Echo Cancellation Techniques for Multimedia Applications - a Survey

by Rainer Storn[1)]

TR-96-046

November, 1996

Abstract

The problem of acoustical echo in a headset-free full duplex communication environment is explained and the potential solutions are sketched. The different methods for acoustic echo cancellation (AEC) via adaptive filters are outlined and their suitability for a 16-bit fixed point implementation on a digital signal processor (DSP) is evaluated. The current prototype for the ICSI Acoustic Echo Canceller (IAEC) which uses an allpass-based subband adaptive approach is introduced and directions for future work are given.

[1)]International Computer Science Institute, 1947 Center Street, Berkeley, CA 94704-1198, Suite 600, Fax: 510-643-7684. E-mail: storn@icsi.berkeley.edu. On leave from Siemens AG, ZFE T SN 2, Otto-Hahn-Ring 6, D-81739 Muenchen, Germany. Fax: 01149-89-636-44577,
E-mail: rainer.storn@zfe.siemens.de.

1. Introduction

The occurrence of echoes is a well known problem for long distance communication links and has been experienced in the telephone network due to the imperfections of the telephone hybrid [1]. The telephone hybrid is unable to completely prevent the incoming speech signal from traveling back to the originator, resulting in echo which the talker hears. An even more challenging problem occurs in hands free telephones and multimedia communication equipment [2], [3] where echoes due to room reverberations occur. An illustration of this problem can be seen in fig. 1.

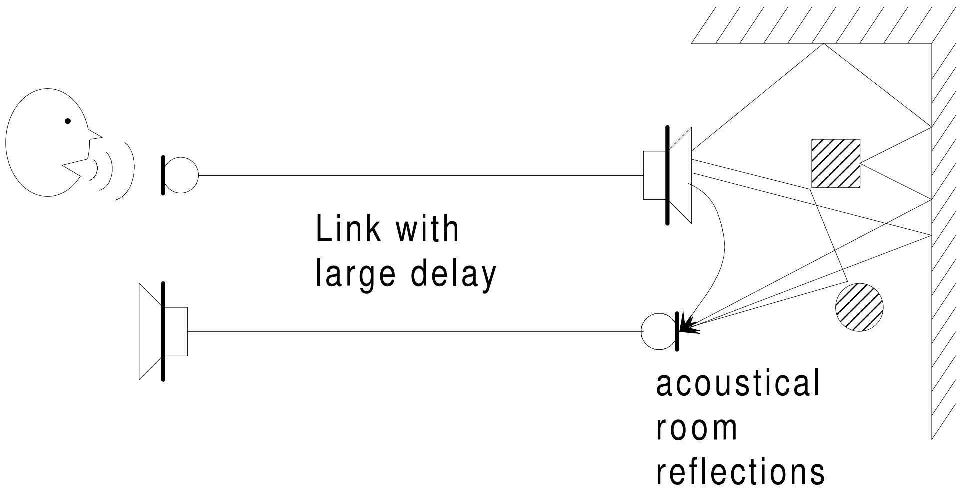


Fig. 1: Illustration of the acoustical echo generation process.

2. Headsets

The simplest method to cope with acoustical echoes is to prevent the remote speech signal from entering the room by using headsets. Headsets can be advantageous, especially in a noisy environment or when the user doesn't want to disturb other persons in the room. Headsets also help to prevent acoustical eavesdropping which can be important if the user communicates while being in public places like trains, hotel lobbies etc.. On the other hand, headsets are not always convenient to carry and constitute additional equipment which, for example, notebook users are reluctant to accept. For car phone applications headsets are not useful either. In virtual reality conferencing the conferencing scenario should be as close as possible to a real conference, and headsets do not support this. In addition headsets hinder the user from moving freely unless it is wireless, e.g. via infrared. A further disadvantage is that the user who takes the inconvenience of wearing the set suppresses the echo of his partners but not of himself. If just one partner refuses to wear a headset the other users hear echo.

3. Echo suppression

An echo suppressor assumes that the communication is essentially half duplex, i.e. there is only one person talking at a time. The echo suppressor detects the talking and the silent party and shuts down the microphone of the latter, so that the signal of the loudspeaker in the silent person's room doesn't travel back to the person who talks.

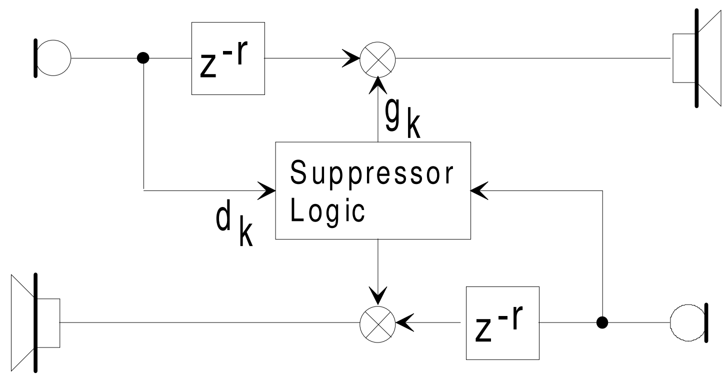


Fig. 2: Simple echo suppressor.

During double talk both microphones are usually opened up causing acoustical echo. Other disadvantages are cutting of beginnings and endings of words, as well as a lack of perceived naturalness of the communication.The advantage of an echo suppressor is its simplicity so that it can be easily implemented in SW.

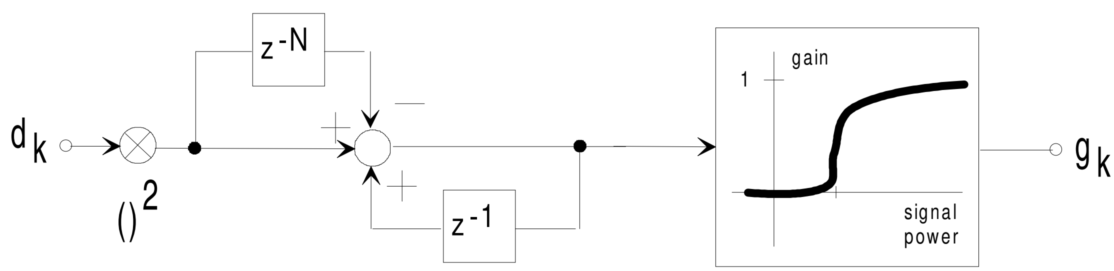


Fig. 3: Main functional unit in the echo suppressor of fig. 2.

Yet speech recognition has done research on this topics decades ago and has found methods how to detect the boundaries of speech signals [4]. These detection methods impose some delay on the audio system which, however, can be kept within reasonable bounds for real time applications. If the on/off switching of the microphone is replaced by a more smooth method of adjusting the gain of the microphone, the quality of echo suppressor can be improved [5]. The echo suppression method can be seen as a "poor man's version" of getting rid of the long distance echo and is of importance for low end workstations. The power estimating filter in fig. 3 generally covers a time window of roughly 75ms [4] which corresponds to 500 taps in case of 8kHz sampling freuency.

4. Acoustic Echo Cancellation

4.1 Echo cancellation of one's own echo

The benefit of this approach is to the owner of the echo canceler board who is able to get rid of his own echo by adding an echo canceler board to his equipment. The basic block diagram is depicted in fig. 4 where an adaptive filter H(z) models the path that the microphone signal is taking. This path, however, can be very complicated, especially for multiparty conferences. Long distance communications can add a substantial delay, and imperfect connections can cause packet loss as well as signal jitter, the latter two of which make it virtually impossible for H(z) to adapt.

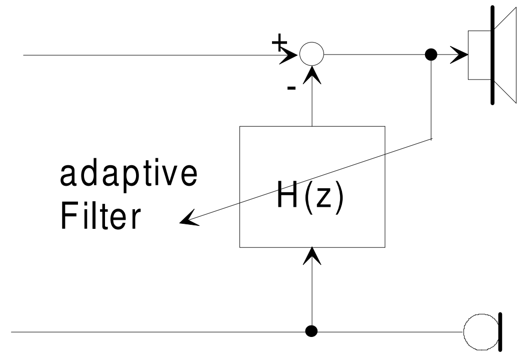


Fig. 4: Echo canceler which cancels the echo of the owner of the canceler.

4.2 Echo cancellation of the partner's echo

Technically this is the most elegant way of an Acoustic Echo Canceler (AEC) the basic idea of which is depicted in fig. 5.

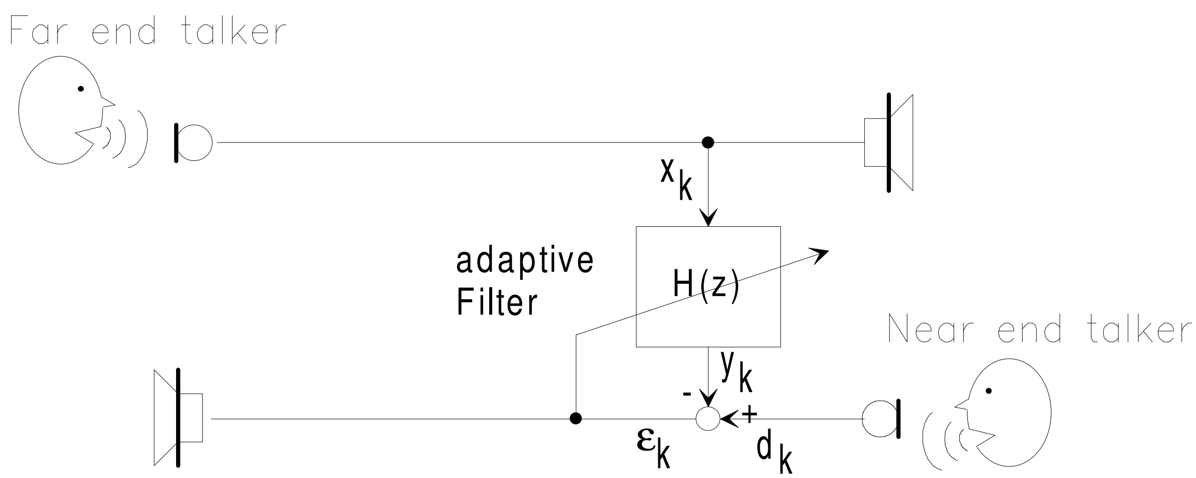


Fig. 5: Echo canceler which cancels the echo of the partner.

The basic idea in fig. 5 is to take the signal of the loudspeaker and subtract it from the microphone signal so that only those components are remaining which stem from signal other than the loudspeaker's [4]. This way the far end talker doesn't hear his own echo. The loudspeaker signal has to be filtered by an adaptive filter H(z) which models the room acoustics. It has been shown that the transfer function of a room changes considerably even if a single chair is changed in the room. A person who is moving around in the room causes even greater changes in the transfer function [3]. Hence H(z) must be able to track the ever changing room acoustics even under adverse conditions like noise or near end talk. The above solution has the drawback that, depending, on the room, the adaptive filter H(z) might be a filter of high degree if an FIR approach is used. This often doesn't allow the filter to be implemented in SW on a general purpose processor alone [6]. An additional signal processor board is usually necessary.

Or subsequent discussion will concentrate on the solution shown in fig. 5.

4.3 Further economic considerations

Since the cancellation of the partner's echo is technically the only viable approach, it is mandatory that all participating conference partners use an AEC. To make this scenario more likely, the AEC must have as low a cost as possible. This in turn makes it highly desirable that the AEC be implementable in fixed point arithmetic since fixed point Digital Signal Processors (DSPs) are much cheaper than floating point DSPs mainly because of the smaller chip area. Hence the AEC-algorithms must be numerically robust and should not require high accuracy computations.

5. The adaptive linear combiner

One of the most common approaches to model the acoustical echo path is to use the adaptive linear combiner (ALC) which is a transversal digital filter with adjustable coefficients as shown in fig. 6. The coefficient update is performed via some adaptation algorithm, details of which will be described later.

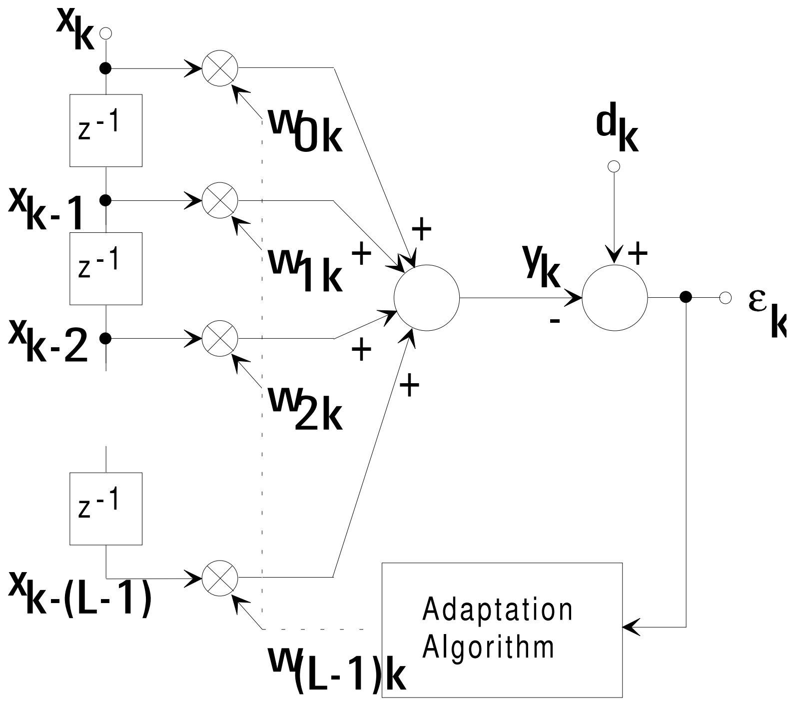


Fig. 6: Basic principle of the adaptive linear combiner.

The output of the adaptive linear combiner can be computed via

$$y_k = \sum_{i=0}^{L-1} w_{ik} \cdot x_{k-i} = \vec{W}_{L,k}^T \cdot \vec{X}_{L,k} = \vec{X}_{L,k}^T \cdot \vec{W}_{L,k} \tag{1}$$

where e.g. $\vec{W}_{L,k}^T$ is the row vector

$$\vec{W}_{L,k}^T = (w_{0k}, w_{1k}, \ldots, w_{(L-1)k}) \tag{2}$$

and $\vec{X}_{L,k}$ is the coloumn vector

$$\vec{X}_{L,k} = \begin{pmatrix} x_k \\ x_{k-1} \\ . \\ . \\ x_{k-(L-1)} \end{pmatrix}. \tag{3}$$

The error signal ε_k is computed as the difference of the desired microphone signal d_k and the filtered loudspeaker signal y_k according to

$$\varepsilon_k = d_k - y_k = d_k - \vec{X}^T_{L,k} \cdot \vec{W}_{L,k} \;. \tag{4}$$

Note that the terminology "error signal" for ε_k, which stems from adaptive filter theory, is somewhat misleading in our case as ε_k is much more than just an error. Instead ε_k is is the desired microphone signal d_k deprived of the echo contribution. In fact, the real desired signal is ε_k, not d_k, but again the terminology from adaptive filter theory doesn't quite fit the current application.

Note also how we consistently use the subscripts L,k to indicate the order L and the time index k of the vectors, or, more generally, matrices. We will need this notation later to better understand some of the fast adaptation algorithms.

If we square (4) and make use of (1), we get

$$\begin{aligned} \varepsilon_k^2 &= (d_k - \vec{X}^T_{L,k} \cdot \vec{W}_{L,k})^2 \\ &= d_k^2 + \vec{X}^T_{L,k} \cdot \vec{W}_{L,k} \cdot \vec{X}^T_{L,k} \cdot \vec{W}_{L,k} - 2d_k \cdot \vec{X}^T_{L,k} \cdot \vec{W}_{L,k} \\ &= d_k^2 + \vec{W}^T_{L,k} \cdot \vec{X}_{L,k} \cdot \vec{X}^T_{L,k} \cdot \vec{W}_{L,k} - 2d_k \cdot \vec{X}^T_{L,k} \cdot \vec{W}_{L,k} \end{aligned} \tag{5}$$

where the product $\vec{X}_{L,k} \cdot \vec{X}^T_{L,k}$ is an LxL matrix as an Lx1 vector is multiplied by an 1xL vector.

Assuming that all signals are stationary, minimizing the mean squared error (MSE) $\xi = E\left[\varepsilon_k^2\right]$, which is the output signal power, should constitute a good adaptation criterion. Intuitively this makes sense as the complete elimination of the echo contribution should render the minimum possible signal power. Unfortunately the signals in AEC-applications are not stationary and not noise free. Short-time stationarity, however, can usually be assumed and justifies to continue working with $E\left[\varepsilon_k^2\right]$. If we use (5) to compute $E\left[\varepsilon_k^2\right]$ we obtain

$$\begin{aligned} \xi = E\left[\varepsilon_k^2\right] &= E\left[d_k^2\right] + \vec{W}^T_{L,k} \cdot E\left[\vec{X}_{L,k} \cdot \vec{X}^T_{L,k}\right] \cdot \vec{W}_{L,k} - 2E\left[d_k \cdot \vec{X}^T_{L,k}\right] \cdot \vec{W}_{L,k} \\ &= E\left[d_k^2\right] + \vec{W}^T_{L,k} \cdot \underline{R}_{L,k} \cdot \vec{W}_{L,k} - 2\,\vec{P}_{L,k} \cdot \vec{W}_{L,k} \end{aligned} \;. \tag{6}$$

$\underline{R}_{L,k}$ is the so called input correlation matrix

$$\underline{R}_{L,k} = E\begin{bmatrix} x_k^2 & x_k \cdot x_{k-1} & \dots & x_k \cdot x_{k-(L-1)} \\ x_{k-1} \cdot x_k & x_{k-1}^2 & & \\ . & & & \\ x_{k-(L-1)} \cdot x_k & & & x_{k-(L-1)}^2 \end{bmatrix} \tag{7}$$

which is symmetric and has size LxL. $\vec{P}_{L,k}$ is the crosscorrelation vector

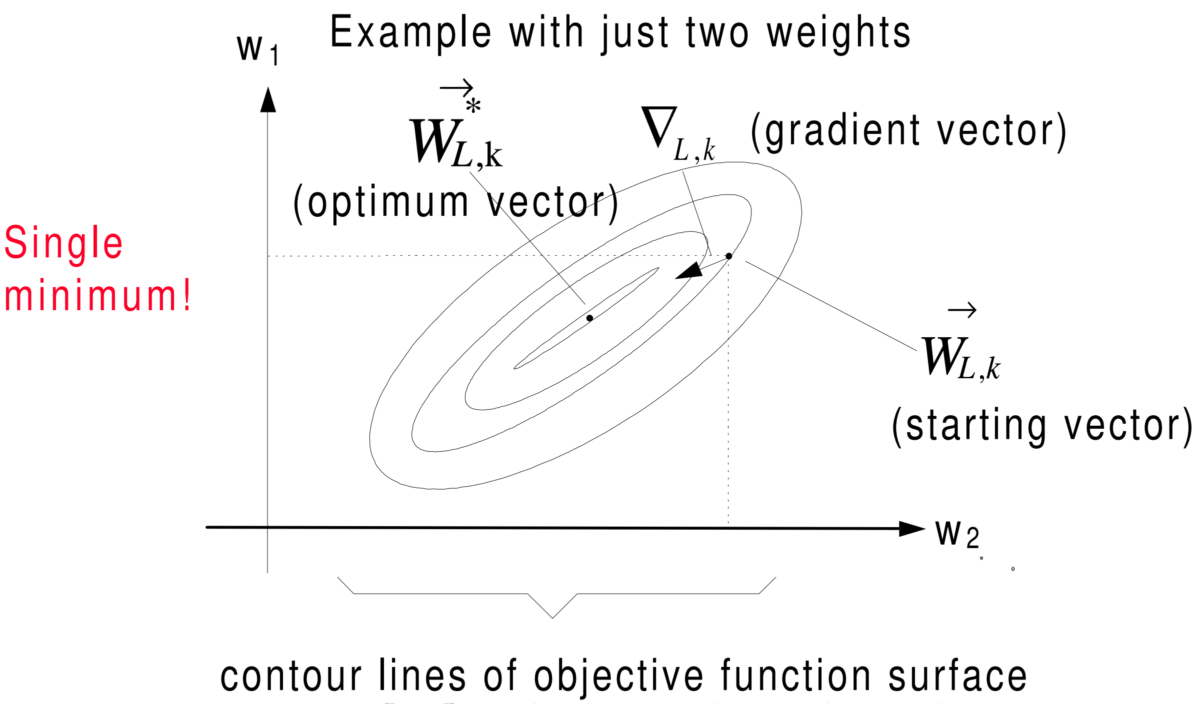
$$\vec{P}_{L,k} = E\begin{bmatrix} d_k \cdot x_k \\ d_k \cdot x_{k-1} \\ \dots \\ d_k \cdot x_{k-(L-1)} \end{bmatrix}. \tag{8}$$

One can easily see from (6) that the mean-square error ξ is precisely a quadratic function of the components of the weight vector $\vec{W}_{L,k}$, i.e. when (6) is expanded, the elements of $\vec{W}_{L,k}$ will appear in first and second degree only. The quadratic error function ξ is also designated as the "performance surface" which must always be >0 as a signal power can never be negative. The minimum of this performance surface can be computed analytically by setting the gradient to zero. As ξ is quadratic there can only be one extremum which has to be a minimum as negative values of ξ are not allowed. Fig. 7 illustrates the quadratic performance surface.

The gradient of ξ is given by

$$\begin{aligned} \nabla_{L,k} &= \frac{\partial \xi}{\partial \vec{W}_{L,k}} = \left[\frac{\partial \xi}{\partial w_{0k}}, \frac{\partial \xi}{\partial w_{1k}}, \dots, \frac{\partial \xi}{\partial w_{(L-1)k}} \right]^T \\ &= \frac{\partial E\left[d_k^2\right]}{\partial \vec{W}_{L,k}} + \frac{\partial \vec{W}_{L,k}^T \cdot \underline{R}_{L,k} \cdot \vec{W}_{L,k}}{\partial \vec{W}_{L,k}} - 2\frac{\partial \vec{P}_{L,k} \cdot \vec{W}_{L,k}}{\partial \vec{W}_{L,k}} \\ &= 0 + \left(\frac{\partial \vec{W}_{L,k}^T}{\partial \vec{W}_{L,k}} \cdot \underline{R}_{L,k} \cdot \vec{W}_{L,k} + \vec{W}_{L,k}^T \cdot \underline{R}_{L,k} \cdot \frac{\partial \vec{W}_{L,k}}{\partial \vec{W}_{L,k}} \right) - 2\,\vec{P}_{L,k} \cdot \frac{\partial \vec{W}_k}{\partial \vec{W}_k} \\ &= \underline{R}_{L,k} \cdot \vec{W}_{L,k} + \vec{W}_{L,k}^T \cdot \underline{R}_{L,k} - 2\,\vec{P}_{L,k} \end{aligned}$$

yielding $$\nabla_{L,k} = 2\underline{R}_{L,k} \cdot \vec{W}_{L,k} - 2\,\vec{P}_{L,k}. \tag{9}$$



$$\xi = E\left[d_k^2\right] + \vec{W}^T_{L,k} \cdot \underline{R}_{L,k} \cdot \vec{W}_{L,k} - 2\,\vec{P}_{L,k} \cdot \vec{W}_{L,k} \qquad ((6))$$

Fig. 7: Illustration of the "performance surface".

