AN EFFICIENT OVERLAP-ADD FREQUENCY DOMAIN ADAPTIVE FILTER STRUCTURE

by

Dr. JAKS Jayasinge,

1.0 Introduction

Adaptive finite impulse response (FIR) digital filters are extremely useful devices in many applications of digital signal processing including channel equalization, sensor array processing, echo and noise interference cancellation. For simplicity we will restrict ourselves to the echo cancellation problem of which the basic scheme is given in Fig.1. Signal x(k) is reflected via an echo path as an echo signal e(k). This echo signal, together with the desired signal s(k) is received. The adaptive filter uses a transversal model for the echo path and makes a replica ê(k) of the echo signal e(k). Theoretically the residual signal r(k) = s(k) + e(k) - ê(k) in steady state will almost be equal to the signal s(k).

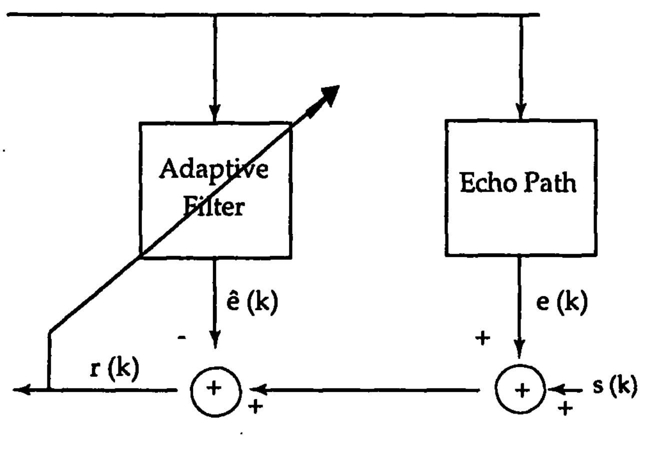

Figure 1: Adaptive Echo Canceler

One of the most popular Time Domain Adaptive Filters (TDAF) is a digital transversal filter using the Least Mean Square (LMS) algorithm to update the adaptive weights. These updates are made along the direction of an estimated gradient vector [1],[5]. When the filter length becomes very large, this filter has the disadvantage that the computational requirements in terms of multiply-adds per output sample increases linearly with the filter length. The computational complexity can be largely reduced by using a Frequency Domain Adaptive Filter (FDAF). This reduction is accomplished by replacing convolution in time domain with multiplication in frequency domain, where the transformations between time and frequency domain can be done very efficiently, e.g. by using an efficient implementation of the Discrete Fourier Transform (DFT); namely the Fast Fourier Transform (FFT). In FDAF configurations, the linear convolution is accomplished by a circular convolution. Because of this circular nature special overlap methods must be used. Two well known overlap methods are the overlap-save and the overlap-add method. Most of the literature on FDAF's deal with overlap-save method [2], hence we concentrate on overlap-add method. Fig.2a depicts the overlap-add method for fixed filter coefficients. The output signal ê(k) is the result of a linear convolution between the input signal x(k) and the impulse response . W'_i (for i = o, . . . , N-1) In all figures the DFT's are of length L + LL'. Furthermore signal paths with double lines refer to paths in frequency domain while signal lines refer to time domain signal paths. In the text we will use lower case characters for the time domain signals while upper case characters are used for frequency domain signals. Exceptions are the upper case letters L,L', N and M which are used to denote block lengths.

The overlap-add procedure is as follows (see Fig.2a):

Dr. J. A. K. S. Jayasinge, BScEng, PhD., MIEE, AMIE (SL), Senior Lecturer, Department of Civil Engineering University of Moratuwa.

