Orthogonally Multiplexed Communication
Using CCSK and Wavelet Bases

A Thesis Presented to
The Faculty of the Russ College of Engineering and Technology
Ohio University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science

By
Ji-Dein Wu
November, 1995
Acknowledgments

The author wishes to acknowledge Dr. Jeffrey Dill, the thesis chairman, for his patience and guidance during the development of this thesis. His knowledge of the thesis subject and support in its development stages were invaluable. His precious advice, unceasing instruction and continuous encouragement contributed greatly to the completion of this masters degree.

The author also wishes to acknowledge the contributions made by others to the preparation and completion of this masters thesis. Special thanks are given to Dr. Michael Braasch, Dr. Jeffrey Giesey, and Dr. Jeffrey Connor for serving as members of the thesis committee. Their willingness to be part of the review process is strongly appreciated.

Finally, the author expresses his sincere appreciation to his family and friends for their support. Two persons, in particular, deserve the author’s gratitude for their extraordinary patience and support, financially and spiritually: his mother and sister. Without them, the author might never have completed his study at Ohio University.
Table of Contents

Acknowledgments .......................................................................................................................... iii

List of Figures ............................................................................................................................... vi

List of Tables ................................................................................................................................. ix

Chapter 1 Introduction .................................................................................................................. 1

Chapter 2 Implementation of the CCSK System ............................................................................ 7
  2.1 Introduction to CCSK and TD ............................................................................................... 7
  2.2 Generation of CCSK Waveforms ......................................................................................... 8
  2.2.1 Binary Maximum Length Sequences ............................................................................. 8
  2.2.2 Improvement for CCSK System ................................................................................... 13
  2.3 Implementation of CCSK System ....................................................................................... 13
    2.3.1 Circular Auto Correlation of CCSK System ................................................................. 16
    2.3.2 Processing Gain .......................................................................................................... 17
  2.4 CCSK/TD System Implemented with Transmitted Reference ........................................... 17
  2.5 Multicarrier Implementation ............................................................................................... 20

Chapter 3 Multiresolution, Filter Banks and Wavelets ............................................................... 22
  3.1 Multiresolution .................................................................................................................... 22
3.2 Quadrature Mirror Filter Bank .................................................. 23
  3.2.1 The Odd Order Filter ..................................................... 26
  3.2.2 The Even Order Filter .................................................... 29
3.3 Wavelet Analysis .............................................................. 31
  3.3.1 Essential Results from MRA .......................................... 32
  3.3.2 Relationship Between Wavelets and Filter Banks .......... 34
  3.3.3 Varieties of wavelet Catalogued ................................... 38

Chapter 4 Periodic Discrete Wavelet Transform and its Fast Algorithm ..... 45
  4.1 Periodic Discrete Wavelets ................................................ 46
  4.2 Periodic Discrete Wavelet Transform .................................. 48
  4.3 Fast Algorithm .................................................................. 50
    4.3.1 Orthogonality of Wavelet Bases .................................... 58
    4.3.2 Computational Cost .................................................... 60
  4.4 Comparison of Results from PRPA and ARPA ..................... 62
    4.4.1 Wavelet Basis Functions ............................................ 62
    4.4.2 Reconstructed Signal ................................................ 73
  4.5 Gray Coding .................................................................... 75

Chapter 5 Conclusion .................................................................. 78

References .................................................................................... 82

Appendix ...................................................................................... 87
List of Figures

Chapter 2
Figure 2.1: Shift Register With Generating Polynomial \( g_1(x) = (x^3 + x^{10}) \) ............ 10
Figure 2.2: Shift Register With Generating Polynomial
\( g_2(x) = (x^2 + x^3 + x^6 + x^8 + x^9 + x^{10}) \) .................................................. 10
Figure 2.3: Circular Auto-Correlation of a BMLS of Length 1023 ................. 11
Figure 2.4: Circular Auto-Correlation of a BMLS of Length 1024 ................. 14
Figure 2.5: Circular Auto-Correlation of a Random Uniform Signal of Length
1024 .................................................................................................................. 14
Figure 2.6: CCSK Transmitter ................................................................. 15
Figure 2.7: CCSK Receiver ................................................................. 15
Figure 2.8: CCSK/TD Transmitter With Transmitted Reference Signal .......... 18
Figure 2.9: CCSK/TD With Transmitted Reference Signal Receiver .............. 18
Figure 2.10: Multicarrier Transmitter ....................................................... 20
Figure 2.11: Multicarrier Receiver ......................................................... 21

Chapter 3
Figure 3.1: Two Channels Quadrature Mirror Filter ............................... 24
Figure 3.2: Frequency Response of Quadrature Mirror Filter .................... 25
Figure 3.3: Daubchies Filter With Order 35 ............................................. 28
Figure 3.4: Squared Root Raised Cosine Filter With Order 36 .................... 30
Figure 3.5: The Calculation of Wavelet Expansion Coefficients Implemented as Analysis Filter Bank .......................................................... 37

Figure 3.6: The Calculation of Scaling Coefficients Implemented as Synthesis Filter bank ........................................................................... 38

Figure 3.7 Dyadic Wavelet With Non-uniform Tree Structure ..................... 39

Figure 3.8 Wavelet With Uniform Tree Structure ...................................... 40

Figure 3.9 M-Bands Wavelet ..................................................................... 42

Figure 3.10 Wavelet Packet ....................................................................... 43

Chapter 4

Figure 4.1: One Stage of the Periodic Wavelet ........................................ 47

Figure 4.2: One Stage of the Inverse Periodic Wavelet .............................. 47

Figure 4.3: Uniform Tree Structured Wavelets With Transformation Depth J=4 48

Figure 4.4: Diagram for Periodic Recursive Pyramid Algorithm ................ 53

Figure 4.5: Basis Functions Constructed From PRPA Algorithm With Square Root Raised Cosine Filter With Length of 37 ................................. 56

Figure 4.6: Basis Functions Constructed From PRPA Algorithm With Daubchies Filter With Length of 36 .................................................. 57

Figure 4.7: Orthogonality With Filter Coefficients .................................... 59

Figure 4.8: Basis Functions Constructed From ARPA With Square Root Raised Cosine .................................................................................. 63

Figure 4.9: The Periodic Sum from the First 16 Sections ............................ 64

Figure 4.10: The Periodic Sum from the First 18 Sections ......................... 65

Figure 4.11: The Periodic Sum from the First 20 Sections .......................... 66

Figure 4.12: The Periodic Sum from the First 22 Sections ....................... 67

Figure 4.13: The Periodic Sum from the First 24 Sections ....................... 68

Figure 4.14: The Periodic Sum from the First 26 Sections ....................... 69

Figure 4.15: The Periodic Sum From All Divided Sections from PRPA With
Square Root Raised Cosine Filter With Length of 37 .................. 70
Figure 4.16: Basis Functions Constructed from ARPA With Daubschies Filter With Length of 36 ................................................................. 71
Figure 4.17: The Periodic Sum From All Divided Sections from ARPA With Daubschies Filter With Length of 36........................................... 72
Figure 4.18: Comparison of the Output for Different Algorithm .................. 74
Figure 4.19: Subband Disordering in Uniform Tree Structure .................... 75
List of Tables

Table 2.1: Example of the number of sequence for sequence of different lengths .................................................. 12
Table 4.1: Example of the TCC for the uniform DWT and FFT ......................... 61
Introduction

In view of the exponential growth of wireless services, one consequence of this growth is that portions of the radio spectrum are very crowded with users; it causes spectral congestion and interference [1]-[4]. One possible solution is spread spectrum, a modulation scheme where the signal bandwidth is much greater than the information bandwidth. Spread spectrum provides a means of intentionally spreading the signal bandwidth without changing the information bandwidth. It is well known that a spread spectrum system has several desirable characteristics, such as antijamming capability, multiple access capability and robustness against multipath fading. Various techniques have been proposed to enhance system performance over and above that offered by the processing gain. For example, the linear and nonlinear filter, and filter banks schemes are applied to suppress strong
narrowband interference [5]-[8], and a rake receiver is used to enhance performance in the presence of frequency selective fading [9]-[10].

Several methods of spread spectrum are currently available, such as direct sequence, frequency hopping and time hopping [11], and a relatively recent spread spectrum system using a cyclic code shift keying (CCSK) implemented with transform domain and rake processing [12]. CCSK is used to modulate the signal in an M-ary orthogonal, or near orthogonal signaling scheme. In this method, a base-vector of length M is first generated. The remaining M-ary signal set is then generated by taking the cyclic shift of the base-vector. In general, the base vector space of the CCSK need not be limited to the binary field, as is the case in direct sequence (DS), and can be expanded to a complex number field. This allows users to achieve better auto-correlation properties.

When a signal is transformed to the frequency domain using a discrete Fourier transform (DFT), the signal processing is said to have been performed in the Transform Domain (TD). Transform domain processing can be utilized to suppress undesired interference and, consequently, to improve system performance. After the TD operation, the signal then is transformed back into a time domain signal using an inverse DFT. With a time invariant impulse response, the CCSK/TD scheme can achieve bit error performance equal to M-ary orthogonal signaling. Furthermore, by adding the reference signal in the transmitter, the system is sufficiently robust to be operated under severe multipath situations. This
process is suitable for time-varying channel conditions, and makes the system relatively immune to imperfect time synchronization.

If the inverse transform operation is implemented in the transmitter, the CCSK/TD system also supports multicarrier modulation (MCM) to form a orthogonally multiplexed communication system [13]. Multicarrier modulation is performed on M-bit symbols from the sequence of binary digits at a time. Because a multicarrier signal is integrated over a long symbol period, the effects of impulse noise are much less than that for a carrier signal. This was one of the original motivations for MCM.

There is an interesting time/frequency duality involved. For example, a complex quadrature amplitude modulation (QAM) signal is sensitive to impulses in the time domain. On the other hand, an MCM signal might be sensitive to impulses in the frequency domain, namely single tone interference. In contrast to the time of occurrence of impulses in the time domain, MCM’s advantages lie in the fact that the sources of these interferences are discrete, and their frequencies are usually stable. Hence, these interferences can be recognized during training [13].

With MCM, the data bandwidth is split into uniform bands and is achieved with reduced complexity using an efficient fast Fourier transform (FFT) based implementation [14]. Another transform, referred to as discrete wavelet transform (DWT), can be used to replace the FFT process. Like the FFT, the DWT is a fast algorithm involving signal decomposition onto a set of basis functions. This DWT
can also be realized in a single chip process [15]. However, unlike FFT, the basis functions (called wavelets) underlie the wavelet analysis; orthonormal wavelets represent finite energy of signals in $L^2(R)$ but not in $L^2(0,2\pi)$ [16]-[19]. In addition, orthonormal wavelets contrast with the traditional Fourier expansion that can only represent signals with complex exponentials as basis functions. The basis functions in the wavelet expansion are obtained by a scaling function and a set of wavelet functions which translate a single prototype wavelet by dilations and scalings as well as shifts [19]. Thus, the wavelet basis is not unique and offers freedom of choice. In view of the signal processing, the prototype wavelet can be thought of as a bandpass filter; others are scaled versions of the prototype.

