number of additions required for the minimum polynomial multiplication algorithm becomes excessive beyond length 2^4.

Table 1 COMPARISON OF DIFFERENT DFT (2^n; 2) ALGORITHMS WITH COMPLEX INPUTS

$N \times N$	Minimal algorithm M	Vector radix algorithm[7] M	Split vector radix algorithm[8] M
8×8	48	48	48
16×16	432	576	528
32×32	2544	4424	3696
64×64	12 528	25 344	21 648
128×128	56 304	136 704	115 056
256×256	240 624	691 200	576 144
512×512	999 408	3 348 480	2 772 336
1024×1024	4 083 696	15 740 928	12 968 592

According to eqns. 3–9, we can also construct an algorithm for computing the two-dimensional DFT (2^n; 2) using the fast discrete cosine transform algorithm presented in Reference 5.

W. MA
D. YANG

16th February 1988

Department of Radio Engineering
South China Institute of Technology
Guangzhou, Guangdong
People's Republic of China

References

1 WINOGRAD, S.: 'On the multiplicative complexity of the discrete Fourier transform', *Math. Comp.*, 1978, **32**, pp. 175–199
2 AUSLANDER, L., FEIG, E., and WINOGRAD, S.: 'New algorithms for the multidimensional discrete Fourier transform', *IEEE Trans.*, 1983, **ASSP-31**, pp. 388–403
3 WINOGRAD, S.: 'Application of complexity of computations to signal processing', *in* SIMON, J. C., and HARALICK, R. M. (Eds.): 'Digital image processing' (Reidel, Dordrecht, 1981)
4 HEIDEMAN, M. T., and BURRUS, C. S.: 'On the number of multiplications necessary to compute a length-2^nDFT', *IEEE Trans.*, 1986, **ASSP-34**, pp. 91–95
5 MA, WEIZHEN, and YIN, RUIXIANG: 'New recursive factorisation algorithms to compute DFT(2^m) and DCT(2^m)'. Proceedings of The IEEE Asian Electronics Conference 1987, Hong Kong, 1987, pp. 71–76
6 AUSLANDER, L., FEIG, E., and WINOGRAD, S.: 'Abelian semi-simple algebras and algorithms for the discrete Fourier transform', *Adv. Appl. Math.*, 1984, **5**, pp. 31–55
7 BLAHUT, R. E.: 'Fast algorithms for digital signal processing' (Addison-Wesley, Reading, Massachusetts, 1985)
8 PEI, SOOCHANG, and WU, JALING: 'Split vector radix 2-D fast Fourier transform'. Proceedings of 1987 IEEE Int. Conf. Acoust., Speech, Signal Processing, Dallas, 1987, pp. 1987–1990

NEAR MINIMAX ALGORITHM OF DECONVOLUTION FILTERING

Indexing terms: Signal processing, Convolution, Algorithms, Deconvolution filtering, Chebyshev minimax norm

A computational algorithm for deconvolution filtering which can be represented as a parallel combination of zero-forcing filters is proposed. The algorithm gives a solution that is close to the Chebyshev minimax-norm-based optimum, which minimises the maximum sidelobe amplitude/mainlobe amplitude ratio in the output signal. Brief examination of the algorithm performance is presented.

Introduction: The problem of deconvolution filtering can be formulated in the following way: for a given discrete input signal $\{s(n)\}$ we seek an FIR type filter impulse response $\{h(n)\}$ such that the filter output signal takes the form of a single pulse

$$\{s(n)\} * \{h(n)\} = \delta(n) \tag{1}$$

where

$$\delta(n) \equiv \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases}$$

and the asterisk denotes convolution. Eqn. 1 is usually assumed to define the deconvolution filtering.

The impulse response $\{h(n)\}$ defined by eqn. 1 is not feasible, and therefore one has to accept approximate solutions. The inevitable approximation error manifests itself in the appearance of additional undesirable pulses, called sidelobes, in the filter output signal. The usual approaches to the problem of $\{h(n)\}$ approximation include: Wiener filtering,[1] truncation inverse filtering,[2] zero-forcing-filtering,[3] and many others. Their common disadvantage is that they do not enable us to control the sidelobe amplitudes relative to the mainlobe amplitude. The definite solution to this problem consists in using the Chebyshev minimax norm for the approximation error. The straightforward application of this norm leads to the Chebyshev solution of inconsistent linear equation systems, which unfortunately must be obtained using rather lengthy iterative procedures.[4]

The aim of this letter is to present a computational non-iterative algorithm that enables us to compute the impulse response $\{h(n)\}$ of an FIR type filter, which for a certain class of input signals gives a solution close to the Chebyshev minimax solution, the latter from now on referred to as the CMS. Moreover, in certain specific cases of input signal the algorithm presented gives the CMS exactly. This algorithm has been examined for those input signals whose mainlobe amplitude is equal to or higher than the sum of the absolute values of sidelobe amplitudes. Therefore, all conclusions which can be drawn from this letter should be restricted to such a class of input signals.

