FAST SLIDING TRANSFORMS IN TRANSFORM-DOMAIN ADAPTIVE FILTERING[1]

*Annamária R. Várkonyi-Kóczy** *Sergios Theodoridis†*

*Department of Measurement and Instrument Engineering,
Technical University of Budapest, H-1521 Budapest, Hungary

†Department of Informatics,
University of Athens, 15771 Athens, Greece

email: koczy@mmt.bme.hu stheodor@di.uoa.gr

ABSTRACT

Transform-domain adaptive signal processing proved to be very successful in very many applications especially where systems with long impulse responses are to be evaluated. The popularity of these methods is due to the efficiency of the fast signal transformation algorithms and that of the block oriented adaptation mechanisms. In this paper the applicability of the fast sliding transformation algorithms is investigated for transform domain adaptive signal processing. It is shown that these sliding transformers may contribute to a better distribution of the computational load along time and therefore enable higher sampling rates. It is also shown that the execution time of the widely used Overlap-Save and Overlap-Add Algorithms can also be shortened. The prize to be paid for this improvements is the increase of the end-to-end delay which in certain configurations may cause some degradation of the tracking capabilities of the overall system. Fortunately, however, there are versions where this delay does not hurt the capabilities of the adaptation technique applied.

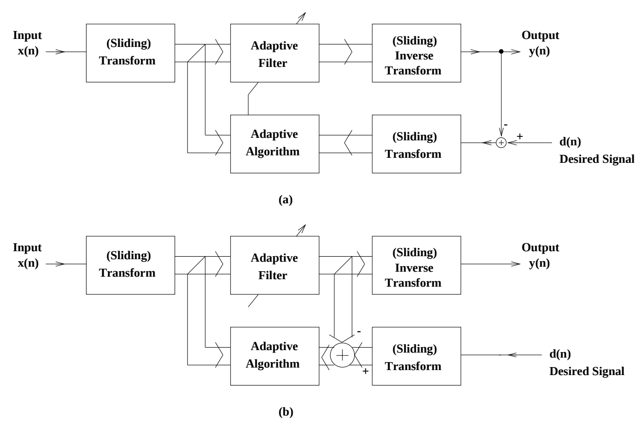

Figure 1. Frequency-domain adaptive filter configurations

1. INTRODUCTION

In recent years transform-domain adaptive filtering methods became very popular especially for those applications where filters with very long impulse responses are to be considered [1]. The basic idea is to apply the fast Fourier Transformation (FFT) for signal segments and to perform adaptation in the frequency domain controlled by the FFT of an appropriate error sequence. There are several algorithms based on this approach [1] and further improvements can be achieved ([2]). The formulation of the available methods follows two different concepts. The first one considers transformations as a "single" operation to be performed on data sequences (block-oriented approach), while the other emphasizes the role of multirate analyzer and synthesizer filter-banks.

In this paper, using some very recent results concerning fast sliding transformation algorithms ([3],[4]), a link is developed which helps to identify the common elements of the two approaches. The fast sliding transformers perform exact transformations, however, operate as polyphase filter-banks. These features together may offer further advantages in real-time applications, especially when standard DSP processors are considered for implementation.

There are two major configurations for transform-domain adaptive filtering (See Figure 1). First let us consider standard (not sliding) transformers. Here transformers perform a serial to parallel conversion while inverses convert to the opposite direction. Adaptation is controlled by the transformed input and error signals once for each input block, i.e. decimation is an inherent operation within these algorithms. Investigations related to the block-oriented approach, however, rarely consider real-time aspects of signal processing. Typically it is supposed that sampling frequency is relatively low compared to the computational power of the signal processors, and therefore, if a continuous flow of signal blocks must be processed, the block-period is enough to compute the transformations and the filter updating equations. Moreover, in the case of the widely used Overlap-Save and Overlap-Add methods (see e.g. [1]) filter updating requires further transformations, therefore further computational power is needed.