Wavelets were taken to a new level of development when multiresolution analysis (MRA) was introduced by Mallat [18] and [19]. Multiresolution representations, which are very effective when analyzing information [20], approximate a signal at a given resolution in which the resolution can be loosely thought of as bandwidth. Differences in information between the approximation of a signal at resolutions $2^{i+1}$ and $2^i$ can be extracted by decomposing the signal on a wavelet orthonormal basis of $L^2(R)$. In addition, MRA analysis also produces a fundamental connection to discrete wavelet analysis, and to a major branch of signal processing called quadrature mirror filter (QMF) banks in a multirate system [21]-[29]. In signal processing, the DWT is seen as an analysis filter bank and the inverse DWT is the corresponding synthesis filter bank. Both analysis and
synthesis are connected to construct the condition for perfect reconstruction with the power complementary property and admissibility condition.

Most work on discrete wavelet analysis is implemented with different structured filter banks focusing on the infinite interval, and which are aperiodic [23]. Recently, another kind of wavelet has been emphasized, called periodic wavelets for finite interval signal [16], [17], [23], [30]-[31]. Instead of $L^2(R)$, the signal is decomposed on a wavelet orthonormal basis of $L^2(0,1)$. The properties for periodic wavelets are almost the same as for aperiodic wavelets. The discrete wavelet transform is called the periodic DWT, or DWT for finite duration [30].

In this thesis, the concepts of CCSK and discrete wavelet transform will be brought together to form an orthogonally multiplexed communication system. Also, the fast algorithm for periodic DWT and inverse DWT will be derived. The source codes developed for the fast algorithms in this thesis are written in MATLAB-callable C language, referred to as MEX-files, while other source codes are written in MATLAB language.

The following chapters are now discussed. Chapter 2 will briefly review the basic theorem of CCSK modulation and CCSK/TD implementation. Improvement in generating base vectors is introduced. Also, the CCSK/TD approach is modified to a multicarrier modulation structure to form an orthogonally multiplexed communication system. Chapter 3 reviews multiresolution representations, quadrature mirror filter banks and aperiodic discrete wavelets. The mathematical
proof for the relationship between filter banks and discrete wavelets is derived. The fast recursive pyramid algorithm for periodic discrete wavelet transform and its inverse is found in Chapter 4. Results are compared with those from the recursive pyramid algorithm for aperiodic DWT. Chapter 5 includes summary comments and conclusions.
Chapter 2

Implementation of the CCSK System

2.1 Introduction to CCSK and TD

Cyclic code is an important class of codes. It can be efficiently encoded by means of a device called a linear shift register. Other decoding schemes utilize shift registers. This encoding scheme is efficient since no storage is required because the code words are generated by shifting and adding [33]. Cyclic Code Shift Keying (CCSK), derived from a cyclic code, is a form of spread spectrum [12]. In CCSK, the signal is an M-ary orthogonal or near orthogonal signaling scheme in which it is spread in a manner like that of DS. One needs first to construct a base-vector of length M, and then take the cyclic shift of the base-vector to form the M-ary orthogonal signals. For a digital communication system, transform domain (TD) signal processing is used to preprocess received signals before demodulation. The received signal is sampled first, and then transformed
into the frequency domain using DFT implemented as an FFT. In the frequency
domain, narrowband interference is suppressed. The signal is then transformed
back into a time domain signal using inverse DFT implemented with inverse FFT,
and then demodulated by conventional means.

2.2 Generation of CCSK Waveforms

Because CCSK is designed to transmit signals which are the circular-shifted
version of the base vector, the CCSK waveforms depend on the base vector.
Several kinds of possible waveforms can be base vectors, such as Linear Frequency
Modulation (FM) signals used in radar applications, Filtered Impulse Trains based
on Nyquist sampling theorem, and Binary Maximum Length Sequences (BMLS)
[12]. To better understand CCSK, the BMLS waveform will be reviewed. BMLS
are commonly used because they are easily generated in shift registers with a
relatively small number of stages, and have good circular auto-correlation
properties.

2.2.1 Binary Maximum Length Sequences

Cyclic codes resulting from maximum-length linear shift register sequences
must satisfy the following conditions [11]:

(1) In each period of the sequence, the number of plus ones differ from the
number of minus ones by exactly one.
(2) Among the runs of plus Ones and minus Ones, in each period one half of
the runs of each kind are of length one, one fourth of each kind are of length two,
one eighth of length three, and so on. Thus, the statistical distributions are well
defined and always the same, but the relative position of the runs vary from code to
code.

(3) Comparing the period sequence term by term with any cycle shift of
itself, the number of terms which are the same differ from those which are different
by at most 1. This implies that auto-correlation is two-valued and can be described
by

\[ R_x(k) = \sum_{n=1}^{N} x_n x_{n+k} \]

\[ = N \quad k = 0, N, 2N, \ldots \] \hspace{1cm} (2.1)

\[ = -1 \quad \text{otherwise} \]

and

\[ x_n = x_{n+N} \]

Where \( x_n \) and \( x_{n+k} \) represent the cyclic code in the \( n \)'th and \((n+k)\)'th
position, and the \( N \), the length of cyclic code.

BMLS may be considered to be a vector \( V=(b_0, b_1, \ldots, b_{n-2}, b_{n-1}) \)
corresponding to a polynomial \( V(x)=b_0+b_1x+\ldots+b_{n-2}x^{n-2}+b_{n-1}x^{n-1} \). The first cyclic
shift version \( V_1=(b_1, b_2, \ldots, b_{n-1}, b_0) \) then corresponds to \( V_1(x)=b_{n-1}+b_0x+\ldots+b_{n-3}x^{n-3} \)
\[ +b_{n-2}x^{n-2}. \] In order to construct a cyclic code, one needs to factor \( x^{n-1} \). Each factor
of \( x^{n-1} \) can be thought of as a generating polynomial, \( g(x) \). This is useful
because the generating polynomial represents shift stages with connections feedback from certain taps to generate the cyclic codes. As an example, consider binary cyclic codes for \( n=10 \). If \( g(x) \) is defined as \((x^3+x^{10})\) and \((x^2+x^3+x^6+x^8+x^9+x^{10})\) which are the codes used in the Global Position System (GPS), then the shift registers with feedback connections for these two generation vectors are shown in Figures 2.1 and 2.2, respectively. If the processing is continued, the output of the shift register will repeat same sequence with length of \( 2^n-1 \) that it has generated from initial entries. For instance, let the initial entries in Figure 2.1 be...
Figure 2.3: Circular auto-correlation of a BMLS of length 1023

The output vector is then

\[
\begin{array}{cccccccc}
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1
\end{array}
\]

For the same initial entries, different cyclic codes are generated from different generating polynomials. The circular auto-correlation of these cyclic codes follows (2.1) and is illustrated in Figure 2.3.

A spread spectrum communication system uses more than one sequence. Thus, using all possible feedback taps for \( m \) stages, the number of \( m \) sequences of given length can be shown to be
\[ N = \frac{1}{m} \phi(2^m - 1) \] (2.2)

Where \( \phi(k) \) is the number of positive integers less than \( k \) and relatively prime to \( k \). This is frequently referred to as the Euler function.

Table 2.1 shows some examples of the number of possible sequences that can be generated from shift registers. If the argument of the Euler function is prime, all of the integers less than the prime number are prime relative to it, and the number of sequences is much greater [11].

<table>
<thead>
<tr>
<th>Register Stages ( m )</th>
<th>Sequence Length ( M = 2^m - 1 )</th>
<th>Total Number of Sequence ( N = \frac{1}{m} \phi(2^m - 1) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>15</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>31</td>
<td>6</td>
</tr>
<tr>
<td>8</td>
<td>255</td>
<td>16</td>
</tr>
<tr>
<td>10</td>
<td>1023</td>
<td>60</td>
</tr>
<tr>
<td>12</td>
<td>4095</td>
<td>144</td>
</tr>
<tr>
<td>15</td>
<td>16383</td>
<td>1800</td>
</tr>
<tr>
<td>16</td>
<td>65535</td>
<td>2048</td>
</tr>
</tbody>
</table>

**Table 2.1:** Example of the number of sequence for sequence of different lengths
2.2.2 Improvements for the CCSK System

In the section above, it was shown that the length of BMLS is \( M = 2^m - 1 \) which is not a power of 2. This implies that the sequence in CCSK can not be applied to FFT operations to take advantage of the transform domain. Thus, one component must be to added BMLS to make it suitable for transform domain processing. However, this destroys the perfect circular auto-correlation property of BMLS. The circular auto-correlation for cyclic codes with length of 1024 is illustrated in the Figure 2.4.

Although a perfect auto-correlation property does not exist, the correlation peak is the same. This means that it can still be applied to the CCSK/TD system to take advantage of transform processing. In addition, as seen in Table 2.1, whether or not the vector length of power is 2, the capacity is not sufficient to permit large numbers of multiusers. According to [34] and [35], if base vectors are generated by a random uniform process method, the channel capacity will be maximized. The circular auto-correlation is illustrated in Figure 2.5. The characteristics shown in Figures 2.4 and 2.5 are almost same.

2.3 Implementation of a CCSK System

A CCSK transmitter and receiver are shown in Figures 2.6 and 2.7. The cyclic shifted version of base vector \( s_0 \) are taken to form an M-ary signals set \( \{s_t\} \) to represent the transmitted data. This is the so-called CCSK modulation.
**Figure 2.4:** Circular auto-correlation of a BMLS of length 1024

**Figure 2.5:** Circular auto-correlation of a random uniform signal of length 1024
Figure 2.6: CCSK Transmitter

Figure 2.7: CCSK Receiver
Let $s_m$ be the $m$'th circulated shift version of base vector $s_0$. Then

$$s_m(n) = s_0(n - m) \quad n, m = 0, 1, \ldots, M - 1 \quad (2.3)$$

The transmitted $s_m$ with length of $M$ represents the true data $m$ in decimal form. This decimal value of $m$ also represents the times of circular shift of base vector $s_0$.

As an example, assume

- Base vector $s_0 = 00100000$ (M=8)
- Transmitted data $m = (100)_2 = (4)_{10}$

then, the transmitted data $s_m = 00000010$ represents message 4.

For CCSK demodulation, the received signal is match filtered at baseband. Since the set $\{s_i\}$ is all possible circular shifts of $s_0$, the symbol decision for the decision variable $U_i$ is the maximum real value of these dot products or the highest real value peak in the circular cross-correlation.

### 2.3.1 Circular Auto Correlation of a CCSK System

If a base vector is chosen such that all of its M-ary signals are orthogonal, the performance of the CCSK waveform will be maximized. That is

$$\sum s_i(n)s_j^*(n) = \begin{cases} E_s, & i = j \\ 0, & \text{otherwise} \end{cases} \quad (2.4)$$

Where $E_s$ is the energy in each symbol.

