned} \qquad (7)$$

The recursive WM filter operation is schematically described in Fig. 1.

Note that if the weights A_ℓ and B_k are constrained to be positive, (7) reduces to the recursive WM smoother previously

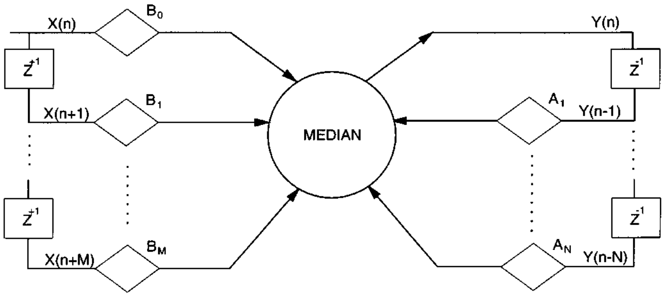

Fig. 1. Structure of a recursive WM filter.

studied in [7] and [8]. The recursive WM filter output for non-integer weights can be determined as follows:

1) Calculate the threshold $T_0 = (1/2)(\sum_{\ell=1}^{N}|A_\ell| + \sum_{k=0}^{M}|B_k|)$.
2) Jointly sort the "signed" past output samples sgn$(A_\ell)Y(n-\ell)$ and the "signed" input observations sgn$(B_k)X(n+k)$.
3) Sum the magnitudes of the weights corresponding to the sorted "signed" samples beginning with the maximum and continuing down in order.
4) If $2T_0$ is an even number, the output is the average between the signed sample whose weight magnitude causes the sum to become $\geq T_0$ and the next smaller signed sample; otherwise, the output is the signed sample whose weight magnitude causes the sum to become $\geq T_0$.

The following example illustrates this procedure. Consider the window size 6 RWM filter defined by the real-valued weights $\langle\langle A_2, A_1, B_0, B_1, B_2, B_3\rangle\rangle = \langle\langle 0.2, 0.4, 0.6, -0.4, 0.2, 0.2\rangle\rangle$. The output for this filter operating on the observation set $[Y(n-2), Y(n-1), X(n), X(n+1), X(n+2), X(n+3)]^T = [-2, 2, -1, 3, 6, 8]^T$ is found as follows. Summing the absolute weights gives the threshold $T_0 = (1/2)(A_1 + A_2 + B_0 + B_1 + B_2) = 1$. The "signed" set of samples spanned by the filter's window, the sorted set, their corresponding weight, and the partial sum of weights (from each ordered sample to the maximum) are

sample set in window	-2,	2,	-1,	3,	6,	8
corresponding weights	0.2,	0.4,	0.6,	-0.4,	0.2,	0.2

sorted signed samples	-3,	-2,	-1,	2,	6,	8
absolute weights	0.4,	0.2,	0.6,	0.4,	0.2	0.2
partial weight sums	2.0,	1.6,	$\underline{1.4}$,	0.8,	0.4	0.2

Thus, the output is $(-1-2)/2 = -1.5$ since when starting from the right (maximum sample) summing the weights, the threshold $T_0 = 1$ is not reached until the weight associated with -1 is added. The underlined sum value above indicates that this is the first sum that meets or exceeds the threshold.

For the sake of notational simplicity, the "signed" samples in the window of the recursive WM filter at time n are denoted by the vector $\mathbf{S}(n) = [\mathbf{S}_Y^T(n), \mathbf{S}_X^T(n)]^T$, where

$$\mathbf{S}_Y(n) = [\text{sgn}(A_1)Y(n-1), \text{sgn}(A_2)Y(n-2), \ldots \text{sgn}(A_N)Y(n-N)]^T \tag{8}$$

is the vector containing the "signed" past output samples, and

$$\mathbf{S}_X(n) = [\text{sgn}(B_0)X(n), \text{sgn}(B_1)X(n+1), \ldots \text{sgn}(B_M)X(n+M)]^T \tag{9}$$

denotes the vector containing the "signed" input observation samples used to compute the filter's output at time n. The ith-order statistic of $\mathbf{S}(n)$ is denoted as $S_{(i)}(n), i = 1, 2, \ldots, L$, where $S_{(1)}(n) \leq S_{(2)}(n) \leq \cdots \leq S_{(L)}(n)$ with $L = N + M + 1$ as the window size. Note that $S_{(i)}$ is the joint order statistic of the signed past output samples in $\mathbf{S}_Y$ and the signed input observation samples in $\mathbf{S}_X$. Furthermore, we let $\mathbf{A} = [A_1, A_2, \ldots, A_N]^T$ and $\mathbf{B} = [B_0, B_1, \ldots, B_M]^T$ be the vectors containing feedback and feed-forward filter coefficients, respectively.

A. Stability of Recursive WM Filters

One of the main problems in the design of linear IIR filters is the stability under the bounded-input bounded-output (BIBO) criterion, which establishes certain constraints on the feedback filter coefficient values. In order to guarantee the BIBO stability of a linear IIR filter, the poles of its transfer function must lie within the unit circle in the complex plane [9]. Unlike linear IIR filters, recursive WM filters are guaranteed to be stable under the BIBO criterion.

Property 1: Recursive weighted median filters, as defined in (7), are stable under the BIBO criterion, regardless of the values taken by the feedback coefficients $\{A_\ell\}$ for $\ell = 1, 2, \ldots, N$.

The proof of Property 1 is straightforward and is not included here. Fig. 2 illustrates, through the impulse response, the stability conditions for the linear IIR filter $Y(n) = A_2Y(n-2) + A_1Y(n-1) + B_0X(n)$ and the recursive WM filter $Y(n) = \text{MEDIAN}(A_2 \diamond Y(n-2), A_1 \diamond Y(n-1), B_0 \diamond X(n))$ for the filter weights: In Fig. 2(a), $\langle\langle A_2, A_1, B_0\rangle\rangle = \langle\langle -0.5, 0, 1\rangle\rangle$. In Fig. 2(b), $\langle\langle A_2, A_1, B_0\rangle\rangle = \langle\langle -1, 0, 1\rangle\rangle$. In Fig. 2(c), $\langle\langle A_2, A_1, B_0\rangle\rangle = \langle\langle 2, 0, 2\rangle\rangle$. Note that in Fig. 2(a), the RWM filter's response is a single pulse at time $n = 0$, whereas the impulse response of the linear IIR filter goes to zero after a few oscillations. Note also, in Fig. 2(b), that the impulse response of the linear IIR filter oscillates indefinitely, whereas the impulse response of the RWM filter reaches its final value after a few oscillations. Fig. 2(c) shows the instability of the linear IIR filter, whereas the RWM filter's response is stable converging to zero after a few oscillations.

III. Recursive WM Filters and Threshold Decomposition

Threshold decomposition is a powerful theoretical tool used in the analysis and design of RWM filters. Conceptually, threshold decomposition was originally formulated to be used only with non-negative integer-valued signals with finite quantization levels [10]. Threshold decomposition was later extended to admit real-valued signal in the analysis of stack smoothers [3] and stack filters [4], [11]. For the purpose of this paper, we adopt a threshold decomposition formulation similar to that described in [4].

