Comparison of Frequency Domain Equalizers to Time Domain Equalizers in WCDMA

Yaqoob, Muhammad Atif

2009

Document Version:
Publisher's PDF, also known as Version of record

Link to publication

Citation for published version (APA):

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.
Comparison of Frequency Domain Equalizers to Time Domain Equalizers in WCDMA

Master Thesis

Muhammad Atif Yaqoob

December 2009
Dedicated to

My Parents
Preface

This thesis work is performed to fulfill the requirements of Master of Science degree. All of the thesis work has been carried out over a period of about 6 months from 01-Jun-09 to 18-Dec-09 at system simulations group EAB/KMT/BDS, ST-Ericsson, Lund and is equivalent to 30 ECTS points. This thesis work is supervised by Elias Jonsson from ST-Ericsson and Fredrik Tufvesson from department of Electrical and Information Technology, LTH, Sweden.

I would like to take this opportunity to express my gratitude to Fredrik Tufvesson who has shown very kind support and guidance as my masters program coordinator at LTH as well as my thesis supervisor. I would like to extend my special thanks to Elias Jonsson for all the technical support and encouragement throughout the period of my thesis work at ST-Ericsson. He remained very kind and supportive during all the time. The frequent discussions I had with him to understand different technical details of the thesis helped me achieve my set objectives timely and effectively. I am also thankful to him for his valuable comments and suggestions for writing this thesis report. Also, I would like to thank my Development Manager Anders Ericsson G for providing me the opportunity to pursue a master thesis in his group.

Finally, I would like to thank to all my friends for their enduring support and encouragement. And to my family for their unconditional love and support during all the time I was involved with my studies. Without their support and encouragement, this work would not be possible.
Abstract

Future mobile platforms will contain an increased amount of wireless technologies. As technologies mature it will be necessary to find synergies between them. This could be in the form of reusing hardware blocks and algorithms. By using Fast Fourier Transforms (FFT/IFFT) and performing channel cancellation/equalization in frequency domain for wideband code division multiple access (WCDMA) will provide us a platform which can support LTE and WiFi together with WCDMA. In this thesis, an equalizer based on Fast Fourier Transforming (FFT) the input signal, cancelling the propagation channel in frequency domain, and finally reverting to time domain using an IFFT would be devised for WCDMA. The performance of frequency domain equalizer is then compared with traditional time domain equalizer based on G-RAKE method for different performance metrics.

To evaluate and compare the performance of frequency domain equalizer (FFT) with time domain equalizer (G-RAKE), a simulator is made in IT++. Several simulations are performed and the obtained results are analyzed in Matlab. Firstly, for channel estimation at the receiver, common pilot channel (CPICH) is used and multiple channel estimates are obtained to get a filtered channel estimate. Along-with CPICH pilot symbols, data symbols of a desired user and four other user’s data symbols are transmitted from the base-station (Node B). Using QPSK signaling, un-coded bit-error-rate (BER) results of the desired data user are compared for the two equalization methods. Simulation results are obtained for single path and multi-path propagation channels with different delay spread values. Included in the modeling are one receive and transmit antenna, other cell interference, i.e., non-white noise, frequency errors at mobile station (UE), and analog-to-digital (A/D) quantization.
Table of Contents

Preface ................................................................................................................................. iii
Abstract ................................................................................................................................. iv
Table of Contents .................................................................................................................. iv
List of Figures ........................................................................................................................ v
List of Tables ........................................................................................................................ viii
List of Abbreviations .............................................................................................................. ix

1 Introduction to Project ...................................................................................................... 1
    1.1 Background .................................................................................................................. 1
    1.2 Motivation ................................................................................................................... 1
    1.3 Previous Work ............................................................................................................. 1
    1.4 Problem Statement ...................................................................................................... 2
    1.5 Contribution ................................................................................................................ 2

2 WCDMA Overview .......................................................................................................... 4
    2.1 WCDMA: An Air Interface for UMTS ...................................................................... 4
    2.2 UMTS System Requirements ..................................................................................... 4
    2.3 WCDMA: Main System Parameters ......................................................................... 5
    2.4 Time Units .................................................................................................................. 5
    2.5 Channels ..................................................................................................................... 5
    2.6 Spreading and modulation ......................................................................................... 6
        2.6.1 Channelization code ............................................................................................ 6
        2.6.2 Scrambling code ................................................................................................. 8
        2.6.3 Spreading and Transmission over the Air .......................................................... 9
    2.7 Common Pilot Channel (CPICH) .............................................................................. 11
    2.8 Example Using WCDMA .......................................................................................... 12
        2.8.1 Propagation Channel Delay Time Estimation ...................................................... 13
        2.8.2 Detection of Propagation Channel Constant ..................................................... 14
        2.8.3 Data Demodulation of the Sent Data ................................................................ 14
            2.8.3.1 RAKE ......................................................................................................... 15
            2.8.3.2 Channel Estimation .................................................................................... 16
            2.8.3.3 Interference Estimation ............................................................................. 16
            2.8.3.4 Signal-to-Interference Ratio (SIR) .............................................................. 16
            2.8.3.5 Combiner .................................................................................................... 17

3 Equalizer Model .............................................................................................................. 18
Chapter 4: Implementation and Results

4.1 Simulation Setup
4.1.1 Transmit Signal from Node B
4.1.2 Propagation Channel
4.1.3 Signal Reception at UE

4.2 GRAKE Equalizer Receiver
4.2.1 Channel Estimation
4.2.2 Equalizer Filter and SIR
4.2.3 Combiner

4.3 FFT Equalizer Receiver
4.3.1 Channel Estimation
4.3.2 Equalizer
4.3.3 Despreader Output, BER, and SIR Estimate

4.4 Results and Discussion
4.4.1 Parameters Optimization
4.4.1.1 GRAKE Parameters Optimization
4.4.1.2 FFT Parameters Optimization
4.4.2 Chip Offset Comparison
4.4.3 Frequency Error Comparison
4.4.4 Colored Noise Comparison
4.4.5 Analog-to-Digital Converter Comparison
4.4.6 Complexity Comparison
4.4.6.1 FFT Equalizer Output Samples
4.4.6.2 Computations For FFT
4.4.6.3 Computations for NP-GRAKE
4.4.7 ITU-Pedestrian B Channel Comparison

5 Conclusion
Appendix .......................................................................................................................... 65

A. Parametric Equalizer ................................................................................................. 65

B. Differentiate $\|Hg - e\|^2$ w.r.t $g$ ............................................................................. 67

References ....................................................................................................................... 69
List of Figures