The circular auto-correlation is simply its orthogonality. That is,
2.3.2 Processing Gain

If all M-ary signals are used, the spread spectrum processing gain, PG, is found to be

\[
R_s(\tau) = \sum_{n=0}^{M-1} s_0(n)s_0(n+\tau)
\]

\[
= \begin{cases} 
E_s, & \tau = 0 \\
0, & \tau = 1, 2, \ldots, M-1 
\end{cases}
\]  \hspace{1cm} (2.5)

\[
PG = \frac{W_s}{R_b} = \frac{MT_c}{bT_c} = \frac{M}{b} = \frac{M}{\log_2 M}
\]  \hspace{1cm} (2.6)

Where \( R_b \) is the data rate in bits per symbol duration of the system and \( T_c \) is the chip rate.

The processing gain will increase greatly with little increase in data bit \( b \).

2.4 CCSK/TD System Implemented with Transmitted Reference

The CCSK/TD system is different from conventional systems in that the CCSK demodulation is implemented in the TD. This modification is shown in Figures 2.8 and 2.9. In the transform domain, signal processing can reduce the complexity of the digital implementation of the receiver by multiplication of one TD vector by the complex conjugate of the other TD vector using FFT and IFFT. For large M-ary CCSK signals, this reduction in the computational load of the
demodulation operation makes it possible to do real time digital demodulation. For the circular auto-correlation of \( s(n) \), the M-pint DFT of \( s(n) \), \( S(k) \) is computed

\[
s(k)_I^2, \quad \text{corresponds to the circular convolution of the finite-length sequence } s(n) \text{ with } s(-n).
\]

That is,

- **Figure 2.8:** CCSK/TD Transmitter with transmitted reference signal

- **Figure 2.9:** CCSK/TD Receiver with transmitted reference signal
\[ |S(k)|^2 = S(k)S^*(k) \quad k = 0, 1, \ldots, M - 1 \quad (2.7) \]

In the transmitter of CCSK/TD system, a reference base-vector, \( s_r \), is added to the modulated CCSK base vector, \( s_m \), at the transmitter. The vector \( s_r \) is also a CCSK base vector, but it is not modulated. The CCSK/TD demodulation operations are each cross-correlation operations of a received signal vector with a stored reference base vector. Following (2.6), the reference vector \( s_r \) should be a base vector with

\[
R_{s_r s_r}(\tau) = \begin{cases} E_s, & \tau = 0 \\ 0, & \tau = 1, 2, \ldots, M - 1 \end{cases} \quad (2.8)
\]

Furthermore, to require that the channel have the same effect on \( s_0 \) and \( s_r \), and thus that the receiver can use the received reference signal (after the reference CCSK demodulation), the magnitude spectrum of \( S_r \) is approximately equal to that of \( S_0 \) but they are uncorrelated. That is

\[
|S_0(k)| \cong |S_r(k)| \quad k = 0, 1, \ldots, M - 1 \quad (2.9)
\]

and

\[
R_{S_0 S_r}(\tau) = \sum_{n=0}^{M-1} S_0^*(n)S_r(n+\tau) \cdot M 
\]

\[
< E_s \quad (2.10)
\]

(2.10) implies that two base vectors are code-separated and that they can be easily separated at the receiver. If \( s_0 \) and \( s_r \) are chosen appropriately, the transmitted
reference signal can be used to produce an estimate of the instantaneous transfer function of the channel [12]. Thus, the receiver can operate efficiently on a time varying channel without tracking the channel with past history.

2.5 Multicarrier Modulation

As described in section 2.1, CCSK waveforms are required to be orthogonal. If these orthogonal signals are subcarriers with properly spaced center frequencies, this signaling technique is referred to as multicarrier modulation (MCM) [13]. In order to implement the MCM in a CCSK/TD system, required modifications for the CCSK/TD system shown in Figures 2.8 and 2.9 are illustrated in Figures 2.10 and 2.11.

MCM is a form of frequency division modulation (FDM) [13]. Like Frequency Domain Interleaving (FDI), modulation is performed on M bits of data at a time. The transmitter interleaves these symbols in the frequency domain by

![Diagram](image)

**Figure 2.10:** MCM transmitter
calculating an M-point FFT. After that, the transformed signal is interleaved, then an M-point IFFT is computed to transform the signal back to the time domain to transmit. In the receiver, the signal is demodulated by assembling samples into a block and performing on FFT. This is equivalent to demodulating each subband separately. With MCM, the data bandwidth is divided into uniform bands [13]. This allows variable coding and equalization strategies in which bands are tailored to the channel distortions within the particular band. This modulation yields superior performance on distorted channels as compared to single carrier signal systems. However, this FFT and IFFF operation will be replaced by discrete wavelet transforms, as discussed in the following chapter.

**Figure 2.11:** MCM receiver
Chapter 3

Multiresolution, Filter Banks and Wavelets

3.1 Multiresolution Analysis

There are several ways to generate wavelets. One of these is through multiresolution analysis (MRA). As in subband decomposition, the concept of MRA is to divide a signal into components of different scales (frequency bands) of $2^m$ where $m$ is an integer, and then treat those subspectra individually. The decomposition of the signal spectrum into subbands provides the mathematical basis for an important and desirable feature in signal analysis and processing [23]: The monitoring of signal energy components within the subbands. Suppose that MRA is applied to a signal which is not bandlimited but has most of its energy in the low frequency region. The subband signals, therefore, can be ranked and processed independently.
In addition, this multiresolution signal decomposition connects to the subband decomposition through multirate signal processing called subband decomposition, implemented with the quadrature mirror filter bank [20]. In this technique, the average number of bits per sample is reduced, though the average number of samples per unit time is unchanged.

3.2 Quadrature Mirror Filter Banks

The basic elements of multirate techniques include a decimator, which is a device that reduces the sampling rate by an integer factor of $M$, and an expander, which is used to increase the rate by $L$ [20]. The operations of $M$ reduction and $L$ expansion are called decimation and interpolation, respectively. The decimation and interpolation of signals are dual processes. In other words, a digital system with a decimator can be transformed into a dual digital system implemented with an interpolator using transformation techniques. Also, both decimation and interpolation can be formulated in terms of linear filter operations. The use of multiple sampling rates offers many advantages, such as reduced computational complexity for a given task, reduced transmission rate, and/or reduced storage requirements, depending on the application. One application of multirate processing is the Quadrature Mirror Filter (QMF). In this thesis, the two-channel QMF is considered and implemented as shown in Figure 3.1. Unless otherwise stated, QMF will refer to two-channel QMF.
The QMF consists of an analysis filter bank followed by a synthesis filter bank to form a subband coding filter bank. A decimated analysis filter bank with filters $H_0(z)$, $H_1(z)$ and decimation by 2 decomposes the input signal $x(n)$ into the subband signal $x_0(n)$ and $x_1(n)$. The filtered signals named subband signals are thus approximately bandlimited (lowpass and highpass, respectively). A synthesis filter is implemented with filters $F_1(z)$, $F_2(z)$ and interpolation by 2. In this manner, an approximation $\hat{x}(n)$ of the signal $x(n)$ is reconstructed from the subband signals. In QMF, the filters $H_0(z)$, $H_1(z)$ in the analysis filter bank must be lowpass and highpass filters with the transfer functions illustrated in figure 3.2 and 3.3. This is the same for $F_1(z)$ and $F_2(z)$ in a synthesis filter bank.

Since decimation is used, the signals alias after filtering and they cannot be reconstructed exactly. To arrive at a perfect reconstruction, many methods have been developed [24] - [28]. The filter design used in this thesis is based on that developed by Vaidyanathan [20] and Jones [36].

**Figure 3.1:** Two-channel quadrature mirror filter

![Diagram of a two-channel quadrature mirror filter](image-url)
Figure 3.2: Frequency Response of Quadrature Mirror Filter
(a) Lowpass Filter and (b) Highpass Filter
3.2.1 The Odd Order Filters

According to Vaidyanathan [20], if filter coefficients are selected carefully, perfect reconstruction can be obtained. The proof for this was demonstrated in [20], and is summarized below.

Let \( H_0(z) \) be a finite impulse response (FIR) filter with the length of \( N \); \( N \) is odd and \( H_0(z) \) and \( H_1(z) \) are lowpass and high pass filters, respectively. With \( h_0(n) \) being real, \( H_0(z) \), the Z transform of \( h_0(n) \), is

\[
H_0(z) = \sum_{n=0}^{N} h_0(n)z^{-n} \quad \text{N is odd} \tag{3.1}
\]

The linear phase constraint in lowpass filter requires

\[
h_0(n) = h_0(N-n) \tag{3.2}
\]

Moreover, let \( H_0(z) \) satisfies the power complementary property,

\[
H_0(z)H_0^*(-z) + H_0(-z)H_0^*(-z) = 2 \tag{3.3}
\]

Where \( H^*(z) \) is the complex conjugate of \( H(z) \) in which the coefficients \( h \) is conjugated and the \( z \) is replaced by \( z^{-1} \).

The decimated signal \( V_k(z) \) in figure 3.1 is

\[
V_k(z) = \frac{1}{2}[X_k(z^{1/2}) + X_k(-z^{1/2})] \quad k = 0,1 \tag{3.4}
\]

and the interpolated signal \( Y_k(z) \) is

\[
Y_k(z) = V_k(z^2) = \frac{1}{2}[X_k(z) + X_k(-z)] \tag{3.5}
\]

\[
= \frac{1}{2}[H_k(z)X(z) + H_k(-z)X(-z)]
\]
The reconstructed signal then is

$$
\hat{X}(z) = F_0(z)Y_0(z) + F_1(z)Y_1(z)
$$

$$
= \frac{1}{2} \left[ H_0(z)F_0(z) + H_1(z)F_1(z) \right] X(z) + \frac{1}{2} \left[ H_0(-z)F_0(z) + H_1(-z)F_1(z) \right] X(-z)
$$

(3.6)

The first term in (3.6) is referred to as the distortion function $T(z)$, and the second term as aliasing. To cancel these two terms, the highpass filter $H_1(z)$ is chosen as

$$
H_1(z) = -z^{-N} H_0^*(-z)
$$

(3.7)

and the analysis filters are related as

$$
F_0(z) = H_1(-z)
$$

(3.8)

$$
F_1(z) = -H_0(-z)
$$

(3.9)

Thus all four filters are completely determined by a single filter, $H_0(z)$. The choices of the filters can be rewritten in the time domain as

$$
h_1(n) = (-1)^n h_0^*(N - n)
$$

(3.10)

$$
f_0(n) = h_0^*(N - n)
$$

(3.11)

$$
f_1(n) = h_1^*(N - n)
$$

(3.12)

The reconstructed signal is then

$$
\hat{X}(z) = T(z)X(z)
$$

(3.13)

Where the $T(z)$ is the distortion function with the form

$$
T(z) = cz^{-\kappa}
$$

(3.14)

The signal in the time domain is thus
Figure 3.3: Daubchies filter with order 35
(a) Lowpass filter and
(b) Highpass filter
Therefore, a QMF bank is free from aliasing as well as distortion, and is said to have the perfect reconstruction property with a scale factor $c$ and a constant delay $n_0$. As an example, Daubchies filters with order 35 are shown in Figure 3.3.