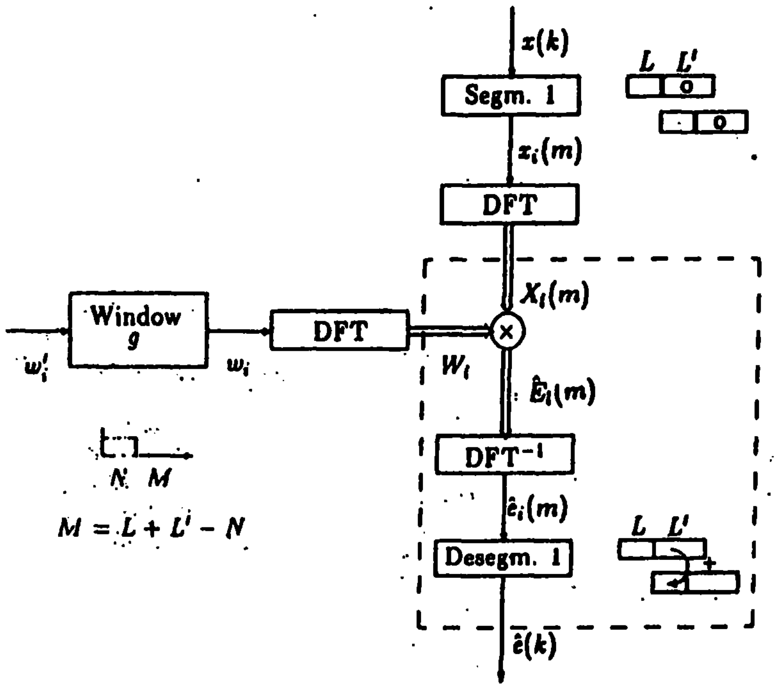

Figure: 2.a General Overlap-add Method Performed in Frequency Domain

1. Select every iteration a block of L new points from the input signal x(k) and augment this block by L/ zeros (Segml). The input signal is now described by blocks which are defined as:

$$X_i(m) = \begin{cases} X(mL = i) & \text{for } i = i, \ldots, l-1 \\ O & \text{for } i = l, \ldots, L+L'-1 \end{cases} \quad (1)$$

where m = 0,1,... is the block index.

2. Transform the blocks containing xi(m) to the frequency domain. The frequency bins are defined as:

$$X_l(m) = \sum_{i=o}^{L+L'-1} x_i(m)e^{-j\theta il} \text{ with } \theta = \frac{2\pi}{L+L'} \quad (2)$$

3. Augment the impulse response , which is of length N, with M = L+L'-N zeros (Window g) to get the block:

$$W_i = \begin{cases} W'_i & \text{for } i = O, \ldots N-1 \\ O & \text{for } i = N, \ldots L+L'-1 \end{cases} \quad (3)$$

These blocks are transformed to the frequency domain which results in:

$$W_l = \sum_{i=o}^{L+L'-1} w_i e^{-j\theta il} \quad (4)$$

4. The convolution is performed in the frequency domain by multiplying $X_l(m)$ with W_l for l = 0,...,L+L'-1.

5. Transform these blocks to time domain by the inverse DFT (DFT^{-1}) which results in the blocks:

$$\hat{e}i(m) = \sum_{1=0}^{L+L'-1} \hat{E}1(m)e^{j\theta il} \quad (5)$$

6. By performing a desegmentation (Desegm 1) on these blocks the output signal is calculated for m = 0,1,... as:

$$\hat{e}(mL+i) = \begin{cases} \hat{e}_i(m) + \hat{e}_{i+L}(m-1) & \text{for } i = O, \ldots, N-2 \\ \hat{e}i(m) & \text{for } i \quad N-1, \ldots, L-1 \end{cases} \quad (6)$$

where m = 0,1,... and $\hat{e}_{i+L}(-1) = 0$ for i = 0,...,N - 2.

The requirements on the block length L and number of augmented zeros L' are:

$L \geq N$ to get an addition of two blocks as given in [4] (7)

$L' \geq N - 1$ to avoid wraparound errors

The output signal $\hat{e}(k)$ in Fig.2a is the linear convolution between the (infinite length) input signal x(k) and the (finite length) impulse response . W'_i We note here that a correlation between an infinite length signal and a finite length sequence can be carried out in a similar way. The only difference is that we have to conjugate the signal $X^l(m)$ before it enters the dashed box.

