

# On the Use of Multiple Constant Multiplication in Polyphase FIR Filters and Filter Banks

Oscar Gustafsson<sup>1</sup> and Andrew G. Dempster<sup>2</sup>

<sup>1</sup>Department of Electrical Engineering  
 Linköping University  
 SE-581 83 Linköping, SWEDEN  
 E-mail: oscarg@isy.liu.se

<sup>2</sup>University of Westminster  
 115 New Cavendish St  
 London W1W 6UW, UK  
 a.dempster@cmsa.westminster.ac.uk

## ABSTRACT

Multiple constant multiplication (MCM) has been shown to be an efficient way to reduce the number of additions and subtractions in FIR filter implementations. However, for polyphase decomposed FIR filters and filter banks, the problem can be formulated in three different ways. Either as one MCM block with all coefficients, one MCM block for each subfilter, or as a matrix MCM block. In this work we compare the approaches in terms of complexity, both for the MCM blocks and for the remaining hardware, such as structural additions and delay elements.

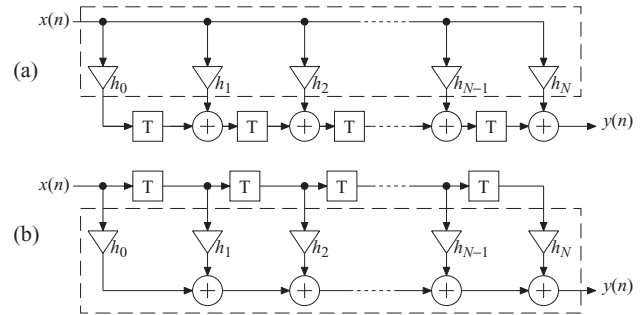
## 1. INTRODUCTION

The multiple constant multiplication (MCM) problem, i.e., multiplying one data with several coefficients has received consideration over the years [1], [2]. By expressing the multiplication using shifts, additions, and subtractions a realization without general multipliers is obtained. The number of additions and subtractions can then be significantly reduced by using common partial results. As additions and subtractions has similar complexity we will from now on only refer to them as additions.

The development of efficient algorithms has to a high extent been motivated by the use of MCM blocks in FIR filters. For direct transposed form FIR filters the input is multiplied with the filter coefficients as shown in Fig. 1 (a), where the MCM block is marked with a dashed box. Using transposition, a direct form FIR filter is obtained, as shown in Fig. 1 (b), where the sum-of-product computation is marked with a dashed box. Hence, MCM is also efficient for sum-of-product computations. We will from now on refer to MCM block for both multiple constant multiplication and for the sum-of-products computation. Sometimes the term multiplier block is used for MCM block [1]. But as that may indicate the algorithm used, we use the more general term MCM block.

However, in some applications, more than one FIR filter is operating on the same data stream. This is the case for, e.g., FIR filters and filter banks using polyphase representation [3]–[5]. In [6] it was shown that by using transposed form FIR filters only one MCM block. However, this leads to that more delay elements and structural additions are required, as each filter has its own delay line. For direct form FIR filters, one must apply MCM to each subfilter.

With the recent advances of algorithms for matrix MCM [7], [8], this gives us a third alternative. Polyphase decom-



**Fig. 1.** (a) Transposed direct form FIR filter. (b) Direct form FIR filter.

posed filters and filter banks can be written as a matrix multiplication, and, hence, a matrix MCM algorithm can be applied.

In this work, we consider the alternatives for use of MCM in polyphase decomposed filter banks and interpolation and decimation filters based on FIR filters. By example filters we show some properties of the three different alternatives, not only in terms of additions in the MCM blocks, but also delay elements and structural additions.

## 2. POLYPHASE STRUCTURES

Any FIR filter can be written in a polyphase form. Assume an  $N$ th order filter with transfer function

$$H(z) = \sum_{n=0}^N h_n z^{-n} \quad (1)$$

Performing a polyphase decomposition leads to that the transfer function is rewritten as

$$H(z) = \sum_{m=0}^{M-1} z^{-m} H_m(z^M) \quad (2)$$

where

$$H_m(z) = \sum_{n=0}^{\frac{N+1}{M}-1} h_{nM+m} z^{-n}, \quad m = 0, 1, \dots, M-1 \quad (3)$$

Hence, the filter now consists of  $M$  subfilters,  $H_m(z^M)$ .

## 2.1. Filter Structures

Assuming that the filter is used as an interpolator filter, polyphase decomposition and the noble identity [4] can be used to derive an efficient interpolator structure as shown in Fig. 2.