### 3.3.2 Even Order Filter

In the example above, the filter is restricted to odd order. If it is of even order, the relationship of the filters chosen in (3.7), (3.8) and (3.9) will cause much amplitude distortion in the reconstructed signal. To avoid this, the relationship between each filter and $H_0(z)$ should be modified. This modification, described in [36], will be briefly reviewed here.

Like the odd order filter, assume that $H_0(z)$ satisfies the power complementary property (3.3). The relationship between filter $H_0(z)$ and the others have been modified as below:

\begin{align}
H_1(z) &= z^{-N} H_0^*(-z) \\
F_0(z) &= H_1(-z) \\
F_1(z) &= H_0(-z)
\end{align}

and the reconstructed signal is

\begin{align}
\hat{X}(z) &= z^{-(N+1)}X(z) \\
\hat{x}(n) &= x(n-N-1)
\end{align}
Figure 3.4: Square root raised cosine filter with order 36
(a) Lowpass filter
(b) Highpass filter
Again, the transfer function is

\[ T(z) = z^{-(N+1)} \]  

(3.21)

Thus, the signal is perfectly reconstructed. Figure 3.4 shows the impulse response of lowpass and highpass filters coming from the square root raised cosine filter developed by Jones [36].

The even order filter, a square root raised cosine, developed by Jones is symmetrical and a truncated version. The property of symmetry enables higher speed operation in digital filter banks by reducing the multiplications. However, a filter with truncated coefficients can not reconstruct the signal exactly, though it can do so with some precision. In contrast this, the Daubchies odd order filter, being not symmetrical has the exactly reconstruction property. For the balance of the discussion, these two filters will be applied through the entire thesis.

3.3 Wavelet Analysis

Wavelet analysis has received much attention since the middle eighties [20]. A great step toward practical wavelet analysis was the introduction of the concept of multiresolution. With multiresolution, wavelets are closely related to tree structured filter banks; the non-uniform tree structured filter bank is related to the dyadic wavelet. Based on this, several kinds of wavelets are developed with different tree structures.
3.3.1 Essential Results from MRA

The following essential MRA result repeats from [36] and [37], and follows the notation of [18]. An MRA is a sequence of closed subspaces \( \{V_m \mid m \in \mathbb{Z} \} \) of \( L^2(\mathbb{R}) \), denoted \( V_2^j \), satisfying the following:

1. **Containment:** \( V_2^j \subset V_2^{j+1}, \quad \forall j \in \mathbb{Z} \)

2. **Scaling Property:** \( f(x) \in V_2^j \iff f(2x) \in V_2^{j+1}, \quad \forall j \in \mathbb{Z} \)

3. **Completeness:** \( \bigcup_{j \in \mathbb{Z}} V_2^j \) is dense in \( L^2(\mathbb{R}) \)

4. **Basis/Frame Property:**

\[ \exists \ g \in V_1 \ni g(x - n), \forall n \in \mathbb{Z}, \text{ is a Riesz basis for } V_1 \]

Where

1. \( \mathbb{Z} \) is the set of integers
2. \( \mathbb{R} \) is the set of real numbers
3. \( L^2(\mathbb{R}) \) means that the space of all square-integrable functions having domain \( \mathbb{R} \).

**Theorem 3.1**

Let \( \left( V_2^j \right)_{j \in \mathbb{Z}} \) be an MRA, then \( \exists ! \phi \in V_1 \ni \sqrt{2^j} \phi(2^j x - n), \forall n \in \mathbb{Z}, \) is an orthonormal basis for \( V_2^j \).
The scaling function $\phi$ is central when constructing the wavelet. One significant consequence of the theorem is that the family of scaling functions is an orthonormal basis for $V_1$. That is, the scaling function with its translate

$$\phi_n(x) = \phi(x - n) \quad n \in \mathbb{Z} \quad (3.22)$$

constitutes an orthonormal basis of the spaces $V_1$.

In addition, because of this orthogonality, the adjacent symbols are free of inter-symbol interference (ISI). ISI is defined as the interference experienced when symbols are filtered improperly; as they pass through a communication system they will spread in time, and the symbols may be smeared into adjacent time slots [38].

Following theorem 3.1, let $W_{2^j}$ be the orthogonal complement of $V_{2^j}$ in $V_{2^{j+1}}$. Then,

$$V_{2^{j+1}} = V_{2^j} \oplus W_{2^j} \quad (3.23)$$

$$V_{2^j} \perp W_{2^j} \quad (3.24)$$

In addition, for $j \geq 2$ the expansion to the direct sum of the possibly finite space $W_{2^j}$ spanning $V_1$ is

$$V_1 = V_{2^{j-1}} \oplus W_{2^{j-1}} \oplus \cdots \oplus W_{2^1} \oplus W_{2^0} \quad (3.25)$$

From (3.22) and (3.23), the following theorem is held.
**Theorem 3.2**

Let \( \left( V_j \right)_{j \in \mathbb{Z}} \) be an MRA. Then \( \exists ! \psi \in V_1, \sqrt{2^j} \psi(2^j x - n), \forall n \in \mathbb{Z}, \) is an orthonormal basis for \( V_j \). In addition, \( \sqrt{2^j} \psi(2^j x - n), \forall n, j \in \mathbb{Z}^2, \) is an orthonormal basis for \( L^2(\mathbb{R}) \).

The wavelet function \( \psi \) is also important when implementing the wavelet.

For subspace \( V_2 \), both the scaling and wavelet functions can be considered as a series expansion in the basis function, such as

\[
\phi(x) = \sqrt{2} \sum_{n \in \mathbb{Z}} h_n \phi(2n - n) \\
\psi(x) = \sqrt{2} \sum_{n \in \mathbb{Z}} g_n \phi(2n - n) \\
g_n = (-1)^n h_{-n}
\]

Therefore, one can see that the wavelet function \( \psi \) is actually determined by the scaling function for the MRA.

**3.3.2 Relationship Between Wavelets and Filter Banks**

There is a close relationship between wavelets and perfect reconstruction filter banks. That is, the wavelets are implemented as a form of filter bank [21]-[29]. In Jones' work, wavelets are linked together with filter banks. However, this conceptionalization is unclear and needs further proof. In this section, it will be investigated through mathematical expression based on [27].
To generate an orthonormal basis for all spaces $V_{2^j}$, one can use the scaling property and the property of the scaling function (3.23). That is,

$$\phi_{j,k}(x) = 2^{j/2} \phi(2^j x - k), \quad j,k \in \mathbb{Z} \tag{3.29}$$

Using the introduced bases, a signal $s_p(x) \in V_{2^j}$ and $s_w(x) \in W_{2^j}$ can be expressed as

$$s_p(x) = \sum_m \alpha_j(m) \cdot \phi_{j,m}(x) \tag{3.30}$$

and

$$s_w(x) = \sum_m \beta_j(m) \cdot \psi_{j,m}(x) \tag{3.31}$$

The expansion $\alpha_j(m)$ is thus an inner product of $s_p(x)$ and $\phi_{j,m}(x)$. In addition, because $s_p(x)$ is also an element of the space $V_{2^j}$, it can be expanded as

$$s_p(x) = \sum_n \alpha_{j+1}(n) \cdot \phi_{j+1,n}(x) \tag{3.32}$$

In order to compute the expansion coefficients for the projection of a signal for the space $V_{2^{j+1}}$ into the subspace $V_{2^j}$, assume signal $s(x)$ with known coefficient $\alpha_{j+1}(n)$ to be in the space $V_{2^{j+1}}$. Then (3.32) implies

$$s(x) = \sum_n \alpha_{j+1}(n) \cdot \phi_{j+1,n}(x) \in V_{2^{j+1}} \tag{3.33}$$

Because of the direct sum property, $V_{2^{j+1}} = V_{2^j} \oplus W_{2^j}$, the signal $s(x)$ can then be represented uniquely as the sum of its projection into the subspaces, whereby the projections are expanded in terms of the respective bases. That is,

$$s(x) = \sum_m \alpha_j(m) \cdot \phi_{j,m}(x) + \sum \beta_j(m) \cdot \psi_{j,m}(x) \tag{3.34}$$
The projections are obtained by calculating the unknown coefficient \( \alpha_j(m) \) and \( \beta_j(m) \) from the known coefficient \( \alpha_{j+1}(n) \). From (3.27) and (3.29),

\[
\phi_{j,m}(x) = 2^{j/2} \phi(2^j x - m) = 2^{j/2} \sum_p \hat{h}_0(p) \phi(2^{j+1} x - 2m - p)
\]  

(3.35)

Let \( 2m + p = n \) and \( \hat{h}_0(p) = 2^{-j} h_0(p) \). Then

\[
\phi_{j,m}(x) = \sum_n h_0(n - 2m) \cdot 2^{-j} \phi(2^{j+1} x - n) = \sum_n h_0(n - 2m) \phi_{j+1,n}(x)
\]  

(3.36)

Here \( h_0(\cdot) \) is related to the lowpass filter in analysis filter banks of QMF. Similarly, the wavelet basis \( \psi_{j,m}(x) \) can be represented with respect to the space \( W_2^n \), as

\[
\psi_{j,m}(x) = \sum_n g(n - 2m) \cdot \phi_{j+1,n}(x)
\]  

(3.37)

Thus, from (3.30), the inner products \( \alpha_j(m) \) and \( \beta_j(m) \) are

\[
\alpha_j(m) = \langle s(x), \phi_{j,m}(x) \rangle = \sum_n h_0(n - 2m) \cdot \langle s(x), \phi_{j+1,n}(x) \rangle = \sum_n h_0(n - 2m) \cdot \alpha_{j+1}(n)
\]  

(3.38)

and

\[
\beta_j(m) = h_0(-n) \ast \alpha_{j+1}(n)|_{n=2m}
\]  

(3.39)

Here \( h_0(\cdot) \) is related to the highpass filter in analysis filter banks of QMF. Thus, (3.38) and (3.39) imply that the expansion coefficients \( \alpha_j(m) \) and \( \beta_j(m) \) can be
found by convolving the $\alpha_{j+1}(m)$ with $h_0(-n)$ and $h_1(-n)$, respectively, followed by a decimation of 2. In considering the connection with the nested subspaces, the projection of $V_{2^{j+1}}$ into the subspace $V_{2^j}$ corresponds to lowpass filtering, and the projection into $W_{2^j}$ corresponds just to highpass filtering. In other words, $h_0(-n)$ is exactly a lowpass filter and $h_1(-n)$ is the associated complementary highpass filter. This is illustrated in Figure 3.5.