In adaptive filter configurations, the impulse response W'_i is updated every iteration. For this reason, a simple addition of $\hat{e}_i(m)$ and $\hat{e}_{i+L}(m-1)$, as given in equation (6), is not possible in adaptive filters. A possible implementation of the overlap-add method in adaptive filter configurations is given in Fig. 2b. In this figure the impulse response is time varying, which implies that all weights do have a time index, e.g. $W_l \rightarrow W_l(m)$. The multiplication in frequency domain has to be carried out separately for $W_l(m)$ with $X_l(m)$ and $W_l(m)$ with $X_l(m-1)$. The result of $W_l(m) \cdot X_l(m-1)$ in time domain is circularly shifted over L′ points to the right, and after that a Window v selects the first N-1 points and augments these with M+1 = L+L′ -N+1 zeros. These results are added to the results of $W_l(m) \cdot X_l(m)$ in time domain which gives a block that contains L correct linear convolution points at the first L places.

2.0 Overlap-add implementation of an FDAF with 7 DFT's

In [2] (Fig.4, page 1080) an overlap-add implementation of an FDAF is given for the case L = L=L′ = N. Here we first give a more general overlap-add implementation of an FDAF, which is depicted in Fig.3, by using the previous results.

Denoting the time domain adaptive weights by the adaptive filter has to perform a linear convolution between the input signal x(k) and these weights. This convolution is performed in the dashed box "CONVOLUTION" in Fig.3, where we used the results of Fig.2b.