Figures 3 (a) and (b) show the resulting interpolator using transposed direct form and direct form FIR filters, respectively. For simplicity, the filter structures are shown for  $M = 2$ , but it is easy to generalize to arbitrary  $M$ . In a similar way can the filters in an analysis bank in, e.g., an  $M$  channel QMF bank be rewritten to yield an identical filter structure. This is also the filter structure used for, e.g., modified Farrow filters (except the delay multipliers) [9].

For a decimator the structures in Fig. 3 can be transposed, resulting in the filter structures in Fig. 4. These filter structures may be used as a synthesis bank in many filter banks.

## 2.2. Complexity

The complexity of the resulting realization will be dependent of three factor. First, the size, numbers, and type of MCM blocks. Second, the number of delay elements, and, finally the number of structural additions, i.e., the additions that are not part of the MCM block (additions outside of the dashed box in Fig. 1 (a)). Here, we focus the discussion on polyphase decomposed interpolation and decimation filters, but identical results can be derived for the filter bank case. We assume that the filter order of the total filter is  $N$  and that the number of subfilters are  $M$ . Hence, each subfilter has a filter order  $(N + 1)/M - 1$  (assuming for simplicity that  $M$  in an integer multiple of  $N + 1$ ).

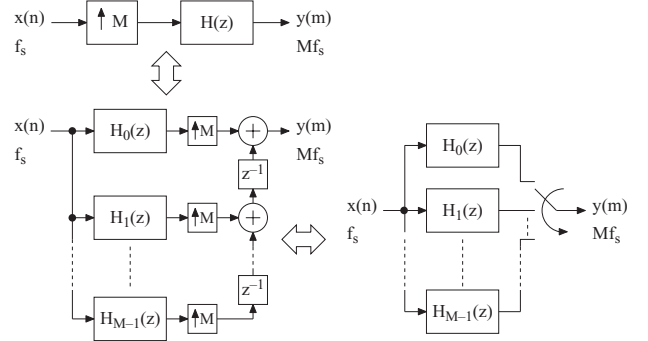
Utilizing either the filter structure in Figs. 3 (a) and 4 (a) one MCM block including all filter coefficients can be used, as indicated by the dashed box. The number of delay elements is

$$\left(\frac{N+1}{M} - 1\right)M = N - M + 1 \quad (4)$$

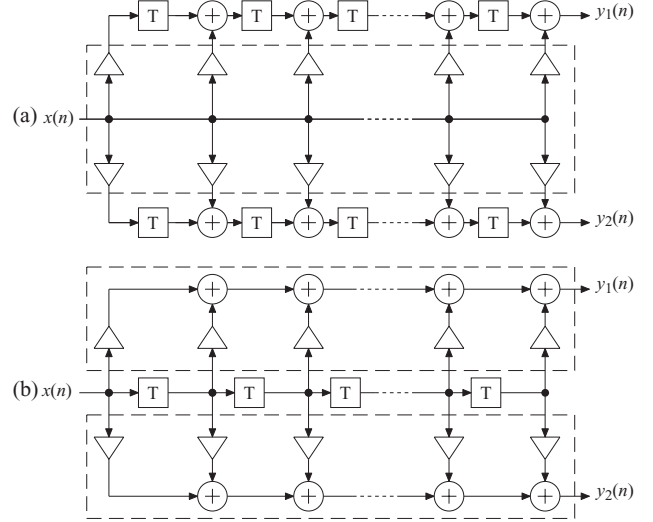
while the number of structural additions are  $N - M + 1$  for interpolators. For the decimator in Fig. 4 (a), the structural additions are included in the MCM block.

For the filter structures in Figs. 3 (b) and 4 (b)  $M$  smaller MCM blocks must be used. However, the number of delay elements is only  $(N + 1)/M - 1$ . For the interpolator the structural additions are included in the MCM block. For the decimator  $N$  structural additions is required. However, each addition between the delay elements in Fig. 4 (b) is in fact two two-input additions ( $M$  for the general case). If a matrix MCM algorithm was used, only one of these structural additions would be required, the remaining would be inside the matrix MCM block. This is illustrated in Fig. 5. It is off course possible to transpose the filter structure in Fig. 5 to obtain an interpolator corresponding to Fig. 3 (b).