For the conversion relationship, it is useful to represent the basis $\phi_{j+1,n}(x)$ in terms of the bases $\phi_{j,m}(x)$ and $\psi_{j,m}(x)$:

$$
\phi_{j+1,n}(x) = 2^{-j} \phi(2^j x - n) = \sum_k 2^{\frac{j}{2}} \hat{f}_0(k) \cdot 2^{\frac{j}{2}} \phi(2^j x - \frac{n}{2} - k) + \sum_k 2^{\frac{j}{2}} \hat{f}_1(k) \cdot 2^{\frac{j}{2}} \psi(2^j x - \frac{n}{2} - k)
$$

Here $\hat{f}_0(\cdot)$ and $\hat{f}_1(\cdot)$ are related to the lowpass and highpass filters in synthesis filter banks of QMF. With substitution from (3.31), (3.33) and (3.34), the unknown coefficient $\alpha_{j+1}(m)$ is
Like subspace decomposition, this merging operation from the subspaces $V_{2^j}$ and $W_{2^j}$ can be realized by the synthesis filter banks illustrated in Figure 3.6.

\[
\alpha_{j+1}(n) = \sum_n f_0(n-2m) \cdot \alpha_j(m) + \sum_m f_1(n-2m) \cdot \beta_j(m)
\]  

(3.41)

Thus, the calculation of the coefficients for the decomposition of a signal which is achieved by an analysis filter bank, and for the merging from the two subband signals in which is achieved by an synthesis filter bank is strongly linked to wavelets.

3.3.3 Varieties of Wavelets Catalogued

Following the essential result of MRA, various kinds of wavelets are developed. Theorems 3.1 3.2 and 3.25 are modified to construct these types of wavelets which include the Dyadic wavelet, M-Band Wavelet and Wavelet Packet...
These modifications are consequential to discussing the implementation and are briefly reviewed in the following section.

**Type 1: Dyadic wavelet with non-uniform tree structure**

The dyadic wavelet is directly derived from theorems 3.1 and 3.2. This originally developed wavelet with octave bands is of a special irregular tree structure. Only the lower half of the spectrum is split into two equal bands at each level of the tree. This means that the detail or higher half-band component of the signal at any level of the tree is decomposed no further. Hence, the frequency band is split in an octave manner. Following the structure of QMF, an irregular tree structure for depth J=3 and its frequency bands split are shown in Figure 3.7.

![Figure 3.7: Dyadic wavelet with non-uniform tree structure](image)
Type 2: Wavelet with uniform tree structure

This uniform tree structure is also called a regular binary subband tree structure, or basic wavelet packet [37]. The theorem is the same as that for the dyadic wavelet, but the decomposition of the detail component of the signal at any level of the tree continues at a higher half band; the operation is done in a manner like at lower half band. The frequency bands split in this manner have equal bandwidth. Figure 3.8 illustrates the wavelet with uniform tree structure for depth $J=3$ and split frequency bands.

![Wavelet with uniform tree structure](image)

**Figure 3.8:** Wavelet with uniform tree structure
Type 3: M-Band Wavelet

The M-band wavelet is derived from Type 2, above. Similar to the uniform tree structured wavelet, the M-Bands wavelet has the same subspectra, but with different wavelet functions. For $M \geq 2$, an M-based wavelet system is a sequence of closed scaling and wavelet subspaces of $L^2(\mathbb{R})$, denoted $V^I_M$ and $W^I_M$, $I \in \{1, \ldots, M-1\}$ satisfying the following conditions:

(1) $V^I_M = V^I_M \oplus W^I_M \oplus \cdots \oplus W^{M-1}_M$

(2) $V^I_M \perp W^I_M \perp \cdots \perp W^{M-1}_M$

(3) $\exists \phi \in V^I \exists \phi(x-n), \forall n \in \mathbb{Z}$, is an orthonormal basis for $V^I$

(4) For each $I \in \{1, \ldots, M-1\}$, $\exists \psi^I \in W^I \exists \psi^I(x-n), \forall n \in \mathbb{Z}$

is an orthonormal basis for $W^I$

**Theorem 3.3**

Let $\left( V^I_M \right)_{j \in \mathbb{Z}}$ be a sequence of scaling subspaces in an M-Band wavelet system, then $\sqrt{M^I} \phi(M^Ix-n), \forall n \in \mathbb{Z}$, is an orthonormal basis for $V^I_M$

**Theorem 3.4**

Let $\left( W^I_M \right)_{j \in \mathbb{Z}}$ be the sequence of $l$th wavelet subspaces in an M-Band wavelet system, $I = \{1, \ldots, M-1\}$. Then, $\sqrt{M^I} \psi(M^Ix-n), \forall n \in \mathbb{Z}$, is an orthonormal basis for $W^I_M$

In this type of wavelet, the wavelet function $\psi^I$ is special. One can see the wavelet function is not uniquely specified by the scaling function. In terms of the
filter bank, the relationship between each filter can be shown as (3.42), and the frequency bands split are shown in Figure 3.9.

\[ H_0'(z) = H_0(z)H_0(z^2)H_0(z^3) \]
\[ H_1'(z) = H_0(z)H_0(z^2)H_1(z^3) \]
\[ \vdots \]
\[ H_6'(z) = H_1(z)H_1(z^2)H_0(z^3) \]
\[ H_7'(z) = H_1(z)H_1(z^2)H_1(z^3) \]

Figure 3.9: M-Bands Wavelet
Type 4: Wavelet Packet

The wavelet packet can be viewed as a general approach for each type of wavelet [37]. In other words, all wavelet types except the M-band wavelet, are special cases of the wavelet packet. As illustrated in Figure 3.10, the wavelet packet is an irregular tree structure and its frequency band split depends on the signal components which are significant or more important than the others. Thus, it is more flexible and it is possible to minimize the computational complexity of the spectral operations. However, this flexibility causes difficulty and requires a special process called the tuning algorithm to optimize the structure. The process is complicated, as is its theorem, and it is not instructive for this review.

Figure 3.10: Wavelet Packet
Among these wavelet structures, this thesis will focus on the uniform tree structure. Also, the orthonormal bases with and without compactly supported wavelets constructed by Jones [36] (the Daubchies and square root raised cosine filters) will be applied.
Chapter 4

Periodic Discrete Wavelet Transform and its Fast Algorithm

In wavelet analysis, if the input is regarded as a infinitely supported sequence, the output will be a finitely supported infinite sequence called an aperiodic discrete wavelet [23]. Hence, those discrete wavelets considered in Chapter 3 are aperiodic wavelets. For many applications, such as numerical and image analysis, they are not sufficient [17]. In these cases, the wavelets are confined to an interval rather than the real line. For an orthonormal system on a finite interval, there is another kind of wavelet, called the periodic wavelet [16], [17], [23] and [30]-[31], which were studied first by Meyer [32]. In the following section, the periodic discrete wavelet and transform will be discussed. Also, the fast algorithm is developed based on the periodic discrete wavelet transform.
Furthermore, some characteristics for periodic wavelets are presented. Finally, the results from periodic wavelets are compared to those from aperiodic wavelets.

4.1 Periodic Discrete Wavelets

The periodic wavelet system is found by periodizing the scaling and wavelet function [23]. In contrast to aperiodic wavelets which are bases of $L^2(\mathbb{R})$, the bases for periodic wavelets are reconstructed in $L^2[0, 1]$. Therefore, these periodic scaling and wavelet functions are both periodic of period 1; this is similar to the FFT. On the other hand, as with aperiodic wavelets, these periodizing scaling and wavelet functions form orthogonal bases. Furthermore, similar ideas used for the construction of aperiodic wavelets succeed in the periodic case [16]. That is, the properties of periodic wavelets follow those of aperiodic wavelets. Finally, the wavelet types for periodic wavelets are the same as for aperiodic wavelets.

Returning to the wavelet theorem, consider $\psi$ and $\phi$ such that

$$\phi_{j,k}(x) = 2^{j/2}(\phi(2^j x - n))$$

$$\phi(x) = \sum_{n=0}^{N-1} h_{n+1} \phi(2x - n)$$

$$\psi(x) = \sum_{n=0}^{N-1} g_{n+1} \phi(2x - n)$$

$$g_n = (-1)^{n-1} h_{N-n+1}, \quad n = 1, 2, \ldots, N$$

Here, $h_n$ is related to $f_0(n)$ in Chapter 3 and $g_n$ is related to $f_1(n)$. 
The one stage of this signal processing operation is illustrated in Figures 4.1 and 4.2, respectively. The construction of the functions $\psi$ and $\phi$ include the properties described in Chapter 3, where it was shown that the coefficients $\{h_n\}$, $n=1, \ldots, N$ define the functions $\psi$ and $\phi$ uniquely.
4.2 Periodic Discrete Wavelet Transform

In Chapter 3, the QMF banks with the perfect reconstruction property were shown to be strongly linked to wavelet analysis. That is, discrete wavelets are essentially a subband coding system. This also holds for periodic discrete wavelets. Upon establishing the link between periodic discrete wavelets and subbands, a recursive

![Uniform tree structured wavelets with transformation depth J=4](image)

**Figure 4.3:** Uniform tree structured wavelets with transformation depth J=4
algorithm is applied for computing the periodic discrete wavelet series expansion coefficients. First, filter the expansion coefficients of an approximation at a particular resolution with digital lowpass and highpass filters. Second, decimate their outputs to produce the coefficients of the approximation and detail signals, respectively, at a coarser resolution [18]. This two step process is recursively continued by operating on the decimated filter output. The algorithm came to be known as the periodic discrete wavelet transform (periodic DWT) for finite duration signals. Similarly, the inverse periodic discrete wavelet transform (periodic IDWT) can be easily obtained by an inverse procedure; that is, interpolating the approximation and detail signals and filtering the expansion respectively with related lowpass and highpass filters and then adding them together. The terms DWT and IDWT will still be used to refer to these periodic discrete wavelet transform and its inverse in general terms. The signal processing for DWT operations is illustrated in Figure 4.3 and will be discussed in detail in the following section.

In Figure 4.3, the uniform wavelet bases are shown corresponding to a uniform two-channel analysis filter bank. This is also called the basic wavelet packet [37]. The gray shades indicate the non-uniform wavelet bases referred to as a dyadic wavelet tree, mentioned in Chapter 3. The \( j \) indicates the \( j \)’th transformation depth.
4.3 Fast Algorithm

The DWT algorithm based on the tree structure can be computed in an efficient manner using the pyramid algorithm (PA) developed by Mallat [18], [19]. Some fast algorithms are then be applied to compute an aperiodic DWT as an alternative [41]-[46]. Among these algorithms, one is called the recursive pyramid algorithm which is a reformulation of the class pyramid algorithm [42]. Although this aperiodic recursive pyramid algorithm (ARPA) causes the basis functions to have good spectrum containment, it also results in group delay when the signal is reconstructed [36]. Thus, in this thesis the recursive pyramid algorithm for periodic DWT will be introduced. In order to distinguish this algorithm from ARPA, the term periodic recursive pyramid algorithm (PRPA) is used. The basic idea is written in Fortran, as shown in [46], and rewritten in C language here. The modification is also made to apply to this thesis.