Figure 2.1: OVSF code tree ................................................................................................................................. 8
Figure 2.2: Relationship between Spreading and Scrambling ................................................................. 8
Figure 2.3: Multi-path propagation channel .................................................................................................. 10
Figure 2.4: Frame structure and modulation pattern for Common Pilot Channel ................................. 11
Figure 2.5: Propagation channel estimates .................................................................................................... 14
Figure 2.6: Data Demodulation Using RAKE ............................................................................................... 15
Figure 2.7: RAKE Receiver Block Diagram ................................................................................................ 17
Figure 3.1: Communication System Block Diagram ...................................................................................... 18
Figure 3.2: FFT Equalizer Receiver Block Diagram ....................................................................................... 24
Figure 4.1: Root Raised Cosine Filter; sample rate = 4 per chip, filter length = 14 chips ......................... 29
Figure 4.2: Transmission of WCDMA signal from Node B ........................................................................... 30
Figure 4.3: Received Signal at UE via Matched Filtering ............................................................................ 32
Figure 4.4: Non-Parametric G-RAKE Receiver Channel Estimation ......................................................... 33
Figure 4.5: Non-Parametric G-RAKE Equalizer Filter and SIR ................................................................... 35
Figure 4.6: Non-Parametric G-RAKE Combiner Output and BER ............................................................. 36
Figure 4.7: FFT Receiver Channel Estimation ............................................................................................... 37
Figure 4.8: FFT Equalizer Input/Output Block Diagram ............................................................................. 39
Figure 4.9: FFT Despreader Output, BER, and SIR Estimate ...................................................................... 40
Figure 4.10: FFT Equalizer Output Chip Samples vs Delay Spread ......................................................... 57
Figure 4.11: BER vs $E_{clor}$ for FFT and GRAKE equalizer using ............................................................... 61
List of Tables

Table 2.1: Main WCDMA parameters........................................................................................................ 5
Table 2.2: WCDMA Downlink Data Rate vs Spreading Factor................................................................. 7
Table 4.1: Simulation setup: different users/channels with their $sf$ and power assigned ..................... 28
Table 4.2: $N_{\text{FFT}}$ vs BER for two path channel, ‘$ds$’ = 5 chip, $\lambda_h = 1/200$, without AWGN .......... 43
Table 4.3: FFT Equalizer BER results for different delay spread values, $N_{\text{FFT}} = 2048$, without AWGN, $\lambda_h = 1/200$, ................................................................................................. 44
Table 4.4: Chip Offset Comparison: Single Path Channel ........................................................................ 45
Table 4.5: FFT equalizer BER vs $\epsilon$, Iorloc = 6 dB, two path channel ‘$ds$’ = 0.75 chip ................. 46
Table 4.6: Chip Offset Comparison: Two Path Channel ‘$ds$’ = 0.75 Chip .............................................. 47
Table 4.7: FFT equalizer BER results at different offset positions using Iorloc = 8 dB ...................... 47
Table 4.8: Chip Offset Comparison: Two Path Channel ‘$ds$’ = 1 Chip ................................................ 48
Table 4.9: Chip Offset Comparison: Two Path Channel ‘$ds$’ = 4.5 Chip ............................................. 49
Table 4.10: Chip Offset Comparison: Two Path Channel ‘$ds$’ = 5 Chip ............................................ 50
Table 4.11: Frequency Error Comparison: Single Path Channel, Offset = 0 Chip ............................. 51
Table 4.12: Frequency Error Comparison: Single Path Channel, Offset = 1/2 Chip........................ 52
Table 4.13: Frequency Error Comparison: Two Path Channel ‘$ds$’ = 5 Chip, Offset = 0 Chip .......... 53
Table 4.14: Colored Noise Comparison ................................................................................................. 54
Table 4.15: A/D Conversion Comparison: Single Path, Offset = 0 Chip ............................................ 55
Table 4.16: A/D Conversion Comparison: Single Path, Offset = 1/2 Chip .......................................... 55
Table 4.17: A/D Conversion Comparison: Two Path ‘$ds$’ = 5 Chip, Offset = 0 Chip ......................... 56
Table 4.18: Complexity Comparison between FFT and G-RAKE.......................................................... 60
### List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2G</td>
<td>2nd Generation Mobile Systems</td>
</tr>
<tr>
<td>3G</td>
<td>3rd Generation Mobile Systems</td>
</tr>
<tr>
<td>3GPP</td>
<td>3rd Generation Partnership Project</td>
</tr>
<tr>
<td>A/D</td>
<td>Analog-to-Digital Converter</td>
</tr>
<tr>
<td>AWGN</td>
<td>Additive White Gaussian Noise</td>
</tr>
<tr>
<td>BER</td>
<td>Bit Error Ratio</td>
</tr>
<tr>
<td>BLER</td>
<td>Block Error Ratio</td>
</tr>
<tr>
<td>BS</td>
<td>Base Station (Node B for UMTS)</td>
</tr>
<tr>
<td>CDMA</td>
<td>Code Division Multiple Access</td>
</tr>
<tr>
<td>CPICH</td>
<td>Common Pilot Channel</td>
</tr>
<tr>
<td>D/A</td>
<td>Digital-to-Analog Converter</td>
</tr>
<tr>
<td>DS-CDMA</td>
<td>Direct Sequence - CDMA</td>
</tr>
<tr>
<td>Eclor</td>
<td>The chip power transmitted from Node B</td>
</tr>
<tr>
<td>EDGE</td>
<td>Enhanced Data Rates for GSM Evolution</td>
</tr>
<tr>
<td>ETSI</td>
<td>European Telecommunications Standards Institute</td>
</tr>
<tr>
<td>FDD</td>
<td>Frequency Division Duplex</td>
</tr>
<tr>
<td>FDE</td>
<td>Frequency Domain Equalization</td>
</tr>
<tr>
<td>FFT</td>
<td>Fast Fourier Transform</td>
</tr>
<tr>
<td>GPRS</td>
<td>General Packet Radio Service</td>
</tr>
<tr>
<td>IFFT</td>
<td>Inverse Fast Fourier Transform</td>
</tr>
<tr>
<td>IMT-2000</td>
<td>International Mobile Telecommunication – 2000</td>
</tr>
<tr>
<td>ISI</td>
<td>Inter Symbol Interference</td>
</tr>
<tr>
<td>ITU</td>
<td>International Telecommunication Union</td>
</tr>
<tr>
<td>Ior</td>
<td>The total transmit power from Node B (normalized to one)</td>
</tr>
<tr>
<td>Ioc</td>
<td>Other-cell interference (including radio noise)</td>
</tr>
<tr>
<td>IorIoc</td>
<td>Amount of inter-cell interference</td>
</tr>
<tr>
<td>LOS</td>
<td>Line of sight</td>
</tr>
<tr>
<td>MAI</td>
<td>Multiple Access Interference</td>
</tr>
<tr>
<td>ML</td>
<td>Maximum Likelihood</td>
</tr>
<tr>
<td>MMSE</td>
<td>Minimum Mean Square Error</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------</td>
<td>------------</td>
</tr>
<tr>
<td>MRC</td>
<td>Maximum Ratio Combining</td>
</tr>
<tr>
<td>MS</td>
<td>Mobile Station (UE for UMTS)</td>
</tr>
<tr>
<td>N_{FFT}</td>
<td>FFT window size</td>
</tr>
<tr>
<td>NP-GRAKE</td>
<td>Non-parametric Generalized RAKE</td>
</tr>
<tr>
<td>PN</td>
<td>Pseudo-Noise</td>
</tr>
<tr>
<td>QPSK</td>
<td>Quadrature Phase Shift Keying</td>
</tr>
<tr>
<td>RRC</td>
<td>Root Raised Cosine</td>
</tr>
<tr>
<td>SIR</td>
<td>Signal to Interference Ratio</td>
</tr>
<tr>
<td>UMTS</td>
<td>Universal Mobile Telecommunication System</td>
</tr>
<tr>
<td>UE</td>
<td>User Equipment</td>
</tr>
<tr>
<td>WCDMA</td>
<td>Wideband Code Division Multiple Access</td>
</tr>
<tr>
<td>ds</td>
<td>delay spread</td>
</tr>
<tr>
<td>os</td>
<td>over sampling</td>
</tr>
<tr>
<td>pdf</td>
<td>probability distribution function</td>
</tr>
<tr>
<td>sf</td>
<td>Spreading Factor</td>
</tr>
<tr>
<td>w.r.t</td>
<td>with respect to</td>
</tr>
</tbody>
</table>
Chapter 1

1 Introduction to Project

1.1 Background

In today's platforms equalizers that neutralize inter-symbol interference (ISI) from the serving cell as well as other cell interference is becoming a standard feature. Future platforms will contain an increased amount of different wireless technologies. As technologies mature it will be necessary to find synergies between them. This could be in the form of reusing hardware blocks and algorithms. One such example is also shown in [11], where the dedicated hardware blocks are shared and reused by different systems such as DAB, DVB-T, HIPERLAN-2 and UMTS.

