All-Digital Aggregator for Multi-Standard Video Distribution

Andreas Norén
Master of Science Thesis in Electrical Engineering

All-Digital Aggregator for Multi-Standard Video Distribution

Andreas Norén

LiTH-ISY-EX--18/5129--SE

Supervisor: Oscar Gustafsson
ISY, Linköping University

Examiner: Håkan Johansson
ISY, Linköping University

Department of Electrical Engineering
Linköping University
SE-581 83 Linköping, Sweden

Copyright © 2018 Andreas Norén
Abstract

In video transmission there is a need to compose a wide-band signal from a number of narrow-band sub-signals. A flexible solution offers the possibility to place any narrow-band sub-signal anywhere in the wide-band signal, making better use of the frequency space of the wide-band signal. A multi-standard supportive solution will also consider the three standard bandwidths of digital and analog video transmissions, both terrestrial and cable (6, 7 and 8 MHz), in use today.

This thesis work will study the efficiency of a flexible aggregation solution, in terms of computational complexity and error vector magnitude (EVM). The solution uses oversampled complex modulated filter banks and inner channelizers, to reduce the total workload on the system.

Each sub-signal is channelized through an analysis filter bank and together all channelized sub-signals are aggregated through one synthesis filter bank to form the wide-band composite signal. The EVM between transmitted and received sub-signals are investigated for an increasing number of sub-signals.

The solution in this thesis work is performing good for the tested number of up to 100 narrow-band sub-signals. The result indicates that the multi-standard flexible aggregation solution is efficient for an increasing number of transmitted sub-signals.
Acknowledgments

I would like to thank my examiner Håkan Johansson and my supervisor Oscar Gustafsson for their help and guiding through my thesis work, as well as the ability to discuss the problems which I encountered. I also would like to thank my parents Inger and Håkan, as well as all my friends, for their support and company during the years at the university. Lastly, but not least, I would like to thank my lovely and wonderful girlfriend Sofia for her support. I love you, always and forever.

Linköping, April 2018
Andreas Norén
# Contents

Notation ix

1 Introduction 1
   1.1 Motivation ........................................... 2
       1.1.1 Ethical Aspects .................................. 2
   1.2 Problem Formulation .................................. 2
       1.2.1 Flexibility ...................................... 3
       1.2.2 Computational Complexity ......................... 3
   1.3 Method .............................................. 3
       1.3.1 Test System Description ......................... 4
       1.3.2 Software ....................................... 4
   1.4 Related Work ....................................... 5
   1.5 Thesis Outline ..................................... 5

2 Theory 7
   2.1 Discrete-Time Filters – Digital Filters .............. 7
       2.1.1 Symmetric Linear-Phase FIR Filters ............... 7
       2.1.2 Synthesis of Linear-Phase FIR Filters ............ 9
   2.2 Multirate Systems .................................... 10
       2.2.1 Interpolation .................................... 10
       2.2.2 Decimation ...................................... 12
       2.2.3 Noble Identities ................................ 13
       2.2.4 Polyphase Representation ......................... 14
       2.2.5 Sample Rate Conversion by a Rational Number .... 16
   2.3 Filter Banks ....................................... 17
       2.3.1 Complex Modulated Filter Banks ................. 17
   2.4 Analysis Filter Banks ................................ 18
       2.4.1 Synthesis Filter Banks ......................... 19
   2.5 Flexible Aggregator .................................. 19
   2.6 Raised Cosine FIR Filters ............................ 20
   2.7 Pulse Shaping ....................................... 21
   2.8 Discrete Fourier Transforms .......................... 21
   2.9 Quadrature Amplitude Modulation ...................... 22
3 Flexible Aggregator Scheme

3.1 Signal Generator .................................................. 23
   3.1.1 Pulse Shaping ........................................... 24

3.2 Aggregator ......................................................... 26
   3.2.1 Prototype Filters ......................................... 26
   3.2.2 Efficient IDFT and DFT Based Implementation of the FBs 28
   3.2.3 Resulting Filter Banks ................................... 29

3.3 Frequency-Band Allocation Schemes .................................. 30
   3.3.1 User Signal Distribution .................................. 30
   3.3.2 Right & Left .............................................. 31
   3.3.3 Center .................................................... 32
   3.3.4 Observations .............................................. 33

3.4 Receiver ........................................................ 33
   3.4.1 Error Vector Magnitude .................................. 33
   3.4.2 Signal Extractor .......................................... 34
   3.4.3 Matched Receiver Filter ................................ 35
   3.4.4 Constellation Scatter Plot ............................... 36
   3.4.5 Error Calculations ...................................... 37

3.5 Straightforward Solution ........................................ 39

3.6 Computational Complexity ......................................... 40
   3.6.1 Flexible Aggregator Scheme ............................. 40
   3.6.2 Straightforward Solution ................................ 41
   3.6.3 Comparison ............................................... 42

4 User Interface ......................................................... 45
   4.1 Main Interface ............................................... 45
   4.2 User Filter Designs ......................................... 46

5 Conclusions .................................................................. 49
   5.1 Overview ....................................................... 49
   5.2 Results ........................................................ 50
   5.3 Future Work .................................................. 50
      5.3.1 Alternative Extractor .................................. 50
      5.3.2 Higher Prototype Filter Orders ...................... 50
      5.3.3 Frequency-Band Allocation Scheme ................ 50

Bibliography .................................................................... 53

Index ............................................................................. 55
# Notation

## Key Signals

<table>
<thead>
<tr>
<th>Signal</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_k(n)$</td>
<td>64-QAM symbol streams</td>
</tr>
<tr>
<td>$z_k(n)$</td>
<td>Pulse shaped $x_k(n)$ – narrow-band sub-signals</td>
</tr>
<tr>
<td>$y(n)$</td>
<td>Composite wide-band signal</td>
</tr>
<tr>
<td>$\hat{z}_k(n)$</td>
<td>Extracted sub-band signals from $y(n)$</td>
</tr>
<tr>
<td>$\hat{x}_k(n)$</td>
<td>Received 64-QAM symbol streams</td>
</tr>
</tbody>
</table>

## Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFB</td>
<td>Analysis Filter Bank</td>
</tr>
<tr>
<td>APF</td>
<td>Analysis Prototype Filter</td>
</tr>
<tr>
<td>DFT</td>
<td>Discrete Fourier Transform</td>
</tr>
<tr>
<td>EVM</td>
<td>Error Vector Magnitude</td>
</tr>
<tr>
<td>FA</td>
<td>Flexible Aggregator</td>
</tr>
<tr>
<td>FB</td>
<td>Filter Bank</td>
</tr>
<tr>
<td>FBAS</td>
<td>Frequency-band Allocation Scheme</td>
</tr>
<tr>
<td>FIR</td>
<td>Finite-length Impulse Response</td>
</tr>
<tr>
<td>GB</td>
<td>Granularity Band</td>
</tr>
<tr>
<td>IDFT</td>
<td>Inverse Discrete Fourier Transform</td>
</tr>
<tr>
<td>ISI</td>
<td>Intersymbol Interference</td>
</tr>
<tr>
<td>MPR</td>
<td>McClellan-Parks-Rabiner (algorithm)</td>
</tr>
<tr>
<td>QAM</td>
<td>Quadrature Amplitude Modulation</td>
</tr>
<tr>
<td>RCFIR</td>
<td>Raised Cosine Finite-length Impulse Response</td>
</tr>
<tr>
<td>SFB</td>
<td>Synthesis Filter Bank</td>
</tr>
<tr>
<td>SPF</td>
<td>Synthesis Prototype Filter</td>
</tr>
<tr>
<td>SRC</td>
<td>Sample Rate Converter</td>
</tr>
<tr>
<td>SRRCFIR</td>
<td>Square-root-raised Cosine Finite-length Impulse Response</td>
</tr>
</tbody>
</table>
This technical report will present the master thesis on a flexible aggregator (FA) for multi-standard video distributions, which utilize inner channelizers to achieve a more efficient solution. In broadband cable networks the problem is to aggregate the different bandwidths narrow-band sub-signals (6, 7 or 8 MHz) into one composite wide-band signal, with an example in Fig. 1.1. These bandwidths cover all standards in use today, both digital and analog, terrestrial and cable [4]. Some examples of digital standards are Advanced Television Systems Committee (ATSC) standards, Digital Terrestrial Multimedia Broadcast (DTMB), Digital Video Broadcasting - Cable / Terrestrial (DVB-C/T) and the Data Over Cable Service Interface Specification (DOCSIS). Examples of analog standards are Phase Alternating Line (PAL) and National Television Systems Committee (NTSC).

While maintaining a low computational complexity, the error vector magnitude (EVM) of the FA is investigated when the number of sub-signals is increased. That is, the received sub-signals should approximate their transmitted counterparts.

This chapter will include a motivation why the multi-standard FA is used, a
formulation of the problem at hand and a look at the method as a whole. The chapter is concluded with an outline of this technical report.

1.1 Motivation

Multi-standard aggregators are desired because they provide a method to handle a mixed set of narrow-band sub-signals with different bandwidths. They also yield the possibility of application in all parts of the world due to its multi-standard bandwidth support. The most frequently used standards in Europe are 7 and 8 MHz, while the most frequently used standard in the United States is 6 MHz [4].

FAs are desired due to the flexibility to place any user anywhere in the wideband composite signal. Evident by Fig. 1.1 it would be possible to place one additional narrow-band sub-signals, of bandwidth 7 MHz, in the wide-band signal. This would maximize the total amount of useful data sent, i.e. minimizing the empty frequency space of the wide-band signal.

There is a simple straightforward solution consisting of a set of digital band-pass filters and sample rate converters (SRCs), but it has a high computational complexity. The scheme in this thesis work offers a computational complexity which is reduced by orders-of-magnitude compared to the complexity of the straightforward solution, when the number of narrow-band sub-signals is increased.

