the contour does not need to be the whole circle, but can be any arc of it. The whole range of the unit circle has to be calculated by the FFT (the undesired range is redundant and wasted in the conventional FFT method).

The conventional Wigner distribution, like the FFT, is intrinsically a global baseband analysis tool. The Wigner spectrum is always computed from DC to the folding frequency of the whole distribution. In many applications, such as speech signal processing and acoustic signature extraction, spectral analysis is concerned over a narrow band of frequencies away from baseband. Although sampling period and sample points can be increased in the FFT to achieve sufficient spectral resolution, this will destroy the short-time stationary assumption of the signal, and also greatly increase the computation time. The proposed CZT approach can select any desired frequency range and choose arbitrary resolution for signal analysis without the above limitations.

IV. Conclusions

The pseudo-Wigner distribution with the chirp Z transform has been studied as a useful tool for the analysis of time-varying signals. It is capable of zooming onto any desired frequency range of interest. This results in more reliable and accurate spectral analysis on the time-frequency plane.

References

[1] T. A. C. M. Claasen and W. F. G. Mecklenbräuker, "The Wigner distribution—a tool for time-frequency signal analysis. Part I: Continuous-time signals; Part II: Discrete-time signals; Part III: Relation with other time-frequency signal transformations," *Phillips J. Res.*, vol. 35, pp. 217–250, pp. 276–300, and pp. 372–389, 1980.

[2] S.-C. Pei and I.-I. Yang, "Additional useful properties for the Wigner distribution," in *Proc. Telecommun. Symp.* (Taipei, Taiwan), Dec. 1989, pp. 482–487.

[3] L. R. Rabiner, R. W. Schafer, and C. M. Radar, "The chirp Z-transform algorithm and its applications," *Bell Syst. Tech. J.*, vol. 48, no. 5, pp. 1249–1291, May–June 1969.

[4] L. R. Rabiner, R. W. Schafer, and C. M. Radar, "The chirp Z-transform algorithm," *IEEE Trans. Audio Electroacoust.*, vol. AU-17, no. 2, pp. 86–92, June 1969.

[5] C. P. Janse and A. J. M. Kaizer, "Time-frequency distributions of loudspeakers: The application of the Wigner distribution," *J. Audio Eng. Soc.*, vol. 31, no. 4, pp. 198–222, 1983.

[6] F. Hlawatsch, "Interference terms in the Wigner distribution," in *Digital Signal Processing-84*, V. Cappellini and A. G. Constantinides, Eds. North-Holland, 1984, pp. 363–367.

Simplified Gradient Calculation in Adaptive IIR Lattice Filters

Jose A. Rodriguez-Fonollosa and Enrique Masgrau

***Abstract*—A simplified algorithm is presented for computing the gradient in adaptive IIR lattice filters. For a filter with N zeros and N poles, this algorithm requires only order N computations and it is shown to be completely equivalent to the previously proposed formulation which has a complexity of order N^2.**

Manuscript received May 7, 1990; revised October 4, 1990. This work was supported by the Spanish Government under Grant PRONTIC 105/88.

The authors are with ETSE Telecomunicació, Department of Signal Theory and Communications, Universitat Politècnica de Catalunya, 08080 Barcelona, Spain.

IEEE Log Number 9144711.

I. Introduction

The lattice realization of adaptive IIR filters has been shown to be an important alternative to the direct form [1]–[4]. It is well known that lattice forms have excellent finite-precision properties. Furthermore, in adaptive applications, its primary advantage is that stability can be controlled with essentially no computation, while in the direct form, stability monitoring is either computationally expensive or too restrictive in the coefficient space [3].

The complexity of the gradient has always been the major drawback of the lattice realization. In the direct form it is possible to calculate the gradient with a simple algorithm of order N, but in the lattice form, the formulation originally proposed by [1] requires order N^2 computations. Nevertheless, in this correspondence it is shown that the lattice form gradient can also be calculated with a simplified algorithm, thus making this structure more competitive with the direct or other IIR realizations.

A first effort to simplify this result was reported in [5], where a simplified algorithm was also derived, but it was based on an approximation which was shown to cause convergence problems. The algorithm we have derived does not neglect any term, and, therefore, it is completely equivalent to the gradient proposed in [1], although it has only a complexity of order N.