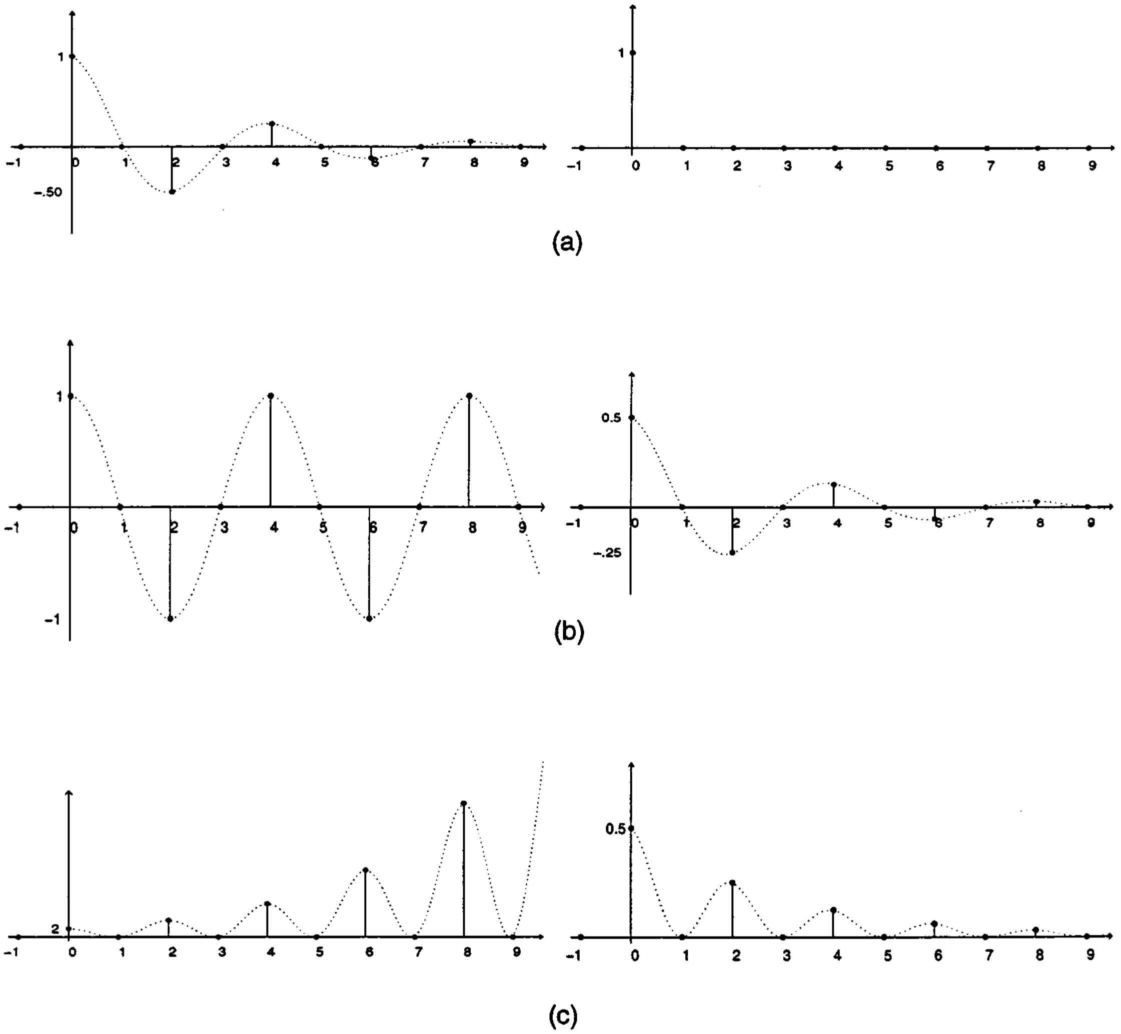

Fig. 2. Impulse responses of linear IIR filter $Y(n) = A_2Y(n-2) + A_1Y(n-1) + B_0X(n)$ (left) and RWM filter $Y(n) = \text{MEDIAN}(A_2 \diamond Y(n-2), A_1 \diamond Y(n-1), B_0 \diamond X(n))$ (right) for (a) $\langle\langle A_2, A_1, B_0\rangle\rangle = \langle\langle -0.5, 0, 1\rangle\rangle$, (b) $\langle\langle A_2, A_1, B_0\rangle\rangle = \langle\langle -1, 0, 1\rangle\rangle$ and (c) $\langle\langle A_2, A_1, B_0\rangle\rangle = \langle\langle 2, 0, 2\rangle\rangle$.

Consider the real-valued vector $\mathbf{Z} = [Z_1, Z_2, \ldots, Z_L]^T$. Threshold decomposition maps this real-valued vector to an infinite set of binary vectors $\mathbf{z}^q \in \{-1, 1\}^L, q \in (-\infty, \infty)$, where

$$\begin{aligned}\mathbf{z}^q &= [\text{sgn}(Z_1 - q), \text{sgn}(Z_2 - q), \ldots, \text{sgn}(Z_L - q)]^T \\ &= [z_1^q, z_2^q, \ldots, z_L^q]^T \end{aligned} \tag{10}$$

where sgn denotes the sign function defined in (3). The original vector $\mathbf{Z}$ can be exactly reconstructed from its binary representation through the inverse process [4] as

$$Z_i = \frac{1}{2}\int_{-\infty}^{+\infty} z_i^q \, dq \tag{11}$$

for $i = 1, 2, \ldots, L$.

Thus, a real-valued vector has a unique threshold signal representation, and vice versa

$$\mathbf{Z} \overset{\text{T.D.}}{\Leftrightarrow} \{\mathbf{z}^q\} \tag{12}$$

where $\overset{\text{T.D.}}{\Leftrightarrow}$ denotes the one-to-one mapping provided by the threshold decomposition operation. Since q can take any real value, the infinite set of binary vectors $\{\mathbf{z}^q\}$ seems redundant in representing the real-valued vector $\mathbf{Z}$. Indeed, some of the binary vectors $\{\mathbf{z}^q\}$ are infinitely repeated. For $Z_{(1)} < q \leq Z_{(2)}$, for instance, all the binary vectors $\{\mathbf{z}^q\}$ are identical. As shown in [3], threshold signal representation can be simplified based on the fact that there are at most $L+1$ different binary vectors $\{\mathbf{z}^q\}$ for each observation vector $\mathbf{Z}$. Using this fact, (12) reduces to

$$\mathbf{Z} \overset{\text{T.D.}}{\Leftrightarrow} \{\mathbf{z}^q\} = \begin{cases} [1, 1, \ldots, 1]^T, & \text{for } -\infty < q \leq Z_{(1)} \\ \left[z_1^{Z_{(i)}^+}, z_2^{Z_{(i)}^+}, \ldots, z_L^{Z_{(i)}^+}\right]^T, & \text{for } Z_{(i)} < q \leq Z_{(i+1)} \\ & \quad 1 \leq i \leq L-1 \\ [-1, -1, \ldots, -1]^T, & \text{for } Z_{(L)} < q < +\infty \end{cases} \tag{13}$$

where $Z_{(i)}^+$ denotes a value on the real line approaching $Z_{(i)}$ from the right. The simplified representation in (13) will be used shortly.

Using the threshold signal decomposition in (10) and (11), the recursive WM operation in (6) can be expressed as

$$\begin{aligned} Y(n) = \mathrm{MEDIAN}\Bigg(& |A_\ell| \\ & \diamond \frac{1}{2}\int_{-\infty}^{+\infty} \mathrm{sgn}[\mathrm{sgn}(A_\ell)Y(n-\ell)-q]\,dq|_{\ell=1}^{N} \\ & |B_k| \diamond \frac{1}{2}\int_{-\infty}^{+\infty} \mathrm{sgn}[\mathrm{sgn}(B_k)X(n+k)-q]\,dq|_{k=0}^{M}\Bigg). \end{aligned} \tag{14}$$

At this point, we resort to the weak superposition property of the nonlinear median operator, which states that applying a weighted median operator to a real-valued signal is equivalent to decomposing the real-valued signal using threshold decomposition, applying the median operator to each binary signal separately, and then adding the binary outputs to obtain the real-valued output [10]. This superposition property leads to interchanging the integral and median operators in the above expression, and thus, (14) becomes

$$\begin{aligned} Y(n) = \frac{1}{2}\int_{-\infty}^{+\infty} & \mathrm{MEDIAN}\big(|A_\ell| \\ & \diamond\, \mathrm{sgn}[\mathrm{sgn}(A_\ell)Y(n-\ell)-q]|_{\ell=1}^{N}, \\ & |B_k| \diamond \mathrm{sgn}[\mathrm{sgn}(B_k)X(n+k)-q]|_{k=0}^{M}\big)\,dq. \end{aligned} \tag{15}$$

To simplify the above expression, let $\{\mathbf{s}_Y^q\}$ and $\{\mathbf{s}_X^q\}$ denote the threshold decomposition of the signed past output samples and the signed input samples respectively, i.e.,

$$\begin{aligned} \mathbf{S}_Y(n) \overset{\text{T.D.}}{\Leftrightarrow} \mathbf{s}_Y^q(n) &= [\mathrm{sgn}[\mathrm{sgn}(A_1)Y(n-1)-q],\ldots \\ &\quad \mathrm{sgn}[\mathrm{sgn}(A_N)Y(n-N)-q]]^T \\ \mathbf{S}_X(n) \overset{\text{T.D.}}{\Leftrightarrow} \mathbf{s}_X^q(n) &= [\mathrm{sgn}[\mathrm{sgn}(B_0)X(n)-q],\ldots \\ &\quad \mathrm{sgn}[\mathrm{sgn}(B_M)X(n+M)-q]]^T \end{aligned}$$

where $q \in (-\infty,+\infty)$. Furthermore, we let $\mathbf{s}^q(n) = [[\mathbf{s}_Y^q(n)]^T,[\mathbf{s}_X^q(n)]^T]^T$ be the threshold decomposition representation of the vector $\mathbf{S}(n) = [\mathbf{S}_Y^T(n),\mathbf{S}_X^T(n)]^T$ containing the signed samples. With this notation and following a similar approach to that presented in [4], it can be shown that (15) reduces to

$$Y(n) = \frac{1}{2}\int_{-\infty}^{+\infty} \mathrm{sgn}\big(\mathbf{A}_a^T\mathbf{s}_Y^q(n) + \mathbf{B}_a^T\mathbf{s}_X^q(n)\big)\,dq \tag{16}$$

where $\mathbf{A}_a$ is the vector whose elements are the magnitudes of the feedback coefficients $\mathbf{A}_a = [|A_1|,|A_2|,\ldots,|A_N|]^T$, and $\mathbf{B}_a$ is the vector whose elements are the magnitudes of the feed-forward coefficients $\mathbf{B}_a = [|B_0|,|B_1|,\ldots,|B_M|]^T$. Note in (16) that the filter's output depends on the signed past outputs, the signed input observations, and the feedback and feedforward coefficients.