1.2 Motivation

In this thesis we will compare an equalizer based on frequency domain equalization (FDE) to traditional time domain method (G-RAKE) for WCDMA in the downlink FDD mode. Frequency domain equalization will be performed by Fast Fourier Transforming (FFT) the input signal, cancelling the influence of propagation channel in frequency domain, and finally reverting to time domain using an IFFT.

Albeit using an FFT for WCDMA equalization might be more computationally expensive than today's time domain methods, however, the total complexity may go down when combined with OFDM based technologies. Using Fast Fourier Transforms (FFT/IFFT) for frequency domain equalization or channel cancellation in WCDMA will provide us a platform which can support LTE and WiFi together with WCDMA.

1.3 Previous Work

Multiple implementation strategies for the frequency domain equalization (FDE) of DS-CDMA signal have been proposed in the literature [12], [13] and [14]. In theses references, the usefulness of FDE over traditional time domain (RAKE) equalization is illustrated by doing channel equalization based on Minimum Mean Square Error (MMSE) criterion to
combat multiple-access interference (MAI) and inter-symbol interference (ISI) and having the same or lesser complexity requirements as well. Also [16] demonstrated that MMSE based equalizer has better BER performance as compared to zero-forcing (ZF) and RAKE equalizer in downlink CDMA.

Moreover, it is proposed that frequency domain equalization combined with single carrier modulation has comparable performance results as of OFDM system having nearly the same complexity requirements [15].

**1.4 Problem Statement**

The objective with the thesis is to evaluate and compare WCDMA equalizers based on the G-RAKE, and FFT methods for multi-path zero Doppler propagation channels.

Metrics used to evaluate are:

1. Bit error rate for QPSK signaling
2. Sensitivity to frequency errors (the mobile phone (UE) internal clock and the base-station (Node B) clocks are not synchronized; the clocks decide the sampling rate in the A/D converters)
3. Sensitivity to analog to digital (A/D) quantization
4. Number of complex operations required per data symbol

**1.5 Contribution**

A simulator is made in IT++ to compare the different WCDMA equalizers and to achieve the above mentioned objectives. ML-based G-RAKE method (Non-parametric G-RAKE) is used for time domain equalization and FFT method is used for frequency domain equalization, where FFT window size is selected as 2048. More details about the equalizer models and implementation is provided in chapter 3 and 4.

Simulations are performed to compare an FFT equalizer to G-RAKE equalizer for non-fading or non-time selective propagation channels. Using QPSK signaling for the desired data user, raw bit-error-rate (BER) performance results are compared to evaluate the performance of the two equalizers. Additive White Gaussian Noise (AWGN) is added to the received signal; and the BER performance is compared w.r.t IoIoc values.
By using a tuned parameter \( \varepsilon \) which represents the noise variance, simulations are performed to estimate the best expected performance of the FFT based equalizer for various channel conditions. Optimum values of \( \varepsilon \) used in the simulations are estimated by observing the un-coded BER performance results. Simulation results are shown in section 4.4 and onwards.

Furthermore, different frequency errors (5, 10, 20 Hz) are generated at the mobile-station (UE) to simulate the synchronization mismatch between the clocks of UE and Node B, and BER performance of the two equalizer methods is compared. Non-white noise, i.e., interference from another Node B is added to the desired signal with appropriate scaling and BER performance results are evaluated using the two equalization methods. Also analog-to-digital quantization results for different (8, 10, and 12-bit) fixed point implementations with appropriate A/D-constant (K) values used to scale the signal for integer representation, are observed. After the simulation results, a basic complexity comparison is made in terms of total complex operations required per data symbol using G-RAKE and FFT method based on the algorithms used for FFT/IFFT and for other matrix operations. Conclusion and possible future work for the FFT based equalization method is given in Chapter 5.
2 WCDMA Overview

2.1 WCDMA: An Air Interface for UMTS

The success of second-generation (2G) cell phones, especially the Global System for Mobile communications (GSM), motivated the development of a successor system [1]. The 2G systems provided not only voice traffic on mobile phones but also some basic data services, e.g., the short message service (sms). But these data services are limited to use as higher data rates are not available which eventually limits the use of mobile phones for high quality multimedia services.

Under the name of IMT-2000, International Telecommunication Union (ITU) asked for proposals to standardize the work of third-generation (3G) mobile telecommunication systems. A standard has been formulated by 3GPP, which is a joint venture of several standard developing organizations from different regions of the world (Europe, Japan, Korea, USA, and China).

2.2 UMTS System Requirements

The requirements for the third-generation systems are given as under [3]:

- Bit rates up to 2 Mbps;
- Variable bit rate to offer bandwidth on demand;
- Multiplexing of services with different quality requirements on a single connection, e.g. speech, video and packet data;
- Co-existence of second and third generation systems and inter-system handovers for coverage enhancements and load balancing;
- High spectrum efficiency.
2.3 WCDMA: Main System Parameters

In the standardization forums, WCDMA technology has emerged as the most widely adopted third generation air interface. Table 2.1 shows the main system design parameters related to the WCDMA air interface [3]:

Table 2.1: Main WCDMA parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiple access method</td>
<td>DS-CDMA</td>
</tr>
<tr>
<td>Duplexing method</td>
<td>Frequency division duplex / time division duplex</td>
</tr>
<tr>
<td>Base station synchronization</td>
<td>Asynchronous operation</td>
</tr>
<tr>
<td>Chip rate</td>
<td>3.84 Mcps</td>
</tr>
<tr>
<td>Frame length</td>
<td>10 ms</td>
</tr>
<tr>
<td>Service multiplexing</td>
<td>Multiple services with different quality of service requirements multiplexed on one connection</td>
</tr>
<tr>
<td>Multirate concept</td>
<td>Variable spreading factor and multicode</td>
</tr>
<tr>
<td>Detection</td>
<td>Coherent using pilot symbols or common pilot</td>
</tr>
<tr>
<td>Multiuser detection, smart antennas</td>
<td>Supported by the standard, optional in the implementation</td>
</tr>
</tbody>
</table>

2.4 Time Units

Transmission Time Interval (TTI): 10, 20, 40, 80 ms, which reflect the different time durations used to transmit coded packets of information.

