MASTER

Subband coding using IIR filters

van Leest, A.J.

Award date:
1996

Link to publication
Subband coding using IIR filters

door

A.J. van Leest

ESP-14-96
Abstract

Subband filtering is a well-known technique with many applications. These filter banks are usually based on FIR filters. However, it is well-known that IIR filters give a larger freedom than FIR filters when approximating a transfer given a certain complexity.

This report considers filter banks which use stable and causal IIR analysis and synthesis filters. The start of this study was a recent publication on perfect reconstruction two-channel filter banks. This two-channel filter bank showed a certain asymmetry in its amplitude transfer. It is found that more symmetrical solutions exist, but at the expense of lower stopband attenuation.

The concept concept of the two-channel filter banks can unfortunately not be extended to the more general case of $M$-channel filter banks. Therefore, a new filter bank is proposed based on IIR filters. In this type of filter bank the $2M$ analysis filters are derived from a prototype filter by exponential modulation. This filter bank has not the perfect reconstruction property. However, the phase distortion, magnitude distortion and the aliasing can be made very small with a very low complexity. This filter bank is not maximally decimated but oversampled by a factor 2. However, the overall complexity of the filter bank is still smaller than that of a filter bank using an FIR prototype with comparable amplitude transfer.
Contents

1 Introduction 2

2 Preliminaries 4
   2.1 The allpass filter 4
   2.2 A halfband filter 5
   2.3 The polyphase representation 6
   2.4 The M-channel filter bank 8

3 Two-channel IIR filter banks 9
   3.1 A two-channel IIR filter bank with the perfect reconstruction property 9
   3.2 Two-channel IIR filter banks which introduce phase distortion 13
   3.3 Almost perfect reconstruction with a two-channel filter bank 14
   3.4 Comparison of the two-channel filter banks 16

4 2M-channel modulated filter bank based on IIR filters 19
   4.1 The prototype 19
   4.2 The analysis bank 20
   4.3 The synthesis bank 21
      4.3.1 An aliasing and amplitude distortion free filter bank 24
      4.3.2 Approximation of the aliasing and amplitude distortion free filter bank 25
      4.3.3 Almost perfect reconstruction 25
   4.4 The implementation of the modulated filter bank 27
   4.5 The design of the filter bank 30
   4.6 An FIR modulated filter bank 33

5 Examples of the Matlab functions 36

6 Concluding remarks 37
   6.1 Discussion 38

A The Matlab functions 41
   A.1 Elliptic filter design 41
   A.2 Allpass filter design 42
   A.3 Almost perfect filter bank design 56
   A.4 Simulations of the filter bank design 58
   A.5 Distortion function of the filter bank 60
   A.6 The peak aliasing distortion of the filter bank 62
Chapter 1

Introduction

In subband coders, the input signal is split into several frequency bands by the analysis filters. Then these subband signals are decimated, coded and transmitted, after which they are processed in the synthesis bank by interpolating each signal, filtering and adding the filtered signals. A common requirement in most applications is that the reconstructed signal should be as close to the input signal as possible. In general, the reconstructed signal suffers from four types of errors: aliasing error, amplitude distortion, phase distortion and quantization errors.

Subband coding is a technique which has a lot of applications. For example subband coding of speech and image signals. In these applications compression of the signals often is the main goal. See e.g. [1], [2], [3] and [4].

Usually, filters used in subband coders are based on finite impulse response (FIR) filters. Infinite impulse response (IIR) filters however achieve better magnitude responses compared to FIR filters with the same complexity. So the use of IIR filters will result in less data storage, less computational requirements and less signal delays. Consequently, if this subband coder is implemented in a chip this will result in less surface and less power consumption. Therefore the question arises if it is possible to design subband coders based on IIR filters so that the complexity of the subband coder decreases.

The most difficult part is stability of the subband coder based on IIR filters. A stable analysis filter bank will not automatically lead to a stable synthesis filter bank. In the case of FIR there are several techniques to achieve stability and still have sufficient freedom to specify the filter characteristics.

In the literature several filter banks are proposed which use IIR filters [5], [6], [7]. Although these analysis filters are stable, the synthesis filters often are unstable. Stable and causal filters are desired in case of real-time applications. Therefore an analysis of subband coders which use stable and causal IIR filters in the analysis and the synthesis filter bank are the subject in this study.

First, some preliminaries are discussed to recall some properties. Subsequently, three different types of two-channel IIR filter banks will be introduced. These are consecutively a two-channel IIR filter bank which

- achieves perfect reconstruction;
- introduces phase distortion;
- achieves almost perfect reconstruction.

These filter banks will be compared. Furthermore, a new filter bank is proposed based on IIR filters. In this type of filter bank the $2M$ analysis filters are derived from a prototype
filter by exponential modulation. This filter bank has not the perfect reconstruction property. However, the phase distortion, magnitude distortion and the aliasing can be made very small with a very low complexity. This filter bank is not maximally decimated but oversampled by a factor 2. However, the overall complexity of the filter bank is still smaller than that of a filter bank using an FIR prototype with comparable amplitude transfer.
Chapter 2

Preliminaries

2.1 The allpass filter

Allpass filters play an important role in this report. Therefore some properties of the allpass filter will be listed here. For a more extensive treatment see e.g. [1].

A transfer function $H(z)$ is said to be allpass if

$$|H(e^{j\theta})| = c, \quad \text{for all } \theta.$$ 

For example, the simplest rational transfer function is given by

$$H(z) = \frac{a^* + z^{-1}}{1 + az^{-1}}$$

representing a first-order allpass filter.

The poles and zeros of an allpass filter occur in reciprocal conjugate pairs. In other words, if $\alpha$ is a pole, then its reciprocal conjugate $1/\alpha^*$ is a zero. Thus, when the allpass filter is stable, that is the poles lie inside the unit circle, the zeros lie outside the unit circle.

An $N$th order allpass filter with real coefficients has the following form:

$$H(z) = \frac{\sum_{k=0}^{N} a_N z^{-k}}{\sum_{k=0}^{N} a_k z^{-k}}.$$ 

Substituting $z = 1$ and $z = -1$ into $H(z)$, which corresponds with the phase responses at 0 and $\pi$ respectively, yield $H(1) = 1$ and $H(-1) = (-1)^N$ independent of the coefficients of the allpass filter.

Let $H(z)$ be an $N$th order allpass filter. If $H(z)$ has all poles inside the unit circle, then the (unwrapped) phase response $\phi(\theta)$ of $H(z)$ is monotone decreasing, and spans a range of $\pi N$ as $\theta$ increases from 0 to $\pi$.

The phase response of an $N$th order allpass filter with real coefficients can be expressed as

$$\phi(\theta) = -N\theta + 2\arctan \frac{\sum_{k=0}^{N} a_k \sin(k\theta)}{\sum_{k=0}^{N} a_k \cos(k\theta)}. \quad (2.1)$$
2.2 A halfband filter

A halfband filter that is frequently used throughout this report has the following form:

\[ H(z) = \frac{1}{2} \left[ A(z^2) + z^{-1}B(z^2) \right] = \frac{1}{2} B(z^2) \left[ A(z^2)/B(z^2) + z^{-1} \right]. \]  

(2.2)

This transfer function \( H(z) \) is an ideal lowpass filter if the ratio \( A(z)/B(z) \) has the following magnitude and phase responses

\[ |A(e^{2j\theta})| = |B(e^{2j\theta})| = 1, \quad \text{for all } \theta. \]

\[ \angle \left( A(e^{2j\theta})/B(e^{2j\theta}) \right) = \begin{cases} -\theta & \text{for } \theta \in [0, \pi/2] \\ -\theta \pm \pi & \text{for } \theta \in (\pi/2, \pi] \end{cases}. \]  

(2.3)

A special case of (2.2) arises if \( A(z) \) is a delay, thus

\[ H(z) = \frac{1}{2} \left[ z^{-2N} + z^{-1}B(z^2) \right]. \]

The filter \( H(z) \) is an ideal lowpass filter if \( B(z) \) has the following magnitude and phase responses:

\[ |B(e^{2j\theta})| = 1, \quad \text{for all } \theta. \]

\[ \angle B(e^{2j\theta}) = \begin{cases} (-2N + 1)\theta & \text{for } \theta \in [0, \pi/2] \\ (-2N + 1)\theta \pm \pi & \text{for } \theta \in (\pi/2, \pi] \end{cases}. \]  

(2.4)

Then

\[ \theta \in [0, \pi/2] : \quad H(e^{j\theta}) = \frac{1}{2} \left( e^{-2j\theta N} + e^{-j\theta}e^{j(-2N+1)\theta} \right) = e^{-2j\theta N} \]

\[ \theta \in (\pi/2, \pi] : \quad H(e^{j\theta}) = \frac{1}{2} \left( e^{-2j\theta N} + e^{-j\theta}e^{j(-2N+1)\theta \pm j\pi} \right) = 0. \]

An allpass filter \( B(z) \) has the property that \( |B(e^{2j\theta})| = 1 \), but cannot have the phase response given in (2.4). This is due to the discontinuity and the fact that \( B(j^2) = (-1)^L \), where \( L \) is the order of \( B(z) \). This means that \( \angle B(e^{2j\theta}) = -L\pi \) for \( \theta = \pi/2 \) independent of the coefficients of \( B(z) \). Therefore (2.4) can only be approximated.

The phase response of a causal stable allpass filter \( B(z^2) \) spans a range of \( 2L\pi \) when \( \theta \) spans a range of \( \pi \). From (2.4) it follows that \( B(z^2) \) has to span a range of \( 2N\pi \) or \( 2(N - 1)\pi \). Since \( B(z) \) has to be a causal stable allpass filter it follows that \( L = N \) or \( L = N - 1 \).

Let \( B(z) \) be an allpass filter, then the phase response of \( B(z^2) \) is point-symmetrical around \( (\theta, \zeta) = (\pi/2, -N\pi) \). (Recall that a function \( f(x) \) is point-symmetrical in the origin if \( f(x) = -f(-x) \).) This can be seen as follows:

Let \( \phi(\theta) \) be the phase response of \( B(z) \), then \( \phi(2\theta) \) is the phase response of \( B(z^2) \). Let \( \phi_t(2\theta) \) be the translated version of \( \phi(2\theta) \) towards the origin, thus (cf. (2.1))

\[ \phi_t(2\theta) = \phi(2(\theta + \pi/2)) + N\pi = -2N(\theta + \pi/2) + 2 \arctan \frac{\sum_{k=0}^{N} b_k \sin(2k(\theta + \pi/2))}{\sum_{k=0}^{N} b_k \cos(2k(\theta + \pi/2))} + N\pi \]

\[ = -2N\theta + 2 \arctan \frac{\sum_{k=0}^{N} (-1)^k b_k \sin(2k\theta)}{\sum_{k=0}^{N} (-1)^k b_k \cos(2k\theta)}. \]
It follows that \( \phi_{c}(2\theta) = -\phi_{c}(-2\theta) \) and so \( \phi(2\theta) \) is point symmetrical around \( (\theta, \zeta) = (\pi/2, -N\pi) \).

If an allpass filter \( B(z) \) is designed such that the slope of its phase response is approximately \( (-N + 1/2) \) between \( 0 \) and \( \theta_p \), where \( 0 < \theta_p < \pi \), then the typical phase response of the allpass filter \( B(z^2) \) looks like the phase response which is shown in figure 2.1. From this figure it follows that the phase response of \( B(z^2) \) will show a transition of \( \pi \). This is a direct result of the point-symmetry around \( (\theta, \zeta) = (\pi/2, -N\pi) \). Thus, in order to approximate (2.4), only the slope of the phase response of \( B(z) \) between \( 0 \) and \( \theta_p \) needs to be \( (-N + 1/2) \). This will result in a filter \( H(z) \) having a bandwidth of \( \pi/2 \) and a stopband which starts at \( \theta = \pi - \theta_p/2 \).

### 2.3 The polyphase representation

Let \( H(z) = \sum_{n=-\infty}^{\infty} h(n)z^{-n} \) be the transfer function of a filter. Then \( H(z) \) can always be decomposed as

\[
H(z) = \sum_{n=-\infty}^{\infty} h(nM)z^{-nM} + \sum_{n=-\infty}^{\infty} h(nM+1)z^{-nM-1} + \ldots + z^{-(M-1)} \sum_{n=-\infty}^{\infty} h(nM+M-1)z^{-nM}.
\]

This can be compactly written as

\[
H(z) = \sum_{t=0}^{M-1} z^{-t} E_t(z^M)
\]

where

\[
E_t(z) = \sum_{n=-\infty}^{\infty} e_t(n)z^{-n}
\]
with

\[ e_\ell(n) = h(Mn + \ell), \quad 0 \leq \ell \leq M - 1. \]

The right-hand side in (2.5) is called the type 1 polyphase representation of \( H(z) \) and the functions \( E_\ell(z) \) are called the polyphase components. It is also possible to write \( H(z) \) in the so-called type 2 polyphase representation

\[
H(z) = \sum_{\ell=0}^{M-1} z^{-(M-1-\ell)} R_\ell(z^M)
\]

where

\[ R_\ell(z) = E_{M-1-\ell}(z). \]

Let \( \{ H_k(z) \mid 0 \leq k \leq M - 1 \} \) be a set of transfer functions. Each \( H_k(z) \) can be expressed in the type 1 polyphase representation according to

\[
H_k(z) = \sum_{\ell=0}^{M-1} z^{-\ell} E_{k\ell}(z^M), \quad 0 \leq k \leq M - 1.
\]

In matrix notation this reads

\[
\begin{pmatrix}
H_0(z) \\
\vdots \\
H_{M-1}(z)
\end{pmatrix} =
\begin{pmatrix}
E_{00}(z^M) & E_{01}(z^M) & \cdots & E_{0,M-1}(z^M) \\
E_{10}(z^M) & E_{11}(z^M) & \cdots & E_{1,M-1}(z^M) \\
\vdots & \vdots & \ddots & \vdots \\
E_{M-1,0}(z^M) & E_{M-1,1}(z^M) & \cdots & E_{M-1,M-1}(z^M)
\end{pmatrix}
\begin{pmatrix}
1 \\
z^{-1} \\
\vdots \\
z^{-(M-1)}
\end{pmatrix}
\]

or in shorthand notation

\[
h(z) = E(z^M)e(z)
\]

where

\[
E(z) =
\begin{pmatrix}
E_{00}(z) & E_{01}(z) & \cdots & E_{0,M-1}(z) \\
E_{10}(z) & E_{11}(z) & \cdots & E_{1,M-1}(z) \\
\vdots & \vdots & \ddots & \vdots \\
E_{M-1,0}(z) & E_{M-1,1}(z) & \cdots & E_{M-1,M-1}(z)
\end{pmatrix},
\]

\[
h(z) =
\begin{pmatrix}
H_0(z) \\
\vdots \\
H_{M-1}(z)
\end{pmatrix}
\quad \text{and} \quad
e(z) =
\begin{pmatrix}
1 \\
z^{-1} \\
\vdots \\
z^{-(M-1)}
\end{pmatrix}.
\]

The matrix \( E(z) \) is called the \( M \times M \) type 1 polyphase matrix.

Similarly by using \( H_k(z) = \sum_{\ell=0}^{M-1} z^{-(M-1-\ell)} R_{k\ell}(z^M) \) this can be expressed in the matrix notation

\[
h^T(z) = f(z)R(z^M)
\]

where \( f(z) = [ z^{-(M-1)} \quad z^{-(M-2)} \cdots \quad 1 ] \) and

\[
R(z) =
\begin{pmatrix}
R_{00}(z) & \cdots & R_{0,M-1}(z) \\
R_{10}(z) & \cdots & R_{1,M-1}(z) \\
\vdots & \ddots & \vdots \\
R_{M-1,0}(z) & \cdots & R_{M-1,M-1}(z)
\end{pmatrix},
\]

The matrix \( R(z) \) is called the type 2 polyphase matrix.
2.4 The $M$-channel filter bank

The purpose of this section is not to explain a filter bank, but only to list some properties which are used in this report, for more details see e.g. [1].

