CHAPTER 1

INTRODUCTION
1. INTRODUCTION

1.1 Introduction:

Discrete-time (or digital) filters are everywhere in today's signal processing applications. Filters are used to achieve desired spectral characteristics of a signal, to reject unwanted signals, like noise or interferers, to reduce the bit rate in signal transmission, etc. The notion of making filters adaptive, i.e., to alter parameters (coefficients) of a filter according to some algorithm, tackles the problems that it might not be known in advance, e.g., the characteristics of the signal, or of the unwanted signal, or of a systems influence on the signal that would like to compensate. Adaptive filters can adjust to unknown environment, and even track signal or system characteristics varying over time.

In many applications, time-variable filters are required whose characteristics can be varied with time. Filters of this type can be obtained by using multipliers with time variable coefficients. A time-variable filter that incorporates some adaptation mechanism by which the multiplier coefficients can be adjusted on line so as to optimize some performance criterion is said to be an adaptive filter [1, 2]. The adaptation mechanism usually incorporates an optimization algorithm that evaluates the instantaneous values of the multiplier coefficients such that some norm of an error function is minimized. Adaptive filters constitutes an important part of statistical signal processing. Whenever there is a requirement to process signals that result from operation in an environment of unknown statistics or one that is inherently non-stationary, the use of an adaptive filter offers a highly attractive solution to the problem as it provides a significant improvement in performance over the use of a fixed filter designed by conventional methods. Furthermore, the use of adaptive filters provides new signal-processing
capabilities that would not be possible otherwise. Thus it is find that adaptive filters have been successfully applied in such diverse fields as communications, control, radar, sonar, seismology, and biomedical engineering, among others.

Over the last two decades, adaptive filtering has been of considerable interest for many applications such as acoustic echo cancellation [3], noise reduction, equalization, and beam forming [4]. Often the adaptive system has to "model" a long duration impulse response, as in the case of, for example, the identification of room acoustics [5], [6]. Hence with the high number of adaptive filter weights required, popular adaptive algorithms based on least mean squares and least squares techniques [7] become very computationally complex and exhibit a slow convergence. In the conventional and normalized LMS algorithms, the tap weights (free parameters) of a finite-duration impulse response (FIR) filter are adapted in the time domain. Recognizing that the Fourier transform maps time-domain signals into the frequency domain and that the inverse Fourier transform provides the inverse mapping that takes us back into the time domain, it is equally feasible to perform the adaptation of filter parameters in the frequency domain.

Multirate schemes such as the subband adaptive filter have been a topic of interest for many years now [8]. They are employed to identify high-order FIR systems and are a promising alternative to the classical LMS algorithm. For this kind of application the LMS adaptive filter is less attractive as it has a larger complexity and its convergence behavior is generally worse.

Subband adaptive filter, which is different from a self-orthogonalizing adaptive filter. In a subband adaptive filter, filters with high stop-band rejection are used to band-partition the input signal and hence possibly provide an improvement in convergence. Moreover, by decimating the subband signals (i.e., down-sampling to a
lower rate), it is possible to achieve a significant reduction in computational complexity. Subband adaptive filters achieve computational efficient by decimating the signals before performing the adaptive process. Decimation refers to the process of digitally converting the sampling rate of a signal of interest from a given rate to a lower rate. The use of this approach makes it possible to implement an adaptive FIR filter of long memory and that is computationally efficient. In this endeavor, the task of designing a single long filter is replaced by one of designing a bank of smaller filters that operate in parallel at a lower rate.

It is often claimed that subband processing leads to a considerable complexity reduction with respect to the fullband approach, offering a complexity gain equal to the number of subbands in the limiting case where the subbands are maximally, i.e. critically downsampled. Furthermore, better performance is expected owing to the fact that after appropriate subsampling each subband signal will have a better spectrum than the fullband signal.

Digital filter bank techniques have been practiced by engineers for more than a decade and a half. This discipline finds applications in speech and image compression, the digital audio industry, statistical and adaptive signal processing, numerical solution of differential equations, and in many other fields. It also fits naturally with certain special classes of time-frequency representations such as the short-time Fourier transform and the wavelet transform, which are useful in analyzing the time-varying nature of signal spectra.
1.2 Adaptive Filtering:

Least Mean Square (LMS) algorithm, introduced by Widrow and Hoff in 1960 [9], is one of the most well-known control algorithms for adaptive filters.

The LMS algorithm is in short summarized using the equations

\[ y(n) = f'(n)x(n) \]  
\[ e(n) = d(n) - y(n) \]  
\[ f(n+1) = f(n) + \mu x^*(n)e(n) \]

where \( f(n) = [f_0(n), \ldots, f_{L_f-1}(n)]^T \) is a vector with the filter coefficients at time instant \( n \). The number of filter coefficients is denoted \( L_f \), and \((\cdot)^*\) denotes complex conjugate. The vector \( x(n) = [x(n), \ldots, x(n - L_f + 1)]^T \) is an input signal buffer at time instant \( n \), which holds the most recent \( L_f \) input samples of \( x(n) \).