The connection between the input signal and DWT is simply the discrete circular convolution. The discrete circular convolution theorem presumes two circumstances, similar [47]. First, the convolution theorem takes the duration of the response to be the same as the period of data; they are both N. Second, it assumes that the input signal is periodic, whereas real data often either exist without repetition or else consist of one nonperiodic stretch of finite length.

In PRPA, the functions $\psi$ and $\phi$ do not appear explicitly at all; all transformations use the coefficients $h_n$ and $g_n$, which are the coefficients of lowpass
and highpass filters, respectively. Both of these two filter lengths are extended to the same length as the signal length N by padding the zero, \{h_n\}, \(n=1, \ldots, N\) and \{g_n\}, \(n=1, \ldots, N\), which are convolved with the entries of the signal vector. The signal x with length N, is a finite-duration discrete-time signal restricted to a length equal to a power of 2.

Since the DWT is the circular convolution-decimation, let \(j \leq k \leq N/2\), and \(j\) indicate the transformation depth, the approximate component and detail component is calculated as

\[
a_k^j = \sum_{n=1}^{N} h_n x_{n+2k-2} \tag{4.5}
\]

\[
d_k^j = \sum_{n=1}^{N} g_n x_{n+2k-2} \tag{4.6}
\]

and

\[
a_k^j = \sum_{n=1}^{N} h_n a_{n+2k-2}^{j-1} \tag{4.7}
\]

\[
d_k^j = \sum_{n=1}^{N} g_n a_{n+2k-2}^{j-1} \tag{4.8}
\]

Similarly, for the IDWT, the inverse process can be done as follows:

\[
a_{2n}^{j-1} = \sum_{k=1}^{\lfloor N/2 \rfloor} h_{2k} a_{n-k+1}^{j} + \sum_{k=1}^{\lfloor N/2 \rfloor} g_{2k} d_{n-k+1}^{j} \tag{4.9}
\]

\[
a_{2n-1}^{j-1} = \sum_{k=1}^{\lfloor N/2 \rfloor} h_{2k-1} a_{n-k+1}^{j} + \sum_{k=1}^{\lfloor N/2 \rfloor} g_{2k-1} d_{n-k+1}^{j} \tag{4.10}
\]
To analyze this algorithm, it is convenient to combine the lowpass filter and highpass filter into a single combined transform matrix and to write the forward recursions as matrix-vector products. Thus, from (4.7) and (4.8) one can write the forward recursion in a type of matrix-vector:

\[ X^j = W^{j-1} \cdot X^{j-1}, \quad j = 1, 2, \ldots, J \]  

(4.11)

For filters with a length of 4 and a signal with a length of 16, the signal vector and combined filter matrix can be represented as:

\[ X = \begin{bmatrix} x_1 & x_2 & \cdots & x_{15} & x_{16} \end{bmatrix} \]  

(4.12)

and

\[
W = \begin{bmatrix}
    h_1 & h_2 & h_3 & h_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
    g_1 & g_2 & g_3 & g_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
    0 & 0 & h_1 & h_2 & h_3 & h_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
    0 & 0 & g_1 & g_2 & g_3 & g_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
    \vdots & \vdots & & & & & & & & & & & & & & \\
    0 & 0 & 0 & 0 & 0 & 0 & \cdots & h_1 & h_2 & h_3 & h_4 & 0 & 0 & 0 & 0 & 0 & 0 \\
    0 & 0 & 0 & 0 & 0 & 0 & \cdots & g_1 & g_2 & g_3 & g_4 & 0 & 0 & 0 & 0 & 0 & 0 \\
    h_3 & h_4 & 0 & 0 & 0 & 0 & 0 & 0 & h_1 & h_2 & 0 & 0 & 0 & 0 & g_1 & g_2 \\
    g_3 & g_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & g_1 & g_2 & \cdots & & & \\
\end{bmatrix}
\]  

(4.13)

A more detailed description for this algorithm is shown below. First, take the full data of length \( N \) and run it through (4.13). One acquires half of the approximated coefficient and half of the detailed coefficient in crossing. Second,
permute the filtered output to make both halves of coefficients in a row. Third, modify (4.13) to make the upper left part of matrix left, and rearrange the coefficient set to a form of cyclic shift, such as that in (4.13) but with size of 1/4.

\[
\begin{bmatrix}
    x_1 & a_1 \\
    x_2 & d_1 \\
    x_3 & a_2 \\
    x_4 & d_2 \\
    x_5 & a_3 \\
    x_6 & d_3 \\
    x_7 & a_4 \\
    x_8 & d_4 \\
    x_9 & a_5 \\
    x_{10} & d_5 \\
    x_{11} & a_6 \\
    x_{12} & d_6 \\
    x_{13} & a_7 \\
    x_{14} & d_7 \\
    x_{15} & a_8 \\
    x_{16} & d_8 \\
\end{bmatrix}
\]

Figure 4.4: Diagram for recursive pyramid algorithm

of previous one. For each permuted approximated and detailed coefficient set, run the same algorithm until one approximated coefficient and detailed coefficient is left. This process is illustrated in Figure 4.4, where W means to take the wavelet transform and P is permuting. Obviously, the final transformed data set from previous data set has two left.
An algorithm for IDWT is easily deduced from that for DWT. If the wavelet forms an orthogonal basis, the exact inverse algorithm is obtained by taking the transpose of the DWT flow graph. That is, similar to the discrete wavelet transform, for the inverse discrete wavelet transform the transpose of combined filter matrix can be written as following:

\[
W^T = \begin{bmatrix}
  h_1 & g_1 & 0 & 0 & 0 & 0 & 0 & h_3 & g_3 \\
  h_2 & g_2 & 0 & 0 & 0 & 0 & 0 & h_4 & g_4 \\
  h_3 & g_3 & h_1 & g_1 & \cdots & 0 & 0 & 0 & 0 \\
  h_4 & g_4 & h_2 & g_2 & 0 & 0 & 0 & 0 & 0 \\
  & \vdots & & & & \ddots & & & \vdots \\
  0 & 0 & 0 & 0 & h_3 & g_3 & h_1 & g_1 & 0 & 0 \\
  0 & 0 & 0 & 0 & \cdots & h_4 & g_4 & h_2 & g_2 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & 0 & h_3 & g_3 & h_1 & g_1 \\
  0 & 0 & 0 & 0 & 0 & 0 & h_4 & g_4 & h_2 & g_4
\end{bmatrix}
\]  

Also, the inverse recursion is

\[
X^{j+1} = (W^{j-1})^T \cdot X^j, \quad j = 1, 2, ..., J
\]  

In the inverse procedure, this algorithm reconstructs the original signal of length \(N\) from \(N/2\) approximated components and \(N/2\) detailed components.

In order to observe the behavior of the wavelet bases, one can simply run unit vectors through the discrete wavelet transform. That takes a unit vector with various components in the input and applies it to (4.15) recursively. As an example, with the signal with length of 16, the unit vectors are arranged from the 1st through the 16th component; that is,
The wavelet bases which are the inverse DWT of the various unit vector with different components are illustrated in Figures 4.5 and 4.6. Figure 4.5 expresses the wavelet bases which use scaling and wavelet functions associated to the square root raised cosine filter with length of 37, and Figure 4.6 the Daubchies filter with length of 36.

Since the signal is intended to be periodic, like the FFT, this algorithm is a fast and linear operation on a data vector whose length is an integer power of two, transforming it into a numerically different vector of the same length. Additionally, the wavelet transform is invertable and in fact orthogonal; the inverse transform, when viewed as a big matrix, is simply the transpose of the transform. Furthermore, because the full procedure is a composition of orthogonal linear operations, the whole of DWT & IDWT are themselves orthogonal linear operators.

\[ x_1 = [1 \ 0 \ 0 \ \cdots \ 0 \ 0 \ 0]_{16}, \]
\[ x_2 = [0 \ 1 \ 0 \ \cdots \ 0 \ 0 \ 0]_{16}, \]
\[ \vdots \]
\[ x_{16} = [0 \ 0 \ 0 \ \cdots \ 0 \ 0 \ 1]_{16} \]
Figure 4.5: Basis Functions constructed from PRPA algorithm with square root raised cosine filter with length of 37
Figure 4.6: Basis Functions constructed from PRPA algorithm with Daubchies filter with length of 36.
4.3.1 Orthogonality of Wavelet Bases

Since $\phi(x)$ is the final function to which the signal converges, and is the product of lowpass filters, the final function is itself lowpass and is used to go from a fine scale to a coarser scale. In a fashion similar to $\phi(x)$, $\psi(x)$ is a bandpass filter as well as its translation except a final highpass filter. Since the filters $h(n)$ and $g(n)$ essentially form an orthonormal set with respect to even shifts, according to wavelet theorem the wavelet basis functions form an orthonormal set.

Let $B_j(n)$ be all wavelet bases. Recall that orthonormality means

$$\sum_{n=1}^{N} B_j(n)B_k^*(n) = \begin{cases} 1 & \text{if } j = k \\ 0 & \text{otherwise} \end{cases}$$

(4.16)

In addition, if a tight frame is used such that all wavelets are necessary to reconstruct a general signal, then the wavelets form an orthonormal basis of the space of signals with finite energy [60]. The reconstructed signal $\hat{x}(n)$ can be expressed as

$$\hat{x}(n) = \sum_{k=1}^{N} c_k(n)B_k(n)$$

(4.17)

Figure 4.7 illustrates the orthogonality of wavelets with two different kinds of filter coefficients, square root raised cosine and Daubchies. The upper right position in these two figures is the original plot of orthonormality, while they are enlarged in the central part. In these enlarged plots, the value in the diagonal position is zeroed out to see clearly the relationship between each basis function.
Figure 4.7a: Orthogonality with filter coefficients, square root raised cosine

Figure 4.7b: Orthogonality with filter coefficients, Daubchies
Comparing Figures 4.7a with 4.7b, one can see that the value in the diagonal position is one; the orthonormality of the remainder is about $10^{-5}$ for the square root raised cosine wavelet and as low as $10^{-16}$ for Daubchies wavelets. Therefore, the orthonormality of wavelet bases with Daubchies is better than that with the square root raised cosine. The better the orthonormality, the more perfect the reconstruction. This is because the coefficients of the square root raised cosine are truncated versions. An arbitrary signal can then be represented closely but not exactly. The reason for using the square root raised cosine filter is to take advantage of the symmetric property in time domain that Daubchies lacks.