An $M$-channel maximally decimated filter bank is shown in figure 2.2. In an $M$-channel filter bank a signal $x(n)$ is decomposed in $M$ subbands. This is done by the analysis filters $H_k(z)$. The analysis filters are followed by downsamplers yielding the subband signals. Then these output signals are recombined to a signal $\hat{x}(n)$. This recombination is done by use of upsamplers followed by the synthesis filters $F_k(z)$. A filter bank achieves perfect reconstruction if $\hat{x}(n) = cx(n - k)$ where $c \neq 0$.

Let $E(z)$ and $R(z)$ be the polyphase matrices of the analysis and the synthesis filters respectively. Then it can be shown that if $R(z)E(z) = cz^{-m}I$, where $I$ is the identity matrix, for some integer $m$, then the filter bank has the perfect reconstruction property. In the maximally decimated case, perfect reconstruction is equivalent to biorthogonality. If perfect reconstruction is not achieved then the reconstructed signal $\hat{x}(n)$ differs from $x(n)$ due to three reasons: aliasing, amplitude distortion and phase distortion. Aliasing is due to the downsamplers.

An $M$-fold downsampler takes an input sequence $x(n)$ and produces the output sequence $y(n) = x(Mn)$ where $M$ is an integer. The z-transform of $y(n)$ can be written as

$$Y(z) = \frac{1}{M} \sum_{k=0}^{M-1} X(z^{1/M}W^k)$$

where $W = \exp(-j2\pi/M)$.

An $M$-fold upsampler takes an input sequence $x(n)$ and produces the output sequence

$$y(n) = \begin{cases} x(n/M) & \text{if } n \text{ is integer-multiple of } M \\ 0 & \text{otherwise} \end{cases}$$

The z-transform of $y(n)$ can be written as $Y(z) = X(z^M)$.

---

Figure 2.2: The $M$-channel maximally decimated filter bank.
3.1 A two-channel IIR filter bank with the perfect reconstruction property

In [8] a new class of two-channel biorthogonal filter banks is proposed. A subclass is the two-channel filter bank with causal stable IIR filters. In figure 3.1 a two-channel filter bank is shown.

\[ H_0(z) = \frac{1}{2} \left( z^{-2N} + z^{-1} \beta(z^2) \right) \]
\[ H_1(z) = -\beta(z^2)H_0(z) + z^{-4N+1} \]
\[ F_0(z) = -H_1(-z) \]
\[ F_1(z) = H_0(-z) \]

(3.1) (3.2)

where \( \beta(z) \) is a stable \( N \)th order allpass filter.

With these filters the filter bank has the perfect reconstruction property. This can be seen as follows

\[ E(z) = \begin{pmatrix} \frac{1}{2}z^{-N} & \beta(z) \\ -\frac{1}{2}\beta(z) & 1 \end{pmatrix} \begin{pmatrix} 0 & z^{-N} \beta(z) \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{2}z^{-2N+1} \beta(z) \\ -\frac{1}{2}z^{-N}\beta(z) - \frac{1}{2}\beta^2(z) + z^{-2N+1} \end{pmatrix} \]

\[ = \begin{pmatrix} \frac{1}{2}z^{-N} & \frac{1}{2}\beta(z) \\ -\frac{1}{2}z^{-N}\beta(z) & -\frac{1}{2}\beta^2(z) + z^{-2N+1} \end{pmatrix} \]

where \( E(z) \) is the polyphase matrix of the analysis bank. The determinant of this polyphase matrix is \( \det[E(z)] = \frac{1}{2}z^{-3N+1} \). The polyphase matrix of the synthesis bank \( R(z) \) becomes
now the delayed inverse of the polyphase matrix of the analysis bank $E(z)$:

$$
R(z) = \begin{pmatrix}
  z^{-2N+1} & -\beta(z) \\
  0 & z^{-N}
\end{pmatrix}
\begin{pmatrix}
  1 & 0 \\
  \frac{1}{2} \beta(z) & \frac{1}{2}
\end{pmatrix}
= \begin{pmatrix}
  z^{-2N+1} - \frac{1}{2} \beta^2(z) & -\frac{1}{2} \beta(z) \\
  \frac{1}{2} z^{-N} \beta(z) & \frac{1}{2} z^{-N}
\end{pmatrix}.
$$

The corresponding synthesis filters can be verified to have the form as in (3.1) and (3.2). Since $R(z)E(z) = \frac{1}{2} z^{-3N+1} I$ it follows that the filter bank has the perfect reconstruction property.

The very efficient implementation, shown in figure 3.2, and the perfect reconstruction property are advantages of this filter bank. A disadvantage can be the non-symmetry of the filters. Substituting $\theta = \pi/2$ into the expression for $H_1(e^{j\theta})$ and using the fact that $\beta(-1) = (-1)^N$ and $H_0(j) = \frac{1}{2} (-1)^N (1 - j)$ yields

$$
H_1(j) = -(-1)^N \frac{1}{2} (-1)^N (1 - j) + j = -\frac{1}{2} + \frac{3}{2} j. \tag{3.3}
$$

This means that $|H_1(e^{j\theta})|$ and $|F_0(e^{j\theta})|$ shows a bump of approximately 4 dB at $\theta = \pi/2$, independent of the specific choice of the allpass filter $\beta(z)$. As a result of this bump the passband will be wider than $\pi/2$ which presumably leads to more aliasing than in the case where a more symmetrical design is available.

Furthermore, the stopband attenuation of $H_1(z)$ and $F_0(z)$ is worse than the stopband attenuation of $H_0(z)$ and $F_1(z)$. In figure 3.3 an example is shown, where the order of $\beta(z)$ is 8. The stopband $\theta_s$ starts at $0.6\pi$.

Is it possible to make the filters $H_0(z)$, $H_1(z)$, $F_0(z)$ and $F_1(z)$ more symmetrical without a bump in the amplitude transfer while retaining the perfect reconstruction property? The answer is affirmative.

Consider (3.3). It can be seen that the complex parts in the middle have the same sign which is not desirable. A minus sign can be introduced by using an allpass filter $\alpha(z)$ of order $N - 1$ in $H_1(z)$ instead of the allpass filter $\beta(z)$ of order $N$. Notice that an order of $N + 1$ leads to instability of $\alpha(z)$. With this allpass filter $\alpha(z)$, $H_1(z)$ becomes now

$$
H_1(z) = -\alpha(z^2) H_0(z) + z^{-4N+1}.
$$

Substituting $z = j$ into $H_1(z)$ yields now

$$
H_1(j) = -(-1)^N \frac{1}{2} (-1)^N (1 - j) + j = -\frac{1}{2} + \frac{1}{2} j.
$$

Figure 3.2: The implementation of the filter bank.
Figure 3.3: The magnitude responses of the filters $H_0(z)$ and $H_1(z)$. The order of $\beta(z)$ is 6. The stopband $\theta_s$ starts at $0.6\pi$.

In figure 3.4 the result is shown. Here the orders of $\alpha(z)$ and $\beta(z)$ are 9 and 10 respectively. It is also possible to interchange the roles of the allpass filters $\alpha(z)$ and $\beta(z)$. Thus the order of $\beta(z)$ becomes $N - 1$ and the order of $\alpha(z)$ $N$. This leads to a better result, this is shown in figure 3.5.

Figure 3.4: The magnitude responses of the filters $H_0(z)$ and $H_1(z)$, where the allpass filter $\alpha(z)$ is used in the expression of $H_1(z)$.

A better performance can be obtained, if the allpass filter $\alpha(z)$ is designed given the phase response of $\beta(z)$, since the phase response of $H_0(z)$ is dependent on $\beta(z)$. Or even better $\alpha(z)$ is designed given the phase response of $H_0(z)$, since then the phase response of $\alpha(z)$ can
Figure 3.5: The magnitude responses of the filters $H_0(z)$ and $H_1(z)$, where the order of $\beta(z)$ is the lowest.

be designed such that the phase response of $-\alpha(z^2)H_0(z)$ in $H_1(z)$ is approximately a delay between 0 and $\pi/2$. The results are shown in figure 3.6 and 3.7 for both design methods. Notice that the latter design method leads to a sharper transition band of $H_1(z)$.

Figure 3.6: The magnitude responses of the filters $H_0(z)$ and $H_1(z)$, where the allpass filter $\alpha(z)$ is designed given the phase response of $\beta(z)$.

It is also possible to decrease or increase the order of the allpass filter $\alpha(z)$, but then the delay in $H_1(z)$ has to be adjusted. Notice that the order of the delay must be odd.

Comparing figure 3.3 with figure 3.6 and figure 3.7 leads to the following conclusions:

If the bump in the magnitude response of $H_1(z)$ and the presumable extra aliasing is not
significant, one can better use the allpass filter $\beta(z)$ in both filters $H_0(z)$ and $H_1(z)$ because of the better stopband attenuation.

If however this is significant, which usually is the case, one can better make use of an other allpass filter $\alpha(z)$, having an order higher than that of $\beta(z)$. The best result is obtained if $\alpha(z)$ is designed given the phase response of $H_0(z)$. A worse stopband attenuation is the price one has to pay.

3.2 Two-channel IIR filter banks which introduce phase distortion

Suitable IIR halfband filters are the well-known elliptic and Butterworth filters. These filters can be used in the two-channel IIR filter bank. Only the elliptic filters will be considered here, since these filters have better magnitude responses given a certain complexity.

Elliptic filters of odd order can be written as [1]

$$H_0(z) = A_0(z^2) + z^{-1}A_1(z^2)$$

where $A_0(z)$ and $A_1(z)$ are stable real allpass filters. So the polyphase components are allpass filters. By taking $H_1(z) = H_0(-z)$, $F_0(z) = H_0(z)$ and $F_1(z) = -H_1(z)$ the filter bank becomes as shown in figure 3.8. The distortion function $T(z)$ of the filter bank is

$$T(z) = \frac{\hat{X}(z)}{X(z)} = z^{-1}A_0(z^2)A_1(z^2). \quad (3.4)$$

Therefore this filter bank introduces only phase distortion which is governed by the phase responses of $A_0(z^2)$ and $A_1(z^2)$.

In figure 3.9 an example is shown where the analysis and the synthesis filters are elliptic. The order of $A_0(z)$ and $A_1(z)$ is 2. The stopband $\theta_s$ of $H_0(z)$ starts at $0.6\pi$. So these filters are very efficient.
3.3 Almost perfect reconstruction with a two-channel filter bank

The phase distortion introduced by the filter bank in the previous section is a disadvantage for certain applications. In [9] a filter bank is proposed which structure is identical to the structure of the filter bank from the previous section. The proposed filter bank is also based on allpass filters as shown in Figure 3.8, but achieves in contrast to the filter bank from the previous section almost perfect reconstruction.

To obtain a lowpass filter the phase difference of the allpass filters $A_N(z)$ and $B_M(z)$ must satisfy (cf.(2.3))

$$\theta_A(\theta) - \theta_B(\theta) = -\frac{1}{2}\theta, \quad (0 \leq \theta \leq 2\theta_p)$$

(3.5)
where $\theta_A$ and $\theta_B$ are the phase responses of $A_N(z)$ and $B_M(z)$ respectively, and $\theta_p$ the passband edge frequency. To force the distortion function $T(z)$ (eq.(3.4)) to have a linear phase characteristic, the sum of the phase responses of $A_N(z)$ and $B_M(z)$ must satisfy

$$\theta_A(\theta) + \theta_B(\theta) = -(N + M)\theta, \quad (0 \leq \theta \leq \pi)$$

(3.6)

where $N$ and $M$ are the orders of the allpass filters $A_N(z)$ and $B_M(z)$ respectively. From (3.5) and (3.6) it follows that the desired phase responses of $A_N(z)$ and $B_M(z)$ in the passband become

$$\theta_A(\theta) = -\left(\frac{N + M}{2} + \frac{1}{4}\right)\theta$$

and

$$\theta_B(\theta) = -\left(\frac{N + M}{2} - \frac{1}{4}\right)\theta$$

(3.7)

From (2.3) with $A(z) = A_N(z)$ and $B(z) = B_M(z)$ it follows that $N$ and $M$ must satisfy $N = M$ or $N = M + 1$.

If $A_N(z)$ and $B_M(z)$ were designed as the desired phase responses in (3.7) then the phase error will not be equiripple, but will show a large ripple in the transition band. Therefore first $A_N(z)$ is designed. Then $B_M(z)$ is designed given the phase response of $A_N(z)$ such that the distortion function (3.4) has an equiripple phase error between 0 and $\pi$. Then $A_N(z)$ is designed again given the phase response of $B_M(z)$ such that the phase difference of $A_N(z)$ and $B_M(z)$ is equiripple. Thus will be approximately $-\theta/2$ between 0 and $2\theta_p$. After this $B_M(z)$ is designed again given the phase response of $A_N(z)$ such that the distortion function
(3.4) has an equiripple phase error between $0$ and $\pi$. This procedure can be repeated to gain better performances.

In figure 3.10 an example is shown, where the order of $A_N(z)$ and $B_M(z)$ is $10$. The stopband of $H_0(z)$ begins at $0.6\pi$. Here the above-mentioned procedure is repeated twice. It can be seen in this figure that the stopband is not equiripple. However the phase error of the distortion function $T(z)$ is.

It is also possible to take $A_N(z)$ a delay. Then the output signal has to be equalized by an allpass filter $C(z^2)$ such that the phase response of the cascade of the allpass filters $B_M(z^2)$ and $C(z^2)$ is approximate linear phase. However, a very high order is needed for $C(z)$ then. If the halfband filter is an elliptic filter then the order of $C(z)$ is very high also (empirical result).

### 3.4 Comparison of the two-channel filter banks

Three types of two-channel IIR filter banks have been presented, which are based on allpass filters. A two-channel filter bank which achieves perfect reconstruction, introduces phase distortion and achieves almost perfect reconstruction have been presented respectively.

If phase distortion is allowed, the filter bank with the elliptic filters achieves the best result given a certain complexity and is therefore recommended. If however the phase distortion has to be very small, only the other two of the presented filter banks are useful. These filter banks will be compared.

Consider figure 3.10 of the previous section. This is the case that almost perfect reconstruction is achieved. The orders of $A_N(z)$ and $B_M(z)$ are both $10$. This result will be compared with the case that the order of the allpass filter $\alpha(z)$ is smaller than the order of $\beta(z)$ and the case that the order of $\alpha(z)$ is higher than the order of $\beta(z)$ in the case of perfect reconstruction (see section 3.1).

In figure 3.11 the magnitude responses of $H_0(z)$ and $H_1(z)$ are shown in the case that the order of $\alpha(z)$ and $\beta(z)$ are $9$ and $11$ respectively. The magnitude response of $H_1(z)$ shows an irregularity. In spite of this, the transition band starts with a steep decline and the stopband still starts at $0.4\pi$. Therefore the observed irregularity is not considered as a disadvantage.

In figure 3.12 the magnitude responses of the analysis filters are shown in the case that the order of $\alpha(z)$ is higher than the order of $\beta(z)$. Here the orders of $\alpha(z)$ and $\beta(z)$ are $14$ and $6$ respectively. Notice that $H_1(z)$ has a smaller transition band than $H_0(z)$. It was even possible to obtain a smaller order for $\alpha(z)$ and still keeping a stopband attenuation of approximately $60$ dB. But in that case the decline at the start of the transition band degraded significantly. This will result in more aliasing.

Although the stopband attenuation in the case of perfect reconstruction is a little worse compared to the almost perfect reconstruction filter bank (see figure 3.10, 3.11 and 3.12), this case is preferable. The implementation complexity is somewhat higher because of the extra delays (compare figure 3.2 and 3.8), but has the perfect reconstruction property.