Interested parties presently use a solution which is both analog and digital [6]. This solution is restricted to only four channels, which makes it harder to further expand, and improve, the solution. An all-digital solution has several advantages such as increased control and adaptability. There are greater possibilities to further improve an all-digital solution than to improve an analog/digital solution.

1.1.1 Ethical Aspects

By using oversampled filter banks (FBs) the number of operations per sample is reduced. This results in less energy required by the transmitter, when composing the wide-band signal. The scheme is henceforth considered free from ethical aspects, if not used for any violent purposes.

1.2 Problem Formulation

The problem addressed is to aggregate a number of narrow-band sub-signals, here called users, with different bandwidths (6, 7, and 8 MHz) into one composite wide-band signal. It is assumed that there are K digital signals \( z_k(n) \), \( k = 1, \ldots, K \), with the same sample rate. The solution is to be all-digital and the aggregation of the K users is to be carried out in a flexible fashion, i.e. the possibility to place each user deemed where best fitted should be available.

The purpose is to extend the solution concept of [10], for an increasing amount of user signals, and to generate code of the solution.
1.2.1 Flexibility

The flexibility will be limited to placing user signals close together, i.e. without any guard band between each user pair. Guard band is in this context a part of the frequency space between each user which is not utilized. An example of a wide-band signal utilizing guard bands between each user pair is shown in Fig. 1.2.

![Spectrum Diagram](image.png)

**Figure 1.2:** A wide-band signal, here composed of 6 user signals, including guard bands between each user.

Guard bands are useful in the sense that they help to easier extract each user from the wide-band signal, at the receiver. However, to identify when extra users can be added to the wide-band signal, as briefly discussed in Section 1.1, will not be considered in this thesis work.

1.2.2 Computational Complexity

The conditions in terms of complexity are in this thesis work restricted to operations (multiplications and additions) per sample, performed in the FA. Note that the number of additions scales with the number of multiplications. The complexity of additional sub-systems used to test the FA are not in focus, in this thesis work. Therefore are the costs and complexities of signal generation and filter implementation, in these parts, of less importance. The computational complexity hence is not the time which the operations take but rather how many operations there are, per sample. The straightforward solution will be compared to the FA given in this thesis work.

1.3 Method

The scheme in this thesis work utilizes complex modulated oversampled FBs, based on [10], which reduce the workload of the system. The work load is reduced further with the aid of inner channelizers. The FBs are here based on symmetric linear-phase finite-length impulse response (FIR) filters of Type I. These types of filters preserve the signal shape and they also have integer valued delays. The design of these filters will be optimized in the minimax sense, further discussed in Section 2.1.2.
A schematic of the multi-standard FA is shown in Fig. 1.3. The scheme carries each user through an analysis filter bank (AFB), utilizing an efficient inverse discrete Fourier transform (IDFT) based implementation. Later all channelized users are aggregated through one synthesis filter bank (SFB), utilizing an efficient discrete Fourier transform (DFT) based implementation, composing the wide-band signal. With the help of a channel select block full flexibility is enabled. Low computational complexity is made possible by this scheme, compared to the straightforward solution, and quality is maintained.

**Figure 1.3:** The multi-standard FA with $K$ inputs and one output.

### 1.3.1 Test System Description

The test system consists of three parts, where a schematic is given in Fig. 1.4. Here $z(n)$ contains all signals $z_k(n), k = 1, \ldots, K$ and $\hat{x}(n)$ contains all signals $\hat{x}_k(n), k = 1, \ldots, K$.

**Figure 1.4:** A schematic of the test system.

It is important to point out that the FA block is the part most relevant to the thesis work. The two other blocks are however necessary to test the implementation of the FA block.

### 1.3.2 Software

The software used in this thesis work is MatLab R2015b - academic use, by MathWorks, Inc. Therefore all functions mentioned, in *italic* script, are MatLab functions.
1.4 Related Work

The solution scheme in this thesis work is inspired by recent results in the area [3, 11]. Unlike [7, 8], which use similar techniques, the aggregation flexibility of the scheme in this thesis work is much more prominent.

Another modulation methods of FBs include fast convolution based modulation [2, 19, 20], which also offers high flexibility and efficiency regarding aggregation. This kind of modulation does however not address the same kind of problem as in this thesis work.

1.5 Thesis Outline

Following this introduction, Chapter 2 gives useful theory for the user to easier understand implementations. The system parts in Fig. 1.4 as well as the straightforward solution of the multi-standard FA is presented in Chapter 3. Chapter 4 describes the user interface and how the user can design new filters or re-design existing filters. Chapter 5 states the conclusions and potential future work, concluding this technical report.
This chapter starts with theory regarding digital filters and moves on to multirate systems and polyphase representation. Complex modulated filter banks are explained and theory concerning FAs is presented. Concluding the chapter are short theory regarding raised cosine FIR filters, pulse shaping, DFT and quadrature amplitude modulation (QAM).

2.1 Discrete-Time Filters – Digital Filters

The discrete-time filters called FIR filters will be considered in this section. The impulse response $h(n)$ of a causal filter of order $N$ is only non-zero for $0 \leq n \leq N$. Its transfer function and frequency response can be written

$$H(z) = \sum_{n=0}^{N} h(n)z^{-n}$$

and

$$H(e^{j\omega T}) = \sum_{n=0}^{N} h(n)e^{-j\omega Tn},$$

respectively.

2.1.1 Symmetric Linear-Phase FIR Filters

FIR filters can be designed to have a linear-phase response, i.e., the phase delay and group delay are constant – all frequency components are delayed equally. The signal shape is preserved at the price of a longer delay in the filter.
**Zero-Phase Frequency Response**

The frequency response is expressed using the real function $H_R(\omega T)$ as

$$H(e^{j\omega T}) = e^{-j\omega TN/2} H_R(\omega T).$$

(2.3)

Here, $H_R(\omega T)$ is called the zero-phase frequency response of $H(e^{j\omega T})$. In Figs. 2.1(a), 2.1(b), 2.1(c) and 2.1(d) are characteristic impulse responses of the four types of symmetric and anti-symmetric linear-phase FIR filters.

![Impulse responses](image)

**Figure 2.1:** Characteristic impulse responses of linear-phase FIR-filters.

The real zero-phase frequency responses of Type I, II, III and IV linear-phase FIR filters are expressed as

$$H_R(\omega T) = h\left(\frac{N}{2}\right) + 2 \sum_{n=1}^{N/2} h\left(\frac{N}{2} - n\right) \cos(\omega T n), \quad \text{Type I},$$

(2.4)

$$H_R(\omega T) = 2 \sum_{n=1}^{(N+1)/2} h\left(\frac{N + 1}{2} - n\right) \cos\left(\omega T\left(n - \frac{1}{2}\right)\right), \quad \text{Type II},$$

(2.5)

$$H_R(\omega T) = 2 \sum_{n=1}^{N/2} h\left(\frac{N}{2} - n\right) \sin(\omega T n), \quad \text{Type III},$$

(2.6)
and
\[
H_R(\omega T) = 2 \sum_{n=1}^{(N+1)/2} h\left(\frac{N+1}{2} - n\right) \sin\left(\omega T \left(n - \frac{1}{2}\right)\right), \quad \text{Type IV},
\]
respectively. Consider \(H_R(\omega T)\) instead of \(H(e^{j\omega T})\). The general specifications of a lowpass filter is then given as
\[
D_1(\omega T) - \delta_1(\omega T) \leq H_R(\omega T) \leq D_1(\omega T) + \delta_1(\omega T), \quad \omega T \in [0, \omega_c T]
\]
\[
-\delta_2(\omega T) \leq H_R(\omega T) \leq \delta_2(\omega T), \quad \omega T \in [\omega_s T, \pi]
\]
where \(\omega_s T\) and \(\omega_c T\) are the stopband edge and passband edge, respectively. Here \(D_1(\omega T)\) is the desired magnitude response in the passband. \(\delta_1(\omega T)\) and \(\delta_2(\omega T)\) denote allowed deviation from \(D_1(\omega T)\) in the passband and zero in the stopband, respectively. This is illustrated in Fig. 2.2.

\[\text{Figure 2.2: Typical specifications for a linear-phase FIR-filter, in analogy with (2.8) and (2.9). Here, } \delta_1(\omega T) = \delta_c, \delta_2(\omega T) = \delta_s \text{ and } D_1(\omega T) = 1.\]

### 2.1.2 Synthesis of Linear-Phase FIR Filters

A frequently used algorithm to design linear-phase FIR filters is the McClellan-Parks-Rabiner (MPR) algorithm [9], which use a weighted error function as
\[
E(\omega T) = W(\omega T) [H_R(\omega T) - D(\omega T)], \quad \omega T \in \Omega.
\]
Here, \(\Omega = [0, \omega_c T] \cup [\omega_s T, \pi]\), \(D(\omega T)\) is a desired function to approximate, and \(W(\omega T)\) is a weighting function, specifying costs of deviation from \(D(\omega T)\).

Minimal maximal ripples in both passband and stopband are obtained through minimax approximation, i.e. the maximum value of \(|E(\omega T)|\) is minimized, formulating the problem as
\[
\text{minimize } E_\infty = \min_{\omega T} \max |E(\omega T)|,
\]
The filter specification in (2.8) and (2.9) is satisfied if $E_\infty \leq \delta_c$. Filters which have minimal maximal ripples are said to be optimized in the minimax sense. This means they are unique in the sense that no other filter, of equal or lower order, has smaller ripples in both passband and stopband.

Another method to synthesize linear-phase FIR-filters, not used in this thesis work, is the frequency response masking (FRM) approach [12, 15, 16]. FRM is a method to achieve steep transition bands at a low implementation complexity. This is however not needed when constructing the FBs, in this thesis work application.

