Approximation problems in decorrelation methods of adaptive filters

A. Venema

Master's degree thesis
Abstract

A commonly used algorithm in the adaptive filter theory is the Least Mean Square (LMS) algorithm. The performance of this algorithm depends on the input signal statistics. This dependency can be reduced by using decorrelation methods.

The Recursive Least Square (RLS) algorithm decorrelates the input signal with an estimate of the inverse autocorrelation matrix of the input signal. If the estimate of this (decorrelation) matrix is perfect, then perfect decorrelation will be achieved. With perfect decorrelation, the performance of the algorithm is independent of the signal statistics.

For RLS the dimension of the decorrelation matrix is equal to the filter length. For large filters this implies a high complexity. In the Block Orthogonal Projection (BOP) algorithm, the dimension of the decorrelation matrix is coupled to the block length. In general, the block length is much smaller than the filter length.

This report will show for the BOP algorithm that even with a reduced dimension decorrelation matrix, good decorrelation can be achieved. For two different signal types, an optimal value for the dimension of the BOP decorrelation matrix will be derived. It will be shown that a further increase than this optimal value will not lead to sufficient better decorrelation.

For large adaptive filters, the BOP algorithm can be implemented efficiently in the frequency-domain by using block processing techniques. But the part where a lot of complexity is involved, the decorrelation, is still carried out in the time-domain. Approximating the BOP decorrelation matrix by a circulant matrix leads to the Block Frequency Domain Adaptive Filter (BFDAF) algorithm. In the BFDAF algorithm also the decorrelation is performed efficiently in the frequency-domain. The influence of the circulant approximation on the decorrelation quality will be discussed in the last part of this report. It will be shown that for large adaptive filters, the BFDAF algorithm is a more efficient alternative than the BOP algorithm with equivalent convergence properties.
Contents

1 Introduction 1

2 Block adaptive filter algorithms 3
  2.1 The BNLMS algorithm 3
  2.2 The BOP algorithm 5

3 Efficient implementation of block adaptive filters 8
  3.1 Efficient implementation of linear convolution with overlap-save method 8
  3.2 Efficient implementation of BNLMS 11
  3.3 Efficient implementation of BOP 12
  3.4 BFDAF an efficient implementation of BOP 14
  3.5 Simulation results 16

4 Problem description 19
  4.1 Ideal signal decorrelation 19
  4.2 Approximation problems 20

5 Convergence improvement by decorrelation 22
  5.1 Convergence properties of a non-decorrelating algorithm 22
  5.2 Decorrelation in the BOP algorithm 24

6 Block length effects on decorrelation quality 26
  6.1 Quality of decorrelation 26
  6.2 Block length effects on ER of input signal 27
  6.3 Perfect decorrelation with a reduced dimension decorrelation matrix 29
  6.4 Decorrelation quality of the BOP algorithm 30
  6.5 Verifications of assertions with simulations 31
  6.6 Optimal B, for the BOP algorithm 34
  6.7 Block lengths effects on decorrelation quality, in conclusion 37

7 BFDAF, decorrelation in the frequency-domain 39
  7.1 Derivation decorrelation part BFDAF algorithm 39
  7.2 Circulant approximation 40
  7.3 Windowing effect 43
  7.4 Convergence of the BFDAF algorithm 45
  7.5 BOP versus BFDAF 47
8 Conclusions

Bibliography

A Power normalization without divisions

B Simulation results of the BOP algorithm 1

C Simulation results of the BOP algorithm 2

D Decorrelation of a male voice
Chapter 1

Introduction

The echo canceller depicted figure 1.1 is a commonly used example to demonstrate the principals of an adaptive filter. An echo canceller tries to eliminate the leakage from the input signal $x$ into the residual signal $r$. A possible application for this system is an acoustic echo canceller for audio/video teleconferencing or loudspeaking telephony systems.

![Figure 1.1: acoustic echo canceller](image)

Suppose two groups of people have a telephoneconference and a speaker at the other end of the line produces a speech signal $x$. This signal enters the room and will be reflected by the walls into the microphone. In a system without echo cancellation there will be a undesired roundsinging effect. The goal of the echo canceller is to make an estimate $\hat{e}$ of the leakage signal $e$. In ideal case $\hat{e} = e$ all echo is removed and the roundsinging effect is eliminated.

The problem here is to make the estimate of the echo because there is no complete a priory knowledge about the signal characteristics. The signals are non-stationair and the character of the echo pad is time variant. For this reason it is not possible to use a Wiener filter solution and an adaptive filter is required. An other problem is the length of the acoustic echo pad, 150-250 msec. To make a good estimate of the leakage signal $e$, large adaptive filter structures are required.
Large adaptive filters imply many calculations every sample and therefore update algorithms with low complexity are required. The Least Mean Square (LMS) update algorithm [WID85] has low complexity but the convergence properties are dependant of the statistic signal characteristics.

Since speech signals, used in the example of figure 1.1, are highly correlated, methods of adaptive filtering are developed which are not to sensitive for input signal characteristics. A well-known algorithm which is less dependant on the statistic signal properties is the Recursive Least Square (RLS) algorithm [HAY86]. The RLS algorithm makes an estimation of the inverse autocorrelation matrix of \( x \). This matrix is used to decorrelate\(^1\) the input signal which yields to better convergence properties of the adaptive filter. A disadvantage of the RLS algorithm is the complexity caused by the calculation of the decorrelation matrix. Especially at large adaptive filters, because the dimensions of the decorrelation matrix equals the adaptive filter length.

This complexity is reduced by the Block Orthogonal Projection (BOP) algorithm. This is a block-based algorithm, meaning that the update of the weights is only performed once every \( B \) samples, where \( B \) denotes the block length. The estimated autocorrelation matrix used in the BOP algorithm to decorrelate the input signal has a \( B \times B \) dimension, in general this is far more smaller than the dimension of the matrix used in RLS.

One step further is the efficient implementation of the BOP algorithm using block processing techniques. By using these techniques, the two main operations of an adaptive filter, linear convolution and linear correlation, are carried out in the frequency domain. By approximating the decorrelation matrix of the BOP algorithm by its circulant version, also decorrelation can be implemented in an efficient way. The result of this is the Block Frequency Domain Adaptive Filter (BFDAF).

This report will examine decorrelation methods used in time- and frequency-domain adaptive filters in order to gain more insight on how decorrelation works and what the effect of the approximations is that have to be made to implement the BOP algorithm in an efficient way.

Chapter 2 explains the principles of block adaptive filter algorithms. Chapter 3 describes how these filters can be implemented efficiently by using block processing techniques. In this chapter also the relation between the BOP and BFDAF algorithm will be shown. In chapter 4 the problem description is given. Chapter 5 describes in which way the input statistics affect the performance of the algorithm and how this performance can be improved by using decorrelation. Decorrelation in time-domain will be discussed in chapter 6. Chapter 7 discusses decorrelation in the frequency-domain. Finally the conclusions will be given in chapter 8.

---

\(^1\)Decorrelation means: removing the correlation, in other words making a coloured signal white again. The matrix which is used to perform decorrelation is called the decorrelation matrix.
Chapter 2

Block adaptive filter algorithms

This chapter gives a short description of the Block Normalized Least Mean Square (BNLMS) and Block Orthogonal Projection (BOP) algorithms. A detailed derivation of these algorithms is given in [SOM92]. The description will be concentrated on the implementation of the algorithms in the time–domain. The convergence properties will only be discussed briefly in the next chapter.

The derivation of the time–domain implementation of block adaptive filters will lead to the next chapter, where the efficient implementation of the block adaptive filters will be discussed. Based on these two chapters the problem description is given in chapter 4.

2.1 The BNLMS algorithm

BNLMS is the block approach of the NLMS algorithm. With $B$ the block length ($\geq 1$), the update of all weight coefficients is performed once every $B$ samples. The update rule for the BNLMS algorithm, given by equation 2.1, shows a great resemblance with the NLMS update rule.

$$w^N((k+1)B) = w^N[kB] + \frac{2\alpha}{\sigma^2[kB]} x^{N,B}[kB] e^B[kB] \quad (2.1)$$

With $w^N[kB]$ defined as the adaptive weight vector containing the $N$ adaptive weights of the adaptive filter:

$$w^N[kB] = (w_{N-1}[kB], \ldots, w_1[kB], w_0[kB])^t \quad (2.2)$$

and $e^B[kB]$ defined as the residual signal vector

$$e^B[kB] = (r[kB-B+1], \ldots, r[kB])^t \quad (2.3)$$

Furthermore $x^{N,B}[kB]$ is defined as the $N \times B$ input signal matrix.

$$x^{N,B}[kB] = \begin{pmatrix} (x^B[kB-N+1])^t \\ \vdots \\ (x^B[kB])^t \end{pmatrix} \quad (2.4)$$
with for \( i = 0, 1, \cdots, N - 1 \):

\[
\mathbf{x}^B[kB - i] = (x[kB - i - B + 1], \cdots, x[kB - i])^t
\]  

(2.5)

In these formulas, a bold-face-underlined character denotes a vector and a bold-face-not-underlined character denotes a matrix. The \( k \) is used as time index, with sample time \( T \). The transpose of a matrix or vector is denoted with \((\cdot)^t\). The adaptation constant is denoted with \( \alpha \). The \( N \) of BNLMS stands for the normalization, carried out by the division with \( \sigma_x^2 \) in equation 2.1, in order to make the convergence properties of the BNLMS algorithm independent of \( \mathcal{E}((x[k])^2) \). The estimation of \( \sigma_x^2 \) can be calculated recursive with the following equation:

\[
\sigma_x^2[(k + 1)B] = (1 - \beta) \cdot \sigma_x^2[kB] + \frac{\beta}{N} (\mathbf{x}^N[kB])^t \mathbf{x}^N[kB] \quad \text{with} \quad 0 < \beta < 1
\]  

(2.6)

The estimate of the gradient vector of the steepest-descent algorithm \( \nabla_N^B[kB] \) is given by:

\[
\nabla_N^B[kB] = -2 \mathbf{x}^N[kB] \mathbf{x}^B[kB]
\]  

(2.7)

The \((N - 1 - i)^{th}\) element of gradient vector \( \nabla_N^B[kB] \) is calculated as follows:

\[
(\nabla[kB])_{N-1-i} = -2 \cdot (\mathbf{x}^B[kB - i])^t \mathbf{x}^B[kB]
\]  

(2.8)

Each gradient vector element is an averaging of the crosscorrelation between the signals \( x \) and \( r \) over \( B \) values. In comparison with the NLMS algorithm, where no averaging is carried out, the estimation of the gradient vector for BNLMS is more accurate but also more complex. But this estimation will only be done once every \( B \) samples so the overall complexity of the BNLMS and the NLMS algorithm is of the same order. Since the updating is performed less frequently for BNLMS, the rate of convergence is slower in comparison with NLMS [SOM92].

As a rough estimate of the complexity, the linear correlation, to calculate the estimation of the gradient vector (eq. 2.7) takes \( N \times B \) multiplications. The linear convolution to calculate the output of the adaptive filter, given in equation 2.9, takes \( B \times N \) multiplications.

\[
\hat{\mathbf{e}}^B[kB] = (\mathbf{X}^N[kB])^t \mathbf{w}^N[kB]
\]  

(2.9)

The \( B \) dimensional vector \( \hat{\mathbf{e}}^B[kB] = (\hat{e}[kB - B + 1], \cdots, \hat{e}[kB])^t \) is produced once every \( B \) samples. The number of (real) multiplications/divisions to produce one new output sample is:

\[
\text{MUL}_{\text{BNLMS}} \approx \frac{2NB}{B} = 2N
\]

\[
\text{DIV}_{\text{BNLMS}} = 1.
\]  

(2.10)

As one can verify, this is the same complexity as for the NLMS algorithm. (Note that the complexity of the calculation of \( \sigma_x^2 \) is not taken into account.) The division which is made to perform the normalisation with \( \sigma_x^2 \), can be turned into a multiplication. This is easier to realize. This is done by calculating the inverse of \( \sigma_x^2 \) in a recursive way. Appendix A gives the deduction of this formula.

\[
(\sigma_x^2[(k + 1)B])^{-1} = (1 + \beta) \cdot (\sigma_x^2[kB])^{-1} - \frac{\beta}{N} ((\sigma_x^2[kB])^{-1})^2 (\mathbf{x}^N[kB])^t \mathbf{x}^N[kB]
\]  

(2.11)
2.2 The BOP algorithm

Although the BNLMS algorithm is normalized with $\hat{e}_n^2$, the convergence properties still depend on second order statistic properties of the input signal. In order to make the convergence properties of the adaptive filter less sensitive to the input signal statistics, a $B \times B$ estimate of the autocorrelation matrix is used to decorrelate the input signal. The derivation of the BOP algorithm can be done in a geometrical way. First some definitions.

The difference vector $d^N_n$ is defined as the difference between the adaptive weight vector $w^N_n$ and the optimal Wiener solution $w^N_{opt}$.

$$d^N_n = w^N_{opt} - w^N_n$$

The geometrical interpretation of $d^N_n$ is a vector that is projected on a plane that is spanned by $B$ vectors $x^N[kB], \ldots, x^N[kB]$ of dimension $N$. This is shown in figure 2.1.