In figure 3.13 the magnitude response of $H_0(z)$ is shown of an FIR two-channel filter bank. This filter has an order of $47$. Due to numerical errors this filter is not exactly power symmetric, i.e.

$$\tilde{H}_0(z)H_0(z) + \tilde{H}_0(-z)H_0(-z) = 1$$

is not fulfilled. Equality is necessary for perfect reconstruction. However it gives an indication of the order for $H_0(z)$ in order to obtain a comparable magnitude response in comparison with the previous examples. As expected the IIR two-channel filter banks have better performances in terms of computational complexity.
Figure 3.11: The magnitude responses of the analysis filters $H_0(z)$ and $H_1(z)$ in the case of the perfect reconstruction filter bank.

Figure 3.12: The magnitude responses of the analysis filters $H_0(z)$ and $H_1(z)$ in the case of the perfect reconstruction filter bank. The orders of $\alpha(z)$ and $\beta(z)$ are 14 and 6 respectively.
Figure 3.13: The magnitude response of the analysis filter $H_0(z)$ of an FIR two-channel filter bank.
Chapter 4

2M-channel modulated filter bank based on IIR filters

In the literature [5], [6], [7] several filter banks are proposed which use IIR filters. Although the analysis filters are stable, the synthesis filters often are unstable. In [10] a method is given to design perfect reconstruction filter banks with IIR filters at the analysis and at the synthesis ends, which both itself are causal and stable. This design method is very general, but does not lend directly itself for the design of analysis filters with good amplitude and phase characteristics.

In this chapter a new filter bank is proposed which uses IIR filters. Although the analysis and the synthesis filters are stable and causal, the filter bank does not have the perfect reconstruction property. However it will be shown that the aliasing, phase and amplitude distortion can be made very small with a very low complexity.

This filter bank is a so-called modulated filter bank, where all the 2M analysis filters are derived from a prototype filter by exponential modulation. As a consequence of the exponential modulation, the modulated filters will have complex coefficients. However the complex computations can be restricted to a minimum by taking the coefficients of the prototype real valued and using a polyphase implementation.

The prototype consists of a cascade of an interpolated halfband filter and a sum of M allpass filters. The halfband filter in turn consists of allpass filters as well. Because of the extensive use of allpass filters, some Matlab functions are developed to simplify the design of the allpass filters. This results in a very simple design of the proposed filter bank.

4.1 The prototype

In [11] an allpass-based structure for an IIR 2Mth-band filter $R(z)$ is proposed. It consists of the filter $H(z)$, cascaded with the correction filter $H_1(z^M)$, i.e. $R(z) = H(z)H_1(z^M)$. $H(z)$ has the following form:

$$H(z) = \frac{1}{M} \sum_{i=0}^{M-1} z^{-i} A_i(z^M)$$  \hspace{1cm} (4.1)

where $A_i(z)$ are stable real allpass filters of order $L$. These allpass filters $A_i(z)$ have approximately linear phase with a slope of $-L + i/M$ between 0 and M times the width of the passband of the $M$th band filter. In [11] it is shown, that if $H(z)$ as in (4.1) is an $M$th-band filter, then $A_0(z) = z^{-L}$. The correction filter $H_1(z^M)$ is designed to suppress the inevitable peaks in the stopband of $H(z)$, which are the result of the allpass filters in $H(z)$. The correction filter $H_1(z^M)$ can be obtained from a halfband filter by $M$-fold interpolation independent of $H(z)$. An example will be given in section 4.5.
A very efficient design of a halfband filter is proposed in [9]. This filter has the following form:

\[ H_1(z) = \frac{1}{2} \left[ D_N(z^2) + z^{-1} D_L(z^2) \right] \]  

\[ = \frac{1}{2} D_L(z^2) \left[ G(z^2) + z^{-1} \right] \]  

\[ G(z) = \frac{D_N(z)}{D_L(z)} \]  

where \( D_N(z) \) and \( D_L(z) \) are stable real allpass filters of order \( N \) and \( L \) respectively. These allpass filters result from the design of \( G(z) \).

It is even possible to obtain a suitable correction filter if \( D_N(z) \) is a delay of order \( L \), thus \( N = L \) and \( D_N(z^2) = z^{-2L} \)

\[ H_1(z) = \frac{1}{2} \left[ z^{-2L} + z^{-1} D_L(z^2) \right] \]  

where \( L \) is the order of the allpass filter \( D_L(z) \).

With the designed \( H(z) \) and \( H_1(z) \), the filter \( R(z) = H_1(z^M)H(z) \) is a 2M-th-band filter that can be used as a prototype for a modulated filter bank. In this type of filter bank the 2M analysis filters are derived from a prototype filter by exponential modulation. The 2M analysis filters are schematically shown in figure 4.1. In this figure \( R_0 \) is the prototype \( R(z) \).

\[ \text{Figure 4.1: The 2M analysis filters of the modulated filter bank, where } R_0 = R(z). \]

### 4.2 The analysis bank

The prototype of the previous section has a passband edge frequency of \( \pi/2M \) and a bandwidth of \( \pi/M \). To modulate this filter with \( \pi/M \), \( z \) has to be replaced with \( z \exp(-j\pi/M) \). This can be achieved by using the 2M polyphase components of the prototype and a DFT matrix. From (4.1) it can be seen that \( H(z) \) is expressed in only \( M \) polyphase components so far. The 2M polyphase \( P_k(z) \) components can be constructed as follows:

\[ P_k(z) = \frac{H_1(z^M)A_k(z^M) + H_1(-z^M)A_k(-z^M)}{2} \]  

\[ 0 \leq k \leq M - 1 \]  

\[ P_k(z) = z^M H_1(z^M)A_{k-M}(z^M) - H_1(-z^M)A_{k-M}(-z^M) \]  

\[ M \leq k \leq 2M - 1. \]  

In fig. 4.2 this construction is shown, followed by the DFT matrix (denoted by \( F^* \)) and the downsamplers. By straightforward calculation it follows that

\[ Q_k(z)/X(z) = \sum_{m=0}^{M-1} (z^{W-k})^{-m} H_1((-1)^kz^M)A_m((-1)^kz^M) \]
where \( W = \exp(-j\pi/M) \) and \( 0 \leq k \leq 2M - 1 \). So these transfers are the modulated versions of the prototype \( R(z) \).

The butterflies in figure 4.2 can be written as a matrix:

\[
K = \frac{1}{2} \begin{pmatrix} I & I \\ I & -I \end{pmatrix} 
\]

where \( I \) is the \( M \times M \) identity matrix.

### 4.3 The synthesis bank

The polyphase matrix of the analysis bank apart from \( F^* \) is a diagonal matrix, where the entries are the polyphase components given in (4.4). For \( N \) is even, where \( N \) is the order of the allpass filters \( A_i(z) \) in (4.1) and thus \( A_0(z^M) = z^{-MN} \), the \( M \)th polyphase component is

\[
P_M(z) = \frac{1}{2} z^M \left[ H_I(z^M)A_0(z^M) - H_I(-z^M)A_0(-z^M) \right] = \frac{1}{2} z^{-MN} D_L(z^{2M}).
\]

For \( N \) is odd, the 0th polyphase component is

\[
P_0(z) = \frac{1}{2} \left[ H_I(z^M)A_0(z^M) + H_I(-z^M)A_0(-z^M) \right] = \frac{1}{2} z^{-MN} z^{-M} D_L(z^{2M}).
\]

This implies that the inverse of the polyphase matrix always has at least one entry where the poles lie outside the unit circle for both proposed halfband filters \( H_I(z) \) (cf.(4.2) and
Thus, perfect reconstruction is not possible with a causal stable synthesis bank for the maximally decimated case.

However, the prototype that is used can have a small transition band. Therefore it is expected that the inverse of the prototype will have approximately the same characteristics. As a result of this approximation, the filter bank will introduce phase, amplitude distortion and aliasing. Nevertheless it will be shown that these errors can be made very small and the consequent analysis will serve as a guideline to improve the synthesis as will be done in the subsequent sections.

In fig. 4.3 the filter bank is shown where the synthesis bank has the same prototype as the analysis bank, with \( B_l(z^M) = A_{M-1-l}(z^M), 0 \leq l \leq M - 1 \) and \( C(z^M) = H_f(-z^M) \). The signals in the analysis bank and the synthesis bank are downsampled and upsampled by \( 2M \) respectively. The \( K \), the DFT matrix and their inverses are not shown, because they have not influence on the transfer function of the filter bank. The transfer function becomes with \( W = \exp(-j\pi/M) \)

\[
P_n = z^{-n}H_f(z^M)A_n(z^M)X(z)
\]

\[
(P_n \downarrow) = \frac{1}{2M} \sum_{k=0}^{2M-1} (z^{\frac{1}{2M}}W^k)^{-n}H_f((z^{\frac{1}{2M}}W^k)^M)A_n((z^{\frac{1}{2M}}W^k)^M)X(z^{\frac{1}{2M}}W^k)
\]
\[
(P_n \downarrow) = \frac{1}{2M} \sum_{k=0}^{2M-1} (zW_k)^{-n} H_I((-1)^k z^\frac{1}{2}) A_n((-1)^k z^\frac{1}{2}) X(zW_k)
\]

where

\[
(P_n \downarrow) = \frac{1}{2M} \sum_{k=0}^{2M-1} (zW_k)^{-n} H_I((-1)^k z^M) A_n((-1)^k z^M) X(zW_k)
\]

(4.6)

\[
P = \sum_{l=0}^{M-1} z^{-(M-1-l)} A_{M-1-l}(z^M) \frac{1}{2M} \sum_{k=0}^{2M-1} (zW_k)^{-l} H_I((-1)^k z^M) A_l((-1)^k z^M) X(zW_k) H(z^M)
\]

\[
P = \frac{1}{2M} \sum_{l=0}^{M-1} \sum_{k=0}^{2M-1-1} z^{-kl} H_I((-1)^k z^M) H_I(z^M) A_l((-1)^k z^M) A_{M-1-l}(z^M) X(zW_k).
\]

(4.7)

In the same manner it can be found that

\[
Q = \frac{1}{2M} z^{-(M-1)} \sum_{l=0}^{M-1} \sum_{k=0}^{2M-1-1} z^{-kl} H_I((-1)^k z^M) H_I(z^M) A_l((-1)^k z^M) A_{M-1-l}(z^M) X(zW_k).
\]

This yields for \(\hat{X}\) that

\[
\hat{X} = P + Q
\]

\[
\hat{X} = \frac{1}{2M} z^{-M-1} \left( \sum_{l=0}^{M-1-1} \sum_{k=0}^{2M-1-1} z^{-kl} H_I((-1)^k z^M) H_I(z^M) A_l((-1)^k z^M) A_{M-1-l}(z^M) X(zW_k) + \right.
\]

\[
\sum_{l=0}^{M-1-1} \sum_{k=0}^{2M-1-1} z^{-kl} H_I((-1)^k z^M) H_I(z^M) A_l((-1)^k z^M) A_{M-1-l}(z^M) X(zW_k)
\]

(4.8)

\[
\approx \frac{1}{2M} z^{-M-1} \sum_{l=0}^{M-1-1} \sum_{k=0}^{2M-1-1} z^{-2kl} H_I^2(z^M) A_l(z^M) A_{M-1-l}(z^M) X(zW^2k) +
\]

\[
\frac{1}{2M} z^{-M-1} \sum_{l=0}^{M-1-1} \sum_{k=0}^{2M-1-1} z^{-2kl} H_I^2(z^M) A_l(z^M) A_{M-1-l}(z^M) X(zW^2k)
\]

(4.9)

\[
\approx \frac{1}{2M} z^{-M-1} \sum_{l=0}^{M-1-1} \sum_{k=0}^{2M-1-1} z^{-2kl} \left\{ H_I^2(z^M) G^2(z^M) + H_I^2(-z^M) G^2(-z^M) \right\} X(zW^2k)
\]

(4.10)

\[
= \frac{1}{2M} z^{-M-1} \left\{ H_I^2(z^M) G^2(z^M) + H_I^2(-z^M) G^2(-z^M) \right\} \sum_{k=0}^{M-1} X(zW^2k) \sum_{l=0}^{M-1} (W^{-2k})^l
\]

\[
= \frac{1}{2M} z^{-M-1} \left\{ H_I^2(z^M) G^2(z^M) + H_I^2(-z^M) G^2(-z^M) \right\} X(z)
\]

(4.11)

\[
\approx \frac{1}{2} z^{-M-1} K(z^M) X(z).
\]

(4.12)

where the following approximations are made:

- (4.9): \(|H_I(-z^M) H_I(z^M)| \approx 0, z \in T;\)
- (4.10): \(H_I(z^M) A_l(z^M) A_{M-1-l}(z^M) \approx H_I(z^M) A^2_{M^2-1/2}, 0 \leq l \leq M - 1 \text{ and } z \in T;\)
- (4.12): \(K(z^M) = H_I^2(z^M) G^2(z^M) + H_I^2(-z^M) G^2(-z^M)\) is a filter with an amplitude approximately 1, \(z \in T;\)
where $T$ is the unit circle of the complex plane. Eq. (4.11) follows from

$$
\sum_{k=0}^{M-1} X(zW^k) \sum_{l=0}^{M-1} (W^{-2k})^l = X(z). 
$$

(4.13)

The approximation used to obtain (4.9) is not valid around the transition bands of $H_I(z^M)$ and $H_I(-z^M)$. From (4.8) it is inferred that the terms with $H_I(z^M)H_I(-z^M)$ in $\hat{X}$ will disappear if the analysis bank and the synthesis are down- and upsampled respectively with a factor $M$ instead of a factor $2M$, which is the obvious way to circumvent this problem. This means there is an oversampling factor 2 necessary. In the subsequent sections this will be done so.

4.3.1 An aliasing and amplitude distortion free filter bank

In the crucial approximation (4.10) it is assumed that

$$
H_I(z^M)A_l(z^M)A_{M-1-l}(z^M) \approx H_I(z^M)A_{M/2-1/2}(z^M), \quad 0 \leq l \leq M - 1 \text{ and } z \in T.
$$

In this way it is possible to use (4.13), because nothing else is dependent on $l$ anymore. An aliasing and amplitude distortion free filter bank can be obtained if $A_l(z^M)A_{M-1-l}(z^M)$ and $A_l(-z^M)A_{M-1-l}(-z^M)$ are replaced by

$$
A_0(z^M)A_1(z^M) \cdots A_{M-1}(z^M)A_0(-z^M)A_1(-z^M) \cdots A_{M-1}(-z^M).
$$

Thus

$$
B_k(z^M) = \prod_{m \neq k}^{M-1} A_m(z^M)\prod_{m=0}^{M-1} A_m(-z^M).
$$

This results in:

$$
(P_n \downarrow \uparrow) = \frac{1}{M} H_I(z^M) \sum_{k=0}^{M-1} (zW^k)^{-n} A_n(z^M)X(zW^k)
$$

$$
P = \frac{1}{M} H_I(z^M) \prod_{m=0}^{M-1} A_m(z^M)A_m(-z^M) \sum_{l=0}^{M-1} z^{-(M-1-l)} \sum_{k=0}^{M-1} (zW^k)^{-l} X(zW^k)
$$

$$
= \frac{1}{M} z^{-(M-1)} H_I(z^M) \prod_{m=0}^{M-1} A_m(z^M)A_m(-z^M) X(zW^k) \sum_{l=0}^{M-1} W^{-kl}
$$

$$
= z^{-(M-1)} H_I(z^M) \prod_{m=0}^{M-1} A_m(z^M)A_m(-z^M)X(z).
$$

In the same manner it can be found that

$$
Q = z^{-(M-1)} H_I(-z^M) \prod_{m=0}^{M-1} A_m(z^M)A_m(-z^M)X(z).
$$

$\hat{X}$ becomes then

$$
\hat{X} = P + Q
$$

$$
= z^{-(M-1)} \prod_{m=0}^{M-1} A_m(z^M)A_m(-z^M)D_N(z^2M)X(z)
$$

24
or
\[
\hat{X} = P - Q
\]
\[
= z^{-(2M-1)} \prod_{m=0}^{M-1} A_m(z^M)A_m(-z^M)D_L(z^{2M})X(z).
\]

Recall that \( D_N(z) \) and \( D_L(z) \) are allpass filters, see (4.2), and so these filter banks only introduce phase distortion. The required number allpass filters in the filter bank will increase quadratically with \( M \).