The LMS adaptive filter is an adaptive solution to the FIR Wiener filter design problem. The FIR Wiener filter is an optimal filter, which minimizes the Mean-Square Error

\[ J = E\{|e(n)|^2\} \]

where \( E\{.\} \) is the expectation operator. The gradient \( \nabla = 2\partial/\partial f^* \) of the Minimum Mean-Square Error (MMSE) with respect to the filter coefficients given by

\[ \nabla E\{|e(n)|^2\} = E\{e(n)\nabla e^*(n)\} = -2E\{e(n)x^*(n)\} \]
Inserting Eqs. (1.1) and (1.2) and setting the gradient to zero yields the system of equations

\[-2E\{d(n)x^*(n)\} + 2E\{x^*(n)x^T(n)\}f = -2r_{dx} + 2R_{xx}f = 0 \quad (1.6)\]

where matrix \( R_{xx} = E\{x^*(n)x^T(n)\} \) is the input signal autocorrelation matrix, vector \( r_{dx} = E\{d(n)x^*(n)\} \) is a desired signal - input signal cross correlation vector. Solving Eq. (1.6) leads to the Wiener solution

\[f_{\text{Wiener}} = R_{xx}^{-1}r_{dx} \quad (1.7)\]

The adaptive LMS algorithm is obtained by using an instantaneous estimate of the gradient. Substituting Eqs. (1.1) and (1.2) into Eq. (1.3) gives

\[f(n + 1) = f(n) + \mu[d(n) - f^T(n)x(n)]x^*(n) \quad (1.8)\]

Using the ensemble average \( \bar{f}(n) = E\{f(n)\} \), and assuming that the input signal samples in \( x(n) \) and LMS coefficient vector \( f(n) \) are statistically independent yields

\[\bar{f}(n + 1) = (I - \mu R_{xx})\bar{f}(n) + \mu r_{dx} \quad (1.9)\]

where \( I \) is the unit matrix. A coefficient error vector is defined as

\[\Delta f(n) = \bar{f}(n) - f_{\text{Wiener}} \quad (1.10)\]

Inserting Eq. (1.5) leads to the recursion

\[\Delta \bar{f}(n) = (I - \mu R_{xx})\Delta \bar{f}(n - 1) \quad (1.11)\]
Since the autocorrelation matrix is hermitian, i.e. $R_{xx} = R_{xx}^H$, the matrix may be factorized using the eigenvalue decomposition $R_{xx} = VAV^H$ (the spectral theorem) with orthogonal eigenvector matrix $V$ and diagonal matrix $A$ with real eigenvalues $\lambda_0, \ldots, \lambda_{L-1}$ on the main diagonal [10]. Using the eigenvalue decomposition and the fact that $V$ is unitary, i.e. $VV^H = I$, yields

$$
\Delta \tilde{f}(n) = V(I - \mu A)V^H \Delta \tilde{f}(n-1)
$$  \hspace{1cm} (1.12)

A modal coefficient error vector is introduced as $\Delta \tilde{f}(n) = V^H \Delta \tilde{f}(n)$ and evolves with time according to

$$
\Delta \tilde{f}(n) = (I - \mu A) \Delta \tilde{f}(n-1)
$$  \hspace{1cm} (1.13)

With an initial modal coefficient error vector $\Delta \tilde{f}(0)$, Eq. (1.13) may be rewritten as

$$
\Delta \tilde{f}(n) = (I - \mu A)^{\star} \Delta \tilde{f}(0)
$$  \hspace{1cm} (1.14)

Since $(I - \mu A)$ is a diagonal matrix, the elements of $\Delta \tilde{f}(n)$ may be expressed as

$$
\Delta \tilde{f}_i(n) = (I - \mu \lambda_i)^{\star} \Delta \tilde{f}_i(0), \quad i = 0, \ldots, L_f - 1
$$  \hspace{1cm} (1.15)

which are referred to as the natural modes of the adaptive filter [36]. In order for $\tilde{f}(n)$ to converge to $\tilde{f}_{\text{wener}}$, $\Delta \tilde{f}(n)$ should converge to zero and therefore $\Delta \tilde{f}(n)$ should converge to zero. This will occur if and only if
\[ |1 - \mu \lambda_i| < 1, \quad \forall i \] (1.16)

The decay for each mode is dependent on the magnitude of \( |1 - \mu \lambda_i| \) and is thus dependent on both \( \mu \) and \( \lambda_i \). Therefore the step-size is restricted by

\[ 0 < \mu < \frac{2}{\lambda_{\max}} \] (1.17)

Eq. (1.11) shows that the convergence of the filter coefficients in the mean, \( \bar{f}(n) \), is dependent on the correlation matrix. If \( x(n) \) is white noise, the correlation matrix is diagonal and all eigenvalues are the same. In this case, the modes will converge in a uniform manner. However, when the input is correlated and the eigenvalues are widely varying, some modes will converge more quickly than others. The choice of the step size is limited by the fastest converging mode. Even though the step size might be large and modes converge fast, there might be modes converging very slowly and thereby causing the filter to converge slowly. So in order to increase the convergence rate, the modes need to be uncoupled so that each filter coefficient controls one and only one mode. Furthermore normalization with the magnitude of the eigenvalues is often required so that all modes converge uniformly.

The most trivial way to uncouple the modes, while simultaneously normalizing with the magnitude of the eigenvalues of the modes, is to insert the inverse correlation matrix, \( R_n^{-1} \), into the coefficient update equation. This yields the Self-Orthogonalizing LMS (SO-LMS). The coefficient update equation for the SO-LMS is given by
\[ f(n + 1) = f(n) + \mu R_\mu^{-1} x^*(n)e(n) \] (1.18)

Since \( R_\mu^{-1}R_\mu = I \), the coefficient error vector \( \Delta \tilde{f}(n) \) is given by

\[ \Delta \tilde{f}(n) = (1 - \mu) \Delta \tilde{f}(n - 1) \] (1.19)

In this case, the convergence speed is not dependent on the input signal characteristics, and all modes decay uniformly to zero, only dependent on the step size \( \mu \). The SO-LMS is not very useful in practical situations since it requires knowledge of the input correlation matrix. The approach is however of theoretical importance since it forms the basic concept of some very powerful orthogonalizing adaptive filters with approximative orthogonalization. Recursive Least Squares (RLS) algorithm is an example of such an algorithm, which estimates the input correlation matrix-inverse recursively and uses the estimate to orthogonalize the input data approximately, and thus increases the converge speed.

