Realisation Techniques For Primitive Operator Infinite Impulse Response Digital Filters

David R. Bull and David H. Horrocks*
Dept Electrical and Electronic Engineering, University of Bristol, Queens Building, University Walk,
Bristol BS8 1TR. UK
*School of Electrical, Electronic and Systems Engineering, University of Wales, PO Box 904,
Cardiff CF1 3YH. UK

Abstract- **This paper presents realisation techniques for IIR digital filters based on the primitive operator graph synthesis method. The approach exploits the redundancy present in the filter multiplier structure when the coefficients are invariant allowing partial result reuse throughout the filter structure. Results are presented for a range of elliptic filters with varying wordlength and order, realised using direct form, parallel and cascade structures. These show significant savings when compared with the equivalent canonical signed digit multiplier coding scheme.**

I. Introduction.

Although possessing the desirable properties of guaranteed stability and linear phase, FIR filters generally exhibit a higher computational complexity for a given filter characteristic than do their IIR counterparts. For this reason, in situations where phase linearity is not critical or can be easily compensated through equalisation, IIR filters are often used in preference to FIR.

The transfer function for an Nth order IIR filter is given by equation (1).

$$H(z)=\frac{\sum_{i=0}^{N} a_i z^{-i}}{1+\sum_{i=1}^{N} b_i z^{-i}} \qquad (1)$$

The canonic direct form II structure which realises this expression is shown in figure 1. This structure may also be transposed to yield an equivalent structure with the numerator and denominator computation orders interchanged. The equivalent non-canonic direct form I structure for equation 1 is given in figure 2. Although often considered superior, the direct form II structure does suffer from a distributed accumulator whereas the direct form I has effectively only one. The latter therefore has the advantage that saturation arithmetic is often not necessary since internal overflows prior to final result formation can occur without error. This in turn increases the effective length of the internal registers thereby reducing the need for input scaling and improving the SNR.

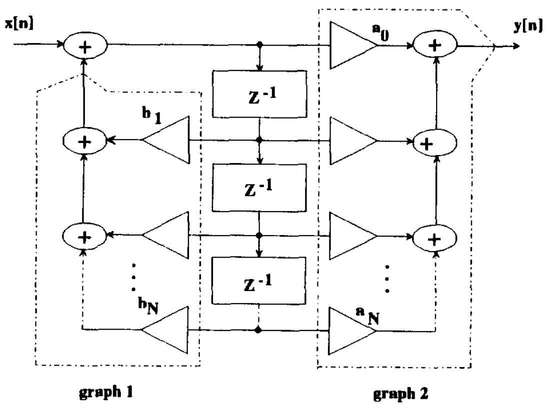


Fig.1. Direct form II IIR structure

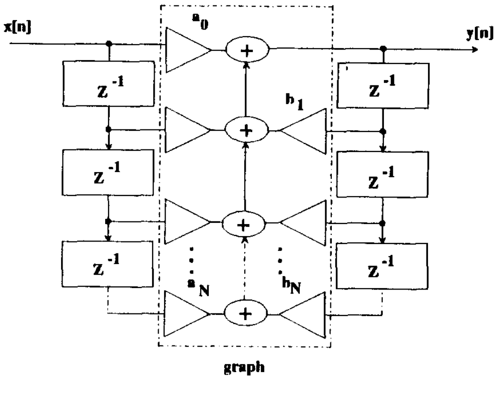


Fig.2. Direct form I IIR structure

Although simply realised, it is well known that, for higher order filters, the direct form structure suffers from a geometric increase in the range of coefficient values coupled with higher pole-zero sensitivity and quantisation noise