### 4.3.2 Approximation of the aliasing and amplitude distortion free filter bank

Because of the increasing complexity of the synthesis bank in the case of the aliasing and amplitude distortion free filter bank, it is desirable to approximate this filter bank. This can be obtained in the following way. Note first that each channel in the upper and the lower part of the filter bank in the previous section has the same phase response. The channels have approximately the same phase response if for example \( B_l(z^M) = A_{M-1-l}(z^M) \) where \( 0 \leq l \leq M - 1 \) and \( C(z^M) = A_{M/2-1}(z^M)A_{M/2}(z^M) \). With these choices, this results in

\[
(P_n \downarrow) \uparrow = \frac{1}{M} H_f(z^M) \sum_{k=0}^{M-1} (z^{Wk})^{-n} A_n(z^M)X(z^{Wk})
\]

\[
P = \frac{1}{M} H_f(z^M)C(-z^M) \sum_{l=0}^{M-1} z^{-(M-1-l)} A_{M-1-l}(z^M) \sum_{k=0}^{M-1} (z^{Wk})^{-l} A_l(z^M)X(z^{Wk})
\]

\[
\approx z^{-(M-1)} H_f(z^M)A_{M/2-1}(-z^M)A_{M/2}(-z^M)A_{M/2-1}(z^M)A_{M/2}(z^M)X(z).
\]

In the same manner it can be found that

\[
Q \approx z^{-(M-1)} H_f(z^M)A_{M/2-1}(-z^M)A_{M/2}(-z^M)A_{M/2-1}(z^M)A_{M/2}(z^M)X(z).
\]

Then \( \hat{X} \) becomes

\[
\hat{X} = P + Q \approx z^{-(M-1)} D_N(z^{2M})A_{M/2-1}(-z^M)A_{M/2}(-z^M)A_{M/2-1}(z^M)A_{M/2}(z^M)X(z)
\]

or

\[
\hat{X} = P - Q \approx z^{-(M-1)} D_L(z^{2M})A_{M/2-1}(-z^M)A_{M/2}(-z^M)A_{M/2-1}(z^M)A_{M/2}(z^M)X(z).
\]

Thus these filter banks approximate the aliasing and amplitude distortion free filter bank, but have less complexity.

### 4.3.3 Almost perfect reconstruction

The disadvantage of the filter bank proposed in the previous section is the non linear phase response of the filter bank. If the \( M \)-fold interpolated halfband filter \( H_f(z^M) \) is composed of a delay and an allpass filter, thus (cf.(4.3))

\[
H_f(z^M) = \frac{1}{2} \left( z^{-2ML} + z^{-M} D_L(z^{2M}) \right)
\]

where \( L \) is the order of the allpass filter \( D_L(z) \), then \( H_f(z^M) + H_f(-z^M) = z^{-2ML} \). As a consequence, if each channel has approximately linear phase, then the filter bank will have a linear phase response as well. Thus in the upper part of the filter bank it is desired that

\[
A_l(z^M)B_l(z^M) \approx z^{-N}
\]
in the pass bands and the transition bands of $H_I(z^M)$ for a certain integer $N$ dependent on
the orders of $A_I(z)$ and $B_I(z)$. In the lower part of the filter bank it is desired that

$$A_I(-z^M)B_I(-z^M) \approx \pm z^{-N}$$

in the pass bands and the transition bands of $H_I(-z^M)$. Since the phase of $A_I(-1)B_I(-1)$ can
be $\pm \pi$, thus if $\theta = 0$, a minus sign can appear. In this case $\hat{X}(z)$ is obtained by subtraction
instead of addition. Notice that in the stopbands of $H_I(z^M)$ and $H_I(-z^M)$ the amplitude of
$H_I(z^M)A_I(z^M)B_I(z^M)$ and $H_I(-z^M)A_I(-z^M)B_I(-z^M)$ are approximately zero. Therefore,
the phase response in the stopbands can be taken $-N\theta$ as well. Then $\hat{X}(z)$ becomes

$$\hat{X}(z) \approx z^{-2ML}z^{-N}X(z).$$

Figure 4.4: The unwrapped phase responses in the case that the slope of the phase response
of $A_I(z^M)B_I(z^M)$ in the pass bands and the transition bands of $H_I(z^M)$ is approximately
$-M(N_A + N_B)$ where $M = 4$, $N_A = 2$ and $N_B = 2$ (upper part), and the case that the slope
of the phase response of $A_I(z^M)B_I(z^M)$ in the pass bands and the transition bands of $H_I(z^M)$
is approximately $-M(N_A + N_B) + M$ (lower part).

There are two obvious choices for $B_I(z)$, namely

- The phase response of $B_I(z)$ is such that the slope of the phase response of $A_I(z)B_I(z)$
in the pass band and the transition band of $H_I(z)$ is approximately $-(N_A + N_B)$, where
$N_A$ and $N_B$ are the orders of the allpass filters $A_I(z)$ and $B_I(z)$ respectively.

- The phase response of $B_I(z)$ is such that the slope of the phase response of $B_I(z)$ in the
pass band and the transition band of $H_I(z)$ is approximately $-(N_A + N_B) + 1$.

With these choices

$$A_I(z^M)B_I(z^M) \approx z^{-N}$$
in the pass bands and the transition bands of $H_I(z^M)$ and

$$A_I(-z^M)B_I(-z^M) \approx \pm z^{-N}$$
in the pass bands and the transition bands of $H_I(-z^M)$. An example is shown in figure 4.4 for both cases where $M = 4$, $N_A = 2$ and $N_B = 2$. Notice that in the second case the unwrapped phase response show transitions of $2\pi$, so the wrapped phase response will not show transitions in the pass bands and transition bands of $H_I(z^M)$.

4.4 The implementation of the modulated filter bank

The DFT matrices in the case of a complex input signal can be implemented with the fast Fourier transformation. If the input signal is real then some simplifications are possible. Consider figure 4.5. The upper part of the filter bank has to modulate the prototype $M/2 + 1$ times. The lower part of the filter bank has to modulate the prototype $M/2$ times. Cf. figure 4.1 on page 20 (it is assumed that $M$ is even). The $(M/2 + 1) \times M$ matrix $W_0^*$ and the $M/2 \times M$ matrix $W_1^*$ can be implemented with the fast Fourier transformation. The $R_k$ and $T_k$ in the figure can be expressed as

$$R_k = \sum_{m=0}^{M} W_2^{km} P_m + \sum_{m=M+1}^{2M-1} W_2^{km} P_{2M-m} = \sum_{m=0}^{M} W_2^{km} P_m + \sum_{m=0}^{M-2} W_2^{k(m+M+1)} P_{M-1-m}$$

$$= \sum_{m=0}^{M} W_2^{km} P_m + \sum_{m=0}^{M-2} (-1)^k W_2^{k(m+1)} P_{M-1-m} = \sum_{m=0}^{M} W_2^{km} P_m + \sum_{m=1}^{M-1} (-1)^k W_2^{km} P_{M-m}$$

Figure 4.5: The matrices between the down- and upsamplers in the modulated filter bank, where $W_0^{mk} = \exp(-j2\pi km/M)$, $W_1^{mk} = \exp(-j\pi k(2m+1)/M)$, $W_2^{mk} = \exp(-j\pi km/M)$ and $K$ the matrix as defined in (4.5) on page 21.
\[
T_k = R_k + R_{k+M} = Re \left\{ P_0 + (-1)^k P_M + 2 \sum_{m=1}^{M-1} W_2^{km} P_m \right\} + \\
Re \left\{ P_0 + (-1)^k P_M + 2 \sum_{m=1}^{M-1} W_2^{(k+M)m} P_m \right\} \\
= Re \left\{ 2P_0 + 2(-1)^k P_M + 2 \sum_{m=1}^{M-1} W_2^{km} P_m + 2 \sum_{m=1}^{M-1} (-1)^m W_2^{km} P_m \right\} \\
= 2P_0 + 2(-1)^k P_M + 4Re \sum_{m=1}^{M/2-1} W_2^{2km} P_{2m}
\]

where \(W_2^{km} = \exp(-j2\pi km/M)\). Notice that \(P_0\) and \(P_M\) are real if the input signal is real. For \(0 \leq k \leq M - 1\), \(T_k\) becomes

\[
T_k = R_k + R_{k+M} = Re \left\{ 2 \sum_{m=1}^{M-1} W_2^{km} P_m - 2 \sum_{m=1}^{M-1} W_2^{(k+M)m} P_m \right\} \\
= 2Re \left\{ \sum_{m=1}^{M/2-1} W_2^{km} P_m - \sum_{m=1}^{M-1} (-1)^m W_2^{km} P_m \right\} \\
= 4Re \sum_{m=0}^{M/2-1} W_2^{k(2m+1)} P_{2m+1} = 4Re \sum_{m=0}^{M/2-1} W_1^{km} P_{2m+1}
\]

where \(W_1^{km} = \exp(-j\pi(2m+1)k/M)\). So \(T_k\) for \(0 \leq k \leq M - 1\) can also be implemented with the fast Fourier transform or with the fast DCT and fast DST [14]. Notice that indeed \(W_0 W_0^\dagger \neq I\)

Where \(W_0^\dagger\) denotes the transpose-conjugate of \(W_0\).

Furthermore, the filters \(H_I(z^M)\) and \(H_I(-z^M)\) in the analysis bank can be combined. This is shown in figure 4.6 (cf. figure 4.3). This results in a more efficient implementation.
A real allpass filter $A(z)$ has the transfer function

$$A(z) = \frac{\sum_{k=0}^{N_A} a_{N_A-k} z^{-k}}{\sum_{k=0}^{N_A} a_k z^{-k}}$$

where $N_A$ is the order of the allpass filter $A(z)$. Thus the difference equation is

$$y(n) = -a_1 y(n-1) - a_2 y(n-2) - \ldots - a_{N_A} y(n - N_A)$$
$$+ a_{N_A} x(n) + a_{N_A-1} x(n-1) + \ldots + a_0 x(n - N_A)$$
$$= x(n - N_A) + a_1 \{x(n - N_A + 1) - y(n-1)\} + a_2 \{x(n - N_A + 2) - y(n-2)\}$$
$$+ \ldots + a_{N_A} \{x(n) - y(n - N_A)\}$$
$$= x(n - N_A) + \sum_{k=1}^{N_A} a_k \{x(n - N_A + k) - y(n - k)\}$$

Recall that $a_0 = 1$. Thus an allpass filter $A(z)$ or $A(-z)$ of order $N_A$ requires in this implementation $N_A$ multiplications and $2N_A + 1$ additions. The implementation of $H_I(\pm z)$ requires $L$ multiplications and $2L + 3$ additions.

Thus the complexity of the analysis bank becomes exclusive the modulation overhead and inclusive the $K$ matrix $M(L + 2N_A)/M = (L + 2N_A)$ multiplications per unit time (MPU) and $M(2L + 3 + 2(2N_A + 1) + 2M)/M = 2L + 4N_A + 7$ additions per unit time (APU).

The complexity of the synthesis bank becomes exclusive the modulation overhead and inclusive the $K$ matrix (cf. figure 4.3) $2MN_B/M = 2N_B$ MPU and $M + (2M(2N_B + 1) + 2M)/M = 4N_B + M + 4$ APU.

Thus the complexity in total becomes exclusive the modulation overhead and inclusive the $K$ matrix $2N_A + 2N_B + L$ MPU and $4N_A + 4N_B + 2L + M + 11$ APU.
4.5 The design of the filter bank

First the $M$th lowpass filter has to be designed. See [11] for the details. The allpass filters can be designed as described in [12] or [13]. These designs are based on the eigenvalue problem. The proposed methods consider the phase error between the the phase response of the allpass network to be realized and the ideal phase response. The algorithms calculate the coefficients of the allpass filter in an iterative way such that the infinity norm of the phase error is minimized.

Because of $H_I(z^M)$ the $M$th lowpass filter $H(z)$ can have a very wide transition band. Non-overlapping of the two filters is the only restriction. This can be obtained if the allpass filters $A_l(z)$ in $H(z)$ have an approximate linear phase between 0 and at least $\pi/2$.

The choice of the halfband filter $H_I(z)$, is dependent on which filterbank is desired. If almost perfect reconstruction is desired $H_I(z)$ has to be composed of a delay and a delayed allpass filter. If phase distortion is allowed, $H_I(z)$ can be better designed as described in [9] because of the better amplitude characteristic.

In the case of almost perfect reconstruction the synthesis bank cannot be built from the analysis bank, but has to be designed separately. A better performance of the filterbank can be achieved if the synthesis bank is designed given the analysis bank. In other words for each channel the phase characteristic of the cascade of the allpass filters $A_l(z)$ and $B_l(z)$ for $0 \leq l \leq M-1$ is equiripple in the passband and the transition band of $H_I(z)$. This means that if the transition band of $H_I(z)$ increases the phase error of the allpass filters $B_l(z)$ increases as well if the order of $B_l(z)$ remains the same.

Here the design method will be explained by an example. An almost perfect reconstruction filterbank will be designed with $M = 32$. The $2M$th lowpass filter must have a stopband attenuation of $90 \text{dB}$. Furthermore the transition bandwidth of $H_I(z)$ has to be $0.1\pi$. Thus the transition bandwidth of $H_I(z^{32})$ will be $0.1\pi/32$.

The phase error of the allpass filters in $H(z)$, thus the allpass filters $A_l(z)$, have to be equiripple from 0 to $0.5\pi$. The cascade of the allpass filters $A_l(z)$ and $B_l(z)$ have to be equiripple from 0 to $0.6\pi$, because the stopband of $H_I(z)$ begins at $0.6\pi$. The slope of the phase response of the cascade of the allpass filters $A_l(z)$ and $B_l(z)$ will be taken approximately $-N_A - N_B + 1$. In figure 4.7 the amplitude responses of $H_I(z^{32})$ and $H(z)$ are shown. Notice the inevitable peak in the magnitude response of $H(z)$. In figure 4.8 the amplitude response of $H_I(z^{32})H(z)$ is shown. Here the allpass filters $A_l(z)$ and $B_l(z)$ have an order of 4 and 7 respectively. The order of the allpass filter $D_L(z)$ is 11.

Ideally this filter bank has a delay as transfer function. Then the transfer function has an amplitude 1 and a linear phase response. Figure 4.9 shows the amplitude error and the phase error of the distortion function of the designed filter bank. So the distortion function has an maximum amplitude distortion of approximate -93.6 dB and a maximum phase distortion of approximately $\pm 3.8 \cdot 10^{-5}$ radians.

The distortion function is the transfer function of the filter bank if there is no aliasing. The modulated filter bank has not the perfect reconstruction property so there will be aliasing. The transfer function of the filter bank is

$$\hat{X}(z) = \sum_{\ell=0}^{M-1} G_\ell(z)X(z^{\ell})$$

where

$$G_\ell(z) = \frac{1}{M} \sum_{k=0}^{2M-1} H_k(z^{\ell})F_k(z), \quad 0 \leq \ell \leq M-1$$

where $H_k(z)$ and $F_k(z)$ are the analysis and synthesis filters respectively and $W = \exp(-j2\pi/M)$. The contribution of the aliasing can be given by the peak aliasing distortion $E_a$ [1]. This
Figure 4.7: The amplitude responses of $H(z)$ and $H_f(z^{32})$.