4.3.2 Computational Cost

To increase computational efficiency and decrease memory storage, (4.11) and (4.15) are used only to show the ideas of implementing periodic DWT and its inverse. The necessary matrix-vector multiplications can be performed implicitly using only the filter vectors, without explicitly constructing the filter matrices. Independent of the depth of the tree, the complexity is linear in the number of input samples, with a constant factor that depends on the length of the filter. With DWT, in other words, the computational cost of convolving a vector with a filter is proportional to the length of the filter * length of the signal vector.
Case 1: Wavelets with Uniform Tree Structure

In the uniform tree structure, the length of the signal vector is halved at each step. Assume the computation of the first stage requires \( C_0 \) operations per input sample in which \( C_0 \) is the length of the filter, and \( M \) the length of input signal, then the total computational cost (TCC) of operations is

\[
TCC = 2C_0 \log_2 M
\]  
(4.18)

Case 2: Dyadic Tree (non-uniform) wavelets

In dyadic tree wavelets, since the convolution-decimation is only performed in the lower half band, the total computational cost is

\[
TCC = C_0 + C_1 + \ldots \leq 2C_0
\]  
(4.19)

In practical and efficient applications, \( C_0 \ll M \). Comparing to FFT algorithm which computational cost is \( M \log_2 M \) [48], the DWT will be faster than FFT if the length of input signal is long enough. The following table 4.1 shows some example of comparison between FFT and the uniform DWT with the wavelet length of 37.

<table>
<thead>
<tr>
<th></th>
<th>M=16</th>
<th>M=32</th>
<th>M=64</th>
<th>M=128</th>
<th>M=256</th>
<th>M=512</th>
<th>M=1024</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uni. DWT</td>
<td>296</td>
<td>370</td>
<td>444</td>
<td>518</td>
<td>592</td>
<td>666</td>
<td>740</td>
</tr>
<tr>
<td>FFT</td>
<td>64</td>
<td>160</td>
<td>384</td>
<td>896</td>
<td>2048</td>
<td>4608</td>
<td>10240</td>
</tr>
</tbody>
</table>

**Table 4.1:** Example of the TCC for the uniform DWT and FFT
4.4 Comparison of results from PRPA and ARPA

The periodic wavelet is one branch in the wavelet family. In contrast to the infinite sequence in aperiodic wavelets, the sequence is of finite duration in periodic wavelets. Although the processing subject of periodic wavelets is different from aperiodic wavelets, there exists some relationship between them. In the following, the wavelet basis functions and characteristics of reconstructed signal, from PRPA and ARPA, will be compared.

4.4.1 Wavelet basis functions

Compared to Jones’ results, the basis functions created from this PRPA is the cyclic shift version with one step of the periodic sum that is from ARPA. As an example, assume a signal with a length of 16. If the square root raised cosine wavelet function with odd number of coefficients is applied, the wavelet basis function created from ARPA is shown in Figure 8.

If each basis function is divided into several parts each with length of 16 and which are then summed together, this is referred as the periodic sum. The approaching results for the periodic sum from the first 16 through 26 sections are shown in the Figures 4.9 through 4.14, and the total periodic sum is shown in Figure 4.15.
Figure 4.8: Basis functions constructed by ARPA with square root raised cosine
Figure 4.9: The periodic sum from the first 16 sections
Figure 4.10: The periodic sum from the first 18 sections
Figure 4.11: The periodic sum from the first 20 sections
Figure 4.12: The periodic sum from the first 22 sections
Figure 4.13: The periodic sum from the first 24 sections
Figure 4.14: The periodic sum from the first 26 sections
Figure 4.15: The periodic sum from all divided sections from PRPA with square root raised cosine filter with length of 37
Figure 4.16: Basis Functions constructed from PRPA with Daubchies filter with length of 36
Figure 4.17: The periodic sum from all divided sections from PRPA with Daubchies filter with length of 36
Comparing the wave amplitudes and positions in Figures 4.5 and 4.15, the basis functions are exactly the same except by one position of cyclic shift. This situation holds for the scaling and wavelet functions associated with Daubechies with an even number of coefficients except one position of cyclic shift. This result is seen by comparing wavelet bases in Figure 4.6 and Figure 4.17.

4.4.2 Reconstructed Signal

Recall the theorem of the quadrature mirror filter in Chapter 3. This filter bank structure, achieving the perfect reconstruction, will cause one unit delay for each filter. If WDT and inverse DWT implement the ARPA, the reconstructed signal will suffer group delay naturally, more seriously for a dyadic wavelet transform; this non-uniform tree structure will result in a non-unit delay and must have additional processing to insert delay to compensate it. Fortunately, it does not occur for PRPA.

Let an input signal with a length of 128 be a random sequence of positive 1 or negative 1, as shown in Figure 4.18 (a). This signal is run into the DWT implemented with even order filters 36, square root raised cosine, with a length of 37. The reconstructed signal from ARPA and PRPA is illustrated in Figure 4.18, part (b) and (c) receptively. The results in Figure 4.18 (b) show the reconstructed signal suffers a constant delay of 37 samples that comprise the (3.22) in which constant delay is order of filter plus 1. On the other hand, Figure 4.18 (c) shows
Figure 4.18: Comparison of the Output for different algorithm
(a) Input signal
(b) Reconstructed signal from ARPA and
(c) Reconstructed signal from PRPA
no delay in the reconstructed signal. This implies that wavelet transform and its inverse is more likely to run the signal in real time processing.

4.5 Gray Coding

In the tree-structured uniform wavelet, because of decimation in the input signal, the analysis filter bank will scramble the bandwidth of the output signal. The adjacent bands in the spectrum are not of linear ordering. This band disordering, called band shuffling by Jones [36], results in inconvenience in recognizing the band ordering when doing the transform domain

![Figure 4.19: Subband disordering in uniform tree structure](image-url)
operation. According to [27], the lowpass and highpass bands of the output of a highpass-decimator operation switch positions relative to the original bandwidth. Let $H\downarrow 2$ be a lowpass-decimation operation and $G\downarrow 2$ a highpass-decimation operation. Assume the signal to be divided into $2^k$ subbands with equal bandwidth $B$ for each band, and $B_i$ represents the $i$'th band of original signal in linear ordering starting at 0. Then Figure 4.19 illustrates this situation of the band disordering clearly. It is obvious that scrambling only occurs at the output of the highpass filter. The band order relative to the previous signal band is switched for consecutive highpass-decimation operations and highpass-decimation followed by the lowpass-decimation operation.

One possible solution to rearrange the disordered band is gray coding. Borrowing the recursive definition of gray coding from [37], let $GC_k$ be the index vector for a gray coded sequence $[0 \ 1 \ \cdots \ 2^{k-1} \ 2^k - 1]$ and range the set of all $k$-bit binary sequences. Then for an arbitrary gray code $GC_k$,

$$GC_k = [GC_k(0) \ GC_k(1) \ \cdots \ GC_k(2^{k-1}) \ GC_k(2^k - 1)]^T$$

$$= [0G_{c,k-1}(0) \ \cdots \ 0GC_{k-1}(2^{k-1}) \ 1GC_{k-1}(2^{k-1}) \ \cdots \ 1GC_{k-1}(0)]^T$$

(4.20)

Related to gray coding, an inverse exists. That is

$$GC^{-1}(((V))_{mod \ 2} \Delta n \in GC_k(n) = ((V))_{mod \ 2}$$

(4.21)

Starting with $GC_1 = [0 \ 1]$, as an example, for $k = 2, 3$ and 4, the gray codes are shown in (4.22).
If the lowpass is represented by 0 and the highpass by 1 in the two channel filter banks, then from Figure 4.10 and gray code \( GC_3 \) in (4.22), one can see that band disordering out of DWT follows the gray coding. In order to obtain the linear ordering subbands, one simply needs to run the inverse gray coding.

\[

gc_2 = \begin{bmatrix}
00 \\
01 \\
11 \\
10
\end{bmatrix} \\
gc_3 = \begin{bmatrix}
000 \\
001 \\
011 \\
010 \\
110 \\
111 \\
101 \\
100
\end{bmatrix} \\
gc_4 = \begin{bmatrix}
0000 \\
0001 \\
0011 \\
0010 \\
0110 \\
0111 \\
0101 \\
0100 \\
1100 \\
1101 \\
1111 \\
1110 \\
1010 \\
1011 \\
1001 \\
1000
\end{bmatrix} 
\]
Chapter 5

Conclusion

This thesis is intended to implement a multicarrier modulation system using CCSK/TD and wavelet bases. Efforts by several professionals and colleagues were reviewed in parts of Chapters 2 and 3. CCSK/TD was reviewed in Chapter 2. The idea was modified and extended to a multicarrier modulation system. For the transform domain operation, the discrete Fourier transform was replaced by the discrete wavelet transform. This MCM scheme, implemented with DWT and its inverse, forms an orthogonally multiplexed communication system. It results in effective mitigation of channel distortions through frequency selective processing. In this system, the duration of the symbol is so long that pulse interference in time domain is minimized by averaging. In addition, single tone interference can be easily suppressed in the transform domain.
In Chapter 3, Multiresolution Representations, filter banks and wavelets were reviewed. The wavelet transform, which can be practically implemented because of the introduction of multiresolution, is strongly and clearly linked to the two channel filter banks based on the mathematical proof. As encountered in Chapter 3, the scaling and wavelet functions in wavelet analysis are not unique, and can be chosen based on certain kinds of applications. Several types of wavelets which can be seen as a special case of the wavelet packet were also discussed. Although the wavelet packet offers minimum cost with an unknown communication environment when decomposing a signal into subbands, it results in high complexity for the receiver to decide this flexible tree structure. Hence, it is better to use another type of wavelet to decompose signals when the environment is known.

The fast algorithm developed in Chapter 4 offers another way to calculate the expansion coefficients of uniform tree structured wavelets. Given a finite interval signal, this algorithm constructs wavelet basis functions which are the same length as the input signal. Based on this algorithm, perfect reconstruction is achieved. Furthermore, this algorithm is faster than the fast Fourier transform when the length of input signal is longer than filter length. In general applications, this high ratio of signal length to filter length is true and practical, especially in an orthogonally multiplexed communication system.
For the scaling function constructed from the square root raised cosine filter with even order, the basis functions are almost orthogonal; the orthonormalities in the diagonal position are 1 and the rest are about $10^{-5}$. It is even better for a scaling function constructed from Daubchies with 4 taps; the orthonormalities in the diagonal position are 1 and the rest are as low as $10^{-16}$. This higher orthonormality offers better reconstruction characteristics.

The orthonormal basis function with flat envelope generated from this periodic recursive pyramid algorithm is a periodic sum of that constructed from the aperiodic recursive pyramid algorithm. As shown in Chapter 4, the orthogonal bases of the latter are good in spectrum containment. That is, the different basis function with different frequency response decays very rapidly. Nevertheless, it is highly redundant; the length of basis functions is long, and most of the positions in the basis functions are zero. This results in inefficiency when they are applied to CCSK/TD scheme implemented as multicarrier modulation.