2.2 Multirate Systems

Multirate systems utilize interpolation and decimation [9], which increase and decrease the sampling frequency, respectively. The corresponding subsystems, simply called interpolator and decimator, make use of frequency selective filters. These SRCs are used to obtain an as low computational workload as possible in the system’s digital parts. Multirate techniques can help reduce the complexity in a wide variety of applications, such as filter banks.

2.2.1 Interpolation

The interpolator, seen in Fig. 2.3, performs the interpolation and consists of an upsampler and a digital filter, called interpolation filter or anti-imaging filter.

\[ \hat{x}(n) \leftarrow \frac{1}{f_s} \rightarrow L \rightarrow \hat{x}_I(m) \rightarrow H(z) \rightarrow \frac{1}{Lf_s} y(m) \]

*Figure 2.3: Interpolator consisting of an upsampler and a digital filter.*

Consider the discrete-time representation $\hat{x}(n)$ of the continuous-time signal $x(t)$. By upsampling $\hat{x}(n)$, $L - 1$ zero-valued samples are inserted between each sample pair. The signal $\hat{x}_I(m)$ is created where correct sample values are only located at samples $m = kL$. After lowpass filtering (filter $H(z)$ with cut-off frequency $\omega_c = \pi/L$) all correct sample values are achieved. An example is illustrated in Fig. 2.4 with $L = 3$.

Figure 2.5 shows a frequency domain visualization of the above example.
2.2 Multirate Systems

\[ \hat{x}(n) = x(nT) \]

(a) The input signal \( \hat{x}(n) \).

\[ x(t) \]

(b) The intermediate signal \( \hat{x}_1(m) \).

\[ y(m) = x(nT/3) \]

(c) The interpolated signal \( y(m) \).

**Figure 2.4:** Time domain visualization of interpolation.

(a) The spectrum \( |\hat{X}(e^{j\omega T})| \) of the input signal \( \hat{x}(n) \).

(b) The spectrum \( |\hat{X}_1(e^{j\omega T_1})| \) of the intermediate signal \( \hat{x}_1(m) \).

(c) Ideal lowpass filter magnitude response \( |H(e^{j\omega T_1})| \).

(d) The spectrum \( |Y(e^{j\omega T_1})| \) of the interpolated signal \( \hat{y}(m) \).

**Figure 2.5:** Frequency domain visualization of interpolation.
As is clear from Figs. 2.4 and 2.5, $\hat{x}_I(m)$ contains the baseband of $\hat{x}(n)$ and $L - 1$ so called images of the baseband. By eliminating these images with an ideal lowpass filter the interpolated sequence $y(m)$ is obtained.

### 2.2.2 Decimation

The decimator in Fig. 2.6 performs the decimation and consists of a digital filter, called decimation filter or anti-aliasing filter, and a downsampler.

![Decimator consisting of a digital filter and a downsampler.](image)

Consider the discrete representation $\hat{x}(m)$ of the continuous signal $x(t)$. By filtering $\hat{x}(m)$ with an ideal lowpass filter, with cut-off frequency $\omega_c T_1 = \pi/M$, a signal $\hat{x}_D(m)$ is created. The decimated signal $y(n)$ is achieved by downsampling $\hat{x}_D(m)$ by a factor $M$. An example is illustrated in Fig. 2.7 with $M = 3$.

![Time domain visualization of decimation.](image)

Figure 2.7: Time domain visualization of decimation.

Figure 2.8 shows a frequency domain visualization of the above example.
2.2 Multirate Systems

2.2.3 Noble Identities

The so called noble identities describe how interchanging upsamplers and downsamplers and filters affect the structure of the filter. They also describe how upsamplers and downsamplers can be moved in branching flow-graphs which include additions and multiplications. The noble identities below are very useful when the polyphase interpolator and decimator are derived, considered in Section 2.2.4.

The first two identities are based upon relations between delays and sample rates. The expansion identity in Fig. 2.9 describes interchange between filters and upsamplers.

\[ \hat{x}(n) \begin{array}{c} \downarrow L \end{array} \hat{x}_I(m) \begin{array}{c} H(z^L) \end{array} \begin{array}{c} y(m) \end{array} \begin{array}{c} Lf_s \end{array} \]

\[ \hat{x}(n) \begin{array}{c} f_s \end{array} \begin{array}{c} \downarrow L \end{array} \hat{x}_I(m) \begin{array}{c} H(z^L) \end{array} \begin{array}{c} y(m) \end{array} \begin{array}{c} Lf_s \end{array} \]

Figure 2.9: The expansion identity.

The decimation identity in Fig. 2.10 describes interchange between filters and downsamplers.

\[ \hat{x}(n) \begin{array}{c} f_s \end{array} \begin{array}{c} H(z) \end{array} \begin{array}{c} \uparrow L \end{array} \hat{x}_I(n) \begin{array}{c} y(m) \end{array} \begin{array}{c} Lf_s \end{array} \]

\[ \hat{x}(n) \begin{array}{c} \downarrow L \end{array} \hat{x}_I(n) \begin{array}{c} H(z) \end{array} \begin{array}{c} \downarrow L \end{array} \hat{x}(n) \begin{array}{c} f_s \end{array} \begin{array}{c} \downarrow L \end{array} \hat{x}_I(m) \begin{array}{c} y(m) \end{array} \begin{array}{c} Lf_s \end{array} \]

Figure 2.10: The decimation identity.
Figure 2.10: The decimation identity.

Additions and multiplications, as well as branching flow-graphs, are independent of the sampling frequency. This means that upsamplers and downsamplers can be moved according to the multiplication identity and the addition identity, respectively. The multiplication identity in Fig. 2.11 describes how upsamplers can be moved in a branched flow-graph, which includes $N$ multipliers $c_n$, $n = 1, \ldots, N$.

Figure 2.11: The multiplication identity.

The addition identity in Fig. 2.12 describes how downsamplers can be moved in a branched flow-graph, which includes a number of adders.

Figure 2.12: The addition identity.

2.2.4 Polyphase Representation

To decrease the computational requirements, and to utilize an efficient implementation, polyphase represented filters are commonly used [17]. The original filter is described as

$$H(z) = \sum_{i=0}^{L-1} z^{-i} H_i(z^L),$$  \hspace{1cm} (2.12)
where $H_i(z)$ are the polyphase component filters, according to

$$H_i(z) = \sum_{n=0}^{\infty} h(Ln + i), \quad i = 0, \ldots, L - 1,$$

(2.13)

where $h(n)$ is the filter impulse response.

**Interpolator and Decimator Structures**

The polyphase represented filter is utilized when deriving the polyphase interpolator and the polyphase decimator. By using the identities described in Section 2.2.3, the polyphase interpolator and decimator can be derived from $H(z)$ described as in (2.12). In Figs. 2.13 and 2.14, two different schematics are given for the polyphase decimator and the polyphase interpolator, respectively.

The use of the expansion identity, in Fig. 2.9, and the multiplication identity, in Fig. 2.11, yields the resulting polyphase interpolator seen in Fig. 2.13(a). The interpolator is however realized as the schematic in Fig. 2.13(b) in practice.

![Polyphase component filters](image)

(a) The structure of a polyphase interpolator.

![Polyphase component filters](image)

(b) Schematic with a commutator.

**Figure 2.13:** A polyphase interpolator, based on the filter $H(z)$.

The upsamplers and adders have been replaced by a commutator in Fig. 2.13(b), which will rotate counter clockwise. Note that for every input value, there are $L$ output values – the output sample rate is $L$ times higher than the input rate.

The use of the decimation identity, in Fig. 2.10, and the addition identity, in Fig. 2.12, yields the resulting polyphase decimator seen in Fig. 2.14(a). The decimator is however realized as the schematic in Fig. 2.14(b) in practice.
2.2.5 Sample Rate Conversion by a Rational Number

It is often needed to convert the sample rate by a rational number. The method is to use one interpolator and one decimator, as the chain in Fig. 2.15

\[
\begin{align*}
\hat{x}(n) & \xrightarrow{f_s} \uparrow L \hat{x}_I(n) \xrightarrow{H_1(z)} z(n) \xrightarrow{L_f_s} \hat{z}_D(n) \xrightarrow{H_D(z)} \downarrow M y(n) \xrightarrow{L f_s} M y(n)
\end{align*}
\]

\[
\omega_c T = \pi/L \quad \omega_c T = \pi/M
\]

**Figure 2.15:** A sample rate conversion by a rational number $L/M$, consisting of one interpolator and one decimator.

It is from here possible to only use one filter by combining the two filters. Let us call this new combined filter $H_{comb}(z)$, with cut-off frequency $\omega_c T = \pi/C$, where $C = \max\{L, M\}$. The simplified chain becomes as in Fig. 2.16.

\[
\begin{align*}
\hat{x}(n) & \xrightarrow{f_s} \uparrow L \hat{x}_I(n) \xrightarrow{H_{comb}(z)} \hat{z}_D(n) \xrightarrow{H_D(z)} \downarrow M y(n) \xrightarrow{L f_s} M y(n)
\end{align*}
\]

\[
\omega_c T = \pi/C
\]

**Figure 2.16:** A sample rate conversion by a rational number $L/M$. 
2.3 Filter Banks

An FB can be described as an array of bandpass filters. There are naturally two roles of an FB; decomposition or reconstruction. The decomposition is called analysis, referring to the analysis of the sub-band signals. The reconstruction is called synthesis, referring to the synthesizing of a new signal based on the input sub-band signals.