![Figure 2.1: geometrical interpretation of the BOP algorithm](image)

The vector $d^N_n$ is decomposed as:

$$d^N_n[kB] = d^N_n[kB]_{\parallel} + d^N_n[kB]_{\perp}$$

Where $d^N_n[kB]_{\parallel}$ is parallel and $d^N_n[kB]_{\perp}$ is perpendicular to the plane. This implies that for all $i = 0, 1, \ldots, B - 1$:

$$d^N_n[kB]_{\parallel} = \sum_{i=0}^{B-1} c_i[kB]x^N[kB - i] = x^N[kB]_{\parallel}$$

and

$$<d^N_n[kB], x^N[kB - i]> = (d^N_n[kB])^t x^N[kB - i] = 0$$
Where \( <a, b> = a^T b \) is defined as the inner product of two vectors and \( g^B[kB] \) is defined as \((c_{B-1}[kB], \ldots, c_1[kB], c_0[kB])^T\). When assuming (for simplicity reasons) that \( s[k] = 0 \), the residual signal \( r^B[kB] \) can be written as:

\[
r^B[kB] = (X^{N,B}[kB])^T d^N[kB]
\]

(2.16)

Using the fact that \( d^N[kB] \) is perpendicular to the plane, the following manner can be derived to calculate the coefficients of the vector \( g^B \):

\[
r^B[kB] = (X^{N,B}[kB])^T d^N[kB] = (X^{N,B}[kB])^T (d^N[kB] + g^N[kB])
\]

\[
= (X^{N,B}[kB])^T d^N[kB] = (X^{N,B}[kB])^T X^{N,B}[kB] g^N[kB]
\]

(2.17)

Now the coefficients of the vector \( g^B \) can be calculated as follows:

\[
g^B[kB] = \frac{1}{N} (\hat{R}^B[kB])^{-1} r^B[kB]
\]

(2.18)

with the estimate of the \( B \times B \) "autocorrelation matrix" defined as:

\[
\hat{R}^B[kB] = \frac{1}{N} (X^{N,B}[kB])^T X^{N,B}[kB]
\]

(2.19)

Each element \((\hat{R}^B[kB])_{p,q}\) can be written as:

\[
(\hat{R}^B[kB])_{p,q} = \frac{1}{N} \langle \hat{x}^{N}[kB - B + 1 + p], \hat{x}^{N}[kB - B + 1 + q] \rangle
\]

(2.20)

When the input signal is stationary, this \((p, q)^{th}\) element is an estimate of \((p - q)^{th}\) element \( \rho_{p-q} \) of the input signal autocorrelation function.

The inner product \( \langle \hat{d}^N[kB], \hat{x}^{N}[kB - i] \rangle \) for \( i = 0, 1, \ldots, B - 1 \) is a measure for the final misadjustment. There are two ways to reduce this inner product:

1. Reduce the length of \( d^N[kB] \).
2. Make \( d^N[kB] \) more orthogonal to the plane spanned by \( \hat{x}^{N}[kB - B + 1], \ldots, \hat{x}^{N}[kB] \).

Both ways can be achieved at ones by subtracting a small part of the vector \( d^N[kB] \). This is shown in figure 2.1 and mathematically described in the next equation:

\[
\hat{d}^N[(k+1)B] = \hat{d}^N[kB] - 2\alpha \hat{x}^{N,B}[kB](\hat{R}^B[kB])^{-1} r^B[kB]
\]

(2.21)

When after a few steps \( \hat{d}^N[kB] \) is rotated in such way that it becomes orthogonal to the plane, the inner product \( \langle \hat{d}^N[kB], \hat{x}^{N}[kB - i] \rangle \) for \( i = 0, 1, \ldots, B - 1 \) becomes zero. This implies that the length of \( \hat{d}^N[kB] \) needs not to be zero for reaching the optimal solution. By using equation 2.12 and 2.21, the update equation for the BOP algorithm is given by:

\[
\hat{w}^N[(k+1)B] = \hat{w}^N[kB] + 2\alpha \hat{x}^{N,B}[kB](\hat{R}^B[kB])^{-1} r^B[kB]
\]

(2.22)
The BOP update equation shows great resemblance with the BNLMS update equation 2.1. The input signal of the BNLMS algorithm is normalized with $\hat{\sigma}_2^2$ in order to make convergence properties independent of $E\{x^2[k]\}$. For BOP, the input signal is normalized (decorrelated) with a $B \times B$ estimate of the autocorrelation matrix $\hat{R}^B[kB]$ in order to make the convergence properties also independent of second order statistics.

Another time-domain algorithm that decorrelates the input signal with an autocorrelation matrix, is the Recursive Least Square (RLS) algorithm [HAY86]. The main difference between the RLS and BOP algorithm is that instead of a $B \times B$ matrix, RLS uses a $N \times N$ estimate of the inverse autocorrelation matrix and updates this estimate every sample period. To overcome the high complexity that is involved by determining $(\hat{R}^N)^{-1}$, efficient methods are developed to determine the inverse autocorrelation matrix in a recursive way.
Efficient implementation of block adaptive filters

The main advantage of block-based-algorithms is the efficient way of implementation by using block processing techniques. The overlap-save method is a commonly used technique to apply this. The infinite input sequence \( x[k] \) is split into finite segments. The convolution of the input signal with the finite impulse response is carried out segmentwise and the output signal is a composition of all these separated signals.

In both (block based) algorithms, BNLMS and BOP, the update equation is of the next form:

\[
\begin{align*}
\mathbf{w}^N[(k+1)B] &= \mathbf{w}^N[kB] + \mathbf{X}^{N,B}[kB] \mathbf{y}^B[kB] \\
\mathbf{z}^B[kB] &= \frac{2\alpha}{\sigma_0^2[kB]} \mathbf{x}^B[kB] & \text{for BNLMS} \\
\mathbf{z}^B[kB] &= 2\alpha(\mathbf{\hat{y}}^B[kB])^{-1} \mathbf{x}^B[kB] & \text{for BOP}
\end{align*}
\]

This chapter shows how for large block lengths \( B \) the calculation of the linear convolution \( (\mathbf{X}^{N,B})^\prime \mathbf{w}^B \) (eq. 2.9) and linear correlation \( \mathbf{X}^{N,B} \mathbf{y}^B \) can be carried out efficiently in the frequency-domain. By approximating \( \mathbf{\hat{y}}^B \) with a circulant matrix also the (time-domain) decorrelation of the BOP algorithm can be implemented efficiently in the frequency-domain. This efficient implementation is called the Block Frequency Domain Adaptive Filter (BFDAF) algorithm and will be discussed at the end of this chapter.

3.1 Efficient implementation of linear convolution with overlap-save method

The (cyclic)convolution of the input signal \( x \) with the adaptive filter \( \mathbf{w}^N[kB] \) is in the frequency-domain is given in equation 3.4. In the frequency-domain, the convolution is carried out as a multiplication of consecutive vector elements [END87], denoted with \( \otimes \).

\[
\mathbf{\hat{y}}^M[kB] = \mathbf{X}^M[kB] \otimes \mathbf{W}^M
\]
$X^M$ is the frequency-domain representation of $x^M$ (the choice of $M$ will be discussed later on, but for now $M \geq N$). A mathematical description of the Discrete Frequency Transformation (DFT$_M$) of $x^M$ is given in the next equation:

$$X^M[kB] = F^M \cdot x^M[kB]$$

(3.5)

The DFT operation is represented by a $M \times M$ Fourier matrix $F^M$. The $(k, l)^{th}$ element of this matrix is defined as:

$$(F^M)_{k,l} = e^{-j\theta_M k l} \quad \text{with} \quad \theta_M = \frac{2\pi}{M} \quad \text{and} \quad k, l \in (0, 1, \cdots, M - 1)$$

(3.6)

To carry out the convolution in the frequency-domain, $w$ has to be of the same length as $x$. Therefore $w^N[kB]$ is first mirrored ($J^N$) and augmented with zeros ($0^{M-N}$) to a vector of length $M$. The $(k, l)^{th}$ element of the $N \times N$ mirror matrix $J^N$ is defined as:

$$(J^N)_{k,l} = \begin{cases} 1 & \text{if } k + l = N - 1 \\ 0 & \text{elsewhere} \end{cases}$$

(3.7)

With the $N \times N$ identity matrix $I^N$ and $(M - N) \times N$ all-zero matrix $0^{M-N,N}$, the mathematical description of the transformation of $w^N$ is as follows:

$$W^M = F^M \left( \begin{array}{cc} I^N \\ 0^{M-N,N} \end{array} \right) J^N w^N$$

(3.8)

Due to the cyclic convolution effect [END87], the first $N - 1$ samples of the inverse transformation (IDFT$_M$) of $\tilde{E}^M$ are worthless. For the computation of $B$ new output samples, $M$ has to be at least $N - 1 + B$, the last $M - B$ samples have to be discarded. Mathematically this can be described as:

$$\tilde{E}^B[kB] = (0^{0,M-B} \quad I^B \quad (F^M)^{-1} \quad \tilde{E}^M[kB]$$

(3.9)

The implementation of the overlap-save implementation of this convolution is depicted in figure 3.1. The DFTs which are used can be replaced by FFTs if $M$ is a power of two. The "S/P" box collects the last $B$ input samples in a tapped−delay−line and produces once every $B$ samples a vector $x^B[kB]$, so a downsample factor $B$ is introduced. The "overlap" box creates a $M$−length vector $x^M[kB]$, which contains the last $B$ input samples augmented with an overlap of the last $M - B$ samples of $x^M[(k - 1)B]$. The "P/S" box stores the vector $\tilde{E}^B[kB]$ in a tapped−delay−line and produces with original sample rate every time unit $T$ an output sample $\tilde{e}[kB - i]$ (for $i = B - 1$ downto 0). As a result of the block processing technique, the processing delay is always $B − 1$ samples.

Mathematically the operations of figure 3.1 can be described as follows: In equation 2.9, $X^{N,B}$ has to be transformed to the frequency−domain instead of $x^M$. From literature [DAV79] it is known that a circulant matrix can be diagonalized by transforming it to the frequency−domain. Therefore and as a consequence of the cyclic nature of the Fourier transform, matrix $X^{N,B}$ has to be extended to a circulant matrix $\tilde{X}^M$. The $M \times M$ circulant matrix $\tilde{X}^M[kB]$ is created by putting the mirrored matrix $J^N \cdot X^{N,B}$

\[1\]$X^{N,B}[kB]$ and $x^M$ contain exactly the same input samples.
in the upper right corner of $\hat{X}^M$ and filling the remaining elements in such a way that $\hat{X}^M$ becomes circulant. Mathematically, the relationship can be described as follows:

$$
\begin{bmatrix}
I^N & 0^{N,M-N}
\end{bmatrix}
\hat{X}^M[kB]
\begin{bmatrix}
0^{M-B,B} \\
I^B
\end{bmatrix}
= J^N \cdot X^{N,B}[kB]
$$

(3.10)

The diagonalization of $\hat{X}^M$ is performed as follows:

$$
F^M(\hat{X}^M[kB]) (F^M)^{-1} = X^M[kB] \quad \text{with} \quad X^M[kB] = \text{diag}(X^M[kB])
$$

(3.11)

The mathematically derivation of the overlap–save method for the linear convolution (eq. 2.9) is given in the next equation by using the relationship given in equation 3.10.

$$
\begin{align*}
\hat{e}^B[kB] &= (X^{N,B}[kB])^T W^N[kB] \\
&= (J^N X^{N,B}[kB])^T J^N W^N[kB] \\
&= (0^{B,M-B} I^B) (F^M)^{-1} F^M (\hat{X}^M[kB]) (F^M)^{-1} F^M (J^N 0^{M-N,N}) J^N W^N[kB] \\
&= (0^{B,M-B} I^B) (F^M)^{-1} F^M (\hat{X}^M[kB]) W^N[kB] \\
&= (0^{B,M-B} I^B) (F^M)^{-1} \left\{ X^M[kB] \otimes W^M[kB] \right\} \\
&= (0^{B,M-B} I^B) (F^M)^{-1} E^M[kB].
\end{align*}
$$

(3.12)

Since the transformations applied to $z$ and $w$ are linear, the time–domain implementation of the convolution (eq. 2.9) has exactly the same output as the discreet frequency–domain implementation of equation 3.12.
3.2 Efficient implementation of BNLMS

The two main operations of the BNLMS algorithm are linear convolution to calculate the filter output and linear correlation to calculate the estimate of the gradient. The overlap-save implementation of linear correlation \( X_{N,B}, y_B \) (at the end of equation 3.1) is equivalent to the overlap-save implementation of linear convolution (eq. 3.12). The derivation is given in the next equation:

\[
X_{N,B}[kB] = J_N (I^N_{0N,M-N}) (F^M - 1) F^M \tilde{Y}_{M}[kB] (F^M - 1) F^M \left( \begin{array}{c}
0^M-B,B \\
I^B
\end{array} \right) Y^B[kB]
\]

\[
= J_N (I^N_{0N,M-N}) (F^M - 1) (X_{M}[kB])^* \tilde{Y}_{M}[kB]
\]

\[
= J_N (I^N_{0N,M-N}) (F^M - 1) \{ (X_{M}[kB])^* \otimes \tilde{Y}_{M}[kB] \}
\]

(3.13)

Where \( \tilde{Y}_{M}[kB] \) is defined as:

\[
\tilde{Y}_{M}[kB] = F^M \left( \begin{array}{c}
0^M-B,B \\
I^B
\end{array} \right) y^B[kB]
\]

(3.14)

For BNLMS, \( y^B = 2a(\tilde{\sigma}_{2}^2[kB])^{-1} x^B[kB] \) (eq. 3.2). The overlap–save of BNLMS implementation is depicted in figure 3.2. The normalization is carried out by multiplying \( R^M \) with \( 2a(\tilde{\sigma}_{2}^2 1^M) \), where \( 1^M \) denotes a vector containing only ones. The number of (real) multiplications without the overlap–save implementation is about \( 2N + 1 \). The overlap–save implementation of figure 3.2 costs:

- 5 FFTs (if \( M \) is a power of two); Each FFT or IFFT costs \( \frac{3}{4} \cdot M \log_2(M) \) real multiplications.
  Totally \( 5 \cdot \frac{3}{4} \cdot M \log_2(M) \) multiplications.