Figure 4.8: The amplitude response of the prototype $R(z) = H_f(z^{32})H(z)$.

quantity $E_a$ is the worst possible peak aliasing distortion. It is defined as

$$E_a = \sqrt{\frac{1}{M-1} \sum_{t=1}^{M-1} |G_t(z)|^2}.$$
Figure 4.9: The amplitude error and the phase error of the distortion function of the designed filter bank.

Figure 4.10: The peak aliasing distortion $E_a$ of the designed filter bank.
The peak aliasing distortion of the modulated filter bank becomes then

\[ E_a = \left( \sum_{k=1}^{M-1} \left| \frac{1}{M} e^{-j\theta(M-1)} H_1(e^{j\theta M}) \sum_{l=0}^{M-1} W^{-kl} A_l(e^{j\theta M}) B_l(e^{j\theta M}) \right| + \right) \left( \sum_{k=1}^{M-1} \left| \frac{1}{M} e^{-j\theta(M-1)} H_1(-e^{j\theta M}) \sum_{l=0}^{M-1} W^{-kl} A_l(-e^{j\theta M}) B_l(-e^{j\theta M}) \right| \right)^{\frac{1}{2}} \]

In practice the aliasing will be in general less. In figure 4.10 the peak aliasing distortion is shown. From this figure it follows that the peak aliasing distortion is approximately -93 dB. Thus a very good performance of the filter bank can be obtained with a very low complexity.

### 4.6 An FIR modulated filter bank

There exists an FIR equivalent for the IIR modulated filter bank, which is shown in figure 4.11. Notice that the DFT matrices are omitted. This filter bank has the perfect reconstruction property. Consider figure 4.11. It follows that

\[ (P_k \downarrow) \uparrow = \frac{1}{M} \sum_{n=0}^{M-1} (zW^n)^{-k} E_k(z^{2M}) X(zW^n) \]

where \( W = \exp(-j2\pi/M) \). Then \( \hat{X}(z) \) becomes:

\[
\hat{X}(z) = \frac{1}{M} \sum_{\ell=0}^{2M-1} z^{-(2M-1-\ell)} z^{-K} \hat{E}_\ell(z^{2M}) \sum_{n=0}^{M-1} (zW^n)^{-\ell} E_\ell(z^{2M}) X(zW^n) = \frac{1}{M} z^{-(2M-1)} z^{-K} \sum_{\ell=0}^{2M-1} \sum_{n=0}^{M-1} W^{-n\ell} E_\ell(z^{2M}) \hat{E}_\ell(z^{2M}) X(zW^n)
\]
\[
\begin{align*}
&= \frac{1}{M} z^{-(2M-1)} z^{-K} \sum_{\ell=0}^{M-1} \sum_{n=0}^{M-1} W^{-n\ell} E_\ell(z^{2M}) \tilde{E}_\ell(z^{2M}) X(zW^n) + \\
&= \frac{1}{M} z^{-(2M-1)} z^{-K} \sum_{\ell=M}^{2M-1} \sum_{n=0}^{M-1} W^{-n\ell} E_\ell(z^{2M}) \tilde{E}_\ell(z^{2M}) X(zW^n) \\
&= \frac{1}{M} z^{-(2M-1+K)} \left( \sum_{\ell=0}^{M-1} \sum_{n=0}^{M-1} W^{-n\ell} \left( E_\ell(z^{2M}) \tilde{E}_\ell(z^{2M}) + E_{\ell+M}(z^{2M}) \tilde{E}_{\ell+M}(z^{2M}) \right) X(zW^n) \right)
\end{align*}
\]

where \( \tilde{E}_k(z) = E_{k*}(z^{-1}) \) and the subscript asterisk notation means that the coefficients are conjugated of those in \( E_k(z) \). If the polyphase components \( E_\ell(z) \) and \( E_{\ell+M}(z) \) are power complementary, i.e.

\[
\tilde{E}_\ell(z) E_\ell(z) + \tilde{E}_{\ell+M}(z) E_{\ell+M}(z) = \alpha, \quad 0 \leq \ell \leq M - 1 \quad (4.14)
\]

then by using

\[
\sum_{k=0}^{M-1} X(zW^k) \sum_{\ell=0}^{M-1} (W^{-k})^\ell = MX(z)
\]

\( \hat{X} \) becomes

\[
\hat{X}(z) = \alpha z^{-(2M-1+K)} X(z).
\]

These filters exist and are used in cosine modulated filter banks [1]. From this it follows that the constant \( K = N - 2M + 1 \) where \( N \) is the order of the prototype and \( N + 1 = 2mM \) and \( m \) is an integer.

An FIR prototype of order 2175 is shown in figure 4.12. The polyphase components of this filter are not exactly power complementary [15], but it gives an indication of the order of the prototype that will be needed. This filter has a stopband attenuation of approximately -92.3 dB and a transition bandwidth of \( 0.1\pi/32 \). The orders of the polyphase components in the \( 2M \) channels are 33.

It is difficult to compare this FIR filter bank with the IIR modulated filter bank, because the latter does not have the perfect reconstruction property. However it was possible to achieve very good results with the IIR modulated filter bank, therefore the filter banks will be compared anyway.

The FIR filter bank requires in total \( 2(2176/32) = 136 \) MPU and \( 2(2176/32) + 31 = 167 \) APU.

The allpass filters \( A_l(z) \) and \( B_l(z) \) of the analysis and synthesis bank of the IIR filter bank which is designed in the previous section had an order of respectively 4 and 7. The allpass filter in the halfband design had an order of 11. Thus the IIR bank requires (see section 4.4) 33 MPU and 109 APU. This means that the IIR filter bank requires a factor 4.12 less MPU and a factor 1.59 less APU in this example.
Figure 4.12: An FIR prototype for an FIR modulated filter bank.
Chapter 5

Examples of the Matlab functions

Here follow some examples to illustrate the developed Matlab functions.

Here it will be shown how $\alpha(z)$ and $\beta(z)$ are designed in the case of figure 3.6 on page 12. The slope of the phase response of the allpass filter $\beta(z)$ of order 9 has to be $-19/2$ between 0 and $0.8\pi$. This can be done by

```matlab
>> beta=desequi(9,0.8*pi,-19/2);
```

Now beta contains the denominator of the allpass filter $\beta(z)$. The allpass filter $\alpha(z)$ of order 10 has to be designed such that the slope of the phase response of $\alpha(z)\beta(z)$ is approximately $-19$ between 0 and $\pi$. This can be done by

```matlab
>> alpha=desequi2(10,pi,-19,beta');
```

Now alpha contains the denominator of the allpass filter $\alpha(z)$ and the slope of the phase response of $\alpha(z)\beta(z)$ will be approximately $-19$.

The design of a filter bank which has the almost perfect reconstruction property is very simple. With

```matlab
>> [A,B,C,d]=desbank2(32,4,7,11,0.6);
```

the 64-channel filter bank will be designed where the orders of the allpass filters $A_0(z)$ and $B_0(z)$ will be 4 and 7 respectively. The order of $D_L(z)$ is 11 and the stopband of $H_I(z)$ starts at $0.6\pi$. The rows of A and B contain the denominators of the allpass filters in the analysis and synthesis filters respectively. $A_0(z)$ and $B_0(z)$ are not contained in A and B, because they are always a delay.

The transfer/distortion function of this filter bank can be shown with

```matlab
>> transf(A,C,B,1);
```

The peak aliasing distortion can be shown with

```matlab
>> peakali(A,C,B,1)
```

A simulation with the filter bank can be done with

```matlab
>> x=simul(512,A,C,B,1);
```

The 1 in transf and simul indicates that the filter bank is designed with desbank2 instead of desbank.
Chapter 6

Concluding remarks

A literature search is done. From this it follows that there is done some research in filter banks with IIR filters. This research falls apart into two subclasses:

- filter banks with only stable and causal filters.
- filter banks with instable or a-causal filters.

For realtime applications only the first item is of interest. This item falls apart in filter banks which achieve perfect reconstruction and which not. For two-channel filter banks there exists a filter bank which achieves perfect reconstruction. In the case of $M$-channel filter banks there is a method to achieve perfect reconstruction, however this design method is very general and leads not directly to analysis filters with good amplitude and phase characteristics. In general it can be concluded that subband coding with causal and stable IIR filters is possible, but a good design method of $M$-channel filter banks which have the perfect reconstruction is still not available. In the case of the two-channel filter bank however there exists a good filter bank with causal and stable IIR filters which achieves perfect reconstruction.

The filters in the above mentioned two-channel filter bank showed a bump in the magnitude responses. These bumps has been eliminated with only minor adjustments. With these adjustments, this filter bank is a suitable candidate for the so-called tree structured filter banks.

If phase distortion, which is introduced by the filter bank, is not of interest, the filter bank with elliptic filters has the best performance. In this case a good magnitude response of the analysis and synthesis filters can be designed with a very low complexity.

Furthermore, a new filter bank is proposed based on IIR filters. It is a so-called modulated filter bank. This filter bank has not the perfect reconstruction property. However, the phase distortion, magnitude distortion and the aliasing can be made very small with a very low complexity. This filter bank is not maximally decimated but oversampled by a factor 2. But because of the very low complexity and the possibility to reduce the modulation overhead in case of real input signals this proposed filter bank is still very efficient. Because of the aliasing it is difficult to determine the transfer function of the filter bank. This is a disadvantage of the proposed filter bank. But by determining the distortion function and the peak aliasing distortion of the filter bank it is possible to get an indication of the expected performance. An advantage is the very simple design and the low complexity. The analysis filters can have very sharp transitions and very good magnitude responses and still have very low complexity. Furthermore, the implementation can be very efficient because the IIR filters are allpass filters which can be implemented very efficiently. Several simulations are done. From this it can be concluded that this filter bank can be very useful.

Some Matlab functions are written to design allpass filters, elliptic filters of a certain form and to design the $2M$-channel modulated filter bank. This designed $2M$-channel modulated
filter bank can be simulated with Matlab. With these Matlab functions the design of the filters and the $2M$-channel modulated filter bank becomes very simple. Some of these Matlab function can be combined into smaller functions.

### 6.1 Discussion

There is a Matlab function developed which designs an elliptic filter of the form

$$E(z) = A_0(z^2) + z^{-1}A_1(z^2)$$  \hspace{1cm} (6.1)

Experimentally it is found that this class of filters are exactly the same as the filters which are proposed in [9]. In case of the elliptic filters only the stopband attenuation and the start point of the stopband is given for the design. The filters proposed in [9] are designed in a different way. In this case the allpass filter

$$G(z) = \frac{A_0(z)}{A_1(z)}$$

is designed such that the phase response of $G(z)$ is approximately equal to $-1/2$ between 0 and $\theta_p$. From the instable allpass filter $G(z)$ the stable allpass filters $A_0(z)$ and $A_1(z)$ are derived. This results in a half band filter which stopband starts at $\pi - \theta_p/2$. In this design the order of $G(z)$ and $\theta_p$ is given. The algorithm which designs the elliptic filter directly is much faster than the algorithm which designs an allpass filter with approximate linear phase.

So two questions arises:

- Is the elliptic filter of the form as in (6.1) the same as the halfband filter proposed in [9]?
- Is it possible to make an algorithm which designs the prototype of the modulated filter bank in a very fast way?

The algorithm to design the prototype is already fast, but of course any improvement is welcome.
Bibliography

[1] P.P. Vaidyanathan
Multirate systems and filter banks.

Subband image coding.

Subband coding of images.
IEEE Trans. on Acoust. Speech and Signal Proc.,

Digital coding of speech in subbands.

Wavelets generalized by IIR filter banks.

IIR analysis and synthesis systems.

A novel implementation of perfect reconstruction QMF filter banks using IIR filters for
infinite length signals.
Proc. IEEE Int. Symp. on CAS,

A new class of two-channel biorthogonal filter banks and wavelet bases.
IEEE Transactions on Signal Processing,

Equiripple design of QMF banks using digital allpass networks.
IEICE Trans. Fundamentals,

Wavelets and perfect reconstruction subband coding with causal stable IIR filters.

Interpolated IIR Mth-band filter design with allpass subfilters.

[12] X. Zhang and H. Iwakura
Design of digital allpass networks based on the eigenvalue problem.

Eigenfilter approach for the design of allpass filters approximating a given phase response.

Fast decimation-in-time algorithms for a family of discrete sine and cosine transforms.

Ontwerp van cosinus-gemoduleerde subband-filterbanken.
Appendix A

The Matlab functions

These Matlab functions are written for Windows. This means that for example `ones([1 N])`, `eye([1 N])` and `zeros([1 N])` are used instead of `ones(1,N)`, `eye(1,N)` and `zeros(1,N)`. Therefore these function will not work for Matlab for DOS. However with some modifications it will be possible.

A.1 Elliptic filter design

An algorithm to design an elliptic filter which has the following form

\[ E(z) = A_0(z^2) + z^{-1}A_1(z^2) \]  

(A.1)

is given in [1]. This algorithm is implemented as follows

```matlab
function [a0,a1]=ellallp(Ws,Rs);
% Deze functie [a0,a1]=ELLALLP(Ws,Rs) ontwerpt een elliptische halfband filter
% die te schrijven is als
% E(z)=A0(z^2)+z^-1A1(z^2)
% De stopband begin bij Ws*pi en heeft een minimale demping in de stopband
% van Rs dB.
% a0 en a1 bevatten de coefficienten van de allpass filters A0(z) en A1(z).
% Referentie:
% Antoniou. A
% Digital filters: analysis and design.
% Ook in:
% Vaidyanathan P.P.
% Multirate systems and filter banks.
% New Jersey: Prentice Hall, Englewood Cliffs

% Specifications
Wp=1-Ws; Wp=Wp*pi; Ws=Ws*pi;
delta2=10^(-Rs/20);
d=[4 -4 delta2^2];
deltal=min(roots(d));
Amax=-20*log10(1-2*deltal);

% Order Estimation
r=tan(Wp/2)/tan(Ws/2);
rd=sqrt(1-r^2);
q0=0.5*(1-sqrt(rd))/(1+sqrt(rd));
qu=2q0^2+16q0^3+180q0^5+1800q0^7+18000q0^9+180000q0^11;
D=(1-delta2^2)/delta2^2;N=ceil(log10(16*D)/log10(1/q));
if rem(N,2)==0, N=N+1; end;
```
Readjusting ripple size

\[ D = 10^{-\left(\text{log}_{10}(1/q)\right)/16}; \]

\[ \delta_2 = \min(\text{sqrt}((\text{roots}([D-1 2 -1])))\); \]

\[ \delta_1 = \min(\text{roots}([4 -4 \delta_2 -2])); \]

\[ A_0 = -20 \times \text{log}_{10}(\delta_2); \]

\[ A_{\text{max}} = -20 \times \text{log}_{10}(1 - 2 \times \delta_1); \]

Compute the filter coefficients

\[ w = \frac{1+r}{\text{sqrt}(r)}; \]

for \( k = 1:(N-1)/2, \)

\[ S_1 = 0; \quad S_2 = 0; \]

for \( i = 0:5, \)

\[ S_1 = S_1 - q^{-1}k \times \sin(k \times i \times \pi/N); \]

\[ S_2 = S_2 + q^{-1}k \times \cos(2 \times \pi \times k \times i \times \pi/N); \]

end;

\[ D = (q^{-1/4} \times S_1) / (1 + 2 \times S_2); \]

\[ v = \text{sqrt}(1 - r \times D^2); \]

\[ b = 2 \times v / (1 + D^2); \]

\[ a = (2 - b) / (2 + b); \]

end;

\[ a = \text{sort}(a); \]

\[ l = \text{length}(a); \]

\[ a_0 = \text{down}(a, 2); \]

\[ a_1 = \text{down}(a(2:1), 2); \]

\[ a_0 = \text{poly}(a_0); \]

\[ a_1 = \text{poly}(a_1); \]

end;

A.2 Allpass filter design

There are three algorithms to design allpass filters with approximately linear phase from 0 to \( \theta_p \). With some modifications it is possible to design allpass filters with arbitrary phase. It is also possible to use the algorithm eigendes instead (see chapter 5). The algorithms are desequi, desequi2 and desequi3.