II. IIR Lattice Filter

The adaptive IIR lattice filter (Fig. 1) consists of a feedback lattice structure characterized by the reflection coefficients $\{f_i(n)\}$ and a feedforward structure characterized by the coefficients $\{v_i(n)\}$.

The forward and backward residuals of each stage are calculated as

$$f_{N+1}(n) = x(n) \tag{1a}$$

for $i = , \cdots , N$ do

$$f_i(n) = f_{i+1}(n) - k_i(n)\, b_i(n) \tag{1b}$$

$$b_{i+1}(n+1) = b_i(n) + k_i(n)\, f_i(n) \tag{1c}$$

end do

$$b_1(n+1) = f_1(n) \tag{1d}$$

where $x(n)$ is the filter input.

Then, the output $y(n)$ is obtained as the sum of the backward residuals weighted by the feedforward coefficients

$$y(n) = \sum_{i=1}^{N+1} b_i(n+1)\, v_i(n). \tag{2}$$

A. Gradient

The gradient vector is formed with the derivative of the output $y(n)$ with respect to the coefficients:

$$\theta_i(n) = \frac{\partial y(n)}{\partial v_i(n)} \qquad \psi_i(n) = \frac{\partial y(n)}{\partial k_i(n)}. \tag{3}$$

The gradient components for the feedforward coefficients are simply computed as

$$\theta_i(n) = b_i(n+1) \tag{4}$$

but the components that correspond to the reflection coefficients need a recursive method [1]. This method computes the auxiliary

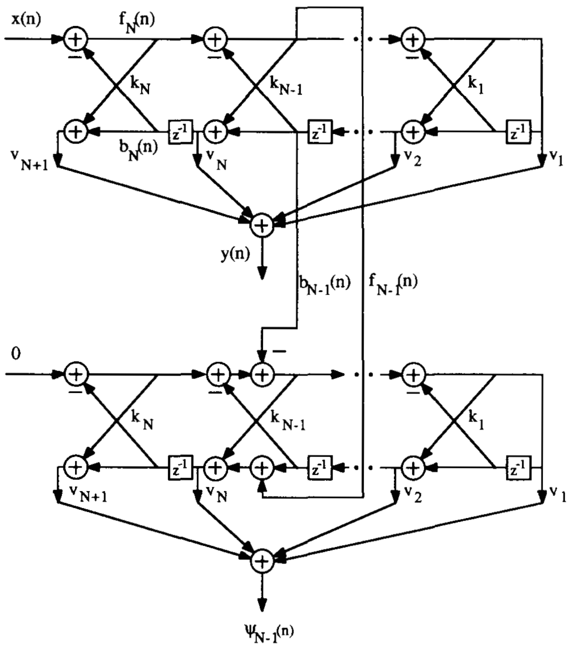

Fig. 1. The IIR lattice filter including one of the additional lattice structures required to compute the gradient with the previous method.

gradients

$$\phi_{ji}(n) \equiv \frac{\partial f_i(n)}{\partial k_j(n)} \qquad \beta_{ji}(n) \equiv \frac{\partial b_i(n)}{\partial k_j(n)}$$

for each i and j using the expressions

$$\phi_{ji}(n) = \phi_{j,i+1}(n) - k_i(n)\,\beta_{ji}(n) - \delta_{ji} b_i(n) \tag{5a}$$

$$\beta_{j,i+1}(n+1) = \beta_{ji}(n) + k_i(n)\,\phi_{ji}(n) + \delta_{ji} f_i(n) \tag{5b}$$

that require the boundary conditions $\beta_{j0}(n+1) = \phi_{j0}(n)$ and $\phi_{j,N+1} = 0$. Then, the gradients $\psi_j(n)$ are computed as

$$\psi_j(n) = \sum_{i=1}^{N+1} \beta_{ji}(n+1)\, v_i(n).$$

This recursive computation of each component for a reflection coefficient corresponds to the equations of a separate lattice filter whose inputs are the forward and backward residuals of the adaptive lattice (Fig. 1). Following this approach, N additional lattice filters are required to compute all the components, and, therefore, the complexity of this method is of order N^2.