- 2 complex vector multiplications; One complex multiplication is equivalent to four real multiplications. Each vector is of length \( M \), so this costs \( 4 \cdot 2 \cdot M \) real multiplications. Since \( x, w \) and \( x \) are real vectors, the frequency–domain vectors have symmetry properties, so only the half of the frequency components have to be calculated. Each complex vector multiplication costs \( 4 \cdot \left( \frac{1}{2}M - 1 \right) + 2 = 2 \cdot (M - 1) \) real multiplications.
  Totally \( 4 \cdot (M - 1) \) multiplications

- 1 real vector multiplication; The calculation of \( (2a(\tilde{\sigma}_{2}^2 1^M) \cdot R^M \) can be carried out in the time–domain. This costs \( B \) multiplications instead of \( M \).
  Totally \( B \) multiplications.

Note that again the complexity of the calculation of \( \tilde{\sigma}_{2}^2 \) is not taken into account. The total number of real multiplications to calculate one new output sample is:

\[
\text{MUL}_\text{BNLMS} \approx \frac{5 \cdot \frac{3}{4} \cdot M \log_2(M) + 4 \cdot (M - 1) + B}{B}
\]

(3.15)

Comparing this with the complexity of the time–domain implementation of BNLMS then it is to be seen that the complexity of the efficient implementation of BNLMS decreases exponential with increasing \( B \) while the time–domain complexity still is \( 2N + 1 \). As mentioned before, this way of implementing has no effect on the output of the adaptive filter, because all performed transformations are linear.
3.3 Efficient implementation of BOP

The implementation of the BOP algorithm is equivalent to the implementation of BNLMS. The difference in equation 3.13 is that now \( \tilde{\mathbf{z}}^B[kB] = 2\alpha (\hat{\mathbf{R}}^B[kB])^{-1} \mathbf{r}^B[kB] \). The implementation of the BOP algorithm is depicted in figure 3.3.

The filter part is equals that of the BNLMS algorithm. The normalization with \( \tilde{\sigma}_z^{-2}[kB] \) in the BNLMS algorithm is now replaced by a decorrelation with the inverse of \( \hat{\mathbf{R}}^B[kB] \), denoted by the box with the cross at the bottom of the figure. This box denotes a matrix-vector multiplication of \( (\hat{\mathbf{R}}^B[kB])^{-1} \) with \( \mathbf{r}^B[kB] \).

The higher complexity of the BOP algorithm is caused by the decorrelation of the input signal. Comparing the complexity of the normalization in the BNLMS algorithm with the complexity of the decorrelation in the BOP algorithm shows:

BNLMS: the calculation of \( \tilde{\sigma}_z^{-2}[kB] \), implemented like in equation 2.11, costs about \( N \) multiplications. The normalization costs \( B \) multiplications.

Totally \( \Rightarrow N + B \) multiplications.
**BOP:** the calculations can be split in three parts: 1. The calculation of $\hat{\mathbf{R}}^{B}[kB]$, implemented like in equation 2.19, costs $B^2 \times N$ multiplications. 2. The calculation of the inverse of $\hat{\mathbf{R}}^{B}[kB]$ costs in the order (denoted by $O$) of $B^3$ multiplications. 3. The calculation of the product of $(\hat{\mathbf{R}}^{B}[kB])^{-1}$ with $\mathbf{X}^{M}[kB]$ costs $N \times B$ multiplications.

Totally $\Rightarrow (B + 1)(B \times N) + O(B^3)$ multiplications.

Both calculations are performed ones every $B$ samples. It does not need a long argument to see that in spite of the better convergence properties of the BOP algorithm, the complexity is so much higher that the algorithm in many cases is not useful. With reduced complexity, the BOP algorithm will be an interesting algorithm. Therefore a closer look will be taken at the part where the most complexity is involved: the decorrelation part of the algorithm.
### 3.4 BFDAF: an efficient implementation of BOP

The DFT of $\chi^M$ for BOP can be written as:

$$\chi^M[kB] = 2\alpha F^M \left( \begin{array}{c} 0^{M-B,B} \\ I_B \end{array} \right) (\hat{R}^B[kB])^{-1} \chi^B[kB]$$  \hspace{1cm} (3.16)

As mentioned before, a circulant matrix can be diagonalized by transforming it to the frequency-domain. When $\hat{R}^B[kB]$ is a circulant matrix, also decorrelation can be performed in the frequency domain. The problem is that $\hat{R}^B[kB]$ is not a circulant matrix but a toeplitz matrix. And even if $\hat{R}^B[kB]$ is a circulant matrix, then the dimension is too small for a diagonalization by $F^M$. This last point can be solved by extending $\hat{R}^B[kB]$ to a $M \times M$ matrix. The approximation of a toeplitz matrix by a circulant matrix is described by [KU92].

The extension and circulant approximation can be described as follows: Extend the first row of $\hat{R}^B[kB]$, which equals the estimated autocorrelation vector $\hat{\rho}^B[kB] = (\hat{\rho}_0[kB], \ldots, \hat{\rho}_{M-1}[kB])^t$, to a vector of length $M$, $\hat{\rho}^M[kB] = (\hat{\rho}_0[kB], \ldots, \hat{\rho}_{M-1}[kB])^t$.

The extended vector $\hat{\rho}^M$ is approximated by the "pseudo-autocorrelation" vector in the following way:

$$\tilde{\rho}_i[kB] = \frac{M - i}{M} \hat{\rho}_i[kB] + \frac{i}{M} \hat{\rho}_{M-i}[kB]$$  \hspace{1cm} (3.17)

with

$$\tilde{\rho}^M[kB] = (\tilde{\rho}_0[kB], \ldots, \tilde{\rho}_{M-1}[kB])^t$$  \hspace{1cm} (3.18)

The "pseudo-autocorrelation" vector is equal to the first row of the circulant matrix (the remaining rows are filled in such way that the matrix becomes circulant). The circulant matrix will be denoted by $\tilde{R}^M[kB]$. Now the matrix $\tilde{R}^M$ is circulant, it can be diagonalized with the Fourier matrix as follows:

$$F^M \tilde{R}^M[kB](F^M)^{-1} = P^M[kB]$$  \hspace{1cm} (3.19)

Where $P^M[kB]$ is a diagonal matrix defined as:

$$P^M[kB] = \text{diag}(P^M[kB]) \quad \text{with} \quad P^M[kB] = F^M \tilde{\rho}^M[kB]$$  \hspace{1cm} (3.20)

By using the just described approach of the approximation/extension of $\hat{R}^B[kB]$ by $\tilde{R}^M[kB]$, the decorrelation in the time-domain, expressed in equation 3.16 for the BOP algorithm, can now be implemented in the frequency-domain. Mathematically this is described in the following equation:

$$\chi^M[kB] \approx 2\alpha F^M (\tilde{R}^M[kB])^{-1}(F^M)^{-1} \chi^B[kB]$$

$$= 2\alpha (P^M[kB])^{-1} \otimes \tilde{R}^M[kB]$$  \hspace{1cm} (3.21)

Note that this is not an exact transformation. $\hat{R}^B$ is now replaced by $\tilde{R}^M$. $\tilde{R}^M$ is not only an approximation of $\hat{R}^B$, but also an extension to a $M \times M$ matrix. Because of this extension, the augmentation with $0^{M-B,B}$ zeros (eq. 3.16) is no longer necessary for this matrix. To perform the transformation of $\tilde{R}^M[kB]$ to the frequency-domain, the augmentation is still necessary.
The second row of equation 3.21 shows how decorrelation is now reduced into an elementwise vector multiplication. The operator $(\cdot)^{-1}$ denotes the elementwise inversion of a vector.

Just like the smoothed estimation of $\sigma^2_s[kB]$ of the BNLMS algorithm, each normalization factor for each frequency bin can be estimated in the same way:

$$
(P^M[(k + 1)B])^{-1}_l = (1 - \beta)(P^M[kB])^{-1}_l + \frac{\beta}{M} ||(X^M[kB])||^2
$$

(3.22)

Also in this case the division of $(P^M[(k + 1)B])$ with the $l^{th}$ frequency bin can be replaced by a multiplication with the approximation of its inverse (see Appendix A):

$$
(P^M[(k + 1)B])^{-1}_l = (1 + \beta)(P^M[kB])^{-1}_l - \frac{\beta}{M} (P^M[kB])^{-2}_l ||(X^M[kB])||^2
$$

(3.23)

Figure 3.4 shows the efficient implementation of the BOP algorithm in the frequency domain. This implementation is called the Block Frequency Domain Adaptive Filter (BFDAF). Note that the most important difference between the implementation BFDAF (fig. 3.4) and the implementation of BOP (fig. 3.3) is that the normalization (decorrelation) is now carried out in the frequency domain by multiplying each frequency bin of the input signal spectrum of $x$ with $(P^M)^{-1}_l$. In mathematics the most important difference is showed in the difference between equation 3.16 and equation 3.21. Here one can see that $\hat{R}^B$ is approximated by $\hat{R}^M$. With the diagonalization of $\hat{R}^M$ the matrix-vector multiplication of the decorrelation in BOP is now reduced to a vector-vector multiplication.

The normalization in the BFDAF algorithm costs:

- A vector multiplication of $R^M$ with $(P^M)^{-1}$. This is not a complete complex vector multiplication. Because of the symmetry properties of $\hat{P}^M$, $\hat{P}^M$ is a real symmetric vector (so is $(P^M)^{-1}$).
  
  Totally $\Rightarrow M$ multiplications.

- The estimation of $(P^M[kB])^{-1}$. This costs 1 complex- and 3 real vector multiplications.
  
  Totally $\Rightarrow 3 \times (\frac{1}{2} \cdot M + 1) + 2 \cdot (M - 1)$ multiplications.

The multiplications are real multiplications and are only performed ones every $B$ samples.
To verify the statements about the convergence properties of the BNLMS, BOP and the BFDAF algorithm, some simulations are performed. For a detailed analyses see [LEE89]. The acoustic echo canceller, depicted in figure 1.1, is implemented in SPOX. SPOX is a high-level software interface to the underlying hardware. In this case a TMS320C30, a Digital Signal Processor (DSP) of Texas Instruments. Vector and matrix math operations and linear filtering are one of the main features of SPOX.

As [HAY86] described, the convergence properties decreases as the eigenvalue ratio of the input autocorrelation matrix increases. The eigenvalue ratio \( ER \) is defined as:

\[
ER = \frac{\lambda_{\text{max}}}{\lambda_{\text{min}}}
\]  

(3.24)

Where \( \lambda_{\text{max}} \) is the maximum and \( \lambda_{\text{min}} \) is the minimum eigenvalue of the input autocorrelation matrix. For a white noise input signal \( z \), no decorrelation is required. The output \( r[k] \) is for all algorithms the same. To determine the difference between the convergence properties of the algorithms, input signals with a high \( ER \) are required. Two
commonly used models of stationary input signals are the Moving Average (MA) and the Auto Regressive (AR) model.

**Moving Average signal model of order 1, MA(1):**
An MA(N) signal is generated by applying a white noise signal \( n[k] \) to a Finite Impulse Response (FIR) filter of order N. The white noise signal has zero mean and variance \( \sigma_n^2 \).

For an MA-signal of order 1, this model is defined as:

\[
\begin{align*}
\text{Model: } & x[k] = \frac{1}{\sqrt{1 + a^2}} \cdot (n[k] + an[k - 1]) \\
\text{Autocorrelation: } & \rho[\tau] = \begin{cases} 
\frac{\sigma_n^2}{1 + a^2} & \text{for } \tau = 0 \\
0 & \text{for } \tau = \pm 1 \\
& \text{elsewhere}
\end{cases} \\
\text{Spectrum: } & P(e^{j\theta}) = \left(1 + \frac{2a}{1 + a^2} \cos(\theta)\right) \sigma_n^2 \\
\text{Eigenvalue Ratio: } & ER \leq \left(\frac{1 + |a|}{1 - |a|}\right)^2
\end{align*}
\]

**Auto Regressive signal model of order 1, AR(1):**
An AR(N) signal is generated by applying the white noise signal \( n[k] \) to a all-pole filter of order N.

\[
\begin{align*}
\text{Model: } & x[k] = (\sqrt{1 - a^2}) \cdot n[k] + ax[k - 1] \\
\text{Autocorrelation: } & \rho[\tau] = a^{\tau} \sigma_n^2 \\
\text{Spectrum: } & P(e^{j\theta}) = \frac{(1 - a^2)\sigma_n^2}{1 + a^2 - 2a \cos(\theta)} \\
\text{Eigenvalue Ratio: } & ER \leq \left(\frac{1 + |a|}{1 - |a|}\right)^2
\end{align*}
\]

The models are scaled in such way that the eigenvalue ratio and the signal power is the same for both if the parameters \( a \) and \( \sigma_n^2 \) are equal.

For this experiment an AR(1) input signal \( x[k] \) is used with a sample frequency of \( 8k\text{Hz} \). The model parameters are, \( a = 0.8182 \) (\( ER \approx 100 \)) and \( \sigma_n^2 = 1 \). With an echo pad of 250ms, an adaptive filter of length 2000 is required to cancel the echo. In this experiment the adaptive filter length (\( N \)) is 1949 and the block length (\( B \)) is 100. Note that in most cases a \( B \) of 100 is not acceptable because the delay is then 12.5ms. The signal \( s \) is a zero mean white noise signal with \( \sigma_s^2 = 0.01 \). The echo pad is a simulated room response, modelled as a FIR filter of order 2000. The results of this experiment are depicted in figure 3.5. The quantity of interest in this figure is the relative misadjustment \( \tilde{J} \) defined as:

\[
\tilde{J}[k] = \frac{\mathcal{E}\{(e[k] - \hat{e}[k])^2\}}{\mathcal{E}\{(e[k])^2\}} = \frac{J[k] - J_{\text{min}}}{J_{\text{min}}}
\]

Where \( J[k] = \mathcal{E}\{(r[k])^2\} \), the misadjustment with as minimum \( J_{\text{min}} = \mathcal{E}\{s[k]\}^2 \). The \( \alpha \) is chosen in such a way that for all algorithms \( \tilde{J}[\infty] = -10\text{dB} \).
It is obvious that the BOP algorithm has much better convergence properties than the BNLMS algorithm. Only the price that have to be paid in terms of complexity is much higher. In this experiment the number of multiplications involved with the normalization is for BNLMS 20 and for BOP $O(10^4)$. This complexity is reduced by using the BFDAF algorithm (92 multiplications), while maintaining good convergence properties. The efficiency of the overlap-save implementation is expressed in the difference of the number of multiplications involved with the time- and frequency-domain implementation of the BNLMS algorithm. The time-domain implementation costs 3899 multiplications each sample (eq. 2.10) and the overlap-save implementation 774 (eq. 3.15).