If we set the gradient to zero to compute the optimum weight vector $\vec{W}^*_{L,k}$ we obtain

$$0 = 2\underline{R}_{L,k} \cdot \vec{W}^*_{L,k} - 2\vec{P}_{L,k}$$

resulting in $\boxed{\vec{W}^*_{L,k} = \underline{R}^{-1}_{L,k} \cdot \vec{P}_{L,k}}.$ (10)

From (10) we can see that we could cancel the echo immediately, i.e. choose the components of the weight vector in correspondence to the minimum of the performance surface, if we knew the matrix $\underline{R}^{-1}_{L,k}$ as well as the crosscorrelation vector $\vec{P}_{L,k}$. Unfortunately those two components are not easily determined.

Let's first have a look at how to change an existing weight vector $\vec{W}_{L,k}$ into the optimum weight vector $\vec{W}^*_{L,k}$. To this end we multiply (9) from the left side with $\frac{1}{2}\underline{R}^{-1}_{L,k}$ to get

$$\frac{1}{2}\underline{R}^{-1}_{L,k}\nabla_{L,k} = \vec{W}_{L,k} - \underline{R}^{-1}_{L,k}\vec{P}_{L,k},$$

and with the help of (10)

$$\vec{W}^*_{L,k} = \vec{W}_{L,k} - \frac{1}{2}\underline{R}^{-1}_{L,k}\nabla_{L,k}. \qquad (11)$$

Equation (11) trades the knowledge of $\vec{P}_{L,k}$ against the knowledge of $\nabla_{L,k}$ and tells how to find the optimum weight vector in one step if one knows the gradient vector and the inverse of the input correlation matrix. Equation (11) is also known as Newton's method [1]. If the signals involved in the echo cancelling process were stationary, one would need to estimate $\nabla_{L,k}$ and $\underline{R}_{L,k}^{-1}$ only once to adapt the filter to the room acoustics. However, especially for speech signals, neither $\underline{R}_{L,k}^{-1}$ nor $\nabla_{L,k}$ are long term stationary which renders an ever changing performance surface and hence requires tracking of the minimum. In addition to the nonstationarity of the signals involved, the estimations of $\underline{R}_{L,k}^{-1}$ and $\nabla_{L,k}$ will be noisy. Hence a slower adaptation process will be advantageous to smoothen the effects of measurement noise. In order to do this we introduce an artificial factor μ in (11) and exchange $\vec{W}_{L,k}^{*}$ with $\vec{W}_{L,k+1}$ to get

$$\boxed{\vec{W}_{L,k+1} = \vec{W}_{L,k} - \mu \underline{R}_{L,k}^{-1} \nabla_{L,k}}. \tag{12}$$

(12) is a recursive equation the solution of which is expected to converge at $\vec{W}_{L,k}^{*}$. We will investigate the convergence properties by combining (9) and (12) into

$$\begin{aligned} \vec{W}_{L,k+1} &= \vec{W}_{L,k} - \mu \underline{R}_{L,k}^{-1} \cdot \left(2\underline{R}_{L,k} \cdot \vec{W}_{L,k} - 2\vec{P}_{L,k} \right) \\ &= \vec{W}_{L,k} - 2\mu \vec{W}_{L,k} + 2\mu \underline{R}_{L.k}^{-1} \cdot \vec{P}_{L,k} \end{aligned}$$

to obtain with the help of (10)

$$\vec{W}_{L,k+1} = \vec{W}_{L,k} \cdot (1 - 2\mu) + 2\mu \vec{W}_{L,k}^{*} . \tag{13}$$

Equ. (13) is a linear difference equation which can be solved e.g. by using the Z-Transform to yield

$$\vec{W}_{L,k} = \vec{W}_{L,k}^{*} + (1 - 2\mu)^{k} \cdot \left(\vec{W}_{L,0} - \vec{W}_{L,k}^{*} \right). \tag{14}$$

In fact we see that (14) converges to the optimum weight vector $\vec{W}_{L,k}^{*}$ if $\mu \in [0.5, 1)$. The value $\mu=0.5$ corresponds exactly to Newton's method defined by (11) as expected.

Although (12) provides a way of finding the optimal weight vector, the estimation of $\underline{R}_{L,k}^{-1}$ still introduces a considerable computational burden. The simplest way to estimate $\underline{R}_{L,k}^{-1}$ is to replace it by the identity matrix $\underline{I}$ rendering

$$\vec{W}_{L,k+1} = \vec{W}_{L,k} - \mu \nabla_{L,k} \tag{15}$$

which will finally lead us to the infamous LMS algorithm. If we treat (15) in the same fashion as we did (12) we obtain the linear difference equation

$$\vec{W}_{L,k+1} = \vec{W}_{L,k} \cdot \left(1 - 2\mu \underline{R}_{L,k}\right) + 2\mu \underline{R}_{L,k} \vec{W}_{L,k}^{*} \tag{16}$$

which, after its solution renders

$$\vec{W}_{L,k} = \vec{W}_{L,k}^{*} + \left(\underline{I} - 2\mu \underline{R}_{L,k}\right)^{k} \cdot \left(\vec{W}_{L,0} - \vec{W}_{L,k}^{*} \right) \tag{17}$$

where $\underline{I}$ is the identity matrix. If we express

$$\underline{R}_{L,k} = \underline{U}_{L,k} \cdot \underline{\Lambda}_{L,k} \cdot \underline{U}_{L,k}^{T} \tag{18}$$

where $\underline{\Lambda}_{L,k}$ is diagonal with the diagonal elements consisting of the eigenvalues λ_{ik} of $\underline{R}_{L,k}$ and the columns of $\underline{U}_{L,k}$ are their corresponding eigenvectors, then defining

$$\vec{G}_{L,k} = \underline{U}^{T}_{L,k} \cdot \vec{W}_{L,k} \tag{19}$$

and

$$\vec{G}^{*}_{L,k} = \underline{U}^{T}_{L,k} \cdot \vec{W}^{*}_{L,k} \tag{20}$$

finally yields

$$\vec{G}_{L,k} = \vec{G}^{*}_{L,k} + \left[\underline{I} - 2\mu\underline{\Lambda}_{L,k}\right]^{k} \cdot \left[\vec{G}_{L,0} - \vec{G}^{*}_{L,k}\right] \tag{21}$$

if all relevant quantities are transformed into the eigenvector space of $\underline{R}_{L,k}$. Each eigen-mode of (21) converges independently according to $\left[\underline{I} - 2\mu\lambda_{ik}\right]^{k}$. In order to achieve overall convergence, we must have

$$-1 < (1 - 2\mu\lambda_{ik}) < 1. \tag{22}$$

The maximum eigenvalue λ_{max} of $\underline{R}_{L,k}$ is less than the sum of all eigenvalues because all eignevalues are positive (since $\underline{R}_{L,k}$ is positive definite). Also the sum of the eigenvalues of $\underline{R}_{L,k}$ is equal to the trace of $\underline{R}_{L,k}$ which is just L times the signal power of x_k, $\sigma^2_x = E\left[x^2_k\right]$. Therefore if we choose μ such that

$$0 < \mu < \frac{1}{L\sigma^2} \tag{23}$$

then the convergence of the algorithm is guaranteed. From (22) we note that once the value of μ is determined, modes with small eigenvalues will converge slower than those with large ones.

Assuming $\underline{R}_{L,k}$ to be the identity matrix corresponds to assuming that the contour lines of the objective function surface be circular. Under this assumption using a single adaptation constant μ for all dimensions of the weight vector makes perfect sense. Another interpretation of $\underline{R}_{L,k}$ being the identity matrix is that the autocorrelation of the input signal must be a dirac delta distribution which corresponds to a totally uncorrelated signal. The power spectrum of such an autocorrelation is a constant for all frequencies and hence dubbed as "white" in analogy to white light which contains all frequencies (i.e. colors) in equal amounts. From an illustrative point of view it is intuitive that a white signal is ideal for sampling an unknown object, just as white light is ideal of analyzing a room visually. Colored light, on the other hand, will render certain objects difficult to distinguish from others, e.g. a red cube in front of a black wall illuminated with red light is invisible. Especially for speech signals the frequency spectrum is highly colored. Yet for the derivation of the LMS algorithm we will continue using the simplification $\underline{R}_{L,k} = \underline{I}$.

5.1 The LMS algorithm

The least mean square (LMS) algorithm by Widrow et alii [1] is a robust and simple method to adapt the weights of an ALC. The weight update equation is (15). The crucial idea, however, is the way in which the gradient $\nabla_{L,k}$ is estimated. As has been mentioned above the gradient is defined by

$$\nabla_{L,k} = \frac{\partial \xi}{\partial \vec{W}_{L,k}} = \frac{\partial E\left[\varepsilon^2_k\right]}{\partial \vec{W}_{L,k}}. \tag{24}$$

In the LMS algorithm the expected value $E\left[\varepsilon^2_k\right]$ is approximated by ε^2_k itself, yielding

$$\nabla_{L,k} \approx \frac{\partial \varepsilon_k^2}{\partial \vec{W}_{L,k}} = 2\varepsilon_k \frac{\partial \varepsilon_k}{\partial \vec{W}_{L,k}} = 2\varepsilon_k \frac{\partial\left(d_k - \vec{X}_{L,k}^T \cdot \vec{W}_{L,k}\right)}{\partial \vec{W}_{L,k}} = -2\varepsilon_k \cdot \vec{X}_{L,k} \tag{25}$$

which finally leads to the simple update equation

$$\boxed{\vec{W}_{L,k+1} = \vec{W}_{L,k} + 2\mu\varepsilon_k \cdot \vec{X}_{L,k}}. \tag{26}$$

In (26) the gradient is just a rough estimate of the true gradient which is why the LMS algorithm belongs to the so called stochastic gradient methods. As the data involved in the adaptation process are noisy anyway the impreciseness of the gradient estimation doesn't affect the quality of the adaptation. As an aside it shall also be mentioned that the gradient estimation (25) can, of course, also be utilized in (12).

Further analysis [9] reveals that, on average, the closer μ is to the midpoint of its region of convergence, the faster the convergence. Conversely, the closer it is to zero from that point, the slower the convergence. This poses a problem for signals with wide dynamic ranges, an important one being speech which has levels that vary over a range of 30dB. So, for a fixed μ the speed of convergence of LMS varies by a factor greater than 100. This makes LMS unacceptable for use in voice excited AEC applications.

The standard fix for this problem is to modify the coefficient update process such that it is normalized with respect to the excitation signal's power. Thus, the normalized LMS (NLMS) coefficient update is

$$\boxed{\vec{W}_{L,k+1} = \vec{W}_{L,k} + \frac{2\mu\varepsilon_k}{L\sigma_x^2} \cdot \vec{X}_{L,k}} \quad . \tag{27}$$

If the above stability analysis is carried out for (27) rather than for (12) we find that

$$\vec{W}_{L,k} = \vec{W}^*_{L,k} + \left(\underline{I} - \frac{2\mu \underline{R}_{L,k}}{L\sigma_x^2}\right)^k \cdot \left(\vec{W}_{L,0} - \vec{W}^*_{L,k}\right) \tag{28}$$

converges for $0<\mu<1$. (29)

For non-stationary signals such as speech the estimate σ_x^2 must be made time varying. A good way of doing this is to use

$$\boxed{L\sigma_x^2 = L \cdot E\left[x_k^2\right] \approx \vec{X}_{L,k}^T \cdot \vec{X}_{L,k}} \tag{30}$$

which is easily updated using the recursion

$$\vec{X}_{L,k}^T \cdot \vec{X}_{L,k} = \vec{X}_{L,k-1}^T \cdot \vec{X}_{L,k-1} + x_k^2 - x_{k-L}^2 . \tag{31}$$

There are methods of normalization, however, other than the euclidean norm $\vec{X}_{L,k}^T \cdot \vec{X}_{L,k} = \left\|\vec{X}_{L,k}\right\|^2$ like in

$$\vec{W}_{L,k+1} = \vec{W}_{L,k} + \frac{2\mu\varepsilon_k}{\sum_{i=k}^{k-L} |x_i|} \cdot \vec{X}_{L,k} \tag{32}$$

which have the advantage of being more efficient in terms of implementation. The normalized LMS algorithm (NLMS) is more robust than its unnormalized counterpart and shows an improved convergence behaviour [8].

5.2 Improved versions of NLMS

As has been elaborated above, the NLMS algorithm is based on the assumption $\underline{R}_{L,k}=\underline{I}$, which is not justified for AEC applications. Yet the NLMS algorithm enjoys wide popularity due to its simplicity and robustness, although its convergence speed is unsatisfactory for large values of L. There are some workarounds, however, which attempt to relieve the problems accompanied with NLMS.

5.2.1 Row array projections

Row array projections (RAP) [10] are a generalization of the NLMS algorithm in that the weight updates are made more often than just once per new sample. While the NLMS algorithm can be written as

$$\vec{W}_{L,k+1} = \vec{W}_{L,k} + \frac{2\mu \cdot \left(d_k - \vec{X}_{L,k}^T \cdot \vec{W}_{L,k} \right)}{\vec{X}_{L,k}^T \cdot \vec{X}_{L,k}} \cdot \vec{X}_{L,k} \tag{33}$$

The RAP method use the update equation

$$\vec{W}_{L,k+1} = \vec{W}_{L,k} + \frac{2\mu \cdot \left(d_{n_k} - \vec{X}_{L,n_k}^T \cdot \vec{W}_{L,k} \right)}{\vec{X}_{L,n_k}^T \cdot \vec{X}_{L,n_k}} \cdot \vec{X}_{L,n_k} \tag{34}$$

where k denotes the iteration and $n_k \leq k$ may refer to older vectors $\vec{X}_{L,n_k}$ and older desired values d_{n_k}. It can be shown that doing more updates during one sample period improves the convergence rate [10]. A prerequisite, of course, is that there is idle time on the DSP which can be utilized for the purpose of doing several weight updates per sample period.

5.2.2 Variable Step Size

A further method to improve the NLMS algorithm is to adjust the stepsize μ depending on either the adaptation progress or the index of the weight vector element to be changed [11]. The motivation behind the first idea is to use small stepsizes for small weights and larger stepsizes for larger weights [12]. The second idea attempts to use large stepsizes at the beginning of the adaptation, when the error is large, and to switch to smaller stepsizes later on when the task is to precisely locate the minimum of ξ.

In order to understand the derivation of these approaches we introduce the distance vector

$$\vec{D}_{L,k} = \vec{W}^*_{L,k} - \vec{W}_{L,k} = \left[\delta_{0k}, \delta_{1k}, \ldots, \delta_{(L-1)k} \right]^T . \tag{35}$$

Further it is assumed that the microphone signal

$$d_k = d_{k,sp} + d_{k,nt} \tag{36}$$

consists of the reverberated loudspeaker signal $d_{k,sp}$ and the near end talker signal $d_{k,nt}$. The first approach, the so-called exponentially weighted step-size algorithm [12] assumes that a room has approximately an exponentially decaying impulse response. The derivation starts off with the equation

$$\mu_{n,k} = \frac{L}{2} \cdot \frac{E\left[\delta_{nk}^2\right]}{E\left[\vec{D}_{L,k}^T \cdot \vec{D}_{L,k}\right]}. \tag{37}$$

To simplify matters $\mu_{n,k}$ is assumed to be time invariant so that

$$\mu_n = \mu_{n,0} \approx \frac{L}{2} \cdot \frac{\left(w_n^*\right)^2}{\sum_{m=0}^{L-1}\left(w_m^*\right)^2} \approx const \cdot \exp\left(-6.9\frac{T_s}{T_R} \cdot n\right) \tag{38}$$

where T_S denotes the sampling interval and T_R the reverberation time of the room. Hence (38) is not room independent. Additionally the derivation assumes $d_{k,nt} \equiv 0$ which is why this algorithm is sensitive to noise.

Another algorithm [13] follows the equation

$$\mu_k = \frac{1}{1 + \frac{E\left[d_{k,nt}^2\right]}{E\left[x_k^2\right]} \cdot \frac{1}{E\left[\vec{D}_{L,k}^T \cdot \vec{D}_{L,k}\right]}} \tag{39}$$

which yields a constant stepsize for all weight vector elements but is time variant instead. It has to be mentioned, however, that the estimation of $E\left[d_{k,nt}^2\right]$ as well $E\left[\vec{D}_{L,k}^T \cdot \vec{D}_{L,k}\right]$ is difficult.

A more practical algorithm is described in [14] which uses

$$\mu_k = \begin{cases} \mu_0 & if \quad \rho_k < r_1 \\ \mu_0 \cdot 2^{-i} & if \quad r_i \le \rho_k < r_{i+1} \quad \text{for i=1,2,...,m} \\ \mu_0 \cdot 2^{-m} & if \quad r_m \le \rho_k \end{cases} \tag{40}$$

where

$$\rho_k = \frac{\sum_{i=0}^{M-1} x_{k-i}^2}{\sum_{i=0}^{M-1} \varepsilon_{k-i}^2} \tag{41}$$

i.e. the quotient of the receive-input signal power and the send-output signal power. Small values of ρ indicate strong misadjustment which justifies large stepsizes μ_k. Large values of ρ indicate advanced adaptation and call for smaller stepsizes to allow fine-adjustment. The constants r_i and M have to be determined experimentally.

5.2.3 Decorrelation approaches

The basic idea in decorrelation approaches is to "whiten" the input signal which results in a reduced spread of the eigenvalues of the input correlation matrix. This in turn allows for faster convergence of the AEC. Fig. 8 illustrates this point where the speech signal is filtered by a highpass in order to emphasize the higher frequencies and attenuate the lower ones.

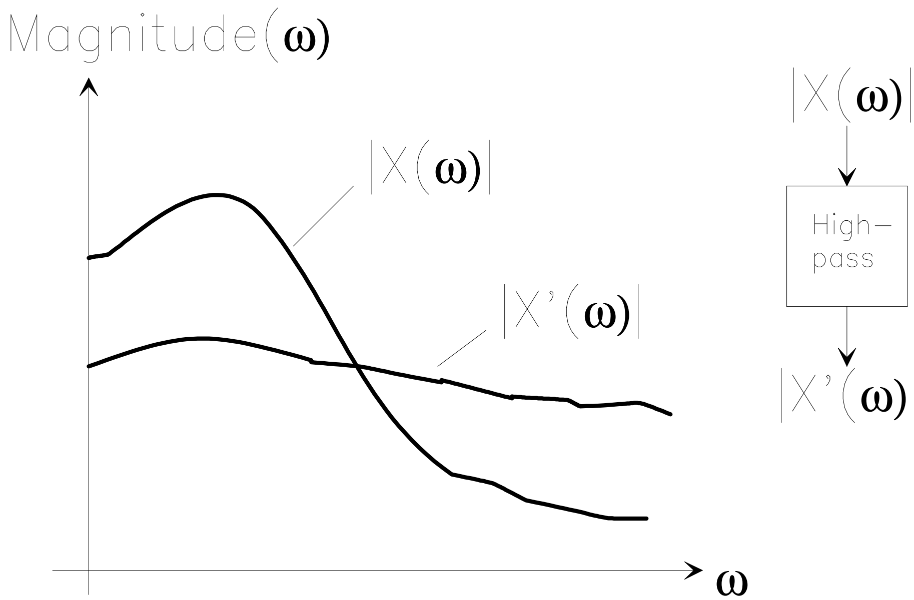


Fig. 8: Whitening of the input signal by highpass filtering.

5.2.3.1 A simple decorrelation approach

A very simple approach to do the highpass filtering is borrowed from delta modulation which is widely used in speech compression technology [15]. Instead of working on the original time sequences, the adaptive filter uses the differential information only which is obtained via the highpass filters hp(z)=(1-z^{-1}). As the highpass filtering operation is a linear one it can be remmoved again by the corresponding lowpass filter lp(z)=1/(1-z^{-1}) at the output. This approach is depicted in fig. 9.

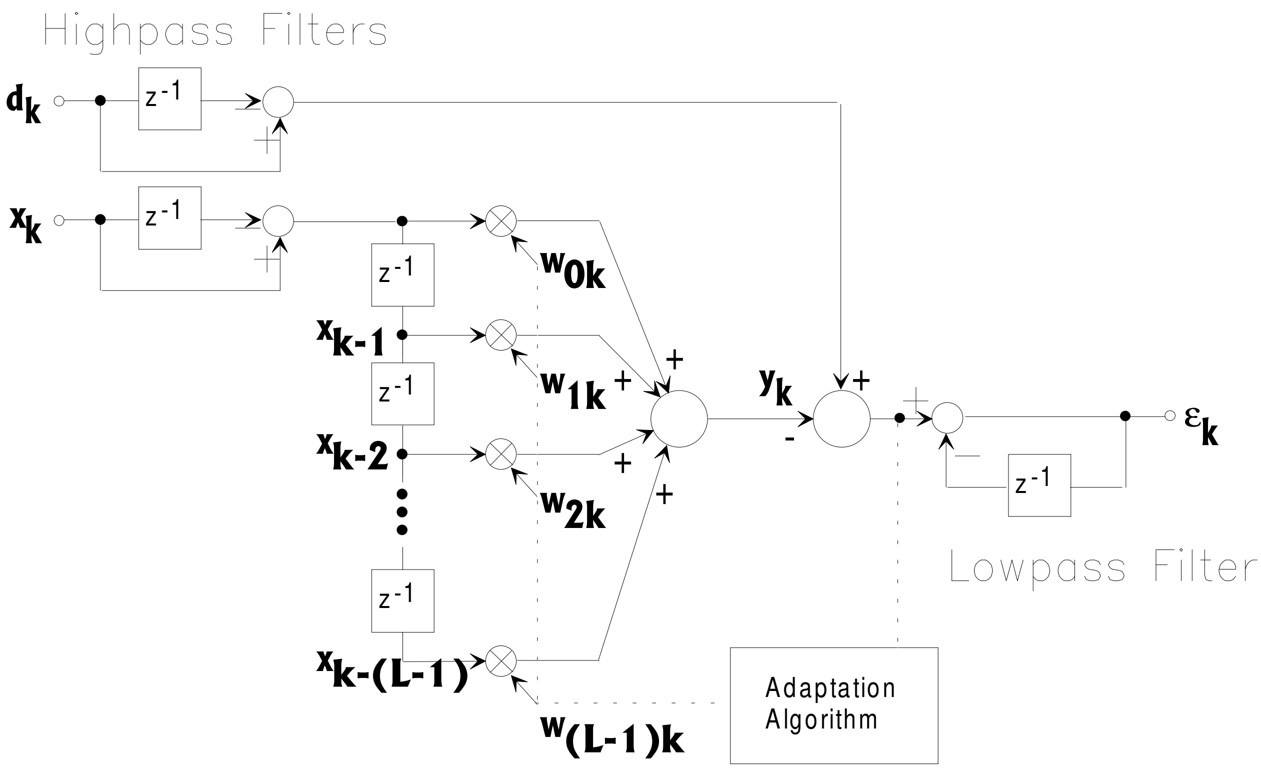


Fig. 9: Simple decorrelation approach with highpass and corresponding lowpass filters.

