Correspondence

A Modified Algorithm for Constrained Least Square Design of Multiband FIR Filters Without Specified Transition Bands

Ivan W. Selesnick, Markus Lang, and C. Sidney Burrus

Abstract—**In a previous paper, we described a constrained least square approach to FIR filter design that does not use "don't care" regions. In that paper, we described a simple algorithm for the design of lowpass filters according to that approach. In this correspondence, we describe a modification of that algorithm that makes it converge for many multiband filter designs. Although no proof of convergence is given, the modified algorithm remains simple and converges rapidly in many cases.**

In this approach, the user supplies a lower and upper bound constraint that is exactly satisfied by the local minima and maxima of the frequency response amplitude. Yet, the constraints can be made as tight as desired—the transition band automatically adjusts (widens) to accommodate the constraints.

Index Terms—**Bandpass filers, Chebyshev approximation, digital filters, FIR digital filters, least squares methods, linear-phase filters.**

I. Introduction

In a previous paper [4], we described a constrained least square approach to FIR filter design that does not use "don't care" regions and described a simple multiple-exchange algorithm for the design of *lowpass* (and highpass) linear phase FIR filters according to this approach. Unfortunately, when applied to the design of *multiband* filters (bandpass, bandstop, etc.), the same algorithm does not converge reliably. In this correspondence, we describe a modification to that algorithm that makes it converge for many multiband filter designs. Although no proof of convergence is given, the modified algorithm remains simple and converges rapidly in many cases.

The approach taken in [4] follows the work on constrained least square filter design by Adams *et al.* [1], [2] and is motivated in part by a paper on error criteria by Weisburn, *et al.* [5]. The algorithm described in this correspondence modifies the algorithm of [4] so that it saves the constraint set of the previous iteration: a concept described previously in [2].

II. Discussion

As stated above, the algorithm of [4] may not converge when it is applied to *bandpass* filter design. In these cases, the failure of the algorithm to converge takes a specific form. Instead of converging to a single filter, the algorithm will eventually cycle between two different filters, neither of which satisfy the specified peak gain constraints. The following example illustrates the way in which the algorithm of [4] may fail when it is used to design a length 63 bandpass filter.

Manuscript received October 29, 1996; revised June 5, 1997. This work was supported by NORTEL under NSF Grant MIP-9316588 and by the Alexander von Humboldt Foundation. The associate editor coordinating the review of this paper and approving it for publication was Dr. Sawasd Tantaratana.

I. W. Selesnick is with Polytechnic University, Brooklyn, NY 11201 USA (e-mail: selesi@radar.poly.edu).

M. Lang is with Me Vis, Bremen, Germany.

C. S. Burrus is with Rice University, Houston, TX 77001 USA.

Publisher Item Identifier S 1053-587X(98)01351-8.

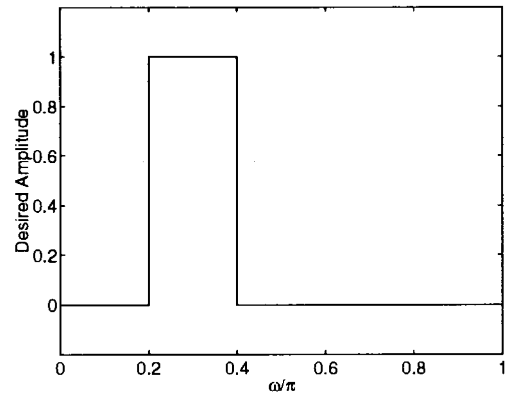

Fig. 1. Desired amplitude of an ideal bandpass filter.

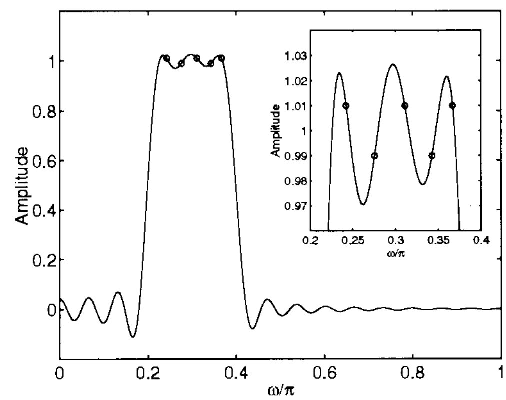

Fig. 2. Algorithm of [4] applied to bandpass filter design. Even iterations after several iterations.