The next chapter will give the problem description and enumerates some points that have to be investigated to gain more insight into decorrelation in time- and frequency-domain.

![Convergence of the BNLMS, BOP and BFDAF algorithm](image)

**Figure 3.5:** convergence of the BNLMS, BOP and BFDAF algorithm
Chapter 4

Problem description

The simulation results in section 3.5 show that for coloured input signals the BOP algorithm has much better convergence properties than the BNLMS algorithm. A disadvantage of the BOP algorithm is its higher complexity.

For this reason methods are developed to implement the BOP algorithm in an efficient way. Chapter 3 described how an efficient implementation of the BOP algorithm leads to the BFDAF algorithm. In spite of the approximations, made to realize this efficient implementation, the simulation results of section 3.5 show that there is no big difference between the convergence properties of both filters.

The goal of this report is to gain insight into which approximations are made and what their effect is on the convergence properties. The first section of this chapter describes the ideal case of signal decorrelation. The second section will evaluate on basis of the previous chapters in which way the decorrelation of the BOP and BFDAF algorithm differ from the ideal case of signal decorrelation.

4.1 Ideal signal decorrelation

The effect of decorrelation on the autocorrelation function of the input signal is depicted in figure 4.1. Figure 4.1.a shows the autocorrelation function of an arbitrarily input signal \( x \). For real signals \( \rho[\tau] \) is a symmetric function, \( \rho[\tau] = \rho[-\tau] \) for \( \tau = -\infty, 0, \infty \). This function can also be represented by a matrix, the autocorrelation matrix \( \mathcal{R} \) (of infinite dimensions). Both descriptions represent the time-domain description of second order statistics of the input signal \( x \).

By applying the Fourier Transform for Discrete signals (FTD) this function can be written as:

\[
P(e^{j\theta}) = \sum_{\tau=-\infty}^{\infty} \rho[\tau] e^{-j\theta \tau}
\]

\[
= \rho[0] + 2 \sum_{\tau=1}^{\infty} \rho[\tau] \cos(\theta \tau), \quad -\pi \leq \theta < \pi
\]

(4.1)

The function \( P(e^{j\theta}) \) is called the power spectral density function (psdf) and represents the frequency-domain description of the second order statistics of input signal \( x \).
psdf is depicted in figure 4.1.b and is also a real symmetric function caused by the real nature of the input signal.

To reduce the dependency of the signal statistics on the convergence properties of the adaptive filter, decorrelation methods are used. In the time-domain, decorrelation can be performed by multiplying the autocorrelation matrix with its inverse. The autocorrelation function of the decorrelated signal is depicted in figure 4.1.c and equals the autocorrelation function of a white noise signal. The psdf of the decorrelated signal is depicted in figure 4.1.d.

Instead of decorrelating with the inverse autocorrelation matrix (the time-domain approach), decorrelation is also possible in the frequency-domain by multiplying the spectrum of the input signal with the inverse psdf of figure 4.1.b. In this case no inverse matrix multiplication is needed, saving a lot of complexity.

4.2 Approximation problems

The implementation of the decorrelation of the input signal, as described in chapter 2 and 3 is not exactly like the ideal case as described in the previous section. To make the implementation possible, some approximation have to be made. This section gives an overview of these approximations.

**Block length B**

In the ideal case input signal $x$ is decorrelated by an autocorrelation matrix with infinite dimensions. In the BOP algorithm, only an autocorrelation matrix with dimension $B \times B$ is used. Chapter 6 will discuss the influence of this reduced dimension on the quality of decorrelation.
Circulant approximation of $\tilde{R}^B$

Section 3.3 described how an efficient implementation of the BOP algorithm leads to the BFDAF algorithm by the approximation of $\tilde{R}^B$ with the circulant matrix $\tilde{R}^M$. The effect of this circulant approximation will be discussed in chapter 7.

Transformation with the DFT

In the ideal case of section 4.1 the transformation from time- to frequency-domain is performed with the FTD. In reality this transformation is performed with a DFT (or FFT). The DFT of a signal is only based on a little piece of its time-domain representation and therefore it is not always a reliable frequency-domain representation of the whole signal.

If the DFT is used as linear transformation to implement for example a convolution efficiently in the frequency-domain, then this last point has no influence. If the result of the DFT is used as source for estimating the spectrum, like in equation 3.22 for determining the normalization vector $p^M$, then the reliability of the frequency-domain representation is very important. Because the quality of the estimating $p^M$ is not a subject of this report and the DFT will only be used as a linear transformation, this last point will not be discussed.

In literature, [DEE90], [EGE92] and [NAR83] quoted some of these points but they never gave a statement about the influences of these approximations and on which conditions the BFDAF algorithm is a good alternative for the BOP algorithm. This report will, based on the examination of time- and frequency-domain decorrelation methods, give answer to these questions.
Chapter 5

Convergence improvement by decorrelation

In the experiment of section 3.5 is shown that filters, performing decorrelation on the input signal, have much better convergence properties. This section describes in which way decorrelation improves convergence properties. For simplicity, from now on all signals are supposed to be stationairy and ergodic.

5.1 Convergence properties of a non-decorrelating algorithm

First some formulas. The residual vector \( r^B[kB] \) of all block adaptive filters can be written as:

\[
r^B[kB] = (X^N,B)^t d^N[kB] + s^B[kB]
\] (5.1)

Where \( d^N[kB] \) is defined as in equation 2.12. By using these definitions, the update equation of the BNLMS algorithm (eq. 2.1) can be written as:

\[
d^N[(k+1)B] = 
\left( I^N - \frac{2\alpha}{\sigma_s^2} X^{N,B}[kB](X^{N,B}[kB])^t \right) d^N[kB] \\
- \frac{2\alpha}{\sigma_s^2} X^{N,B}[kB] s^B[kB]
\] (5.2)

The average behaviour of \( d^N[kB] \) can be split in two parts. A part that is changing fast in time, \( X^{N,B}[kB] \) and a part that is changing much slower, \( d^N[kB] \), caused by the adaptation constant \( \alpha \). For this reason \( X^{N,B}[kB] \) and \( d^N[kB] \) may be separated under \( \mathcal{E}\{\} \). Assuming that \( x \) and \( s \) are independent and using \( \mathcal{E}\{s[k]\} = 0 \), the average of the difference vector is approximated by:

\[
\mathcal{E}\{d^N[(k+1)B]\} \approx \left( I^N - \frac{2\alpha}{\sigma_s^2} \mathcal{R}^N \right) \mathcal{E}\{d^N[kB]\}
\] (5.3)

With the autocorrelation matrix \( \mathcal{R}^N \) of the stationairy input signal \( x \) defined as:

\[
\mathcal{R}^N = \frac{1}{B} \mathcal{E} \left\{ X^{N,B}[kB](X^{N,B}[kB])^t \right\}
\] (5.4)

22
When \( z \) is a white noise signal there is no difference between the BNLMS algorithm and an algorithm using decorrelation. Since for that case the autocorrelation matrix \( R^N \) is a diagonal matrix with \( \sigma_z^2 \) on the diagonal. The result is that all elements of \( d^N[(k+1)B] \) are updated independent of each other with the same amount of power.

If \( z \) is a non-white signal then \( R^N \) is a matrix with non-zero elements besides the main diagonal. In this case the update of an element of \( d^N[(k+1)B] \) depends on more than one value of \( d^N[kB] \). The problem of the mutual dependency of the difference vector elements can be conquered by transforming the difference vector equation 5.3 with the matrix \( Q^N \). The matrix \( Q^N \) is an orthonormal matrix containing the eigenvectors of the autocorrelation matrix \( R^N \). The matrix \( R^N \) can be diagonalized in the following way:

\[
(Q^N)^h R^N Q^N = \Lambda^N \quad \text{with} \quad \Lambda^N = \text{diag}\{\lambda_0, \ldots, \lambda_{N-1}\}
\]  
(5.5)

Where \( \Lambda^N \) is a diagonal matrix, containing the eigenvalues of \( R^N \). The \( h \) of \( Q^h \) denotes the hermitian transpose of the matrix ( \( Q^h = (Q^*)^t \) ). The transformation of \( d^N[kB] \) is performed as follows:

\[
D^N[kB] = (Q^N)^h d^N[kB] = (Q^N)^h (w_{\text{opt}}^N - w^N[kB])
\]  
(5.6)

By using the property \( Q^h Q = I \) of orthonormal matrices, the difference vector equation of the BNLMS algorithm can be transformed in the following way:

\[
\mathcal{E}\{D^N[(k+1)B]\} \approx \left( I^N - \frac{2\alpha}{\sigma_z^2} \Lambda^N \right) \mathcal{E}\{D^N[kB]\}
\]  
(5.7)

In this equation all elements of \( D^N[kB] \) are updated independently of each other. For each element can be determined if it converges to the optimal Wiener solution. The optimal Wiener solution is reached if \( \mathcal{E}\{D^N[kB]\} = 0 \) for \( k \to \infty \). By using the initial value \( \mathcal{E}\{D^N[0]\} = w_{\text{opt}}^N - w^N[0] = D^N[0] \), the update of each individual element of the vector of equation 5.7 can be written as:

\[
\mathcal{E}\{D_i[kB]\} = \left( 1 - \frac{2\alpha}{\sigma_z^2} \lambda_i \right)^k D_i[0] \quad \text{for} \quad i = 0 \ldots N - 1
\]  
(5.8)

The transformed difference vector \( D^N \) shows that the dependency on the signal statistics can be expressed in terms of eigenvalues. When \( z \) is a white noise signal all eigenvalues are equal, the update of each element \( D_i[kB] \) is performed with the same amount of power. If \( z \) is a non-white signal, then the eigenvalues are not equal. In this case the elements \( D_i[kB] \) are updated with a different amount of power.

From equation 5.8 can be derived that the optimal Wiener solution will be reached if:

\[
\lim_{k \to \infty} \left( 1 - \frac{2\alpha}{\sigma_z^2} \lambda_i \right)^k = 0 \quad \text{for} \quad i = 0, \ldots, N - 1
\]  
(5.9)
This implies that the convergence region is bounded by:

\[ 0 < \alpha < \frac{\sigma^2}{\lambda_{\max}} \] (5.10)

Another convergence property is the rate of convergence \( v_{20} \). This quantity is defined as the number of iterations needed to decrease the relative misadjustment \( \tilde{J} \) (eq. 3.27) by 20 dB. Without proof (see [SOM92]), this quantity is:

\[ v_{20} = \frac{20\tau}{10 \log(e)} \] (5.11)

Where \( \tau \) is the time constant. The time constant is a composition of \( N - 1 \) different time constants and is defined as the number of samples needed to decrease \( \lambda_i \mathcal{E} \{ |D_i[kB]|^2 \} \) by a factor \( e \). In mathematics (for proof see [SOM92]), for BNLMS:

\[ \tau_i = \frac{-1}{\ln |1 - 4\frac{\sigma^2}{\lambda_i} \lambda_i|} \approx \frac{1}{4\frac{\sigma^2}{\lambda_i} \lambda_i} \] (5.12)

This equation implies that the eigenvalue distribution has also influence on the rate of convergence. For \( \lambda_{\max} \) the time constant is small, which implies a fast rate of convergence. But the time constant is also bounded by \( \alpha \), because \( \alpha < \frac{\sigma^2}{\lambda_{\max}} \) and this restriction will slow down the rate of convergence. For \( \lambda_{\min} \) the corresponding \( \tau_i \) is the largest, which implies a slow rate of convergence.

To make it even more complex, not every \( \tau_i \) delivers the same contribution to the overall time constant \( \tau \). The weight of the contribution of \( \tau_i \) to \( \tau \) depends on the product \( \lambda_i \mathcal{E} \{ |D_i[kB]|^2 \} \). The largest product delivers the largest contribution to the composition of \( \tau \) and the corresponding \( \tau_i \) is called the local time constant \( \tau_i \).

The influence of the eigenvalue distribution on the convergence properties is evaluated in [SOM92]. [SOM92] shows that the eigenvalue distribution does not always have to have a negative influence on the convergence properties. But in general the white noise case has a faster rate of convergence and/or a better final misadjustment.