In contrast, although the wavelet basis functions with flat envelope are not as good as that in frequency containment, they have no redundancy. This offers better efficiency when data are modulated in multicarrier form. When certain data are transmitted, one need not wait for a long time to transmit the next symbol. However, the almost flat envelope property sometimes implies that it is not suitable in some kinds of applications. Because the wavelet bases generated from ARPA and PRPA have different properties, one needs to trade off in this situation to make
algorithms and applications suitable. It is also possible to modify the periodic recursive pyramid algorithm to extend the length of wavelet bases, resulting in waves which decay faster and are less redundant. These were mentioned in some papers and are left for further research [49]-[50].

The requirements of no additional process to insert delay to compensate in dyadic wavelet, and no group delay in both of dyadic and uniform tree structured wavelet, make real time processing of DWT and inverse DWT possible. In addition, this algorithm is not only used for uniform wavelets. With minor modification, it can be applied to all types of wavelets except the wavelet packet.
References


Appendix

The following are the major parts of the actual CMEX-files and M-files generating for this thesis.

```c
#include <math.h>
#include "mex.h"
#define X_INPUT  prhs[0]
#define H_FILTER prhs[1]
#define Y_OUTPUT plhs[0]

static double *h, *g, *a, *wksp, *xin, *yout, junk;
static int nn, ncof, ioff, joff;

/**
   Periodic Discrete Wavelet Transform
   Y = UFWT(x, h)

   This program is written in MATLAB-callable C language which is referred to as
   an MEX-file. The program implements the fast recursive algorithm for a periodic
   uniform tree structured Discrete Wavelet Transform based on the analysis filter
   banks. The input transformed signal, x, is the projection of DATA onto the wavelet
   basis. No matter what type of scaling and wavelet function and the filter order
   (even and odd) is used, the output, Y, is reconstructed with the length same as
   the input signal, which is N-point finite interval signal. Also, for the input
   lowpass filter coefficients, h, the program automatically determines the coefficient
   of highpass filter, g.

   #include <math.h>
   #include "mex.h"
   #define X_INPUT   prhs[0]
   #define H_FILTER prhs[1]
   #define Y_OUTPUT plhs[0]

   static double *h, *g, *a, *wksp, *xin, *yout, junk;
   static int nn, ncof, ioff, joff;

   /*******************************************************************************
   *                                                                              *
   * Periodic Discrete Wavelet Transform                                          *
   * Y = UFWT(x, h)                                                              *
   *                                                                              *
   * This program is written in MATLAB-callable C language which is referred to   *
   * as an MEX-file. The program implements the fast recursive algorithm for a     *
   * periodic uniform tree structured Discrete Wavelet Transform based on the     *
   * analysis filter banks. The input transformed signal, x, is the projection of  *
   * DATA onto the wavelet basis. No matter what type of scaling and wavelet       *
   * function and the filter order (even and odd) is used, the output, Y, is      *
   * reconstructed with the length same as the input signal, which is N-point     *
   * finite interval signal. Also, for the input lowpass filter coefficients, h,   *
   * the program automatically determines the coefficient of highpass filter, g.  *
   *******************************************************************************/

void s2p(buff,n)
  double buff[];
```

int n;
{
    int nmod, nl, nh;
    int j, ii, i, ni, nj, jf, jr, k;

    nmod = ncof * n;
    nl = n - 1;
    nh = n >> 1;

    for (j = 0; j <= n - 1; j++)
        wksp[j] = 0.0;

    for (ii = 1, i = 1; i <= n; i += 2, ii++)
        {
            ni = i + nmod + ioff;
            nj = i + nmod + joff;

            for (k = 1; k <= ncof; k++)
                {
                    jf = nl & (ni + k);
                    jr = nl & (nj + k);
                    wksp[ii - 1] += h[k - 1] * buff[jf];
                    wksp[ii + nh - 1] += g[k - 1] * buff[jr];
                }
        }

    for (j = 0; j <= n - 1; j++)
        buff[j] = wksp[j];

    n = n >> 1;
    if (n < 2)
        {
            n = n << 1;
            return;
        }
    s2p(buff, n);
    s2p(&buff[n], n);
}

/***************************************************************************/
/*
 *
 * Main
 *
 */

mexFunction(nlhs, plhs, nrhs, prhs)

int nlhs;
Matrix *plhs[];
int nrhs;
Matrix *prhs[];
{
    int j, sig;

    h = mxGetPr(H_FILLET);
    xin = mxGetPr(X_INPUT);
    ncof = mxGetN(H_FILLET);
    g = mxAlloc(ncof, sizeof(junk));
    nn = mxGetN(X_INPUT);
wksp = mxMalloc(nn, sizeof(junk));
a = mxMalloc(nn, sizeof(junk));
Y_OUTPUT = mxCreateFull(1, nn, 0);
yout = mxGetPr(Y_OUTPUT);
sig = (2 * (ncof % 2)) - 1;

for (j = 0; j <= ncof - 1; j++)
    
    g[ncof - j - 1] = sig * h[j];
    sig = -sig;

for (j = 0; j <= nn - 1; j++)
    a[j] = xin[j];

ioff = -2;
joff = -ncof + 2;
s2p(a, nn);

for (j = 0; j <= nn - 1; j++)
    yout[j] = a[j];

mxFree(g);
mxFree(a);
mxFree(wksp);
Periodic Inverse Discrete Wavelet Transform
with uniform tree structure

Y=UIFWT(x,h)

This program is written in MATLAB-callable C language which is
referred to as an MEX-file. The program implements the fast recursive
algorithm for periodic inverse Discrete Wavelet Transform with
uniform tree structure based on the synthesis filter banks. The
N-point finite interval, input signal is transformed into the wavelet
basis with the same length as the input signal no matter what type of
scaling and wavelet function and the filter order (even and odd) is
used. Also for the input lowpass filter coefficient, h, the program
automatically determines the coefficient of the highpass filter, g.

#include <math.h>
#include "mex.h"
define X_INPUT    prhs[0]
define H_FILTER   prhs[1]
define Y_OUTPUT   plhs[0]

static double *h, *g, *a, *wksp, *xin, *yout, junk;
static int nn, ncof, ioff, joff;

void p2s(buff,n)
double buff[];
int n;
{
    double ai, ail;
    int nmod, nl, nh;
    int c, p, j, ii, i, ni, nj, jf, jr, k;
    for (p=2; p<=n; p<<=1)
    {
        nmod=ncof*p;
        nl=p-1;
        nh=p>>1;
        for (j=0; j<=n-1; j++)
            wksp[j]=0.0;
        for (c=0; c<n; c+=p)
        {
            for (ii=1, i=1; i<=p; i+=2, ii++)
            {
                ai=buf[c+ii-1];
                ail=buf[c+ii+nh-1];
                ni=i+nmod+ioff;
                nj=i+nmod+joff;
                for (k=1; k<=ncof; k++)
                {
                    jf=nl & (ni+k);
                }
            }
        }
    }
}
jr=nl & (nj+k);
wksp[jf+c] += h[k-1]*ai;
wksp[jr+c] += g[k-1]*ail;
}
}
for (j=0; j<=n-1; j++)
buff[j]=wksp[j];

/**
 * Main
 */

mexFunction(nlhs, plhs, nrhs, prhs)

int nlhs;
Matrix *plhs[];
int nrhs;
Matrix *prhs[];
{
    int j, sig;
    h = mxGetPr(H_FILTER);
xin = mxGetPr(X_INPUT);
ncof = mxGetN(H_FILTER);
g = mxMalloc(ncof,sizeof(junk));
nn = mxGetN(X_INPUT);
wksp=m_malloc(nn,sizeof(junk));
a=m_malloc(nn,sizeof(junk));
Y_OUTPUT = mxCreateFull(1,nn,0);
yout=mxGetPr(Y_OUTPUT);
sig=(2*(ncof-1));
for (j=0; j<=ncof-1; j++)
{
    g[ncof-j-1]=sig*h[j];
sig=-sig;
}
for (j=0; j<=nn-1; j++)
a[j]=xin[j];

ioff=-2;
jooff=-ncof+2;
p2s(a,nn);
for (j=0; j<=nn-1; j++)
yout[j]=a[j];
m_free(g);
m_free(a);
m_free(wksp);
}
function y=uwpbase(lx,h,g)

% Construction of Basis Function
% This program is written in MATLAB. For the N-point input unit vector Y
% the output is an N by N matrix which presents 16 wavelet bases with length
% of 16 each.
% lx : The length of input signal
% h & g: The coefficients of lowpass and highpass filters

ch_max=lx;

lf=length(h);
tmax=lx-1;
tmin=0;
tinc=1;

xax=tmin:tinc:tmax;
inx=graycode(lx);

figure (1)
clf
hold on

% Generation of Scaling function
X=zeros(1,lx);
X(1,inx(1))=1;
x=uifwt(X,h);
y(1,:)=x;

subplot(ch_max,1,1), plot(xax,x)
set(gca,'XLim',[0 tmax])
set(gca,'XTickLabels',[])
ylabel('s')
set(gca,'Box','off')

% Generation of Wavelet function
X=zeros(1,lx);
X(1,inx(2))=1;
x=uifwt(X,h);
y(2,:)=x;
hold on

subplot(ch_max,1,2), plot(xax,x)
set(gca,'XLim',[0 tmax])
set(gca,'XTickLabels',[])
ylabel('w')
set(gca,'Box','off')

% Generation of the rest of Wavelet function
for kk=3:ch_max
hold on
X=zeros(1,lx);
X(1,inx(kk))=1;
x=uifwt(X,h);

end
\begin{verbatim}
y(kk,:) = x;
hold on
subplot(ch_max, 1, kk), plot(xax, x)
set(gca, 'XLim', [0 tmax])
set(gca, 'XTickLabels', [])
ylabel(['num2str(kk)])
set(gca, 'Box', 'off')
end

set(gca, 'XTickLabelMode', 'auto')
\end{verbatim}
function y=uworth(lx,h,g)

% Calculation of Orthonomality
% This program is written in MATLAB. For the N-point input signal, the
% program calculates Orthonomality between each Wavelet basis function
% as well as themselves. For convenient visualization, the plot is 3-
% dimensions. Also, the Orthonomality for each Wavelet basis function
% itself is zeroed out.
% lx : The length of input signal
% h & g: The coefficients of lowpass and highpass filters

ch_max=lx;

lf=length(h);
sf=1;
tmax=15;
tmin=0;
tinc=1/sf;
xax=tmin:tinc:tmax;
inx=graycode(lx);

for kk=1:ch_max
    XXX=zeros(lx);
    XXX(1,inx(kk))=1;
    xxx=uifwt(XXX,h);
    y(kk,:)=xxx;
end

figure(2)
clf
for mm=1:lx
    for nn=1:lx
        a(mm,nn)=y(mm,:)*y(nn,:); % Calculation of Orthonomality
        if mm==nn
            a(mm,nn)=0; % Zero out the Orthonomality in
            % the diagonal line
        end
    end
end
mesh(a)
end