function [a,delta,o] = desequi(N,wp,rc,w,fast)

This function \([a,delta,o] = \text{DESEQUI}(N,wp,rc,w)\) designs an allpass filter of order \( N \), which has approximately equiripple linear phase from 0 to \( \theta_p \). When \( w \) is given, it is possible to design an allpass filter with arbitrary phase. If \( w \) is \( \pi \), this function returns the coefficients of the filter.

function \([a,\delta,0] = \text{DESEQUI}(N,wp,rc,w) \)

for \( i = 0:5, \)

\[ S_1 = S_1 - q^{-1}k \times \sin(k \times i \times \pi/N); \]

\[ S_2 = S_2 + q^{-1}k \times \cos(2 \times \pi \times k \times i \times \pi/N); \]

end;

\[ D = (q^{-1/4} \times S_1) / (1 + 2 \times S_2); \]

\[ v = \text{sqrt}(1 - r \times D^2); \]

\[ b = 2 \times v / (1 + D^2); \]

\[ a = (2 - b) / (2 + b); \]

end;

\[ a = \text{sort}(a); \]

\[ l = \text{length}(a); \]

\[ a_0 = \text{down}(a, 2); \]

\[ a_1 = \text{down}(a(2:1), 2); \]

\[ a_0 = \text{poly}(a_0); \]

\[ a_1 = \text{poly}(a_1); \]

end;

Zhang, X. and H. Ivakura


\[ w = \pi; \]

\( w = \pi; \)

\[ w = \pi; \]

\( w = \pi; \)

\[ w = \pi; \]

\( w = \pi; \)

\[ w = \pi; \]

\( w = \pi; \)

\[ w = \pi; \]

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)

\( w = \pi; \)
eps=1e-10;

w=eps;
P=zeros([N+1 N+1]);
a=zeros([1 N+1]); a(1)=1; a=a';
lser=1000;
while lser>eps,
    %Bepaal P en Q
    for c=1:N+1, P(:,c)=((c-1-N/2)*w-rc*w/2); end;
    Q=P;
    P=sin(P); Q=cos(Q);
    for r=2:2:N, Q(r,:)=-Q(r,:); end;
    D=P-'(-1)*Q;
    %Bepaal iteratief de coefficients a
    lser=1000;
    while lser>eps,
        a=a*D*a;
        %Bepaal eigenvector van de maximale eigenvector
        %z.b.v. de power methode
        delta=a(1);
        if delta==0 | delta==NaN, a=aold; break; end;
        a=a/delta;
        dif=aold-a;
        lser2=dif'*dif;
        end;
        delta=1/delta;
        if nargin<5,
            t=[(0:wp/255:wp)];
        if vs<pi, th=[(ws:ws/255:pi)]; end;
            t=[t;th];
            [n,d]=allpass(a);
            [h,wn]=freqz(n,d,t); p=unwrap(angle(h));
            plot(t/pi,abs(p-t-p));
            xlabel('Normalized frequency');
            ylabel('Phase error (rad)');
        hold off
    end;

end;

%Bepaal de maxima
w=eps;
for k=1:N+1,
    lser=1000;
    while lser>eps,
        afg=abs(d1(a,k),rc)/d2(a,o(k)));
        if afg>maxafg/8/k, break
        else
            oold=o(k);
            o(k)=o(k)-d1(a,o(k),rc)/d2(a,o(k));
            e=o(k)-oold;
            lser=e*e;
        end;
    end;
end;

%Delete enkele maxima
o=del(o,a,ep,vs,rc);
end;

%Bepaal verschuiving van de maxima
e=w-o;

lser1=e*e';

w=0;
end;

delta=max(abs(rc*t-p));
for k=1:length(v), x(k)=diff2(a,w(k)); y(k)=diff1(a,w(k)); end,[x;y],pause end;

function [a,del,co]=desequi2(N,ep,rc,b,v)

43
functie (a,delta,o)=DESEQUI2(N,vp,rc,b,w)
ontvanger

een allpass

Xfilter van orde N, die met een andere allpass filter in cascade een

Xphasekarakteristiek krijgt die van 0 tot wp ongeveer (equiripple) een

Xlimsere fase met een afgeleide rc heeft.

%b de noemer van de allpass filter dat in cascade komt
%a is de noemer van de ontworpen allpass filter
%delta is de maximale afwijkning van de rechte lijn met afgeleide rc
%0 is een vector die de locaties van de maxima aangeeft

%w kan eventueel als initialisatie meegegeven worden voor de (verwachte)
%locaties van de maxima. Hierdoor kan de ontwerpprocedure versneld worden.
%Debe lengte van v moet N+1 zijn.
%De ontwerpmethode staat beschreven in (met enkele modificaties):
%Zhang, X and H. Iwakura
%Design of digital allpass networks based on the eigenvalue problem
%Electronics and communications in Japan, part 3,
%Vol. 77 (1994), No. 4, p.99-109

b=b';
w=pi;
if nargin==4,
   w=[wp/N:wp*(1-1/N)/N];
X=['(c^2+1)^N/2];
X=log(N+2); Xk=2:N+2, w1(k-1)=vp*log(k)/C; end;
X=vp=log(w1)/C;
else,
   w=[wp/N:wp*(1-1/N)+2/N:wp];
   vh=(vs:(pi-vs)*2/(N-1):pi); end;
maxafg=wp/(N+1)/2;
es=1e-10;

a=w;
P=zeros([N+1 N+1]);
a=zeros([1 N+1]); a=a';
lser1=1000;
while lser1>1e-10,
XZBereken P en Q
for c=1:N+1, P(:,c)=((c-1-N/2)c-v-(rcv-vals(b,w))/2); end;
Q=Q;
for r=2:2:N, Q(r,: )=-Q(r, :); end;
D=P-(-1).Q;
%Bepaal iteratief de coëfficiënten a
% [V,E]=eig(D),pause
lser2=1000;
while lser2>eps,
XZBepaal eigenvector van de maximale eigenvector
Zm.b.v. de power methode

delta=a(1);
if delta==0 | delta==NaN, a=aold; break; end;
a=a/delta;
dif=aold-a;
lser2=dif'.dif;
end;
delta=1/delta;
t=[0:wp/255:wp];
[n,d]=allpass(a);
[nb,db]=allpass(b);
[hb,wn]=freqz(nb,db,t); pa=angle(h);
plot(t,abs(p)); pause(0.1);
hold on
Xfor k=1:length(w),
%plot([w(k),w(k)],[0 .1])
%end,pause,hold off
%Bepaal de maxima
ows;
for k=1:N,
lser=1000;
while lser>eps,

afg=abs(dlb(a,b,rc,o(k))/d2b(a,b,o(k)));
if afg=maxafg/k/2, break
else

wold=o(k);
o(k)=o(k)-dlb(a,b,rc,o(k))/d2b(a,b,o(k));
e=o(k)-wold;
lser=e*e;
end;
end;
end;
End;
End;
O=peak2(a,b,wp,ws,delta,o,rc);
%hold on
%for k=1:length(o),
% plot([o(k),o(k)],[O max(abs(p))].'r')
%end;
%hold off %pause
if length(o)<(N+1), o(N+1)=wp; end;

%Delete enkele maxima
O=del2(o,a,b,wp,ws,rc);
%Bepaal verschuiving van de maxima
ows=O;
lserl=a*a';
wo=O;
end;
delta=max(abs(p));
%for k=1:length(w), x(k)=diff2(a,w(k)); y(k)=diff1(a,w(k)); end,[x;y],pause end;

function [a,delta,o]=desequil(N,wp,rc,nH,dH,w)

w=pi;
if nargin==5,

w=[(wp/N:wp*(1-1/N)/N:wp)];
End;
maxafg=wp/(N+1)/2;
eps=le-10;
Ow=
P=zeros(N+1 N+1);
a=zeros([1 N+1]); a(1)=1; a=a';
lserl=1000;
while lserl>le-10,

%Bereken P en Q
h=freqz(nH,dH,w/2); x=unwrap(angle(h));
for c=1:N+1, P(:,c)=((c-1-N/2)*w-(rc+w-x)/2)'; end;
Q=P;
P=cos(Q);
for r=2:2:N, Q(r,:)=Q(r,:); end;
D=P'*(-1)*Q;
%Bepaal iteratief de coefficienten a
% [V,E]=eig(D),pause
lser2=1000;
while lser2>eps,
aold=a;

%Bepaal eigenvector van de maximale eigenvector
%z.m.b.v. de power methode

delta=a(1);
if delta=0 | delta==NaN, a=aold; break; end;
a=a/delta;
dif=aold-a;
lser2=dif'*dif;
end;
delta=1/delta;

[t,n,d]=allpass(a);
[b,hn]=freqz(n,d,t); pa=angle(h);
[hh,hn]=freqz(nH,dH,t/2); pb=unwrap(angle(hh)); pa=unwrap(pa-rc*t+pb);
hold off
for k=1:length(w),
    subplot(1,1,k),
    %plot([v(k),w(k)],0 .1)
end, pause, hold off

X Bepaal de maxima
o=v;
for k=1:N,
    lser=1000;
    while lser>eps,
        afg=abs(dlc(a,nH,dH,rc,o(k))/d2c(a,nH,dH,o(k)));
        if afg>maxafg/k/2, break
        else
            o(k)=o(k)-dlc(a,nH,dH,rc,o(k))/d2c(a,nH,dH,o(k));
            o(k)=o(k)-ddc(a,nH,dH,rc,o(k))/d2c(a,nH,dH,o(k));
            lser=e/e;
        end;
    end;
end;
o=peak3(a,nH,dH,vp,vs,delta,o,rc); hold on
for k=1:length(o),
    plot([o(k),o(k)],0 max(abs(p)), 'r')
end;
hold off, pause
if length(o)<(N+1), o(N+1)=vp; end;

% Delete enkele maxima
o=del3(o,a,nH,dH,vp,ws,rc);
% Bepaal verschuiving van de maxima
e=peak3(a,nH,dH,vp,ws,delta,o,rc);

function p=peak(a,wp,ws,delta,pold,rc)
% Deze functie hoort bij de functie DESDEQ, en bepaalt de maxima van het
% verschil tussen de fase karakteristiek en de rechte.
% Het algoritme staat beschreven in:
% J.A. Antoniou
% Accelerated procedure for the design of equiripple nonrecursive digital filters
% IEEE proc.,
% Vol. 129 (1982), Part C, No. 1, p.1-10

r=length(a);
I=(wp-pi/2)/r; m=floor(wp/I*0.5); m1=floor((pi-ws)/I*0.5);
if m=0, Ip=wp/m; end;
if m1=0, Im=(pi-ws)/m1; end;
if Ip=64, ia=ia/16;
p(1)=0;
% Step 1
i=1;j=1;d=abs(delta);
G2=dfe2(a,0,rc);
M=a(0,rc);
if G2<0 & (M>d),
p(i)=0; i=i+1;
end
%Step 2 De passband
while (j<=m),
  if (w'=0) & (G2=0), % (a)
    while 1,
      G1=diffe1(a,w,rc); M=e(a,w,rc);
      if (G1>0) & (M>d), break;
      elseif abs(G1)<eps, (i)=w; i=i+1; break;
      elseif (w=wp) & (G1<0) & (M>d),
        p(i)=w; i=i+1; break;
      elseif ((G1>0) & (w'=0)) | ((G2>0) & (w=0)), % (d)
        Mold=0;
      while (M>Mold) & (w<wp),
        Mold=M; v=v+ip; M=e(a,v,rc);
      end;
      if (w=wp),
        v=v-ip; p(i)=wp; i=i+1; break; % (e)
      end;
    end;
  end;
end;
end;

%Step 3
if m'=0,
  if (pold(m)=wp) & (p(i-1)=wp), % (a)
    w=wp;
    G1=diffe1(a,w,rc); M=e(a,w,rc);
    if (G1>0) & (M>d), p(i)=wp; i=i+1; end; % (b)
    end;
end;

%Step 4
if ml'=0,
  if pold(m+1)=vs, % (a)
    w=vs;
    G1=diffe1(a,w,rc); M=e(a,w,rc);
    if (G1<0) & (M>d),
      p(i)=wp; i=i+1;
      end;
    end;
end;

%Step 5 Stopband
step7=0;
while j<e,
  if (w=wp) & (G2=0), % (a)
    w=pold(j); G2=diffe2(a,w,rc);
    if (w'=0) | (G2>0), % (b)
      if (w=wp) & (G2<0) & (M>d),
        step7=1; break;
      elseif (w=wp) & (G2>0) & (M>d),
        p(i)=wp; i=i+1; step7=1; break;
      else,
        G1=diffe1(a,w,rc);
        if (G1>0),
        Mold=0;
        % (d)
      end;
    end;
  end;
end;
while (M>Mold) & (w<pi),
    w+=ia; Mold=M; M=e(a,w,rc);
end;
if w'=pi,
    w+=ia; p(i)=w; i=i+1; break; \%(c)
end; \%(e)
elseif (G1<0) & (w='pi) | ((G2>0) & (w=pi)),
    Mold=0;
while (M>Mold) & (w>ws),
    w+=ia; Mold=M; M=e(a,w,rc);
end;
if w'=ws,
    w+=ia; p(i)=w; i=i+1; break; \%(e)
else break;
end; \%(e)
end; \%(d)
end;
if step7==1, break; end;
end; \%(c)
end;
j=j+1;
end; \%(b)
end; \%(a)

Xstep 6
% if step7=1,
% if pold(r)=pi,
% w=pi; G2=diffe2(a,w,rc); M=e(a,w,rc);
% if (G2<0) & (M>d),
% p(i)=pi; i=i+1;
% end;
end;
end;

Xstep 7
function p=peak(a,b,vp,vs,delta,pold,rc)
% Accelerated procedure f l the design of equiripple nonrecursive digital filters
%A. Antoniou
r=length(a);
I=(vp+pi-vs)/r; m=floor(vp/I+0.5); ml=floor((pi-vs)/I+0.5);
    if m=0, Ip=vp/m; end;
    if ml=0, la=(pi-vs)/ml; end;
    Ip=Ip/32; ia=la/16;
p(i)=0;

Xstep 1
i=1; j=1; d=abs(delta);
G2=diffe2b(a,b,0,rc);
M=e2(a,b,0,rc);
if (G2<0) & (M>d),
p(i)=0; i=i+1;
end;

Xstep 2 De passband
while (j<m),
    w=pold(j); G2=diffe2b(a,b,w,rc);
if (w=0) | (G2=0),
    while 1,
        G1=diffe1b(a,b,w,rc); M=e2(a,b,w,rc);
        \%\(c\)
        if (w=up) & (G1>0) & (M<d), break;
        elseif abs(db(a,b,rc,w))<eps, p(i)=w; i=i+1; break;
        elseif (w=wp) & (G1>0) & (M>d),
            p(i)=w; i=i+1; break;
        elseif ((G1>0) & (w=0)) | ((G2>0) & (w=0)), \%\(d\)
            Mold=0;
        while (M>Mold) & (w<vp),
        %\(a\)
        %\(b\)
        \%\(c\)
    end; \%\(e\)
end; \%\(d\)
end; \%\(b\)
end; \%\(a\)

48
Hold\[H: w=w+ip; H=e2(a,b,w,rc):
end;
if \(w=wp,\)
w=w-ip; p(i)=w; i=i+1; break; \(\%(a)\)
end;
elseif \(G1<0),\)
Mold=0;
while \(M>Mold \& \ w=0 \& \ w=wp(1),\)
Mold=M; w=w-ip; H=e2(a,b,w,rc);
end;
if \(w=0 \mid w=p(1),\) break;
else,
w=w-ip; p(i)=w; i=i+1; break;
end;
end;
ej=j+1;
end;
end;