Many different adaptive filtering techniques are introduced during the years. The Normalized LMS (NLMS) has made the LMS adaptive filter less sensitive to non-stationary input. The Affine Projection Algorithm (APA) is based on the same idea as the RLS, but the APA is a generalization between the NLMS and the RLS, providing a tradeoff between increased convergence speed and low computational complexity. Gradient Adaptive Lattice prediction filters are used to de-correlate the input samples, and in this way improve the convergence speed. In transform domain adaptive filters, orthogonalizing transforms are exploited to improve the convergence speed. Frequency domain adaptive filtering, which was initially proposed in [12], increased the convergence speed but also enabled implementations with low computational costs.
In the next sections, an introduction is given to fast implementations using fast Fourier transform and transform domain adaptive filtering for increased convergence speed. Finally an introduction to subband domain adaptive filtering is given, which may be regarded as a generalized form of adaptive filtering. An excellent overview of all these techniques is found in [13].

1.2.1 Algorithms with Reduced Computational Complexity:

The Frequency Domain Block LMS, or Fast Block LMS, was initially proposed by Ferrara [14]. It was found that by performing the signal filtering and the adaptation of the filter coefficients in the frequency domain, the computational complexity of large filters could be reduced significantly. Frequency domain adaptive filters use block updating strategies where the Fast Fourier Transforms (FFT) is used to perform the filtering and coefficient adaption [15]. Other examples of frequency domain adaptive filters are the adaptive frequency sampling filter [16], where the filter coefficients are sampling points of the frequency response [17].

The fast Block LMS is based on the Block implementation of the LMS adaptive filter. In the Block-LMS (B-LMS) algorithm [18], the filter output is calculated block-wise and the filter coefficients are adapted block-wise instead of sample-wise. Using matrix notations of block-wise filtering and block-wise correlation, the B-LMS is summarized by the following equations

\[ y(n) = X(n)f(n) \]  
\[ e(n) = d(n) - y(n) \]
\[ f(n + L_f) = f(n) + \mu X^H(n) e(n) \]  \hspace{1cm} (1.22)

which are performed at time instances \( n = kL_f \), where \( k = 0, 1, 2, 3, \ldots \). The filter length is denoted with \( L_f \) and parameter \( k \) is the block index. The output signal \( y(n) \) is also calculated block-wise, defining \( y(n) = [y(n), \ldots, y(n-B+1)]^T \), where \( B \) denotes the block length. The vectors \( d(n) \) and \( e(n) \) are the corresponding desired signal and error signal blocks. The input sample matrix \( X(n) \) is a convolution matrix according to

\[
X(n) = \begin{bmatrix}
x(n) & \cdots & x(n - L_f + 1) \\
x(n - B + 1) & \cdots & x(n - B - L_f + 2)
\end{bmatrix}
\]  \hspace{1cm} (1.23)

It can be shown that the B-LMS minimizes \( J_{B\text{-LMS}} = E\{e^H(n)e(n)\} \) instead of \( J_{LMS} = E\{|e(n)|^2\} \). For wide-sense stationary signals, \( J_{B\text{-LMS, min}} = B \cdot J_{LMS, min} \), and the Wiener solution, misadjustment and time constants are the same as for the LMS. An analysis of the performance of the Block LMS algorithm is given in [19].

The block convolution and block correlation operations for the filtering and gradient calculations, respectively, can be implemented efficiently using fast fourier transforms. The block-wise convolution in Eq. (1.20) with a filter length \( L_f \) and the block length set to \( B = L_f \) can be implemented efficiently using the Fast Fourier Transform as follows

\[ X(k) = \text{FFT}\{x(kL_f - 2L_f + 1), \ldots, x(n)\} \]  \hspace{1cm} (1.24)

\[ Y(k) = X(k) \odot F(k) \]  \hspace{1cm} (1.25)
\[ y(kL_f - L_f + 1, \ldots, y(kL_f)) \] = last \( L_f \) elements of \( \text{IFFT}\{Y(k)\} \) \hspace{1cm} (1.26)

which is performed for \( k = 0, 1, 2, 3, \ldots \). The parameter \( k \) denotes block index. The FFT-domain coefficient vector is

\[ F(k) = \text{FFT}\{f_0(kL_f), \ldots, f_{L_f - 1}(kL_f), 0, \ldots, 0\} \]

and \( \odot \) denotes element-wise multiplication. The method is known as the overlap-save FFT implementation of block-wise convolution. An illustration is given in Fig. 1.1.

The block-wise correlation operation for the calculation of the gradient estimate \( \hat{g}(n) = X^H(n)e(n) \) can be implemented efficiently by

\[ e(k) = \text{FFT}\{0, \ldots, 0, e(kL_f - L_f + 1), \ldots, e(kL_f)\} \]

\[ \hat{G}(k) = X^*(k) \odot e(k) \]

\[ [\hat{g}_0(kL_f), \ldots, \hat{g}_{L_f - 1}(kL_f)] = \text{first} \ L_f \ \text{elements of IFFT}\{\hat{G}(k)\} \]

which is performed for \( k = 0, 1, 2, 3, \ldots \).
The fast B-LMS is summarized by:

\[ X(k) = \text{FFT}\{x(kL_f - 2L_f + 1), \ldots, x(kL_f)\} \]  
(1.31)

\[ Y(k) = X(k) \oplus F(k) \]  
(1.32)

\[ [y(kL_f - L_f + 1), \ldots, y(kL_f)] = \text{last } L_f \text{ elements of } IFFT\{Y(k)\} \]  
(1.33)

\[ e(i) = d(i) - y(i), \text{ for } i = kL_f - L_f + 1, \ldots, n \]  
(1.34)

\[ e(k) = \text{FFT}\{0, \ldots, 0, e(kL_f - L_f + 1), \ldots, e(kL_f)\} \]  
(1.35)

\[ \hat{G}(k) = X^\ast(k) \oplus e(k) \]  
(1.36)

\[ [\hat{g}_0(kL_f), \ldots, \hat{g}_{L_f-1}(kL_f)] = \text{first } L_f \text{ elements of } IFFT\{\hat{G}(n)\} \]  
(1.37)