B. New Algorithm

As it can be observed in Fig. 1, the gradient component $\psi_i(n)$ is the output of a time-varying filter whose input is $x(n)$. In the following, in order to work with transfer functions, we will make the assumption that the coefficients adapt slowly. This approximation was, in fact, used to derive the previous recursive gradient equations (5) and it introduces essentially no degradation in the performance of adaptive IIR filters [3], [6].

The derivation of the proposed algorithm starts expressing the transfer function $\Psi_i(z)$ of the above mentioned filter as the combination of two terms (Fig. 1)

$$\Psi_i(z) = F_i(z)\, G_i(z) - B_i(z)\, E_i(z) \tag{6}$$

where $F_i(z)$ and $B_i(z)$ are the transfer functions from the input to the upper and lower outputs of the first lattice, respectively, and $E_i(z)$ and $G_i(z)$ are the transfer functions of the second lattice that go, respectively, from the upper and lower input to the output. In the rest of this section, it is shown that there is a relation between the two terms of $\Psi_i(z)$ and the terms of $\Psi_{i-1}(z)$ and $\Psi_{i+1}(z)$. This relation allows a recursive computation of these terms and avoids the necessity of using a separate lattice structure for each gradient $\psi_i(n)$.

It can easily be derived from (1) that

$$F_i(z) = F_{i+1}(z) - k_i B_i(z) \tag{7a}$$

$$zB_{i+1}(z) = k_i F_i(z) + B_i(z) \tag{7b}$$

with $F_{N+1}(z) = 1$ and $zB_1(z) = F_1(z)$. It can also be shown from the transposed IIR lattice structure that

$$E_{i+1}(z) = E_i(z) + k_{i+1} G_{i+1}(z) \tag{8a}$$

$$zG_i(z) = -k_{i+1} E_{i+1}(z) + G_{i+1}(z) + z v_{i+1} \tag{8b}$$

with $E_N(z) = H(z)$ and $G_N(z) = v_{N+1}$, and where $H(z)$ is the transfer function of the adaptive lattice filter.

Multiplying (7) by $E_i(z)$ and $G_i(z)$, and (8) by $F_i(z)$ and $B_i(z)$, we obtain the following eight equations (Fig. 2):

$$F_i(z)\, E_i(z) = F_{i+1}(z)\, E_i(z) - k_i B_i(z)\, E_i(z) \tag{9a}$$

$$zB_{i+1}(z)\, E_i(z) = k_i F_i(z)\, E_i(z) + B_i(z)\, E_i(z) \tag{9b}$$

$$F_i(z)\, G_i(z) = F_{i+1}(z)\, G_i(z) - k_i B_i(z)\, G_i(z) \tag{9c}$$

$$zB_{i+1}(z)\, G_i(z) = k_i F_i(z)\, G_i(z) + B_i(z)\, G_i(z) \tag{9d}$$

$$F_{i+1}(z)\, E_{i+1}(z) = F_{i+1}(z)\, E_i(z) + k_{i+1} F_{i+1}(z)\, G_{i+1}(z) \tag{9e}$$

$$zF_{i+1}(z)\, G_i(z) = -k_{i+1} F_{i+1}(z)\, E_{i+1}(z) + F_{i+1}(z)\, G_{i+1}(z) + z v_{i+1} F_{i+1}(z) \tag{9f}$$

$$B_{i+1}(z)\, E_{i+1}(z) = B_{i+1}(z)\, E_i(z) + k_{i+1} B_{i+1}(z)\, G_{i+1}(z) \tag{9g}$$

$$zB_{i+1}(z)\, G_i(z) = -k_{i+1} B_{i+1}(z)\, E_{i+1}(z) + B_{i+1}(z)\, G_{i+1}(z) + z v_{i+1} B_{i+1}(z) \tag{9h}$$

and the boundary conditions

$$F_{N+1}(z)\, E_N(z) = H(z)$$

$$F_{N+1}(z)\, G_N(z) = v_{N+1}$$

$$zB_1(z)\, E_1(z) = F_1(z)\, E_1(z)$$

$$zB_1(z)\, G_1(z) = F_1(z)\, G_1(z).$$

Note that in Fig. 2, we have represented the above equations in the time domain. For example, $f_i e_i(n)$ corresponds to $F_i(z)\, E_i(z)$, i.e., $f_i e_i(n)$ is the output of a filter whose transfer function is $F_i(z)\, E_i(z)$ and its input $x(n)$.