Formulation and solution to the filtering problem: Consider an input signal $\{s(n)\}$;

$$\{s(n)\} = \delta(n) + \sum_{\substack{i=-l \\ i \neq 0}}^{m} s(i) \circ \delta(n-i)$$

$$l, m \quad \text{nonnegative integers} \tag{2}$$

such that:

$$\sum_{\substack{i=-l \\ i \neq 0}}^{m} |s(i)| \leqslant 1 \tag{3}$$

The elements $\delta(n)$ and $s(i) \circ \delta(n-i)$ in eqn. 1 are called the input signal mainlobe and the sidelobes, respectively. Now, consider the impulse response $\{h(n)\}$ that is sought:

$$\{h(n)\} = \delta(n) + \sum_{\substack{i=-q \\ i \neq 0}}^{p} h(i) \circ \delta(n-i)$$

$$p, q \quad \text{nonnegative integers} \tag{4}$$

The convolution of $\{s(n)\}$ and $\{h(n)\}$ gives the output signal $\{g(n)\}$;

$$\{g(n)\} = g(0) \circ \delta(n) + \sum_{\substack{i=-r \\ i \neq 0}}^{v} g(i) \circ \delta(n-i) \tag{5}$$

where the properties of the convolution yield:

$$r + v = m + l + p + q \tag{6}$$

The element $g(0) \circ \delta(n)$ is called the output signal mainlobe, while the elements $g(i) \circ \delta(n-i)$ are the sidelobes.

The Chebyshev minimax approach to our filtering problem consists in the determination of the $\{h(n)\}$ impulse response, subject to the following minimax criterion:

$$\inf_{\{h(n)\} \in H} \left[\frac{\max |g(i)| \quad i \neq 0}{|g(0)|} \right] \tag{7}$$

where H denotes the set of all functions $\{h(n)\}$ which take the value 0 for $n = p + 1, p + 2, \ldots$, and $n = -(q + 1), -(q + 2), \ldots$. This approach, though the optimum one, does not provide closed form analytic solutions, and requires time-consuming iterative procedures, in which the number of iterations strongly depends on guessing the initial set of functions. Therefore, it is postulated here, that for input signals satisfying eqn. 3, the filters which gives solutions close to the CMS have the form shown in Fig. 1.

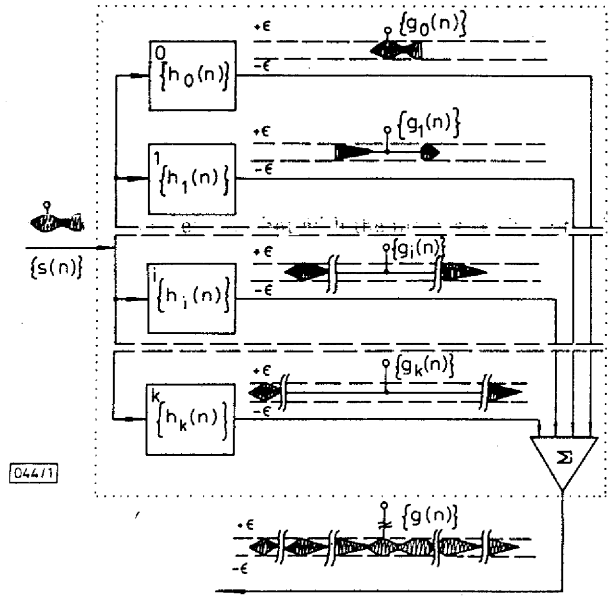

Fig. 1 *Diagram of the algorithm: parallel combination of ZFFs*

As can be seen in Fig. 1 the impulse response $\{h(n)\}$ of the filter proposed can be represented as the sum of properly delayed component impulse responses $\{h_0(n)\}, \{h_1(n)\}, \ldots, \{h_i(n)\}, \ldots, \{h_k(n)\}$. Each of the component impulse responses $\{h_i(n)\}$ represents a finite-length zero-forcing filter (in brief ZFF).[3] Basically, the ZFF causes a certain number of zeroes on both sides of its output signal mainlobe to appear. This specific number equals the number of sidelobes in the ZFF impulse response. In the case of filter shown in Fig. 1, all component impulse responses $\{h_i(n)\}$ should be weighted in such a way that the maximum sidelobe amplitudes at the output of all ZFFs are equal to an arbitrarily chosen constraint ε. At the same time all ZFF output mainlobe amplitudes should be of the same sign. Thus, the mainlobe amplitude in the output signal $\{g(n)\}$ equals the sum of all ZFFs mainlobe amplitudes, while the maximum sidelobe amplitudes do not exceed ε. With reference to eqns. 2 and 4, one finds that the specific number i of each component impulse response $\{h_i(n)\}$ in Fig. 1 determines the number of its elements as follows:

(1) for arbitrary input signals,

$$p = 0 \quad q = 0 \qquad \text{for } i = 0$$

$$p + q = (m + l) \circ 2^{i-1} \quad \text{for } i = 1, 2, \ldots, k \tag{8}$$

(2) for even input signals in which, owing to their symmetry $p = q$ and $m = l$, which in turn makes possible

$$p = q = i \circ m = i \circ l \tag{9}$$

For even input signals, the use of eqn. 9 gives more favourable performance than eqn. 8. The number k of component impulse responses $\{h_i(n)\}$, which comprise the filter discussed will be hereafter referred to as the filter order. Each $\{h_i(n)\}$ can be obtained by solving the appropriate system of 2^{i-1} or i linear equations, in the case of eqns. 8 or 9, respectively. These equations can be determined by writing the expression for the convolution of $\{s(n)\}$ and $\{h_i(n)\}$ explicitly, and requesting that the appropriate number (given by eqn. 8 or 9), of the convolution product elements be zero.

Examination results: The filtering algorithm discussed has so far been tested in application to even input signals, using eqn. 9. It has been proved that in the case of the three-element input signal, i.e., consisting of a mainlobe and two symmetrically spaced sidelobes, the algorithm gives the CMS exactly.[5,6] For the five-element even input signal with sidelobes of equal amplitudes and arbitrary signs, and for the seven-element even input signal with sidelobes of equal amplitudes and signs, it has been proved analytically that for $k = 1$ the algorithm discussed also gives CMS (a paper in preparation). Preliminary tests with short even input signals, whose length did not exceed seven elements, and for $k \leq 2$, have proved that the approximation error with the algorithm discussed was only slightly larger than in the case of the CMS. Moreover, it was found that in all cases examined, the solutions obtained with this algorithm enabled us to determine correctly the initial system of functions for the iterative ascent algorithm,[4] such that the CMS could be obtained directly, without iteration. One may expect that with the increase of both the input signal length and the order of filtering k, the discrepancy between the solution obtained with the algorithm presented and the CMS will also increase. However, it is believed that the solution obtained with this algorithm will be a good beginning to determining the initial set of functions in the ascent algorithm, reducing thus the number of iterations in the computation of the CMS. Moreover, as the ascent algorithm underestimates the CMS approximation error, and the algorithm discussed overestimates this error, the interval in which the CMS approximation error lies can be determined after the first iteration. The behaviour of the algorithm presented for long input signals and large values of k is under investigation.

Acknowledgments: This work has been carried out with the financial support of the CPBP 02.16.4.2 research project. The author wishes to acknowledge the assistance of and fruitful discussions with H. Ugowski.

A. DYKA

Institute of Telecommunications
Technical University of Gdańsk
80-952 Gdańsk, Poland

19th February 1988

References

1 ZIOLKOWSKI, A.: 'Deconvolution' (D. Reidel, 1984)

2 DYKA, A.: 'Sidelobe reduction filtering technique using truncated inverse filter method'. Proc. of the 8-th Colloquium on Microwave Communication, Microcoll-86, Budapest, 1986, pp. 123–124

3 LUCKY, R. W.: 'Automatic equalization for digital communication', *Bell. Syst. Tech. J.*, 1965, **44**, pp. 547–588

4 CHENEY, E. W.: 'Introduction to approximation theory' (McGraw-Hill, 1966)

5 DYKA, A., and UGOWSKI, H.: 'Minimax sidelobe reduction filtering for Huffman sequence autocorrelation function type signals', *IEEE Trans.*, 1988, **CAS-35**, accepted for publication

6 UGOWSKI, H., and DYKA, A.: 'On a certain minimax problem in deconvolution-inverse filtering', *COMPEL-Int. J. Comput. & Math. Electr. & Electron. Eng.*, 1988, 7, accepted for publication

ON *RC* LINE DELAYS AND SCALING IN VLSI SYSTEMS

Indexing terms: Integrated circuits, Large-scale integrations, Clocks, Metal-oxide-semiconductor structures and devices

We propose that a special interconnection metal layer is introduced into the VLSI process and used for global clock distribution and communication. We show that by this method the interconnection delay will not be a limiting factor for the speed of synchronous circuits in CMOS down to 0·3 μm feature sizes and up to 20 mm chip sizes.