The integral term in (16) required to compute the output of the recursive WM filter may seem difficult to implement. It should be emphasized, however, that the representation on (16) will be used for analysis and not for computation. Further simplifications of this expression can be achieved, however, if the fact that the theshold decomposition of the signed samples in the window of the recursive WM filter can take at most $L+1$ different binary vectors is used. This leads to

$$\begin{aligned} Y(n) = & \frac{1}{2}\int_{-\infty}^{S_{(1)}} \mathrm{sgn}\Big(\mathbf{A}_a^T\mathbf{s}_Y^{S_{(1)}}(n) + \mathbf{B}_a^T\mathbf{s}_X^{S_{(1)}}(n)\Big)\,dq \\ & + \frac{1}{2}\sum_{i=1}^{L-1}\int_{S_{(i)}}^{S_{(i+1)}} \mathrm{sgn}\Bigg(\mathbf{A}_a^T\mathbf{s}_Y^{S_{(i)}^+}(n) + \mathbf{B}_a^T\mathbf{s}_X^{S_{(i)}^+}(n)\Bigg)\,dq \\ & + \frac{1}{2}\int_{S_{(L)}^+}^{+\infty} \mathrm{sgn}\big(\mathbf{A}_a^T\mathbf{s}_Y^{+\infty}(n) + \mathbf{B}_a^T\mathbf{s}_X^{+\infty}(n)\big)\,dq. \end{aligned} \tag{17}$$

After some simplifications, (17) reduces to

$$\begin{aligned} Y(n) = & \frac{S_{(1)}(n)+S_{(L)}(n)}{2} + \frac{1}{2}\sum_{i=1}^{L-1}\big(S_{(i+1)}(n) - S_{(i)}(n)\big) \\ & \times \mathrm{sgn}\Bigg(\mathbf{A}_a^T\mathbf{s}_Y^{S_{(i)}^+}(n) + \mathbf{B}_a^T\mathbf{s}_X^{S_{(i)}^+}(n)\Bigg) \end{aligned} \tag{18}$$

where $S_{(i)}$ is the ith-order statistic of the "signed" sample. The filter representation in (18) provides us with an interesting interpretation of recursive WM filters. The filter output at time n is computed by the sum of the midrange of the signed samples $((S_{(1)}+S_{(N)})/2)$ and the linear combination of the differences between successive order statistics $(S_{(i)}(n) - S_{(i-1)}(n))$.

IV. Adaptive Recursive WM Filtering Algorithm

In general, the coefficients of the recursive WM filter have to be designed in some optimal fashion. In this section, we develop the first adaptive optimization algorithm for the design of recursive WM filters. Threshold decomposition developed in the last section is used to find a closed-form and adaptive expression for the optimal weights. Since recursive WM smoothers are included in the class of recursive WM filters, the optimization algorithms developed here will be suited for the optimization of recursive WM smoothers.

The main objective of the optimization is to find the best filter coefficients such that a performance cost criterion is minimized. A criterion widely used in the design of median-based filters is the mean absolute error (MAE) between the filter's output and the desired signal. Consider an observed process $\{X(n)\}$ that is statistically related to a desired process $\{D(n)\}$. Further, assume that both processes are jointly stationary. Under the MAE criterion, the goal is to determine the weights $\{A_\ell\}|_{\ell=1}^{N}$ and $\{B_k\}|_{k=0}^{M}$ to minimize the cost function

$$J(A_1,\ldots,A_N,B_0,\ldots,B_M) = E\{|D(n)-Y(n)|\} \tag{19}$$

where $E\{\cdot\}$ denotes the statistical expectation, and $Y(n)$ is the output of the recursive WM filter given in (7).

To form an iterative optimization algorithm, the steepest descent algorithm is used, in which the filter coefficients are updated according to

$$\begin{aligned} A_\ell(n+1) = & A_\ell(n) \\ & + 2\mu\left[-\frac{\partial}{\partial A_\ell}J(A_1,\ldots,A_N,B_0,\ldots,B_M)\right] \\ & \ell = 1,\ldots,N \end{aligned}$$

$$B_k(n+1) = B_k(n) + 2\mu\left[-\frac{\partial}{\partial B_k}J(A_1,\ldots,A_N,B_0,\ldots,B_M)\right] \qquad k=0,\ldots,M. \quad (20)$$

Note that in (20), the gradient of the cost function (∇J) has to be previously computed to update the filter weights. Due to the feedback operation inherent in the recursive WM filter, however, the computation of ∇J becomes intractable.

To overcome this problem, the optimization framework refered to as *equation error formulation* is used [12]. Equation error formulation is used in the design of linear IIR filters and is based on the fact that ideally the filter's output is close to the desired response. The lagged values of $Y(n)$ in (7) can thus be replaced with the corresponding lagged values $D(n)$. Hence, the previous outputs $Y(n-\ell)|_{\ell=1}^{N}$ are replaced with the previous desired outputs $D(n-\ell)|_{\ell=1}^{N}$ to obtain a two-input, single-output filter that depends on the input samples $X(n+k)|_{k=0}^{M}$ and on delay samples of the desired response $D(n-l)|_{l=1}^{N}$, namely

$$\begin{aligned}\hat{Y}(n) = \text{MEDIAN}(&|A_N|\diamond\text{sgn}(A_N)D(n-N),\ldots\\ &|A_1|\diamond\text{sgn}(A_1)D(n-1),|B_0|\diamond\text{sgn}(B_0)X(n),\ldots\\ &|B_M|\diamond\text{sgn}(B_M)X(n+M)).\end{aligned} \quad (21)$$

The approximation leads to an output $\hat{Y}(n)$ that does not depend on delayed output samples, and, therefore, the filter no longer introduces feedback reducing the output to a two-input, single-output nonrecursive system. This "recursive decoupling" optimization approach provides the key to a gradient-based optimization algorithm for recursive WM filters.

According to the approximate filtering structure, the cost function to be minimized is

$$\hat{J}(A_1,\ldots,A_N,B_0,\ldots B_M) = E\{|D(n)-\hat{Y}(n)|\} \quad (22)$$

where $\hat{Y}(n)$ is the nonrecursive filter output (21). Since $D(n)$ and $X(n)$ are not functions of the feedback coefficients, the derivative of $\hat{J}(A_1,\ldots,A_N,B_0,\ldots B_M)$ with respect to the filter weights is nonrecursive, and its computation is straightforward. The adaptive optimization algorithm using the steepest descent method (20), where $J(A_1,\ldots,A_N,B_0,\ldots,B_M)$ is replaced by $\hat{J}(A_1,\ldots,A_N,B_0,\ldots,B_M)$, is derived as follows. Define the vector $\mathbf{S}(n) = [\mathbf{S}_D^T(n),\mathbf{S}_X^T(n)]^T$ as that containing the signed samples in the sliding window of the two-input, single-output nonrecursive filter (21) at time n, where $\mathbf{S}_D(n) = [\text{sgn}(A_1)D(n-1),\text{sgn}(A_2)D(n-2),\ldots,\text{sgn}(A_N)D(n-N)]^T$, and $\mathbf{S}_X(n)$ is given by (9). With this notation and using threshold decomposition, (22) becomes

$$\hat{J}(A_1,\ldots,A_N,B_0,\ldots,B_M) = \frac{1}{2}E\left\{\left|\int_{-\infty}^{+\infty}[\text{sgn}(D(n)-q) - \text{sgn}\big(\mathbf{A}_a^T\mathbf{s}_D^q(n)+\mathbf{B}_a^T\mathbf{s}_X^q(n)\big)\,dq]\right|\right\} \quad (23)$$

where $\{\mathbf{s}_D^q(n)\}$ is the corresponding threshold decomposition of the vector $\mathbf{S}_D(n)$.