Filter-bank and consequently fast sliding transformation techniques do not offer extra savings in computations, however, they may provide a much better distribution of the computational load with time. This means that potentially they give better behavior when real-time requirements are to be met. The paper is organized as follows: Section 2 describes the basic idea of the fast sliding transforms while Section 3 is devoted to review the concepts of transform-domain adaptive filtering. The novelty of the paper is introduced in Section 4 where the combined structure is characterized.

[1]This work was supported by the Hungarian Fund for Scientific Research (OTKA) under contract T 017448, and TEMPUS under contract S_JEP-08333-94.

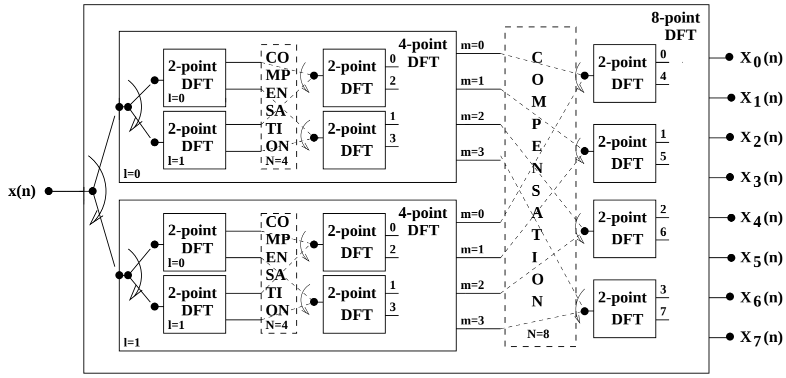

Figure 2. Polyphase DFT analyzer for $N = 8$

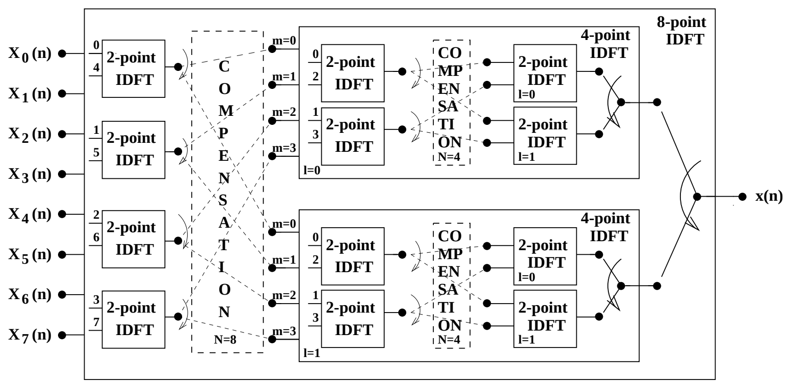

Figure 3. Polyphase DFT synthesizer for $N = 8$

2. FAST SLIDING TRANSFORMATIONS

Recently a fast implementation of the recursive Discrete Fourier Transformation (DFT) has been proposed [3] which combines the idea of polyphase filtering and the Fast Fourier Transformation (FFT) algorithm. Figs. 2 and 3 show the analyzer and the synthesizer DFT filters, respectively. The operation can be easily understood if we observe that e.g. the analyzer at its input follows the decimation-in-time while at its output the decimation-in-frequency principle. The computational complexity of these structures is in direct correspondence with that of the FFT and its parallel nature provides additional advantages in parallelization.

The analyzer bank can be operated as well as a sliding-window DFT or as a block-oriented transformer. This latter means that the parallel outputs are maximally decimated as it is typical with serial to parallel converters. However, if overlapped data segments are to be transformed, the structure is well suited to support decimation by any integer number. The widely used Overlap-Save Method (see Section 3) concatenates two blocks of size N to perform linear convolution and calculates $2N$-point FFTs. A $2N$-point sliding transformer can easily produce output in every Nth step.