\[ F(n) = F(n - L_f) + \ldots \]  
\[ + \mu \text{FFT}\{\hat{g}_0(kL_f), \ldots, \hat{g}_{L_f-1}(kL_f), 0, \ldots, 0\} \]  
(1.38)

which is performed for \( k = 0, 1, 2, 3, \ldots \). Note that the gradient is calculated in the time domain, Eq. (1.37), and then truncated to \( L_f \) values before it is transformed and added to the coefficients, Eq. (1.38). This procedure is called the gradient constraint and ensures that the coefficient vector never exceeds \( L_f \) taps in the time domain, i.e. in order to avoid circular convolution.
An illustration is given in Fig. 1.2. Another implementation is the overlap-add version, where the “old” and “new” data blocks simply trade place. A thorough analysis of the converge behavior of the fast Block LMS algorithm is given in [20].

**Fig. 1.2: Overlap-Save FFT implementation of the B-LMS.**

A disadvantage of the Fast Block LMS is that of the block length and thus the lengths of the fast fourier transforms are connected to the length of the adaptive filter. This leads to very large transforms for very large filters and therefore a large block delay. This disadvantage is circumvented in the Partitioned Block Frequency Domain adaptive filters [21, 22]. A convergence analysis of such frequency domain adaptive filters is presented in [23].

### 1.2.2 Algorithms with Increased Convergence Speed:

In transform domain adaptive filters, transforms are used to improve the convergence rate by generating uncorrelated signals [24, 25, 26]. The class of LMS based adaptive filters using orthogonalizing transforms is known as Transform Domain LMS adaptive filters.
In these adaptive filters, the input sample vector $\mathbf{x}(n)$ is transformed into a vector $\mathbf{x}_T(n)$ by a linear transform $\mathbf{T}$, i.e. $\mathbf{x}_T(n) = \mathbf{T}\mathbf{x}(n)$, prior to further processing. Refer Fig. 1.3. In other words, the Transform Domain LMS is summarized by

$$y(n) = (\mathbf{T}\mathbf{x}(n))^T f_r(n) \tag{1.39}$$

$$e(n) = d(n) - y(n) \tag{1.40}$$

$$f_r(n+1) = f_r(n) + \mu(T\mathbf{x}(n))^* e(n) \tag{1.41}$$

The Wiener-Hopf equations for the transform domain LMS are given by

$$\mathbf{R}_r f_{r,\text{Wiener}} = r_f \tag{1.42}$$

with $\mathbf{R}_f = \mathbf{T}\mathbf{R}_x\mathbf{T}^H$ and $r_f = \mathbf{T}r_d$. Eq. (1.42) is rewritten as

$$\mathbf{T}\mathbf{R}_x\mathbf{T}^H f_{r,\text{Wiener}} = \mathbf{T}r_d \tag{1.43}$$

The coefficient error vector $\Delta f_r(n)$ for the transform domain LMS is given by
the eigenvalues of $R_T$. In the previous section it was already shown that the correlation matrix is preferably diagonal, to ensure uncoupled modes.

The transform $T$ is selected such that $R_T$ is exactly or approximates a diagonal matrix. The Karhunen-Loève (KLT) transform is the optimal transform: $T = V''$, where $V$ is the eigenvector matrix corresponding to $R_{xx}$, i.e.

$$R_T = TR_{xx}T = V''V\Lambda V''V = \Lambda$$

Approximations of the optimal transform are needed, since the KLT requires access to $R_{xx}$. The KLT is also known as Principal Component Analysis. Discrete Cosine Transform (DCT) and the Discrete Fourier Transform (DFT) are the other transforms that may be used in transform domain adaptive filters.

The effect of the transform is that the modes are exactly or approximately uncoupled. The convergence speed is still dependent on the eigenvalues of the input correlation matrix. Uniform convergence (all modes converge at the same speed) is obtained by inserting the inverse correlation matrix $R_T^{-1}$ into the coefficient update equation, similar to the SO-LMS approach. When using the KLT, the correlation matrix is $R_T = \Lambda$, i.e. the diagonal eigenvalue matrix corresponding to $R_{xx}$. Inserting $R_T^{-1} = \Lambda^{-1}$ in the coefficient update equation of the TD-LMS gives

$$\Delta f_T(n) = (1 - \mu R_T') \Delta f_T(n - 1)$$

and correspondingly

$$\Delta f_{T,i}(n) = (1 - \mu \lambda_i) \Delta f_{T,i}(n)$$

where $\lambda_i$ are the eigenvalues of $R_T$. In the previous section it was already shown that the correlation matrix is preferably diagonal, to ensure uncoupled modes.
Inserting \( x_T(n) = T \mathbf{x}(n) \), \( x_T(n) = [x_{T,0}(n), \ldots, x_{T,L-1}(n)]^T \), and rewriting the coefficient update for each coefficient, we obtain

\[
f_{T,i}(n+1) = f_{T,i}(n) + \mu \frac{1}{\hat{\lambda}_i} x_{T,i}(n)e(n), \quad k = 0, \ldots, L_i - 1
\]

which is the normalized TD-LMS. In a practical situation with an approximation of the KLT, the eigenvalues are estimated. This is for example obtained by exponential averaging of instantaneous estimates

\[
\hat{\Lambda}(n) = \hat{\mathbf{R}}_T(n) = \beta \hat{\mathbf{R}}_T(n-1) + (1 - \beta) \mathbf{x}_T(n) \mathbf{x}_T^H(n)
\]

Since \( \Lambda \) is diagonal, \( \hat{\Lambda} \) is approximately diagonal and the individual eigenvalue estimates may be expressed as

\[
\hat{\lambda}_i(n) = \beta \hat{\lambda}_i(n-1) + (1 - \beta) |x_{T,i}(n)|^2, \quad i = 0, \ldots, L_i - 1
\]

Hence \( \hat{\lambda}_i(n) \) are estimates of the power in the components \( x_{T,i}(n) \). In summary, the normalized transform domain LMS is given by

\[
x_T(n) = T \mathbf{x}(n)
\]

\[
y(n) = x_T^H(n) \mathbf{f}(n)
\]

\[
e(n) = d(n) - y(n)
\]

\[
\hat{\lambda}_i(n) = \beta \hat{\lambda}_i(n-1) + (1 - \beta) |X_i(n)|^2, \quad i = 0, \ldots, L_i - 1
\]
A generalized method of transform domain LMS is the Filter Bank Adaptive Filter proposed in [28].