The AEC of fig. 9 has several drawbacks: The highpass filter hp(z) has only a slight whitening effect as the filter order is too low and hence is unable to compensate for the lowpass characteristic of the speech signal. In addition, the spectrum of the speech signal changes over time. Another disadvantage of decorrelation approaches in general is that the decorrelated signals are usually low in magnitude which causes accuracy problems in fixed point implementations.

5.2.3.2 Time Variant Decorrelation Filters

This approach uses "prewhitened" signals x'_k and d'_k which are obtained by higher order time variant decorrelation filters [16], [17]. As these signals are distorted versions of the loudspeaker signal x_k and the microphone signal d_k, the basic version of this approach requires two identical adaptive filters as can be seen in fig. 10. One filter is used for the NLMS adaptation process and the other one performs the actual echo cancelation.

The basic idea when working with decorrelation filters F(z) is to predict the future signal vector $\vec{X}^p_{L,k}$ from the most recent signal vector $\vec{X}_{L,k-1}$ by the Levinson-Durbin [18] algorithm and to subtract the predicted vector $\vec{X}^p_{L,k}$ from the actual vector $\vec{X}_{L,k}$. As only correlated components can be predicted, ideally only the uncorrelated components remain after subtraction and yield a much better input signal for the NLMS algorithm. The Levinson-Durbin algorithm computes the coefficients of the decorrelation filters A(z) which actually perform the linear prediction and subtract $\vec{X}^p_{L,k}$ from $\vec{X}_{L,k}$. The filters A(z) usually have a low degree M, e.g. M=6 for a canceling filter of order L=1500. The computation performed by A(z) having the transfer function

$$A(z) = \sum_{n=0}^{M-1} a_n \cdot z^{-n} \tag{42}$$

is

$$x_k^{'} = x_k - \sum_{n=0}^{M-1} a_n \cdot x_{k-n-1} \tag{43}$$

and

$$d_k^{'} = d_k - \sum_{n=0}^{M-1} a_n \cdot d_{k-n-1}\,. \tag{44}$$

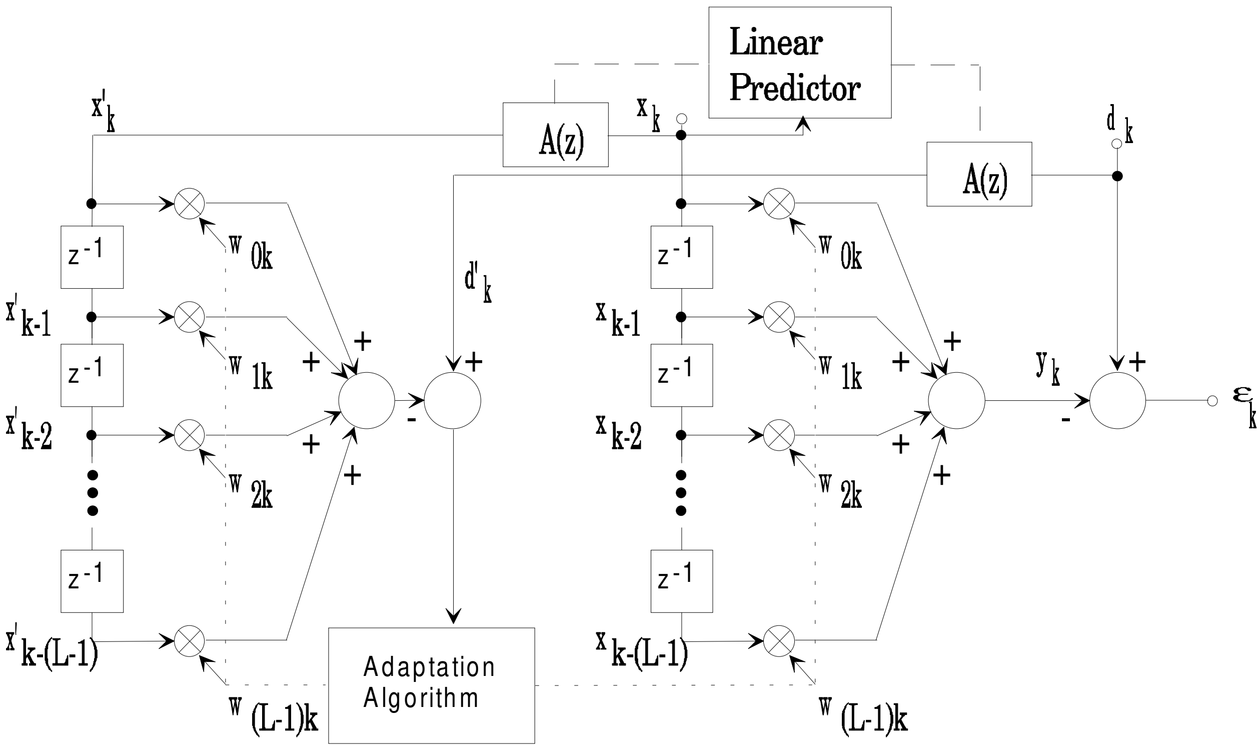


Fig. 10: Echo Canceler with NLMS algorithm and decorrelation filters A(z).

The coefficients of A(z) are usually refreshed after about 25ms in case of speech signals. During the time where the coefficients are constant, A(z) has to compute just one output value x'_k and d'_k respectively. After the refresh-operation, however, the entire signal vector $\vec{X}'_{L,k}$ must be recomputed to initialize the buffers of the NLMS filter correctly.

For completion the Levinson-Durbin algorithm is stated in the following.

1) Initialize r=0 (counter variable) and $a_0 = \dfrac{-x_1}{x_0}$

2) r=r+1

3) $\gamma_r = -\sum_{i=1}^{r} x_i \cdot a_{r-i}$

$\gamma'_r = -\sum_{i=1}^{r} x_i \cdot a_{i-1}$

4) $\beta_r = -\dfrac{x_{r+1} - \gamma_r}{x_0 - \gamma'_r}$

5) $$\begin{bmatrix} a_0^{(r+1)} \\ a_1^{(r+1)} \\ \ldots \\ a_{r-1}^{(r+1)} \\ a_r^{(r+1)} \end{bmatrix} = \begin{bmatrix} a_0^{(r)} \\ a_1^{(r)} \\ \ldots \\ a_{r-1}^{(r)} \\ 0 \end{bmatrix} + \beta_r \cdot \begin{bmatrix} a_{r-1}^{(r)} \\ a_{r-2}^{(r)} \\ \ldots \\ a_0^{(r)} \\ 1 \end{bmatrix}$$

6) If r=M-1 finish, else goto 2).

Again, this approach requires high precision computation which makes it less suitable for fixed point implementations. It is also much more complex than the previous approach as it requires two adaptive filters.

5.2.3.3 The Signal Decorrelation Algorithm

A compromise between good decorrelation and low implementation complexity is the Signal Decorrelation Algorithm (DCR) by Yasukawa et alii [14], [42], [43].

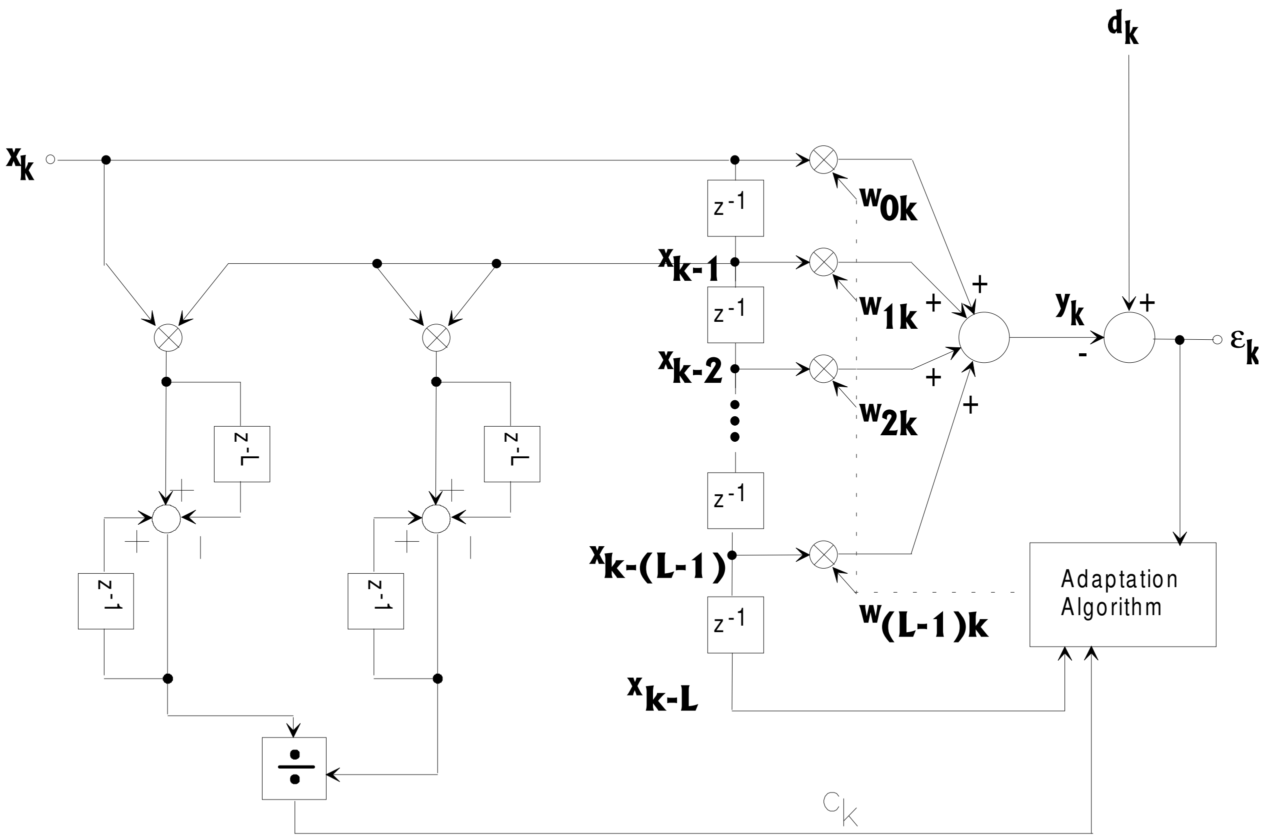


Fig. 14: Block diagram of the DCR-AEC.

Here the weight update equation is given by

$$\vec{W}_{L,k+1} = \vec{W}_{L,k} + \frac{2\mu_k \cdot \varepsilon_k \cdot \vec{Z}_{L,k}}{\vec{X}_{L,k}^T \cdot \vec{Z}_{L,k}} \qquad (45)$$

with $\vec{Z}_{L,k} = \vec{X}_{L,k} - c_k \cdot \vec{X}_{L,k-1}$ (46)

and $c_k = \dfrac{\vec{X}^T_{L,k} \cdot \vec{X}_{L,k-1}}{\vec{X}_{L,k-1} \cdot \vec{X}^T_{L,k-1}}$ (47)

The DCR algorithm is attractive due to its simplicity which stems from the fact that just the first order correlation coefficient c_k is exploited. Again, small numbers are prevalent in the DCR-calculations which call for high precision computation, especially when large values of L are used.

5.2.4 Perfect Sequences

Perfect Sequences [16], [19] constitute an optimal excitation signal for the NLMS algorithm such that an L-tap adaptive filter is completely adapted after 2L input signal samples. The prerequisites that this holds true are:

1) The unknown system to be identified (the reverberating room) is linear and time invariant (LTI)

2) The system is noiseless

3) The step-size $2\mu_0$ is equal to 1

4) The adaptive filter and the unknown system are of the same length L

Perfect sequences s_i are characterized by their periodic autocorrelation function which vanishes for all out of phase values [20]. Hence we get

$$\sum_{k=0}^{L-1} s_{<i>_L} \cdot s_{<k+i>_L} = \begin{cases} \left\| s_{<i>_L} \right\| & for \quad <i>_L = 0 \\ 0 & else \end{cases} \tag{48}$$

where $<>_L$ denotes the modulo function with modulus L. Let's regard the NLMS update equation (27), (30)

$$\begin{aligned} \vec{W}_{L,k+1} &= \vec{W}_{L,k} + \frac{2\mu_0 \cdot \varepsilon_k}{\vec{X}^T_{L,k} \cdot \vec{X}_{L,k}} \cdot \vec{X}_{L,k} \\ &\overset{(4)}{=} \vec{W}_{L,k} + \frac{2\mu_0 \cdot \left(d_k - \vec{X}^T_{L,k} \cdot \vec{W}_{L,k} \right)}{\vec{X}^T_{L,k} \cdot \vec{X}_{L,k}} \cdot \vec{X}_{L,k} \\ &= \vec{W}_{L,k} + \frac{2\mu_0 \cdot \left(\vec{X}^T_{L,k} \cdot \vec{W}^*_{L,k} - \vec{X}^T_{L,k} \cdot \vec{W}_{L,k} \right)}{\vec{X}^T_{L,k} \cdot \vec{X}_{L,k}} \cdot \vec{X}_{L,k} \end{aligned} \tag{49}$$

which, after applying $2\mu_0$=1 and

$$\vec{X}_{L,k} = \vec{S}_{L,k} = \left[s_{<k>_L}, s_{<k+1>_L}, \ldots, s_{<k+L-1>_L} \right]^T \tag{50}$$

turns into

$$\vec{W}_{L,k+1} = \vec{W}_{L,k} + \frac{\vec{S}^T_{L,k} \cdot \left(\vec{W}^*_{L,k} - \vec{W}_{L,k} \right)}{\vec{S}^T_{L,k} \cdot \vec{S}_{L,k}} \cdot \vec{S}_{L,k} \tag{51}$$

If we now multiply (51) by $\vec{S}^T_{L,j}$ we obtain

$$\vec{S}^T_{L,j} \cdot \vec{W}_{L,k+1} = \vec{S}^T_{L,j} \cdot \vec{W}_{L,k} + \frac{\vec{S}^T_{L,k} \cdot \left(\vec{W}^*_{L,k} - \vec{W}_{L,k} \right)}{\vec{S}^T_{L,k} \cdot \vec{S}_{L,k}} \cdot \vec{S}^T_{L,j} \cdot \vec{S}_{L,k} . \tag{52}$$

Because perfect sequences obey

$$\vec{S}^T_{L,j} \cdot \vec{S}_{L,k} = \begin{cases} 0 & for \quad j \neq k \\ \vec{S}^T_{L,k} \cdot \vec{S}_{L,k} & for \quad j = k \end{cases} \tag{53}$$

(52) can be expressed as

$$\vec{S}^T_{L,k} \cdot \vec{W}_{L,k+1} = \begin{cases} \vec{S}^T_{L,k} \cdot \vec{W}_{L,k} & for \quad j \neq k \\ \vec{S}^T_{L,k} \cdot \vec{W}^*_{L,k} & for \quad j = k \end{cases} . \tag{54}$$

(54) illustrates that the jth component of vector $\vec{W}_{L,k+1}$ and $\vec{W}^*_{L,k}$ do match while the remaining L-1 components are still unchanged. During the adaptation process in each iteration one component of vector $\vec{W}^*_{L,k}$ is identified. To be of use for AEC the room impulse response has at least to be short time LTI and the perfect sequence must serve as a training signal. In order to satisfy prerequisite 2) the power of the training signal has to be large enough. Such a training signal is usually well audible and hence not acceptable for convenient audio communication. Therefore this method, despite its mathematical beauty, is not practical for AEC. Simulations where low powered training signals were used to help improve the convergence behaviour of an NLMS-AEC have been conducted and have shown indeed some improvement. Subjective tests via DSP implementations in real audio systems, however, indicated that no perceivable improvement can be obtained [21].

5.2.5 Transform Domain LMS (TDLMS)

All of the above approaches to improve the behaviour of the LMS algorithm are operating in the time domain. An intriguing idea, however, is to use the frequency domain in order to decorrelate the input signal of the LMS algorithm [22], [23]. An intuitive approach is to apply a Fourier series decomposition to the input signal x_k which yields the coefficients for each of the frequency components. As the Fourier basis functions form an orthogonal set the different frequencies should be totally uncorrelated. After weighting each frequency component and normalizing it with the inverse of its signal power, a Fourier synthesis is done again and the synthesized signal is subtracted from the desired signal d_k. The adjustment of the weights can, again, be done by the LMS algorithm. Fourier analysis is, of course, a difficult task to do as indefinite integrals are involved. Hence the Fourier analysis is approximated by the Discrete Fourier Transform (DFT) for which fast algorithms exist. The block diagram for this approach is depicted in fig. 15.

Not only the DFT but any other orthogonal transform can be used to decorrelate the input signal [22], [23]. In real-time (or nonblock) algorithms, the flow of input samples is continuously transformed by a fixed, data-independent transform which, in case of speech signals, is often chosen to be the Discrete Cosine Transform (DCT). This way of continuously transforming is also called "sliding transform". After the preprocessing, which decorrelates the input signal, a power normalization stage follows. The latter causes the eigenvalues of the LMS filter inputs to cluster around one and speeds up the convergence of the adaptive weights. The performance of these algorithms clearly depends on the orthogonalizing capabilities of the data independent transform.

In mathematical terms we can write

$$u_{ik} = \sum_{l=0}^{L-1} T(i,l) \cdot x_{k-l} \tag{55}$$

where T(i,l) denotes the element of the ith row and the lth coloumn of the matrix $\underline{T}_L$ which describes the orthogonal transform. For the DFT we get

$$T(i,l) = W_L^{-il} = e^{-j\frac{2\pi il}{L}} \quad with \quad j = \sqrt{-1} \tag{56}$$

For the DCT we obtain

$$T(i,l) = \sqrt{\frac{2}{L}} \cdot k_i \cdot \cos\left(\frac{i(l+0.5)\cdot\pi}{L}\right), \quad k_i = \begin{cases} \frac{1}{\sqrt{2}} & for \quad i=0 \\ 1 & otherwise \end{cases}. \tag{57}$$

The weight update equation of the TDLMS can be formulated as

$$w_{i(k+1)} = w_{ik} + \frac{2\mu\varepsilon_k \cdot u_{ik}^*}{\sqrt{P_{ik} + \eta}} \tag{58}$$

with the asterisk in (58) denoting complex conjugation and the signal power P_{ik} being approximated by

$$P_{ik} = \beta \cdot P_{i(k-1)} + (1-\beta) \cdot u_{ik}^2 . \tag{59}$$

The constant η in (58) should be positive and small compared to P_{ik} as it is just a safeguard to prevent the denominator of (58) from becoming zero. TDLMS offers fairly fast convergence but requires a sliding transform to be executed. For the FFT the L-point sliding transform can be computed in O(L) computations [24], the memory requirement, however, increases with $O\left(L \cdot \log_2(L)\right)$. Additionally L is constrained to be a power of two. A computationally more efficient way to evaluate the sliding DFT as well as DCT can be obtained by using the Goertzel algorithm [8]. Several authors have investigated this possibility [25], [26], [27], and the basic block diagram is sketched in fig. 16 for the DCT. A crucial ingredient in the recursive computation of the sliding transform is the damping factor α which prevents the poles of the recursive filters to lie on the unit circle which wo5uld make the filter unstable. Therefore α must be positive and smaller than 1, but for for satisfactory AEC performance α has to be very close to 1. The ensuing precision requirements are difficult to meet when it comes to fixed point implementations. Another disadvantage of

TDLMS in general is the large number of power estimations in (58) which requires L divisions for one coefficient update. As divisions are usually very costly on DSPs, TDLMS is rarely implemented for AEC-applications despite its simplicity and favourable convergence behaviour.

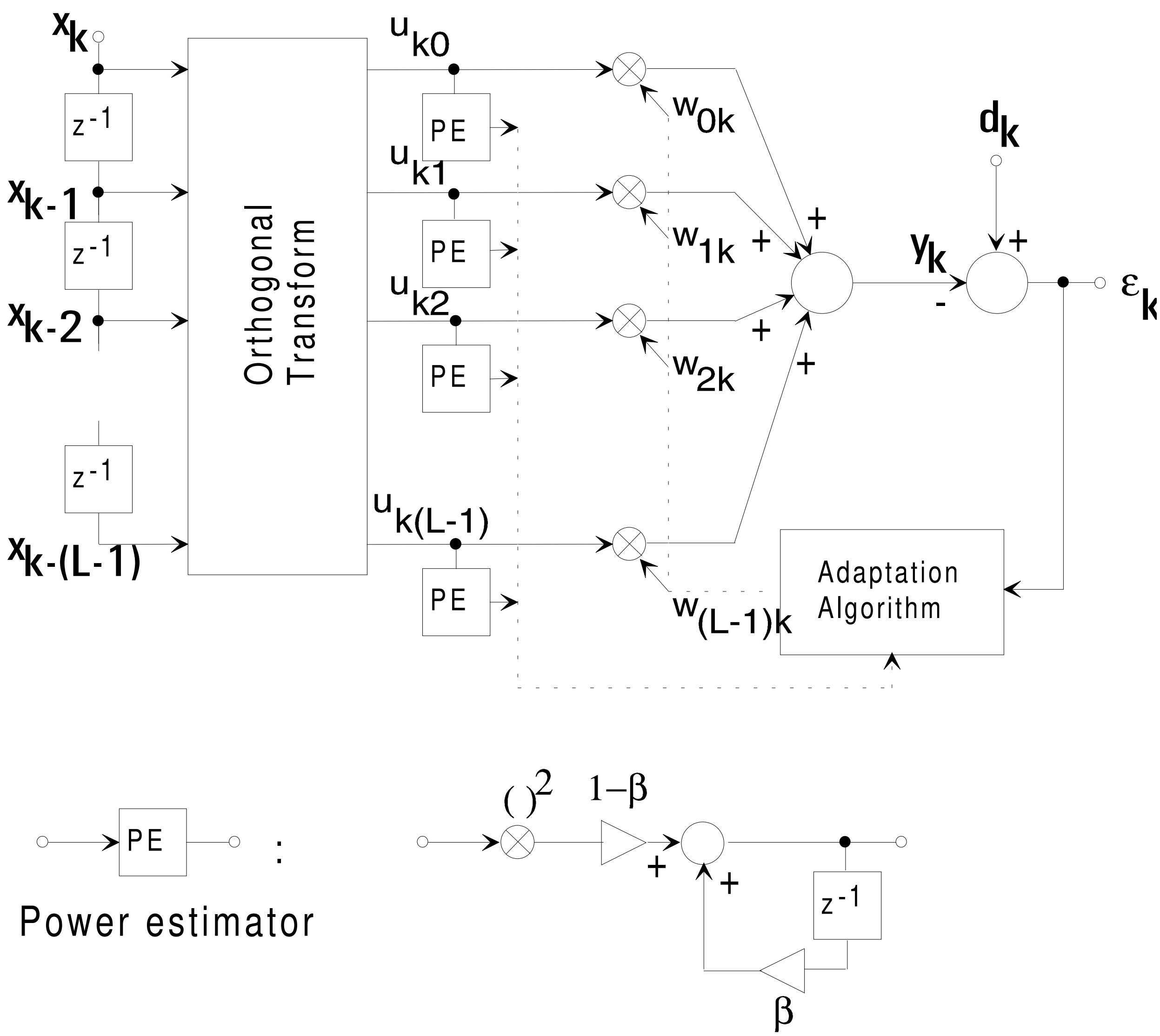


Fig. 15: Block diagram of the transform domain LMS algorithm.