2.3.1 Complex Modulated Filter Banks

In an \( N \)-channel complex modulated FB all filters are modulated versions of the same prototype filter \( H(z) \) as

\[
H_k(z) = \beta_k H(z W_N^{\alpha+k}), \quad k = 0, 1, \ldots, N - 1,
\]

(2.14)

where

\[
\beta_k = W_N^{(\alpha+k)D/2}, \quad W_N = e^{-j2\pi/N},
\]

(2.15)

where \( D \) is the order of the prototype filter. The constant \( \alpha \in \mathbb{R} \) is used to place the filters at desired center frequencies. In this thesis work \( \alpha = 1/2 \) to obtain eight equidistant granularity bands (GBs) in \([0, 2\pi]\) and also to achieve a simple implementation, where the modulators become simple. The \( \beta_k \) are constants which compensate for the introduced phase shift, when \( H(z) \) is replaced with \( H(z W_N^{\alpha+k}) \), rendering all FB filters linear-phase FIR filters with the same delay \( D/2 \) as the prototype filter. All \( \beta_k \) can however be made equal to unity \([10]\), by selecting a proper prototype filter order \( D \). The factor \( W_N \) is a primitive root of unity, referred to as twiddle factor in fast Fourier transform (FFT) algorithms.

Example

An \( M \)-channel FB is based on a prototype filter \( H(z) \) with cut-off frequency \( \omega_c T = \pi/M \), with magnitude response in Fig. 2.17(a). The resulting FB will here consist of a total of \( M \) filters \( H_k(z) \), \( k = 0, \ldots, M - 1 \), to cover \([0, 2\pi]\), all complex modulated versions of \( H(z) \). By selecting the filter order as \( D = 16 \) (or multiples of 16), all \( \beta_k \) can be made unity. The FB spectra is given in Fig. 2.17(b).
2.4 Analysis Filter Banks

The AFB separates the input signal $z(n)$ into $C$ sub-band signals $z_c(n)$, $c = 0, \ldots, C−1$, by filtering out different spectral components with the bandpass filters $G_c(z)$. An AFB with one input and $M$ outputs is shown in Fig. 2.18.

Figure 2.18: An AFB with one input and $C$ outputs.
2.4.1 Synthesis Filter Banks

The SFB creates a signal $y(n)$ using the provided $P$ sub-band input signals $z_p(n)$, $p = 0, \ldots, P-1$, by placing the input signals on the corresponding filter $F_p(z)$ spectral space. An SFB with $P$ inputs and one output is shown in Fig. 2.19.

![Figure 2.19: An SFB with $P$ inputs and one output.](image)

2.5 Flexible Aggregator

The function of an FA is to compose a wide-band signal from narrow-band sub-signals. The flexibility is achieved with a channel select block to direct each narrow-band sub-signal to specific parts of the wide-band signal. This can be used to utilize as much space of the available frequency space as possible, which increase the flexibility and functionality of the system since any user can be placed where it is seemed best fit. Figure 2.20 shows an FA with $K$ AFBs with $C$ channels and one $P$-channel SFB.

![Figure 2.20: An FA with $K$ inputs and one output.](image)

In Fig. 2.21, the AFB block and the SFB block used in Fig. 2.20 are shown.
2.6 Raised Cosine FIR Filters

The raised cosine finite-length impulse response (RCFIR) filter is one of the most frequently used filters when it comes to pulse shaping [1, 5]. The beginning and the end of each symbol period are typically the most susceptible to multi-path disturbances. The RCFIR filters attenuate these portions to help minimize the intersymbol interference (ISI) [18]. The frequency response of such a filter is described by [21],

\[
H_{rc}(f) = \begin{cases} 
\frac{T_s}{2}, & 0 \leq |f| \leq \frac{1-\beta}{2T_s} \\
\frac{T_s}{2} \left( 1 + \cos \left( \frac{\pi T_s}{\beta} \left( |f| - \frac{1-\beta}{2T_s} \right) \right) \right), & \frac{1-\beta}{2T_s} \leq |f| \leq \frac{1+\beta}{2T_s} \\
0, & |f| > \frac{1+\beta}{2T_s}
\end{cases}, \tag{2.16}
\]

where the roll-off parameter, \( 0 \leq \beta \leq 1 \), is a measure of the excess bandwidth of the filter, i.e. bandwidth beyond the Nyquist bandwidth \( 1/(2T_s) \), see Fig. 2.22.

![Figure 2.21: The AFB block and the SFB block used in Fig. 2.20.](image)

![Figure 2.22: Magnitude responses with different roll-off factors \( \beta \).](image)
Therefore, to ensure near, or equal to, zero ISI the total channel frequency response must be raised cosine. This can be achieved by using matched square-root-raised cosine finite-length impulse response (SRRCFIR) filters at the receiver and transmitter. The frequency response of the SRRCFIR filter equals the square root RCFIR filter frequency response as

\[ H_{rc}(f) = H_{srrc}(f)H_{srrc}(f), \text{ and } |H_{srrc}(f)| = \sqrt{|H_{rc}(f)|}. \] (2.17)

### 2.7 Pulse Shaping

Pulse shaping is the process of making a transmitted signal better suited for the communication channel. By filtering the signal this way it changes the effective bandwidth of the transmission and the ISI can be better controlled. In this thesis work all signals have a bandwidth of 6, 7 or 8 MHz and SRRCFIR filters are used to minimize the ISI, see Fig. 2.23, which can be compared to the SRC, with a factor of a rational number, in Fig. 2.16.

![Figure 2.23: A schematic of the pulse shaper.](image)

Here is \( C = \max\{L, M\} \), which matches \( H(z) \) to the largest factor, i.e. \( H(z) \) has a cut-off frequency of \( \omega_c T = \pi/C \).

### 2.8 Discrete Fourier Transforms

The DFT is the tool used when calculating the transform of finite-length sequences, since we must consider a finite amount of frequencies. There exist efficient algorithms to compute the DFT, such as the FFT.

The DFT of a finite-length sequence \( x(n), n = 0, 1, \ldots, N - 1 \), is defined as

\[
X(k) = \sum_{n=0}^{N-1} x(n)W_N^{nk}, \quad k = 0, 1, \ldots, N - 1
\] (2.18)

where \( W_N \) is the twiddle factor, defined in Section 2.3.1 in (2.15). \( X(k) \) is a complex sequence of length \( N \), giving it the commonly used name \( N \)-point DFT. \( x(n) \) can be obtained from \( X(k) \) by using the IDFT according to

\[
x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)W_N^{-nk}, \quad n = 0, 1, \ldots, N - 1.
\] (2.19)
2.9 Quadrature Amplitude Modulation

QAM is a standard format by which digital television cable channels are encoded. In the digital sense, it conveys two bit streams by amplitude modulating two carrier waves, using amplitude-shift modulation (ASM). The two carrier waves (usually sinusoids of the same frequency) are out of phase by $\pi/2$ with each other. Therefore they are called quadrature carriers or components. The final waveform is the sum of the two waves, which is a combination of the phase-shift modulation and the ASM. Note that this is one way of implementing a QAM signal.

The most commonly used types of QAM encoding regarding digital television signals are 16-QAM and 64-QAM. This means that the QAM either uses a 16 symbol or a 64 symbol library when encoding the signals.

In this thesis work 64-QAM signals are used, which are the result of four steps [13]. First a stream of random binary bits are generated, using the function `randi`. This bit stream is reshaped into binary $m$-tuples, $m = \log_2(64)$, using `reshape`, which in turn are converted into integers, using `bi2de`. Lastly is 64-QAM applied on the integers using binary coding with a 64 symbol alphabet, using `qammod`. The final signal is a complex column vector whose values are elements of the 64-QAM signal constellation. A scatter plot of a 64-QAM signal constellation is illustrated in Fig. 2.24.

![Scatter plot](image)

**Figure 2.24:** Scatter plot of the 64-QAM signal constellation.
Flexible Aggregator Scheme

To be able to test the FA scheme of this thesis work two additional sub-systems are implemented. These two sub-systems are the signal generator and the receiver, whose computational complexities are not taken into account in this thesis work. This chapter will therefore also consider the signal generator and the receiver. The components of the signal generator is described in the first section, as the second section moves on to explain the aggregation scheme. The different flexibility schemes in this thesis work are then described followed by a section about the receiver. Concluding the chapter is the straightforward solution and calculations of the computational complexities, as well as a comparison between the two solutions.

3.1 Signal Generator

In this thesis work the used signals are random pulse shaped 64-QAM symbol streams. These signals are to model the standards of digital television signals in use today, i.e. signals of bandwidths 6, 7 and 8 MHz [4]. The signal generator, seen in Fig. 1.4, consists of two parts. The first part is referred to as a QAM generator, which generate random QAM symbol streams $x_k(n)$, and the second part is referred to as a pulse shaper, generating signals $z_k(n)$. A schematic of the signal generator is shown in Fig. 3.1.

Figure 3.1: A schematic of the signal generator.

Here, $x(n)$ contains all generated 64-QAM symbol streams $x_k(n)$, $k = 1, \ldots, K$,
and \(z(n)\) contains all signals \(z_k(n), \ k = 1, \ldots, K\). Note that this sub-system is not in focus, therefore the filters are over designed to reduce interference with the multi-standard FA.

### 3.1.1 Pulse Shaping

The different bandwidths (6, 7 and 8 MHz) are achieved by following the schematic in Fig. 3.2. The bandwidths of all users are stored in a vector \(f_{bw}\), in MHz, i.e. \(f_{bw}(k) \in \{6, 7, 8\}\). All user signals \(z_k(n)\) are here desired to have the same length, therefore it is of interest to modify the sample rate by a rational number. All user signals are to be used to compose the wide-band signal, therefore to be able to do this at all, the user signals must be of equal lengths.

\[
\begin{align*}
\frac{x_k(n)}{f_s} & \quad \uparrow L \quad H_{ps}(z) \quad \downarrow M \quad \frac{z_k(n)}{\frac{L}{M} f_s} \\
\omega_c T &= \pi/L
\end{align*}
\]

**Figure 3.2:** A schematic of the pulse shape chain.

The upsampler has a factor \(L = 10\) and the filter \(H_{ps}(z)\) is an SRRCFIR filter matched to the upsampler, i.e., its cut-off frequency is \(\omega_c T = \pi/L\). The downsampler factor \(M \in \{6, 7, 8\}\) corresponds to the desired bandwidth, in MHz.