### 5.2 Decorrelation in the BOP algorithm

Following the same approach as with BNLMS, the update equation for the BOP algorithm can be written as:

\[
\hat{d}^N[(k+1)B] = \left( I^N - 2\alpha \chi^N[B][\hat{R}^B[kB])^{-1}(\chi^N[B][kB])^t \right) d^N[B] \\
- 2\alpha \chi^N[B][kB][\hat{R}^B[kB])^{-1} s^B[kB] \tag{5.13}
\]

Average this equation implies:

\[
\mathcal{E}\{d^N[(k+1)B]\} \approx \left( I^N - 2\alpha \mathcal{E}\{\chi^N[B][kB][\hat{R}^B[kB])^{-1}(\chi^N[B][kB])^t\} \right) \mathcal{E}\{d^N[B]\} \tag{5.14}
\]

If no decorrelation is carried out, that is if \( \hat{R}^B[kB] = \sigma^2 I^B \), then the BNLMS update equation (eq. 5.2) equals the BOP update equation. This implies that the only difference between the two algorithms is the matrix \( \hat{R}^B[kB] \). The goal of this matrix is to make
the convergence properties of the algorithm independent of the signal characteristics by decorrelating the input signal. Mathematically, decorrelation in the BOP algorithm can be expressed in the following way:

\[
\frac{1}{B} X^{N,B}[kB][\bar{R}^B[kB]]^{-1}(X^{N,B}[kB])'
\]

(5.15)

This expression is a part of equation 2.21 and is called the decorrelation part of the BOP algorithm. The fraction \( \frac{1}{B} \) is added for scaling. When the average of this part equals \( 1^N \), all elements of the difference vector are updated independent of each other with the same amount of power and perfect decorrelation is achieved. With perfect decorrelation the convergence region and time constant equals:

\[
0 < \alpha < 1 \quad \text{and} \quad \tau_r \approx \frac{1}{4\alpha}
\]

(5.16)
Chapter 6

Block length effects on decorrelation quality

For RLS an $N \times N$ autocorrelation matrix is used to decorrelate the input signal. For BOP this is only a $B \times B$ matrix. Which is in general much smaller. This reduced dimension saves a lot of complexity, but the question is whether this matrix has enough decorrelation power. In other words, is a $B \times B$ autocorrelation matrix big enough to decorrelate the input signal?

This question will be examined on basis of the influence of $B$ on the BOP decorrelation part. It is shown that decorrelation quality improves with an increasing $B$. It will also be shown that till a certain $B$, the decorrelation quality improves very fast. This particular $B$ is a candidate for the optimal $B$: the $B$ at which a further increase will not lead to sufficient better decorrelation quality.

With this in mind it is to be expected that also the convergence properties of the whole algorithm will not improve sufficiently when $B$ is increased further than the optimal $B$. But the influence of $B$ on the convergence behaviour is not univocal. With an increasing $B$ the decorrelation quality will improve, resulting in better convergence properties. But when $B$ increases the update of the algorithm is performed less frequently, this will slow down the convergence process. These influences are contradictorily, therefore on optimal $B$ is hard to find.

When the results of the examination of the decorrelation part and the convergence behaviour of the whole algorithm are combined, it follows that in spite of the contradictorily influences of $B$, there still is a point that can be defined as the optimal $B$.

6.1 Quality of decorrelation

The decorrelation of the input signal $x$ is included in equation 5.14. The part of interest is:

$$\frac{1}{B}E\left\{X^{N,B}[kB] (\hat{R}^{B}[kB])^{-1} (X^{N,B}[kB])^{t}\right\}$$  \hspace{1cm} (6.1)

If this part is equal to the identity matrix $I^N$, then perfect decorrelation is achieved. But when it is not equal to the identity matrix, how can quality be determined?
[GRA72] describes that two toeplitz matrices are asymptotically equivalent when their eigenvalues behave similarly. In this case this implies that a matrix is asymptotically equivalent to the identity matrix when the eigenvalue ratio is equivalent to one.

For BNLS the convergence properties depends on the eigenvalue ratio of the autocorrelation matrix $R^N$ of the input signal $x$. For BOP the convergence properties depend on the eigenvalue ratio of equation 6.1. When the eigenvalue ratio is reduced to 1, good decorrelation is achieved. Therefore as measurement for the quality of decorrelation the eigenvalue ratio, $ER$, will be used. ($ER$ is defined in equation 3.24)

### 6.2 Block length effects on $ER$ of input signal

Before something can be said about the decorrelation in the BOP algorithm, first the initial case has to be examined, the case without decorrelation, like in BNLS. For BNLS the eigenvalues of the input signal autocorrelation matrix $R^N$ (eq. 5.4) are of great interest. The matrix $R^N$ is only a small cut out of the full autocorrelation matrix $R$, having infinite dimensions. But what is small? This section will investigate when the reduced dimension autocorrelation matrix $R^N$ is a good representation of $R$.

[HAY86] describes that the eigenvalues of an autocorrelation matrix are bounded by the maximum and minimum values of the associated power spectrum density function (psdf) like in the next equation:

$$P_{\min} \leq \lambda_i \leq P_{\max} \tag{6.2}$$

Where $P_{\min}$ is the minimum and $P_{\max}$ is the maximum value of the psdf. But there are more relations between the eigenvalue distribution and the psdf. Figure 6.1 shows for different signal models the eigenvalue distribution and the psdf.

---

**Figure 6.1:** eigenvalue distribution and psdf for the MA(1) and AR(1) signal models
For the MA(1) and AR(1) signal models the parameters are chosen in such way that the eigenvalue ratio is about 100. The solid lines represent the psdf and the dots represent the eigenvalues of the autocorrelation matrix. Each type of dot denotes the eigenvalue distribution of the autocorrelation matrix of a different dimension.

As the matrix dimension of $\mathcal{R}^N$ increases, the eigenvalues fit more and more on the psdf curve. For large $N$ there can be said:

$$P(j^{i\theta_i})|_{\theta_i=\theta_N} \approx \lambda_i \quad \text{with} \quad \theta_N = \frac{2\pi}{N} \quad (6.3)$$

This has also influence on the eigenvalue ratio of $\mathcal{R}^N$. For the MA(1) signal of figure 6.1, $ER(\mathcal{R}^4) = 8.7$ and $ER(\mathcal{R}^{32}) = 81.5$. In general the eigenvalue ratio increases with an increasing dimension of the matrix $\mathcal{R}^N$, bounded by $\frac{P_{\max}}{P_{\min}}$. This effect is shown in figure 6.2. The $ER_{max}$ in this figure is defined as: $ER_{max} = ER(\mathcal{R}^\infty)$ or equivalent $ER_{max} = \frac{P_{\max}}{P_{\min}}$. The horizontal axis denotes the $n \times n$ matrix dimension and vertically the eigenvalue ratio of $\mathcal{R}^n$ is plotted.

If $ER(\mathcal{R}^N) \approx ER_{max}$, then the eigenvalue distribution of the MA(1) and AR(1) signals matches with the psdf. So the eigenvalue distribution of $\mathcal{R}^N$ matches with the eigenvalue distribution of $\mathcal{R}$. On basis of the theorem of [GRA72] (also previous section) this implies that $\mathcal{R}^N$ is asymptotically equivalent with $\mathcal{R}$.

Figure 6.2: eigenvalue ratio of the $n \times n$ autocorrelation matrix $\mathcal{R}^n$
6.3 Perfect decorrelation with a reduced dimension decorrelation matrix

The ideal case in section 4.1 showed that with \( R^{-1} \) (of infinite dimension) perfect decorrelation can be achieved. In the RLS algorithm, decorrelation is performed with a \( N \times N \) (approximated) inverse autocorrelation matrix. This section will show that even with this reduced dimension, \( N \) instead of \( \infty \), perfect decorrelation is possible.

In the next equation the update rule of the RLS algorithm is given:

\[
\mathbf{w}^N[k + 1] = \mathbf{w}^N[k] + 2\alpha(\hat{R}^N[k])^{-1}x^N[k]r[k]
\] (6.4)

With \( r[k] = (x^N[k])^Td^N[k] + s[k] \) and \( d^N[k] = w_{\text{opt}}[k] - \mathbf{w}^N[k] \), this equation can also be written as:

\[
d^N[k + 1] = \left( \mathbf{I}^N - 2\alpha(\hat{R}^N[k])^{-1}x^N[k](x^N[k])^T \right) d^N[k]
- 2\alpha(\hat{R}^N[k])^{-1}x^N[k]s[k]
\] (6.5)

The average behaviour of the difference vector can be determined in the same way as that of the BNLMS. The result is given in the next equation:

\[
\mathcal{E}(d^N[k + 1]) = \left( \mathbf{I}^N - 2\alpha(\hat{R}^N[k])^{-1}R^N[k] \right) \mathcal{E}(d^N[k])
\] (6.6)

With \( R^N = \mathcal{E}(x^N[k](x^N[k])^T) \) the autocorrelation matrix. To examine the decorrelation quality, the part of interest of equation 6.6 is \((\hat{R}^N[k])^{-1}R^N[k]\). Suppose the signals are stationary and the inverse autocorrelation matrix is perfectly estimated: \((\hat{R}^N[k])^{-1} = (R^N)^{-1}\), then \((\hat{R}^N[k])^{-1}R^N[k] = I^N\) and perfect decorrelation is achieved.

By applying the orthogonal transform with \( Q^N \) to equation 6.6 the difference equation becomes as follows:

\[
\mathcal{E}(D^N[k + 1]) = \left( \mathbf{I}^N - 2\alpha(\hat{A}^N[k])^{-1}A^N \right) \mathcal{E}(D^N[k]) \quad \text{"RLS"}
\] (6.7)

In the "eigenvalue" domain (for example fig. 6.1) the decorrelation is performed by multiplying each separate \( \lambda_i \) by his inverse \( \lambda_i^{-1} \). From this the conclusion can be drawn that to perform perfect decorrelation the eigenvalue distribution of the decorrelation matrix \((R^N)^{-1}\) does not have to match the (inverse)psdf of the input signal \( x \), but has to match with the (inverse)eigenvalue distribution of \( R^N \)!

\(^1\)Note that "RLS" means that with the assumption \((\hat{R}^N[k])^{-1} = (R^N)^{-1}\) the algorithm is no longer equal to the RLS algorithm. Now it is equal to the LMS/Newton algorithm.
6.4 Decorrelation quality of the BOP algorithm

In the previous section it is shown that for perfect decorrelation a decorrelation matrix of infinite dimensions is not necessary. For RLS the dimension of the decorrelation matrix is reduced to $N$. It was shown that even with this reduced dimension (and with some assumptions) perfect decorrelation can be achieved. By the derivation of the BOP algorithm (chap. 2) a method of decorrelation is derived where the dimension of the decorrelation matrix is decoupled from the filter length. The dimension of the decorrelation matrix $(\hat{R}^B[kB])^{-1}$ is coupled to the block length $B$.

In first instance the decorrelation part of the BOP algorithm will be examined. The goal of decorrelation is removing the colouring of a signal. The previous sections shows that the eigenvalue distribution is a good reflection of the colouring a signal. To simplify the examination, only AR(1) and MA(1) signals are used. For these signals the eigenvalue ratio is a good reflection of the colouring of the signals, caused by the monotonously increasing/decreasing eigenvalue distribution.

Later on the convergence properties of the whole BOP algorithm will be discussed. To avoid the influence of the quality of the estimation of $\hat{R}^B[kB]$, this estimation is assumed to be perfect, $\hat{R}^B[kB] = R^B$.

Decorrelation part of the BOP algorithm

Where the decorrelation part of the RLS algorithm, $(R^N)^{-1} \mathcal{E}(x^N[k][x^N[k]]^t)$, was easy to examine, for BOP things get more complicated. The average of the decorrelation part of the BOP algorithm is given in the next equation:

$$\tilde{I}^N = \frac{1}{B} \mathcal{E} \{ x^{N,B}[kB](R^B)^{-1}(x^{N,B}[kB]^t) \} \quad (6.8)$$

Where $\tilde{I}^N$ represents the "autocorrelation" matrix after decorrelation. The tilde is used to express that this matrix is not exactly, but approximately equal to the identity matrix $I^N$.

For RLS the average behaviour of the input signal $z$ could be easy separated from the decorrelation matrix. For BOP this separation is not possible, even a transformation to the "eigenvalue"-domain is out of the question. Therefore an instinctively approach will be followed to examine the decorrelation part of the BOP algorithm. The assertions will be verified by earlier results and simulations.

For BNLMS is showed in section 5.1 that the convergence properties depend on the eigenvalue distribution of $R^N$ and in section 6.2 it is shown that this distribution strongly depends on $N$. In the BOP algorithm the dependency of the eigenvalue distribution is reduced by decorrelation with $(R^B)^{-1}$. The eigenvalue distribution of $R^B$ is given by $(Q^B)^h R^B Q^B = \Lambda^B$. The eigenvalue distribution of $R^N$ is given by $(Q^N)^h R^N Q^N = \Lambda^N$.

It is mathematically not true that decorrelation in the BOP algorithm can be seen as a normalization of $\Lambda^N$ with $\Lambda^B$ as with RLS. Because $\Lambda^N$ and $\Lambda^B$ can not be separated under equation 6.8.
Even when this separation is possible a normalization of $\mathbf{A}^N$ with $\mathbf{A}^B$ with $N \neq B$ is hard to imagine. But there are two cases that are reasonable trivial:

1. If $B = 1$, then $\text{ER}(\hat{\mathbf{I}}^N) = \text{ER}(\mathcal{R}^N)$: no decorrelation is carried out.

2. If $B = N$, then $\text{ER}(\hat{\mathbf{I}}^N) \approx 1$: almost perfect decorrelation.

Note that point 2 says almost perfect decorrelation. The case for $B = N$ is comparable with the decorrelation of RLS, but not equal. Although there can not be spoken about a normalization of the eigenvalues it is obvious that, when the eigenvalue distribution of $\mathcal{R}^B$ match with the eigenvalue distribution of $\mathcal{R}^N$, good decorrelation quality will be achieved. For the used signal models, MA(1) and AR(1), there will only be looked at the maximal and minimal eigenvalue, because these values are characteristic for the total shape of the eigenvalue distribution.

In figure 6.1 it is to be seen that for the MA(1) signal the minimum eigenvalue, $\lambda_{\text{min}}$, of the matrices $\mathcal{R}^B$ and $\mathcal{R}^N$ lie on the same place. Therefore it is reasonable that the minimum eigenvalue of $\hat{\mathbf{I}}^N$ (in this case) can be estimated by $\frac{(\mathbf{A}^N)_{\text{min}}}{(\mathbf{A}^B)_{\text{min}}}$. When $N \neq B$ the maximum eigenvalues of $\mathcal{R}^N$ and $\mathcal{R}^B$ lie not on the same place, but for both matrices this is the last eigenvalue. So (in this case) the maximum eigenvalue of $\hat{\mathbf{I}}^N$ can be estimated by $\frac{(\mathbf{A}^N)_{N-1}}{(\mathbf{A}^B)_{B-1}}$.