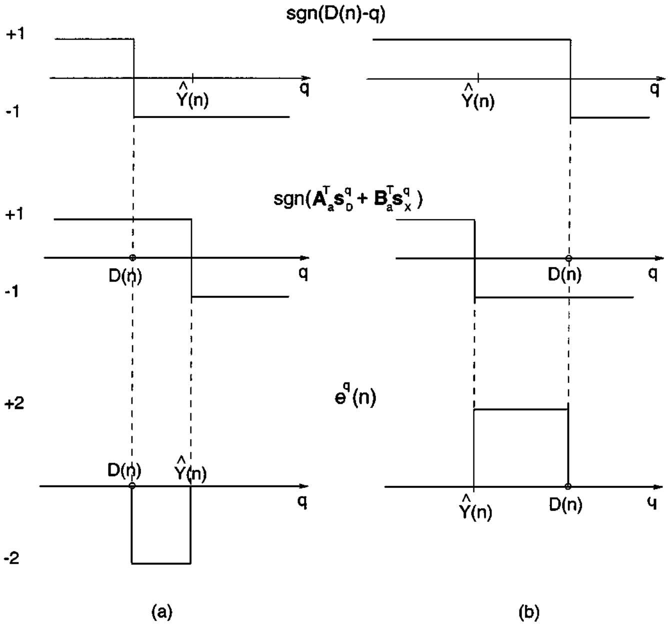

Fig. 3. Threshold decompositions of the desired signal $D(n)$, filter output $\hat{Y}(n)$, and error function $e(n) = D(n) - \hat{Y}(n)$. (a) $D(n) < \hat{Y}(n)$. (b) $D(n) > \hat{Y}(n)$.

Now, let $e^q(n)$ be the argument inside the integral operator, i.e., $e^q(n) = \text{sgn}(D(n)-q)-\text{sgn}(\mathbf{A}_a^T\mathbf{s}_D^q(n)+\mathbf{B}_a^T\mathbf{s}_X^q(n))$. Note that $e^q(n)$ can be thought of as the threshold decomposition of the error function $e(n) = D(n) - \hat{Y}(n)$ for a fixed n. Fig. 3 shows $e^q(n)$ for two different cases. Fig. 3(a) shows the case where the desired filter's output $D(n)$ is less than the filter's output $\hat{Y}(n)$. Fig. 3(b) shows the second case where the desired filter output $D(n)$ is greater than the filter output $\hat{Y}(n)$. The case where the desired response is equal to the filter's output is not shown in Fig. 3. Note that for a fixed n, the integral operator in (23) acts on a strictly negative function [Fig. 3(a)] or a strictly positive function [Fig. 3(b)], and therefore, the absolute value and integral operators in (23) can be interchanged, leading to

$$\hat{J}(A_1,\ldots,A_N,B_0,B_1,\ldots,B_M) = \frac{1}{2}\int_{-\infty}^{+\infty} E[|e^q(n)|]\,dq \quad (24)$$

where we have used the linear property of the expectation.

Fig. 3 also depicts that $e^q(n)$ can only take on values in the set $\{-2,0,2\}$; therefore, the absolute value operator can be replaced by a properly scaled second power operator. Thus

$$\hat{J}(A_1,\ldots,A_N,B_0,B_1,\ldots,B_M) = \frac{1}{4}\int_{-\infty}^{+\infty} E[(e^q(n))^2]\,dq. \quad (25)$$

Taking derivatives of the above expression with respect to the filter coefficients A_ℓ and B_k yields, respectively

$$\frac{\partial}{\partial A_\ell}\hat{J}(A_1,\ldots,A_N,B_0,\ldots B_M) = -\frac{1}{2}\int_{-\infty}^{+\infty} E\left[e^q(n)\frac{\partial}{\partial A_\ell}\text{sgn}\big(\mathbf{A}_a^T\mathbf{s}_D^q+\mathbf{B}_a^T\mathbf{s}_X^q\big)\right] \quad (26)$$

$$\frac{\partial}{\partial B_k}\hat{J}(A_1,\ldots,A_N,B_0,\ldots B_M) = -\frac{1}{2}\int_{-\infty}^{+\infty} E\left[e^q(n)\frac{\partial}{\partial B_k}\text{sgn}\big(\mathbf{A}_a^T\mathbf{s}_D^q+\mathbf{B}_a^T\mathbf{s}_X^q\big)\right]. \quad (27)$$

Since the sgn function has a discontinuity at the origin, it introduces the *dirac* function in its derivative, which is not convenient for further analysis. In order to overcome this difficulty, the sgn function is approximated by the differentiable hyperbolic tangent function $\mathrm{sgn}(x) \approx \tanh(x)$ whose derivative is $(\partial/\partial x)\tanh(x) = \mathrm{sech}^2(x)$. Using this approximation in (26), it follows that

$$\begin{aligned}\frac{\partial}{\partial A_\ell}&\,\mathrm{sgn}\big(\mathbf{A}_a^T\mathbf{s}_D^q + \mathbf{B}_a^T\mathbf{s}_X^q\big)\\ &\approx \mathrm{sech}^2\big(\mathbf{A}_a^T\mathbf{s}_D^q + \mathbf{B}_a^T\mathbf{s}_X^q\big)\frac{\partial}{\partial A_\ell}\big(\mathbf{A}_a^T\mathbf{s}_D^q + \mathbf{B}_a^T\mathbf{s}_X^q\big)\\ &= \mathrm{sech}^2\big(\mathbf{A}_a^T\mathbf{s}_D^q + \mathbf{B}_a^T\mathbf{s}_X^q\big)\,\mathrm{sgn}(A_\ell)s_{D_\ell}^q \qquad (28)\end{aligned}$$

where $s_{D_\ell}^q$ is the ℓth component of the vector $\mathbf{s}_D^q$. Similarly

$$\begin{aligned}\frac{\partial}{\partial B_k}&\,\mathrm{sgn}\big(\mathbf{A}_a^T\mathbf{s}_D^q + \mathbf{B}_a^T\mathbf{s}_X^q\big)\\ &\approx \mathrm{sech}^2\big(\mathbf{A}_a^T\mathbf{s}_D^q + \mathbf{B}_a^T\mathbf{s}_X^q\big)\frac{\partial}{\partial B_k}\big(\mathbf{A}_a^T\mathbf{s}_D^q + \mathbf{B}_a^T\mathbf{s}_X^q\big)\\ &= \mathrm{sech}^2\big(\mathbf{A}_a^T\mathbf{s}_D^q + \mathbf{B}_a^T\mathbf{s}_X^q\big)\,\mathrm{sgn}(B_k)s_{X_k}^q \qquad (29)\end{aligned}$$

where $s_{X_k}^q$ is the kth component of the vector $\mathbf{s}_X^q$. Using (28) and (29) in (26) and (27), respectively, it follows that

$$\begin{aligned}\frac{\partial}{\partial A_\ell}&\hat{J}(A_1,\ldots,A_N,B_0,\ldots,B_M)\\ &\approx -\frac{1}{2}\int_{-\infty}^{+\infty} E\left[e^q(n)\,\mathrm{sech}^2\big(\mathbf{A}_a^T\mathbf{s}_D^q + \mathbf{B}_a^T\mathbf{s}_X^q\big)\right.\\ &\quad\left.\times\ \mathrm{sgn}(A_\ell)s_{D_\ell}^q\right] dq \qquad (30)\\ \frac{\partial}{\partial B_k}&\hat{J}(A_1,\ldots,A_N,B_0,\ldots,B_M)\\ &\approx -\frac{1}{2}\int_{-\infty}^{+\infty} E\left[e^q(n)\,\mathrm{sech}^2\big(\mathbf{A}_a^T\mathbf{s}_D^q + \mathbf{B}_a^T\mathbf{s}_X^q\big)\right.\\ &\quad\left.\times\ \mathrm{sgn}(B_k)s_{X_k}^q\right] dq.\end{aligned}$$