- Frame: 10 ms
- Slot: 15 slots = 1 frame
- Chip: 38400 chips = 1 frame or 2560 chips = 1 slot

2.5 Channels

UMTS uses a number of logical and physical channels. The logical channels in UMTS are of two types: common transport channels and dedicated transport channels. Example of common channels is BCH (broadcast channel), PCH (paging channel), and RACH (random access channel) and are shared by a group of MSs in a cell.

Physical channels are used to carry different control and user information. Information is mapped on to the physical channels for transmission from base station (Node B) to mobile
station (MS) or vice versa. A physical channel that is used for data and control signaling in WCDMA is known as dedicated physical data channel (DPDCH) and is used for both uplink and downlink transmission. Common pilot channel (CPICH) is a downlink physical channel used to transmit known pilot symbols for channel estimation at UE and is discussed in more detail in section 2.5. Details about other different channels used in WCDMA can be found in the standard and also in literature [3], and [4].

**2.6 Spreading and modulation**

Direct-Sequence code division multiple-access (DS-CDMA) is employed in WCDMA. Different users are assigned different codes (in the form of chips) derived from CDMA. Codes are used to separate different users, and also they distinguish between users and other control channels. Two types of codes: channelization codes and scrambling codes are used to spread the user information and to assign identification tag to the different users. Basic theory about CDMA is described in the references [5], [6], and [7].

**2.6.1 Channelization code**

Orthogonal channelization codes are used in WCDMA. Let $C^{(n)}$ be a vector of length 4, 8, 16, 32, 64, 128, or 256. The length of the vector is called the spreading factor ($sf$). And a channelization code is characterized by

$$
\sum_{i=1}^{sf} C^{(n)}_i C^{(m)}_i = \begin{cases} 
    sf, & n = m \\
    0, & n \neq m 
\end{cases}
$$

where each element of $C^{(n)}$ equals $\pm 1$, and $C^{(n)}$, $C^{(m)}$ are two channelization codes.

Orthogonal variable spreading factor (OVSF) code tree is used in WCDMA to assign the channelization codes to different users. Depending on the user’s data rate, code is assigned to the user from the code tree. Where a short code implies high data rate and long code is used for low data rate user. WCDMA downlink data rates (modulation scheme: QPSK) w.r.t the spreading factor used are listed in table 2.2.
Table 2.2: WCDMA Downlink Data Rate vs Spreading Factor

<table>
<thead>
<tr>
<th>Code length (spreading factor)</th>
<th>WCDMA (Downlink) Data Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>1.92 Mbps</td>
</tr>
<tr>
<td>8</td>
<td>960 kbps</td>
</tr>
<tr>
<td>16</td>
<td>480 kbps</td>
</tr>
<tr>
<td>32</td>
<td>240 kbps</td>
</tr>
<tr>
<td>64</td>
<td>120 kbps</td>
</tr>
<tr>
<td>128</td>
<td>60 kbps</td>
</tr>
<tr>
<td>256</td>
<td>30 kbps</td>
</tr>
<tr>
<td>512</td>
<td>15 kbps</td>
</tr>
</tbody>
</table>

An OVSF code tree is shown in figure 2.1. Since a constant chip rate (3.84 Mcps) is used in WCDMA, different data rates are achieved by directly changing the spreading factor. Multiplying the spreading factor by 2 means the data rate is reduced by 2 correspondingly. Spreading factor of 1 can only be used if only one data stream is transmitted, because no other code can be used in this case. Also if parent code e.g. C_{(4,3)} is assigned to one user, then the codes e.g., C_{(8,5)}, C_{(8,6)}, … from the parent tree can not be assigned to other users at the same time.
Scrambling code is uniquely assigned to different base stations in the network. Also each (UE) has its own scrambling code. It is used as an identification tag; it separates the base stations (in downlink) and UEs (in uplink). Scrambling does not spread any further the chip sequence generated by spreading, but only scramble it by simple multiplication. In figure 2.2, relationship between spreading and scrambling is shown:

![Figure 2.1: OVSF code tree](image1)

**2.6.2 Scrambling code**

Scrambling code is uniquely assigned to different base stations in the network. Also each (UE) has its own scrambling code. It is used as an identification tag; it separates the base stations (in downlink) and UEs (in uplink). Scrambling does not spread any further the chip sequence generated by spreading, but only scramble it by simple multiplication. In figure 2.2, relationship between spreading and scrambling is shown:

![Figure 2.2: Relationship between Spreading and Scrambling](image2)
They are quasi-orthogonal random sequences; e.g., a scrambling code is a vector $q^{(n)}$ of length 38400, where each vector entry of $q^{(n)}$ is a complex number of type $(\pm 1 \pm i)/\sqrt{2}$. The scrambling code is constructed to approximate a random sequence, that is,

$$\frac{1}{38400} \sum_{i=0}^{38399} q_i^{(n)} \cdot (q_{(i-m) \mod 38400})^* = \begin{cases} 1, & m = 0 \\ \approx 0, & m \neq 0 \end{cases},$$

observe that if $m \neq 0$ the correlation is not equal to zero, but almost.

### 2.6.3 Spreading and Transmission over the Air

Let $S_n$ denote the symbols to be sent. Spread the symbols $S_n$ using the channelization code $C$, that is,

$$S_n \rightarrow S_nC_1, S_nC_2, \ldots, S_nC_{sf},$$

then multiply each spread term with the cell specific scrambling code $q$,

$$S_nC_1q_1, S_nC_2q_2, \ldots, S_nC_{sf}q_{sf},$$

If $S_{n+1}$ is the next symbol to be transmitted, we get similarly, first by spreading

$$S_{n+1} \rightarrow S_{n+1}C_1, S_{n+1}C_2, \ldots, S_{n+1}C_{sf},$$

and afterwards by scrambling

$$\rightarrow S_{n+1}C_1q_{sf+1}, S_{n+1}C_2q_{sf+2}, \ldots, S_{n+1}C_{sf}q_{2sf}$$

Observe that the channelization code is reused for each new symbol, but the scrambling code vector continues to be read until we reach the end of it. When the end of the scrambling code has been reached, we start reading it from the beginning.

The purpose of the channelization codes is for the cell to be able to transmit simultaneously several symbols to different UEs.

We have that at any given time a “CHIP” of the type $S_nC_jq_k = X_n$ which needs to be transmitted.

Alternatively, the above operation can be described as:
Symbols $\rightarrow$ Spreading $\rightarrow$ Scrambling $\rightarrow$ Transmit $\rightarrow$ Propagation Channel Distortion $\rightarrow$ UE-reception

Also, pulse shaping is applied on the generated chips before transmission. In WCDMA, standard root raised cosine (rrc) pulse shape filter is used at the transmitter and matched filter with the same filter is performed at the receiver. More details about the pulse shape filter used is given section 4.1.1.

The sent $X_n$ are affected by the propagation channel. The propagation channel will change the amplitude and phase of the chips.