In the same way this can be derived for the AR(1) signal of figure 6.1. From this can be concluded that the eigenvalue ratio of $\hat{\mathbf{I}}^N$ is estimated by:

$$
\text{ER}(\hat{\mathbf{I}}^N) \approx \frac{\text{ER}(\mathcal{R}^N)}{\text{ER}(\mathcal{R}^B)}
$$

(6.9)

Note that this estimation is only valid for signals with an monotonously increasing or decreasing psdf. For other signals further examination is required.

### 6.5 Verifications of assertions with simulations

In this section all assertions made about decorrelation in the BOP algorithm will be checked on basis of simulation results. To measure the eigenvalue ratio of the decorrelation part $\hat{\mathbf{I}}^N$ the next computation is implemented in SPOX:

$$
\mathcal{A}\{X^{N,B}[kB](\mathcal{R}^B)^{-1}(X^{N,B}[kB])^t\}
$$

(6.10)

The average in this computation is not an ensemble average $\mathcal{E}\{\cdot\}$ like in equation 6.8, but a time average $\mathcal{A}\{\cdot\}$. Due to the stationairy and ergodic nature of the input signal, as supposed in the introduction of the previous chapter, these two types of averaging are the same. The implemented part is equal to the $\hat{\mathbf{I}}^N$ of equation 6.8.

The time average is approximated by an average over 10.000 iterations. The number of 10.000 is chosen because an increase of this number will not lead to sufficient better results (with 20.000 iterations the results only differ a few percent). The goal of this experiment is to measure the quality of decorrelation as function of the dimension of the decorrelation matrix, this will be defined as the decorrelation function. As measurement for “quality of decorrelation” the eigenvalue ratio of $\hat{\mathbf{I}}^N$ is used. To avoid the influence of the quality of the estimation of $\mathcal{R}^B[kB]$, this matrix is replaced by the matrix $\mathcal{R}^B$. 

31
For this experiment the used signals are generated by the AR(1) and MA(1) model. The parameters of the models are \(|\alpha| = 0.8182\) \((ER_{\text{max}} \approx 100)\) and \(\sigma_\varepsilon^2 = 1\). The decorrelation functions for the different signal models are plotted in figure 6.3.

The solid lines represent the decorrelation functions of \(\hat{I}^N\), generated by the simulations. The dashed lines are an estimate of this function. This estimate is made by the approximation made in equation 6.9, \(ER(\hat{I}^N) \approx \frac{ER(\hat{R}^N)}{ER(\hat{R}^N)}\). For \(B = 1\) (not shown in this plot) no decorrelation is carried out and \(ER(\hat{I}^N) = ER(\hat{R}^N)\). For \(B > 20\) the \(ER(\hat{I}^N) \approx 1\).

In figure 6.3 only the decorrelation function for a filter of length 100 is plotted. For this filter length the eigenvalue ratio of \(\hat{R}^N\) is almost equal to the maximum, see figure 6.2. When \(N > 100\) the eigenvalue ratio will only increase a little, the decorrelation function will approximately be the same.

For a smaller \(N\) the eigenvalue ratio of \(\hat{R}^N\) will decrease, which implies by the approximation of equation 6.9 that the eigenvalue ratio of the decorrelation part decreases. This effect is showed by the (SPOX) simulation of figure 6.4. The simulations are carried out for the same models but for different filter lengths.

![Diagram showing decorrelation functions for AR and MA models.](image-url)
Decorrelation part of the BOP algorithm, in conclusion

The theory (or hypothesis) of section 6.4 can be verified on the basis of the simulation results of this section.

- For $B < N$ the theory matches with the results of the simulations. The estimate of the eigenvalue ratio by equation 6.9 is a reasonable approximation of the simulated eigenvalue ratio.

- For $B = N$ the simulation results show that perfect decorrelation, which according to the estimate of equation 6.9 should be achieved, is not totally reached. For the filters of length $N = 100$ and $N = 20$ the difference is very small. For the filter of length $N = 8$, the simulation shows that $ER(\hat{\mathbf{I}}^N) = 2.1$ for the MA signal\(^2\). This difference can be explained by the use of $\mathcal{R}^B$ as decorrelation matrix, instead of $\hat{\mathcal{R}}^B$. If $\hat{\mathcal{R}}^B$, as defined in equation 2.19, is used as decorrelation matrix, then it can be proved that perfect decorrelation will be achieved. For large filters each element of $\hat{\mathcal{R}}^B$ will be averaged over a lot of samples, because of that $\hat{\mathcal{R}}^B$ is approximately equal to $\mathcal{R}^B$. For small filters (like $N = 8$) this is no longer the case.

- For $B > N$ the theory is no longer valid. In theory perfect decorrelation should be achieved at $B = N$. A further increase of $B$ leads by the approximation $ER(\hat{\mathbf{I}}^N) \approx \frac{ER(\mathcal{R}^N)}{ER(\mathcal{R}^B)}$ to an $ER(\hat{\mathbf{I}}^N) < 1$, which make no sense.

\(^2\) $ER(\hat{\mathbf{I}}^N) \approx 1.2$ for the AR signal
In conclusion can be said that when \( \hat{R}^B \) is a good approximation of \( R^B \), equation 6.9 is a good approximation of the decorrelation function for \( B \leq N \). Equation 6.9 expresses that the quality of decorrelation improves with an increasing \( B \).

Note that examination of the BOP decorrelation part is only carried out for AR(1) and MA(1) signals. For other input signals, further examination is required.

### 6.6 Optimal \( B \), for the BOP algorithm

Figure 6.4 shows for all simulations that in the lower range of \( B \) the eigenvalue ratio decreases more than linear with increasing \( B \). For \( B > 15 \) this decrease only a little. It does not need a long approach to see that a further increase of \( B \) is quite useless. This section will examine the influence of the block length on the whole algorithm.

**Criterion for optimal block length.**

Based on the knowledge of the previous section the optimal \( B \) of the BOP algorithm can be defined as:

The block length at which a further increase of \( B \) will only lead to an asymptotic increase of the convergence properties.

With asymptotic increase will be meant the little decrease of the eigenvalue ratio of \( \tilde{I}^N \) to the asymptote of \( ER = 1 \) by an increasing \( B \) (see fig. 6.3).

Defining a criterion is not the problem, but how can the optimal \( B \) be determined? Therefore in first instance a look will be taken at the convergence of the BOP algorithm. For \( N = 100 \) and for different \( B \), the convergence curves of the BOP algorithm are plotted in figure 6.5. The \( \alpha \) is chosen in such way that the final misadjustment \( J = -10 \) dB.

For \( B = 20 \), where \( ER(\tilde{I}^N) \approx 1 \), the algorithm acts almost as in the ideal (white noise) case and perfect decorrelation is almost achieved. For a decreasing \( B \) the curves differ more and more from the ideal case. But it can not be said that the performance is worse over the whole range. Initially some curves show even a better performance than the white noise case. This can be explained by the following points:

- If \( B \) decreases then the \( ER \) of \( \tilde{I}^N \) increases. Because of this the convergence properties will decrease.

- With a decreasing \( B \) the update of the adaptive filter is performed more frequently, which results in a better rate of convergence.

- Also the nature of the signal and the shape of the echo pad is of influence. Chapter 5 (and [SOM92]) describes that it is possible that the eigenvalue distribution delivers a positive contribution to the rate of convergence. When \( B \) increases the signal is more decorrelated, implying a decrease of the positive influence of the eigenvalue distribution. The convergence rate will increase.

Some of these statements are opposite to each other, therefore an optimal \( B \) is hard to find. Also the criterion for determining the difference between the curves is hard to find. For the AR signal the quantity \( u_{20} \) (eq. 5.11) can be used as criterion. For the MA signal
Figure 6.5: convergence of BOP for different block lengths B

This quantity shows no big difference for different block lengths (see fig. 6.5). This is caused by the just described contradictorily influences generated by a decreasing B.

Although the rate of convergence and the final misajustment \( \tilde{J} \) are almost the same for \( B > 4 \), there is definitely a difference between the points of where the final state is reached. Determining this point is rather difficult caused by the fluctuations in \( \tilde{J}[k] \). Determining the point where \( \tilde{J} \) is reached by 3 dB is an option. A better option is the point where the AR signal stops to decrease linear. At this point \( v_{20} \) is almost a linear extrapolated, so the results are for the AR signal basically the same. This point will be defined as \( \psi_{5} \), the point where the final misajustment is reached by 5 dB. Although this point looks a little bit arbitrary, in this case it can be used to determine the difference between the convergence curves. A definition of this point is given by;

\[
\psi_{5} = k \quad \text{for the } k \text{ where } 10 \log J[k] = 10 \log \tilde{J} + 5 \text{dB} \tag{6.11}
\]

35
Determining the optimal $B$, simulation results

To determine the optimal $B$ of the BOP algorithm, many simulations for different filter and block lengths have been done. The results of the simulations on the convergence behaviour are compared to the results of the decorrelation part (fig. 6.4). On the basis of this comparison a criterion for the optimal $B$ will be determined.

To get an impression in which range the optimal $B$ has to be found, a closer look is taken at the convergence of the AR signal of figure 6.5. For $B = 20$ the algorithm acts almost the same as in the white noise signal. For $B = 8$ the convergence properties are somewhat decreased and for $B = 4$ there is a significant difference with the convergence of the ideal (white noise) case. The corresponding eigenvalue ratio of $\tilde{I}^N$ (fig. 6.4) for $B = 4 \ldots 20$ are within the range of 5 downto 1.8.

For other AR(1) and MA(1) signals it is to be expected that they behave in the same way for the same eigenvalue ratio. After all, when $ER = 1$ they act just like a white noise signal. Therefore, only will be looked at the convergence for block lengths at which the $ER$ of $\tilde{I}^N$ are within the range of 1.8 to 5. This range is only a small cut out of the total range. To increase the distinction capability the values in figure 6.6 are plotted on a log-base.

![logarithm of decorrelation function](image)

The results are shown in table 6.1. The items at the left side of the table denote a range of $x - 0.5 \leq x < x + 0.5$ dB. The simulation of the BOP algorithm is carried out for the smallest possible $B$, fitting in this range. From each simulation $\psi_6$ (eq. 6.11) is given in the table. The submarkers of $\psi_6$ denote the $B$ at which this value is achieved. The submarker $\infty$ denotes that this is the ideal case. As input signal for the ideal case a white noise signal is used (simulated with small $B$). The simulation for the MA signal with $N = 8$ and $B = 11$ is not carried out because the implemented BOP algorithm is not capable of handle block lengths larger than $N$. The plots of these simulations can be found in appendix B.
The figures of this table are visualized in figure 6.7. The values of $\psi_5$ are represented as the difference with the ideal case, expressed in percents. The difference between the values of $\psi_5$ for the different signal models at the same filter length is only 10%. This confirms the earlier made assertion that the filter behaves the same for different input signals when the eigenvalue ratio of $\hat{I}_N$ is the same. Based on these results the optimum for the eigenvalue ratio can defined as:

The $ER$ at which an decrease of 1 dB will only lead to an asymptotic improvement of $\psi_5$.

Applying this definition to figure 6.7 leads not to a clear answer. Although not very clear, it is to be seen that when $ER$ is further decreased than 5 dB, $\psi_5$ will not decrease in the same order as it did before. If the eigenvalue ratio is 5 dB$^3$ ($ER \approx 3.2$), then all decorrelation functions of figure 6.4 behave asymptotically.

### 6.7 Block lengths effects on decorrelation quality, in conclusion

Perfect decorrelation does not have to be achieved with an autocorrelation matrix of infinite dimensions. The RLS algorithm shows, when the dimension of the autocorrelation matrix equals the filter length, perfect decorrelation can be achieved. It is also derived that to perform perfect decorrelation, the eigenvalue distribution of the (inverse) decorrelation matrix, does not have to match with the psdf of the input signal, but with the eigenvalue distribution of the input signal autocorrelation matrix.

For BOP the decorrelation quality improves with an increasing block length. For input signals generated by the MA(1) and AR(1) signal models it is shown that, till an $ER$ of 3 (5 dB), the decorrelation function decreases very fast. A further increase of $B$ leads not to a sufficient decrease of the eigenvalue ratio.

\[3 \text{the corresponding values of } B \text{ at } 10 \log ER = 5dB \text{ can be found in table 6.1}\]
Figure 6.7: percentage difference of $\psi_5$ with respect to the ideal case

Although not explicitly shown as for the decorrelation function, also for the convergence of the whole BOP algorithm, it can be said that a further decrease of the eigenvalue ratio than 5 dB leads not to sufficient better convergence properties.

The reason why the optimum of $B$ is expressed more clearly in the decorrelation functions than in the convergence curves of the BOP algorithm, can be found in the direct relation between the decorrelation part and the decorrelation function. Changes in $B$ will be directly reflected on the decorrelation function. But the decorrelation part is only a part of the BOP algorithm. By changing $B$ also other parts of the algorithm change. Therefore changes in the decorrelation part will not be reflected directly on the convergence properties of the whole algorithm.

Based on the behavior of the decorrelation function and the convergence properties of the algorithm, it can be concluded that for input signals generated by the MA(1) and AR(1) model, the $B$ where the eigenvalue ratio is decreased till 5 dB, is the optimal $B$. For other input signals, further examination is required for determining the optimal $B$. 

38
Chapter 7

BFDAF, decorrelation in the frequency-domain

For the BOP algorithm decorrelation is performed in the time-domain. In chapter 3 it is shown that decorrelation can be performed much more efficiently in the frequency-domain. The result of this is called the Block Frequency Domain Adaptive Filter (BFDAF) algorithm.