Consider the design of a bandpass filter with cut-off frequencies at $\omega_1 = 0.2\pi$ and $\omega_2 = 0.4\pi$. The ideal frequency response amplitude $D(\omega)$ is shown in Fig. 1. Further, suppose that at the local maxima and minima of $A(\omega)$ in the passband, $A(\omega)$ is required to lie between 0.99 and 1.01. To simplify this illustrative example, the peak errors in the stopbands are not required to meet any ripple size constraints. When the algorithm of [4] is applied to the design of a length 63 bandpass filter with these constraints and cut-off frequencies, it fails to converge. After several iterations, that algorithm will cycle between the two filters shown in Figs. 2 and 3.

The algorithm of [4] employs a constraint set (a set of interpolation points) and proceeds as follows. On each iteration, 1) the set of interpolation points is updated and 2) the least square error filter satisfying the interpolation constraints is found. The set of interpolation points is updated from one iteration to the next by setting it equal to the set of local minima and maxima where the new frequency response amplitude violates the lower and upper bound constraints.

In this example, the exchange algorithm of [4] fails to converge because on some iteration, the new set of interpolation points is the same as a previous set. This is made clear in Figs. 2 and 3, where it can be seen that the extremal points in one figure are the interpolation points in the other figure.

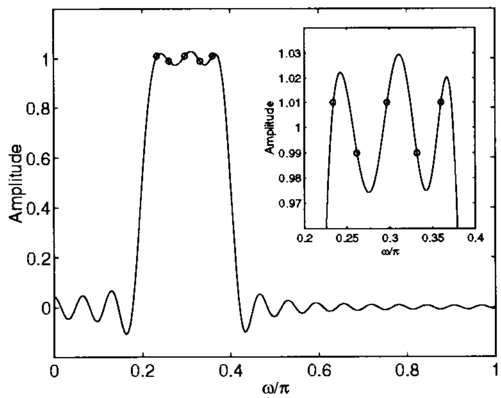

Fig. 3. Algorithm of [4] applied to bandpass filter design. Odd iterations after several iterations.

III. New Algorithm

The modified algorithm remains simple, and although we have not proven its convergence, it converged for all examples with which it was tested and, for most of those examples, converged rapidly. Like the algorithm of [4], the new algorithm is a multiple-exchange algorithm that uses Lagrange multipliers and the Kuhn–Tucker conditions on each iteration. It also gives the best L_2 filter and a continuum of Chebyshev filters as special cases. The algorithm is similar to that described in [4]; however, it employs an additional inner loop.

To describe the algorithm for multiband filter design, let $\omega_1, \cdots, \omega_K$ be the cut-off frequencies and $m_0, \cdots, m_K$ be the magnitudes of a $K+1$ band filter whose desired frequency response amplitude is given by

$$D(\omega) = \begin{cases} m_0, & \text{for } 0 \le \omega < \omega_1 \\ m_k, & \text{for } \omega_k \le \omega < \omega_{k+1} \\ m_K, & \text{for } \omega_K \le \omega < \pi. \end{cases} \quad (1)$$

For a bandpass filter, for example, we might have $\omega_1 = 0.2\pi$, $\omega_2 = 0.4\pi$, $m_0 = 0$, $m_1 = 1$, and $m_2 = 0$. In this case, $D(\omega)$ is shown in Fig. 1.

Let the lower and upper bound functions $L(\omega)$ and $U(\omega)$ be specified by the user such that they are constant within each band[1] and satisfy the following.

1) $L(\omega) \le D(\omega)$.
2) $U(\omega) \ge D(\omega)$.
3) $U(\omega) > L(\omega)$.

As in [4], we propose that the integral square error be minimized such that the local minima and maxima of $A(\omega)$ lie within the lower and upper bound functions $L(\omega)$ and $U(\omega)$.

The amplitude $A(\omega)$ of the filter minimizing the L_2 error subject to these constraints will touch the lower and upper bound functions at certain extremal frequencies of $A(\omega)$. [By extremal frequencies of $A(\omega)$, we mean local minima and maxima of $A(\omega)$]. If these frequencies were known in advance, then the filter could be found by minimizing $\|E\|_2^2$ subject to equality constraints (interpolation requirements) at these frequencies. The iterative procedure below determines these frequencies by updating a set of candidate frequencies (a constraint set). Each filter in this iterative procedure is found by minimizing the L_2 error subject to interpolation requirements at these candidate frequencies. The sets of candidate frequencies will be called "constraint sets" because constraints will be imposed on $A(\omega)$ at these frequencies.