Using (30) in (20), the weighted updates reduce to

$$\begin{aligned}A_\ell(n+1) &= A_\ell(n)\\ &\quad + 2\mu\left[\frac{1}{2}\int_{-\infty}^{+\infty} E\left[e^q(n)\,\mathrm{sech}^2\big(\mathbf{A}_a^T\mathbf{s}_D^q + \mathbf{B}_a^T\mathbf{s}_X^q\big)\right.\right.\\ &\quad\left.\left.\times\ \mathrm{sgn}(A_\ell)s_{D_\ell}^q\right] dq\right] \qquad (31)\\ B_k(n+1) &= B_k(n)\\ &\quad + 2\mu\left[\frac{1}{2}\int_{-\infty}^{+\infty} E\left[e^q(n)\,\mathrm{sech}^2\big(\mathbf{A}_a^T\mathbf{s}_D^q + \mathbf{B}_a^T\mathbf{s}_X^q\big)\right.\right.\\ &\quad\left.\left.\times\ \mathrm{sgn}(B_k)s_{X_k}^q\right] dq\right] \qquad (32)\end{aligned}$$

for $\ell = 1,\ldots,N$, and $k = 0,\ldots,M$. Now, as shown in Fig. 3, the threshold decomposition of the error term $e^q(n)$ takes nonzero values only if q is between the desired output $D(n)$ and the actual filter output $\hat{Y}(n)$. Assuming that the desired output $D(n)$ is one of the signed samples,[2] say $S_{(m)}$, and that the actual output $\hat{Y}(n)$ is $S_{(j)}$, $e^q(n) = 0$ for $q \in \{(-\infty, \min(S_{(m)}, S_{(j)})] \cup (\max(S_{(m)}, S_{(j)}), +\infty)\}$. Thus, the adaptive optimization algorithm reduces to

$$\begin{aligned}A_\ell(n+1) = A_\ell(n) + \mu\int_{\min(S_{(m)},S_{(j)})}^{\max(S_{(m)},S_{(j)})} e^q(n)&\\ \times\ \mathrm{sech}^2\big(\mathbf{A}_a^T\mathbf{s}_D^q + \mathbf{B}_a^T\mathbf{s}_X^q\big)\,\mathrm{sgn}(A_\ell)s_{D_\ell}^q\,dq&\end{aligned}$$

where the instantaneous estimate for the gradient is used. Evaluating the above integral leads to

$$\begin{aligned}A_\ell(n+1) &= A_\ell(n)\\ &\quad + \mu\,\mathrm{sgn}(A_\ell)\sum_{i=\min(m,j)}^{\max(m,j)-1}\big(S_{(i+1)} - S_{(i)}\big)e^{S_{(i)}^+}(n)\\ &\quad \times\ \mathrm{sech}^2\left(\mathbf{A}_a^T\mathbf{s}_D^{S_{(i)}^+} + \mathbf{B}_a^T\mathbf{s}_X^{S_{(i)}^+}\right)s_{D_\ell}^{S_{(i)}^+} \qquad (33)\end{aligned}$$

for $\ell = 1,\ldots,N$. Similar simplifications can be made to (32), leading to

$$\begin{aligned}B_k(n+1) &= B_k(n)\\ &\quad + \mu\,\mathrm{sgn}(B_k)\sum_{i=\min(m,j)}^{\max(m,j)-1}\big(S_{(i+1)} - S_{(i)}\big)e^{S_{(i)}^+}(n)\\ &\quad \times\ \mathrm{sech}^2\left(\mathbf{A}_a^T\mathbf{s}_D^{S_{(i)}^+} + \mathbf{B}_a^T\mathbf{s}_X^{S_{(i)}^+}\right)s_{X_k}^{S_{(i)}^+} \qquad (34)\end{aligned}$$

for $k = 0, 1, \ldots, M$.

Since the MAE criterion was used in the derivation, the recursion in (33) and (34) is referred to as the *least mean absolute* (LMA) recursive weighted median adaptive algorithm. The principle of the adaptive optimization algorithm can be explained as follows. Assume that the desired signal is larger than the filter's output, i.e., $m > j$. From Fig. 3(b), the threshold decomposition of the error signal $e^q(n)$ takes positive values for $q \in (S_{(j)}, S_{(m)}]$, which is the interval of interest where $e^q(n) \neq 0$. Now, for those signed samples whose magnitudes are smaller than the actual filter's output, their corresponding weights are reduced since the threshold decompositions of those signed samples ($S_{(i)}$ for $i < j$) are -1 in the interval $(S_{(j)}, S_{(m)}]$. On the other hand, for those signed samples whose magnitudes are larger than the desired signal, their corresponding weights are either increased (more positive) if the weights take positive values or decreased (more negative) if the weights are negatives. Consequently, both cases will lead to updated weights that will push the estimate toward the desired output. Similar conclusions can be reached when the desired signal is smaller than the filter's output. Note that if the desired response is equal to the filter's output, $e^q(n)$ is equal to zero for all q, and therefore, the filter coefficients remain unchanged, and no update is needed. The following are some remarks of the optimization algorithm.

An interesting characteristic of this optimization algorithm is that at each iteration, the feedback filter coefficients do not have to satisfy any stability constraints since, as shown in Sec-

[2] For simplicity in the analysis, we will approximate $D(n)$ as one of the signed samples. This approximation is not necessary, but since it provides accurate results and simplifies the notation significantly, we find it useful.

tion II, the recursive WM filters are always stable under the BIBO criterion, regardless of the values taken by the feedback filter weights. This advantage of the recursive WM filters is not shared by the linear IIR filters, where the feedback filter coefficients are contrained such that the poles of the transfer function fall inside the unit circle.

The adaptive optimization algorithm is suitable for the design of recursive WM smoothers that do not admit negative weight values. On closer examination, it turns out that the constraint of having non-negative weight can be accomplished by a projection operator that maps all the negative weights to zero. Using this constraint, the adaptive optimization algorithm for recursive WM smoothers reduces to

$$\begin{aligned} A_\ell(n+1) = P\Bigg[& A_\ell(n) \\ & + \mu\,\text{sgn}(A_\ell) \sum_{i=\min(m,j)}^{\max(m,j)-1} \left(S_{(i+1)} - S_{(i)}\right) e^{S^+_{(i)}}(n) \\ & \times \text{sech}^2\left(\mathbf{A}_a^T \mathbf{s}_D^{S^+_{(i)}} + \mathbf{B}_a^T \mathbf{s}_X^{S^+_{(i)}}\right) s_{D_\ell}^{S^+_{(i)}} \Bigg] \end{aligned} \tag{35}$$

$$\begin{aligned} B_k(n+1) = P\Bigg[& B_k(n) \\ & + \mu\,\text{sgn}(B_k) \sum_{i=\min(m,j)}^{\max(m,j)-1} \left(S_{(i+1)} - S_{(i)}\right) e^{S^+_{(i)}}(n) \\ & \times \text{sech}^2\left(\mathbf{A}_a^T \mathbf{s}_D^{S^+_{(i)}} + \mathbf{B}_a^T \mathbf{s}_X^{S^+_{(i)}}\right) s_{X_k}^{S^+_{(i)}} \Bigg] \end{aligned}$$

where $P(\cdot)$ is the projection operator

$$P(x) = \begin{cases} x & x \geq 0 \\ 0 & x < 0. \end{cases} \tag{36}$$

This LMA optimization algorithm can be further simplified to obtain a faster adaptive optimization algorithm following similar arguments to those used in [3] and [4]. On closer examination of (33) and (34), it turns out that the contribution to each term is to a large extent determined by the nonlinear function $\text{sech}^2(\mathbf{A}_a^T \mathbf{s}_D^q + \mathbf{B}_a^T \mathbf{s}_X^q)$. The $\text{sech}^2(\cdot)$ function achieves its maximum value when the argument is zero. Its value decreases rapidly and monotonically to zero as the argument departs from zero. For some value q, $\mathbf{A}_a^T \mathbf{s}_D^q + \mathbf{B}_a^T \mathbf{s}_X^q$ takes its closest value to zero, and therefore, the update term corresponding to this q will produce the bigest contribution in the update of the weights. The value of q for which the largest update contribution occurs can be found through the definition of the RWM filter. The output of the RWM filter is $S_{(j)}(n)$ if and only if the following three inequalities are satisfied simultaneously:

$$\begin{aligned} \mathbf{A}_a^T \mathbf{s}_D^{S_{(j-1)}} + \mathbf{B}_a^T \mathbf{s}_X^{S_{(j-1)}} &> 0 \\ \mathbf{A}_a^T \mathbf{s}_D^{S_{(j)}} + \mathbf{B}_a^T \mathbf{s}_X^{S_{(j)}} &\geq 0 \\ \mathbf{A}_a^T \mathbf{s}_D^{S_{(j+1)}} + \mathbf{B}_a^T \mathbf{s}_X^{S_{(j+1)}} &< 0 \end{aligned} \tag{37}$$

where we assume $S_{(j-1)}, S_{(j)}$, and $S_{(j+1)}$ are distinct. Since $S_{(j)}$ is the output of the RWM filter at time n, $S_{(j)}$ is also equal to $\hat{Y}(n)$. Thus, $q = \hat{Y}(n)$ produces the largest update contribution. Since

$$\begin{aligned} \mathbf{S}_D^{\hat{Y}(n)} = \Big[& \text{sgn}\left(S_{D_1} - \hat{Y}(n)\right), \text{sgn}\left(S_{D_2} - \hat{Y}(n)\right), \ldots \\ & \text{sgn}\left(S_{D_N} - \hat{Y}(n)\right)\Big]^T \end{aligned}$$