The difference between the BOP and BFDAF algorithm is the way of implementing the decorrelation part. One of the differences is that for BFDAF decorrelation is performed with the inverse circulant matrix \( \tilde{R}^M \), instead of \( (\tilde{R}^B)^{-1} \), as with BOP.

The question rises in which way the circulant approximation, and all other differences between BOP and BFDAF, affect the properties of the adaptive filter. This chapter derives under which circumstances the BFDAF algorithm is an useful alternative for the BOP algorithm.

To examine this, the same approach will be followed as in chapter 6. In first instance the behaviour of the BFDAF decorrelation part will be discussed. It will be shown that because of the differences in the decorrelation part, BFDAF is sensitive for other parameters than BOP. The second part shows the behaviour of the whole BFDAF algorithm, influenced by examined parameters of the decorrelation part. From the results of both parts, finally the conclusion will be drawn under what circumstances BFDAF is a good alternative for the BOP algorithm. For simplicity reasons all signals are supposed to be stationary.

7.1 Derivation decorrelation part BFDAF algorithm

Section 3.4 derives that to implement the BOP decorrelation part in the frequency-domain, the following approximation has to be done\(^1\):

\[
\begin{pmatrix}
0^{M-B,B} \\
I^B
\end{pmatrix}
(\tilde{R}^B[kB])^{-1} \approx (\tilde{R}^M[kB])^{-1}
\begin{pmatrix}
0^{M-B,B} \\
I^B
\end{pmatrix}
\]

\(\text{(7.1)}\)

\(^1\)The approximation results from the difference between equation 3.16 and equation 3.21.
Instead of approximation, better can be spoken about replacement. In this equation \( \hat{R}^B \) is replaced by a matrix \( \hat{R}^M \) which is a circulant approximation of \( R^M \). The matrix \((0 \ 1^B)^t\) in the BOP algorithm used for augmenting zeros to \((\hat{R}^B)^{-1}\), is now used as a matrix which cuts out a window of \((\hat{R}^M)^{-1}\). Mainly there are two points that have to be examined:

1. Circulant approximation, influenced by \( M \)
2. Windowing effect, influenced by \( B \)

With circulant approximation is meant: the approximation of an autocorrelation matrix \( R^M \) by a circulant matrix \( \hat{R}^M \). With windowing effect is meant: the influence of cutting out a window of \((\hat{R}^M)^{-1}\) by multiplying this matrix with \((0 \ 1^B)^t\).

For the examination of these points, the BFDAF decorrelation part will be used. The BFDAF decorrelation part can be derived by transforming the BOP decorrelation part in such way that equation 7.1 can be filled in. This can be achieved by the substitution of equation 3.10 (transformation \( X^{N,B} \rightarrow \hat{X}^M \)) in the BOP decorrelation part (eq. 5.15). With this the BOP decorrelation part can be written as:

\[
\frac{1}{B}J^N (I^N \ 0^{N,M-N}) \hat{X}^M[kB] \left( \begin{array}{c} 0^{M-B,B} \\ I^B \end{array} \right) (\hat{R}^B[kB])^{-1}(X^{N,B}[kB])^t \quad \text{BOP (7.2)}
\]

Substituting the approximation of equation 7.1 in equation 7.2 gives the time-domain representation of the BFDAF decorrelation part:

\[
\frac{1}{B}J^N (I^N \ 0^{N,M-N}) \hat{X}^M[kB](\hat{R}^M[kB])^{-1} \left( \begin{array}{c} 0^{M-B,B} \\ I^B \end{array} \right) (X^{N,B}[kB])^t \quad (7.3)
\]

For an efficient implementation this equation has to be transformed to frequency-domain. However this is not necessary for the examination of the properties of the decorrelation part. After all, the transformation to the frequency-domain has no influence on the eigenvalue distribution of the decorrelation part. The circulant approximation will be discussed in section 7.2 and the windowing effect in section 7.3.

### 7.2 Circulant approximation

This section examines the influence of the approximation of an autocorrelation matrix \( R^L \) by a circulant matrix \( \hat{R}^L \) on the decorrelation quality of an adaptive filter. The circulant approximation is carried out according to equation 3.17.

The influence of the circulant approximation will not examined on basis of the BFDAF decorrelation part, here also the windowing effect is of influence.

Two algorithms that clearly express the effect of the circulant approximation are the RLS and the FDAF (Frequency Domain Adaptive Filter) algorithm. The FDAF algorithm can be seen as an frequency-domain implementation of the RLS algorithm, where decorrelation is performed in the frequency-domain. To achieve decorrelation in the frequency-domain, the decorrelation matrix of the RLS algorithm has to approximated by a circulant matrix. This can be derived as follows:
The RLS update rule is given by equation 6.4. This equation can be transformed to the frequency domain by multiplying both sides with $(\mathcal{F}^N)^*$. Using the fact that $(\mathcal{F}^N)^{-1} = \frac{1}{N} (\mathcal{F}^N)^*$, it can be written as follows:

$$(\mathcal{F}^N)^* \mathbf{w}^N[k + 1] = (\mathcal{F}^N)^* \mathbf{w}^N[k] + 2\alpha (\mathcal{F}^N)^{-1} (\tilde{\mathbf{R}}^N[k])^{-1} \mathcal{F}^N (\mathcal{F}^N)^* \mathbf{x}^N[k] r[k]$$ (7.4)

With the following approximation:

$$(\mathcal{F}^N)^{-1} \tilde{\mathbf{R}}^N[k] \mathcal{F}^N \approx (\mathcal{F}^N)^{-1} \tilde{\mathbf{R}}^N[k] \mathcal{F}^N = \mathbf{P}^N[k]$$ (7.5)

and the next definitions:

$$(\mathbf{W}^N)^*[k] = (\mathcal{F}^N)^* \mathbf{w}^N[k]$$
$$(\mathbf{X}^N)^*[k] = (\mathcal{F}^N)^* \mathbf{x}^N[k]$$ (7.6)

the update rule of the FDAF algorithm is given by:

$$(\mathbf{W}^N)^*[k + 1] = (\mathbf{W}^N)^*[k] + 2\alpha (\mathbf{P}^N[k])^{-1} (\mathbf{X}^N[k])^* r[k] \quad \text{FADF}$$ (7.7)

A more detailed derivation of the FDAF algorithm is given in [NAR83].

The average of the RLS decorrelation part is equal to $(\tilde{\mathbf{R}}^N[k])^{-1} \mathbf{R}^N[k]$ (derived in section 6.3). With the assumption that $x$ is stationary and the estimate of $(\tilde{\mathbf{R}}^N[k])^{-1}$ is perfect, the decorrelation part becomes $(\mathbf{R}^N)^{-1} \mathbf{R}^N$. With this, perfect decorrelation will always be achieved.

In the time-domain the only difference between the RLS and the FDAF algorithm is that $\mathbf{R}^N[k]$ is approximated by $\tilde{\mathbf{R}}^N[k]$. This implies that the average time-domain representation of the FDAF decorrelation part is equal to $(\tilde{\mathbf{R}}^N)^{-1} \mathbf{R}^N$. Also here $x$ is supposed to be stationary and the estimate of the underlying autocorrelation function of $\tilde{\mathbf{R}}^N$ is supposed to be perfect.

The influence of the circulant approximation can be determined by the examination of the FDAF decorrelation part. If $(\tilde{\mathbf{R}}^N)^{-1} \mathbf{R}^N$ is approximately equal to $\mathbf{I}^N$, then $\tilde{\mathbf{R}}^N$ is a good approximation of $\mathbf{R}^N$.

The influence of the circulant approximation on the decorrelation quality is expressed in figure 7.1. Using the $ER$ as measurement for decorrelation quality is in this case not sufficient. The reason for this can be found in the shape of the "eigenvalue" spectrum. As mentioned before, because of the monotonously increasing/decreasing eigenvalue distribution of an AR(1) or MA(1) signal, the $ER$ is a reasonable measure of the whiteness of these signals. This is no longer the case for the decorrelated signal of the FDAF algorithm. The eigenvalue distribution of the FDAF decorrelation part shows a few eigenvalues which are much higher, and a few eigenvalues which are much lower than the mean eigenvalue. The majority of the eigenvalues are almost equal to the mean eigenvalue. A judgement about decorrelation quality on basis of only a few side-slips in the eigenvalue distribution is not fair.
Therefore the decorrelation quality will be determined in the following way:

\[ WM = \frac{1}{N} \sum_{i=0}^{N-1} \left( \max\left( \frac{\lambda_i}{\bar{\lambda}}, \frac{\bar{\lambda}}{\lambda_i} \right) - 1 \right) \]  

(7.8)

with

\[ \bar{\lambda} = \frac{1}{N} \sum_{i=0}^{N-1} \lambda_i \]  

(7.9)

\( WM \) stands for whiteness measure. The decorrelation quality is determined on basis of all eigenvalues. In case of a white noise signal, all \( \lambda_i \) are the same. The corresponding \( WM \) is 0.

The FDAF decorrelation part is examined in the same way as the BOP decorrelation part. In first instance there will be looked at the input signal and subsequently there will be looked at the decorrelated signal.

For the input signal, the function \( WM(\mathcal{R}^n) \) for \( n = 1, \ldots, \infty \) behaves basically the same as the function \( ER(\mathcal{R}^n) \) of figure 6.2 and is therefore not depicted. Instead of \( ER(\mathcal{R}^\infty) \approx 100 \), now \( WM(\mathcal{R}^\infty) \approx 5 \).

For the decorrelated signal, the \( WM \) of \((\tilde{\mathcal{R}}^N)^{-1}\mathcal{R}^N\) is depicted in figure 7.1 as function of the filter length \( N \). This decorrelation function can be split in tree parts, namely:

1. A part where \( WM = 0 \): if \( N = 1 \) or 2 then \( \tilde{\mathcal{R}}^N = \mathcal{R}^N \), perfect decorrelation is achieved.

2. A part where \( WM \) increases: For the MA(1) signal only the \((\hat{\mathcal{R}}^N)_1\) and \((\hat{\mathcal{R}}^N)_{N-1}\) are affected by the circulant approximation. For \( N = 3 \) the circulant approximation is
at worse. For a further increase of \( N \) the circulant approximation only gets better. This is not expressed in figure 7.1, here the top of the MA(1) decorrelation function lies at \( N = 4 \). The increase of the decorrelation function between \( N = 3 \) and \( N = 4 \) is caused by the fast increase of the \( WM \) of \( \mathcal{R}^N \).

For the AR(1) signal: the influence of the approximation of \( \rho^N \) by \( \hat{\rho}^N \) is large till \( \frac{N}{2} > |\tau_{\text{max}}| \). With \( \tau_{\text{max}} \) defined as the \( \tau \) where \( \rho[\tau] \approx 0 \) for \( \tau > \tau_{\text{max}} \). An exact value for \( \tau_{\text{max}} \) is hard to give, but it is for sure that the \( \tau_{\text{max}} \) for the AR(1) signal is larger than that of the MA(1) signal (here \( \tau_{\text{max}} = 1 \)). This explains why the top of the decorrelation function of the AR(1) signal is higher and appears in a later stage (for \( N = 6 \)).

3. A part where \( WM \) decreases: The influences of point 2 lose their power. The \( WM \) of \( \mathcal{R}^N \) reaches its maximum and \( \frac{N}{2} \) exceeds \( \tau_{\text{max}} \). As described by [GRA72] and [DAV79], for \( N \to \infty \) the matrices \( \mathcal{R}^N \) and \( \mathcal{R}^N \) are asymptotically equivalent, implying perfect decorrelation for \( N \to \infty \).

Figure 7.1 shows the remarkable fact that the decorrelation of the MA(1) signal shows better performance than that for the AR(1) signal. Figure 6.3, the decorrelation function of the BOP decorrelation part shows the opposite. This is explained by the fact that the \( \tau_{\text{max}} \) of the MA(1) signal is reached earlier than the \( \tau_{\text{max}} \) of the AR(1) signal.

About the decorrelation quality of the FDAF algorithm can be said that, except for the (small) first part of the decorrelation function, the quality becomes better with an increasing \( N \). A statement about the decorrelation quality of the BFDAF algorithm will be given after the examination of the windowing effect, which will be treated in the next section.

7.3 Windowing effect

For the BOP algorithm the dimension of the decorrelation matrix is \( B \times B \), for BFDAF \( M \times M \), with \( M \geq N + B - 1 \). At first side it seems that BFDAF uses a much larger decorrelation matrix, but this is not totally true. For BOP every element of the matrix \((\mathcal{R}^B)^{-1}\) is used to perform decorrelation. For BFDAF only the last \( B \) columns of \((\mathcal{R}^M)^{-1}\) are used, caused by the windowing with \((0 \, 1^B)^T\). Because the decorrelation matrix is the inverse of \( \mathcal{R}^M[kB] \), it can not be said that only \( B \) or \( N \) samples of \( \hat{\rho}^M \) are used for decorrelation. The influence of the windowing effect on the decorrelation quality can not be explained easily with mathematics. In this section the windowing effect will be determined on basis of simulation results.

When \( M \) is constant, the windowing effect can be examined by varying \( B \). This will be examined by the average the behaviour of the BFDAF decorrelation part, which is implemented in SPOX as follows:

\[
\frac{1}{B^{M}} \left\{ \mathbf{J}^N \begin{pmatrix} \mathbf{I}^N & \mathbf{0}_{N,M-N} \end{pmatrix} (\mathcal{R}^M[kB])(\mathcal{R}^M)^{-1} \begin{pmatrix} 0^{M-B,B} \\ \mathbf{I}_B \end{pmatrix} (\mathbf{X}^{N,B}[kB])^T \right\}
\]

(7.10)

As input signals, signals generated by the MA(1) and AR(1) models are used. The model parameters are \(|a| = 0.8182 \) and \( \sigma^2_n = 1 \). Caused by the stationary and ergodic nature
of the input signals the ensemble average $\mathcal{E}\{\cdot\}$ can be replaced by a time average $\mathcal{A}\{\cdot\}$. The time average is approximated by an average over 10,000 iterations.