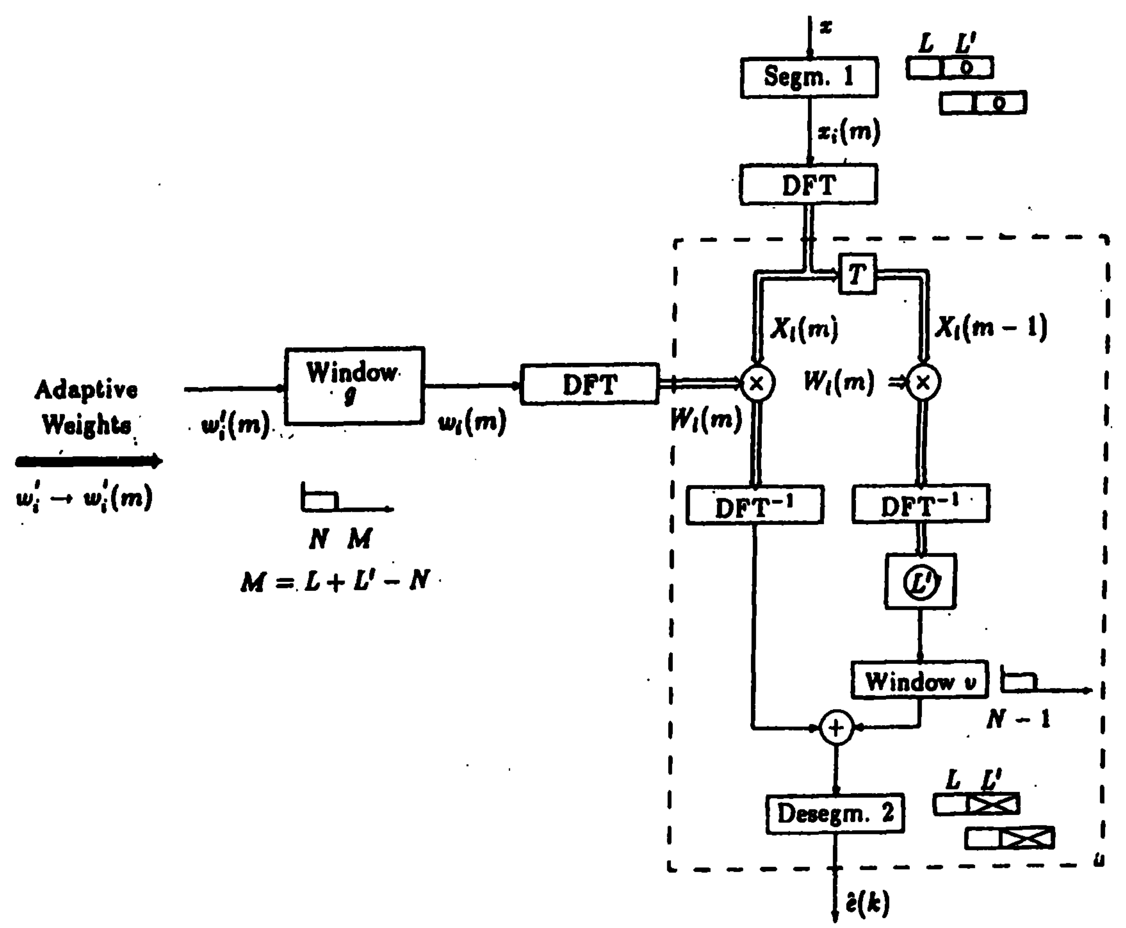

Fig: 2.b Implementation of Figure. 2a in Adaptive Filter Configurations

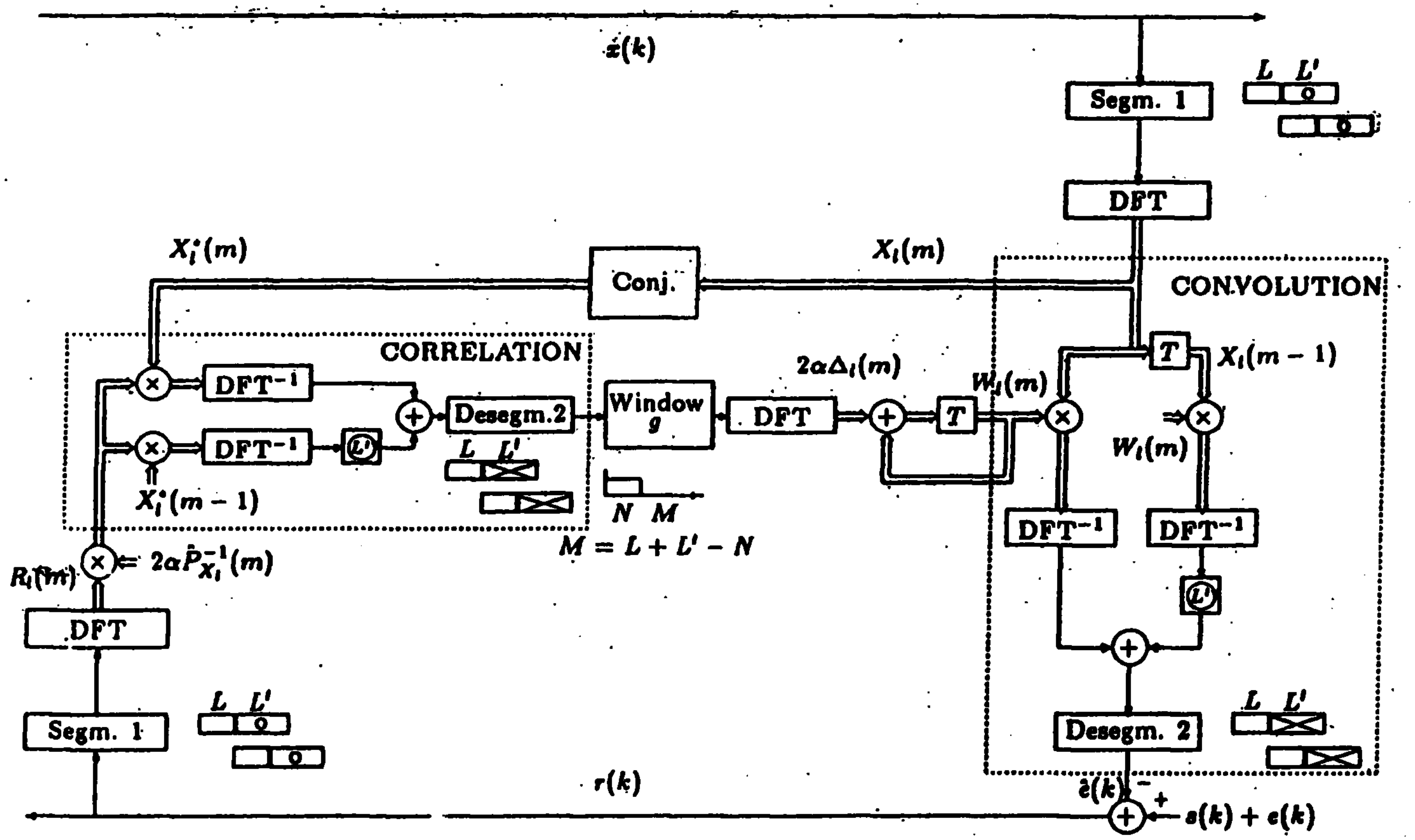

Figure 3: Overlap -add Impementation of an FDAF with FFT's