### 1.2.3 Subband Adaptive Filtering:

Frequency domain and transform domain adaptive filtering may be regarded as special cases of subband adaptive filtering. As a new concept, subband adaptive filtering was first introduced in the second half of the 1980s [29, 30, 31, 32]. Since the introduction of the concept of subband adaptive filtering, several new structures are proposed using general multirate structures and filter banks. A new subband adaptive filter using a weighted mean square criterion was proposed in [33]. A similar criterion was introduced in [34] where a fullband adaptive filter is adapted in the subband domain using the a polyphase decomposition. A new subband adaptive filtering structure with critical sampling was introduced in [35]. Adaptive noise cancellation [36], acoustic echo cancellation [37], microphone arrays [38] and channel equalization [39, 40] are a few applications where Subband adaptive filtering is successfully applied.

The main idea in subband adaptive filtering is to split a high order adaptive filtering problem into a number of low order adaptive filters. The general concept of subband adaptive filtering may involve a number of properties, such as improved convergence speed and low computational costs. Normally a combination of both is preferred. Another significant property is low delay, which for example is achieved at the expense of computational complexity.
Different configurations of subband adaptive filters exist. In some configurations, the desired signal is decomposed into subband signals. In [41], this is referred to as the open loop configuration, with individual adaptive filters operating in the subbands. In the open loop configuration, the adaptive filters are controlled using individual (local) subband error signals, refer Fig. 1.4. The open loop approach is also used in [31, 42], where a generalized adaptive filtering approach is presented with adaptive cross-filters operating between the subbands. When the open loop approach is used in a system identification scenario, it is shown that cross-filters are necessary to identify the unknown system correctly. The approach with individual filters in the subbands, as depicted in Fig. 1.4, has limited performance in terms of fullband Mean Square Error. This is essentially the same in a frequency domain adaptive filter, with an individual frequency domain tap update, where circular correlation limits the performance when the gradient constraint is not imposed on the gradient.

In another configuration for subband adaptive filtering, an error signal is calculated in fullband, and decomposed into subband error signals which are used to control the adaptive filters, refer Fig. 1.5. In [41], this is referred to as the closed loop configuration. In this approach, individual adaptive filters operating in the subbands are
sufficient for high performance in terms of fullband MSE. However, the converge speed may be lower because filter banks impose a signal delay in the control loop.

In a subband adaptive filter, the number of subbands \( M \), the decimation factor \( D \) and the analysis and synthesis filters \( H_m(z) \) and \( G_m(z) \) are filter bank parameters respectively, which influence the performance of the adaptive filter in terms of fullband MSE and convergence speed.

An important issue in subband adaptive filtering is that multirate filter banks introduce aliasing and imaging distortion, caused by multirate building blocks. They also introduce system and/or block delay. Too much delay is an undesired effect in subband adaptive filtering. For example, in acoustic echo cancellation algorithms the maximum tolerable delay is 2 ms for stationary telephones and 39 ms for mobile telephones [37].
Delayless subband adaptive filters are introduced to deal with the drawback of delay in subband adaptive filters, at the expense of extra computational complexity [41], refer Fig. 1.6. These subband adaptive filters introduce no delay in the signal path; however delay may still be present in the algorithm path. Since the filtering is done in time domain, rather than subband domain, there is no aliasing or imaging distortion present in the output signal. However, in-band aliasing effects may still impair the performance of the adaptive algorithm. The algorithm uses a mapping of the adaptive subband filter coefficients to the fullband domain by a coefficient transform. An application specific integrated circuit (ASIC) implementation is presented in [43]. After the initial paper by Morgan and Thi [41], several new filter bank and transform configurations are proposed. Hirayama et. al. proposed a tree structured uniform filter bank scheme with Hadamard transforms for the subband-to-fullband transform [44]. In Huo et. al., several improved schemes for the subband-to-fullband transformation are presented [45]. Besides improvements of the original approach, new delayless subband adaptive filters are developed [46, 47, 48].
1.3 Multirate Digital Filter Banks:

Filter banks play an important role in several of signal processing applications, such as tele-transmission, speech processing, audio/video signal compression, signal detection and spectral analysis. The steadily growing interest in filter banks, is mainly due to the wide range of applications where filter banks successfully have been applied.

The first notion of analysis filter banks is systems where short-time signal segments are transformed into a set of transform domain coefficients. One of the most important transforms is the Discrete Fourier Transform. The DFT, and its fast implementation the Fast Fourier Transform (FFT), is an important tool in spectral analysis [49]. By performing the transform at pre-defined intervals, a time dependence is introduced in the transform domain coefficients and they may be regarded as decimated signals. This may also be regarded as the discrete-time version of the Short-Time Fourier Transform (STFT). The magnitude of the STFT is sometimes referred to as the Spectrogram and is an important tool in time-frequency analysis of signals. Analysis filter banks are successfully applied in analysis and estimation of signals [50, 51].

The first and one of the most important applications of filter banks is subband coding of speech signals [52, 53]. In subband coding, signals are decomposed into subband signals using filter banks and then coding techniques are used to convert the subband signals into bits. Major advantages of subband coding, is that good compression is achieved, where quantization is performed using perceptual criteria, which means that errors are distributed in a way such that they are not perceivable [54, 55]. Furthermore, an additional important feature is that operations on the signals in the subband domain are performed at a lower computational cost. Apart from speech signals, subband coding is
successfully introduced in compression of music [56, 57], images [58] and video, which has resulted in a number of industrial standards, such as JPEG for images [59, 60] and MPEG for audio [61] and video [62, 63].