In certain applications it may be advantageous to produce signal components instead of the Fourier coefficients as it is dictated by the definition of the DFT. In this case the filter-bank is a set of band-path filters with center frequencies corresponding to the Nth roots of unity values. Such DFT filter-banks can be easily derived using the ideas valid for the DFT transformers. With this DFT filter-bank approach transform-domain signal processing can have the following interpretation: The input signal to be processed first is decomposed by a filter-bank into components and the actual processing is performed on these components. The modified components enter into a N-input single output filter-bank (the so-called synthesizer bank) which produces the output sequence.

3. TRANSFORM-DOMAIN ADAPTIVE FILTERING

The concept of transform-domain adaptive filtering offers real advantages if adaptive FIR filters with very long impulse responses are to be handled. The first important aspect is the possible parallelization described above achievable using fast transformation algorithms. The second is the applicability of block adaptive filtering: the parallel channels enable decimation and therefore a coefficient update only once in every Nth step. In the meantime, however, a much better gradient estimate can be derived.

Figure 1 shows two possible forms of frequency-domain adaptive filtering. In the first version adaptation is controlled by the time-domain difference of the filter output and the desired signal. The adaptive filter performs N multiplications using the N-dimensional weighting vector generated by the adaptive algorithm. From the viewpoint of this paper the adaptation mechanism can be of any kind controlled by an error signal, however, in the majority of the applications the least-mean-square (LMS) algorithm is preferred (see e.g. [1]) for its relative simplicity. Adaptation can be performed in every step, however, drastic reduction of the computations can be achieved only if the transformer outputs are maximally decimated. The techniques developed for this particular case are the so-called block adaptive filtering methods. [1] gives a very detailed analysis of the most important approaches. It is emphasized that the classical problem of linear versus circular convolution appears also in this context. This problem must be handled because in the majority of the applications a continuous data flow is to be processed and therefore the dependence of the neighboring blocks can not be neglected without consequences. The correct solution is either the Overlap-Save or the Overlap-Add Method. Both require calculations where two subsequent data blocks are to be concatenated and double-sized transformations are to be performed.

If we consider the system of Figure 1.a from timing point of view it is important to observe that at least two transformations must be calculated within the adaptation loop. If real-time requirements are also be fulfilled the time needed for these calculations may be a limiting factor. Moreover, if we investigate more thoroughly e.g. the Overlap-Save Method it turns out that the calculation of the proper gradient requires the calculation of two further transformations, i.e. there are altogether four transformations within the loop. This may cause considerable delay and performance degradation especially critical in tracking nonstationary signals.

The adaptation of the transform-domain adaptive filtering scheme on Figure 1.b is controlled by an error vector calculated in the transform-domain. Due to this solution the transformer blocks are out of the adaptation loop, therefore the delay within the loop can be kept at a lower level. Here the adaptation is completely parallel and is to be performed separately for every "channel". The operation executed in this scheme corresponds to the circular convolution which may cause performance degradation due to severe aliasing effects.

In the literature of the analysis filter-banks the so-called subband adaptive filters (see e.g. [1]) are suggested for such and similar purposes which at the prize of smaller $L < N$ decimation rates provide better aliasing suppression. The fast sliding transformers are in fact efficiently implemented special filter-banks. If they are maximally decimated they suffer from the side-effects of the circular convolution. In order to reduce aliasing the application of $L = N/2$ can be advised. If the the number of the adaptive filter channels is lower than the size of the sliding transformer the standard windowing techniques (see e.g. [7]) can be used for channel-filter design.

4. ADAPTIVE FILTERING WITH FAST SLIDING TRANSFORMERS

In this Section the timing conditions of the frequency-domain adaptive filters using the famous Overlap-Save Method are investigated. This technique is carefully described in [1] therefore here only the most critical elements are emphasized. Other techniques can be analyzed rather similarly.