\%Step 3
if \(m=0,\)
if \((\text{old}(m))^2=wp \& (p(i-1))^2=wp),\)
\(w=wp;\)
G1=diffe1b(a,b,w,rc); H=e2(a,b,w,rc);
if \((G1>0) \& (M>d),\)
p(i)=wp; i=i+1; end;
end;

\%Step 4
if \(m=0,\)
if \(\text{old}(m+1)^2=wp,\)
w=wp;
G1=diffe2b(a,b,w,rc); H=e2(a,b,w,rc);
if \((G1<0) \& (M>d),\)
p(i)=wp; i=i+1;
end;
end;

\%Step 5 Stopband
step7=0;
while \(j<r,\)
w=old(j); G1=diffe1b(a,b,w,rc); H=e2(a,b,w,rc);
if \((w=wp) \mid (G1>0) \mid (M>d),\)
if \((w=wp) \& (G1<0) \& (M=d),\)
p(i)=wp; i=i+1; step7=1; break;
else,
G1=diffe2b(a,b,w,rc);
if \((G1>0) \& (M>d),\)
Mold=0;
while \(M>Mold \& \ w>wp1,\)
w=w+ia; Mold=M; H=e2(a,b,w,rc);
end;
if \(w=pi,\)
w=w-ia; p(i)=w; i=i+1; break; \(\%(e)\)
end; \(\%(e)\)
elseif \((G1<0) \& (w=pi)) \mid ((G1>0) \& (w=pi)),\)
Mold=0;
while \(M>Mold \& \ w>wp,\)
w=w-ia; Mold=M; H=e2(a,b,w,rc);
end;
if \(w=ws,\)
w=w+ia; p(i)=w; i=i+1; break; \(\%(e)\)
else break;
end; %
end; %

if step7==1, break; end;
end; %

j=j+1;
end; %
%

%Step 6
%If step7==1,
% if pold(r)==pi,
%x w=pi; G2=diffe2b(a,b,w,rc); M=ae2(a,b,w,rc);
% else if (G2>0) & (M>d),
% p(i)=pi; i=i+1;
% end;
% end;
%
end;

function p=peak(a, n, den, wp, ws, delta, pold, rc)
%Accelerated procedure for the design of equiripple nonrecursive digital filters
%A. Antoniou

r=length(a);
I=evp+pi-vs)/r;
m=floor(wp/I+0.6); m1=floor((pi-vs)/I+0.5);
if m==0, Ip=wp/m; end;
if m1==0, Ia=(pi-vs)/m1; end;
Ip=Ip/128; Ia=Ia/16;
p(1)=0;

%Step 1
i=1; j=1; d=abs(delta);
G2=diffe2c(a, n, den, 0, rc);
M=ae3(a, n, den, 0, rc);
if (G2<0) & (M>=d),
else if (v==vp) & (G1>O) & (M>d),
Kold=O;
while (M>Mold)
if v==p(l), break;
else, v=v-i; p(i)=w; i=i+1; break;
end;
end;
else if (v==vp) & (G1>O) & (M>d),
Kold=O;
while (M>Mold)
if v==p(l), break;
else, v=v-i; p(i)=w; i=i+1; break;
end;
end;
else, v=v-i; p(i)=w; i=i+1; break;
end;
end;

%Step 2 De passband
while (j<=m),
wpold(j); G2=diffe2c(a, n, den, w, rc);
if (v==0) | (G2==0),
while 1,
G1=diffe1c(a, n, den, w, rc);
M=ae3(a, n, den, w, rc);
% (c)
if (v==wp) & (G1>O) & (M>d), break;
elseif abs(dic(a, n, den, rc, v)<eps, p(i)=w; i=i+1; break;
elseif (v==wp) & (G1>O) & (M>d),
Kold=O;
while (M>Mold)
if v==p(l), break;
else, v=v-i; p(i)=w; i=i+1; break;
end;
end;
elseif G1<0,
Mold=0;
while M>Mold & w==0 & w==p(l),
Mold=M; w=w-1p; M=ae3(a, n, den, w, rc);
end;
if v==0 | v==p(l), break;
else, w=w-1p; p(i)=w; i=i+1; break;
end;
end;

50
XStep 3
if m=0,
if (pold(m)=wp) & (p(i-1)=wp),
   v=wp;
   Gl=diffe1c(a,n,den,v,rc); M=e3(a,n,den,v,rc);
   if (Gl>0) & (M>d), p(i)=wp; i=i+1; end;
   % (b)
end;
end;

XStep 4
if m=0,
if pold(m+1)=vs, % (a)
   v=vs;
   Gl=diffe1c(a,n,den,v,rc); M=e3(a,n,den,v,rc);
   if (Gl<d) & (M>v),
   p(i)=v; i=i+1;
end;
end;

XStep 5 Stop band
step7=0;
while j<r,
   if pold(j)=wp, % (a)
      v=pold(j); Gl=diffe1c(a,n,den,v,rc); M=e3(a,n,den,v,rc);
      if (Gl>0) & (M>d), % (b)
         if v=vs, % (c)
            if (Gl<d) & (M>v),
               p(i)=v; i=i+1;
            else
               Gl=diffe1c(a,n,den,v,rc);
               if Gl>0,
                  Mold=0;
                  while (M>Mold) & (v<pi),
                     w=v; Mold=M; M=e3(a,n,den,v,rc);
                     end;
                     if v=pi,
                        w=v; p(i)=w; i=i+1; break; % (d)
                     else
                        Gl=diffe1c(a,n,den,v,rc);
                        if Gl>0, % (d)
                           Mold=0;
                           while (M>Mold) & (v<pi),
                              w=v; Mold=M; M=e3(a,n,den,v,rc);
                              end;
                              if v=pi,
                                 w=v; p(i)=w; i=i+1; break; % (d)
                              else
                                 Gl=diffe1c(a,n,den,v,rc);
                                 if Gl>0, % (d)
                                    Mold=0;
                                    while (M>Mold) & (v<pi),
                                       w=v; Mold=M; M=e3(a,n,den,v,rc);
                                       end;
                                       if v=pi,
                                          w=v; p(i)=w; i=i+1; break; % (d)
                                       else break;
                                       end; % (d)
                                  end; % (d)
                              end; % (d)
                          end; % (d)
                      end;
                      step7=1; break;
                   end;
                end;
            end;
         else
            Gl=diffe1c(a,n,den,v,rc);
            if Gl<0, % (d)
               Mold=0;
               while (M>Mold) & (v<vs),
                  w=v; Mold=M; M=e3(a,n,den,v,rc);
                  end;
                  if v=vs,
                     w=v; p(i)=w; i=i+1; break; % (e)
                  else break;
                  end; % (e)
              end;
          end;
      end;
   else
      if step7=1, break; end;
      % (c)
   end;
   j=j+1;
end; % (b)
end; % (a)

XStep 6
% if step7=1,
% if pold(r) = pi,
% w = pi; G2 = d2(a, b, v, rc); M = e2(a, b, v, rc);
% if (G2 < G) & (v = v'),
% p(i) = pi; i = i + 1;
% end;
% end;
%end;

%Step 7
end;

function o = delete(w, a, b, wp, vs, rc)
%Deze functie hoort bij de functie DESEQU. en verwijdert eventueel de
%overbodige maxima.

% Het algoritme staat beschreven in:
IA, Antoniou
%Accelerated procedure for the design of equiripple nonrecursive digital filters
%IEE Proc. .
%Vol. 129 (1982), Part G, No. 1, p.1-10

o = w; l = length(w); al = length(a);
while l > al,
mu = 1; Ep = 0; Ea = 0;
while w(mu) < wp, mu = mu + 1; end;
if w(mu) > wp, mu = mu - 1; end;
for k = 1:mu, Ep = Ep + e(a, b, v(k), rc); end;
if w = pi,
for k = mu + 1:1, Ea = e2(a, b, v(k), rc); end;
else Ea = 1000;
Ep = Ep / (mu + 1);
if Ep < Ea,
for l = 1:mu, er(l) = e2(a, v(l), rc); end;
[s, i] = min(er);
ob = o(1:i-1); o = [o1 oh];
else o = o(1:i-1); end;
l = l - 1;
end;
end;

function o = delete3(a, n, d, wp, vs, rc)
o = w; l = length(w); al = length(a);
while l > al,
mu = 1; Ep = 0; Ea = 0;
for k = 1:mu, Ep = Ep + e3(a, n, d, v(k), rc); end;
if w = pi,
for k = mu + 1:1, Ea = e3(a, n, d, v(k), rc); end;
else Ea = 1000;
Ep = Ep / (mu + 1);
if Ep < Ea,
for l = 1:mu, er(l) = e3(a, v(l), rc); end;
[s, i] = min(er);
ob = o(1:i-1); o = [o1 oh];
else o = o(1:i-1); end;
l = l - 1;
end;
end;

function o = delete(v, a, b, wp, vs, rc)
o = w; l = length(v); al = length(a);
while l > al,
mu = 1; Ep = 0; Ea = 0;
for k = 1:mu, Ep = Ep + e(a, b, v(k), rc); end;
if w = pi,
for k = mu + 1:1, Ea = e2(a, b, v(k), rc); end;
else Ea = 1000;
Ep = Ep / (mu + 1);
if Ep < Ea,
for l = 1:mu, er(l) = e2(a, v(l), rc); end;
[s, i] = min(er);
ob = o(1:i-1); o = [o1 oh];
else o = o(1:i-1); end;
l = l - 1;
end;
end;
function G1=d1(a,w,rc);
    b=a';
    l=length(a)-1;
    kw=[0:1:1];
    t=[0:1:1];
    v=ones(1,1+l);
    CO=b.*cos(kw);
    SO=b.*sin(kw);
    Cl=t*CO';
    S1=t*SO';
    CO=v*CO';
    SO=v*SO';
    G1=(-1+2*(CO*Cl+S0*Sl)/(CO^2+S0-2)-rc);
end;

function G2=d2(a,w);
    b=a';
    l=length(a)-1;
    kw=[0:1:1];
    t=[0:1:1];
    v=ones(1,1+l);
    x=t-2,
    CO=b.*cos(kw);
    SO=b.*sin(kw);
    Cl=t*CO';
    S1=t*SO';
    CO=v*CO';
    SO=v*SO';
    G2=2*(-CO-3*S2-C0*Cl+S0*Sl)/(CO^2+S0-2^2)-2;
end;
function M=E(a, v, rc, absolute)
    b=a';
    l=length(a); c=(l-1)/2;
    t=((0:1:1-l)-c)*v-theta(w)/2;
    SO=sin(t); CO=cos(t);
    CO=CO';
    SO=SO';
    SO=0; CO=0;
    for k=1:l,
        SO=SO+a(k)*sin((k-l)-c)*v-rc*v/2);
        CO=CO+a(k)*cos((k-l)-c)*v-rc*v/2);
        end;
    M=(2*atan(SO/CO));
    if nargin==3, M=abs(M); end;
    end;

function M=E2(a, b, v, rc, absolute)
    l=length(a); c=(l-1)/2;
    SO=0; CO=0;
    x=evalplp(b, v);
    for k:l:1.
        SO:SQ+a(k)*sin((k-1)-c)*v-(rc*v-x)/2);
        CO=CO+a(k)*cos((k-1)-c)*v-(rc*v-x)/2);
        end;
    M=(2*atan(SO/CO));
    if nargin==4, M=abs(M); end;
    end;

function M=E3(a, n, d, v, rc, absolute)
    l=length(a); c=(l-1)/2;
    SO=0; CO=0;
    h=freqz(n, d, [0 v/2]);
    x=angle(h(2);
    for k=1:1,
        SO=SO+a(k)*sin(k)*v-(rc*v-x)/2);
        CO=CO+a(k).cos(k)*v-(rc*v-x)/2);
        end;
    M=(2*atan(SO/CO));
    if nargin==5, M=abs(M); end;
    end;

function G1=diffEl(a, v, rc);
    b=a';
    l=length(a)-1;
    XC0=0; SO=0; Cl=0; Sl=0;
    %for k=0:1,
    % ac=a(k+1)*cos((k)*v); as=a(k+1)*sin((k)*v);
    % CO=CO+ac;
    % SO=SO+as;
    % Cl=Cl+(k)*ac;
    % Sl=Sl+(k)*as;
    %end;
    XC1, Sl, CO, SO.pause
    %G1=-((1+2*(CO*Cl+SO*Sl)/(CO^2+SO^2)+1/2);

    k=v*([0:1:1]);
    t=([0:1:1]);
    v=ones([1 l+1]);
    CO=b.*cos(k);
    CO=CO';
    Cl=t*CO';
    Sl=t*SO';
    CO=CO';
    SO=SO';
    G1=sign(e(a, v, rc, 1))*(-1+2*(CO*Cl+SO*Sl)/(CO^2+SO^2)-rc);
    end;

function G2=diffe2(a, v, rc);
    b=a';
    l=length(a)-1;
    kw=[0:1:1];

t=([0:1:1]);
v=ones([1 1+1]);
x=t.^2;
CO=b.*cos(k);
SO=b.*sin(k);
Cl=t*CO';
Sl=t*SO';
C2=x*CO';
S2=x*SO';
CO=v*CO';
SO=v*SO';
G2=G2/(CO-2+S0-2)-2;
G2=sign(e(a,v,rc,l»*G2;
XCO=0;SO=0;Cl=0;Sl=0;C2=0;S2=0;
X1=length(a);
Xfor k=0:1:-1,
X  ac=a(k+1)*cos(k*w);
X  as=a(k+1)*sin(k*w);
X  CO=CO+ac;
X  SO=SO+as;
X  Cl=Cl+k*ac;
X  Sl=Sl+k*as;
X  C2=C2+k-2*ac;
X  S2=S2+k-2*as;
Xend;
XG2=2*(-CO-3*S2-CO*S2-2*CO-3*S2-CO*S2*SO-2-2*Cl*Sl*CO-2-2*Cl*SO-2*Sl+2*SO*Sl-2*CO+SO*C2*CO-2+S0-3*C2-2*CO*Cl-2*SO):
XG2=G2/(CO-2+S0-2)-2;
XG2=sign(E(a,w,1»*G2;
end;
function Gl=diffelb(a,b,w,rc):
Gl=e2(a,b,w,rc,l);
if abs(Gl)>eps,
Gl=sign(Gl)*(dl(a,w,rc)+dl(b,w,0));
else Gl=-1;
end;
end;
function G2=diffe2(a,b,w,rc);
G2=(e2(a,b,w,rc,l»;
if abs(G2)>eps,
G2=(d2b(a,b,w)+d2b(a,b,w,0));
else G2=-1;
end;
end;
function Gl=diffelc(a,n,d,w,rc):
Gl=e3(a,n,d,w,rc,l);
if abs(Gl)>eps,
Gl=sign(Gl)*(d1(a,w,rc)+d1h(n,d,w));
else Gl=-1;
end;
end;
function G2=diffe2c(a,n,d,w,rc);
G2=(e3(a,n,d,w,rc,l»;
if abs(G2)>eps,
G2=(d2a(a,v)+d2h(d,n,v))*sign(G2);
else G2=-1;
end;
end;
function Gl=evallp(a,w);
l=length(a)-1;
if l==0, Gl=1;
else,
for n=1:length(w),
x=w(n)*([0:1:1]);
A.3 Almost perfect filter bank design

The following algorithms desbank and desbank2 design the almost perfect filter banks. The algorithm desbank designs the filter bank such that the slope of the phase response of the cascade of the allpass filters $A_t(z)$ and $B_t(z)$ is approximately $-N_A - N_B + 1$ where the order of $A_t(z)$ and $B_t(z)$ are $N_A$ and $N_B$ respectively. If desired the order of $B_t(z)$ can be different from $N_A$. Notice that a minus sign can be necessary then. The algorithm desbank2 designs the filter bank such that $\angle(A_t(z)B_t(z)) \approx -N_A - N_B$ where the order of $A_t(z)$ and $B_t(z)$ are $N_A$ and $N_B$ respectively. Here it is also possible to change the order or $B_t(z)$. Notice that a minus sign can be necessary then.