In MatLab the SRRCFIR filter order is determined by the product of the samples-per-symbol (SPS) and the symbol span. A cut-off frequency of \(\omega_c T = \pi/L\) yields \(SPS = L\). The span determines the number of symbols which the filter is truncated to. The SRRCFIR filter order is thus \(N = SPS \cdot span\).

The filter \(H_{ps}(z)\) yield \(SPS = 10\) and it is truncated to 400 symbols, \(span = 400\). \(H_{ps}(z)\) is constructed using \texttt{rcosdesign} and has a roll-off factor described by

\[
\Delta = \frac{f_d}{8} \frac{L}{M} - 1 = \frac{L}{8} - 1 = 0.25,
\]

where \(f_d\) is the desired bandwidth given in MHz. This \(\Delta\) becomes the same for all desired bandwidths, since \(f_d\) has the same value as \(M\). Fig. 3.3 shows the magnitude response \(|H_{ps}(e^{j\omega T})|\).
3.1 Signal Generator

The Pulse Shaping SRRCFIR-filter

\[ |H_{ps}(j\omega T)| \text{ [dB]} \]

\[ \omega T/\pi \]

\[ -200 \]

\[ -100 \]

\[ 0 \]

\[ 100 \]

\[ 200 \]

\[ 0 \]

\[ 0.2 \]

\[ 0.4 \]

\[ 0.6 \]

\[ 0.8 \]

\[ 1 \]

**Figure 3.3:** The SRRCFIR filter used to pulse shape the 64-QAM signals.

For the pulse shaped signals \( z_k(n) \) to have the same length, all \( x_k(n) \), representing the different bandwidths, must have different lengths. Each user signal \( z_k(n) \) has a length of \( l \) samples, which yield a number of \( l/8 \) samples per user signal when inputted to the FA, due to downsampling. Since \( z_k(n) \) are downsampled by a factor 8, any length evenly divisible by 8 is recommended. With a length of \( l \) samples per \( z_k(n) \), each 64-QAM symbol stream therefore has a length of \( (Ml - N_{ps})/L \), where \( N_{ps} \) is the order of the filter \( H_{ps}(z) \). As a standard, the length \( l = 3200 \) samples has been used, in this thesis work.

In Fig. 3.4 the spectra of the three different types of user signals are shown. The 64-QAM symbol streams \( x_k(n) \) and the narrow-band signals \( z_k(n) \) are to the left and right, respectively, in the figure. Note that the vertical axes are not the same.

**Figure 3.4:** The spectra of \( x_k(n) \) (left) and \( z_k(n) \) (right).
3.2 Aggregator

The digital frequency space available to each user is divided into $Q = 8$ granularity bands (GBs), leaving 8 GBs of width $2\pi/8$ equidistantly in $[0, 2\pi]$. It is assumed that users $f_{bw}(k) = 8$ cover all eight GBs, while users $f_{bw}(k) = 7$ and $f_{bw}(k) = 6$ cover seven and six GBs, respectively. Based upon the above assumptions, each user spectrum is divided using an AFB with $C$ channels. The outputs of these channel filters are subsequently downsampled by a factor $M = C/2$. To avoid aliasing effects and secure an efficient implementation, the number of channels are chosen as $C = 16$. As with the GBs, it is assumed that users $f_{bw}(k) = 8$ use all sixteen channels, while users $f_{bw}(k) = 7$ and $f_{bw}(k) = 6$ use fourteen and twelve channels, respectively.

The FA scheme, as mentioned, carries each user through an AFB, channelizing the users into 1/2 MHz sub-bands. Later all channelized users are aggregated through one SFB, composing the wide-band signal. The SFB is typically chosen to have $P = K \cdot C$ channels and upsampled by a matched factor $L = P/2$, to enable an efficient implementation. This enables full flexibility, low computational complexity, compared to the straightforward solution, and maintained quality.

3.2.1 Prototype Filters

Of interest, in this thesis work, are the symmetric linear-phase FIR filter of Type I [14]. These filters have a symmetric impulse response around $n = N/2$, i.e., $h(n) = h(N - n)$, where $n = 0, 1, \ldots, N$ and $N$ is even, see Fig. 2.1a, and they have integer valued delays.

The analysis prototype filter (APF) is independent of the number of users in the network, therefore the same AFB is used for every user signal. The synthesis prototype filter (SPF) however depends on the number of users. When designing the prototype filters, their maximum passband and stopband ripples are minimized, i.e. optimized in the minimax sense discussed in Section 2.1.2. The minimax optimization is carried out according to

$$
\begin{align*}
\text{minimize} & \quad \max_{\omega T} |E(\omega T)| \\
\text{subject to} & \\
|H(e^{j\omega T}) - e^{jD\omega T}| & \leq \delta_c, \quad \omega T \in [0, \omega_c T], \\
|H(e^{j\omega T})| & \leq \delta_s, \quad \omega T \in [\omega_s T, \pi], \\
\sum_{k=-1}^{1} & \left| H(e^{j\omega T} \cdot e^{j(\omega T-2\pi k/N)}) \right|^2 - 1 \leq \delta_p,
\end{align*}
$$

(3.2)

where $\delta_c, \delta_s, \delta_p \in \mathbb{R}$ are some small numbers and $E(\omega T) = e^{jW\omega T} \left[ H(e^{j\omega T}) - e^{jD\omega T} \right]$ is an error function. Here, $e^{jW\omega T}$ is a weighting function, specifying costs of deviation from the desired function $e^{jD\omega T}$. The filter frequency response $H(e^{j\omega T})$ is desired to approximate $e^{jD\omega T}$ in the passband. The third constraint checks the
Power Complementary

When constructing the prototype filters they are desired to be power complementary, i.e., the sum of the squared magnitude responses is approximately unity. However, the power complementary is approximated in the design of the prototype filters. In this thesis work, each prototype filter is power complementary with frequency shifted versions of itself as

$$S = \sum_{k=-1}^{1} \left| H(e^{j\omega T}, e^{j(\omega T - 2\pi k/N)}) \right|^2 \approx 1,$$

where $N$ is the number of desired channels of the FB and $H(e^{j\omega T})$ is the frequency response of a prototype filter. Since all filters in the FBs are frequency shifted versions of each other, it is enough that a filter and two shifted versions of itself are power complementary.

This approximation is based upon the fact that when the AFB channel filters are upsampled, by a factor $K$, they are approximately equal to the SFB channel filters. The case where $K = 1$ the APF and SPF are equal, and could therefore be power complementary with themselves, respectively. For larger $K$ the design of the APF pretend that the SPF is equal to the APF. This approximation is commonly used in this area of application [17].

Example

The used prototype filters when $K = 4$ are shown in Fig. 3.5. Note that the vertical axes are not the same, to include the whole filters.

![Graphs](image1)  
(a) Analysis prototype filter.  
(b) Synthesis prototype filter.

Figure 3.5: The prototype filters used for four users.
3.2.2 Efficient IDFT and DFT Based Implementation of the FBs

By making use of the polyphase form given in [17] the channel filters in (2.14) can be used to describe the analysis channel filters as

\[ G_k(z) = \beta_k \sum_{i=0}^{C-1} z^{-i} \alpha_i G_i(z^C W^i C) W^{-ik}, \quad k = 0, \ldots, C - 1, \quad (3.4) \]

where \( \alpha_i = W^{-\alpha_i} \) and \( G_i(z) \) are polyphase component filters of the APF \( G(z) \) as

\[ G(z) = \sum_{i=0}^{C-1} z^{-i} G_i(z^C). \quad (3.5) \]

By making use of (3.4) and (3.5), known properties of SRCs, IDFT and DFT FBs, each of the \( C \)-channel AFBs can be realized with the help of a \( C \)-point IDFT [10], illustrated in Fig. 3.6, where \( M = C/2 \).

\[ z_k(n) \quad \downarrow M \quad \downarrow M \quad \downarrow M \quad G_0 \left( z^2 W^\alpha C \right) \quad G_1 \left( z^2 W^\alpha C \right) \quad \cdots \quad G_{C-1} \left( z^2 W^\alpha C \right) \quad \downarrow \text{IDFT} \quad \beta_0 \quad \beta_1 \quad \cdots \quad \beta_{C-1} \]

**Figure 3.6:** The AFB block, based on an efficient IDFT implementation, with one input and \( C \) outputs.

The corresponding polyphase representation is used when implementing the synthesis channel filters. The \( P \)-channel SFB can be realized with the help of a \( P \)-point DFT, illustrated in Fig. 3.7, where \( L = P/2 \) and

\[ \gamma_k = \beta_k W^k_p, \quad k = 0, \ldots, P - 1. \quad (3.6) \]
3.2 Aggregator

3.2.3 Resulting Filter Banks

The resulting FBs with prototype filters as in Fig. 3.5 are illustrated in the figures below. In Fig. 3.8(a) is an AFB with \( C = 16 \) channels according to the AFB block in Fig. 3.6, where each filter \( G_c(z) \) utilizes (3.4). In Fig. 3.8(b) is a SFB with \( P = 64 \) channels used with four users according to the SFB block in Fig. 3.7, where each filter \( F_p(z) \) utilizes (3.4).
3.3 Frequency-Band Allocation Schemes

For this thesis work three different frequency-band allocation schemes (FBASs) are considered. These schemes place user signals in different parts of the outputted wide-band signal, referred to as Right, Left and Center. The only difference between the FBASs is that they group the user signals in different parts of the wide-band signal.

3.3.1 User Signal Distribution

Say that there are users according to $f_{bw} = \{7, 6, 8, 7\}$, i.e. $z_1(n)$ and $z_4(n)$ are 7 MHz signals, $z_2(n)$ is a 6 MHz signal, and $z_3(n)$ is an 8 MHz signal. These users will be placed in the order dictated by the vector distribution. With $distribution = \{1, 2, 3, 4\}$ the users are placed in the order $f_{bw}$ describes, see Fig. 3.9(a), but if e.g. the first and third user want to change positions, $distribution$ have to change to $distribution = \{3, 2, 1, 4\}$, see Fig. 3.9(b).