[1] It is not necessary that $L(\omega)$ and $U(\omega)$ be constant in each band, but they should be smooth. It is assumed that they are constant within each band for simplicity.

On each iteration, the constraint set is updated so that at convergence, the only frequency points at which equality constraints are imposed are those where $A(\omega)$ touches the constraint. The equality constrained problem is solved with Lagrange multipliers. The equality constraints are the same as those of [4] and give rise to the same equations as those given in that paper.

A. The Modified Exchange Iterations

The equality constrained optimization procedure described in [4] is performed at each step of an iterative algorithm. The way in which the constraint set S is updated is now described. This part of the algorithm differs from the algorithm of [4]. To avoid the cycling that may occur when the algorithm of [4] is applied to multiband filter design, two constraint sets are used. The second constraint set, which we call R, is used to store the elements of the constraint set S of the previous iteration of the algorithm.

After each iteration, the algorithm checks the values of $A(\omega)$ over the previous constraint set frequencies. If $A(\omega)$ is within the lower and upper boundary functions $L(\omega)$ and $U(\omega)$ over these frequencies, then the algorithm proceeds exactly as does the algorithm of [4]. However, if it is found that $A(\omega)$ violates the constraints at some frequency belonging to the previous constraint set, then i) that frequency where the violation is greatest is appended to the current constraint set S and ii) the same frequency is removed from the record of previous constraint set frequencies R.

The algorithm begins with an empty constraint set S so that the first filter designed is the best unconstrained L_2 filter. Then, constraints are iteratively imposed on $A(\omega)$ at selected frequencies until the best constrained L_2 filter is obtained.

The algorithm can be summarized in the following steps. In this description, the set R records the constraint frequencies of the previous iteration. The remaining notation is the same as that of [4].

1) **Initialization:** Initialize both constraint sets to the empty set: $R = \emptyset$, $S = \emptyset$.
2) **Minimization with Equality Constraints:** Calculate the Lagrange multipliers associated with the filter that minimizes $\|E(\omega)\|_2$ subject to the equality constraints $A(\omega_i) = L(\omega_i)$ for $\omega_i \in S_l$ and $A(\omega_i) = U(\omega_i)$ for $\omega_i \in S_u$. (Solve [4, (32)]).
3) **Kuhn–Tucker Conditions:** If there is a constraint set frequency $\omega_i \in S$ for which the Lagrange multiplier μ_i is negative, then remove from the constraint set S the frequency corresponding to the most negative multiplier, and go back to step 2). Otherwise, calculate the new cosine coefficients using (33) in [4], and proceed to step 4).
4) **Check for Violation Over $\boldsymbol{R}$**: Calculate $\mathbf{a}$ and $A(\omega)$. If $A(\omega_i) < L(\omega_i)$ or $A(\omega_i) > U(\omega_i)$ for some $\omega_i \in R$, then remove from R the frequency corresponding to the greatest violation, append to S the same frequency, and go back to step 2. Otherwise, proceed to step 5).
5) **Multiple Exchange of Constraint Set:** Set the constraint set S equal to $S_l \cup S_u$, where S_l is the set of frequency points ω_i in $[0, \pi]$ satisfying both $A'(\omega_i) = 0$ and $A(\omega_i) \le L(\omega_i)$, and where S_u is the set of frequency points ω_i in $[0, \pi]$ satisfying both $A'(\omega_i) = 0$ and $A(\omega_i) \ge U(\omega_i)$.
6) **Check for Convergence:** If $A(\omega) \ge L(\omega) - \epsilon$ for all frequency points in S_l and if $A(\omega) \le U(\omega) + \epsilon$ for all frequency points in S_u, then convergence has been achieved. Otherwise, go back to step 2).