Figure 4 shows the timing diagram of the standard block-oriented solution. Here the acquisition of one complete data block (N samples) is followed by the processing of this data vector. A continuous sequence of data blocks can be processed if the execution time of one block adaptation is less or equal to the corresponding acqusition time ($t_e \leq t_a$). One block adaptation consists of several steps, among them the transformation of the input and the error sequences, respectively (see Fig. 1.a). The generation of the output sequence requires an additional transformation. Due to the requirements of the linear convolution all these transformations work on double blocks, i.e. on $2N$ data points. The calculation of the gradient and coefficient update requires further two such transformations. A detailed analysis of the steps using the $2N$-point transformations shows that none of them is "complete", i.e. some savings in the computations are possible.

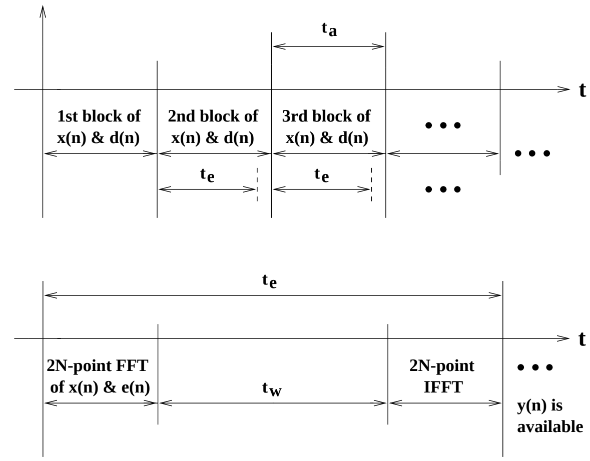

Figure 4. Timing diagram of the standard overlap-save frequency-domain adaptive algorithms (t_a denotes the acquisition time of one data block of N samples, t_e stands for the execution time of one block adaptation, and t_w for the calculation time of the gradient and the W update). The end-to-end delay is of N samples.

With the introduction of the sliding transformers the acquisition of the input blocks and the processing can be overlapped, since the sliding transformers can start working already before having the complete block. If we permit an end-to-end delay of $2N$ samples then the processing can be extended for three acqusition intervals (see Fig. 5). During the first interval the data acqusition is combined with the transformation, the second can be devoted for finishing the transformation, for updating the coefficient vector and to start the inverse transformation which can be continued in the third interval because the data sampling performed parallelly provides the $d(n)$ samples (see Fig. 1) sequentially. At the prize of larger end-to-end delay the execution time can be extended for more blocks, as well.

In the case of the Overlap-Save Method the gradient calculation consists of an inverse transformation, some simple manipulations and a transformation. Since with the sliding inverse transformation a parallel to serial conversion is performed and the sliding transformer implements a serial to parallel conversion, further overlapping in the execution is possible as it is indicated on Fig. 5.

The above considerations can result savings if the granu-

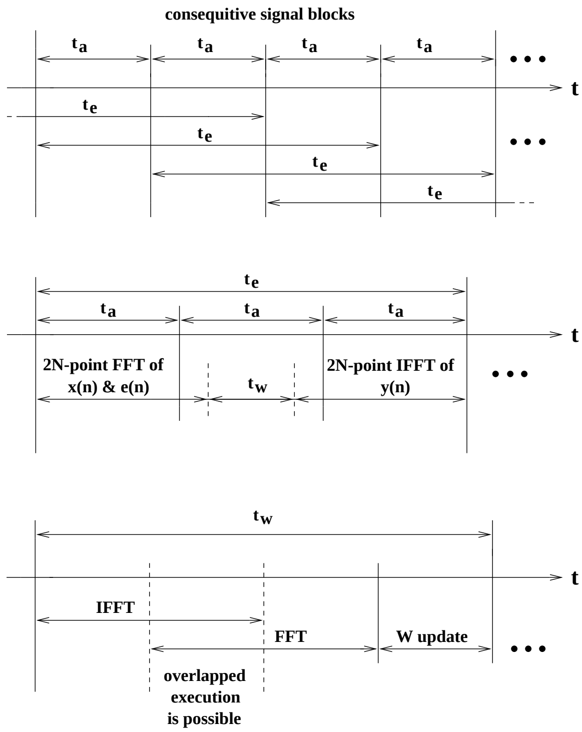

Figure 5. Timing diagrams of a possible frequency-domain adaptive algorithm using sliding transformations. (t_a denotes the acquisition time of one data block, t_e stands for the execution time of one block adaptation, and t_w for the calculation time of the gradient and the W update). The end-to-end delay is of $2N$ samples.