The estimate of the autocorrelation function $\hat{\rho}$ used as basis for the creation of $\tilde{\rho}$ is supposed to be perfect. Due to the stationary properties of the input signals $\tilde{\mathcal{R}}^M[kB]$ is replaced by $\tilde{\mathcal{R}}^M$.

The previous section shows that for a circulant matrix with large dimension the influence of the circulant approximation is almost negligible. For AR(1) and MA(1) signal models it can be seen in figure 7.1 that a dimension of 256 is "large".

So when in the simulation of the BFDAF decorrelation part $M$ (dimension of circulant matrix) is chosen 256, it is to be expected that without the without the windowing effect the decorrelation is almost perfect.

In other words a derivation from the situation of "perfect" decorrelation will be mainly caused by the windowing effect. Mainly, because $B$ does not only affect the windowing effect but also the input signal matrix $\mathcal{X}^{N,B}$ (and hence $\tilde{\mathcal{X}}^M$). To measure the influence of a changing dimension of $\mathcal{X}^{N,B}$, the simulation is repeated twice, for $N = 8$ and $N = 100$. But there were no important differences.

The influence of the windowing effect is depicted in figure 7.2, starting at $B = 1$. As measurement for decorrelation quality the $ER$ is used (the use of $WM$ shows the same result).

At first sight it seems that the influence of $B$ on the BFDAF decorrelation part is the same as the influence of $B$ on the BOP decorrelation part. But this is not true. Without decorrelation the $ER$ is about 100. Without the windowing effect and with decorrelation performed by a circulant matrix of dimension 256 the $ER$ is about 1. With windowing effect the $ER$ is at worst 2.4. So the increase in $ER$ is at worst only 1.2 out of a 100.

Figure 7.2: eigenvalue ratio of BFDAF decorrelation part, as function of $B$
From this it can be concluded that $B$ and hence the windowing effect has almost no effect on the decorrelation quality. The decorrelation quality of the BFDAF algorithm depends mainly on influence of the circulant approximation. In the area where BFDAF is interesting, large adaptive filters, this influence is negligible and therefore a good decorrelation quality will be achieved.

### 7.4 Convergence of the BFDAF algorithm

In section 7.3 the influence of $B$, the block length is only examined for the decorrelation part. The influence of the circulant approximation is only examined for the FDAF algorithm. In this section the influence of the windowing and the circulant approximation will on the convergence properties of the BFDAF algorithm will be determined.

#### Influence of block length on BFDAF algorithm

In section 7.3 is shown that the influence of the block length $B$ on the decorrelation part is mainly expressed in the windowing effect. It is also shown that with an increasing $B$ the decorrelation quality only improves a little.

For the simulation of the whole BFDAF algorithm, the same parameters are used as with the simulation of section 7.3. For the input signal model, $\alpha = 0.8182$ and $\sigma_n^2 = 1$. For the adaptive filter, $M = 256$ and $N = 100$. Figure 7.3 shows the influence of $B$ on the convergence properties. It is to be seen that an increasing $B$ has a negative influence on the rate of convergence. This can be explained by the contradictorily influences of an increasing $B$, as mentioned for BOP in section 6.6. The difference with BOP is now that with an increasing $B$ the decorrelation quality improves only a little, not enough to beat the negative influences of an increasing $B$.

![Figure 7.3: convergence of the BFDAF algorithm, as function of $B$](image-url)
Influence of DFT length $M$ on BFDAF algorithm

Figure 7.1 shows that when the dimension of the (circulant) decorrelation matrix is small, the decorrelation quality strongly deviates from the ideal case. The examination of the properties of the BFDAF algorithm in this area will be interesting. For the input signals used in figure 7.1 this area lies between $M = 3$ and $M = 20$. Caused by constrains of SPOX, $M$, the FFT length must be at least 16 and must be increased by a power of two. So the input signals used in figure 7.1 are not useful for this examination.

Therefore an AR(12) signal is used as input signal $z$. The $W_M$ of the input signal and the $W_M$ of $(\hat{R}^M)^{-1}R^M$ is given in appendix D. The results are shown in figure 7.4.

For $M = 16$, the circulant approximation is almost at worst (see the top at dimension 16 in figure D.2). For $M = 32$, the result is almost equal to the ideal case.

![Figure 7.4](image)

*Figure 7.4: convergence of the BFDAF algorithm, as function of $M$*

It can be concluded that with an increasing $M$, the convergence properties gets better caused by the better approximation of the circulant matrix. When $B$ increases the decorrelation quality increases only a little, the negative influence of an increasing convergence rate dominates, the convergence properties get worse.
7.5 BOP versus BFDAF

In this chapter and the previous chapter the influences of different parameters (like \( M \), \( N \) and \( B \)) on the BOP and BFDAF algorithm are examined. It is shown that the parameters have a different influence on both algorithms. The differences are:

The influence of \( M \):

- **BOP**: no influence (of course \( M \geq N + B - 1 \)).
- **BFDAF**: the circulant approximation gets better with an increasing \( M \), resulting in a better decorrelation quality and better convergence properties.

The influence of \( B \):

- **BOP**: the decorrelation quality increases with an increasing \( B \), resulting in better convergence properties.
- **BFDAF**: has almost no influence on the decorrelation quality.

The influence of \( N \) is not mentioned, this parameter is equal to the adaptive filter length which is for both filters the same.

With this the comparison between BOP and BFDAF is ended, a general statement about the choice of BOP or BFDAF will be given in the next chapter, the conclusions.
Chapter 8

Conclusions

In this report several methods of decorrelation are presented, performed in time- and frequency-domain. The relation between time- and frequency-domain decorrelation is shown on the basis of the Block Orthogonal Projection (BOP) algorithm and the Block Frequency Domain Adaptive Filter (BFDAF) algorithm.

For the BOP algorithm the dimension of the decorrelation matrix is equal to the block length $B$. A decorrelation quality comparable with that of RLS can be reached with a decorrelation matrix with a dimension far more smaller than the filter length. Till a certain $B$, the decorrelation quality increases very fast. A further increase of $B$ leads only to higher complexity, not to sufficient better decorrelation quality. This certain $B$ is called the optimal $B$.

The optimum value of $B$ depends on the input statistics. For input signals generated by the MA(1) and AR(1) signal models, the optimal $B$ is determined as the $B$ where the eigenvalue ratio of the decorrelated signal, $ER(\tilde{I}^N)$, is 3. The eigenvalue ratio of $\tilde{I}^N$ can be estimated by:

$$ER(\tilde{I}^N) \approx \frac{ER(\mathcal{R}^N)}{ER(\mathcal{R}^B)}$$

To determine the optimal $B$ for arbitrary input signals, further examination is required.

For the BFDAF algorithm the decorrelation quality mainly depends on the influence of the circulant approximation. With an increasing DFT length $M$, the circulant approximation becomes better and better, resulting in better decorrelation quality and better convergence properties.

Decorrelation in the BFDAF algorithm is not performed with the whole decorrelation matrix, only the last $B$ columns are used. With an increasing $B$, the decorrelation quality increases only a little, not enough to be expressed in better convergence properties.

In general a frequency-domain implementation of a small adaptive filter is not useful. For small filter lengths the implementation can be done efficiently in the time-domain. Frequency-domain adaptive filters are interesting for large adaptive filters. For these filters like BFDAF, decorrelation can be performed efficient in the frequency-domain, saving a lot of complexity. For BFDAF a large filter length implies a large decorrelation matrix, implying a negligible influence of the circulant approximation.

In conclusion can be said that in the area where frequency-domain adaptive are interesting, BFDAF is a more efficient alternative than the BOP algorithm with equivalent convergence properties.
Bibliography


Appendix A

Power normalization without divisions

In this appendix is shown that the following equation

\[
\hat{\sigma}_x^{-2}[(k + 1)B] = \frac{1}{(1 - \beta)\hat{\sigma}_x^2[kB] + \beta \frac{1}{N}(x^N[kB])^t x^N[kB]}
\]  

(A.1)

can be approximated, for small \(\beta\), by an equation that needs no divisions. For this we rewrite the equation as follows:

\[
\hat{\sigma}_x^{-2}[(k + 1)B] = \frac{\frac{1}{1 - \beta} \hat{\sigma}_x^2[kB]}{1 + \frac{\beta}{1 - \beta} \hat{\sigma}_x^{-2}[kB] \frac{1}{N}(x^N[kB])^t x^N[kB]}
\]

\[
\approx \frac{1 - \beta}{1 - \beta} \hat{\sigma}_x^{-2}[kB] - \frac{\beta}{(1 - \beta)^2} (\hat{\sigma}_x^{-2}[kB])^2 \frac{1}{N}(x^N[kB])^t x^N[kB]
\]

\[
\approx (1 + \beta)\hat{\sigma}_x^{-2}[kB] - \beta(\hat{\sigma}_x^{-2}[kB])^2 \frac{1}{N}(x^N[kB])^t x^N[kB]
\]  

(A.2)

Note that the division \(\frac{1}{N}\) can be combined in one constant with \(\beta\) as \(\frac{\beta}{N}\), and no divisions are required.
Appendix B

Simulation results of the BOP algorithm 1

The simulations are carried out with a SPOX implementation of the BOP algorithm. The next signal models are used for generating the input:

- The moving average model of order 1, MA(1):
  \[ x[k] = \frac{1}{\sqrt{1 + a^2}} \cdot (n[k] + an[k - 1]) \] (B.1)

- The auto regressive signal model of order 1, AR(1):
  \[ x[k] = (\sqrt{1 - a^2}) \cdot n[k] + az[k - 1] \] (B.2)

With \(|a| = 0.8182\) and \(\sigma_n^2 = 1\). The \(ER_{\text{max}}\) of both signals \(\approx 100\). The signal \(s\), the speakers signal in the BOP system is chosen as a white noise signal with \(\sigma_s^2 = 0.001\). The echo pad \(h[k]\) is equal to \(h[k] = \delta[k - \frac{N}{2}]\).

The simulations are done for different filter lengths, \(N = 100, N = 20, N = 8\). For each filter length the simulations are performed with different block lengths. The selected \(B_s\) are the smallest possible \(B\) in the range \(z - 0.5 \leq x < z + 0.5 \text{ dB}\) of the corresponding decorrelation function. The value on the vertical axis of the figures B.1...B.3 is the relative misadjustment \(J[k]\), expressed in dB.

The \(\alpha\) is chosen in such a way that the final misadjustment \(10 \log J = -10 \text{ dB}\).
Figure B.1: convergence of BOP algorithm for filter of length 100 as function from ER.
Figure B.2: convergence of BOP algorithm for filter of length 20 as function from ER
Figure B.3: convergence of BOP algorithm for filter of length 8 as function from ER
Appendix C

Simulation results of the BOP algorithm 2

The simulations are carried out with a SPOX implementation of the BOP algorithm. The next signal models are used for generating the input:

- The moving average model of order 1, MA(1):
  \[ x[k] = \frac{1}{\sqrt{1 + \alpha^2}} \cdot (n[k] + an[k - 1]) \]  
  (C.1)

- The auto regressive signal model of order 1, AR(1):
  \[ x[k] = (\sqrt{1 - \alpha^2}) \cdot n[k] + az[k - 1] \]  
  (C.2)

With |\(a| = 0.8182 and \(\sigma_n^2 = 1\). The \(ER_{max}\) of both signals \(\approx 100\). The signal \(s\), the speakers signal in the BOP system is chosen as a white noise signal with \(\sigma_s^2 = 0.001\). The echo pad \(h[k]\) is equal to \(h[k] = \delta[k]\).

The simulations are done for different filter lengths, \(N = 100\), \(N = 20\), \(N = 8\). For each filter length the simulations are performed with different block lengths. The selected block lengths are the smallest possible \(B\) in the range \(z - 0.5 \leq z < z + 0.5\) dB of the corresponding decorrelation function. The value on the vertical-axis of the figures C.1, . . . , C.3 is the relative misadjustment \(\hat{J}[k]\), expressed in dB.

\(\alpha = 7e - 4\) for all simulations.
Figure C.1: convergence of BOP algorithm for filter of length 100 as function from ER
Figure C.2: convergence of BOP algorithm for filter of length 20 as function from ER
Figure C.3: convergence of BOP algorithm for filter of length 8 as function from ER
Appendix D

Decorrelation of a male voice

The characteristics of a male voice can be modelled by an AR(12) model. For this signal, figure D.1 of this appendix shows how the $WM$ (defined in equation D.1) of the input signal autocorrelation matrix $\mathcal{R}^N$ increases with an increasing $N$. Figure D.2 shows the decorrelation function $WM((\mathcal{R}^N)^{-1}\mathcal{R}^N)$ as function of $N$.

Definition of $WM$:

\[
WM = \frac{1}{N} \sum_{i=0}^{N-1} \left( \max(\frac{\lambda_i}{\overline{\lambda}}, \frac{\overline{\lambda}}{\lambda_i}) - 1 \right) \tag{D.1}
\]

with

\[
\overline{\lambda} = \frac{1}{N} \sum_{i=0}^{N-1} \lambda_i \tag{D.2}
\]

$WM$ stands for whiteness measure. The decorrelation quality is determined on basis of all eigenvalues. In case of a white noise signal, all $\lambda_i$ are the same. The corresponding $WM$ is 0.
Figure D.1: the $WM$ of $\mathcal{R}^N$, as function of $N$.

Figure D.2: the $WM$ of $(\mathcal{R}^N)^{-1}\mathcal{R}^N$, as function of $N$. 

60