According to the Kuhn–Tucker conditions, because $\boldsymbol{\mu} \ge \mathbf{0}$ is ensured for each set of computed cosine coefficients $\mathbf{a}$, each filter minimizes the L_2 error subject to the inequality constraints (19), (20) in [4] over some set of frequencies. At convergence, the constraint set

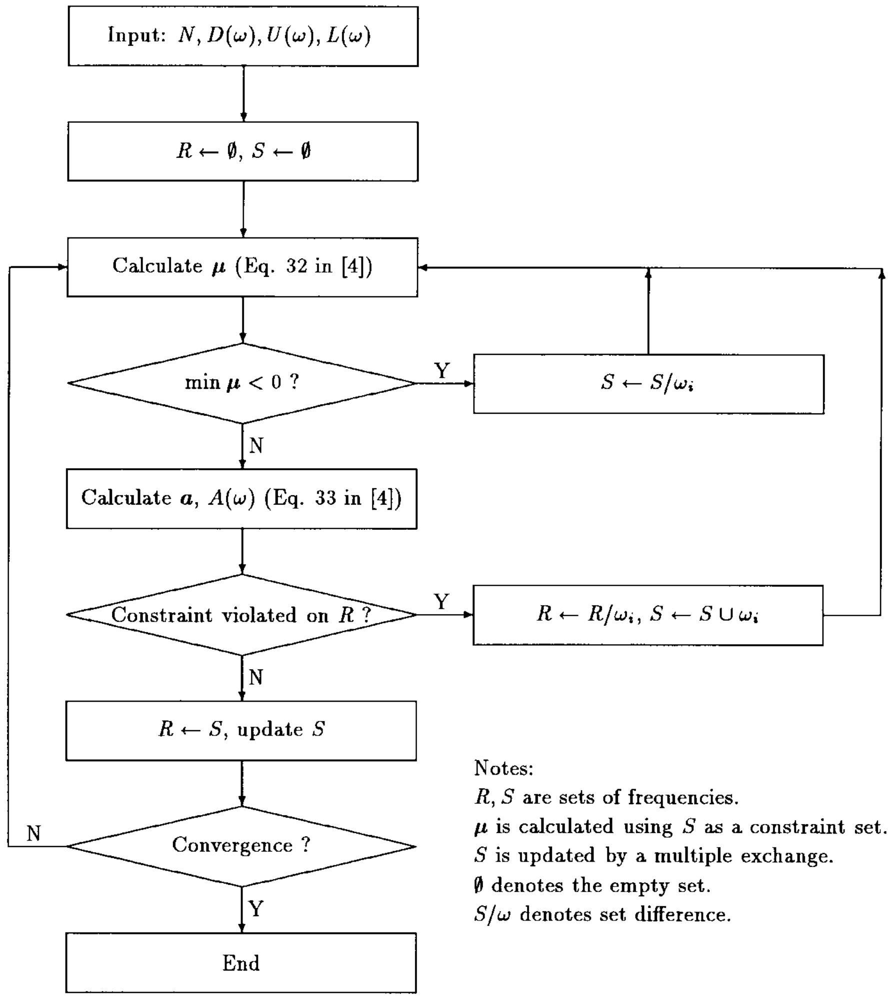

Fig. 4. Flowgraph for the exchange algorithm for the constrained least square design of multiband filters.

frequencies are exactly those extrema of $A(\omega)$, where $A(\omega)$ touches the lower and upper bound function. ϵ in step 4) is a small number (like 10^{-6}) indicating the numerical accuracy desired.

A flowgraph is shown in Fig. 4. Comparing this flowgraph with the flowgraph given in [4], it can be seen that the two algorithms are very similar. The only modification of the earlier algorithm is the test for constraint violation over the previous constraint set R. This approach was first discussed in [2]. Note, however, that when a frequency in R is appended to S, it is also removed from R. Therefore R diminishes in size. This is important because it ensures that eventually, the inner loop will terminate. In some cases, many of the frequencies of R are transferred to S, in which case, the new constraint set is essentially replaced by the previous one. If the frequencies transferred from R to S were not deleted from R, then there are cases where the inner loop will not terminate.

Notice that the algorithm begins by initializing the sets R and S to the empty set. Therefore, on the first iteration of the algorithm, steps 2)–4) are trivial: the set of Lagrange multipliers in step 2) is the null set. The algorithm essentially begins by calculating the best unconstrained least square solution in step 5).

This approach does not exclude from the integral square measure of approximation error any region around the cut-off frequencies; therefore, it does not implicitly assume that input signals have no energy in those regions. At the same time, the algorithm does not preclude the use of "don't care" regions if their use is desired—appropriate L_2 weighting functions can be incorporated if desired.