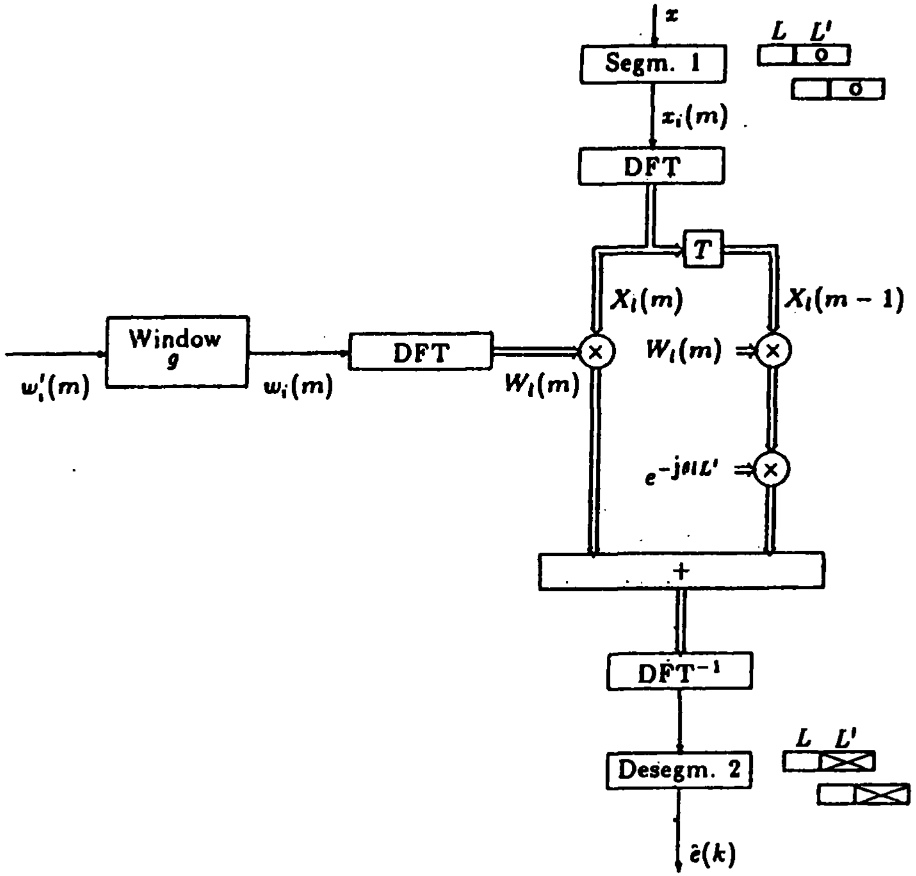

Figure: 4a Overlap-add Method of Fig .2b with Circular Shift Over L´ Points and Addition Performed in frequency Domain