Another variant of TDLMS which is better suited for AEC applications has been described in [28] and is called the generalized subband decomposition (GSD). Fig. 17 shows the block diagram of GSD which exhibits adaptive filters $G_i(z^L)$ instead of just single weights w_{ik} as used in the regular TDLMS algorithm. The main idea is to use a relatively small-sized orthogonal M-point transform, e.g. M=8 or M=16 as opposed to M=512 or more in the traditional TDLMS approach, and obtain the required length of the impulse response through the adaptive filters $G_i(z^L)$. The crucial finding in GSD is that the $G_i(z^L)$ may be sparse so that even though they contain a delay chain of L elements the number of nonzero coefficients is just L/K, where K is the so-called interpolation factor. The nonzero coefficients are adaptive and occur every K delay elements. For K=M there is already some convergence improvement compared to the NLMS baseband AEC, yet much greater benefits emerge if K<M [28]. GSD uses much less divisions than the traditional TDLMS and hence is computationally more efficient.

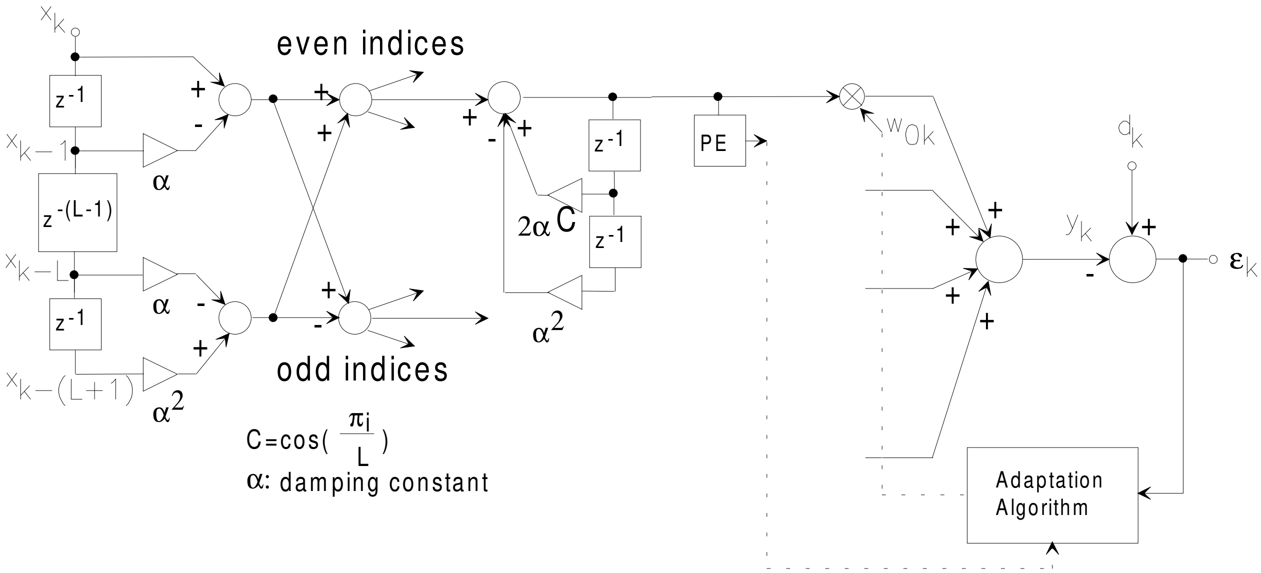


Fig. 16: DCT-TDLMS algorithm with recursive filters.

For AEC applications which require a large L, however, the computational load is still substantial and also the precision requirements are high, because the increments by which the adaptive coefficients are altered are fairly small.

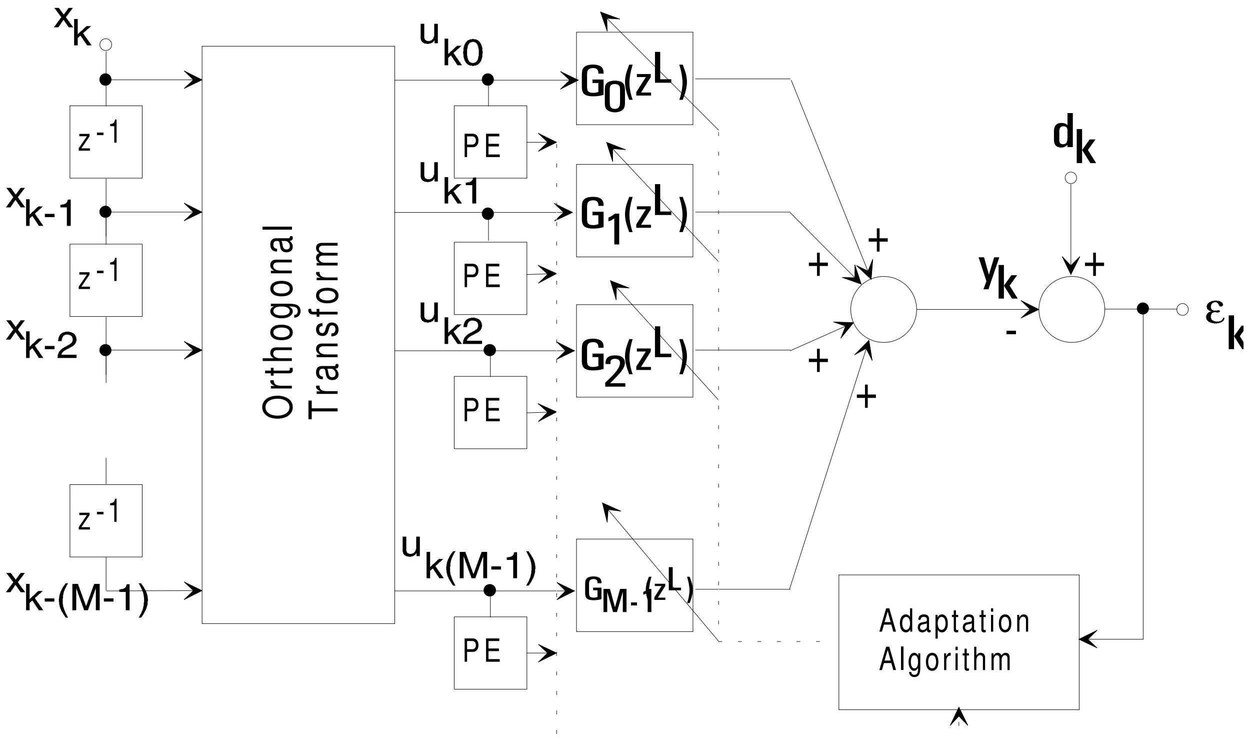


Fig. 17: Generalized Subband Decomposition (GSD).

6. Other than LMS approaches

6.1 Recursive Least Squares Approach

As the input correlation matrix $\underline{R}_{L,k}$ provides valuable information about the performance surface one of the most obvious approaches to improve the convergence behaviour of the weight adaptation in the ALC is to look for fast algorithms in order to compute $\underline{R}_{L,k}^{-1}$.

As opposed to the LMS algorithm in the recursive least squares (RLS) algorithm the expected value $E\left[\varepsilon_k^2\right]$ is not approximated by ε_k^2 itself but is computed by averaging several samples of ε_k^2 according to

$$E\left[\varepsilon_k^2\right] \approx \frac{1}{k+1}\sum_{i=0}^{k}\left(d_i - \vec{X}_{L,i}^T \cdot \vec{W}_{L,k}\right)^2 \cdot \lambda^{k-i} \tag{60}$$

with $\lambda \leq 1$ which is an approximation for

$$E\left[\varepsilon_k^2\right] = \lim_{k\to\infty}\frac{1}{k+1}\sum_{i=0}^{k}\left(d_i - \vec{X}_{L,i}^T \cdot \vec{W}_{L,i}\right)^2. \tag{61}$$

Note that the weight vector $\vec{W}_{L,k}$ has a constant time index, i.e. it is assumed that the weight vector changes at a much slower rate than the input signal vector $\vec{X}_{L,k}^T$. This assumption seems to be reasonable for AEC applications. The method to approximate $E\left[\varepsilon_k^2\right]$ by (60) is called "exponential windowing". The factor λ is a so called "forgetting factor" and should not be confused with the eigenvalues of the correlation matrix $\underline{R}_{L,k}$. Another method to estimate $E\left[\varepsilon_k^2\right]$ is to compute

$$E\left[\varepsilon_k^2\right] \approx \frac{1}{l+1}\sum_{i=k-l}^{k}\left(d_i - \vec{X}_{L,i}^T \cdot \vec{W}_{L,k}\right)^2 \tag{62}$$

which is the "sliding window" method. However, we will only consider exponential windowing in the following.

As in the previous chapter the goal is to minimize $E\left[\varepsilon_k^2\right]$. We obtain the same solution if we drop the factor $(k+1)^{-1}$ and try to minimize

$$
\begin{aligned}
E'\left[\varepsilon_k^2\right] &\approx \sum_{i=0}^{k}\left(d_i - \vec{X}_{L,i}^T \cdot \vec{W}_{L,k}\right)^2 \cdot \lambda^{k-i} \\
&= \sum_{i=0}^{k}\left(d_i^2 + \vec{W}_{L,k}^T \cdot \vec{X}_{L,i} \cdot \vec{X}_{L,i}^T \cdot \vec{W}_{L,k} - 2 d_i \cdot \vec{X}_{L,i}^T \cdot \vec{W}_{L,k}\right) \cdot \lambda^{k-i} \\
&= \sum_{i=0}^{k} d_i^2 \cdot \lambda^{k-i} + \vec{W}_{L,k}^T \cdot \left(\sum_{i=0}^{k} \vec{X}_{L,i} \cdot \vec{X}_{L,i}^T \cdot \lambda^{k-i}\right) \cdot \vec{W}_{L,k} - 2\left(\sum_{i=0}^{k} d_i \cdot \vec{X}_{L,i}^T \cdot \lambda^{k-i}\right) \cdot \vec{W}_{L,k} \\
&= \sum_{i=0}^{k} d_i^2 \cdot \lambda^{k-i} + \vec{W}_{L,k}^T \cdot \underline{\Phi}_{L,k} \cdot \vec{W}_{L,k} - 2\,\vec{\Theta}_{L,k} \cdot \vec{W}_{L,k}
\end{aligned}
$$

We notice the resemblance with (6) and can easily verify that

$$
\vec{W}_{L,k}^{*} = \underline{\Phi}_{L,k}^{-1} \cdot \vec{\Theta}_{L,k} \tag{63}
$$

defines the optimum weight vector $\vec{W}_{L,k}^{*}$ in analogy to (10). Let's have a closer look at the correlation matrix $\underline{\Phi}_{L,k}$ and the correlation vector $\vec{\Theta}_{L,k}$. We can write

$$
\begin{aligned}
\underline{\Phi}_{L,k} &= \sum_{i=0}^{k} \vec{X}_{L,i} \cdot \vec{X}_{L,i}^T \cdot \lambda^{k-i} \\
&= \lambda \cdot \sum_{i=0}^{k-1} \vec{X}_{L,i} \cdot \vec{X}_{L,i}^T \cdot \lambda^{k-1-i} + \vec{X}_{L,k} \cdot \vec{X}_{L,k}^T \\
&= \lambda \cdot \underline{\Phi}_{L,k-1} + \vec{X}_{L,k} \cdot \vec{X}_{L,k}^T
\end{aligned} \tag{64}
$$

to get a recursive formula for the correlation matrix $\underline{\Phi}_{L,k}$. Similiarly the correlation vector $\vec{\Theta}_{L,k}$ can be expressed as

$$
\begin{aligned}
\vec{\Theta}_{L,k} &= \sum_{i=0}^{k} \vec{X}_{L,i} \cdot d_i \cdot \lambda^{k-i} \\
&= \lambda \cdot \sum_{i=0}^{k-1} \vec{X}_{L,i} \cdot d_i \cdot \lambda^{k-1-i} + \vec{X}_{L,k} \cdot d_k \\
&= \lambda \cdot \vec{\Theta}_{L,k-1} + \vec{X}_{L,k} \cdot d_k
\end{aligned} \tag{65}
$$

providing another recursive formula. In order to proceed with the derivation of the recursive least squares (RLS) algorithm we have to introduce **Woodbury's identity** [29], a useful mathematical lemma which states the following:

Let $\underline{A}$ and $\underline{B}$ be two positive-definite MxM matrices related by

$$
\underline{A} = \underline{B}^{-1} + \underline{C} \cdot \underline{D}^{-1} \cdot \underline{C}^H \tag{66}
$$

where $\underline{D}$ is another positive-definite NxN matrix and $\underline{C}$ is an MxN matrix, then the inverse of $\underline{A}$ may be expressed by

$$\underline{A}^{-1} = \underline{B} - \underline{B} \cdot \underline{C} \cdot \left(\underline{D} + \underline{C}^H \cdot \underline{B} \cdot \underline{C}\right)^{-1} \cdot \underline{C}^H \cdot \underline{B}. \tag{67}$$

As a reminder we repeat that a positive-definite matrix has eigenvalues which are all >0 and that the notation $\underline{C}^H$ refers to hermitian transposition of matrix $\underline{C}$, i.e. complex conjugation and transposition combined.

If we choose

$$\begin{aligned} \underline{A} &= \underline{\Phi}_{L,k} \\ \underline{B}^{-1} &= \lambda \cdot \underline{\Phi}_{L,k-1} \\ \underline{C} &= \vec{X}^T_{L,k} \\ \underline{D} &= 1 \end{aligned} \tag{68}$$

and use Woodbury's identity together with (64), we obtain

$$\begin{aligned} \underline{\Phi}^{-1}_{L,k} &= \lambda^{-1} \cdot \underline{\Phi}^{-1}_{L,k-1} - \lambda^{-1} \cdot \underline{\Phi}^{-1}_{L,k-1} \cdot \vec{X}^T_{L,k} \cdot \frac{\vec{X}_{L,k} \cdot \lambda^{-1} \cdot \underline{\Phi}^{-1}_{L,k-1}}{\left(1 + \vec{X}_{L,k} \cdot \lambda^{-1} \cdot \underline{\Phi}^{-1}_{L,k-1} \cdot \vec{X}^T_{L,k}\right)} \\ &= \lambda^{-1} \cdot \underline{\Phi}^{-1}_{L,k-1} - \frac{\lambda^{-2} \cdot \underline{\Phi}^{-1}_{L,k-1} \cdot \vec{X}^T_{L,k} \cdot \vec{X}_{L,k} \cdot \underline{\Phi}^{-1}_{L,k-1}}{\left(1 + \vec{X}_{L,k} \cdot \lambda^{-1} \cdot \underline{\Phi}^{-1}_{L,k-1} \cdot \vec{X}^T_{L,k}\right)} \\ &= \lambda^{-1} \cdot \left(\underline{\Phi}^{-1}_{L,k-1} - \frac{\underline{\Phi}^{-1}_{L,k-1} \cdot \vec{X}^T_{L,k}}{\left(\lambda + \vec{X}_{L,k} \cdot \underline{\Phi}^{-1}_{L,k-1} \cdot \vec{X}^T_{L,k}\right)} \cdot \vec{X}_{L,k} \cdot \underline{\Phi}^{-1}_{L,k-1} \right) \end{aligned}$$

and finally

$$\boxed{\underline{\Phi}^{-1}_{L,k} = \lambda^{-1} \cdot \left(\underline{\Phi}^{-1}_{L,k-1} - \vec{C}_{L,k} \cdot \vec{X}_{L,k} \cdot \underline{\Phi}^{-1}_{L,k-1} \right)} \tag{69}$$

where

$$\boxed{\vec{C}_{L,k} = \frac{\underline{\Phi}^{-1}_{L,k-1} \cdot \vec{X}^T_{L,k}}{\left(\lambda + \vec{X}_{L,k} \cdot \underline{\Phi}^{-1}_{L,k-1} \cdot \vec{X}^T_{L,k}\right)}}. \tag{70}$$

Eq. (69) provides a recursive update formula for the inverse of the correlation matrix and constitutes the central idea in RLS. Let's now recast (70) to obtain

$$\vec{C}_{L,k} \left(1 + \lambda^{-1} \cdot \vec{X}_{L,k} \cdot \underline{\Phi}^{-1}_{L,k-1} \cdot \vec{X}^T_{L,k}\right) = \lambda^{-1} \cdot \underline{\Phi}^{-1}_{L,k-1} \cdot \vec{X}^T_{L,k}$$

i.e. $$\vec{C}_{L,k} = \lambda^{-1} \cdot \left(\underline{\Phi}_{L,k-1}^{-1} - \vec{C}_{L,k} \cdot \vec{\vec{X}}_{L,k} \cdot \underline{\Phi}_{L,k-1}^{-1} \right) \cdot \vec{X}_{L,k}^{T}$$

which shows us by regarding (69) and the fact that $\underline{\Phi}_{L,k}^{-1}$ is symmetric that the relationship

$$\vec{C}_{L,k} = \underline{\Phi}_{L,k}^{-1} \cdot \vec{X}_{L,k}^{T} = \underline{\Phi}_{L,k}^{-1} \cdot \vec{X}_{L,k} \tag{71}$$

holds. We will use now (65) and (69) to express (63) in a different way to get

$$\begin{aligned} \vec{W}_{L,k}^{*} &= \lambda^{-1} \cdot \left(\underline{\Phi}_{L,k-1}^{-1} - \vec{C}_{L,k} \cdot \vec{X}_{L,k} \cdot \underline{\Phi}_{L,k-1}^{-1} \right) \cdot \left(\lambda \cdot \vec{\Theta}_{L,k-1} + \vec{X}_{L,k} \cdot d_k \right) \\ &= \underline{\Phi}_{L,k-1}^{-1} \cdot \vec{\Theta}_{L,k-1} - \vec{C}_{L,k} \cdot \vec{X}_{L,k} \cdot \underline{\Phi}_{L,k-1}^{-1} \cdot \vec{\Theta}_{L,k-1} + \underline{\Phi}_{L,k}^{-1} \cdot \vec{X}_{L,k} \cdot d_k \\ &\overset{(71)}{=} \vec{W}_{L,k-1}^{*} - \vec{C}_{L,k} \cdot \vec{X}_{L,k} \cdot \vec{W}_{L,k-1}^{*} + \vec{C}_{L,k} \cdot d_k \end{aligned}$$

and finally $$\boxed{\vec{W}_{L,k}^{*} = \vec{W}_{L,k-1}^{*} + \vec{C}_{L,k} \cdot \left(d_k - \vec{X}_{L,k} \cdot \vec{W}_{L,k-1}^{*} \right)}$$

or $$\boxed{\begin{aligned} &\vec{W}_{L,k}^{*} = \vec{W}_{L,k-1}^{*} + \vec{C}_{L,k} \cdot \alpha_k \\ &with \quad \alpha_k = \left(d_k - \vec{X}_{L,k} \cdot \vec{W}_{L,k-1}^{*} \right) \end{aligned}} \tag{72}$$

Vector $\vec{C}_{L,k}$ is usually referred to as the KALMAN VECTOR and the expression $\alpha_k = \left(d_k - \vec{X}_{L,k} \cdot \vec{W}_{L,k-1}^{*} \right)$ as the A PRIORI ERROR. For comparison regard equ. (4), $\varepsilon_k = d_k - y_k = d_k - \vec{X}_{L,k}^{T} \cdot \vec{W}_{L,k}$, which describes the so called A POSTERIORI ERROR.

The idea of the RLS algorithm is to compute $\vec{C}_{L,k}$ by using (70), updating $\underline{\Phi}_{L,k}^{-1}$ via (69) and then update $\vec{W}_{L,k}^{*}$ using (72). In order to get the RLS algorithm to work, we have to initialize both $\vec{W}_{L,k-1}^{*}$ and $\underline{\Phi}_{L,k-1}^{-1}$. $\vec{W}_{L,k-1}^{*}$ is usually chosen to be 0 while $\underline{\Phi}_{L,k-1}^{-1}$ should be initially set to some small diagonal matrix, say $0.001 \cdot \sigma_x^2 \cdot \underline{I}$, to avoid division by zero in the first iteration of this algorithm. The variance σ_x^2 can be computed by

$$\sigma_x^2 = Var\left[\vec{X}_{L,k} \right] = \frac{1}{L} \sum_{i=0}^{L-1} x_i^2 - \left(\frac{1}{L} \sum_{i=0}^{L-1} x_i \right)^2 . \tag{73}$$

A closer look at the various computations reveals that the vector by vector multiplications are of order O(L) whereas the vector by matrix multiplications are of order O(L^2) meaning that the entire RLS algorithm is of order O(L^2). In acoustic echo cancelers L can easily turn out to be in the range of 1000-3000 resulting in a significant computational burden. However, it has been shown in [8] that the RLS algorithm is superior to the NLMS algorithm in terms of convergence speed and tracking if the eigenvalue spread of the input correlation matrix is large. This large eigenvalue spread is characteristic in AEC scenarios. Extensive research has been going on to compute the RLS problem by fast algorithms like Fast Transversal Filters (FTFs) as described in [30], [31] and [32].

6.2 Fast Transversal Filters

The basic idea in FTFs is to find a scheme for the computation of $\vec{C}_{L,k}$ which is O(L). In fact this can be achieved, if linear forward and backward prediction filters are used as auxiliary filters. The forward linear predictor computes the estimate

$$x_i \approx \vec{X}_{L,i-1}^T \cdot \vec{F}_{L,k} \tag{74}$$

whereas the backward linear predictor computes the estimate

$$x_{i-L} \approx \vec{X}_{L,i}^T \cdot \vec{B}_{L,k} . \tag{75}$$

Both forward and backward linear predictor are adaptive filters, similiar to the actual echo cancellation filter, the coefficients of which can be updated according to the RLS algorithm yielding

$$\vec{F}_{L,k}^* = \vec{F}_{L,k-1}^* + \vec{C}_{L,k-1} \cdot \eta_k \tag{76}$$

and

$$\vec{B}_{L,k}^* = \vec{B}_{L,k-1}^* + \vec{C}_{L,k} \cdot \psi_k \tag{77}$$

with η_k and ψ_k being the pertinent A PRIORI ERRORs. The important finding in (76), (77) and (72) is that all three equations are referring to the same Kalman vector. It can be shown that the Kalman vector can be updated recursively with complexity O(L) by using the coefficients and estimation errors of the forward and backward prediction filters. This is indicated in fig. 18.