and

$$\begin{aligned} \mathbf{S}_X^{\hat{Y}(n)} = \Big[& \text{sgn}\left(S_{X_1} - \hat{Y}(n)\right), \text{sgn}\left(S_{X_2} - \hat{Y}(n)\right), \ldots \\ & \text{sgn}\left(S_{X_M} - \hat{Y}(n)\right)\Big]^T \end{aligned}$$

the algorithm in (33) and (34) is simplified, leading to the recursion referred to as the fast LMA adaptive algorithm

$$\begin{aligned} A_\ell(n+1) &= A_\ell(n) + \mu(D(n) - \hat{Y}(n)) \\ &\quad \times \text{sgn}(A_\ell(n))\,\text{sgn}\left(S_{D_\ell} - \hat{Y}(n)\right) \end{aligned} \tag{38}$$

$$\begin{aligned} B_k(n+1) &= B_k(n) + \mu(D(n) - \hat{Y}(n)) \\ &\quad \text{sgn}(B_k(n))\,\text{sgn}\left(S_{X_k} - \hat{Y}(n)\right) \end{aligned} \tag{39}$$

for $\ell = 1, 2, \ldots, N$ and $k = 0, 1, \ldots, M$, where $S_{X_k} = \text{sgn}(B_k)X(n+k)$, and $S_{D_\ell} = \text{sgn}(A_\ell)D(n-\ell)$.

Due to the nonlinear nature of the adaptive algorithm, a convergence analysis cannot be derived. Thus, exact bounds on the step-size μ are not available. We have observed in practice that a reliable guideline to select the step size of this algorithm is to select a step size on the order of that required for the standard LMS algorithm. Another approach is to use a variable step size $\mu(n)$, where $\mu(n)$ decreases as the training progresses.

V. Applications of Recursive WM Filters

This section illustrates the performance of the RWM filters. In the first example, the performance of the RWM filters is compared with that of the nonrecursive counterpart in image denoising. In the second example, a bandpass recursive WM filter is designed using the LMA and fast LMA adaptive optimization algorithms described in the previous section. The performance of the resultant filters is compared with that of a linear IIR filter and of a nonrecursive WM filter designed for the same application.

A. Image Denoising

Fig. 4(a) shows the original "portrait" image used in the simulations. The noisy image in Fig. 4(b) is obtained by corrupting the original image with impulsive noise. Each pixel in the image has a 10% probability of being contamined with an impulse. The impulsives occur randomly and were generated using MATLAB's imnoise function.

The noisy image is filtered by a 3 × 3 nonrecursive center WM filter[3] and by a 3 × 3 nonrecursive center WM filter with the same set of weights [13]. Fig. 4(c) and (d) show their respective filter outputs with a center weight $W_c = 5$. Note that the recursive WM filter is more effective than its nonrecursive counterpart.

[3] The center WM operation refers here to the WM filter where all the samples in the window are weighted by 1, except for the center sample, which is weighted by $W_c > 1$ [13].

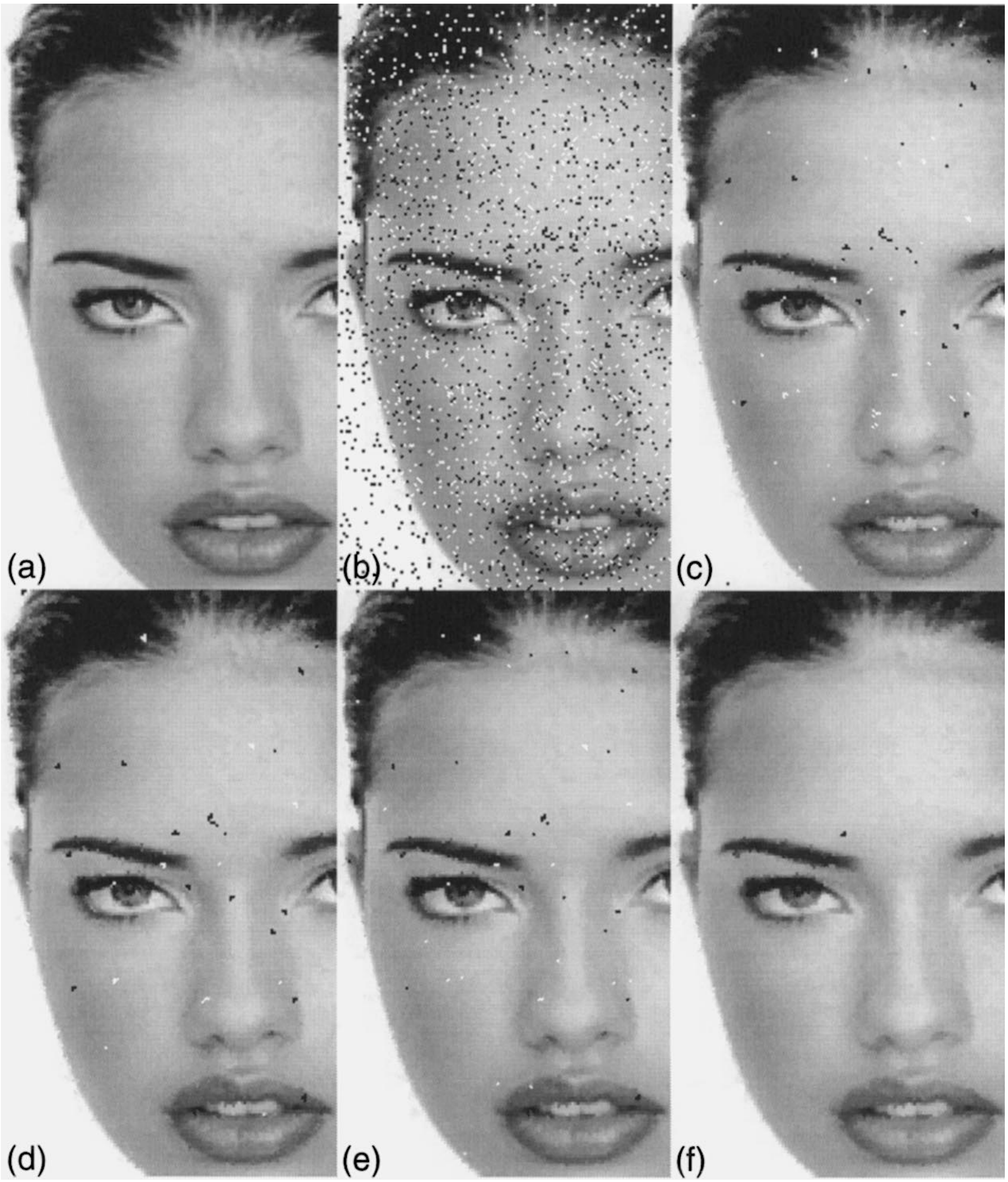

Fig. 4. Image denoising using 3 × 3 recursive and nonrecursive WM filters. (a) Original. (b) Image with salt and pepper noise. (c) Nonrecursive center WM filter. (d) Recursive center WM filter. (e) Optimal nonrecursive WM filter. (f) Optimal RWM filter.

A small 60 × 60 pixel area in the upper left part of the original and noisy images are used to train the recursive WM filter using the adaptive algorithm given by (35). The same training data are used to train a nonrecursive WM filter using the adaptive algorithm described in [3]. The initial conditions for the weights for both algorithms were the filter coefficients of the center WM filters described above. The step size used was 10^{-3} for both adaptive algorithms. The optimal weights found by the adaptive algorithms are $\langle 1.38, 1.64, 1.32 \mid 1.50, \underline{5.87}, 2.17 \mid 0.63, 1.36, 2.24\rangle$ for the nonrecursive WM filter and $\langle\langle 1.24, 1.52, 2.34 \mid 2.07, \underline{4.89}, 1.45 \mid 1.95, 0.78, 2.46\rangle\rangle$ for the RWM filter, where the underline weight is associated with the center sample of the 3× 3 window. The optimal filters determined by the training algorithms were used to filter the entire image. Fig. 4(f) and (e) show the output of the optimal RWM filter and the output of the nonrecursive WM filter, respectively. The normalized mean square errors and the normalized mean absolute errors produced by each of the filters are listed in Table I. As can be seen by comparing the images and the error values, recursive WM filters outperform nonrecursive WM filters.