As Fig. 2 illustrates, these equations cannot be used as they appear above, but they can be rearranged in a computable way. In fact, several combinations are possible, and in Fig. 3 we have represented one of them. The resulting algorithm, in the time domain, is the following:

$$b_1 e_1(n) = f_1 e_1(n-1) \tag{10a}$$

$$b_1 g_1(n) = f_1 g_1(n-1) \tag{10b}$$

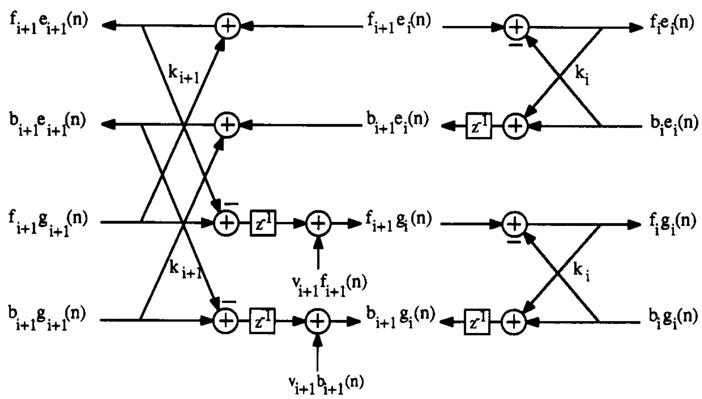

Fig. 2. Graph illustrating the equations (9) that relate the terms involved in the gradient computation of the proposed method.

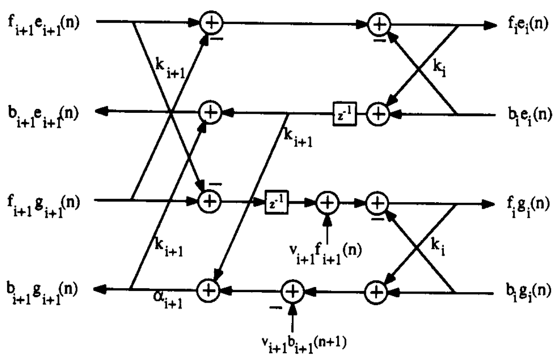

Fig. 3. Graph illustrating the proposed algorithm for calculating the gradient.

for $i = 1, \cdots, N-1$ do

$$b_{i+1}e_i(n) = k_i(n)f_ie_i(n-1) + b_ie_i(n-1) \tag{11a}$$

$$f_{i+1}g_i(n) = -k_{i+1}(n)f_{i+1}e_{i+1}(n-1) + f_{i+1}g_{i+1}(n-1) + v_{i+1}(n)f_{i+1}(n) \tag{11b}$$

$$f_ig_i(n) = f_{i+1}g_i(n) - k_i(n)b_ig_i(n) \tag{11c}$$

$$\psi_i(n) = f_ig_i(n) - b_ie_i(n) \tag{11d}$$

$$b_{i+1}g_i(n+1) = k_i(n)f_ig_i(n) + b_ig_i(n) \tag{11e}$$

$$\alpha_{i+1} = 1/(1 - k_{i+1}^2(n)) \tag{11f}$$

$$b_{i+1}g_{i+1}(n) = \alpha_{i+1}(b_{i+1}g_i(n+1) - v_{i+1}(n)b_{i+1}(n+1) + k_{i+1}(n)b_{i+1}e_i(n)) \tag{11g}$$

$$b_{i+1}e_{i+1}(n) = b_{i+1}e_i(n) + k_{i+1}(n)b_{i+1}g_{i+1}(n) \tag{11h}$$

end do

$$f_Ng_N(n) = v_{N+1}(n)x(n) - k_N(n)b_Ng_N(n) \tag{12a}$$

$$\psi_N(n) = f_Ng_N(n) - b_Ne_N(n) \tag{12b}$$

$$f_Ne_N(n) = y(n) - k_N(n)b_Ne_N(n) \tag{12c}$$

for $i = N-1, \cdots, 1$ do

$$f_{i+1}e_i(n) = f_{i+1}e_{i+1}(n) - k_{i+1}(n)f_{i+1}g_{i+1}(n) \tag{13a}$$

$$f_ie_i(n) = f_{i+1}e_i(n) - k_i(n)b_ie_i(n) \tag{13b}$$

end do

where (11g) is the result of combining (9g) and (9h).