function [A,C,B,\delta]=desbank(M,Na,Ns,Nb,vpb)
% Deze functie [A,B,C,\delta]=DESBANK(M,Na,Ns,Nb,wpb)
% Ontwerpt een almost perfect reconstruction filterbank.
% Deze functie geeft een beter resultaat dan DESBANK2.
% M is de helft van het aantal kanaal.
% Na zijn de ordes van de allpass filters van de analysebank.
% Nb zijn de ordes van de allpass filters van de synthesebank.
% Ns is de orde van de allpass filter in het halfband filter ontwerp.
% Vpb de stopband van de halfband begint bij wpb*pi.
% Deze allpass fasekarakteristieken van de analyse- en de synthesebank
% worden m.b.v. wpb bepaald.
%
ÌA bevat de M-1 x Na+1 matrix. De rijen bevatten de noemers van de
% allpass filters van de analysebank. AO is niet gegeven:
% deze is een delay.
ÌB bevat de 2 x Ns+1 matrix. De rijen bevatten de noemers van de
% allpass filters van de halfband filter.
ÌC bevat de M-1 x Ns+1 matrix. De rijen bevatten de noemers van de
% allpass filters van de synthesebank. CO is niet gegeven:
% deze is een delay.
Ìdelta bevat de grootste phasefout van de allpassfilters in de analyse-
% en synthesebank ontwerp.

subplot(111)
[A,B,C,delta1]=mband(Na,M,0.5*pi,2*pi*(1-vpb),Nb);
[C,delta2]=dessynth(Ns,M,vpb*pi,A);
d=[delta1 delta2];
delta=delta2;
end;

function [A,B,C,delta]=desbank2(H,Na,Ns,Nb,vpb)
%Deze functie (A,B,C,delta]=DESBANK2(H,Na,Ns,Nb,vpb)
%ontverpt een almost perfect reconstruction filterbank.
%(De functie DESBANK geeft een beter resultaat dan DESBANK2.)
%XM
%Na
%Ns
%Nb
%vpb
%De allpass phasekarakteristieken van de analyse- en de synthesebank
%vorden m.b.v. vpb bepaald.
%ÌA bevat de M-1 x Na+1 matrix. De rijen bevatten de noemers van de
% allpass filters van de analysebank. AO is niet gegeven:
% deze is een delay.
ÌB bevat de 2 x Ns+1 matrix. De rijen bevatten de noemers van de
% allpass filters van de halfband filter.
ÌC bevat de M-1 x Ns+1 matrix. De rijen bevatten de noemers van de
% allpass filters van de synthesebank. CO is niet gegeven:
% deze is een delay.
Ìdelta bevat de grootste phasefout van de allpassfilters in de analyse-
% en synthesebank ontwerp.

subplot(111)
[A,B,C,delta1]=mband(Na,M,0.5*pi,2*pi*(1-vpb),Nb);
[C,delta2]=dessynth2(Ns,M,vpb*pi,A);
d=[delta1 delta2];
delta=delta2;
end;

function [C,d]=dessynth(N,M,vp,A);
w=[(2:1:N+2)];
L=log(N+2);
w=log(w)/L;
l=length(A(1,:))-1;
[c,delta,v]=desequi2(N,vp,-(N+1)+1,A(1,:));
C=c';
D=delta;
for k=2:M-1,
k
[c,delta,v]=desequi2(N,vp,-(N+1)+1,A(k,:),w);
C=[C,c'];
D=[D;delta];
end;
d=max(D);
end;

function [C,d]=dessynth2(N,M,vp,A);
w=[(2:1:N+2)];
L=log(N+2);
w=log(w)/L;
A.4 Simulations of the filter bank design

With the following algorithm it is possible to simulate the designed filter bank from the previous section. With some modifications it is also possible to simulate filter banks which introduce phase distortion.

```
function [x1,mag,pe]=simul(R,A,C,H,type);

% with this function [x1,mag,pe]=SIMUL(R,A,C,H,type) is possible to simulate the filterbank
% which is designed by DESBANK2.
% A is the analysisbank matrix generated by DESBANK2.
% C is the synthesisbank matrix generated by DESBANK2.
% H is the halfband filter generated by DESBANK2. Another value is given.
% x1 is the input signal vector.
% mag is the amplitude difference of the input signal and the output signal.
% pe is the phase difference with respect to the ideal phase characteristic.
% This function will plot the last described.

[M,W]=size(A); N=M+1; R=N-1;
[MC,NC]=size(C); NC=NC-1;

x=zeros([1 R]);
for k=1:R,
    x=[cos((pi*(k-1/2))/M/2)*t),zeros([1 R-321])]+x;
end;
t=[(0:2/(R-1):2); zeros([1 R-2*M+1])];
x=[randn([1 R-4096]),zeros([1 R-4096])];
X=xft(x)/sqrt(R);
hold off
subplot(111)
plot(x)
xlabel('input signal (push a key)'),pause
plot(t,abs(X));
xlabel('FFT of the input signal (push a key)'),pause
xlabel('Simulating...'),pause(0.1)

% Add AO to A (AO=z^{-(N)})
AO=eye([1 N+1]);
if nargin==4,
    AO=eye([1 NC]);
else
    C=eye([1 NC+1]);
end;

% Add CO to C (CO=z^{-(NC)})
CO=eye([1 NC]);
```
if nargin==3, 
    H=[AO; A(M/2+1,:)]; 
end; 
NH=length(H(1,:))-1; 

%uitgangsvector y1: na H1(Z^M) 
dly=fliplr(eye([1 M+1])); 
[nh, dh]=allpassm(H(1,:),2*M); 
f1=filter(nh, dh, x); 
[y1, uitgangsvector y1: na H1(-Z^M)] 
y1=(f1-f2)/2; hold off 

%uitgangsvector y2: na H1(-Z^M) 
y2=(f1-f2)/2; hold off 

%delay z^(-M) 

%vermenigvuldig A_sigma met z^(-M) 

%Bereken uitgangsmatrix van de analysebank voor y1 met down- en upsamplers 
for k=1:M, 
    %dly=fliplr(eye([1 k])); 
    [n, d]=allpassm(A(k,:), M, 1); 
    %n=conv(n, dly); 
    res=delay(y1, k-1); 
    res=filter(n, d, res); 
    res=up(down(res, M, M)); 
    %h1=fft(res); plot(t/pi, abs(h1), 'g'); pause 
    Y1=[Y1; res]; 
end; 

%Bereken uitgangsmatrix van de analysebank voor y2 down- en upsamplers 
for k=1:M, 
    %dly=fliplr(eye([1 k])); 
    [n, d]=allpassm(A(k,:), M, 1); 
    %n=conv(n, dly); 
    res=delay(y2, k-1); 
    res=filter(n, d, res); 
    res=up(down(res, M, M)); 
    %h1=fft(res); plot(t/pi, abs(h1), 'g'); pause 
    Y2=[Y2; res]; 
end; 

%Bereken uitgangsmatrix van de synthesebank tot de delayline voor Y1 
for k=1:M, 
    [n, d]=allpassm(C(k,:), M); 
    Y1(k,:)=filter(n, d, Y1(k,:)); 
end; 

%Bereken uitgangsmatrix van de synthesebank tot de delayline voor Y2 
for k=1:M, 
    [n, d]=allpassm(C(k,:), M); 
    Y2(k,:)=filter(n, d, Y2(k,:)); 
end; 

%Bereken uitgang van de delayline voor Y1 
xapp1=zeros([1 R]); 
for k=M:-1:1, 
    xapp1=xapp1+delay(Y1(M-k+1,:), k-1); 
end; 

%Bereken uitgang van de delayline voor Y2 
xapp2=zeros([1 R]); 
for k=M:-1:1, 
    xapp2=xapp2+delay(Y2(M-k+1,:), k-1); 
end; 

if nargin==4, 
    x1=xapp1*(-1)^(NC+N)*xapp2; 
else 
    x1=xapp1*(-1)^(NC+N)*xapp2; 
end;
\[ p = \text{angle}(X); \]
\[ g = \text{fft}(x_1) / \sqrt{R}; \]
\[ d = \text{flip}(\text{eye}([1 H (N+NC-1)+NH*2*M+(M-l)+1])); \]
\[ dly = [H (N+NC-1)+NH*2*M+(M-1)]; \]
\[
\text{if nargin} = 6, \\
\text{dly} = \text{dly} + M; \\
\text{end}; \]
\[ dly; \]
\[ xd = \text{delay}(x, dly); \]
\[ \text{subplot}(122); \]
\[ \text{plot}(\text{abs}(x_1), dly); \]
\[ \text{g} = \text{fft}(x_1) / \sqrt{R}; \]
\[ p = \text{angle}(X_d); \]
\[ \text{subplot}(122); \]
\[ \text{plot}(\text{abs}(g - x_1)); \]
\[ \text{xlabel}(’\text{Difference delayed input and output}’); \]
\[ e = \text{abs}([g - x_1]); \]
\[ \text{mag} = 20 * \log10(e); \]
\[ \text{subplot}(121); \]
\[ \text{plot}(\text{abs}(g - x_1)); \]
\[ \text{ylabel}(’\text{Normalized frequency}’); \]
\[ \text{ylabel}(’\text{Amplitude error (dB)}’); \]
\[ \text{hold off}; \]
\[ \text{end}; \]

A.5 Distortion function of the filter bank

With the following algorithm it is possible to determine the distortion function of the almost perfect filter bank designed by desbank or desbank2 without the down- and upsamplers. Thus, the transfer function of the filter bank if there is no aliasing.

```
function dummy=transf(A,C,H,type);
  % Deze functie dummy=TRANSF(A,C,H,type) bepaalt de overdracht van de
  % almost perfect reconstruction filterbank die ontworpen is door
  % DESBANK of DESBANK2, maar zonder down- en upsamplers.
  % A bevat de analysebankmatrix ontworpen door DESBANK of DESBANK2.
  % C bevat de synthesebankmatrix ontworpen door DESBANK of DESBANK2.
  % H bevat de matrix van de halfband filter ontworpen door DESBANK of DESBANK2.
  % Neem type=1 als de filterbank is ontworpen door DESBANK2. Anders niets meegeven.

  M=4096;
  [M,N]=size(A); M=M+1; N=N-1;
  [MC,NC]=size(C); NC=NC-1;
  NH=length(H(1,:))-1;

  % Voeg AO aan A toe (AO=z^(-N))
  AO=eye([1 N+1]);
  A=[AO; A];

  Hp=zeros([K 1]); Hn=zeros([K 1]);
  if nargin=4,
  % Voeg CU aan C toe (AO=z^(-NC))
  CC=eye([1 NC+1]);
  C=[C0; C1];
  [n1,d1]=allpassm(A(1,:),M);
  [n2,d2]=allpassm(C(1,:),M);
  Hn1=freqz(n1,d1,K);
  Hn2=freqz(n2,d2,K);
  Hp=Hp+Hn1.*Hn2;

  [n1,d1]=allpassm(A(1,:),M,1);
  [n2,d2]=allpassm(C(1,:),M,1);
  Hn1=freqz(n1,d1,K);
  Hn2=freqz(n2,d2,K);
  Hn=Hn+Hn1.*Hn2;
```
else
    CO=[eye(1 NC)];
    C=[1 C0;C ];
    [n1,d1]=allpassm(A(1,:),M);
    n2=flipud(eye([1 M*(NC-1)+1]));
    d2=1;
    Hn1=freqz(n1,d1,K);
    Hn2=freqz(n2,d2,K);
    Hp=Hp+Hn1.*Hn2;

    [n1,d1]=allpassm(A(1,:),M,1);
    if rem(NC-1,2)==1, n1=-n1; end;
    Hn1=freqz(n1,d1,K);
    Hn2=freqz(n2,d2,K);
    Hn=Hn+Hn1.*Hn2;
end;

    [n1,d1]=allpassm(H(1,:),2*M);
    [n2,d2]=allpassm(H(2,:),2*M);
    n2=zeros([1 K]) n2];
    Fl1=freqz(n1,d1,K);
    [Fl1, w]=freqz(n2,d2,K);
    H1=(Fl1+Fl2)/2;
    H2=(Fl1-Fl2)/2;
    clear Fl1 Fl2

for k=2:K,
    [n1,d1]=allpassm(A(k,:),M);
    [n2,d2]=allpassm(C(k,:),M);
    Hn1=freqz(n1,d1,K);
    Hn2=freqz(n2,d2,K);
    Hp=Hp+Hn1.*Hn2;

    [n1,d1]=allpassm(A(k,:),M,1);
    [n2,d2]=allpassm(C(k,:),M,1);
    Hn1=freqz(n1,d1,K);
    Hn2=freqz(n2,d2,K);
    Hm=Hm+Hn1.*Hn2;
end;

Hp=Hp.*H1; Hm=Hm.*H2;

if nargin==4,
    T=Hp+(-1)^(N+NC)*Hm;
else
    T=Hp-(-1)^(N+NC)*Hm;
end;

Hn1=freqz(fliplr(eye([1 M]).1,K);
T=T.*Hn1/M;

subplot(121)
e=20*log10(abs(1-abs(T)));
plot(w/pi,e);
xlabel('Normalized frequency');
ylabel('Amplitude error (dB)');

d=M*(N+NC-1)+KH*2*M*(M-1);
if nargin==4,
    d=d*M;
end;
p=unwrap(angle(T));
pe=(p.*.e);

subplot(122);
plot(w/pi,pe);
xlabel('Normalized frequency');

61
A.6 The peak aliasing distortion of the filter bank

With the following algorithm it is possible to determine the peak aliasing distortion of the filter bank.

```matlab
function dummy=peakali(A,C,H,type);

% Deze functie dummy=PEAKALI(A,C,H,type) bepaalt de 'peak aliasing
% distortion' van de almost perfect reconstruction filterbank die
% ontworpen is door DESBANK of DESBANK2.
% 
% A bevat de analysebankmatrix ontworpen door DESBANK of DESBANK2.
% C bevat de synthesebankmatrix ontworpen door DESBANK of DESBANK2.
% H bevat de matrix van de halfband filter ontworpen door DESBANK of DESBANK2.
% Neem type=1 als de filterbank is ontworpen door DESBANK2. Anders niets meegeven.

K=2048;
[M,N]=size(A); M=M+1; N=N-1;
[NC,NC]=size(C); NC=NC-1;

%Voeg AO aan A toe (AO=z^(-N))
AO=(eye([1 N+1]));
A=[AO; A];

if nargin==3,
%Voeg CO aan C toe (AO=z^(-NC))
CO=eye([1 NC]));
C=[CO;C ];
else
CO=eye([1 NC+1]));
C=[CO;C ];
end;

%H(z^M) en H(-z^M)
W=exp(-j*2*pi/K);
[n1,d1]=allpassm(H(1,:)
2*M);
[n2,d2]=allpassm(H(2,:)
2*M);
n2=[zeros([1 M]) n2];
F1=[freqz(n1,d1,K)];
[F12,w]=freqz(n2,d2,K);
H1=(F1+F12)/2;
H2=(F1-F12)/2;
clear F1 F12

da=fliplr(eye([1 M]));
D=freqz(d,1,K);

A3=zeros([K 1]); A4=zeros([K 1]);
for l=2:M,
A1=zeros([K 1]); A2=zeros([K 1]);
for k=1:N,
[n1,d1]=allpassm(A(k,:),M);
[n2,d2]=allpassm(C(k,:),M);
Hn1=freqz(n1,d1,K);
Hn2=freqz(n2,d2,K);
A1=W^(-(k-1)*(1-l)))+Hn1.*Hn2+A1;
end;
A1=H1.*A1; A2=H2.*A2;
A3=A1+A2;
A4=A4+abs(D.*A3/K).^-2;
end;
```

%noemer=1 met N nullen
end;

Ea=(sqrt(A4));
mag=20*log10(Ea);
plot(w/pi,mag);
end;