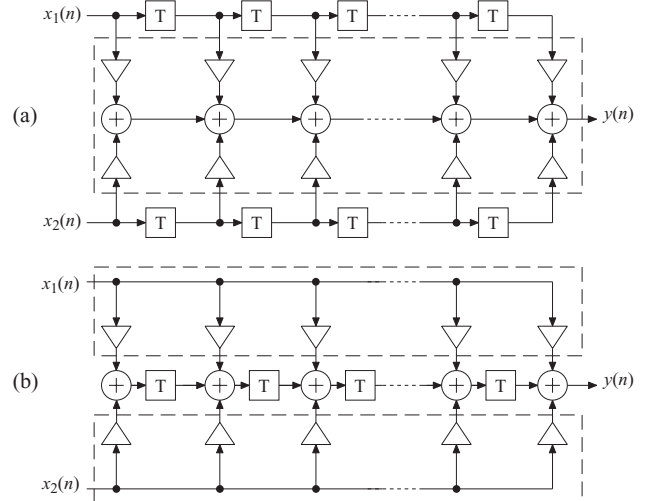
If the FIR filter has linear phase-response, it is possible to utilize the symmetry of the filter coefficients to reduce the number of multiplications. However, among the subfilters resulting from the polyphase decomposition only 1 or 2 may have symmetrical coefficients [10]. Hence, the total number of multiplications in the subfilters are as in Table 1.



**Fig. 2.** Polyphase decomposition of an interpolator.



**Fig. 3.** Analysis filter bank or polyphase interpolator based on a (a) transposed direct form and (b) direct form FIR filter.

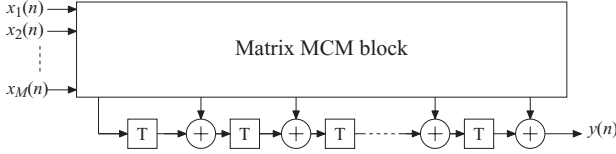


**Fig. 4.** Synthesis filter bank or polyphase decimator based on a (a) direct form and (b) transposed direct form FIR filter.

The complexity for the different cases are summarized in Table 2.

## 2.3. Transposition

Transposition of a matrix multiplication is useful as the matrix MCM algorithms [7], [8] will produce different re-



**Fig. 5.** The filter structure in Fig. 4 (b) using a matrix MCM block.

**Table 1.** Total number of multiplications utilizing symmetry for polyphase decomposed direct form FIR filters.

|          | $N$ even                                            | $N$ odd                                             |
|----------|-----------------------------------------------------|-----------------------------------------------------|
| $M$ even | $N + 1 - \left\lfloor \frac{N+1}{M} \right\rfloor$  | $N + 1$                                             |
| $M$ odd  | $N + 1 - \left\lfloor \frac{N+1}{2M} \right\rfloor$ | $N + 1 - \left\lfloor \frac{N+M}{2M} \right\rfloor$ |

sults depending on transposition. Because of how the algorithms work, both algorithms generally produces better results for matrices with more rows than columns. An optimal algorithm would always give the same results, apart from the inherent differences because of the different numbers of inputs and outputs.

Assume a network  $N_1$  has  $N_{1i}$  inputs and  $N_{1o}$  outputs. It consists of  $N_{1A}$  two-input additions and  $N_{1F}$  forks, i.e., where a single signal is sent to two places.

Transposition of the network  $N_1$  results in a network  $N_2$ . This network will have  $N_{2i}$  inputs,  $N_{2o}$  outputs,  $N_{2A}$  two-input additions and  $N_{2F}$  forks. The rules of transposition [11] give that

$$\begin{aligned}
 N_{2i} &= N_{1o} \\
 N_{2o} &= N_{1i} \\
 N_{2A} &= N_{1F} \\
 N_{2F} &= N_{1A}
 \end{aligned} \tag{5}$$

The total out-degree of the nodes in the network  $N_1$  is

$$N = N_{1i} + N_{1A} + 2N_{1F} \tag{6}$$

i. e., each input and each addition has one outgoing edge and the fork has two. Similarly, the total in-degree of the nodes in the network  $N_1$  is

$$N = N_{1o} + 2N_{1A} + N_{1F} \tag{7}$$

As each edge in the network is an incoming edge to one node and an outgoing edge of another, we have that

$$N_{1i} + N_{1A} + 2N_{1F} = N_{1o} + 2N_{1A} + N_{1F} \tag{8}$$

Equations (5) and (8) gives

$$N_{2A} = N_{1A} + N_{1o} - N_{1i} \tag{9}$$

Hence, if a matrix multiplication with an  $N \times M$  matrix is realized using  $A$  additions, the realization of the transposed  $M \times N$  matrix would require  $A + M - N$  additions.