B. Design of a Bandpass RWM Filter

In this example, we use both LMA and fast LMA adaptive optimization algorithms developed in Section V to design a robust bandpass recursive WM filter. The performance of the designed recursive WM filter is compared with the performances

TABLE I
RESULTS FOR IMPULSIVE NOISE REMOVAL

	Normalized mean square error	Normalized mean absolute error
Noisy image	2545.20	12.98
Recursive center WM filter	189.44	1.69
Non-recursive center WM filter	243.83	1.92
Optimal non-recursive WM filter	156.30	1.66
Optimal RWM filter	88.13	1.57

of a linear FIR filter, a linear IIR filter, and a nonrecursive WM filter all designed for the same task. Moreover, to show the noise attenuation capability of the recursive WM filter and compare it with those of the other filters, an impulsive noisy signal is used as test signal.

The application at hand is the design of a 62-tap bandpass RWM filter with passband $0.075 \leq \omega \leq 0.125$ (normalized frequency with Nyquist $= 1$). We use white Gaussian noise with zero mean and variance equal to one as input training signal. The desired signal is provided by the output of a large FIR filter (122-tap linear FIR filter) designed by MATLAB's fir1 function. The 31 feedback filter coefficients were initialized to small random numbers (on the order of 10^{-3}). The feedforward filter coefficients were initialized to the values outputed by MATLAB's fir1 with 31 taps and the same passband of interest. A variable step size $\mu(n)$ was used in both adaptive optimizations, where $\mu(n)$ changes according to $\mu_0 \exp[-n/100]$ with $\mu_0 = 10^{-2}$.

As a test signal, we need to use a signal that spans all the range of frequencies of interest. Fig. 5(a) depicts a linear swept-frequency signal spanning instantaneous frequencies form 0 to 400 Hz, with a sampling rate of 2 kHz. Fig. 5(b) shows the chirp signal filtered by the 122-tap linear FIR filter that was used as the filter that produced the desired signal during the training stage. Fig. 5(c) shows the output of a 62-tap linear FIR filter used here for comparison purposes. The adaptive optimization algorithm described in [4] was used to optimize a 62-tap nonrecursive WM filter admitting negative weights. The filtered signal attained with the optimized weights is shown in Fig. 5(d). Note that the nonrecursive WM filter tracks the frequencies of interest but fails to attenuate completely the frequencies out of the desired passband. MATLAB's yulewalk function was used to design a 62-tap linear IIR filter with passband $0.075 \leq \omega \leq 0.125$. Fig. 5(e) depicts the linear IIR filter's output. Finally, Fig. 5(f) shows the output of the optimal recursive WM filter determined by the LMA training algorithm. Note that the frequency components of the test signal that are not in the passband are attenuated completely. Moreover, the RWM filter generalizes very well on signals that were not used during the training stage. The optimal RWM filter determined by the fast LMA training algorithm yields similar performance to that of the optimal RWM filter determined by the LMA training algorithm, and therefore, its output is not shown.

Comparing the different filtered signals in Fig. 5, it can be seen that the recursive filtering operation performs much better than its nonrecursive counterpart having the same number of coefficients. Alternatively, to achieve a specified level of performance, a recursive WM filter generally requires considerably far fewer filter coefficients than the corresponding nonrecursive WM filter.

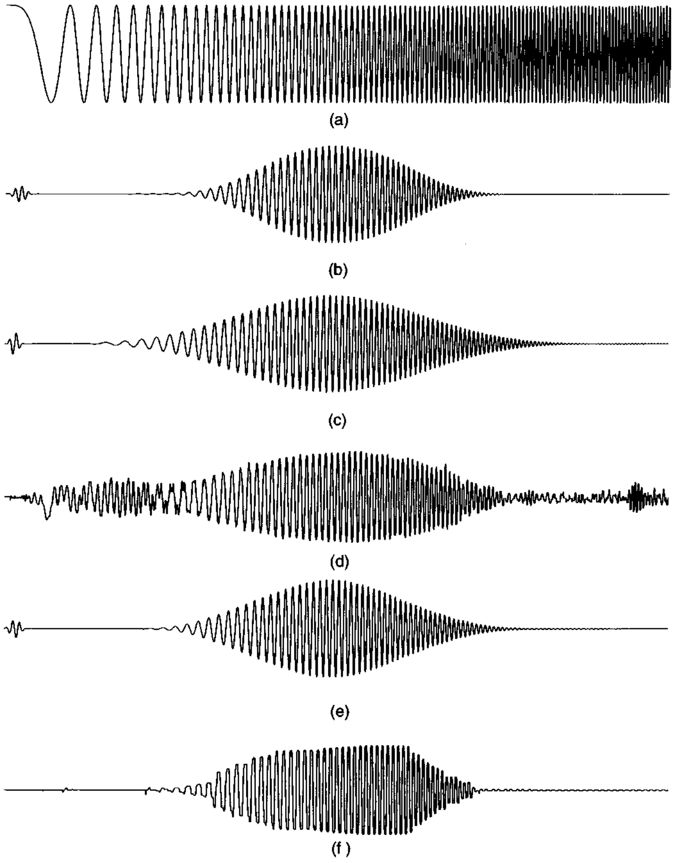

Fig. 5. Bandpass filter design. (a) Input test signal. (b) Desired signal. (c) Linear FIR filter output. (d) Nonrecursive WM filter output. (e) Linear IIR filter output. (f) RWM filter output.

In order to test the robustness of the different filters, the test signal is contaminated with additive α-stable noise, as shown in Fig. 6(a). The parameter $\alpha = 1.4$ was used, simulating noise with impulsive characteristics [14]. Fig. 6(a) is truncated so that the same scale is used in all the plots. Fig. 6(b) and 6(d) show the filter outputs of the linear FIR and the linear IIR filters, respectively. Both outputs are severely affected by the noise. On the other hand, the nonrecursive and recursive WM filters' outputs, shown in Fig. 6(c) and 6(e), respectively, remain practically unaltered. Fig. 6 clearly depicts the robust characteristics of median-based filters.

To better evaluate the frequency response of the various filters, a frequency domain analysis is performed. Due to the nonlinearity inherent in the median operation, traditional linear tools, like transfer function-based analysis, cannot be applied. However, if the nonlinear filters are treated as a single-input single-output system, the magnitude of the frequency response can be experimentally obtained as follows. A single tone sinusoidal signal $\sin(2\pi f t)$ is given as the input to each filter, where f spans the complete range of possible frequencies. A sufficiently large number of frequencies spanning the interval $[0, 1]$ is chosen. For each frequency value, the mean power of each filter's output is computed. Fig. 7(a) shows a plot of the normalized mean power versus frequency attained by the different filters. On closer examination of Fig. 7(a), it can be seen that the recursive WM filter yields the flattest response

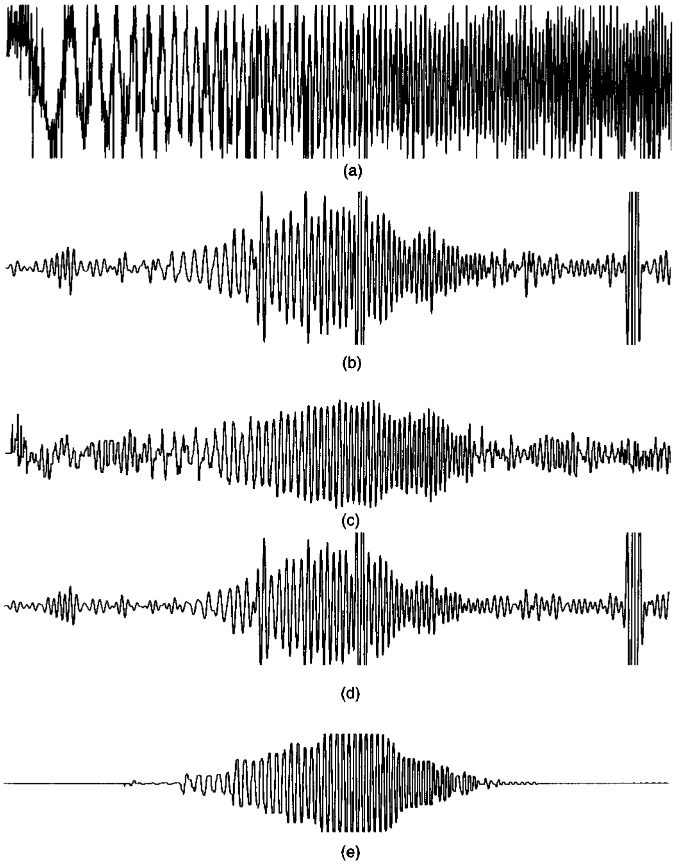

Fig. 6. Performance of the bandpass filter in noise. (a) Chirp test signal in stable noise. (b) Linear FIR filter output. (c) Nonrecursive WM filter output. (d) Linear IIR filter output. (e) RWM filter output.