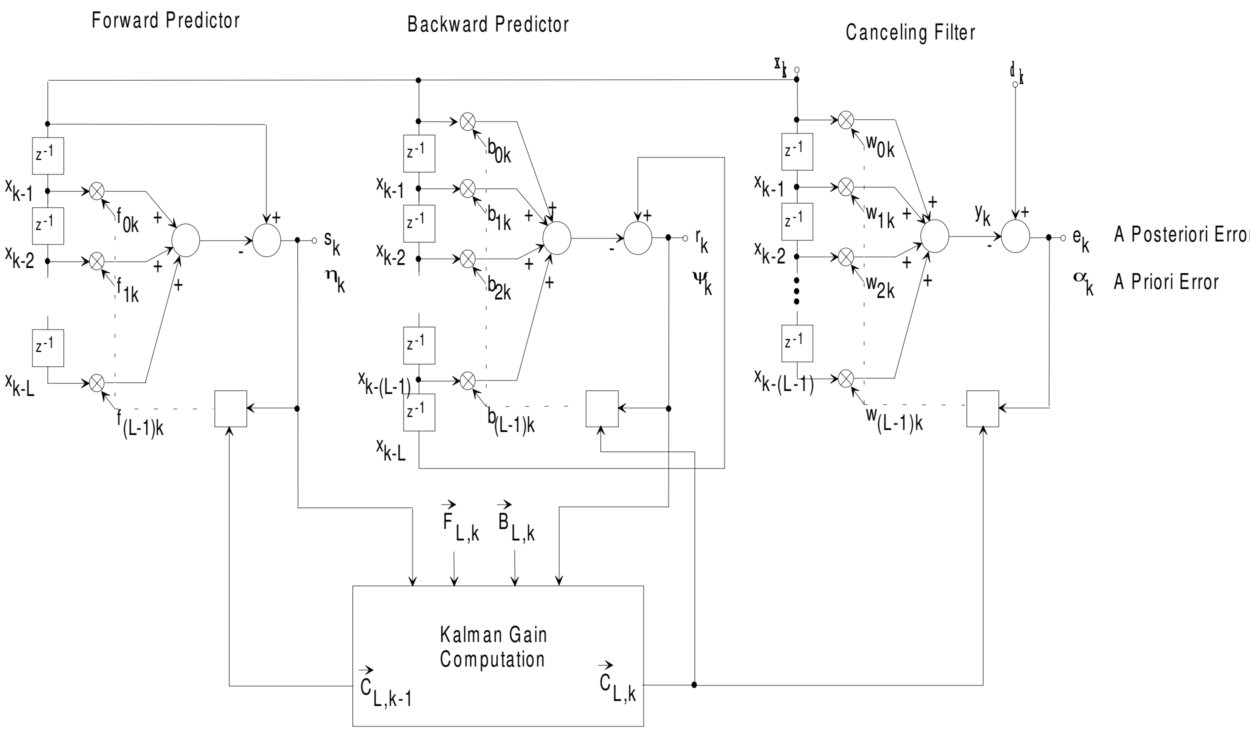


Fig. 18: Overview of the FTF-algorithm.

Before we have a closer look at forward and backward linear prediction filters, let us find out an important relationship concerning $\underline{\Phi}_{L,k}$. We recall that

$$\underline{\Phi}_{L,k} = \sum_{i=0}^{k} \vec{X}_{L,i} \cdot \vec{X}_{L,i}^{T} \cdot \lambda^{k-i}. \tag{78}$$

If we set i=u-1 and k=m-1 in (78) we obtain

$$\underline{\Phi}_{L,m-1} = \sum_{u=1}^{m} \vec{X}_{L,u-1} \cdot \vec{X}_{L,u-1}^{T} \cdot \lambda^{m-u} . \tag{79}$$

If we further define that $\vec{X}_{L,u} \equiv \vec{0} \quad for \quad u < 0$

we can finally write $\boxed{\underline{\Phi}_{L,k-1} = \sum_{i=0}^{k} \vec{X}_{L,i-1} \cdot \vec{X}_{L,i-1}^{T} \cdot \lambda^{k-i}}.$ (80)

Now let's turn to the estimation scheme for $\vec{C}_{L,k}$ which is used in FTFs makes use of linear prediction. Fig. 19 shows a simple model of a predictor.

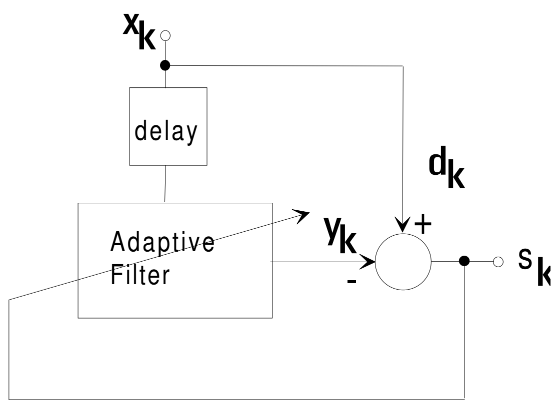


Fig. 19: Principle of a forward prediction filter.

The desired signal d_k is the input signal x_k itself.

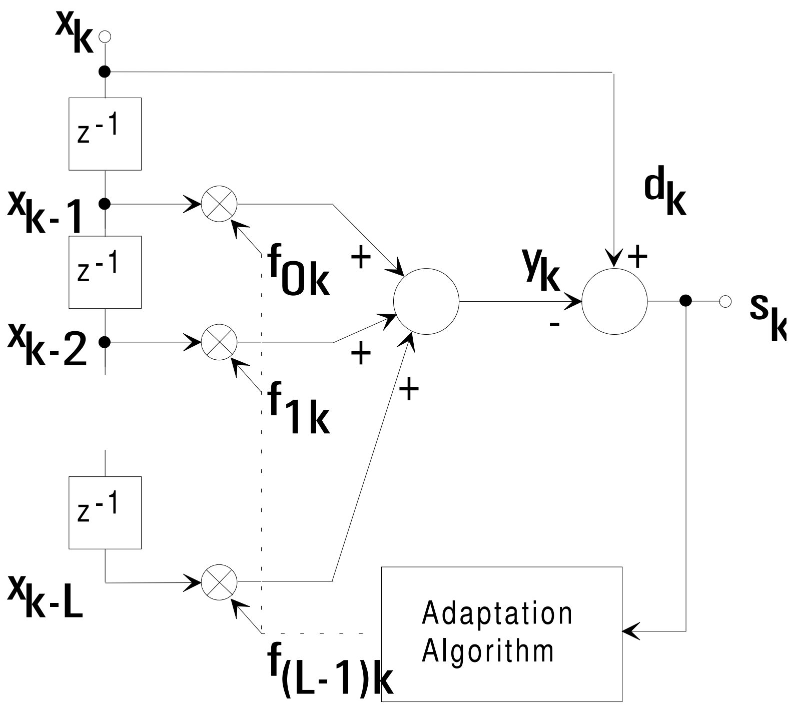


Fig. 20: Forward linear predictor (FLP) with just one clock cycle delay.

A delayed version of x_k is sent to the adaptive filter which therefore must try to "predict" the current input x_k in order to have y_k cancel d_k and drive s_k towards zero. Let's have a look at the special case of a forward linear predictor where the delay is just one clock cycle. The corresponding filter arrangement is shown in fig. 20 and looks very similiar to the echo cancelling filter in fig. 6 with some subtle differences, however. The desired signal is x_k instead of the microphone signal and the signal imposed upon the filter is $\vec{X}_{L,k-1}$ instead of $\vec{X}_{L,k}$. In analogy to the derivation of (63) we start defining

$$E'\left[s_k^2\right] \approx \sum_{i=0}^{k} \left(x_i - \vec{X}_{L,i-1}^T \cdot \vec{F}_{L,k} \right)^2 \cdot \lambda^{k-i} \tag{81}$$

where $\vec{F}_{L,k}$ is the weight vector of the forward prediction filter. Carrying through all the arithmetics which lead to (63) and making use of (80) we finally obtain

$$\vec{F}_{L,k}^{*} = \underline{\Phi}_{L,k-1}^{-1} \cdot \vec{\Theta}_{L,k}^{f} \tag{82}$$

with
$$\vec{\Theta}_{L,k}^{f} = \sum_{i=0}^{k} \vec{X}_{L,i-1}^{T} \cdot x_i \cdot \lambda^{k-i}. \tag{83}$$

We now follow the same procedures as we did for the canceling filter and try to find a recursive formula for $\vec{F}_{L,k}^{*}$. First of all we use (69) and substitute k-1 instead of k to get

$$\underline{\Phi}_{L,k-1}^{-1} = \lambda^{-1} \cdot \left(\underline{\Phi}_{L,k-2}^{-1} - \vec{C}_{L,k-1} \cdot \vec{X}_{L,k-1} \cdot \underline{\Phi}_{L,k-2}^{-1} \right). \tag{84}$$

Accordingly we obtain

$$\vec{C}_{L,k-1} = \underline{\Phi}^{-1}_{L,k-1} \cdot \vec{X}^{T}_{L,k-1} = \underline{\Phi}^{-1}_{L,k-1} \cdot \vec{X}_{L,k-1} \tag{85}$$

by employing (71). Using (84) and (85) in (82) yields in analogy to (72) the recursion formula

$$\vec{F}^{*}_{L,k} = \vec{F}^{*}_{L,k-1} + \vec{C}_{L,k-1} \cdot \left(x_k - \vec{X}_{L,k-1} \cdot \vec{F}^{*}_{L,k-1} \right). \tag{86}$$

So far it is not evident yet what we have gained by regarding the forward prediction filter in fig. 20. We have to be patient though and first regard the backward predictor in fig. 21.

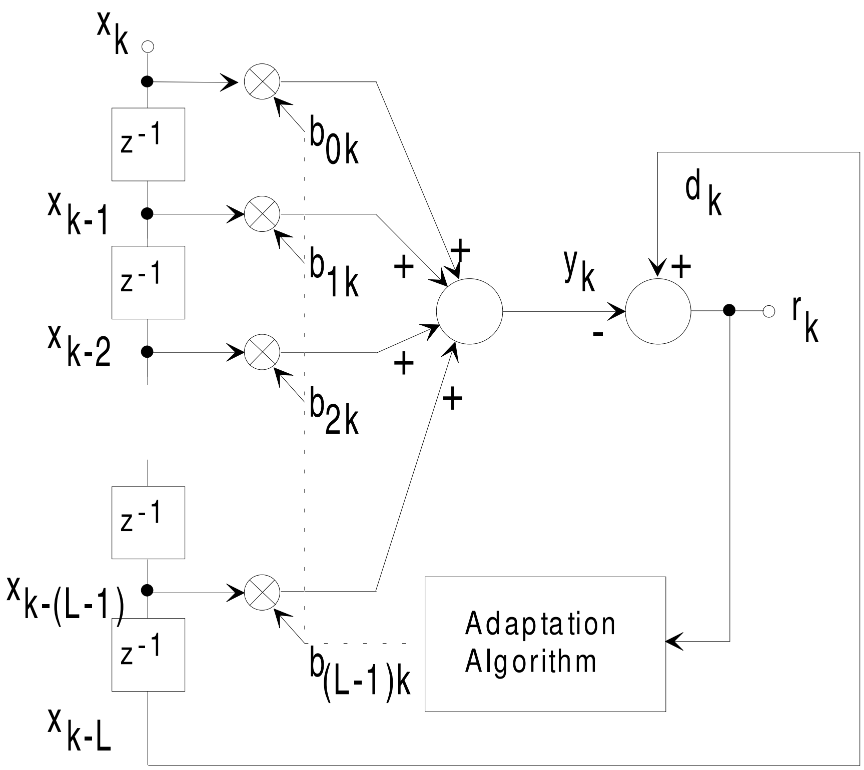


Fig. 21: Backward linear predictor (BLP) with just one clock cycle delay.

In case of the backward linear predictor, the filter tries to recover an old signal x_{k-L} which is "lost" already because it has been shifted out from the backward prediction filter. The backward linear predictor resembles the adaptive filter from fig. 6 even more than the forward linear predictor from fig. 20. The only difference to fig. 6 is that the desired signal d_k is specialized to be x_{k-L}. We therefore can immediately write in analogy to (72)

$$\vec{B}^{*}_{L,k} = \vec{B}^{*}_{L,k-1} + \vec{C}_{L,k} \cdot \left(x_{k-L} - \vec{X}_{L,k} \cdot \vec{B}^{*}_{L,k-1} \right). \tag{87}$$

Similiarly to (82) and (83) we can also deduce

$$\vec{B}^{*}_{L,k} = \underline{\Phi}^{-1}_{L,k} \cdot \vec{\Theta}^{b}_{L,k} \tag{88}$$

with

$$\vec{\Theta}^{b}_{L,k} = \sum_{i=0}^{k} \vec{X}^{T}_{L,i} \cdot x_{i-L} \cdot \lambda^{k-i}. \tag{89}$$

Using (71) we now define the extended gain vector $\vec{C}_{L+1,k}$ according to

$$\vec{C}_{L+1,k} = \underline{\Phi}_{L+1,k}^{-1} \cdot \vec{X}_{L+1,k}^{T} = \underline{\Phi}_{L+1,k}^{-1} \cdot \vec{X}_{L+1,k} \,. \tag{90}$$

The noteworthy feature of the extended gain vector is that, as will be shown later, it incorporates both the gain vector $\vec{C}_{L,k-1}$ in the adaptive forward linear predictor as well as the gain vector $\vec{C}_{L,k}$ in the adaptive backward linear predictor and the cancellation filter itself. This will lead to an efficient update strategy for the gain vector which is exactly what we have been looking for. In order to move on, we have to check how we can express $\underline{\Phi}_{L+1,k}^{-1}$ in terms of quantities that we already know. To this end we first write

$$\begin{aligned}
\underline{\Phi}_{L+1,k} &= \sum_{i=0}^{k} \vec{X}_{L+1,i} \cdot \vec{X}_{L+1,i}^{T} \cdot \lambda^{k-i} \\
&= \sum_{i=0}^{k} \begin{bmatrix} x_i \\ \vec{X}_{L,i-1} \end{bmatrix} \cdot \begin{bmatrix} x_i , \vec{X}_{L,i-1}^{T} \end{bmatrix} \cdot \lambda^{k-i} \\
&= \sum_{i=0}^{k} \begin{bmatrix} (x_i)^2 & x_i \cdot \vec{X}_{L,i}^{T} \\ \vec{X}_{L,i}^{T} \cdot x_i & \vec{X}_{L,i-1} \cdot \vec{X}_{L,i-1}^{T} \end{bmatrix} \cdot \lambda^{k-i} \\
&= \begin{bmatrix} \sum_{i=0}^{k} (x_i)^2 \cdot \lambda^{k-i} & \vec{\Theta}_{L,k}^{f} \\ \vec{\Theta}_{L,k}^{f} & \underline{\Phi}_{L,k-1} \end{bmatrix}
\end{aligned} \tag{91}$$

In a similiar fashion we can deduce

$$
\begin{aligned}
\underline{\Phi}_{L+1,k} &= \sum_{i=0}^{k} \vec{X}_{L+1,i} \cdot \vec{X}^{T}_{L+1,i} \cdot \lambda^{k-i} \\
&= \sum_{i=0}^{k} \begin{bmatrix} \vec{X}_{L,i} \\ x_{i-L} \end{bmatrix} \cdot \begin{bmatrix} \vec{X}^{T}_{L,i}, x_{i-L} \end{bmatrix} \cdot \lambda^{k-i} \\
&= \sum_{i=0}^{k} \begin{bmatrix} \vec{X}_{L,i} \cdot \vec{X}^{T}_{L,i} & \vec{X}_{L,i} \cdot x_{i-L} \\ x_{i-L} \cdot \vec{X}^{T}_{L,i} & \left(x_{i-L}\right)^2 \end{bmatrix} \cdot \lambda^{k-i} \\
&= \begin{bmatrix} \underline{\Phi}_{L,k} & \sum_{i=0}^{k} \vec{X}_{L,i} \cdot x_{i-L} \cdot \lambda^{k-i} \\ \sum_{i=0}^{k} x_{i-L} \cdot \vec{X}^{T}_{L,i} \cdot \lambda^{k-i} & \sum_{i=0}^{k} \left(x_{i-L}\right)^2 \cdot \lambda^{k-i} \end{bmatrix} \\
&= \begin{bmatrix} \underline{\Phi}_{L,k} & \vec{\Theta}^{b}_{L,k} \\ \vec{\Theta}^{b}_{L,k} & \sum_{i=0}^{k} \left(x_{i-L}\right)^2 \cdot \lambda^{k-i} \end{bmatrix}
\end{aligned}
\tag{92}
$$

Now let's reexamine the expression (81) for the expected error in the forward error predictor of fig. 19 to find

$$
\begin{aligned}
E'\left[s_k^2\right] &\approx \sum_{i=0}^{k} \left(x_i - \vec{X}^{T}_{L,i-1} \cdot \vec{F}_{L,k} \right)^2 \cdot \lambda^{k-i} \\
&= \sum_{i=0}^{k} \left(\begin{bmatrix} x_i, \vec{X}^{T}_{L,i-1} \end{bmatrix} \cdot \begin{bmatrix} 1 \\ -\vec{F}_{L,k} \end{bmatrix} \right)^2 \cdot \lambda^{k-i} \\
&= \sum_{i=0}^{k} \left(\vec{X}^{T}_{L+1,i} \cdot \begin{bmatrix} 1 \\ -\vec{F}_{L,k} \end{bmatrix} \right)^2 \cdot \lambda^{k-i} \\
&= \ldots \\
&= \begin{bmatrix} 1, -\vec{F}^{T}_{L,k} \end{bmatrix} \cdot \underline{\Phi}_{L+1,i} \cdot \begin{bmatrix} 1 \\ -\vec{F}_{L,k} \end{bmatrix}
\end{aligned}
$$

and together with (91)

$$
\begin{aligned}
E'\left[s_k^2\right] &\approx \left[1, -\vec{F}_{L,k}^T\right] \cdot \begin{bmatrix} \sum_{i=0}^{k} (x_i)^2 \cdot \lambda^{k-i} & \vec{\Theta}_{L,k}^{f} \\ \vec{\Theta}_{L,k}^{f} & \underline{\Phi}_{L,k-1} \end{bmatrix} \cdot \begin{bmatrix} 1 \\ -\vec{F}_{L,k} \end{bmatrix} \\
&= \left[1, -\vec{F}_{L,k}^T\right] \cdot \begin{bmatrix} \sum_{i=0}^{k} (x_i)^2 \cdot \lambda^{k-i} - \vec{\Theta}_{L,k}^{f} \cdot \vec{F}_{L,k} \\ \vec{\Theta}_{L,k}^{f} - \vec{\Theta}_{L,k}^{f} \cdot \vec{F}_{L,k} \end{bmatrix} \\
&= \left[1, -\vec{F}_{L,k}^T\right] \cdot \begin{bmatrix} \sum_{i=0}^{k} (x_i)^2 \cdot \lambda^{k-i} - \vec{\Theta}_{L,k}^{f} \cdot \vec{F}_{L,k} \\ \vec{0}_L \end{bmatrix} \\
&= \sum_{i=0}^{k} (x_i)^2 \cdot \lambda^{k-i} - \vec{\Theta}_{L,k}^{f} \cdot \vec{F}_{L,k}
\end{aligned}
\tag{.93}
$$

The derivation of (93) reveals that the equation

$$
\underline{\Phi}_{L+1,i} \cdot \begin{bmatrix} 1 \\ -\vec{F}_{L,k} \end{bmatrix} = \begin{bmatrix} E'\left[s_k^2\right] \\ \vec{0}_L \end{bmatrix} \tag{94}
$$

must hold. Now we are prepared to prove the validity of the ingenious identity

$$
\boxed{\underline{\Phi}_{L+1,k}^{-1} = \begin{bmatrix} 0 & \vec{0}_L^T \\ \vec{0}_L & \underline{\Phi}_{L,k-1}^{-1} \end{bmatrix} + \frac{1}{E'\left[s_k^2\right]} \cdot \begin{bmatrix} 1 \\ -\vec{F}_{L,k} \end{bmatrix} \cdot \left[1, -\vec{F}_{L,k}^T\right]}. \tag{95}
$$

To this end we multiply (95) from the left with (91) and find after some lengthy arithmetic that (95) is indeed valid. We eventually can use (95) in (90) to get

$$
\begin{aligned}
\vec{C}_{L+1,k} &= \underline{\Phi}_{L+1,k}^{-1} \cdot \vec{X}_{L+1,k}^T = \underline{\Phi}_{L+1,k}^{-1} \cdot \begin{bmatrix} x_i \\ \vec{X}_{L,i-1} \end{bmatrix} \\
&= \begin{bmatrix} 0 & \vec{0}_L^T \\ \vec{0}_L & \underline{\Phi}_{L,k-1}^{-1} \end{bmatrix} \cdot \begin{bmatrix} x_i \\ \vec{X}_{L,i-1} \end{bmatrix} + \frac{1}{E'\left[s_k^2\right]} \cdot \begin{bmatrix} 1 \\ -\vec{F}_{L,k} \end{bmatrix} \cdot \left[1, -\vec{F}_{L,k}^T\right] \cdot \begin{bmatrix} x_i \\ \vec{X}_{L,i-1} \end{bmatrix} \\
&= \begin{bmatrix} 0 \\ \vec{C}_{L,k-1} \end{bmatrix} + \frac{x_i - \vec{F}_{L,k}^T \cdot \vec{X}_{L,i-1}}{E'\left[s_k^2\right]} \cdot \begin{bmatrix} 1 \\ -\vec{F}_{L,k} \end{bmatrix}
\end{aligned}
$$

and finally
$$\boxed{\vec{C}_{L+1,k} = \begin{bmatrix} 0 \\ \vec{C}_{L,k-1} \end{bmatrix} + \frac{s_k}{E'\left[s_k^2\right]} \cdot \begin{bmatrix} 1 \\ -\vec{F}_{L,k} \end{bmatrix}}. \tag{96}$$

In a similiar fashion we can prove the identity
$$\boxed{\underline{\Phi}_{L+1,k}^{-1} = \begin{bmatrix} \underline{\Phi}_{L,k}^{-1} & \vec{0}_L \\ \vec{0}_L^T & 0 \end{bmatrix} + \frac{1}{E'\left[r_k^2\right]} \cdot \begin{bmatrix} -\vec{B}_{L,k} \\ 1 \end{bmatrix} \cdot \left[-\vec{B}_{L,k}^T, 1 \right]} \tag{97}$$

which, together with (90) yields
$$\boxed{\vec{C}_{L+1,k} = \begin{bmatrix} \vec{C}_{L,k} \\ 0 \end{bmatrix} + \frac{r_k}{E'\left[r_k^2\right]} \cdot \begin{bmatrix} -\vec{B}_{L,k} \\ 1 \end{bmatrix}} \tag{98}$$

We have now all results to move on to the fast computation of the RLS algorithm via FTFs.

The fast computation of the RLS algorithm employs the following basic idea:

1) Compute an estimate of $\vec{C}_{L,k-1}$
2) Update the FLP according to (86)
3) Compute $\vec{C}_{L+1,k}$ according to (96)
4) Use (87) in (98) and solve for $\vec{C}_{L,k}$
5) Update the BLP according to (87)
6) Finally update the echo cancelling filter in (72)

Now the recursion begins when we start again with 2) while the time index k has incremented by one. RLS and FTFs have the disadvantage of exhibiting numerical instabilities so that the filters have to be reinitialized [33]. In addition RLS and FTF require high numerical precision which makes this class of algorithms unattractive for fixed point implementations.