This also has an intuitive relation with FIR filters and MCM blocks. A one-in, several-out MCM block can be attached to a chain of additions and delays to produce a transposed direct form FIR filter. Transposing the FIR filter results in the additions merging with the resulting MCM block yielding in total the same number of additions (the delay line is a several-in, one-out network).

For a complete interpolator and decimator with a factor  $M$ , the relation between the number of additions after transposition is as follows. Assuming that the interpolator requires  $A$  additions, the decimator requires, according to (9),  $A + M - 1$  additions. If the decimator requires  $A$  additions, the corresponding interpolator requires  $A + 1 - M$  additions.

### 3. RESULTS

In this section a number of different FIR filters are designed using the three different approaches discussed in the previous section. The single input MCM blocks are designed using the RAGn algorithm [1]. The matrix multiplier blocks are designed using the matrix MCM algorithms in [8], referred to as MMST. Because of problems with negative numbers in the current implementation of the algorithm in [7], this will only be applied to cases where all filter coefficients are positive.

In both cases the transposition of the matrix is selected to that there are more rows than columns as the matrix MCM algorithms generally prefers that case.

#### 3.1. Example 1

In this example a linear-phase FIR filter for decimation with a factor three is designed. The passband and stopband edges are at  $0.3\pi$  and  $0.35\pi$ , respectively. The filter is designed for a passband ripple of 0.01 and a stopband ripple of 0.001. Using a filter order of 110 this filter is synthesized in MATLAB using *remez.m* and rounding the coefficients to 12 bits precision.

Using direct form subfilters a total of 136 additions are required for the sum-of-products MCM block. For the original MCM block 28 additions are required plus 108 for the transposition (two coefficients are zero). In addition to that 109 delay elements are required.

**Table 2.** Complexity for the different filter structures.

| Filter structure              | Subfilter                           | MCM block                                         | Delay elements  | Structural additions |
|-------------------------------|-------------------------------------|---------------------------------------------------|-----------------|----------------------|
| Interpolator or analysis bank | Transposed direct form (Fig. 3 (a)) | One with $N + 1$ coefficients                     | $N - M + 1$     | $N - M + 1$          |
|                               | Direct form (Fig. 3 (b))            | $M$ sum-of-products with $(N + 1)/M$ coefficients | $(N + 1)/M - 1$ | -                    |
|                               | Direct form (Fig. 5 transposed)     | Matrix with $M$ rows and $(N + 1)/M$ columns      | $(N + 1)/M - 1$ | -                    |
| Decimator or synthesis bank   | Direct form (Fig. 4 (a))            | One sum-of-products with $N + 1$ coefficients     | $N - M + 1$     | -                    |
|                               | Transposed direct form (Fig. 4 (b)) | $M$ with $(N + 1)/M$ coefficients each            | $(N + 1)/M - 1$ | $N$                  |
|                               | Transposed direct form (Fig. 5)     | Matrix with $(N + 1)/M$ rows and $M$ columns      | $(N + 1)/M - 1$ | $(N + 1)/M - 1$      |

For transposed direct form FIR filters the number of additions in the three multiplier blocks are 19, 14, and 19 giving a total of 52 additions. Furthermore, 108 structural additions are required leading to a total of 160 additions. 36 delay elements are required.

Using the algorithm in [8], 99 additions are required for the matrix MCM block. In addition to that 36 structural additions are required according to Table 2. 36 delay elements are required for this case as well.

For this example the matrix approach was clearly the best, both in terms of additions and delay elements.

### 3.2. Example 2

In this example a decimation filter for decimation with a factor of 16 is designed using four cascaded 15th-order comb filters. The impulse response is, hence, obtained by convolution of four vectors with 16 ones in each, resulting in a 60th-order filter. The results are summarized in Table 3. Here, the algorithm in [7] (referred to as BHMM) is also used.

For this filter the direct form approach uses 13 additions less than the BHMM matrix approach at the cost of 42 delay elements. The trade-off between additions and delay elements is circuit and technology dependent, and, hence, should be evaluated on the circuit level.

### 3.3. Example 3

In this example an interpolator with a factor of two is designed. We will consider both the cases with odd and even order filters, yielding different symmetry properties, as can be seen from Table 1.

The passband and stopband edges are at  $0.4\pi$  rad and  $0.55\pi$  rad, respectively. Both the passband and the stopband ripples are 0.01. Using a 30th-order (#1) and 31st-order (#2) filter with rounding to 10 bits the results in Table 4 are obtained.