Filter banks are successfully applied in tele-transmission. Transmultiplexer filter banks convert time-division multiplexed signals (TDM) into frequency division multiplexed (FDM) signals. In a frequency division multiplexed signal, individual signals occupy different frequency bands of the spectrum. A transmultiplexer filter bank is a multirate system consisting of a synthesis filter bank connected to an analysis filter bank, i.e. a multirate multiple-input multiple-output (MIMO) system. In [64], it is shown that if the order of analysis and synthesis filter banks is interchanged, the dual filter bank is formed. A perfect reconstruction filter bank is the dual system of a transmultiplexer filter bank that is free of crosstalk. Crosstalk is the leakage of information among the individual signals due to the multirate structure in the same way that aliasing appears across the subbands of a analysis-synthesis filter bank. Transmultiplexer filter banks have for example been applied in digital audio broadcasting (DAB) and asymmetrical digital subscriber lines (ADSL). In the case of wireless communications, this is known as orthogonal frequency division multiplexing (OFDM) [65, 66], and in the case of line transmission, this is known as discrete multitone (DMT).

Multirate filter banks are special cases of multirate systems [67, 68]. In multirate systems, digital signals are processed using more than one sampling rate. In multirate systems, the sampling rate is kept as low as possible, which leads to more efficient processing. Multirate techniques are for example used in filters with stringent requirements and low complexity [69]. Sampling of analog signals and lowering the sampling rate of digital signals by decimation introduces an undesired effect, known as
aliasing. In the design of filter banks, aliasing may be minimized or cancelled. Many design techniques for different filter banks are proposed during the years. Key developments in this field include two-channel Quadrature Mirror Filter (QMF) Banks [70], the polyphase representation [71, 72], and the extension from two to M channels [72]. It is shown in [70] that aliasing may be fully cancelled in the two-channel QMF synthesis filter bank. If in conjunction, the transfer function of the analysis-synthesis filter bank is a pure delay, i.e. linear phase and constant magnitude, the system is said to have the perfect reconstruction (PR) property. Approximate aliasing cancellation has first been introduced for M-channel filter banks [73] and the general theory for PR is fully developed subsequently [74]. The polyphase representation and paraunitary polyphase matrices have lead to a considerable simplification of the theory [64, 75]. The theory of aliasing cancellation is the main focus of filter bank theory and design techniques. Optimization methods have later been developed to minimize remaining distortions. Many of these optimization methods have their origin in classical filter design [67, 68].

1.3.1 Uniform Filter Banks:

A uniform M-channel analysis filter bank is a structure consisting of analysis filters, $H_m(z), m = 0, \ldots, M - 1$, and decimators with decimation factor $D$. The task of the analysis filter bank is to transform an input signal $X(z)$ to a set of $M$ subband signals $X_m(z)$, which are sampled at a lower rate, refer Fig. 1.7.
A corresponding $M$-channel synthesis filter bank consists of synthesis filters $G_m(z)$, and interpolators with interpolation rate equal to $D$. The task of the synthesis filter bank is to transform $M$ subband signals $Y_m(z)$ to a full band signal $Y(z)$, which is sampled at the original higher rate, refer Fig. 1.7.

Uniform filter banks are filter banks where the sampling rate, and thus the decimation factor, in all the channels is the same. Different structures of $M$-channel filter banks exist, for example the tree structure [76] and the parallel structure [77]. The parallel, or direct form structure is illustrated in Fig. 1.7. A special uniform filter bank is the modulated filter bank where the individual filters in the filter bank are modulated from prototype filters, $H(z)$ and $G(z)$. Examples of such filter banks are the uniform DFT filter bank and the cosine modulated filter bank. Both filter banks may be efficiently implemented using the polyphase implementation employing fast transforms, the discrete fourier transform (DFT) and discrete cosine transform (DCT), respectively, and the noble identity properties of decimation and interpolation [67].
The subband signals from the analysis filter bank are expressed by [67]

\[ X_n(z) = \frac{1}{D} \sum_{d=0}^{D-1} H_n(z^{-1}W_n^d)X(z^{-1}W_n^d) \]  

(1.56)

where \( W_D = e^{-j2\pi D} \). By examining Eq. (1.56), the effect of decimation on the signal spectrum is explained as follows. The spectrum of the filtered subband signal is first expanded. Then, different shifted spectral terms are added and constitute the decimated signal spectrum. This is illustrated in Fig. 1.8 for a critically sampled filter bank with \( M = 4 \) subbands and decimation factor \( D = 4 \). Some spectral parts among these terms are undesired in-band aliasing. The task of the stopband in the analysis filters is to suppress these spectral parts, so that in-band aliasing is minimized.

![Fig. 1.8: Magnitude of in-band aliasing transfer functions \( H_n(z^{-1}W_n^d) \) for a critically sampled \((D = M)\) modulated DFT filter bank with \( M = 4 \) subbands.](image-url)
An input-output expression for the system in Fig. 1.7 is given by

\[ Y(z) = \frac{1}{D} \sum_{m=0}^{M-1} \sum_{d=0}^{L-1} H_m(zW_d^m)G_m(z)X(zW_d^m) \]  \hspace{1cm} (1.57)

Hence, the output signal \( Y(z) \) consists of \( MD \) terms where the input signal \( X(z) \) is present in a modulated and filtered form. The terms where \( X(z) \) is present in modulated form (for \( d > 0 \)) are the undesired aliasing terms and the terms where \( X(z) \) is present in un-modulated form are the desired terms. Eq. (1.57) is rearranged in desired and undesired terms as

\[ Y(z) = \frac{1}{D} \sum_{m=0}^{M-1} \left[ \sum_{d=0}^{L-1} H_m(zW_d^m)G_m(z)X(zW_d^m) \right] + \sum_{d=1}^{L-1} H_m(zW_d^m)G_m(z)X(zW_d^m) \]  \hspace{1cm} (1.58)