6.3 Fast Newton Filters

The basic idea is to reduce the order of the FLP and BLP in the FTF-algorithm and use a Levinson-type algorithm to do the necessary order update of the FLP and BLP. We don't go further into the details here but refer to [34] and [35].

6.4 The Fast Affine Projection Algorithm

The Fast Affine Projection Algorithm (FAP) [36] ... [43] has spawned quite some interest recently as FAP provides RLS-like convergence properties with O(L) computational effort. FAP is the fast evaluation of the Affine Projection Algorithm (APA) [9], [37], [41], which can be formulated according to [36]
$$\vec{W}_{L,k+1} = \vec{W}_{L,k} + 2\mu \cdot \underline{X}_{LxN,k} \cdot \vec{\varepsilon}_k \tag{99}$$

where $$\vec{\varepsilon}_k = \left[\underline{X}^T_{LxN,k} \cdot \underline{X}_{LxN,k} + \delta \cdot \underline{I}\right]^{-1} \cdot \vec{e}_k \tag{100}$$

and $$\vec{e}_k = \vec{d}_k - \underline{X}^T_{LxN,k} \cdot \vec{W}_{L,k} \tag{101}$$

with $$\underline{X}_{LxN,k} = \left[\vec{X}_{L,k}, \vec{X}_{L,k-1}, \ldots, \vec{X}_{L,k-N+1}\right], \tag{102}$$

$$\vec{e}_k = \begin{pmatrix} e_k \\ e_{k-1} \\ . \\ . \\ e_{k-N+1} \end{pmatrix} \tag{103}$$

and $$\vec{d}_k = \begin{pmatrix} d_k \\ d_{k-1} \\ . \\ . \\ d_{k-N+1} \end{pmatrix}. \tag{104}$$

Similiar to the Row Array Projections (RAP) from chapter 5.2.1 the N most recent input signal vectors $\vec{X}_{L,k}, \vec{X}_{L,k-1}, \ldots, \vec{X}_{L,k-N+1}$ are used to form a matrix $\underline{X}_{LxN,k}$ where N < L. This matrix has a lower rank than the input correlation matrix $\underline{R}_{L,k}$. Note that for N=1 (99) reduces to the NLMS algorithm. Like for RLS fast techniques for the matrix inversion can be used, but again high precision computations are required to do this.

6.6 Subband Approaches

The motivation for adaptive filtering in subbands stems from several well-known problems in full-band filtering:

1) Convergence and tracking can be very slow if the input correlation matrix is ill conditioned, as in the case with speech input, and if NLMS is used.
2) If fast-converging full-band methods like RLS, FTF or FAP are used, high numerical precision is required and the algorithms are potentially unstable.
3) Even for NLMS-based AECs high numerical precision is required if the number of weights is large (e.g. L=512). This is due to the fact that the weight update increments are getting smaller with increasing L.
4) High order full-band adaptive filters are computationally very expensive so that a single DSP is often not sufficient to handle the computational load.

Subbanding as a divide and conquer approach splits the signals into several frequency bands and employs independently operating adaptive filters within each subband. Subbanding is designed to fulfill two tasks:

i) Split a signal into several independent frequency bands to whiten the signal in each band [44]. The decorrelating effect of subbanding is already known from the TDLMS approach and helps to improve convergence speed.

ii) Employ multirate methods [45] to reduce the sampling rate within the subbands to save computational effort and to reduce the filter length of the subband adaptive filters. Reduced length adaptive filters have relaxed precision requirements which is crucial in fixed point implementations. The reduced length also helps in terms of convergence speed which, however, is counteracted by the fact that each adaptive filter is updated less often.

For low-precision fixed point implementations of AEC subbanding is indeed the only viable approach.

Fig. 22 shows a block diagram of a subband AEC with two subbands but no sampling rate reduction. Fig. 22 also indicates that real-world highpass and lowpass filters are not perfect "brick wall" filters but have transition bands in their frequency response which separates pass band and stop band. The non-perfect nature of the transition band is one of the reasons why the reconstruction of a signal from subband signals is only an approximation. Bandgaps and/or aliasing lead to lost information or artifacts in the reconstructed signal. Other imperfections are amplitude and phase distortions in the pass band as well as insufficient stop band attenuation. The goal, of course, is to approximate an ideal "brick wall" filter as best as possible because it not only causes the least reconstruction errors but also provides the best whitening effect for the input signal [46]. This whitening effect, however, is not as good as one could hope for [44], [46]. Especially the band edges are the cause for some small eigenvalues in the subbands which slow down the convergence of the subband adaptive filters especially if NLMS is the adaptive algorithm. In [44] it was also mentioned that the occurrence of band gaps aggravates the problem of small eigenvalues. In order to approximate an ideal filter, the transition band has to be very narrow which in turn requires a very large number of weights if the subband filters are of FIR type. For practical purposes the number of filter weights lies in the hundreds, resulting in non-negligible computational effort and, most notably, delay in the AEC. Henceforth IIR-filters are worth considering, as they require less effort and cause less delay. A drawback is that linear phase is not possible with IIR-Filters, yet mild phase distortions are usually not perceived by the human auditory system if the input signal is speech.

The idea of sample rate alteration is sketched in fig. 23, again for a two-band scheme. It can be seen that filtering and subsampling leads to either bandgaps or aliasing. Yet there are filter design strategies which provide perfect reconstruction due to aliasing cancellation [45]. Perfect reconstruction, however, requires linear phase FIR filters with the associated drawbacks of high cost and large delay. In addition, the perfect reconstruction constraints are violated by the artificial echo path which is modeled by the adaptive filters. In order to reestablish perfect reconstruction, cross terms [44], [47] have to be taken into consideration which increase the computational cost but seem to have only very little beneficial effect in terms of the cancellation of the echo signal [47].

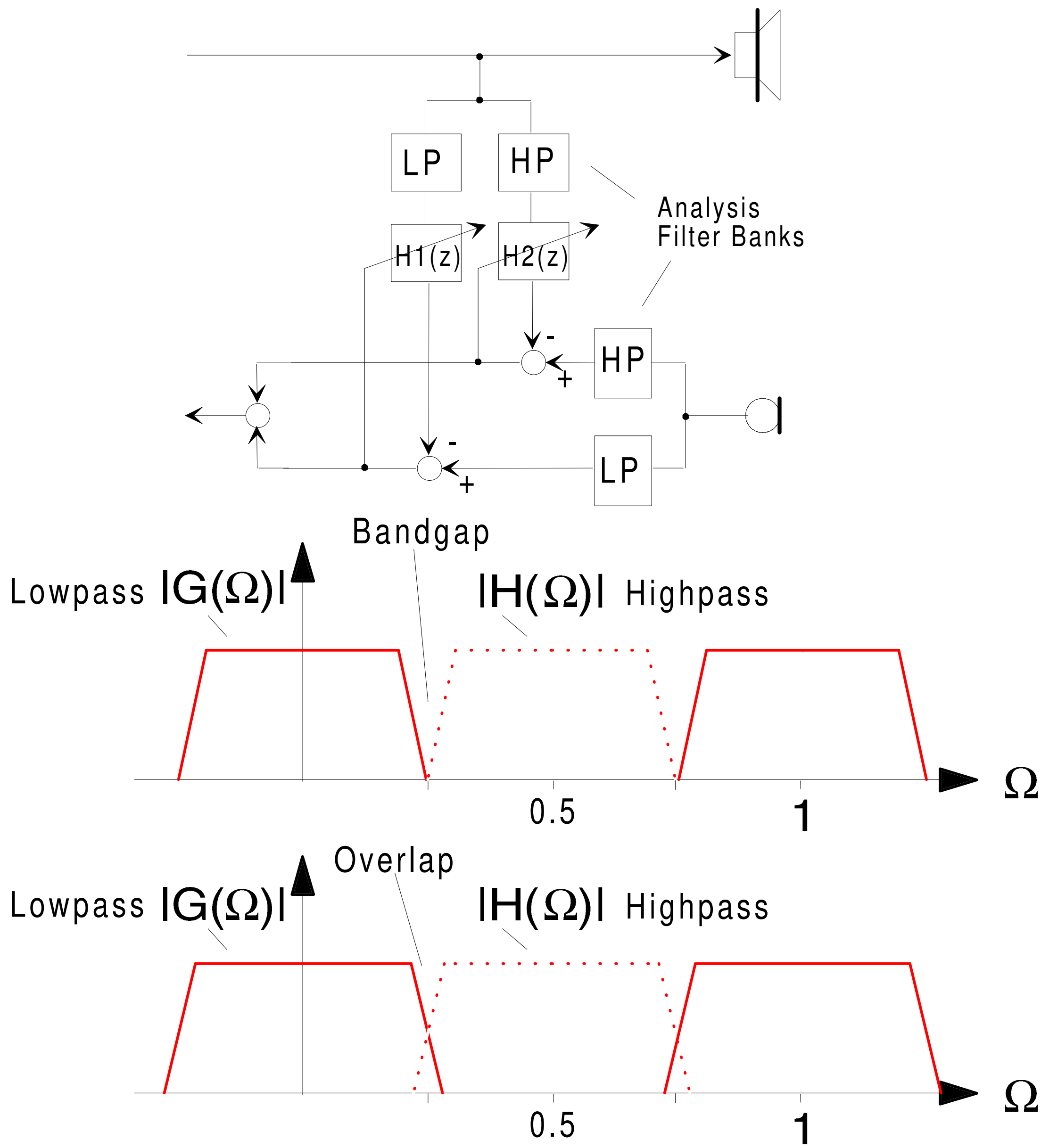


Fig. 22: Subband decomposition showing bandgaps and aliasing as causes for reconstruction errors.

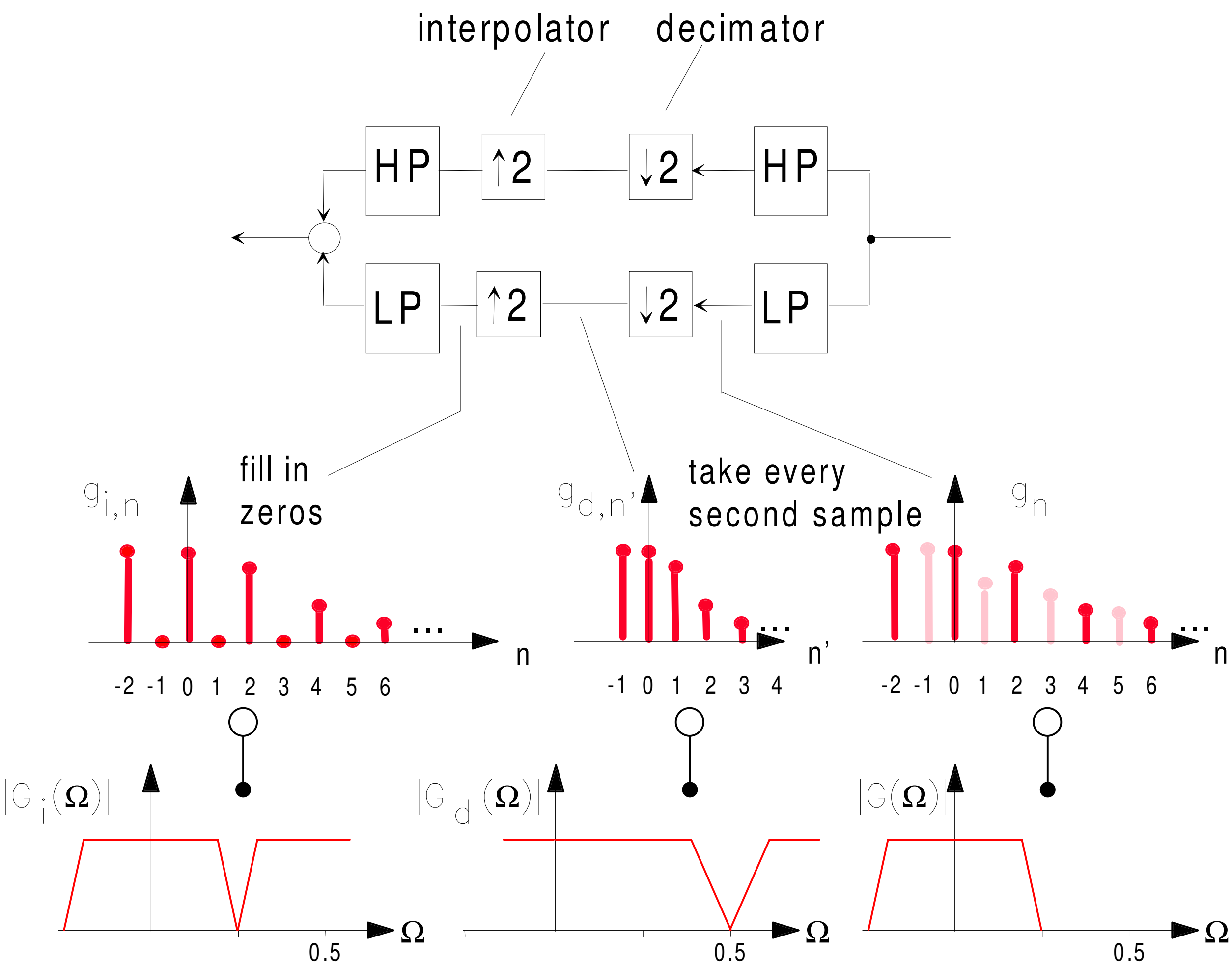


Fig. 23: Subbanding employing multirate methods.

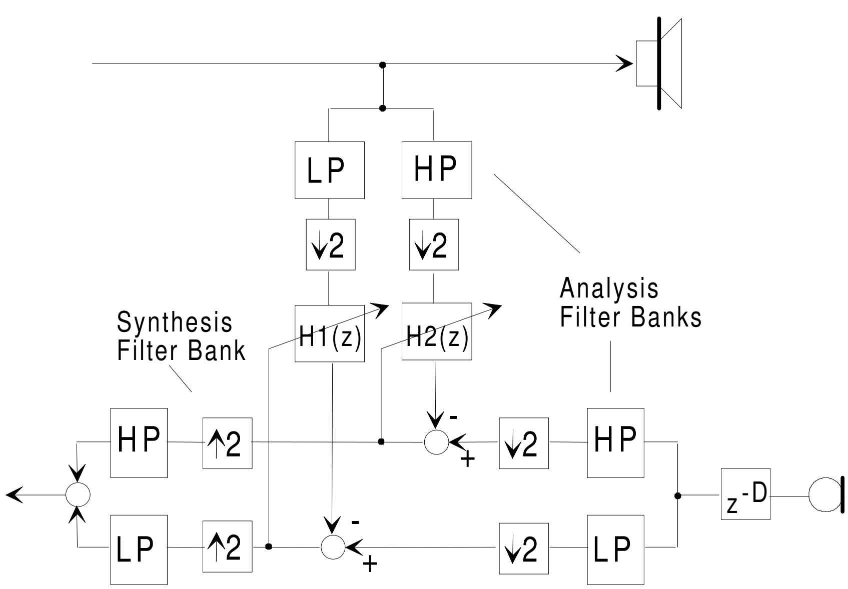


Fig. 24: Subband Echo Canceler with two subbands and critical sampling.

A basic two-subband AEC structure is shown in fig. 24. Most notably there is a delay z^{-D} which delays the microphone signal and enables the adaptive filters to have "leading taps", i.e. a slightly non-causal impulse response. This interesting phenomenon that non-causality appears in subband AECs is further explained in [9] and [48].

Most subband implementations to date make use of fractional sampling in order to prevent aliasing as aliasing is a major cause for residual echo. In addition the stopband attenuation requirements can be fulfilled more easily. The idea of fractional sampling is indicated in fig. 25. Disadvantages are increased computational cost and the presence of band gaps which slow down convergence of the adaptive filters [44].

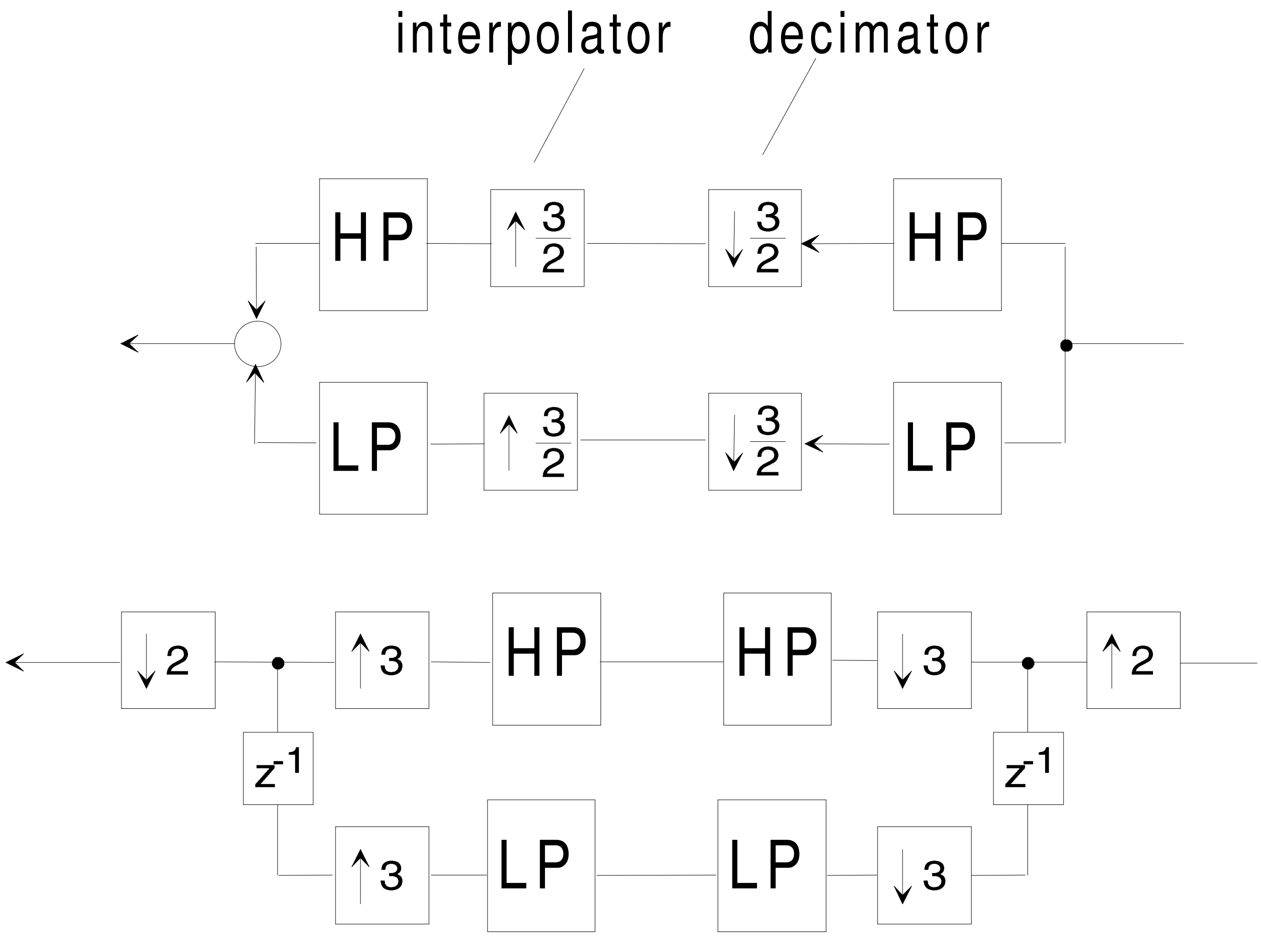


Fig. 25: Fractional subsampling with theoretical structure (above) and efficient polyphase structure (below).

As has been mentioned before, IIR-filters have distinct advantages when it comes to generating subband signals. One of the most cost-efficient solutions is the realization via allpass based filters [45], [49], [50]. It can be shown that if $A_1(z)$ and $A_2(z)$ are allpass exhibiting certain additional properties [45], then

$$H_0(z) = \frac{A_1(z) + A_2(z)}{2} \quad \text{is lowpass} \tag{105}$$

and

$$H_1(z) = \frac{A_1(z) - A_2(z)}{2} \quad \text{is highpass} \tag{106}$$

where $H_0(z)$ and $H_1(z)$ are power-complementary, i.e. at least the magnitude can be perfectly reconstructed in a multirate system with critical sampling. An additional advantage of allpass-based IIR-filters is their low sensitivity to coefficient quantization, something very important in fixed point implementations [49]. A subband analysis/synthesis unit with two subbands is depicted in fig. 26.

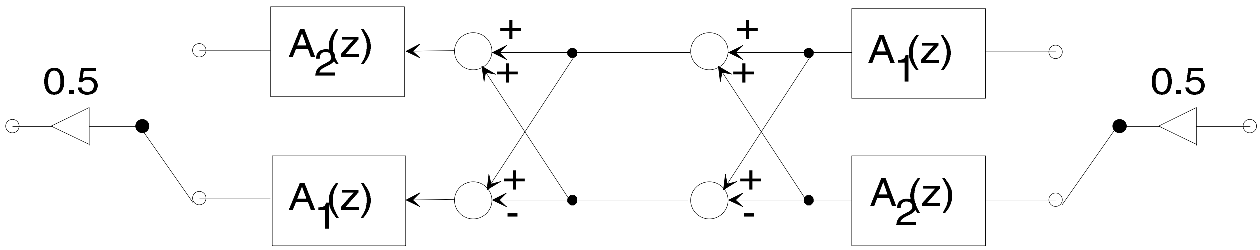


Fig. 26: Analysis/Synthesis unit based on an allpass decomposition.

$A_1(z)$ and $A_2(z)$ are conventiently implemented via first and second order allpass structures, some suitable examples of which can be found in fig. 27.