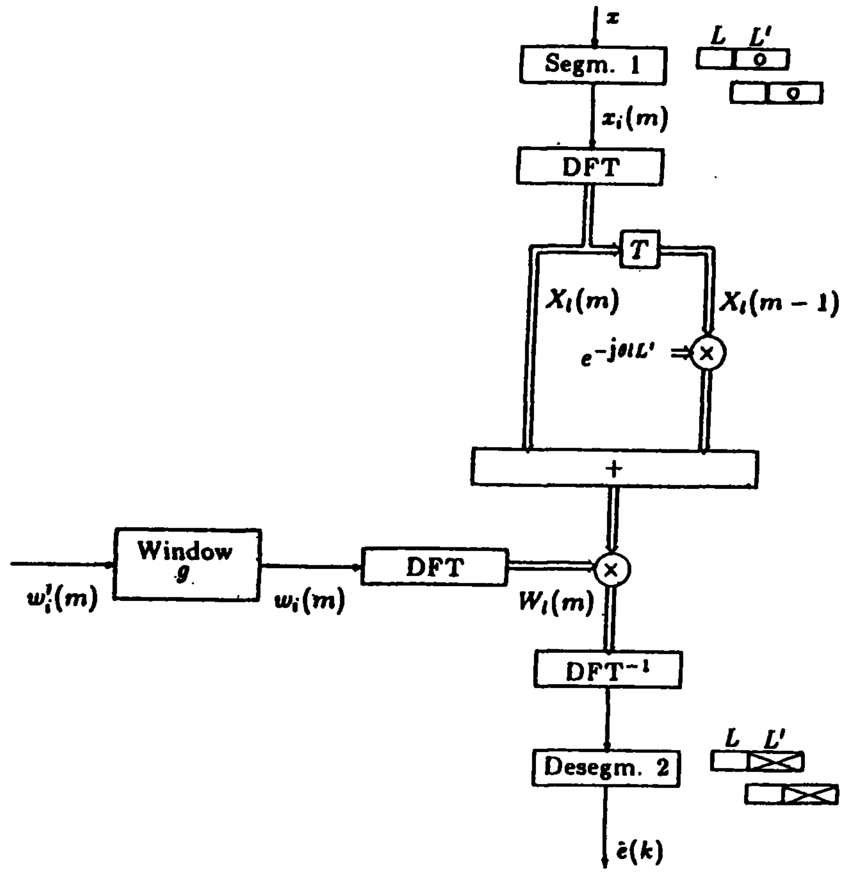

Figure: 4b Equivalent Structure of Fig 4a

The principle of the FDAF is to update the weights as long as there is correlation between the input signal x(k) and the residual signal r(k). In adaptive filter configurations however both the input signal x(k) and the residual signal r(k) are of infinite length. Since it is practically impossible to calculate the correlation between two infinite sequences, we assume r(k) as the finite sequence. Every iteration L new points of the signal r(k) are available, which is the result of the desegmentation box Desegm2 in Fig.3. For this reason we choose the length of the finite sequence r(k) to be equal to L points. As mentioned before the correlation of an infinite signal with a finite sequence can be calculated in a similar way as the convolution just by applying the complex the conjugate $Xl^*(m)$ as input to the dashed box "CORRELATION" in Fig.3. An additional requirement to (7) will now be $L/ \geq L-1$ to get no wraparound errors in the calculation of the correlation. For this reason Window v of Fig.2b is not needed. The requirement (7) on the block length L and the number of augmented zeros L/ for adaptive filter configurations using the overlap-add method now becomes:

$$L \geq N \quad \text{and} \quad L' \geq L-1 \tag{8}$$

Furthermore, we note that we have calculated L correlation points from which only N points are needed. The Window g is now used to take the first N points out of L+L′ points and augment these with M = L+L′-N zeros. It is obvious that the operations of the desegmentation (Desegm2) and the Window g can be combined in the Window g.

By decorrelating the input signal the convergence speed can be accelerated [5]. This can be accomplished by normalizing the input power spectrum. In Fig.3 this is done by multiplying the frequency bins $R_l(m)$ of the residual signal with the inverse of an estimate of the l^{th} input power spectral bin$P_{xl}^{\wedge -1}$ (m). This power normalization however is out of the scope of this paper, hence we will not go into further detail on that point.

The update algorithm is now given by:

$$W_l(m+1) = W_l(m) + 2a\Delta_l(m) \tag{9}$$

Where is the adaptation constant(m) and $\Delta\tau$ is an estimate of the gradient.