Eq. (1.58) is simplified as

\[ Y(z) = T(z)X(z) + \sum_{m=0}^{M-1} \sum_{d=1}^{L-1} A_{m,d}(z)X(zW_d^m) \]  \hspace{1cm} (1.59)

where \( T(z) \) represents the overall transfer function for the terms of interest, and \( A_{m,d}(z) \) represents the transfer functions which give rise to the undesired residual aliasing terms in the output signal \( Y(z) \). Refer Fig. 1.9. The filter bank design problem comprises a specification on the behavior of the transfer functions \( T(z) \) and \( A_{m,d}(z) \).
Fig. 1.9: Magnitude of residual aliasing transfer functions $A_{m_d}(\omega)$ for a critically sampled (D = M) modulated DFT filter bank with $M = 4$ subbands.

1.3.2 Nonuniform Filter Banks:

Nonuniform filter banks are multirate filter banks with nonuniform frequency resolution where the decimation factors are not necessarily equal, unlike the uniform filter banks. These filter banks have gained much attention since they were introduced in systems for coding of images [78] and speech and audio signals [70]. The use of nonuniform filter banks in speech processing is motivated by the properties of the human auditory system. By appropriate design it is possible to get a cochlear model [79], i.e. a model of the frequency selective organs of the inner ear. For this same reason, nonuniform filter banks are suggested for speech enhancement and they are also applied in speech recognition. Nonuniform filter banks have also been proposed for subband adaptive filtering, e.g. in spectral subtraction for speech enhancement [80], and beamforming for subband microphone arrays [38]. A special type of nonuniform filter banks are octave filter banks [77]. Closely related to these are dyadic wavelets [81].
Octave filter banks are constructed using a tree structure, where the low-frequency channel is successively split by two-channel filter banks. Filter banks with nonuniform frequency resolution can also be obtained by applying frequency transformation to a uniform filter bank. They utilize a lossless frequency transformation similar to a bilinear transform [82]. Frequency transformed filter banks have previously been presented [83, 84]. These filter banks are known to approximate the Bark frequency scale (or critical band scale) accurately [85].

1.4 Design by Optimization:

1.4.1 Digital Filter Design:

Many efficient methods exist for the design of digital filters [86, 87]. Digital filters can be represented in many ways. One of the most common representation is the frequency domain representation \( H(\omega) \), where \( \omega \) denotes normalized angular frequency, which is a special case of the z-domain representation \( H(z) (z = e^{j\omega}) \) [17].

Important filter properties can be derived from the transfer function, e.g. the magnitude function \(|H(e^{j\omega})|\) and the phase function \( \theta(\omega) = \angle H(e^{j\omega}) \). Another important property, which in turn is derived from the phase function, is the group delay \( \tau(\omega) = -d\theta(\omega)/d\omega \). The design of digital signal processing systems can be divided into two parts: architectural design and behavioral design [88]. In filter design, the goal is to find filter parameters, so that the behavior of the filter matches a filter specification. In the specification, the desired behavior is described, given a certain architecture with architectural design parameters. For example, a (direct-form) FIR architecture has the transfer function
where the filter length \( L_h \) is a pre-chosen architectural design parameter and \( h(n) \) are filter parameters which have to be found by the design method. A filter specification can be given in the form of a desired transfer function \( H_d(\omega) \). A distance measure has to be defined between the desired transfer function and the actual transfer function. Two common distance measures are the \( L_2 \)-norm and the \( L_\infty \)-norm. A problem formulation of filter design using the \( L_2 \)-norm is the complex least squares criterion

\[
\min_{h(n)} \int_{\pi} \left| H(e^{j\omega}) - H_d(e^{j\omega}) \right|^2 d\omega
\]

where the objective function is a quadratic function. In the case of direct-form FIR filters and discretization of the optimization frequency interval, the \( L_2 \)-norm objective function leads to the method of least squares.

A problem formulation of filter design using an \( L_\infty \)-norm, the complex minimax criterion is

\[
\min_{h(n)} \max_{\omega \in \Omega} \left| H(e^{j\omega}) - H_d(e^{j\omega}) \right| d\omega
\]

which can be formulated as a semi-infinite linear program. By discretization, such problems can be approximated by finite dimensional linear programs, which can be solved using simplex or interior point algorithms [89]. A combination of linear and quadratic cost function can also be used. Such design problems can be formulated as semi-infinite quadratic programs. Combinations of linear and quadratic objective functions in combination with linear constraints can be reformulated as quadratic programs. An example of such a combination is presented for the design of windows in [90]. In [91] the versatility of the method is presented with additional specifications in filter design. Semi-
infinite programming problems can be solved using the simplex method for linear programming and the active set method for quadratic programming [89].

A number of numerical methods exists for the solution of general semi-infinite programming problems, refer for example [92, 93]. However, most methods are computationally complex and employ approximation by discretization in some step of the optimization procedure. In order to overcome the difficulties mentioned above, a new front-end for semi-infinite re-formulated complex linear and quadratic programming is presented in [94], known as the Dual Nested Complex Approximation (DNCA) algorithm, which is based on conventional finite-dimensional linear and quadratic programming. Design examples with DNCA are presented for the design of antenna arrays [95] and high accuracy windows [96].

1.4.2 Digital Filter Bank Design:

The design of filter banks comprises the following architectural aspects:

- Individual filter issues: filter type (IIR and FIR), filter order, filter structure.
- Filter bank issues: number of bands, decimation factors, filter bank structure.