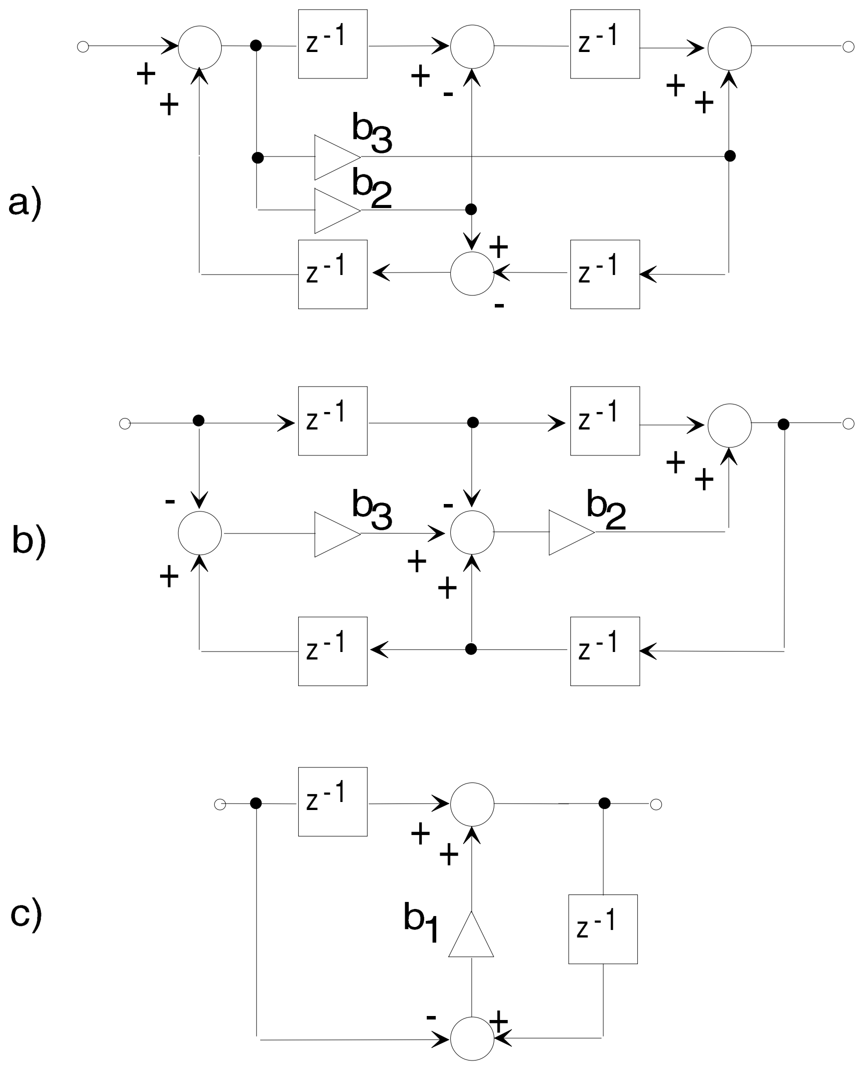


Fig. 27: First and second order allpass structures taken from [50].

A nice property of the decomposition following eqs. (105) and (106) as described in [45], [49] is the fact that a wide family of practical transfer functions including Butterworth, Chebyshev, and elliptic filters can be represented according to (105), (106) thus mitigating the design task of the allpass filters.

Another approach for the design of allpass filters is taken in [51], where

$$H_0(z) = 0.5 \cdot \left(\prod_{j=0}^{P_0-1} \frac{\alpha_{0,j} + z^{-2}}{1 + \alpha_{0,j} \cdot z^{-2}} + z^{-1} \cdot \prod_{j=0}^{P_1-1} \frac{\alpha_{1,j} + z^{-2}}{1 + \alpha_{1,j} \cdot z^{-2}} \right) \tag{107}$$

and

$$H_1(z) = 0.5 \cdot \left(\prod_{j=0}^{P_0-1} \frac{\alpha_{0,j} + z^{-2}}{1 + \alpha_{0,j} \cdot z^{-2}} - z^{-1} \cdot \prod_{j=0}^{P_1-1} \frac{\alpha_{1,j} + z^{-2}}{1 + \alpha_{1,j} \cdot z^{-2}} \right) \tag{108}$$

The filter structures which are suited to implement (107) and (108) are depicted in fig. 28.

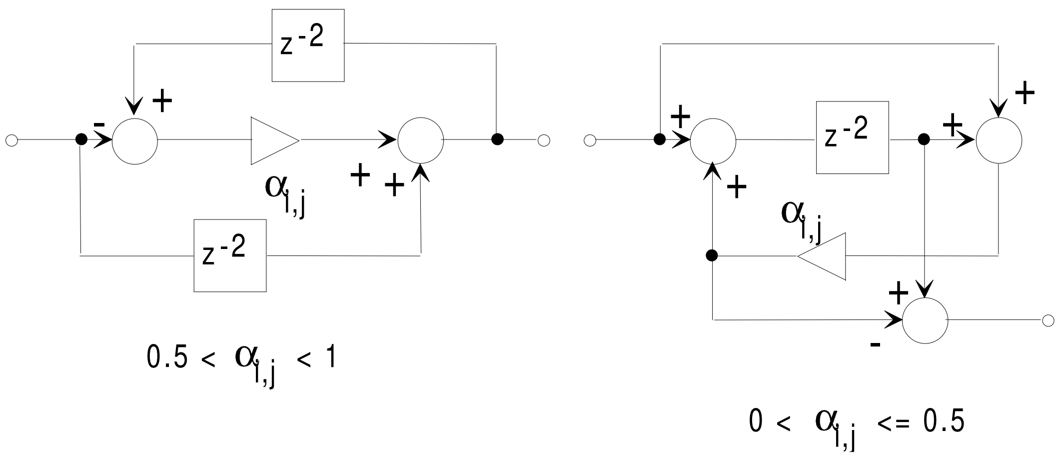


Fig. 28: Second order filter structures used to implement (107) and (108).

The coefficients of the filters of fig. 28 are designed via nonlinear optimization techniques and chosen such that the poles of $H_0(z)$ are located on the imaginary axis while the zeroes stay on the unit circle.

6.8 Adaptive IIR-Filters

Room impulse responses are characterized by fairly long tails due to multiple reflections of acoustic waves in the room. Hence it is intriguing to use adaptive IIR-Filters instead of adaptive linear combiners to model the echo path. Adaptive IIR-Filters seem promising as they are able to create long inpulse responses with very little computational effort. Indeed there has been already substantial research in the area of adaptive IIR-filters [52] ... [57] and there are already several investigations about IIR-based AECs, e.g. in [58], [59]. Yet there are many problems with adaptive IIR-Filters such as multiple minima, stability problems and high precision requirements.

6.9 Additional Techniques

6.9.1 Silence and Near-end Talk Detection

Silence and double talk detection can help to prevent an already adapted filter from loosing its proper weights by freezing the weight update once a silent period or a near end talk situation is detected. During silent periods the ambient noise can cause the filter weights to drift which is why the weight update should be stopped. Silence detection is performed by either counting zero crossings of the microphone signal or by measuring the signal power of the microphone signal. Many zero crossings indicate noise while a low number of zero crossings indicate speech. Power measurements use long and short term averages. The short term average value has to exceed the long term value by a certain amount in order to detect speech (either near or far end). This way speech can be distinguished from a rise in noise level. If the difference between short and long term averages is not large enough, speech is defined to be absent.

The second problematic situation occurs when the near end speaker talks. As the adaptive filter tries to minimize the output power the filter will change the weights to grossly wrong values due to the increase in output power. The near end talker acts as a noise source for the adaptive filter and hence the weight update should be stopped, once near end talk is detected. No standard method for near end talk detection has been established yet. One of the most common methods though is to run a second full band NLMS filter of short length. If this filter shows an increase in output power and the rate of increase is high enough it is assumed that near end talk has caused the power increase. Slowly increasing power values might be due to a change in room acoustics which should, of course, be tracked by the AEC.

6.9.2 Suppression of Residual Echo

Most AECs are not able to cancel the echo completely, yet customers generally demand to hear no echo at all. Therefore residual echo is usually suppressed rather than cancelled [7]. The prerequisite is that the AEC meets a certain quality standard. If this is the case, the microphone gain can be lowered if the echo-cancelled microphone signal falls below a certain threshold. Assuming that the AEC always manages to cancel the echo sufficiently such that the residual echo always falls below the mentioned threshold, residual echo can be suppressed. Suppression can also be used if the output signal rises above a certain threshold while the near end talker is not active. This way, echo can be suppressed if the far end talker talks very loud, increases his microphone gain a lot, or the near end talker increases the loudspeaker volume too much.

6.9.3 Compensation of Nonlinearities

Some improvement has been achieved by realizing that the path the potential echo signal is travelling through is not linear. Some of the nonlinearities are caused by the coil/loudspeaker-unit and have been taken into consideration by the use of a neural network approach [60]. Other nonlinearities such as appearing in D/A-converters have been treated in [61] and shown to have a non-negligible effect.

6.9.4 Beamforming and Directional Microphones

Some research has gone into the use of microphone beam narrowing or even beam steering via microphone arrays. Beam narrowing can also be realized via appropriate microphones. While this technique tries to prevent the loudspeaker signal from entering the microphone it also prevents side remarks from third persons e.g. in a conference. Therefore omnidirectional microphones are the preferred solution to make the conversation as natural as possible although this is the most difficult situation for an AEC.

7. The ICSI AEC (IAEC)

In the course of ICSIs "Robust Audio"-Project there have been several attempts to implement an AEC on a standard Unix-based workstation [62]. AEC, however, requires direct control over the microphone and loudspeaker to ensure proper timing relations of x_k, y_k, d_k and ε_k. Unfortunately this control is not supported by a standard Unix-workstation. In addition it has been found that an AEC requires a considerable amount of computation time so that all other applications run much slower. Especially for videoconferencing scenarios with application sharing a major slowdown can not be tolerated. The AEC-development was hence shifted to a low-cost digital signal processor (DSP) implementation on a 16-bit fixed point DSP. Table 1 provides a summary of the most important basic AEC-algorithms with regard to implementability under limited precision and limited computational resources.

AEC-Algorithm	Suitability for 16-bit fixed point implementation	computational cost	precision requirements	robustness	convergence speed	residual echo suppression
NLMS (fullband)	no	high	high	high	low	moderate
NLMS with decorrelation filters	no	high	very high	moderate	high	high
RAP	no	high	high	high	moderate	moderate
RLS	no	very high	very high	low	high	high
FTF	no	very high	very high	low	high	high
FAP	no	very high	very high	low	high	high
FIR-Subband-NLMS	yes	moderate	moderate	high	high	moderate
IIR-Subband-NLMS	yes	moderate	moderate	high	high	moderate
adaptive IIR	no	high	very high	low	high	moderate

Table 1: Overview of the most important AEC-algorithms.

The ADSP2181® from Analog Devices turned out to be the processor of choice as a low-cost evaluation board, the EZ-Kit Lite ®, is available (<$100) which offers analog/digital stereo in- and output. The basic experimental setup is sketched in fig. 30.

The Unix-workstations were located in different rooms and ran a full-duplex communication-SW like the one described in [62]. The experimental setup and subjective assessment was chosen over a simulation-based study for several reasons. Firstly, there are simulation studies galore in the scientific literature which, however, almost never take finite precision arithmetic into account. In addition, the room impulse response the AEC has to adapt to is usually a fixed one and does not consider the fact that an actual room impulse response is non-LTI. Yet both non-LTI properties and finite precision arithmetic fundamentally affect the convergence and cancellation properties of an AEC. In order to consider these effects theoretically and implement them on a suitable simulator the time frame for this investigation was too short. For this reason and because an AEC was required for ICSI's MAY project, a working prototype had priority even over meaningful simulations.

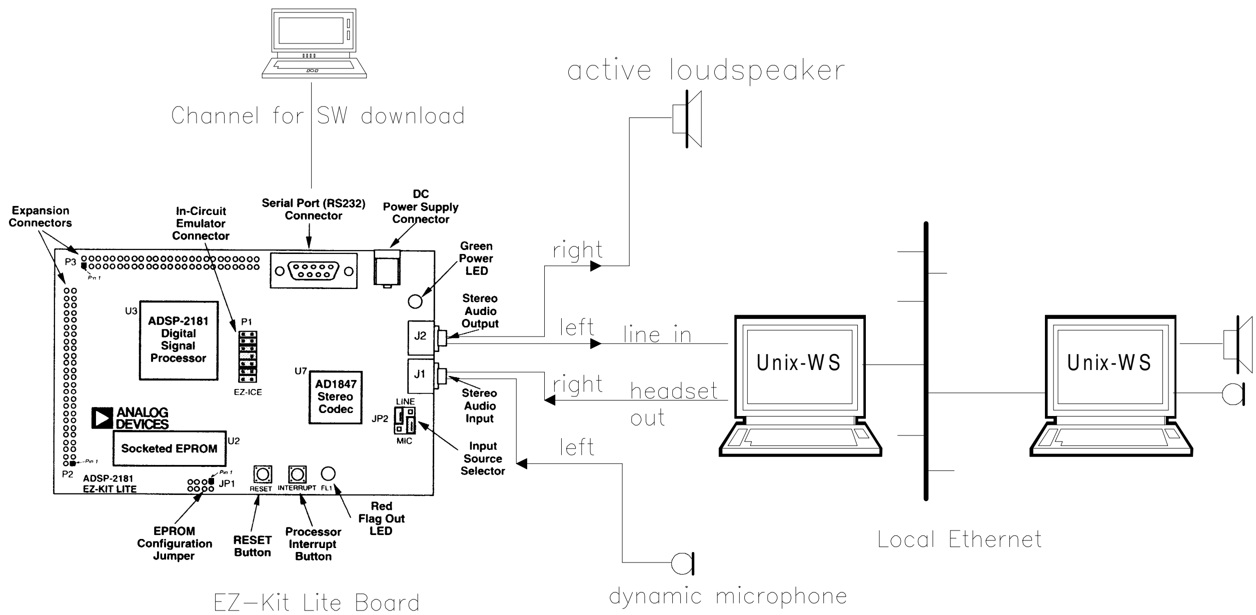


Fig. 30: Experimental setup for the development of the IAEC.

The basic structure of the IAEC is shown in fig. 31 with some implementation details in fig. 32. Apart from an allpass-based two-subband structure with a 256-tap variable step size NLMS adaptive filter in each subband, further elements such as bandstop filters to eliminate aliasing, microphone delay to implement leading taps as well as silence detection were added to the IAEC. The sampling frequency was 8kHz. In addition to the two subband solution, a 4- and 8-band solution was implemented which, however, subjectively brought no improved AEC behaviour. Although the 4- and 8-band solution converged faster than the 2-band AEC the cancellation depth was slightly inferior.

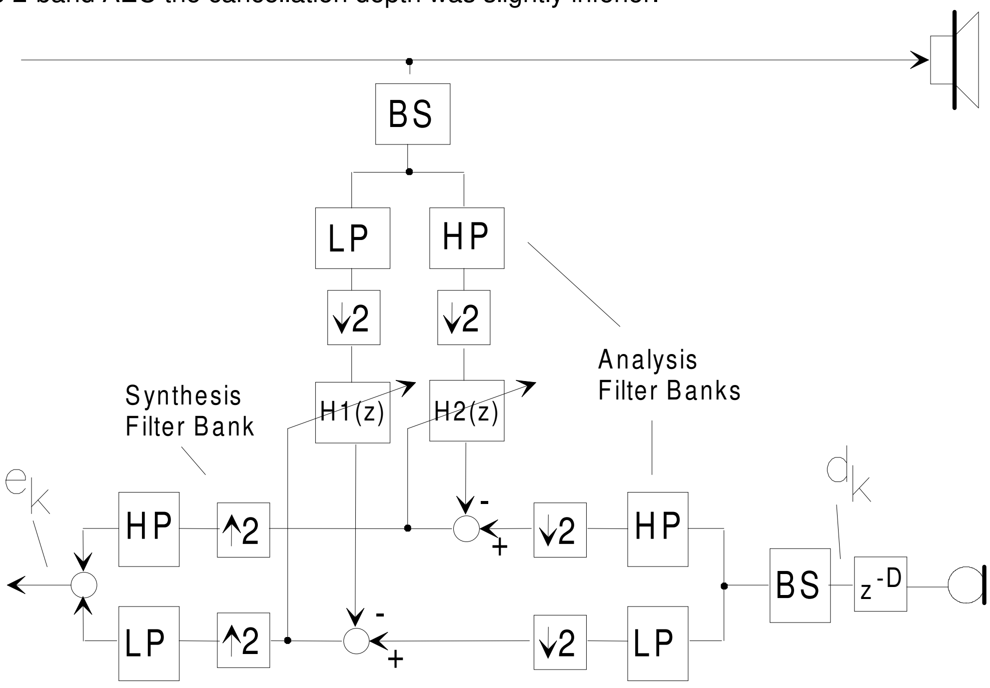


Fig. 31: Block diagram of the IAEC.

These findings are in agreement with the results presented in [44]. The reason for the subjectively superior performance of the 2-band solution is most probably due to the smaller reconstruction error and hence less residual echo.

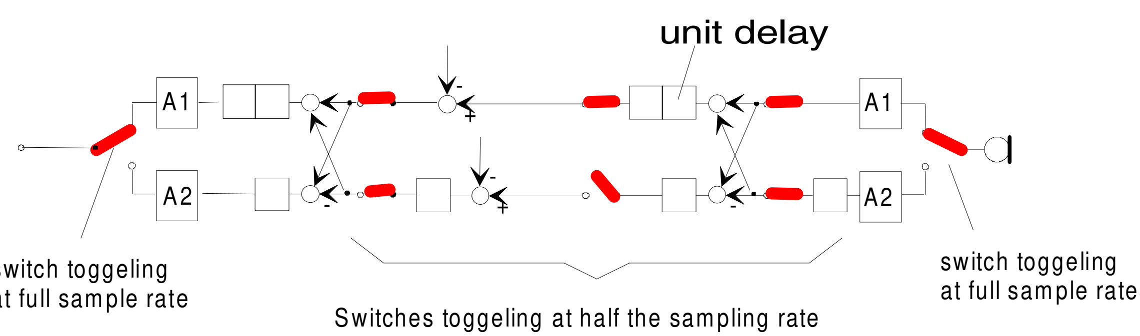


Fig. 32: Implementation details of the analysis/synthesis filter bank used in the IAEC

For the implementation of the filter bank the filters structures of figs. 26 and 27 were used with the lowpass prototype being based on a 19th degree elliptical filter the magnitude response of which is shown in fig. 33. The corresponding responses of the allpass based filters are depicted in fig. 34.

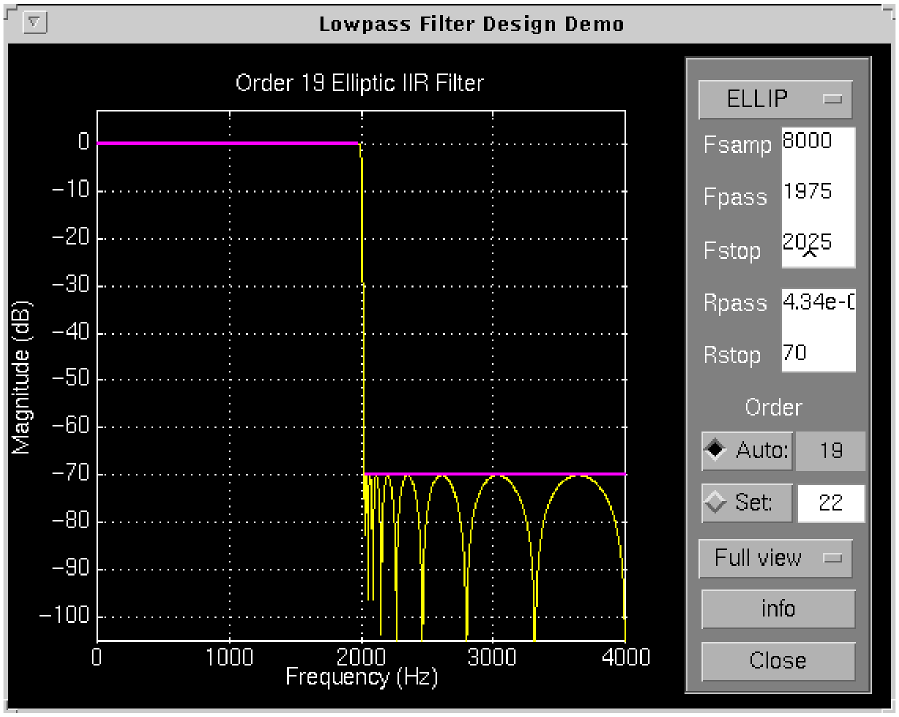


Fig. 33: MATLAB environment for the design of a 19th degree elliptical filter with a passband ripple of -4.343e-7dB and stopband attenuation of -70dB.

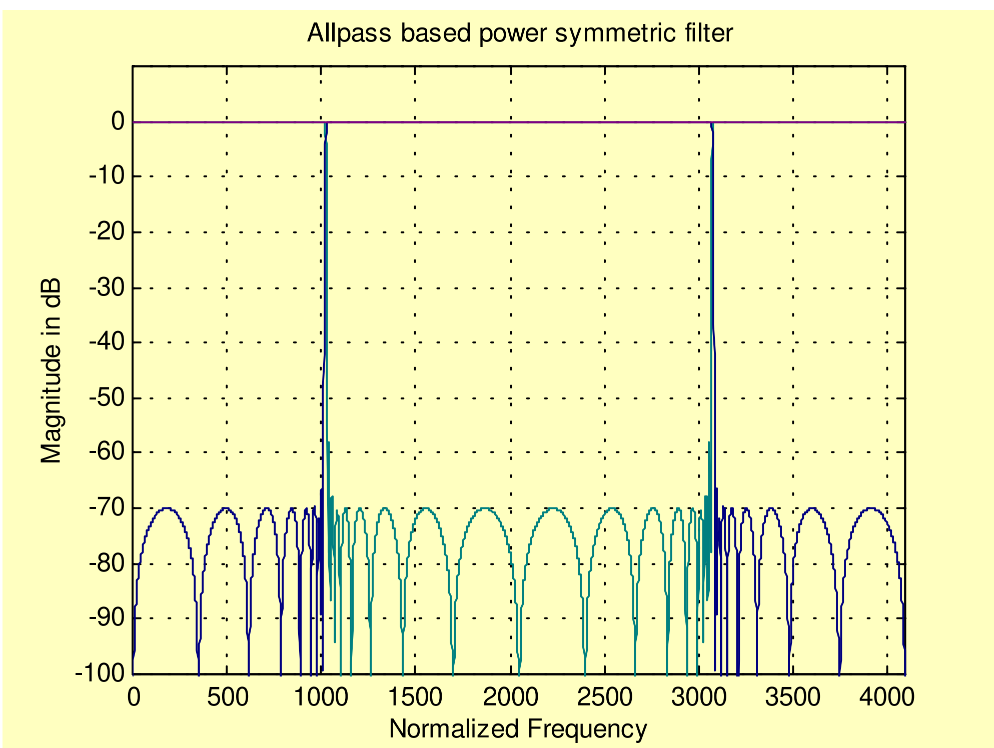


Fig. 34: Magnitude responses of the allpass-based high and lowpass filters and their sum.

The bandstop i.e. notch filters for the IAEC were designed ccording to [63] and have the transfer function

$$N(z) = 0.5 \cdot \left(\frac{\gamma_2 + \gamma_1 \cdot z^{-1} + z^{-2}}{1 + \gamma_1 \cdot z^{-1} + \gamma_2 \cdot z^{-2}} \cdot \frac{\gamma_2 - \gamma_1 \cdot z^{-1} + z^{-2}}{1 - \gamma_1 \cdot z^{-1} + \gamma_2 \cdot z^{-2}} + \frac{\varsigma + z^{-2}}{1 + \varsigma \cdot z^{-2}} \right) \quad (109)$$

with

$$\begin{aligned} \gamma_1 &= 0.171050 \\ \gamma_2 &= 0.891673 \\ \varsigma &= 0.806325 \end{aligned} \quad (110)$$

Fig. 35 shows the magnitude response of the allpass filters combined with the above notch filter. Experimental results with the IAEC have revealed that computing the NLMS-algorithm with 32-bit accuracy yields a notable improvement in cancellation depth. The disadvantage on a 16-bit DSP is, of course, that 32-bit computation has to be emulated and needs a significant amount of computation time. Another non-negligible improvement was caused by the introduction of leading taps as indicated in fig. 31 via the delay-element z^{-D}. The two-band version of the IAEC showed only little sensitivity to the stop-band attenuation of the subband filters. A reduction of the stopband attenuation from -70dB to -45dB caused no perceivable loss in cancellation quality. Also the inclusion of the notch-filters had, as opposed to the simulations given in [64], only a small positive effect in terms of perceived quality.