0-7803-1254-6/93$03.00 © 1993 IEEE

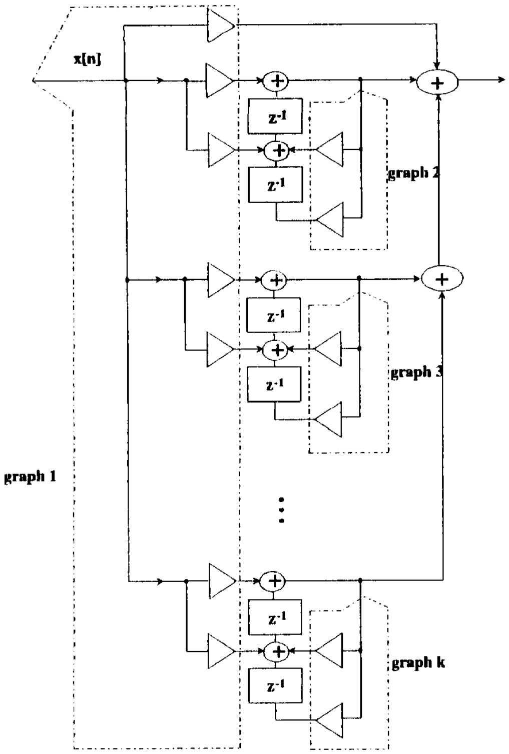


Fig. 3. Parallel decomposition

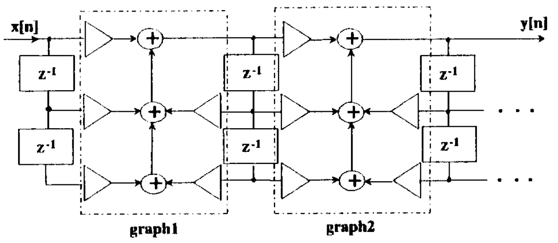


Fig.4. Cascade decomposition

recirculation [1]. As a consequence of this it is usual to construct higher order filters from parallel combinations (equation (2)) or cascades (equation (3)) of quadratic sections. These are illustrated in figures 3 and 4 respectively.

$$H(z) = c_0 + \sum_{i=1}^{L} \frac{c_{i0} + c_{i1}z^{-1}}{1 + d_{i1}z^{-1} + d_{i2}z^{-2}} \qquad (2)$$

$$H(z) = \prod_{i=1}^{L} \frac{a_{i0} + a_{i1}z^{-1} + a_{i2}z^{-2}}{1 + b_{i1}z^{-1} + b_{i2}z^{-2}} \qquad (3)$$

The former, while sometimes offering improved noise performance [1], has higher sensitivity to coefficient quantisation and so cascades are often preferable. For audio applications however, graphic equalisers are traditionally realised as parallel combinations while parametric equalisers are realised as cascades. [2]. Despite the fact that lower sensitivity IIR structures such as wave filters do exist, implementations using the structures of figures 3 and 4 are still commonplace.

With ASIC technology becoming more readily available to an increasing number of end users it is becoming common for products to incorporate high performance dedicated function filtering functions in place of programmable DSP devices, the performance of the latter often being insufficient to satisfy application requirements. For such dedicated functions it is desirable to make the most efficient use of silicon area. To this end there has been much work reported in the literature over the past few years concerning complexity reduction techniques for filter realisation. Examples of this include realisations based on distributed arithmetic [3], the use of restricted coefficient values [4], MSB first bit-level systolic arrays [5] and, for FIR filters, primitive operator techniques [6].

The aim of this paper is to introduce and demonstrate the applicability of the primitive operator graph synthesis technique to the realisation of IIR filters. The approach exploits the redundancy present in the coefficient-sample multiplication process when one vector is invariant, the underlying principle relying on the fact that partial results formed in any one multiplication can be reused in the formation of other product terms. Through the application of efficient synthesis algorithms, the inner product process may be represented as a directed graph comprising only primitive arithmetic operations (addition, subtraction and k-bit shift). This method is well suited to the realisation of FIR filters where the higher number of coefficients result in the formation of an efficient graph. Typical results have shown that, for many filters the complexity of the multiplication process is reduced to less than one addition per coefficient. For example we have realised a 32nd order linear phase FIR filter with 8-bit coefficients which replaces seventeen multipliers by only ten additions [6].

II. IIR FILTER REALISATION USING PRIMITIVE OPERATOR METHODS

Two problems arise when primitive operator methods are applied to IIR filters. Firstly, the order of IIR filters is generally lower than their FIR counterparts resulting in a less efficient graph (graph efficiency here being defined in terms of vertex reuse). This is compounded when the filter is realised in terms of first and second order sections. Secondly, the IIR filter transfer function comprises both recursive and non-recursive components each appearing to require an independent graph.