The complexity of the FDAF is roughly made by the number of DFT's. From Fig.3 we see that the complexity of the FDAF using the overlap-add method is in the order of 7 DFT's.

3.0 Problem formulation

From literature [2] it is known that the complexity of an FDAF configuration using the overlap-save method is in the order of 5 DFT's. It is also stated in literature [2] (p1073) that the FDAF configuration using the overlap-save method is to be preferred above the one using the overlap-add method, because of complexity reason. As shown in Section 2, overlap-add method needs 7 DFT's.

4.0 New overlap-add implementation of an FDAF with 5 DFT's

In this section, we will first take a closer look to the overlap-add method separately. For this we make use of the requirements given in (8) for adaptive filter configurations using the overlap-add method. We refer to Fig.2b and note that the Window v is not needed here. By using the property that a circularly shift over L′ points in time-domain is equal to a multiplication in frequency domain with $e^{-j\theta iL'}$, we can derive Fig.4a from Fig.2b, where we also used the linearity property of the DFT. We note here that in Fig.4a only one inverse DFT is needed while we need two of them in Fig.2b. This is the main reason of reducing the number of DFT's in an overlap-add implementation of an FDAF.

Since both $X_l(m)$ and $X_l(m-1)$ are multiplied by $W_l(m)$, we can further reduce Fig.4a by shifting this multiplication with $W_l(m)$ after the addition point. This results in Fig.4b.

Using the above results in the dashed boxes of the FDAF configuration in Fig.3, leads to the FDAF configuration which is given in Fig.5. This new overlap-add implementation contains 5 DFT's while the original one (Fig.3) which was derived from [2] contains 7 DFT's.

5.0 Implementation

In most practical situations, we will use an efficient realization of the DFT, namely the Fast Fourier Transform (FFT). A requirement for this is that the length of the transformation L+L′ has to be a power of two. A possible choice is L = L′. In this case the length of the FFT's is 2L while moreover the multiplication with $e^{-j\theta iL'}$ becomes very simple, namely:

$$e^{-j\theta iL'} \Big/_{L=L'} = e^{-j\frac{2\pi}{2L}lL} \quad (-1)^l \tag{10}$$

This implies that even frequency bins are added to their previous values, while for the odd bins we have to subtract their previous values.