It has to be mentioned that the standard EZ-Kit Lite development environment is not well suited for measuring and tuning any application. Every change in parameters has to go through a reassembly and loading phase. The development process would be greatly simplified if paramters could be changed interactively. Likewise, the measurement of results is inconvenient to do.

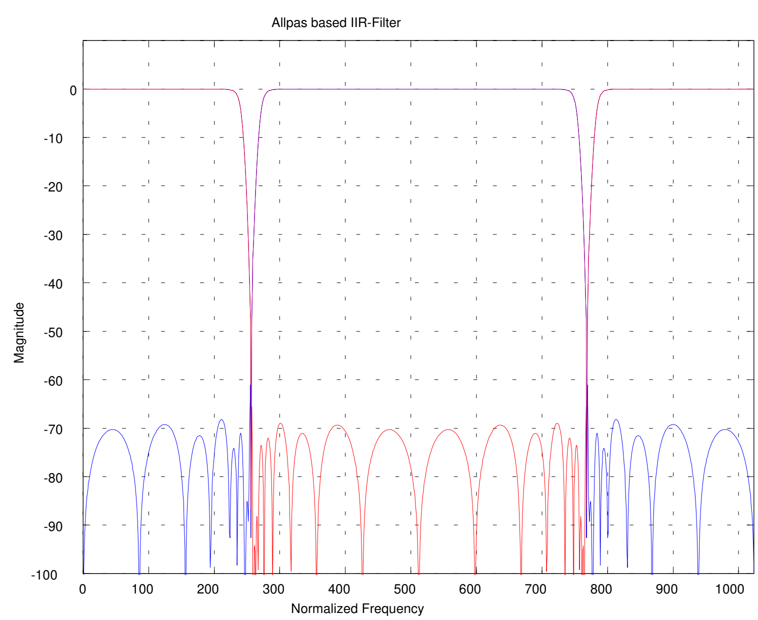


Fig. 34: Magnitude responses of the allpass-based high and lowpass filters combined with the notch filter from equ. (109), (110).

The AEC-code has to be instrumented with tracing-facilities the content of which is freezed in a pre- or posttrigger fashion. As there is no additional memory on the EZ-Kit Lite board other than the on-chip memory of the ADSP2181 the size of the trace stores is limited severely. After the trigger event the AEC has to be stopped, and control has to be returned to the monitor program so that the trace data have can be uploaded into the host computer. The uploaded data contain a header and a trailer which both have to be removed before the data, which are in hexadecimal format, can be converted to a decimal format which then can be processed further.

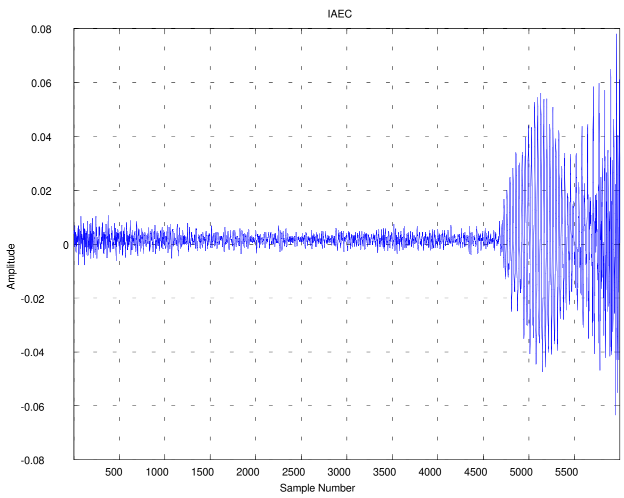


Fig. 35: Sample of microphone signal d_k (see fig. 31) resembling the start of the word "vier".

An integrated development environment which automates and facilitates these steps as far as possible would be of help to speed the development process. Figs. 35, 36 and 37 provide some measurement results obtained with the experimental setup. Figs. 34 and 35 demonstrate how the IAEC cancels the acoustical echo.

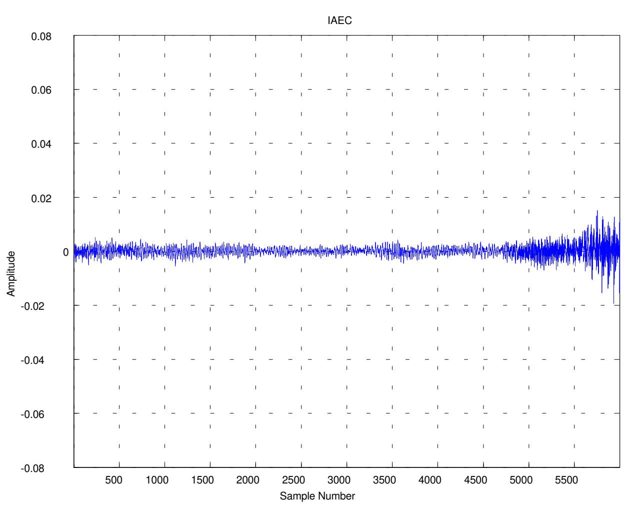


Fig. 35: Sample of echo-reduced microphone signal e_k (see fig. 31) resembling the start of the word "vier".

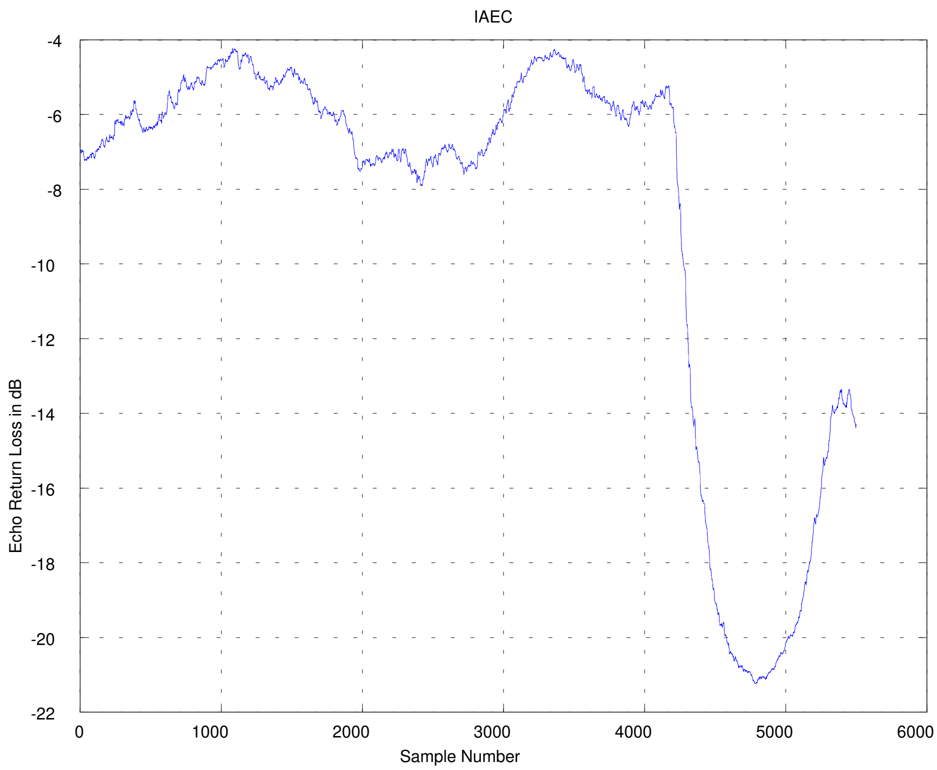


Fig. 36: Echo Return Loss computed with the help of the sequences in figs. 34 and 35.

Fig. 36 provides an illustration of the Echo Return Loss (ERL) which is the standard measure for AEC-performance and is computed via:

$$ERL = 10 \cdot \log_{10}\left(\frac{E\left[e_k^2\right]}{E\left[d_k^2\right]}\right). \tag{111}$$

The expected value is estimated as follows:

$$E\left[s_k^2\right] = \frac{1}{K}\sum_{i=1}^{K} s_k^2 \,. \tag{112}$$

It is apparent from fig. 36 that the ERL attenuation is very low during the noisy preamble and increases when the speech signal appears. As the sentence spoken into the microphone was "eins, zwei, drei, vier" with office noise in between the words, one can deduce that the IAEC "learns" the room impulse response during spoken phases and "forgets" it again when just the office noise remains. A remedy for this behaviour is to activate silence detection and freeze the coefficient update in the IAEC once the absence of speech is detected. Experiments with the IAEC do in fact support this conjecture and lead to deeper cancellation, yet howling removal is inferior. Apart from acoustical echo, howling is another feedback phenomenon which quickly builds up during noisy periods and is subjectively even more disturbing than the echo itself.

9. Conclusion and Future Work

In this work most of the currently known methods for acoustic echo cancellation (AEC) have been sketched and evaluated in terms of feasibility for implementation on a 16-bit fixed point digital signal processor (DSP). Fixed point implementation is considered important to obtain low cost AEC solutions which, in turn, are mandatory as the owner of the AEC only keeps his environment from producing acoustical echo, but does not cancel his own echo. Therefore all conference participants need to have an AEC in order to be able to conduct a natural full duplex conversation. It has turned out that subband AECs have the greatest potential of reducing the computational load and relaxing the accuracy requirements, both crucial for a fixed point DSP implementation. Due to the small inherent delay a subbanding scheme using allpass-based recursive digital filters was implemented on the ADSP2181. The AEC under investigation used a sampling frequency of 8kHz and two 256-tap NLMS adaptive filters of 32-bit wordlength. The implemented AEC manages to remove acoustic feedback howling completely but leaves a perceivable residual echo and hence has the status of "usable but improvable". Two subbands do not suffice to reduce the computational load and relax the accuracy requirements sufficiently. Eight or sixteen subbands seem more appropriate as the adaptive filters inside the subbands can be shorter and 16-bit computation might then suffice for the adaptation process. However, in order to get good results, the subband filters have to be of high quality, i.e. sharp cutoff and high stopband attenuation to yield a small reconstruction error. Introducing several bandstops to mask the reconstruction error are also worth considering. It has to be investigated whether or not the nonlinear phase of IIR-based subbanding schemes renders the audio quality still acceptable. In the negative case linear-phase FIR filters and fractional sampling has to be used, although overall delay will be increased and convergence be slowed down because of the softer band edges of the required medium-sized FIR filters. Special attention has to be paid to several nonlinear auxiliary techniques like silence/noise-detection which freezes the coefficient update during the absence of far-end speech. It is important, though, that the howling removal capability of the AEC be maintained. Also other auxiliary techniques such as double talk detection and residual echo suppression, possibly via frequency dependent suppressors, have to be examined further. Many more issues concerning AEC are of interest and have not been resolved yet. It is hoped that this survey serves as a motivation to continue research on this important and exciting topic.

10. References

[1] Widrow, B. and Stearns, S.D., Adaptive Signal Processing, Prentice-Hall, 1985.

[2] Hänsler, E., Adaptive Echo Compensation Applied to the Hands-Free Telephone Problem, IEEE Proc. CAS 1990, pp. 279 - 282.

[3] McCanne, S., An Echo Canceler for Workstation Audio, EE225A Term Project, University of California, Berkeley, 1993.

[4] Lamel, L.F. et alii, An Improved Endpoint Detector for Isolated Word Recognition, IEEE Trans. ASSP, Aug. 1981, pp. 777 - 785.

[5] Heitkaemper, P. and Walker, M., Adaptive Gain Control for Speech Quality Improvement and Echo Suppression, Proc. ISCAS 93, pp. 455 - 458.

[6] Tahernezhadi, M. et alii, Acoustic Echo Cancellation Using Subband Technique for Teleconferencing Applications, IEEE Globecom, 1994, pp. 243 - 247.

[7] Heitkaemper, P., Optimization of an Acoustic Echo Canceller Combined with Adaptive Gain Control, Proc. ICASSP 95, pp. 3047 - 3050.

[8] Mitra, S.K. and Kaiser, J.F., Handbook for Digital Signal Processing, John Wiley, 1993.

[9] Gay, S.L., Fast Projection Algorithms with Application to Voice Echo Cancellation, Diss. 1994, Rutgers University.

[10] Gay, S.L. and Mammone, R.J., Fast Converging Subband Acoustic Echo Cancellation Using RAP on the WE®DSP16A, Proc. ICASSP 1990, pp. 1141 - 1144.

[11] Antweiler, C., Schiffer, A. and Doerbecker, M., Acoustic Echo Control with Variable Individual Step Size, 4th Int. Workshop on Acoust. Echo Control, Roros, Norway, June 1995, pp. 15 - 18.

[12] Makino, S., Kaneda, Y. and Koizumi, N., Exponentially Weighted Stepsize NLMS Adaptive Filter Based on the Statistics of a Room Impulse Response, IEEE Trans. ASSP,

[13] Yamamoto, S. and Kitayama, S., An Adaptive Echo Canceller with Variable Step Gain Method, Trans. IECE Japan, Vol. E65, No. 1, Jan. 82, pp. 1 - 8.

[14] Yasukawa, H. and Shimada, S., An Acoustic Echo Canceller Using Subband Sampling and Decorrelation Methods, IEEE Trans. Signal Proc., Vol. 41, No. 2, Feb. 93, pp. 926 - 930.

[15] Barnwell, T., Nayebi, K. and Richardson, C.H., Speech Coding - A Computer Laboratory Textbook, John Wiley, 1996.

[16] Antweiler, C., , Orthogonalisierende Algorithmen fuer die digitale Kompensation akustischer Echos, Dissertation 1995, Univ. Aachen, Inst. fuer Nachrichtengeraete und Datenverarbeitung.

[17] Poltmann, R. D., A Fast Method for the NLMS Algorithm with Time-Variant Decorrelation Filters, European Trans. on Telecomm., vol. 6, No. 2, March - April 1995, pp 203 - 206.

[18] Blahut, R. E., Fast Algorithms for Digital Signal Processing, Addison-Wesley, 1985.

[19] Antweiler, C. and Antweiler, M., System Identification with Perfect Sequences Based on the NLMS Algorithm, Archiv. fuer Elektronik und Uebertragungstechnik, Vol. 49, No. 3, 1995, pp. 129 - 134.

[20] Lueke, H.D., Binare Folgen und Arrays mit optimalen ungeraden Autokorrelationsfunktionen, Frequenz 48 (1994), pp 213 - 220.

[21] Klisch, R., Implementation von Algorithmen fuer ein digitales Freisprechtelefon auf einem Signalprozessorsystem, Diploma Thesis at the Institut fuer Nachrichtengeraete und Datenverarbeitung, TH Aachen.

[22] Narayan, S.S., Peterson, A.M. and Narasimha, M.J., Transform Domain LMS Algorithm, IEEE Trans. ASSP, ASSP-31, No. 3, June 1983, pp. 609 - 615.

[23] Beaufays, F., Transform-Domain Adaptive Filters: An Analytical Approach, IEEE Trans. Sign. Proc., Vol. 43, No. 2, Feb. 1995, pp. 422 - 431.

[24] Farhang-Boroujeny, B. and Gazor, S., Generalized Sliding FFT and Its Applications to Implementation of Block LMS Adaptive Filters, IEEE Trans. Sign. Proc., Vol. 42, No. 3, March 1994, pp. 532 - 538.

[25] Ramamurthy, N. and Swamy, M.N.S., On the Computation of Running Discrete Cosine and Sine Transforms, IEEE Trans. Sign. Proc., Vol. 40, No. 6, June 1992, pp. 1430 - 1437.

[26] Shynk, J.J. and Widrow, B., Bandpass Adaptive Pole-Zero Filtering, Proc. ICASSP 86, pp. 2107 - 2110.

[27] Clark, G.A., Soderstrand, M.A. and Johnson, T.G., Transform Domain Adaptive Filtering Using a Recursive DFT, Proc. ISCAS 85, pp. 1113 - 1116.

[28] Petraglia, M.R. and Mitra, S.K., Adaptive FIR Filter Structure Based on the Generalized Subband Decomposition of FIR Filters, IEEE Trans. CAS-2, vol. 40, No. 6, June 1993, pp. 354 - 362.

[29] Haykin, S., Adaptive Filter Theory, Prentice Hall, 1991.

[30] Spring, A., Unger, H. and Wardenga, W., Identifikation in Multirate-Systemen mittels mehrkanaliger schneller Transversalfilter, Frequenz, Vol. 47, No. 7 - 8, 1993, pp. 201- 208.

[31] Carayannis, G., Manolakis, D.G. and Kalouptsidis, N., A Fast Sequential Algorithm for Least-Squares Filtering and Prediction, IEEE Trans. ASSP, vol. ASSP-31, No. 6, Dec. 1983, pp. 1394 -1402.

[32] Cioffi, J.M. and Kailath, T., Fast Recursive Least Squares Transversal Filters for Adaptive Filtering, IEEE Trans. ASSP, vol. ASSP-32, No. 2, April 1984, pp. 304 - 337.

[33] Haetty, B., Recursive Least Squares Algorithms Using Multirate Systems for Cancellation of Acoustical Echoes, ICASSP 1990, pp. 1145 - 1148.

[34] Moustakides, G.V. and Theodoridis, S., Fast Newton Transversal Filters - A New Class of Adaptive Estimation Algorithms, IEEE Trans. Sign. Proc., Vol. 39, No. 10, Oct. 91, pp. 2184 - 2193.

[35] Petillon, T., Gilloire, A. and Theodoridis, S., The Fast Newton Transversal Filter: An Efficient Scheme for Acoustic Echo Cancellation in Mobile Radio, IEEE Trans. Sign. Proc., Vol. 42, No. 3, March 1994, pp. 509 - 518.

[36] Morgan, D. R. and Kratzer, S. G., On a Class of Computationally Efficient, Rapidly Converging, Generalized NLMS Algorithms, IEEE Sign. Proc. Lett., Vol. 3, No. 8, Aug. 1996, p. 245- 247.

[37] Gay, S. L. and Tavathia, S., The Fast Affine Projection Algorithm, Proc. ICASSP 95, pp. 3023 - 3026.

[38] Tanaka, M. et alii, Fast Projection Algorithm and its Step Size Control, Proc. ICASSP 95, pp. 945 - 948.

[39] Gay, S. L., Dynamically Regularized Fast RLS with Application to Echo Cancellation, Proc. ICASSP 96, pp. 957 - 960.

[40] Makino, S., SSB Subband Echo Canceller Using Low Order Projection Algorithm, Proc. ICASSP 96, pp. 945 - 948.

[41] Ozeki, K. and Umeda, T., An Adaptive Filtering Algorithm Using an Orthogonal Projection to an Affine Subspace and its Properties, Trans. IECE Japan, vol. J67-A, Feb. 1984, pp. 126 - 132 (in Japanese).

[42] Yasukawa, H., Shimada, S. and Furukawa, I., Acoustic Echo Cancellation with High Speech Quality, ICASSP 87, pp. 2125 - 2128.

[43] Yasukawa, H., Furukawa, I. and Ishiyama, Y., Acoustic Echo Control for High Quality Audio Teleconferencing., ICASSP 1989, pp. 2041 - 2044.

[44] Hart, J., Multirate Sub-Band Structures with Applications to Adaptive Acoustic Echo Cancellation, PhD-Thesis, Imperial College of Science and Technology, Univ. London, 1996.

[45] Vaidyanathan, P.P., Multirate Systems And Filter Banks, Prentice Hall, 1993.

[46] Morgan, D.R., Slow Asymptotic Convergence of LMS Acoustic Echo Cancelers,IEEE Trans. ASSP, Vol. 3, No. 2, March 1995, pp. 126 - 136.

[47] Gilloire, A. and Vetterli, M., Adaptive Filtering in Subbands with Critical Sampling: Analysis, Experiments, and Applications to Acoustic Echo Cancellation, IEEE TRans. Signal Proc., Vol. 40, No. 8, Aug. 1992, pp. 1862 - 1875.

[48] Kellermann, W., Analysis and Design of Mltirate Systems for Cancellation of Acoustical Echoes, Proc. ICASSP 88, pp. 2570 - 2573

[49] Vaidyanathan, P.P., Mitra, S.K. and Neuvo, Y., A New Approach to the Realization of Low-Sensitivity IIR Digital Filters, IEEE Trans. ASSP-34, No. 2, April 1986, pp. 350 - 361.

[50] Mitra, S.K. and Hirano, K., Digital All-Pass Networks, IEEE Trans. CAS-21, No. 5, Sept. 74, pp. 688 - 700.

[51] Tanrikulu, O. et alii, Finite-Precision Design and Implementation of All-Pass Polyphase Networks for Echo Cancellation in Sub-Bands, Proc. ICASSP 95, pp. 3039 - 3042.

[52] Diniz, P.S.R., Cousseau, J.E. and Antoniou, A., Fats Parallel Realization of IIR Adaptive Filters, IEEE Trans. CAS-2, Vol. 41, No. 8, Aug. 94, pp. 561 - 567.

[53] Nayeri, M. and Jenkins, W.K., Alternate Realizations to Adaptive IIR Filters and Properties of Their Performance Surfaces, IEEE TRans. CAS, Vol. 36, No. 4, April 89, pp. 485 - 496.

[54] Shynk, J.J., Adaptive IIR-Filtering, IEEE ASSP Magazine, April 89, p. 4 - 21.

[55] Regalia, P.A., Stable and Efficient Lattice Algorithms for Adaptive IIR-Filtering, IEEE Trans. Signal Proc., Vol. 40, No. 2, Feb. 92, pp. 375 - 388.

[56] Davidson, G. and Falconer, D.D., Reduced Complexity Echo Cancellation Using Orthonormal Functions, IEEE Trans. CAS, Vol. 38, No. 1, Jan. 91, pp. 20 - 28.

[57] Harteneck, M. et alii, An Effective Approach to Adaptive IIR Filtering, Proc. ICASSP 96.

[58] Chao, J., Kawabe, S. and Tsuji, S., A New IIR Adaptive Echo Canceler: GIVE, IEEE J. Sel. Areas Comm., Vol. 12, No. 9, Dec. 94, pp. 1530 - 1539.

[59] Tahernezhadi, M. and Liu, L., Real-Time Implementation of an IIR Acoustic Echo Canceller on ADSP21020, Proc. ICASSP 95, pp. 2723 - 2726.

[60] Birkett, A.N. and Goubran, R.A., Acoustic Echo Cancellation Using NLMS-Neural Network Structures, Proc. ICASSP 95, pp. 3035 - 3038.

[61] Agazzi, O., Messerschmitt, D.G. and Hodges, D.A., Nonlinear Echo Cancellation of Data Signals, IEEE Trans. Comm., Vol. COM-30, No. 11, Nov. 82, pp. 2421 - 2433.

[62] Chua, R., The ICSI Audio Laboratory, Masters-Thesis, Univ. of California Berkeley, 1996.

[63] Tanrikulu, O., Residual Signal in Subband Acoustic Echo Cancellers, EUSPICO-96, Trieste.