Consider first the usual canonic structure of figure 1. This clearly requires two independent graphs as indicated. A improved situation however results from the non-canonic structure of figure 2 where all of the multipliers can be realised in a single graph. Although non-canonic, the additional z^{-1} registers in this structure will not present a significant overhead in any bit-parallel architecture.

Due to the reasons outlined in section 1, direct form IIR filter realisations are seldom employed. Instead the transfer function is decomposed either by factorisation to yield a cascade structure or using partial fractions to yield a parallel structure. The parallel arrangement appears to offer greater potential in the context of a primitive operator realisation. As can be observed from figure 3 all transfer function numerator coefficients in all second order sections may be formed using a single graph. The recursive part in each section, comprising only two coefficient values, must however be realised independently. The transpose structure of that shown in figure 3 can also be treated similarly.

In the case of the cascade realisation of figure 4, savings can be made through the choice of the non-canonic direct form I structure (figure 4) as the second order building block. This offers the advantage of allowing each graph to represent five coefficients and facilitates the sharing of z^{-1} registers between sections in the cascade.

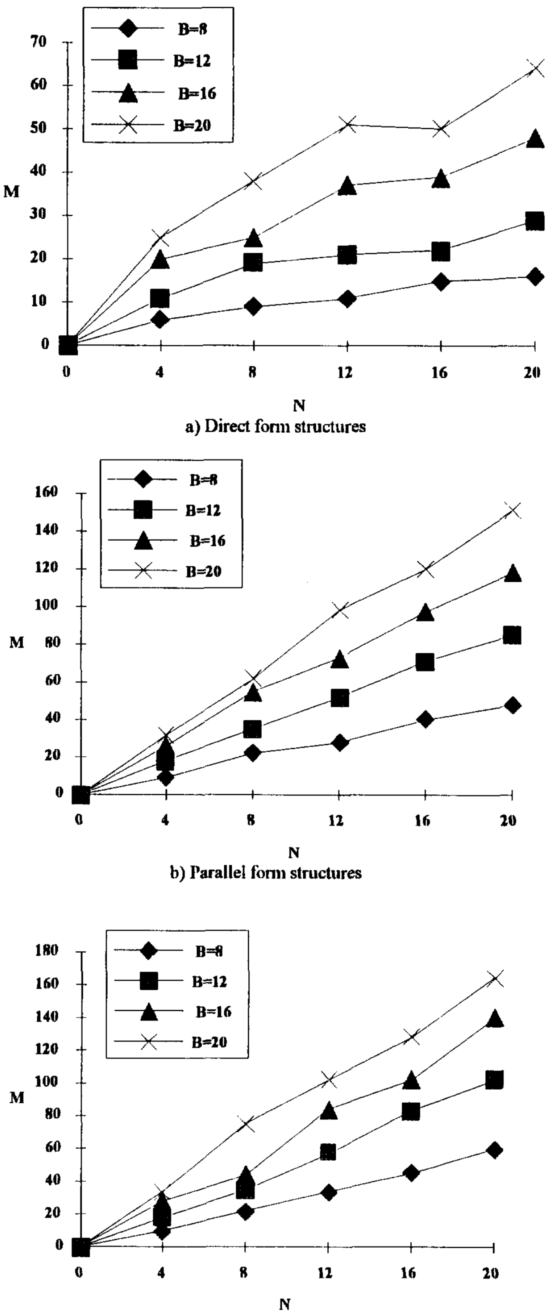
Certain types of filter, for example those requiring low frequency, high Q responses with limited precision coefficients, are difficult to implement using the conventional IIR structures due to the scarecety of pole locations near z=1. Structures such as the Gold-Rader coupled form [7] which overcome this problem are also amenable to primitive operator realisations. Systems employing error feedback can also benefit from the technique.