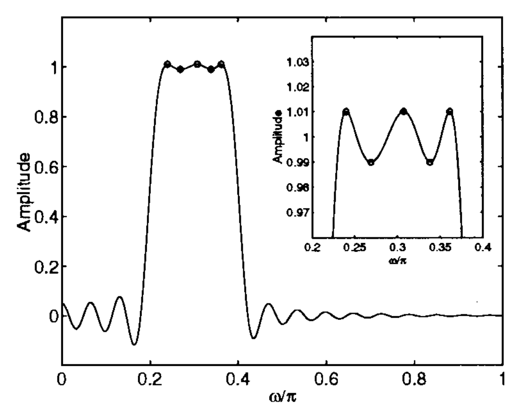

Fig. 5. Modified algorithm applied to bandpass filter design. The filter length is 63.

When implementing the algorithm described herein, the frequency response can be calculated over a grid of frequency values. However, it is advised that the location of the extremal frequencies be refined by Newton's method; otherwise, a rather dense grid is sometimes required for convergence. The use of Newton's method is easily incorporated.

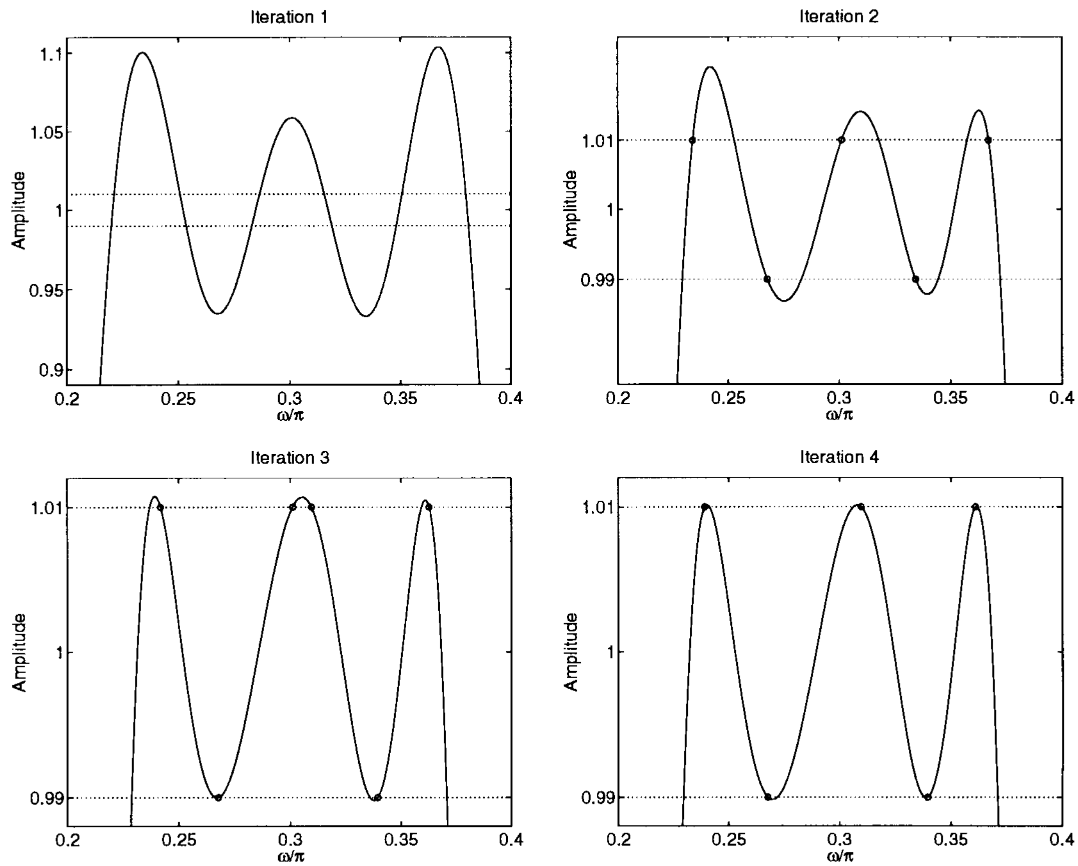

Fig. 6. First four (outer) iterations of the modified algorithm for the example.