The received chips at the UE can be modeled as

$$y_n = \sum_{l=1}^{L} h_l \cdot X_{n-\tau(l)} + \nu_n$$

Figure 2.3: Multi-path propagation channel

where $h_l$ is a complex number modeling path $l$, and $\tau(l)$ is the time delay between the cell and the UE (time unit in chips) and $\nu_n$ is the disturbance (noise) from other cells transmission. Different multi-path signal components arriving at the MS from two different BSs are shown in the above figure 2.5.
2.7 Common Pilot Channel (CPICH)

The CPICH is a fixed rate (30kbps, SF=256) downlink physical channel that carries a pre-defined bit sequence. Figure 2.5 shows the frame structure and modulation pattern of the CPICH. The figure shows that one radio frame having duration of 10ms is composed of 15 slots where each slot carries 2560 chips. 10 CPICH symbols (QPSK modulated of the form \(1-i\)) corresponding to 20 bits are mapped onto these 2560 chips for channel estimation at the UE. In total 38,400 = 2560x15 chips are transmitted in one frame.

![Diagram of CPICH frame structure and modulation pattern](image)

---

**Figure 2.4:** Frame structure and modulation pattern for Common Pilot Channel

CPICH symbols → Spreading using a 256 chip long channelization code = \(C_{(256,1)}\), all ones \((1, 1, \ldots, 1)\) → Apply cell specific scrambling code → Send over the air
\section*{2.8 Example Using WCDMA}

Assume an AWGN channel with the following impulse response

\[ h_l = \begin{cases} 
1, & l = 1 \\
0, & l = 2, 3, 4, \ldots 
\end{cases} \]

Assume that 256 chips are received corresponding to one CPICH symbol. The chips received at the UE can be expressed as

\[ y_n = X_n^{(i)} \]

\[ + O_n (\text{other channels or users sent from the same cell using orthogonal channelization codes}) \]

\[ + q_n (\text{inter-cell interference, chips received from other BSs}) \]

The subscript ‘n’ shows the chip index and superscript ‘i’ is used for a channelization code.

Now if the received signal \( y_n \) is correlated with the conjugate of \( X_n^{(i)} \) and normalized by \( 1/256 \), we have the following expression

\[ \frac{1}{256} \sum_{n=0}^{255} y_n^* X_n^{(i)} = \frac{1}{256} \sum_{n=0}^{255} |X_n^{(i)}|^2 \]

\[ + O_n \left( \text{other channels use channelization codes orthogonal to the CPICH} \right) \]

\[ + \frac{1}{256} \sum_{n=0}^{255} q_n^* (X_n^{(i)})^* \]

\[ \Rightarrow \quad \frac{1}{256} \sum_{n=0}^{255} y_n^* (X_n^{(i)})^* = 1 + v \]

And the expected value of the power is then

\[ E \left\{ \left| \frac{1}{256} \sum_{n=0}^{255} y_n^* (X_n^{(i)})^* \right|^2 \right\} = E \left\{ |1 + v|^2 \right\} \]
Now

\[ E\left[ |1 + v|^2 \right] = E\left[ |1 + v|^2 + 2 \Re(v) \right] \]

Since \( q_n \) & \( X_n^{(i)} \) are uncorrelated and \( E\left[ q_n^2 \right] = E\left[ X_n^{(i)} \right] = 0 \), this implies

\[ E\left[ |1 + v|^2 \right] = 1 + \frac{1}{256} \sum_{n=0}^{255} E\left[ |q_n|^2 \right] \]

\[ E\left[ |X_n^{(i)}|^2 \right] = 1 + \frac{\sigma^2}{256} \]

where \( \sigma^2 = E\left[ |q_n|^2 \right] \) and equals the average power of the inter-cell interference.

### 2.8.1 Propagation Channel Delay Time Estimation

We will use the CPICH again in our analysis now to detect the time location of the propagation channel paths. The signal received at the UE is given by

\[ y_n = \sum_{l=1}^{L} h_l \left[ X_{n-l}^{(i)} + \text{other channels from the cell } (O_{n-l}) \right] \]

\[ + \text{ Inter-cell Interference } (q_n) \]

(2.3)

where \( X_n \) is the chip sequence containing the CPICH symbols, channelization code, and scrambling code. And \((L)\) represent the number of multi-paths arriving at the MS from Node B.

For different time/chips shifted version of \( X_n \), lets define the correlation of \( y_n \) with \( (X_{n-m})^\ast \) as follows:

\[ Z_m = \sum_{n=0}^{255} y_n (X_{n-m})^\ast = \sum_{l=1}^{L} \sum_{n=0}^{255} h_l \left[ X_{n-l}^{(i)} + O_{n-l} \right] X_{n-m}^\ast + q_n X_{n-m}^\ast \]

The expectation value of \( Z_m \) is

\[ E\left[ Z_m \right] = \begin{cases} 256 h_l, & m = \tau(l) \text{ for some } l = 1, \ldots, L, \\ 0, & m \neq \tau(l) \text{ for some } l = 1, \ldots, L \end{cases} \]

(2.4)

The result follows from the fact that \( q_n \) & \( X_{n-m}^\ast \) are uncorrelated due to different scrambling codes, \( O_n \) & \( X_{n-m}^\ast \) are orthogonal if \( m = 0 \), and uncorrelated if \( m \neq 0 \) due to scrambling code
properties. Similarly $X_n$ & $X_{n-m}$ are uncorrelated if $m \neq 0$ from the scrambling code properties.

Figure 2.6 shows different peaks corresponding to different propagation paths (different values of $m$).

![Figure 2.5: Propagation channel estimates](image)

A window, that is, the number of $m$ used, typically spans 64-128 chips. A typical propagation channel has a delay spread of roughly 5 chips.

### 2.8.2 Detection of Propagation Channel Constant

From the previous section; to find the propagation channel amplitude and phase distortions, the use of CPICH gives us the following result:

$$
\frac{1}{256} \sum_{n=0}^{255} y_n \cdot (X_{n-\tau(l)})^* \approx h_i
$$

where $h_i$ includes the over all channel response (Tx filter + propagation channel + Rx filter).

### 2.8.3 Data Demodulation of the Sent Data

The received signal is as before

$$
y_n = \sum_{l=1}^{L} h_i \cdot [X_{n-\tau(l)} + Other\ channels\ from\ the\ cell\ (O_n)] + Inter-cell\ Interference\ (q_n)
$$

where $X_n$ is $n^{th}$ chip containing the data we want to extract, multiplied by a channelization and scrambling code. To simplify the notation assume data symbol $S$ corresponds to chip indexes 1, …, $sf$. We want to extract the symbol $S$.  