III. Results

This section presents arithmetic complexity results for direct form, parallel and cascade IIR filter structures when realised using primitive operator techniques. All filters are based on low-pass elliptic prototypes with the following specification:

Normalised pass-band edge	$\Omega_p= 0.1$
Normalised stop-band edge	$\Omega_s= 0.105$
Pass-band ripple	$\Delta p= 0.14345$
Stop-band ripple	$\Delta s=$ free

The filter order has been allowed to vary between N=4 and N=20 and coefficients have been quantised to wordlengths in the range B= 8 to 20 bits. All results presented were obtained using a graph synthesis algorithm employing addition, subtraction and shift operators.



a) Direct form structures

b) Parallel form structures

c) Cascade structures

Fig. 5. Arithmetic complexity, M, plotted for filters with varying order, N, and coefficient wordlength, B. All filters are derived from elliptic prototypes using the bilinear transform.

Figures 5a, b and c show results for direct form, parallel and cascade realisations respectively. The ordinate, M, indicates the number of addition / subtraction operations required to form the primitive operator graph(s) in each case. These values do not therefore take into account the accumulator adders. As can be observed from figure 5, the best performance is obtained in the case of the direct form structure. This is not unexpected as all coefficients can be embodied in a single graph.

Arithmetic complexity results for the cascade and parallel structures are similar for all values of N and B, being approximately twice that of the direct form case. The advantage gained in the case of the parallel decomposition from the single graph formed from all the numerator coefficients is in practice counteracted by the need to form lower order graphs for each set of denominator coefficients.

The results from the POF approach have been compared with those obtained from forming each coefficient individually using a canonical signed digit coding scheme. Savings for the cascade and parallel decompositions ranged between 26% and 35% with this increasing to 56% for the direct form case. Results from the primitive operator addition / subtraction / shift algorithm were also compared with those from an add / shift only algorithm. Unlike the FIR case, where filter order tends to be relatively large and the incorporation of subtraction yields little benefit, in the IIR case savings of up to 27% were obtained.

Table 1

	Arithmetic Complexity		
B	**Add/Sub**	**Add Only**	**CSD**
8	19	21	31
12	37	44	60
16	54	65	89
20	73	101	126

In addition to the results shown in figure 5 we have also synthesised structures for a cascade decomposition into quartic sections. Since each graph now embodies 9 coefficients rather than 5 as in the second order case, the efficiency of the synthesis algorithm increases. Results for the N=12 case are given in table 1 and can be seen to compare favourably with the results from figure 5c.

IV. Conclusions

This paper has demonstrated the applicability of primitive operator based realisation methods in the context of fixed function IIR filters. Although generally of lower order than their FIR counterparts, significant savings can be made in the IIR case if an appropriate filter topology is adopted. The direct form structure, although rarely used in practice, exhibits the greatest reduction in arithmetic complexity. In the cascade and parallel forms, relative savings are less due to the fact that each graph is restricted to the formation of a small number of coefficients. Even so, savings over canonical signed digit coding of up to 35% have been reported. Decomposition of a transfer function into a cascade of quartic sections has been shown to offer improved realisation complexity when compared to the second order case. This however is at the expense of an increased sensitivity structure.

Acknowledgement

The authors would like to thank Mr. Graham Wacey for his assistance in preparing the results presented here

References

[1] Jackson, L.B. 'Digital filters and Signal Processing', Kluwer Academic Publishers, Boston, MA, 1986

[2] Dattorro, J.,'The Implementation of Recursive Digital Filters for High Fidelity Audio', J. Audio Eng. Soc., Vol. 36, No. 11, November 1988, pp851-878.

[3] Peled, A. and Liu, B.,'A New Hardware Realisation of Digital Filters', IEEE Trans. ASSP, Vol.22, No.6, December 1974, pp456-462.

[4] Leuder, E.,'Fast Digital Filters Without Multipliers', IEEE Intl. Symp. on Circuits and Systems, Rome 1982, pp1037-1040

[5] Knowles, S.C., McWhirter, J. G., Woods, R. F. and McCanny, J. V.,'A Bit-Level Systolic Architecture for Very High Performance IIR Filters' IEEE Intl. Conf. on ASSP, Glasgow, Scotland, May 1989, pp2449-2452

[6] Bull, D. R. and Horrocks, D. H., 'Primitive Operator Digital Filters' IEE Proc. Pt. G, Vol. 138, No.3, June 1991, pp401-412.

[7] Gold, B. and Rader, C.M., 'Effects of Parameter Quantisation on the Poles of a Digital Filter', Proc. IEEE, May 1967, pp688-689