The design of filter banks comprises the following behavioral aspects:

- Individual filter issues: passband size and ripple, stopband size and ripple, transition width and shape, delay.
- Filter bank issues: frequency coverage, overall delay, overall magnitude, phase and aliasing distortions.
All design aspects are interrelated, which means that there is a need for design methods, which take as many of these aspects as possible into account. In the behavioral design, the desired behavior needs to be identified and accordingly, design objectives need to be defined, i.e. distortion minimization, distortion cancellation, distortion limitation. Subsequently, a method needs to be developed designing the filter banks according to the objectives.

The ideal filter bank transfer function (the desired transfer function) has unit magnitude and zero phase for all frequencies, i.e. $T(z) = 1$. Due to the natural limitations of causal digital filters and architectural limitations, a relaxed desired transfer function $T_d(z)$ is usually defined with a unit magnitude frequency response and a constant group-delay $\tau_\omega (\omega) = c$, hence $T(z) = z^{-c}$.

Design methods with $L_2$-norm objective functions are proposed for the design of filter banks. In [97, 98], iterative constrained least squares methods are proposed for the design of perfect reconstruction and near-perfect reconstruction $M$-Channel filter banks. In that paper, the least squares method is used because of the simplicity. In [99] a design method using least squares approximation of perfect reconstruction $M$ channel filter banks is presented. The main idea behind the method is to find causal and stable synthesis filters such that the analysis-synthesis system approximates the perfect reconstruction system.

In [100, 101], design methods using an $L_\infty$-norm are presented. In [102] iterative least squares methods are proposed for the design of half-band filter banks.

Since the introduction of subband adaptive filtering, special design methods are developed for the multirate filter banks. Optimal wavelet packet filter banks for signal
decomposition for echo-cancellation are presented in [103]. An iterative least squares method is developed in [104], where the main aim is to suppress aliasing in the subbands while employing oversampling. The main purpose of the analysis filter bank is to provide whitened signals with low aliasing to the adaptive filters operating on the subband signals. Hence, in the analysis filter bank design, the shape of the analysis filters and the suppression of aliasing terms is important. The purpose of the synthesis filters is to provide a distortion-free reconstruction of the output signal, although adaptive filtering is used in the subbands.

The design of modulated filter banks is reduced to the design of the analysis and synthesis prototype filters. Prototype filter design methods using iterative optimization schemes are presented in [105]. A simple design method for uniform DFT filter banks is presented in [106] with application in echo cancellation. In Section 1.3.1, it was shown that the complete cancellation of aliasing distortion in uniform DFT filter banks is only possible under special conditions. In the case with FIR analysis and synthesis filters, the synthesis filters will become of very high order. Furthermore, it was shown that although aliasing is completely cancelled, the filter bank may still have remaining magnitude- and phase distortion, which has to be addressed by a design method.

As in the design of digital filters, a optimization method may utilize the degrees of freedom in a structure optimally. Many trade-offs between architectural and behavioral aspects exists. For example, when a delay requirement is relaxed and/or the number of filter coefficients is increased, a higher stopband attenuation in a lowpass filter may be obtained [86]. As with conventional digital filters, similar trade-offs can be made in the design of analysis and synthesis filters in a filter bank. An important aspect in filter bank design is the decimation rate. By using a decimation factor $D < M$, which is lower
than the critical sampling rate, more degrees of freedom can be introduced in the filter bank design and distortions can further be reduced. A disadvantage of oversampling is that the computational complexity of the filter bank and the subband domain processing is increased, due to the higher sampling rate. If adaptive filters are used in the subbands, longer filters need to be used to obtain the same effective time-span.

1.5 Thesis Organization:

The thesis is organized as follows:

In Chapter 1, introduction of Discrete-time filters, Multirate schemes, Subband adaptive filter, Digital filter bank techniques are discussed.

In Section 1.2, fundamentals of Adaptive filtering Least Mean Square Algorithm, FIR Weiner filter which is an optimal filter which minimizes the Mean Square Error, RLS algorithm with relevant mathematical expressions are explained. An introduction to fast implementations using fast Fourier transform and transform domain adaptive filtering for increased convergence speed and subband domain adaptive filtering, which may be regarded as a generalized form of adaptive filtering are explained. In Section 1.3, Uniform & Non-uniform Multirate Digital Filter banks are described. In Section 1.4, Digital Filter & Digital Filter Bank design with Optimization methods are explained.
In Chapter 2, a Design of Uniform DFT filter banks for subband adaptive filters is described.

In Section 2.2, design issues of $M$-channel filter banks are discussed. The uniformly modulated filter bank is presented. In Section 2.3 & 2.4, the proposed analysis & synthesis filter bank design method is described. In Section 2.5, the influence of the design parameters on the performance of the filter bank itself is evaluated. In Section 2.6, the implementation and the computational complexity of 2-times oversampled uniformly modulated filter banks compared with critically sampled uniformly modulated filter banks are presented. In Section 2.7, four different filter banks are designed using the proposed method and they were compared against four conventional filter banks with known structure as reference. In Section 2.8, conclusions of this chapter are presented.

In Chapter 3, Design of Nonuniform filter banks using Unconstrained Quadratic Optimization, Linear Optimization with Linear Constraints and Quadratic Optimization with Linear Constraints are presented.

In Section 3.2, a detailed description of analysis and synthesis filter bank structure is given, as well as the properties of the analysis-synthesis system are discussed. In Section 3.3, design methods using unconstrained quadratic optimization using least squares with design examples are presented with and without phase compensation. In Section 3.4, design methods using Linear optimization with Linear Constraints with & without Phase compensation are presented. In Section 3.5, design methods using Quadratic optimization with Linear Constraints with & without phase compensation are
presented. Design Examples are included in these sections. In Section 3.6, a comparison of the design examples is given. In Section 3.7, the filter bank complexity is discussed. In Section 3.8, conclusions of the chapter are given.

In Chapter 4, the main Conclusions of this thesis are given.