IV. Example

When the modified algorithm is applied to the design problem described in the example above, the filter illustrated in Fig. 5 is obtained. The behavior of the algorithm is illustrated in Fig. 6, which shows the filter response for the first few (outer) iterations of the algorithm. The subfigures of Fig. 6 show the frequency response amplitude at the point in the algorithm at which convergence is tested. The circular marks in the figure indicate the interpolation points in the set S used to obtain the filter. Each amplitude shown in Fig. 5 is calculated after several *inner* loop iterations. The algorithm converges after several more iterations.

Notice that the middle local maximum on the third outer iteration shown in Fig. 6 is flanked by two interpolation points. The simpler algorithm of [4] cannot achieve this behavior. It is the ability of the modified algorithm to arrive at this type of constraint set that allows it to solve the multiband case.

An additional example illustrates a constrained least square filter where constraints have been imposed in the stopbands as well; see Fig. 7. The same ideal response as above is used and again, the filter length is 63. The local minima and maxima of the filter response amplitude are constrained to lie within -0.01 and 0.01.

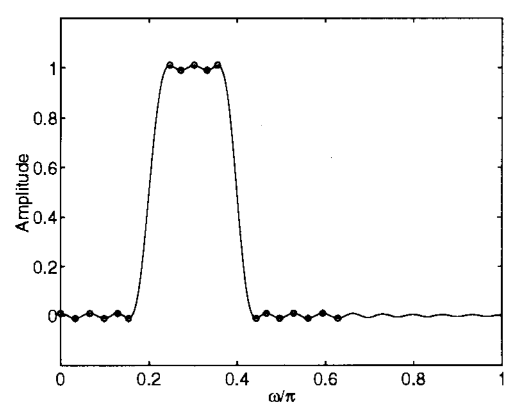

Fig. 7. Modified algorithm applied to bandpass filter design. Constraints in both passband and stopband. The filter length is 63.

V. Conclusions

This correspondence shows that there exists a simple and effective multiple-exchange algorithm for the design of multiband linear phase FIR filters according to a constrained least square approach that does not use specified transition bands. The user supplies a lower and upper bound constraint that is satisfied by the local minima and maxima of the frequency response amplitude. The constraints can be made as tight as desired—the transition band automatically adjusts (widens) to accommodate the constraints.

The algorithm of this correspondence gives a design approach that is a hybrid of the Parks–McClellan algorithm [3] and the method of least squares. Accordingly, the algorithm of this correspondence produces both least square filters and equiripple filters as special cases. A Matlab program is available on the World Wide Web at URL *http://www-dsp.rice.edu*.

References

[1] J. W. Adams, "FIR digital filters with least squares stop bands subject to peak-gain constraints," *IEEE Trans. Circuits Syst.*, vol. 39, pp. 376–388, Apr. 1991.

[2] J. W. Adams, J. L. Sullivan, R. Hashemi, C. Ghadimi, J. Franklin, and B. Tucker, "New approaches to constrained optimization of digital filters," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 1993, vol. 1, pp. 80–83.
[3] T. W. Parks and J. H. McClellan, "Chebyshev approximation for nonrecursive digital filters with linear phase," *IEEE Trans. Circuit Theory*, vol. CT-19, pp. 189–194, Mar. 1972.
[4] I. W. Selesnick, M. Lang, and C. S. Burrus, "Constrained least square design of FIR filters without specified transition bands," *IEEE Trans. Signal Processing*, vol. 44, pp. 1879–1892, Aug. 1996.
[5] B. A. Weisburn, T. W. Parks, and R. G. Shenoy, "Error criteria for filter design," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, Apr. 1994, vol. 3, pp. 565–568.

Fast CWT Computation at Integer Scales by the Generalized MRA Structure

K. C. Ho

***Abstract*—This correspondence proposes a fast algorithm for continuous wavelet transform (CWT) at linear scale without decimation by using the generalized multiresolution analysis (MRA) structure. The constraints required on the lowpass and bandpass filters in the generalized MRA structure are derived. A possible solution for the lowpass filters and a least-squares design of the bandpass filters are given. The computational complexity of the algorithm is $O(N)$ per scale, where N is the data length. The fast algorithm is verified by computer simulations.**

I. Introduction

The wavelet transform (WT) is a useful tool for nonstationary signal analysis. Unlike the short-time Fourier transform (STFT), which utilizes sine and cosine functions as basis to expand a signal in the $\boldsymbol{L}^2(\boldsymbol{R})$ space, WT uses basis functions that are dilations and translations of a single function known as the mother wavelet $\psi(t)$. Constant-Q analysis is a major advantage of WT. It has a high-frequency resolution at low frequency and a high time resolution at high frequency. This is in contrast with STFT, in which the time and frequency resolutions are fixed. WT has found wide applications in time-frequency analysis [1], transient detection [2], image processing [3] and speech processing [4].