larity of the hardware and software elements of implementation enable smooth distribution of the computational load. If separate hardware units are available for the sliding transformations then the parallelism of the execution can be considerably improved. It is important to note that the precedence conditions of block processing do not support such parallelizations: the complete data block must be available for a block-oriented operation like a signal transformation.

To illustrate the achievable gain using fast sliding transforms here a comparison is made concerning the timing and computational conditions of the two approaches. In the usual implementations of the conventional method data acquisition and processing are separated in time, i.e. after the arrival of a complete input data block an efficient DSP program calculates the transformed values. The sliding FFT introduced in [3] has the same computational complexity as the traditional algorithm, however, in the proposed polyphase filter-bank data acquisition and processing overlap, therefore calculations start well before the complete block becomes available. There are very many ways to characterize the complexity of the FFT, one possible figure can be the number of complex multiplications. This figure for the maximally decimated sliding N-point FFT equals (see [3])

$$C1 = \frac{N}{4} \log_2 \left(\frac{N}{8} \right) + 1 \, , \qquad (1)$$

i.e. this figure is one of the possible characterizations. These operations, however, are performed partly during the data acqusition, and the number of complex multiplications which can not be executed before the arrival of the last sample of the block remains only

$$C2 = \frac{N}{2} - \log_2 N \, . \qquad (2)$$

As an example for $N = 1024$ $C1 = 1793$ and $C2 = 502$, i.e. the possible time gain due to the overlap can be considerable. Similar figures can be given for other operations, as well.

5. CONCLUSIONS

In this paper the transform-domain adaptive signal processing is investigated in that particular case where the necessary transformations are performed by fast sliding techniques, i.e. with filter-banks corresponding to fast transformations. The significance of these investigations is that with such techniques further parallelism can be achieved and utilized for applications where higher sampling rates are required. The solutions which follow the scheme of Fig.1.b. are in good correspondence with some successful multirate filter-bank techniques while the other group related to Fig.1.a. can be significantly improved. The prize to be paid for the additional parallelism is the increase of the overall end-to-end delay which requires further investigations concerning step-size and stability issues. This can be started following the ideas of [5] where a similar problem was to be solved. In this paper everywhere the term "transform-domain" was used instead of the explicit term "frequency-domain". The reason for this is that all the above developments can be extended for other type of transformations, as well. The application of other transformers may reduce either the computational load or in certain cases improve the adaptation performance.

REFERENCES

[1] J. J. Shynk, "Frequency-Domain and Multirate Adaptive Filtering", *IEEE Signal Processing Magazine*, pp. 15-37, Jan. 1992.

[2] H.-C. Chiang, J.-C. Liu, "Fast Algorithm for FIR Filtering in the Transform Domain", *IEEE Transactions on Signal Processing*, Vol. 44, No.1, pp.126-129, Jan. 1996.

[3] A. R. Várkonyi-Kóczy, "A Recursive Fast Fourier Transformation Algorithm", *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, Vol. 42, No.9, pp. 614-616, Sept. 1995.

[4] A. R. Várkonyi-Kóczy, "Multi-Sine Synthesis and Analysis Via Walsh-Hadamard Transformation", *Proc. of the 1996 IEEE International Symposium on Circuits and Systems, ISCAS'96*, May 12-15, Atlanta, Vol. 2, pp. 457-460.

[5] D. L. Jones, "Learning Characteristics of Transpose-Form LMS Adaptive Filters", *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, Vol. 39,, No. 10, pp. 745-749, Oct. 1992.

[6] R. E. Crochiere, L. R. Rabiner, *Multirate Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[7] L. R. Rabiner, B. Gold, *Theory and Application of Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1975.