The vector $distribution$ contains values telling which user goes in position $k$, e.g. $distribution(1) = 3$ implies that user number three goes in the position one.

![Figure 3.9: An example of two different distribution vectors.](image)

The vector $distribution$ and the FBASs can together be seen as the channel select block in Fig. 2.20. The FBASs could possibly be used to make better use of the frequency space of the composite wide-band signal.
3.3 Frequency-Band Allocation Schemes

3.3.2 Right & Left

FBASs Right and Left gather the users in the rightmost and leftmost frequency space of the wide-band signal, respectively.

The set of SFB channels for each user signal, Right FBAS, are determined by the following algorithm

\[
\text{start}_r = \text{start}_r - 2f_{bw} (K - [\text{distribution}(k) - 1])
\]

\[
P_{\text{used}}(K - [k - 1]) = [\text{start}_r, \text{stop}_r - 1]
\]

\[
\text{stop}_r = \text{start}_r
\]

with starting values \(\text{stop}_r = P\) and \(\text{start}_r = \text{stop}_r\). As noticed the loop starts with the rightmost user and iterates to the left, which gives an easier implementation, since the need to calculate the number of unused channels is removed. Figure 3.10 shows an example of four users with \(f_{bw} = \{7, 6, 8, 7\}\) and \(\text{distribution} = \{2, 3, 1, 4\}\) using Right FBAS.

![Wideband output signal; users: 4, bandwidth: 32 MHz, orientation: Right](image)

**Figure 3.10:** Four users utilizing Right FBAS.

Likewise are the Left FBAS set of SFB channels for each user signal determined by the following algorithm

\[
\text{stop}_l = \text{stop}_l + 2f_{bw}(\text{distribution}[k])
\]

\[
P_{\text{used}}(k) = [\text{start}_l, \text{stop}_l - 1]
\]

\[
\text{start}_l = \text{stop}_l
\]

with starting values \(\text{start}_l = 0\) and \(\text{stop}_l = \text{start}_l\). In Fig. 3.11 an example of four users is shown with \(f_{bw} = \{7, 6, 8, 7\}\) and \(\text{distribution} = \{2, 3, 1, 4\}\) using Left FBAS.
Flexible Aggregator Scheme

3.3.3 Center

The FBAS Center gather all user signals in the center of the wide-band signal. Center calculates the number of unused SFB channels and place users at

\[ \text{stop} = \text{stop} + 2f_{bw}(\text{distribution}[k]) \]

\[ P_{\text{used}}(k) = [\text{start}, \text{stop} - 1] \]

\[ \text{start} = \text{stop} \]

with starting values \( \text{start} = \left\lceil \frac{P_{\text{unused}}}{2} \right\rceil - \text{mod} \left( \frac{P_{\text{unused}}}{2}, 2 \right) \) and \( \text{stop} = \text{start} \), leaving the most space to the right. The extra \( \text{mod} \left( \frac{P_{\text{unused}}}{2}, 2 \right) \) added to \( \text{start} \) will be explained in Section 3.3.4. \( P_{\text{unused}} \) is the number of unused SFB channels and it is determined as

\[ P_{\text{unused}} = P - 2 \sum_{k=1}^{K} f_{bw}(k), \quad (3.7) \]

where \( K \) is the number of users and \( P \) is the number of channels of the SFB. Center could be interpreted as Left which has been shifted to the right. An example is shown in Fig. 3.11 with the same four users as above, but with \( \text{distribution} = \{4, 1, 3, 2\} \), using Center FBAS.

Figure 3.11: Four users utilizing Left FBAS.

Figure 3.12: Four users using Center FBAS.
3.3.4 Observations

It is, even though desired to, not possible to place the users exactly anywhere in the output wide-band signal. Since the FBs consist of a multiple of $C = 16$ channels or $Q = 8$ GBs, it is not possible to move users an uneven amount of channels $c$. This is due to the size of each GB being 2 channels wide – users cannot be placed in the middle of a GB. This can be a problem when trying to implement FBAS which utilizes guard bands between each user. A guard band is a portion of the wide-band signal between each user which is heavily attenuated, in comparison to the user signals. This means that these guard bands have to be at least one GB, or 2 channels, wide. This is why the start value of $\text{start}$ in the Center scheme has to be an even number too, hence the extra mod $\left( \frac{\text{Pängeled}}{2}, 2 \right)$.

A potential solution is to frequency shift the user signals, which need to be placed in the middle of a GB, before inputting them to the FA. By shifting the center frequency by $\pi/8$ (the width of a FB channel) the user signal could be placed in the middle of a GB.

3.4 Receiver

The receiver consists of a signal extractor and a matched receiver filter, as in Fig. 3.13, where $\hat{z}(n)$ contains all signals $\hat{z}_k(n)$, $k = 1, \ldots, K - 1$, and $\hat{x}(n)$ contains all symbol streams $\hat{x}_k(n)$, $k = 1, \ldots, K - 1$.

![Figure 3.13: A schematic of the Receiver.](image)

The signals $\hat{z}_k(n)$ are extracted from the received wide-band signal $y(n)$. These $\hat{z}_k(n)$ are filtered to form 64-QAM symbol streams $\hat{x}_k(n)$. It is of interest to study the effects on the error vector magnitude (EVM), between the transmitted $x_k(n)$ and the received $\hat{x}_k(n)$, when $K$ increases.

The complexity of the receiver is not in focus. Therefore, filters in these steps are designed using a very high order in order to evaluate the FA.

3.4.1 Error Vector Magnitude

The error between the received and transmitted signals, is estimated using the EVM. The EVM in dB is given as

$$e_{\text{EVM}} = 10 \log_{10} \left\{ \frac{\mathbb{E} \{ |x(n)|^2 \}}{\mathbb{E} \{ |x(n) - \hat{x}(n)|^2 \}} \right\}, \quad (3.8)$$

where $\mathbb{E} \{ \cdot \}$ denotes the expectation value, whereas $x$ and $\hat{x}$ denote the transmitted and received signals, respectively. An EVM greater than 0 dB indicates that
there is more of the useful signal than of the noise. The EVM is used to measure error caused by filters and aliasing.

### 3.4.2 Signal Extractor

The role of the signal extractor is to extract all $K$ users $\hat{z}_k(n)$ from the received wide-band signal $y(n)$. A schematic of the sub-signal extractor is shown in Fig. 3.14, consisting of a modulator $e^{-jS_k n \pi}$, a variable FIR filter $H_K(z)$ and a downsampler of factor $K$.

\[
\begin{align*}
y(n) & \xrightarrow{K f_s} \hat{y}_k(n) \\
& \xrightarrow{K f_s} H_V(z) \\
& \xrightarrow{K f_s} \hat{z}_k(n)
\end{align*}
\]

**Figure 3.14:** A schematic of the sub-signal extractor.

The modulator $e^{-jS_k n \pi}$ shifts the center frequency of user $k$ down to $\omega T = 0$. To get the correct shift, the current orientation and bandwidth of each user is needed. The filter $H_V(z)$ depends on the user signal and therefore has a cut-off frequency $\omega_c = \frac{f_{bw}}{8} \pi / K$, and a roll-off factor $\beta \approx 0$. As an example, in Fig. 3.15 are filter used to extract users when $K = 4$ is shown.

When the signals $\hat{y}_k(n)$ later are downsampled by a factor $K$ their spectra are stretched from $[-\pi/K, \pi/K]$ to $[-\pi, \pi]$, resulting in the signals $\hat{z}_k(n)$.

Consider a wide-band signal $y(n)$ with $f_{bw} = \{8, 6, 7, 8\}$ as in Fig. 3.16, where the second user will be extracted.
Figure 3.16: A wide-band signal, here composed of 4 user signals.

Firstly the center frequency of the second user is shifted down to $\omega T = 0$, resulting in $\hat{y}_2(n)$ with spectrum as shown in Fig. 3.17(a). After both filtering and downsampling of $\hat{y}_2(n)$ its spectrum becomes as in Fig. 3.17(b), and furthermore the signal is referred to as $\hat{z}_2(n)$.

(a) The second user signal frequency shifted to $\omega T = 0$.

(b) The second user extracted.

Figure 3.17: The extraction of the second user signal.

The current extraction scheme gets increasingly slower with an increasing $K$. This is due to the number-of-users dependent order $N_{RC} = K \cdot 1000$, of the filter $H_V(z)$. With a higher order the EVM becomes barely some dB higher but the time it takes to extract each user increases. With a lower order the EVM drops a lot, in comparison to increasing the order. E.g. halving the order drops the EVM by 10 dB, instead of doubling the order which increases the EVM by approximately 1 dB. This however is not in focus when evaluating the FA.

3.4.3 Matched Receiver Filter

The schematic in Fig. 3.18 is used to acquire a 64-QAM symbol stream from a received narrow-band sub-signal. The pulse shaped signals $\hat{z}_k(n)$ are filtered to form $\hat{x}_k(n)$, which should approximate the transmitted signals $x_k(n)$. The filter $H_{mrf}(z)$ is matched to the pulse shape filter $H_{ps}(z)$, i.e., they are designed to be equal.
Flexible Aggregator Scheme

\[ \frac{L_{\text{in}}}{M} f_s \triangleq \omega_c T = \frac{\pi}{L} \]

Figure 3.18: A schematic of the matched receiver filter.

### 3.4.4 Constellation Scatter Plot

The received symbol streams for the example with four users given in Section 3.3, utilizing Center orientation, have an approximate EVM of 45 dB, compared to the transmitted symbol streams. The scatter plot of the received signals constellations are presented in Fig. 3.19.

(a) The first 7 MHz user signal.  
(b) The 6 MHz user signal.  
(c) The 8 MHz user signal.  
(d) The second 7 MHz user signal.