14
2.8.3.1 RAKE

The despreading is done in the RAKE, that is, correlate $y_n$ with $(\tilde{X}_{n-\tau(t)})^*$, where $\tilde{X}_n$ is the chip sequence formed by the channelization code and scrambling code of the desired user or symbol. Rake outputs are obtained for $l$ paths or branches as described in the following equation and also shown in figure 2.7.

$$r_j = \frac{1}{sf} \sum_{n=0}^{sf-1} y_n (\tilde{X}_{n-\tau(t)})^* \approx h_i S$$

by using the below mentioned results:

$$\frac{1}{sf} \sum_{n=0}^{sf-1} (\tilde{X}_{n-\tau(t)})^* X_{m-\tau(t)} = \begin{cases} S, & m = n \\ \approx 0, & m \neq n \end{cases}$$

$$\frac{1}{sf} \sum_{n=0}^{sf-1} (\tilde{X}_{n-\tau(t)})^* (\text{Other channels}) \approx 0$$

$$\frac{1}{sf} \sum_{n=0}^{sf-1} (\tilde{X}_{n-\tau(t)})^* (\text{Intercell Interference}) \approx 0$$

A simple block diagram showing the operations required to extract the sent symbol $S$ is shown in figure 2.8.
2.8.3.2 Channel Estimation

Using the CPICH we can estimate $h_l$, since the symbol $S$ is then known. In a slot, we have 10 CPICH symbols and consequently we have 10 channel samples per slot, call them

$$h_{l,i}^{samp}, \quad i = 1, \cdots, 10$$

To calculate a channel estimate, use the average of the samples over a slot, that is,

$$h_l^{est} = \frac{1}{10} \sum_{i=1}^{10} h_{l,i}^{samp}$$

Or the channel samples can be filtered by using the following equation:

$$h_{l,i}^{est} = \lambda h_{l,i}^{samp} + (1 - \lambda) h_{l,i-1}^{est}, \quad i = 1, \cdots, 10$$

2.8.3.3 Interference Estimation

An estimate of the inter-cell interference is found from

$$I_i = \frac{1}{9} \sum_{i=1}^{10} \left| h_{l,i}^{samp} - h_l^{est} \right|^2, \quad l = 1, \cdots, L$$

To be precise, the expectation value of $I_i$ also contains inter-symbol interference from the cell we are trying to demodulate. That is, the different paths carry own copy of the sent signal and since they are not time-aligned, they will interfere with each other.

2.8.3.4 Signal-to-Interference Ratio (SIR)

SIR is defined as:

$$SIR = \sum_{l=1}^{L} \frac{\left| h_l^{est} \right|^2}{I_l}$$

and is computed once per slot. It is a quality measure of the propagation channel. The higher the ratio the lesser the errors will be made when determining the sent data.
2.8.3.5 Combiner

The sent symbol $S$ is recovered by multiplying the RAKE output with the conjugate of the channel estimates, that is,

$$\sum_{l=1}^{L} \left( h_{l}^{est} \right)^{*} r_{l} = \left( \sum_{l=1}^{L} \frac{\left( h_{l}^{est} \right)^{*}}{I_{l}} \right) S$$

A real positive value is obtained if $h_{l}^{est}$ is well estimated. We divide by $I_{l}$ to favor paths experiencing low interference. Figure 2.8 shows the operations of a simple RAKE receiver with combiner output.

![RAKE Receiver Block Diagram](image)

**Figure 2.7:** RAKE Receiver Block Diagram
Chapter 3

3 Equalizer Model

3.1 Communication System Diagram

A simplified RAKE receiver block diagram is shown in figure 2.7. The filter (equalizer) weights shown in that figure can be determined in several ways for time domain equalization. Here, we consider the following two methods

1) Non-parametric MMSE-based Equalization
2) ML-based Equalization (Nonparametric GRAKE)
3.2.1 Non-parametric MMSE-based Equalization

Statistical signal model:

\[ y_m = h_m s_0 + \frac{\text{other chips / symbols transmitted from serving cell}}{m} + \frac{\text{other chips / symbols transmitted from other cells}}{(a)} + \text{radio noise} \]

(3.1)

where \( y_m \) is the received sampled signal (after descrambling/despreading), the sampling rate could be arbitrary (any os),

\( s_0 \) is the transmitted symbol/chip to demodulate, with \( \| s_0 \| = 1 \),

\( h_m \) is the propagation channel including \( R_x \& T_x \) radio filters affecting symbol/chip \( s_0 \) at time instant \( t_m \), and

\( t_m \) is absolute sampling time.

Find a filter \( g \) that solves

\[
\min_g E \left[ \left( \sum_{m=-M}^{M} y_m g_m - s_0 \right)^2 \right]
\]

(3.2)

Here we assume that the time coefficients of \( g \) are placed on a fixed grid ranging from \( -M \) to \( M \).

(P1): Also assume that \( s_0 \) is uncorrelated with the noise in term (i), (ii) & (iii) in the statistical model.

Expanding (3.2) gives

\[
\min_g \left[ \tilde{g}^H E \left\{ \tilde{y} \tilde{y}^H \right\} \tilde{g} - \operatorname{Re} E \left\{ s_0 \sum_{m=-M}^{M} y_m g_m \right\} \right]
\]

(3.3)

where \( \tilde{g} = (g_{-M}, \ldots, g_0, \ldots, g_M) \),

\( \tilde{y} = (y_{-M}, \ldots, y_0, \ldots, y_M) \),

\( \tilde{y}^H = \text{Hermitian Transpose of } \tilde{y} \), and

\( \tilde{g}^H = \text{Hermitian Transpose of } \tilde{g} \)
Since \( \| s_0 \|^2 = 1 \), the assumptions in (P1) gives us
\[
\text{Re} \mathbb{E} \left\{ s_0^* \sum_{m=-M}^{M} y_m g_m \right\} = \text{Re} \sum_{m=-M}^{M} h_m g_m
\]
\[
\therefore s_0 s_0^* = \| s_0 \|^2 = 1
\]
Thus (3.3) can be written as,
\[
\min_g \left[ \mathbf{g}^H \mathbb{E} \left\{ \mathbf{y} \mathbf{y}^H \right\} \mathbf{g} - \text{Re} \sum_{m=-M}^{M} h_m g_m \right]
\]
(3.4)
Taking the derivative of (3.4) w.r.t \( g \) and solving for zero gives, see Appendix B
\[
g = \left( R_y^t \right)^{-1} \tilde{h}^*
\]
(3.5)
where \( \tilde{h} = (h_{-M}, \cdots, h_0, \cdots, h_M) \) and by omitting the scalar multiplier in the (3.5).
Here \( R_y \) can be computed by filtering the received samples as
\[
R_y(l, k) = \lambda \cdot R_y^\text{old}(l, k) + (1 - \lambda) y_m^* y_{m-(k-l)}
\]
(3.6)
where the filtering parameter \( \lambda \) depends on the Doppler spread.

### 3.2.2 ML-based Equalization (Non-parametric G-RAKE)

Referring back to (3.1)
\[
y_m = h_m s_0 + n_m
\]
where \( n_m \) is modeling the unwanted part of the received signal, that is, the noise.
Assume we have \( m = 1, \cdots, M \) sampled \( y_m \); collect them into a vector \( \mathbf{y} = (y_1, \cdots, y_M) \). Let’s then try to fit the vector \( \mathbf{y} \) to a multi-dimensional probability distribution. A first choice is the Gaussian probability distribution, since it is simple to manipulate and contains few parameters to estimate.
Assuming the probability distribution for \( \mathbf{y} \) to be Gaussian, we have the following pdf
\[
e^{-\left( \mathbf{y} - \mathbf{h} s_0 \right)^H R_n^{-1} \left( \mathbf{y} - \mathbf{h} s_0 \right)}
\]
(3.7)
where \( \mathbf{h} = (h_{-M}, \cdots, h_0, \cdots, h_M) \)
and \( R_n \) is the covariance matrix of the noise defined as

\[
(R_n)_{m,m} = E\{n^*_m n_m\}
\]

Maximizing (3.7) with respect to \( s_0 \), that is, take the derivative of (3.7) with respect to \( s_0 \) and solve the result for \( s_0 \). Then, \( s_0 \) is given by

\[
s_0 = \left( R^{-1}_n \tilde{h}^* \right)^T \tilde{y},
\]

(3.8)

by omitting a scalar factor \( \left( \tilde{h}^* R^{-1}_n \tilde{h} \right)^{-1} \).