Observe that, apart from the computation of α_i, only $5N - 1$ multiplications per iteration are required to obtain the gradient components, while the previous algorithm requires $3N^2 + N$. The major drawback of the proposed algorithm is the computation of α_i from k_i because it requires $N - 1$ multiplications and divisions.

III. Comments

The proposed order N formulation is nothing more than an efficient structure for the realization of the filters that compute the gradient. Therefore, if the coefficients were kept fixed, the gradient computed with the proposed algorithm would be the same as the one given by the previous order N^2 formulation.

When the coefficients are adapted, the computed values will not be exactly the same, nevertheless, no differences are expected in the behavior of the adaptive IIR algorithms using the conventional gradient computation and those using the proposed formulation. In fact, it is known that both algorithms can be associated to the same ordinary differential equation (ODE) [6], which governs their convergence properties.

We have performed several computer simulations to compare the performance of the proposed $O(N)$ formulation and the conventional $O(N^2)$ formulation, and we have found no difference in their behavior. In these simulations we used the full Hessian and the diagonal Hessian adaptive algorithms [4] in a system-identification configuration.

IV. Conclusions

An efficient method has been derived to compute the gradient in adaptive IIR lattice filters. It reduces the complexity of the previous algorithm by an order of magnitude and allows the development of robust adaptive IIR algorithms with a cost proportional to the filter order.

References

[1] D. Parikh, N. Ahmed, and S. D. Stearns, "An adaptive lattice algorithm for recursive filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, no. 1, pp. 110–111, Feb. 1980.

[2] J. J. Shynk, "On lattice-form algorithms for adaptive IIR filtering," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Apr. 1988, pp. 1554–1557.

[3] J. J. Shynk, "Adaptive IIR filtering," *IEEE ASSP Mag.*, pp. 4–21, Apr. 1989.

[4] J. A. Rodríguez-Fonollosa and E. Masgrau, "Spectral sensitivity and convergence rate in adaptive IIR filtering," in *Proc. Int. Symp. Circuits Syst.*, May 1990, pp. 1959–1962.
[5] J. L. Ayala, "On a new adaptive lattice algorithm for recursive filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-30, no. 2, pp. 316–319, Apr. 1982.
[6] L. Ljung, "Analysis of recursive stochastic algorithms," *IEEE Trans. Automat. Contr.*, vol. AC-22, pp. 551–575, Aug. 1977.

NFIR Nonlinear Filter

Xin Wang

***Abstract*—A new nonlinear filter is introduced in this correspondence. As the output of this filter is nearest in distance to the output of a FIR linear phase filter, we call it the NFIR filter. After choosing the impulse response of the FIR filter properly, the NFIR filter can clean impulsive noise and preserve edges of signals. As a special case of the NFIR filter, a nearest mean (NEME) filter is derived. Theoretical analysis shows that the NEME filter is better than the median filter in smoothing additive Gaussian noise. In addition, based on the definition of the NEME filter, a new interpretation of the low-pass property of the median filter with window width 3 is described. Also, the performance of the nearest mean filter on a noisy edge is quantitatively compared with that of the median and mean filters.**

I. Introduction

As our visual perception is heavily based on edge information, a number of nonlinear filters have been presented [1]. They attempt to remove the effects of noise while retaining the edges of signals.

The median filter is one well-known method. Its simplicity and edge preservation characteristics made it attractive in speech and image processing. Indeed, the median filter is used in an ever-growing variety of applications. Many authors have analyzed the statistical and root signal properties of the median filter [2]–[5].

Though the median filter can preserve edges, it is not good at smoothing Gaussian noise. A worthwhile goal is to find some new nonlinear filters which are better than the median filter in smoothing Gaussian noise.

Based on the median filter, some novel filters have been proposed. Bednar and Watt [6] presented an alpha-trimmed filter. Heinonen and Neuvo [7] introduced a class of FIR-median hybrid filters. Recently, Coyle *et al.* [8] presented an algorithm for optimal stack filtering.