In this example the direct form subfilters for the 30th-order are symmetric, and, hence, the number of additions for the multiplications is low. This gives that for the 30th-order filter 14 delay elements are saved for direct form subfilters compared with the transposed direct form subfilters at the cost of three additions. It may be worth noting that an optimal matrix MCM algorithm will never yield results that are worse than for the direct form subfilters.

For the 31st-order filter the matrix approach saves 15 delay elements compared with the transposed direct form subfilters at the cost of three additions.

## 4. CONCLUSIONS

In this work, the usage of multiple constant multiplication (MCM) techniques for polyphase decomposed filters and filter banks was discussed. We have shown that there are three different cases that could be applied. Either one MCM block with all coefficients leading to many delay elements and possibly many structural additions. The second alternative is to use separate multiplier blocks for each subfilter. This reduces the number of delay elements. The third approach, introduced in this paper, is to use a matrix MCM

**Table 3.** Results of Example 2 (decimator).

| Filter structure  | MCM block additions | Structural additions | Delay elements |
|-------------------|---------------------|----------------------|----------------|
| Direct form       | 86                  | -                    | 45             |
| Transposed        | 91                  | 60                   | 3              |
| Transposed (MMST) | 102                 | 3                    | 3              |
| Transposed (BHMM) | 96                  | 3                    | 3              |

**Table 4.** Results of Example 3 (interpolator).

| # | Filter structure | MCM block additions | Structural additions | Delay elements |
|---|------------------|---------------------|----------------------|----------------|
| 1 | Transposed       | 10                  | 29                   | 29             |
|   | Direct form      | 42                  | -                    | 15             |
|   | Direct (MMST)    | 46                  | -                    | 15             |
| 2 | Transposed       | 11                  | 30                   | 30             |
|   | Direct form      | 52                  | -                    | 15             |
|   | Direct (MMST)    | 44                  | -                    | 15             |

block. This leads to few structural additions and few delay elements, but requires an algorithm for matrix MCM, a research area under development.

Example filters showed that there is a trade-off between additions and delay elements, depending on the filter structure, and that the best approach depends on the filter coefficients. However, the matrix approach was shown to be competitive for all example filters.

## 5. REFERENCES

- [1] A. G. Dempster and M. D. Macleod, "Use of minimum-adder multiplier blocks in FIR digital filters," *IEEE Trans. Circuits Syst.-II*, vol. 42, no. 9, pp. 569–577, Sep. 1995.
- [2] R. Pasko, P. Schaumont, V. Derudder, S. Vernalde, and D. Durackova, "A new algorithm for elimination of common subexpressions," *IEEE Trans. Computer-Aided Design Integrated Circuits*, vol. 18, no. 1, pp. 58–68, Jan. 1999.
- [3] M. Bellanger, G. Bonnerot, and M. Coudreuse, "Digital filtering by polyphase network: application to sample rate alterations and filter banks," *IEEE Trans. Acoust. Speech Signal Processing*, vol. 24, pp. 109–114, April 1976.
- [4] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*, Prentice Hall, 1993.
- [5] L. Wanhammar and H. Johansson, *Digital Filters*, Linköping University, 2003.
- [6] A. G. Dempster and N. P. Murphy, "Efficient interpolators and filter banks using multiplier blocks," *IEEE Trans. Signal Processing*, vol. 48, no. 1, pp. 257–261, Jan. 2000.
- [7] A. G. Dempster, O. Gustafsson, and J. O. Coleman, "Towards an algorithm for matrix multiplier blocks," in *Proc. European Conf. Circuit Theory Design*, Kraków, Poland, Sept. 1–4, 2003.
- [8] O. Gustafsson, H. Ohlsson, and L. Wanhammar, "Low-complexity constant coefficient matrix multiplication using a minimum spanning tree approach," in *Proc. Nordic Signal Processing Symposium*, Espoo, Finland, June 9–11, 2004.
- [9] J. Vesma and T. Saramäki, "Optimization and efficient implementation of FIR filters with adjustable fractional delay," in *Proc. IEEE Int. Symp. Circuits Syst.*, Hong Kong, June 9–12, 1997, vol. IV, pp. 2256–2259.
- [10] R. A. Hawley, B. C. Wong, T.-J. Lin, J. L. Laskowski, and H. Samuelli, "Design techniques for silicon compiler implementations of high-speed FIR digital filters," *IEEE J. Solid-State Circuits*, vol. 31, pp. 656–667, May 1996.
- [11] R. E. Crochiere and A. V. Oppenheim, "Analysis of linear digital networks," *Proc. IEEE*, vol. 63, pp. 581–595, April 1975.