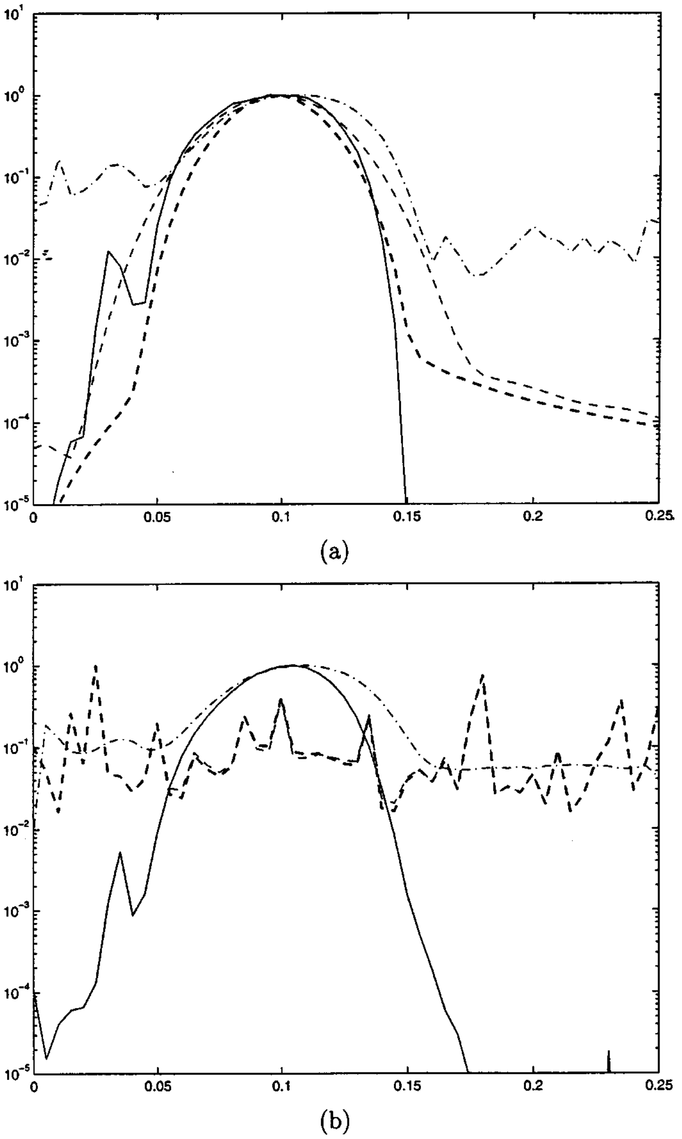

Fig. 7. Frequency response (a) to a noiseless sinusoidal signal (b) to a noisy sinusoidal signal. (—) RWM. (— · — · —) nonrecursive WM filter (- - -) linear FIR filter. (- - -) linear IIR filter.

in the passband of interest. A similar conclusion can be drawn from the time domain plots shown in Fig. 5.

In order to see the effects that impulsive noise has over the magnitude of the frequency response, a contamined sinusoidal signal $\sin(2\pi ft) + \eta$ is given as the input to each filter, where η is α-stable noise with parameter $\alpha = 1.4$. Following the same procedure described above, the mean power versus frequency diagram is obtained and shown in Fig. 7(b). As expected, the magnitudes of the frequency responses for the linear filters are highly distorted, whereas the magnitudes of the frequency responses for the median-based filters do not change significantly with noise.

VI. Conclusion

In this paper, two important contributions were presented. First, the class of recursive WM filters admitting negative weights was introduced. This new filtering framework is useful in applications that require a robust bandpass or highpass characteristic, together with near-perfect “stopband” characteristics. This filtering structure is analogous to linear IIR filters, but unlike linear IIR filters, RWM filters are always stable under the BIBO criterion, regardless of the values taken by the feedback coefficients. Moreover, in comparing the performance of the RWM filters with that of linear IIR filters, RWM filters perform as well as linear IIR filters acting on noiseless signals. In the presence of impulsive noise, linear IIR filters perform poorly, whereas the performance of RWM filters remains practically unalterable by the noise components.

The second contribution of this paper is the introduction of the first adaptive optimization algorithm for the design of recursive WM filters. The “recursive-decoupling” algorithm introduced in Section V uses the threshold decomposition representation to find a closed-form and adaptive expression for the update of the filter coefficients.

Computer-based simulations illustrate the advantages of recursive WM filter over their nonrecursive counterparts. Alternatively, to achieve a specified level of performance, a recursive WM filter generally requires considerably fewer filter coefficients than the corresponding nonrecursive WM filter.

References

[1] D. R. K. Brownrigg, “The weighted median filter,” *Commun. Assoc. Comput. Mach.*, vol. 27, Aug. 1984.

[2] O. Yli-Harja, J. Astola, and Y. Neuvo, "Analysis of the properties of median and weighted median filters using threshold logic and stack filter representation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 39, Feb. 1991.

[3] L. Yin and Y. Neuvo, "Fast adaptation and performance characteristic of FIR-WOS hybrid filters," *IEEE Trans. Signal Processing*, vol. 42, July 1994.

[4] G. R. Arce, "A general weighted median filter structure admitting negative weights," *IEEE Trans. Signal Processing*, vol. 46, pp. 3195–3205, Dec. 1998.

[5] ——, "Statistical threshold decomposition for recursive and nonrecursive median filters," *IEEE Trans. Inform. Theory*, vol. IT-32, Mar. 1986.

[6] G. R. Arce and N. C. Gallagher, "Stochastic analysis of the recursive median filter process," *IEEE Trans. Inform. Theory*, vol. 34, July 1988.

[7] Y. Han, I. Song, and Y. Park, "Some root properties of recursive weighted median filters," *Signal Process.*, vol. 25, 1991.

[8] Y. Han and I. Song, "Some useful weights of recursive weighted median filters," *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 1, June 1991.

[9] A. Antoniou, *Digital Filter. Analysis, Design and Applications*, New York: McGraw-Hill, 1993.

[10] J. Fitch, E. Coyle, and N. Gallagher, "Median filtering by threshold decomposition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, Dec. 1984.

[11] J. L. Paredes and G. R. Arce, "Stack filters, stack smoothers, and mirrored theshold decomposition," *IEEE Trans. Signal Processing*, vol. 47, pp. 2757–2767, Oct. 1999.

[12] J. J. Shynk, "Adaptive IIR filtering," *IEEE Acost., Speech, Signal Processing Mag.*, pp. 4–21, Apr. 1989.

[13] S.-J. Ko and Y. H. Lee, "Center weighted median filters and their applications to image enhancement," *IEEE Trans. Circuits Syst.*, vol. 38, Sept. 1991.

[14] C. Nikias and M. Shao, *Signal Processing with Alpha-Stable Distributions and Applications*. New York: Wiley, 1995.

Gonzalo R. Arce (F'00) received the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN.

He then joined the Department of Electrical and Computer Engineering at the University of Delaware, Newark, where he is currently Professor and Interim Chair. He is a frequent consultant to federal and industrial research centers. His research interests include nonlinear signal processing and its applications, wireless communications, multimedia communications and security, and electronic imaging. He holds three U.S. patents.

Dr. Arce has served as Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING and as Guest Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING. He is a member of the Digital Signal Processing Technical Committee of the IEEE Circuits and Systems Society. He is currently a Guest Editor of *Optics Express* of the Optical Society of America, and a Founding Member of the Nonlinear Signal and Image Processing Board. He has served as Chair, Co-Chair, and in the advisory committee of several conferences in nonlinear signal processing. He is a Fellow of the Center for Advanced Studies at the University of Delaware.

José L. Paredes was born in Mérida, Venezuela. He received the Diploma in electrical engineering from the Universidad de Los Andes, Mérida, in 1995, being the first student to graduate summa cum laude from his department. In 1999, he received the M.S. degree in electrical engineering from the University of Delaware, Newark, where he is currently pursuing the Ph.D. degree.

From 1995 to 1996, he was an Assistant Professor with the Electrical Engineering Department, Universidad de Los Andes. Since September 1996, he has been a Research Assistant with the Department of Electrical and Computer Engineering, University of Delaware. He has consulted with industry in the areas of digital communication and signal processing. His research interests include robust signal and image processing, nonlinear filter theory, adaptive signal processing, image compression, and digital communications.

Mr. Paredes is the recipient of several awards. In 1997, he won the Best Paper award from the U.S. Army's ATIRP Federated Laboratory Consortium. In 1995, he received several medals and awards for being the first electrical engineering student to graduate from the Universidad de Los Andes with the highest honors. He also was awarded three scholarships from the Government of Venezuela (in 1988, 1992, and 1996).