Comparing with (3.5) shows that the filter \( g \) here is given by \( \left( R'_n \right)^{-1} \tilde{h}^* \) and in (3.5) is given by \( \left( R'_n \right)^{-1} \tilde{h}^* \).

Assuming the noise \( n_m \) is uncorrelated with \( h_m s_0 \), we have

\[
R = E\left\{ \left( \tilde{h}s_0 + \tilde{n} \right)\left( \tilde{h}s_0 + \tilde{n} \right)^H \right\}
= E\left\{ \tilde{h}\tilde{h}^H \right\} = \tilde{h}\tilde{h}^H + R_n
\]

(3.9)

In practice, it is difficult to estimate \( R_n \) without knowing the sent symbol \( s_0 \). If, however, we have a pilot channel, modeled as

\[
y_m = h_m p_0 + \tilde{n}_m
\]

(3.10)

The noise \( \tilde{n}_m \) is of the same nature as \( n_m \) (which is most often the case); we can now estimate \( R_n \) based on the pilot channel, that is,

\[
(R_n)_{m,m} = (y_m p_0^* - h^{est}_m) (y_m p_0^* - h^{est}_m)^H
\]

Here \( h^{est}_m \) is an averaged or filtered version of the samples \( y_m p_0^* \), for example,

\[
h^{est}_m = \lambda_h (y_m p_0^*) + (1 - \lambda_h) h^{est}_{m,old}
\]

(3.11)

Also less variations in \( R_n \) can be had by applying filtering as below

\[
R_{n,flt} = \lambda_{R_n} (R_n) + (1 - \lambda_{R_n}) R_{n,old}^{flt}
\]

(3.12)
3.2.3 SIR

For both MMSE and ML methods, the demodulated symbol/chip is given by the following equation

$$s_o = \left(R^{-1}\tilde{h}^*\right)\tilde{y}$$

where the matrix $R$ is given by $R'$ if the filter $g$ is computed using the MMSE method and $R_n'$ if the ML method is used.

Given the model in (3.1)

$$\tilde{y} = \tilde{h}s_o + \tilde{n} \quad |s_o| = 1$$

by multiplying the above equation with the filter $g$, we get

$$\left(R^{-1}\tilde{h}^*\right)\tilde{y} = \left(R^{-1}\tilde{h}^*\right)\tilde{h}s_o + \left(R^{-1}\tilde{h}^*\right)\tilde{n}$$  (3.13)

The signal-to-interference ratio is defined as

$$SIR = \frac{\left\|\left(R^{-1}\tilde{h}^*\right)\tilde{h}s_o\right\|^2}{E\left\{\left(R^{-1}\tilde{h}^*\right)\tilde{n}\tilde{n}^*\left(R^{-1}\tilde{h}^*\right)^\dagger\right\}} = \frac{\left\|\left(R^{-1}\tilde{h}^*\right)\tilde{h}\right\|^2}{E\left\{\left(R^{-1}\tilde{h}^*\right)\tilde{n}\tilde{n}^*\left(R^{-1}\tilde{h}^*\right)^\dagger\right\}}$$

For the ML method $R^{-1}R_n = I$ (identity matrix)

$$\Rightarrow SIR = \frac{\left\|\left(R^{-1}\tilde{h}^*\right)\tilde{h}\right\|^2}{\left\|\tilde{h}\right\|^2\left\|\left(R^{-1}\tilde{h}^*\right)^\dagger\tilde{h}\right\|^2} = \left(\left\|\tilde{h}\right\|^2\right)^{-1} = \left(\left\|\left(R^{-1}\tilde{h}^*\right)^\dagger\tilde{h}\right\|^2\right)^{-1}$$

For the MMSE method no such reduction is possible.

In practice, the numerator in $SIR$ is replaced by $\left\|\left(R^{-1}\tilde{h}^*\right)\tilde{y}\right\|^2$. The expectation value of this is

$$\left\|\left(R^{-1}\tilde{h}^*\right)\tilde{y}\right\|^2 + E\left\{\left\|\left(R^{-1}\tilde{h}^*\right)\tilde{n}\right\|^2\right\}$$

$$\Rightarrow SIR_{practical} = SIR + 1$$
Also \( E \left\{ \left\| \left( R^{-1} h^* \right)' \tilde{n} \right\|^2 \right\} \) in the denominator of \( \text{SIR} \) is computed by averaging the square of

\[
\left( R^{-1} h^* \right)' \tilde{y} - \left( R^{-1} h^* \right)' \tilde{h} p = n_{sample}
\]

(3.14)

Where \( p \) are known pilot symbols. From (3.13) we get that the above difference equals \( \left( R^{-1} \tilde{h}^* \right)' \tilde{n} \). Let \( n_{sample,i} \) be a number of collected samples, then an estimate of \( E \left\{ \left\| \left( R^{-1} \tilde{h}^* \right)' \tilde{n} \right\|^2 \right\} \) is given by

\[
\frac{1}{\# \text{Samples}} \sum_{i=1}^{\# \text{Samples}} \left| n_{sample,i} \right|^2
\]

If the ML-method is used and we are confident that the used \( R \) approximates well \( R^* \), then the simplified expression \( \left( R^{-1} \tilde{h}^* \right)' \tilde{h} \) can be used for \( \text{SIR} \).

### 3.3 FFT (Frequency domain)

#### 3.3.1 System Model

Time domain signal \( y(t) \) at the receiver is modeled as below

\[
y(t) = \sum_{p=1}^{P} \sum_{m=\infty}^{\infty} \sum_{l=1}^{L} h_l^{\text{prop}} a_p c_m^{(p)} \rho(t - m\Delta_{\text{chip}} - \tau_l) + \text{noise},
\]

(3.15)

where \( p \) enumerates the physical channels;
\( m \) enumerates the chips;
\( l \) enumerates the propagation paths;
\( h_l^{\text{prop}} \) is a complex valued propagation channel;
\( a_p \) is the signal amplitude;
\( c_m^{(p)} \) is the chip at index \( m \) for \( p^{th} \) channel (composed of channelization code, scrambling code, and the sent symbol);
\( \rho \) is the pulse shaping filter; and
\( \tau_l \) is the propagation path delay for signal path \( l \).

Figure 3.2 shows the different blocks used in the FFT equalizer receiver. Functioning detail about respective blocks is provided in the coming sections.
3.3.2 Demodulation

Sample the received signal at \( t_n = n \Delta_{\text{chip}} \), then

\[
y_n = \sum_{p=1}^{P} \sum_{m=-\infty}^{\infty} a_p h_m^{(n \epsilon)} c_{n-m}^{(p)} + \text{noise} = \sum_{p=1}^{P} \tilde{h}_n^{(n \epsilon)} \ast a_p c_n^{(p)} + \text{noise} \tag{3.16}
\]

where \( \ast \) represents convolution,

\[
h_m^{(n \epsilon)} = h(t_n)_{\text{prop}} \ast \rho(t_n - m \Delta_{\text{chip}} - \tau_i), \text{ where } t_n \text{ is the absolute time.}
\]

**Figure 3.2**: FFT Equalizer Receiver Block Diagram

Fourier transform of \( y_n \) will provide \( Y_n \) which can be described as follows

\[
Y_n = \sum_{p=1}^{P} H_n^{(n \epsilon)} \cdot c_n^{(p)} + \text{noise} \tag{3.17}
\]
where \( H_n^{(net)} \) is the Fourier transform of \( h^{(net)} \) and 
\( C_n^{(p)} \) is the Fourier transform of \( q_p \).