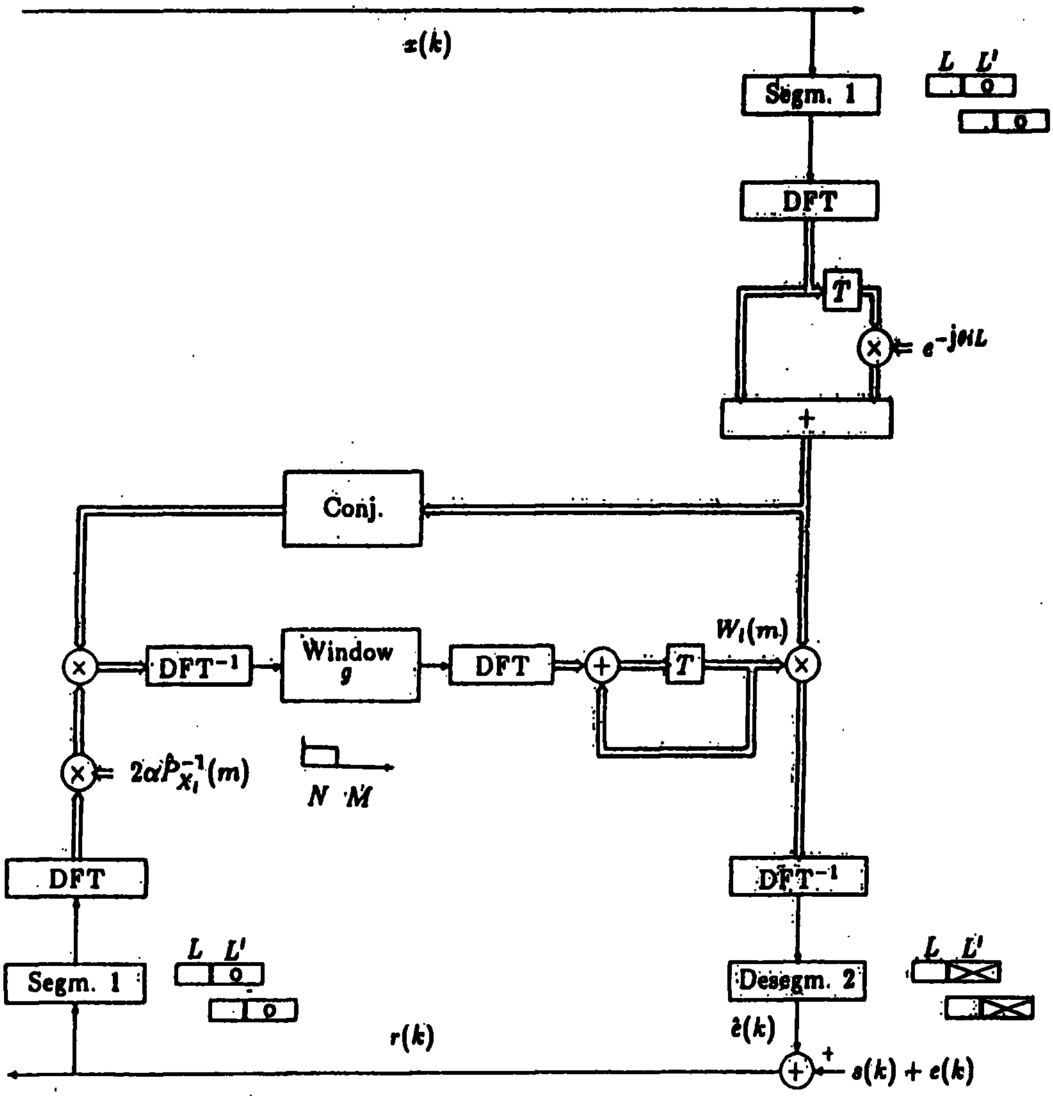

Figure 5: New Overlap-add Implementation of an FDAF with 5 FFT's

As a final comment of this section we note that when shifting the multiplication of the factor $e^{-j\theta L'}$ and the addition to the input signal (time domain), the segmentation of the input signal becomes different and the scheme is "overlap-save" like. However the segmentation and desegmentations of Fig.3 and Fig.5 are equivalent. For this reason, it is obvious that the new FDAF configuration of fig.5 is an overlap-add implementation.

6.0 Conclusions

In contrary to the statement in [2], we have introduced an FDAF configuration using the overlap-add method which has equal complexity as the FDAF using the overlap-save method. In Fig.5 we can see that the segmentation and desegmentations are "equivalent", which is not the case for the overlap-save configuration. This implies that the configuration using the overlap-add method has DFT's which have more "equivalent" structures. This "equivalency" make the overlap-add configuration more efficient.

7.0 References

1. Benesty J., Duhamel P, "A Fast Exact Least Mean Square Adaptive Algorithm", IEEE Trans. on Signal Processing, vol. 40, no. 12, dec. 1992, pp 2904-2920.
2. Clark G.A., Parker S.R., Mitra S.K, " A unified approach to time and frequency-domain realization of FIR adaptive digital filters", IEEE Trans. on ASSP, vol. ASSP-31, no.5, oct. 1983, pp 1073-1083.
3. Markel J.D., "FFT Pruning", IEEE Trans. on Audio and Electroacoustics, vol. AU-19, no.4 dec. 1971, pp 305-311.
4. Pelkowitz L., "Frequency Domain Analysis of Wraparound Error in Fast Convolution Algorithms", IEEE Trans. on ASSP, vol. ASSP-29, no.3, June 1981, pp 413-422.
5. Widrow B., Stearns S.D., "Adaptive Signal Processing", Prentice Hall Inc., Englewood Cliffs, New Jersey 07632, 1985.