Figure 3.19: The constellations of the four received signals.
3.4.5 Error Calculations

The current code will, implementation wise, work for any number of users. But, the EVM will decrease with an increasing amount of users. This is most probably due to the summation of leakage, contributed by each user.

Say, \( K \) users has leakage power of \( P_{\text{leak}} \). For any \( N \cdot K \) users the leakage noise becomes multiples of \( P_{\text{leak}} \), i.e. \( N \cdot P_{\text{leak}} \). It is noted that for each added user, for \( 1 < K < 10 \), the EVM decreases by approximately 1 dB. Therefore for a multiple of 10 added users, the EVM decreases by approximately 10 dB. This behaviour is illustrated with the help of the data in Table 3.1, where the EVM is given by a mean. Three different constellations \( S_i \) of user configurations were used for each amount of user signals – e.g. for one user, the three constellations were \( S_1 = 6 \), \( S_2 = 7 \) and \( S_3 = 8 \). A variance is not calculated since a small amount of user constellations were used. However, note that each individual constellation yields the same result each time it is used. The mean is calculated as

\[
EVM_{\text{mean}} = \text{mean} \{ [E(S_1), E(S_2), E(S_3)] \}. \tag{3.9}
\]

Naturally the EVM in Table 3.1 is determined as \( EVM = EVM_{\text{mean}} \).

<table>
<thead>
<tr>
<th>User Signals ([K])</th>
<th>EVM ([\text{dB}])</th>
<th>User Signals ([K])</th>
<th>EVM ([\text{dB}])</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>49.7</td>
<td>10</td>
<td>39.9</td>
</tr>
<tr>
<td>2</td>
<td>48.6</td>
<td>20</td>
<td>38.8</td>
</tr>
<tr>
<td>3</td>
<td>46.7</td>
<td>30</td>
<td>37.3</td>
</tr>
<tr>
<td>4</td>
<td>45.9</td>
<td>40</td>
<td>31.4</td>
</tr>
<tr>
<td>5</td>
<td>44.6</td>
<td>50</td>
<td>37.7</td>
</tr>
<tr>
<td>6</td>
<td>43.3</td>
<td>60</td>
<td>34.1</td>
</tr>
<tr>
<td>7</td>
<td>42.5</td>
<td>70</td>
<td>31.9</td>
</tr>
<tr>
<td>8</td>
<td>41.5</td>
<td>80</td>
<td>30.8</td>
</tr>
<tr>
<td>9</td>
<td>40.6</td>
<td>90</td>
<td>29.4</td>
</tr>
<tr>
<td>10</td>
<td>39.9</td>
<td>100</td>
<td>30.9</td>
</tr>
</tbody>
</table>

*Table 3.1: The errors between transmitted and received signal with an increasing amount of users.*

**Increased Prototype Filter Order**

A way to increase the EVM from the levels given in Table 3.1 is to double the prototype filter orders. The order of the synthesis prototype filters is doubled to \( N_s = 128K \), \( K \) users. To ensure a number of \( 16K \) channels, the number of filter taps of each channel filter is increased to \( R = 8 \). The order of the analysis prototype filter is increased to \( N_a = 128 \), also doubling the number of taps.

When utilizing these doubled filter orders, the stopbands of the prototype filters yield an increased attenuation of 100 dB, instead of the 60 dB attenuation without the doubled filter orders, see Fig. 3.20.
The best results are given when using 8 MHz user signals, when double filter orders are used. The EVM of an 8 MHz user, compared to 6 MHz or 7 MHz user, can be 10 dB higher. The EVM in Table 3.2 are the result when both prototype filters have double orders.

<table>
<thead>
<tr>
<th>User Signals [K]</th>
<th>EVM [dB]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>81.7</td>
</tr>
<tr>
<td>2</td>
<td>55.9</td>
</tr>
<tr>
<td>3</td>
<td>55.7</td>
</tr>
<tr>
<td>4</td>
<td>54.8</td>
</tr>
</tbody>
</table>

Table 3.2: The errors between transmitted and received signal when utilizing higher prototype filter orders.

It is evident that the EVM is higher when the prototype filter orders are increased. To achieve a proper and successful filter design the resolution in the passband have to be increased, or else the design might fail. These improved filters, even though they yield a better EVM, are more expensive implementation wise, i.e. the filter design would take even longer than those described in Section 4.2.

Variable Signal Length

A potential way to increase the EVM could be to let the user signal length increase. The length used has been \( l = 3200 \) samples for this thesis work, but for this section other lengths have been tested. In Table 3.3 are the resulting error calculations, when \( K = 4 \).
### Table 3.3: The errors between transmitted and received signal when varying the user signal lengths, for $K = 4$.

<table>
<thead>
<tr>
<th>Signal Length [samples]</th>
<th>EVM [dB]</th>
</tr>
</thead>
<tbody>
<tr>
<td>800</td>
<td>42.3</td>
</tr>
<tr>
<td>1600</td>
<td>42.4</td>
</tr>
<tr>
<td>3200</td>
<td>41.9</td>
</tr>
<tr>
<td>6400</td>
<td>41.8</td>
</tr>
<tr>
<td>12800</td>
<td>41.7</td>
</tr>
<tr>
<td>32000</td>
<td>41.6</td>
</tr>
<tr>
<td>64000</td>
<td>41.6</td>
</tr>
</tbody>
</table>

As seen in Table 3.3 the EVM do not depend significantly on the length of the user signals. However, the EVM does decrease with an increasing length, and increase with and decreasing length. By this comparison, it appears to be no major difference in the resulting EVM, if the length of the user signals are varied.

### 16-QAM Signals Comparison

The most commonly used formats by which video signals are coded are 16-QAM and 64-QAM. In this thesis work 64-QAM signals are used, and there should be no significant difference between using either of the two formats. In Table 3.4 the EVM between the two formats are compared.

<table>
<thead>
<tr>
<th>User Signals [K]</th>
<th>EVM [dB], 16-QAM</th>
<th>EVM [dB], 64-QAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>47.9</td>
<td>47.8</td>
</tr>
<tr>
<td>2</td>
<td>45.4</td>
<td>45.2</td>
</tr>
<tr>
<td>3</td>
<td>43.3</td>
<td>42.9</td>
</tr>
<tr>
<td>4</td>
<td>42.2</td>
<td>41.9</td>
</tr>
</tbody>
</table>

*Table 3.4: The errors between transmitted and received signal when using both 16-QAM and 64-QAM signals.*

As seen in Table 3.4 the difference between the two formats is insignificant. However, the EVM is higher by a small amount when using 16-QAM signals, which could be because 16-QAM is the modulation with a smaller library.

### 3.5 Straightforward Solution

The straightforward solution to the aggregator problem consists of four different components; upsamplers, variable filters, modulators and adders, see Fig. 3.21.
The upsampler will compress the spectrum of each $z_k(n)$ from $[-\pi, \pi]$ to $[-\pi/K, \pi/K]$. Each filter $H_k(z)$ is specifically designed for each type of user signal, i.e. 6, 7 or 8 MHz. If, for instance, the signal $z_1(n)$ at first is a 6 MHz signal a filter $H_1(z)$ is designed. If then another signal is inputted in the place of $z_1(n)$, e.g. an 8 MHz signal, then the filter $H_1(z)$ would have to be re-designed. The same applies to all filters $H_k(z)$, which can be polyphase decomposed to make the implementation more efficient. However the straightforward solution will execute all operations at a higher sample rate than the efficient solution. The modulators $e^{-jC_kn\pi}$ shift user $z_k(n)$ to a designated frequency space of the wide-band signal $y_k(n)$. The final composite wide-band signal $y(n)$ is the summation of all $y_k(n)$.

### 3.6 Computational Complexity

The computational complexity, in terms of operations (multiplications and additions) per sample, will be considered in this section. The complexity of the FA scheme, presented in this thesis work, will first be calculated and secondly the complexity of the straightforward solution to the aggregation problem will be computed. The section is concluded by a comparison between the two computational complexities.

#### 3.6.1 Flexible Aggregator Scheme

Due to the polyphase decomposition, of the filter banks in the aggregator, all operations take place at a sample rate $f_s = 1$ MS/s. Each of the AFBs for each sample has a complexity of $C \times R + C \times (\log_2 |C| - 2) / 2$. Here $C \times (\log_2 |C| - 2) / 2$ comes from the IDFT of the realization and $R$ is the number of filter taps used for...
3.6 Computational Complexity

each channel filter, which sum to

\[ C \times K \times R + C \times K(\log_2{C} - 2)/2 \]  (3.10)

for all the AFBs, at the rate \(f_s\). The SFB on the other hand has a complexity of

\[ P \times R + P \times (\log_2{P} - 2)/2 + P, \]  (3.11)

at the rate \(f_s\), where \(P = K \times C\) are the number of channel filters in the SFB and \(P \times (\log_2{P} - 2)/2\) comes from the DFT of the realization. The extra \(P\) multiplications come from the constants \(\gamma_k\). In total, by combining (3.10) and (3.11), the computational complexity \(C_{fa}\) becomes

\[ C_{fa} \approx (2 \times R + \log_2{K}/2) \times 16 \times K \times f_s, \]  (3.12)

with \(C = 16\). The computational complexity is illustrated in Fig. 3.22.