Based on the mean filter, Kundu *et al.* [9] proposed a generalized mean filter for the removal of impulsive noise. Pitas and Venetsanopulos [10] presented a nonlinear mean filter. According to the varied nonlinear functions, the output of this filter can be the arithmetic mean, harmonic mean, geometric mean, or Lp mean. They can smooth Gaussian noise more effectively than the median filter but cannot simultaneously remove both positive and negative spikes.

In general, all the nonlinear filters can be divided into two classes, A and B, which are described as follows:

Let set X and Y consist of the discrete input signals $x(n)$ and the output signals $y(n)$, respectively. If $Y \subseteq X$, the filter belongs to class A; otherwise the filter belongs to class B. The median filter and nonlinear mean filter are two representatives of these two classes.

Manuscript received October 14, 1989; revised June 22, 1990.

The author is with the Department of Electrical and Computer Engineering, Shandong Polytechnic University, Jinan 250014, People's Republic of China.

IEEE Log Number 9144718.

In this correspondence, a NFIR nonlinear filter is introduced. It can clean impulsive noise and preserve edges effectively.

The statistical properties of this filter are analyzed for input signals with Gaussian density function. As a special case of the NFIR nonlinear filter, a NEME filter is derived.

Theoretical analysis shows that the NEME filter is the best one from class A in smoothing Gaussian noise.

II. Definition of the NFIR Filter

Let $x(n)$ $(-\infty < n < \infty)$ be a discrete input sequence with additive noise, where n is integer value. One purpose is to smooth noise without smearing the edges of signals.

A simple method for smoothing noise is to let the noisy signal $x(n)$ be filtered by a FIR filter.

Let $h(n)$ $(-N \le n \le N)$ be the impulse response of a linear phase FIR filter. Suppose that $h(k) = h(-k)$ $(1 \le k \le N)$ and $h(k) \ge 0$ $(-N \le k \le N)$. The output of the FIR filter at position n is

$$\mu(n) = \sum_{K=-N}^{N} h(k)\, x(n-k). \tag{1}$$

After applying the FIR filter to the input signal $x(n)$, the edges of signals would be blurred at the output. In addition, as impulsive noise consists of very large positive or negative spikes of short duration, the linear FIR filtering is not a useful tool in impulse noise removal. However, this disadvantage can be avoided by using a nonlinear operation on the output sequence of the FIR filter.

Consider a window with length $2M + 1$ and $x(n)$ is the central sample, where $0 < M \le N$. If

$$|x(u) - \mu(n)| = \min_{n-M \le m \le n+M} |x(m) - \mu(n)|. \tag{2}$$

Then the output of the new filter is defined as

$$y(n) = x(u). \tag{3}$$

Since the output of this filter is nearest in distance to the output of the FIR filter, we call it the NFIR filter.

If $M = 0$ in (2), then $y(n) = x(n)$. In this case, the output of the NFIR filter is equal to the input sequence $x(n)$.

With different values of $h(n)$, the NFIR filter has varied properties. Our attempt is to determine the impulse response $h(n)$ to make the NFIR filter clean impulsive noise and preserve edges of signals. For this purpose some constraints on $h(n)$ are derived below:

$$\text{Condition 1:} \quad \sum_{k=-N}^{N} h(k) = 1. \tag{4}$$

If the input $x(n) = d$ $(-\infty < n < \infty)$, condition 1 makes the output $\mu(n) = d$.

$$\text{Condition 2:} \quad h(0) < 1/2. \tag{5}$$

This condition enables the NFIR filter to clean impulsive noise. To see this, suppose there is an impulse at position n: then $x(n) = d$ and $x(m) = 0$ $(m \ne n)$. From (1), we have $\mu(n) = h(0)d$. According to $h(0) < 1/2$ and the definition of the NFIR filter, we obtain $y(n) = 0$. Thus, condition 2 is proven.

$$\text{Condition 3:} \quad \sum_{k=-N}^{-1} h(k) < 1/2 \quad \text{and} \quad \sum_{k=-N}^{0} h(k) > 1/2. \tag{6}$$

With condition 3, the NFIR filter can preserve edges of signals.