The continuous WT (CWT) of a signal $s(t)$ is defined as

$$\mathrm{CWT}(a,\tau) = \frac{1}{\sqrt{a}} \int s(t)\psi^*\left(\frac{t-\tau}{a}\right) dt \tag{1}$$

where

- * complex conjugate;
- a scale;
- τ translation.

Direct evaluation of (1) is very computationally intensive. When dyadic sampling the scale at $a = 2^m$ and downsampling τ such that $\tau = 2^m n$, Shensa [5] has shown that under some conditions the multiresolution analysis (MRA) structure from Mallat [6] computes $\mathrm{CWT}(2^m, 2^m n)$. The conditions are 1) a proper initialization of the MRA input and 2) a satisfaction of the two-scale equations [7], [8] in the lowpass and bandpass filters. The computational complexity is only $O(N)$, where N is the data length. The algorithm can be extended to evaluate CWT without decimation. A review of fast CWT algorithms is in [7].

Manuscript received February 14, 1995; revised June 18, 1997. The associate editor coordinating the review of this paper and approving it for publication was Dr. Nurgun Erdol.

The author is with the Department of Electrical Engineering, University of Missouri-Columbia, Columbia, MO 65211 USA.

Publisher Item Identifier S 1053-587X(98)01343-9.

In some signal processing tasks, such as EEG analysis, dyadic sampling in scale is too coarse, and linear scale sampling is desired. In [9], a noniterative algorithm to compute linear-scale CWT without decimation was proposed. The required computation per scale is only $O(N)$. The method assumes that the input signal and the wavelet are splines of certain degrees.

This correspondence proposes another fast algorithm to compute linear-scale CWT without decimation. It is based on a generalized MRA structure that computes the transform coefficients iteratively. The structure is highly regular and can be implemented by recursion. No assumption on the input and the wavelet is required, although some approximation is necessary to find the bandpass filters in the generalized structure. Section II will introduce the proposed generalized MRA structure. The set of conditions on the structure such that it evaluates decimated CWT at linear scale will be derived. In the special case of dyadic sampling, the conditions reduce to those found in literature. The structure is then extended to compute linear-scale CWT without decimation. A method is proposed to further decrease the computation. Section III provides a possible solution to the lowpass filters and gives a least-squares design of the bandpass filters when a mother wavelet function is given. Section IV studies the computational complexity of the proposed algorithm. Section V presents simulation results to verify the computational structure and examine the accuracy. Section VI concludes the correspondence.

II. Relating the Generalized MRA and CWT

This study uses a frequency domain approach. Any continuous-time signal is assumed to be bandlimited to 0.5. Denote $\mathcal{F}\{*\}$ as the continuous time Fourier transform of $\{*\}$, and let f be a continuous frequency variable that takes values from $-\infty$ to ∞. Derivation for the conditions relating the generalized MRA and CWT requires only three basic identities in digital signal processing. They are summarized below for clarity [10]:

I1) If $\mathcal{F}\{y(t)\} = Y(f)$, sampling $y(t)$ at $t = l$ gives

$$\mathcal{F}\{y(l)\} = \sum_{k=-\infty}^{\infty} Y(f+k).$$

I2) If $\mathcal{F}\{y(l)\} = \hat{Y}(f)$, downsampling $y(l)$ by an integer a yields

$$\mathcal{F}\{y(an)\} = \frac{1}{a}\sum_{r-0}^{a-1} \hat{Y}\Big(f + \frac{r}{a}\Big).$$

.

I3) The noble identity shown in Fig. 1, where $\downarrow M$ represents downsampling by M and $T(f)$ is any transfer function.

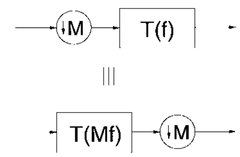

Fig. 1. Noble identity.