![Computational complexity of the FA scheme.](image)

**Figure 3.22:** Computational complexity of the FA scheme.

### 3.6.2 Straightforward Solution

The straightforward solution will commit all operations at a higher rate, than the FA scheme, that is \(8f_s = 8\) MS/s. The filter order of the bandpass filters are roughly the same as the filters used in the SFB, i.e. \(R \times P\). The complexity \(C_{sf}\) therefore becomes

\[ C_{sf} \approx R \times K^2 \times 128f_s, \]  (3.13)

with \(C = 16\) and where the extra 1 multiplication comes from the modulators. The complexity \(C_{sf}\) is illustrated in Fig. 3.23, which makes it clear that \(C_{sf}\) grows fast.
3.6.3 Comparison

The computational complexity ratio between the FA with efficient DFT and IDFT implementation and the straightforward solution is

\[
\frac{C_{fa}}{C_{sf}} \approx \frac{(2 \times R + \log_2 \{K\}/2)}{R \times K \times 8} \approx \frac{1}{4K} + \frac{\log_2 \{K\}}{R \times K \times 16},
\]

which means that even when \( K \) is small the computational complexity will be much less for the FA than the straightforward solution. If the number of users is increased to a more realistic situation, say \( K = 100 \), then the FA would have 2 - 3 orders of magnitude better computational complexity compared to the straightforward solution. The ratio is illustrated in Fig. 3.24.

\[
\text{Figure 3.23: The computational complexity of the straightforward solution.}
\]

\[
\text{Figure 3.24: The computational complexity ratio between the solutions.}
\]
As evident by Fig. 3.24 the computational complexity of the straightforward solution grows much faster than that of the FA with efficient DFT and IDFT. E.g. the number of operations for the FA solution, when \( K = 100 \), is smaller than the number of operations for the straightforward solution, when \( K = 7 \). An illustration of this is seen in Fig. 3.25.

**Figure 3.25:** Comparison of computational complexity between the solutions.
This chapter starts with the main interface between user and the implemented multi-standard FA. Possible ways to implement and design filters with the current code conclude the chapter.

4.1 Main Interface

The main user interface is a MatLab script to access the main function. There are three parameters here to achieve different simulations. An excerpt of the file main.m is shown in Fig. 4.1.

![Figure 4.1: An excerpt of the Matlab file of the user interface, main.m.](image)

The first called *users* is a vector containing three or no elements. The first through third element represent how many 6, 7 and 8 MHz sub-signals, respectively, to be used. While *users* is empty, the user signals are defined by the second parameter, instead.

The second parameter called *Dist* is also a vector, but it contains a distribution telling which user goes where in the wide-band signal *y(n)*. When *Dist* is empty the distribution is randomized, or when *users* is empty *Dist* must contain elements equal to the number of users *K*. The elements must be either 6, 7 or 8
to distribute the different types of signals in the order which Dist dictates. The
users can also be grouped together by letting Dist only contain three elements,
either 6, 7 or 8. E.g. Dist = [6, 7, 8] group user signals together as dictated by
Dist, i.e. 6s to the left, 7s in the middle and 8s to the right.

The third parameter is the string called Orientation which can take the values
‘Center’, ‘Left’ or ‘Right’. This groups all users towards the given direction.

4.2 User Filter Designs

With an increasing $K$ the synthesis prototype filter will become more and more
narrow. The optimal filter design will therefore consume more and more time.
With a larger $K$ the size of the stopband increases, therefore must the resolution
of the stopband be larger. This means that the stopband has to be made up of
more coefficients for the filter design to be successful. If the resolution would
be too small, the stopband of the filter rise to approximately 0 dB, rendering the
filter useless.

The file create_filters_to_be_used.m exist for the user to create new or re-create
already existing filters. The file features creation of prototype filters (both analysis
and synthesis), synthesis filter banks and extraction filters outside of the
main function. To change the resolution of the pass-band of the prototype filters,
change the length of the vector $wT$ in the file lowpassFIR.m, marked in red in Fig.
4.2.

```
function h = lowpassFIR(Nh,wT,wH,order2k)

% This function takes a filter order N, passband edge wcl and stopband edge wcl.
% and the arrival error between N and an ideal lowpass filter. 

% Define parameters
w = (1-rho)*pi/C;
wc = (1-rho)*pi/C;

% Initial solution filter estimate
h = firpm(Nh,fo,wo,w);

% Power spectrum of filter

% Defining a filter

% Closest frequency equal to

% Minmax optimization of prototype filter

% Desired filter (optimal lowpass filter)

% Extract final filter
```

**Figure 4.2:** An excerpt of the Matlab file lowpassFIR.m.

The resolution of the pass-band can be reduced when $K$ is increased, since the
pass-band becomes more and more narrow.

To change the resolution of the stop-band of the prototype filters, change the
length of the vector $wH$ in the file constr_lp.m, marked in red in Fig. 4.3.
4.2 User Filter Designs

Figure 4.3: An excerpt of the Matlab file constr_lp.m.

The resolution of the stop-band can be increased when $K$ is increased, since the stop-band becomes wider and wider.

The time required increase from a few minutes ($K \leq 10$) to a few hours ($10 < K < 30$). Prototypes for $30 \leq K < 60$ take somewhere in the magnitude of half a day. For the largest $K$s the magnitude of time required is days, where a prototype filter for $K = 100$ can take up to five days to create.

Successfully designed synthesis prototype filters are those used for $K = 1, 2, \ldots, 50, 60, \ldots, 100$ and the analysis prototype filter.
Conclusions

The conclusion made will be considered in this chapter. An overview of what has been done is followed by a discussion regarding the results. Potential future work concludes both the chapter and the technical report.

5.1 Overview

By utilizing oversampled complex modulated FBs and inner channelizers, the number of operations performed per sample can be reduced. All frequency components are delayed equally, and the signal shape is preserved, since the prototype filters are linear-phase FIR filters. By designing the prototype filters using the MPR algorithm a minimax optimization is secured. The FB channel filters are made more efficient with the use of a DFT and IDFT based implementation.

To be able to test the multi-standard FA, two additional sub-systems are constructed. The first additional sub-system is a signal generator, generating narrowband signals $z_k(n)$ by pulse shaping 64-QAM symbol streams $x_k(n)$. The second additional sub-system is a receiver, generate 64-QAM symbol streams $\hat{x}_k(n)$ from extracted signals $\hat{z}_k(n)$ from the wide-band signal $y(n)$. The EVM is calculated between the transmitted and received symbol streams.

The FA scheme has a high flexibility thanks to the FBASs Right, Left and Center. By using these FBASs the user signals can be grouped together in the leftmost, rightmost or center part of the wide-band signal, respectively. It is also possible to distribute the user signals as deemed best fit.

The computational complexity, in terms of performed operations (multiplications and additions) per sample, of the implemented FA scheme is in focus. The scheme has 2 - 3 orders-of-magnitude better complexity than the straightforward solution, when $K$ is increased to a more real life case.
5.2 Results

The FA solution used in this thesis work is an efficient way to achieve an multi-standard and flexible aggregation of narrow-band signals. The scheme has proven to be efficient for an increasing number of transmitted narrow-band signals, where the EVM between transmitted and received signals also is high.

The computational complexity in terms of performed operations per sample, of the efficient solution, is by 2 - 3 orders of magnitude superior to that of the straightforward solution.

There is no significant difference between the 16-QAM and 64-QAM coded user signals. When varying the length of the user signals, no major difference was present in the result. However, the EVM can be increased if the prototype filter order is increased.

5.3 Future Work

Potential future work will be discussed in this section. A potential future work could be to implement an alternative receiver, investigate how the implementation of prototype filters with higher orders can be carried out or an additional FBAS.

5.3.1 Alternative Extractor

A possible alternative extractor could be similar in appearance to the transmitter, i.e., a solution that is based on FBs. The current implementation of the extractor is a straightforward solution.

5.3.2 Higher Prototype Filter Orders

The prototype filter orders $N_a = 128$ and $N_s = 128K$ has briefly been tested in this thesis work. A potential future work could be to investigate the possibility to use some arbitrary prototype filter orders, e.g. $N_a = 96$ and $N_s = 96K$.

5.3.3 Frequency-Band Allocation Scheme

One additional FBAS could be considered in the FA scheme. To utilize guard bands between each user both lower introduced errors between signals but also makes it easier to distinguish different users from each other. The concept of an FBAS, here called, Guard is considered below.

Guard

The FBAS Guard leaves a guard band between each user signal pair. This makes the task of detecting the different users easier, but the total occupied space of the
wide-band signal cannot be optimized as easy. Each user passes through the set of SFB channels

\[ P_{used}(k) = [(k - 1)C + C_{unused}(k)/2, kC - C_{unused}(k)/2 - 1], \]  

(5.1)

where \( C_{unused}(k) \) is the number of unused channels of user \( k \), determined by

\[ C_{unused}(k) = C - 2f_{bw}(k), \]  

(5.2)

where \( f_{bw} \) is a vector containing user bandwidths in MHz, i.e., \( f_{bw}(k) \in \{6, 7, 8\} \).

Figure 5.1 shows an example with four users, \( f_{bw} = \{6, 8, 6, 7\} \), with distribution = \( \{4, 2, 1, 3\} \) using Guard.

\[ \text{Figure 5.1: Four users using Guard with distribution} = \{4, 2, 1, 3\}. \]

This FBAS could also be combined with the current FBASs Right, Left and Center. The combination would in this case make the FBAS Guard act as a parameter which is either true or false.


Analysis Filter Bank, ix, 4, 18, 26
Analysis Prototype Filter, ix, 26
Discrete Fourier Transform, ix, 21
Error Vector Magnitude, iii, ix, 1, 33, 37
Filter Bank, ix, 17, 29
Finite-length Impulse Response, ix, 7, 9
Flexible Aggregator, ix, 1, 19, 50
Frequency-Band Allocation Scheme, 30, 50
Frequency-band Allocation Scheme, ix
Granularity Band, ix, 26, 33
Intersymbol Interference, ix, 20, 21
Inverse Discrete Fourier Transform, ix, 21
McClellan-Parks-Rabiner (algorithm), ix, 9
Quadrature Amplitude Modulation, ix
Raised Cosine Finite-length Impulse Response, ix, 20
Sample Rate Converter, ix, 2, 10
Square-root-raised Cosine Finite-length Impulse Response, ix, 21, 24
Synthesis Filter Bank, ix, 4, 19, 26
Synthesis Prototype Filter, ix, 26