For the MMSE based equalizer, the influence of \( H_n^{(net)} \) is removed to obtain \( X_n \) as shown below in equation (3.18)

\[
X_n = Y_n \cdot \frac{(H_n^{(net)})^*}{|H_n^{(net)}|^2 + \sigma^2}
\]

(3.18)

where \( \sigma^2 \) is the estimate of noise variance and \((H_n^{(net)})^*\) is the complex conjugate of \( H_n^{(net)} \).

Eq. (3.18) refers to the MMSE based equalizer, which is explained next. Now referring back to (3.1): for a signal model

\[
y = h \cdot s + \text{noise}
\]

where ‘s’ is the transmit symbol/chip. Find a filter ‘w’, such that the following expression is minimized:

\[
\min_w E\left\{ |y \cdot w - s|^2 \right\}
\]

\[
= \min_w E\left\{ |h \cdot s \cdot w - s + w \cdot \text{noise}|^2 \right\}
\]

\[
= \min_w \left( |h \cdot s|^2 \cdot |w|^2 + |s|^2 + |w|^2 \cdot \sigma^2 - h \cdot s \cdot w \cdot s^* - (h \cdot s \cdot w)^* \cdot s \right)
\]

\[
= \min_w \left( |h \cdot s|^2 + \sigma^2 \cdot |w|^2 - 2 \text{Re}(h \cdot w) \right)
\]

By solving for ‘w’ gives

\[
w = \frac{h^*}{|h|^2 + \sigma^2}
\]

\[
\rightarrow s = y \cdot w = y \cdot \frac{h^*}{|h|^2 + \sigma^2}
\]

In the simulations, we will use (3.19) for the required channel cancellation operation as follows:
\[ X_n = Y_n \cdot \frac{(H_n^{(net)})^*}{|H_n^{(net)}|^2 + \varepsilon} \]  

(3.19)

where ‘\( \varepsilon \)’ can be called a regularization term which needs to be tuned depending on the propagation channel profile and on the signal-to-noise ratio value at the receiver.

### 3.3.3 Approximations to investigate

The following approximations will be investigated:

1. The Fourier transform needs to be finite, that is, a finite number of samples can only be handled at a time. Investigate optimal (smallest) number of samples for different channels.
2. The amount of filtering needs to be tuned for filtered channel estimate \( h_{filtered,n}^{(net)} \) as shown in (3.23).
3. The amount of filtering needs to be tuned for interference estimation \( \sigma_{filtered,k}^2 \) as shown in (3.22).

### 3.3.4 Interference Calculation

After applying IFFT on \( X_n \) to come back to time domain, we get ‘\( x \)’ which can be modeled as

\[ x_n = \sum_{p=1}^{P} a_p c_n^{(p)} + \text{noise} \]  

(3.19)

Assume \( P = 1 \) for the CPICH. Despreading of ‘\( x \)’ with the scrambling code and then the channelization code of \( P = 1 \) will give

\[ \frac{1}{256} \sum_{n=1}^{256} x_n c_n^{(1)*} = x_{CPICH}^{sample,k} \]  

(3.20)

where ‘\( k \)’ refers to the CPICH symbols received by despreading of the chip samples. Ten CPICH symbols are received from one WCDMA slot.

The received CPICH symbols are filtered as below:

\[ x_{filtered,k}^{CPICH} = (1 - \lambda_n) x_{filtered,k-1}^{CPICH} + \lambda_n x_{sample,k}^{CPICH} \]

The interference is further estimated as,

\[ \left| x_{filtered,k}^{CPICH} - x_{sample,k}^{CPICH} \right|^2 = \sigma_{sample,k}^2 \]  

(3.21)
To get the final $\sigma^2$, filter $\sigma^2_{\text{sample},k}$

$$\sigma^2 = \sigma^2_{\text{filtered},k}, \text{ where}$$

$$\sigma^2_{\text{filtered},k} = (1 - \lambda) \sigma^2_{\text{filtered},k-1} + \lambda \sigma^2_{\text{sample},k} \quad (3.22)$$

### 3.3.5 Power Calculation

To calculate the received signal power, let $p_{\text{est}} = \left| x_{\text{filtered},k}^{\text{CPICH}} \right|^2$. Filter $p_{\text{est}}$ as in (3.22) and call it $P$.

### 3.3.6 Calculation of Filtered Channel Estimate

Put RAKE fingers on all $N_{\text{FFT}}$ samples used in the FFT of $y_n$, despread these fingers using the CPICH channelization code. If the CPICH is for $p = 1$, then

$$h_{\text{sample},n}^{(\text{net})} = \frac{1}{256} \sum_{r=1}^{256} y_{n+r-1} \cdot (c_n^{(1)})^*, \quad r = 1, \ldots, N_{\text{FFT}}$$

where ‘$r$’ is the RAKE finger index and $(c_n^{(1)})^*$ includes the complex conjugate of scrambling code, channelization code for the CPICH, and sent QPSK pilot symbol.

Assume $\tilde{h}_{\text{sample},n}^{(\text{net})} = (h_1^{(\text{net})}, h_2^{(\text{net})}, \ldots, h_{N_{\text{FFT}}}^{(\text{net})})$, and then filter $\tilde{h}_{\text{sample},n}^{(\text{net})}$ as shown in (3.23) to get

$$\tilde{h}_{\text{filtered},n}^{(\text{net})} = (1 - \lambda) \tilde{h}_{\text{filtered},n-1}^{(\text{net})} + \lambda \tilde{h}_{\text{sample},n}^{(\text{net})} \quad (3.23)$$

$\tilde{h}_{\text{filtered},n}^{(\text{net})}$ is also alternatively called as $\tilde{h}_{\text{filtered},n}^{\text{CPICH}}$ in this thesis.
Chapter 4

4 Implementation and Results

4.1 Simulation Setup

Both the methods described in sections 3.2 and 3.3 for channel equalization are implemented in IT++. Standard library features of IT++ are used to construct different blocks of the transmitter, receiver, and the propagation channel. Simulation parameters used for the WCDMA transmit signal are given below.

With reference to figure 2.1, OVSF code tree, the channelization codes used in the simulations are shown in table 4.1.

Table 4.1: Simulation setup: different users/channels with their sf and power assigned

<table>
<thead>
<tr>
<th>USER</th>
<th>Channelization Code</th>
<th>Spreading Factor</th>
<th>Power (Ec/Io)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPICH</td>
<td>$C_{(256,1)}$</td>
<td>256</td>
<td>-10 dB</td>
</tr>
<tr>
<td>Data (desired user)</td>
<td>$C_{(16,2)}$</td>
<td>16</td>
<td>-11 dB</td>
</tr>
<tr>
<td>user1</td>
<td>$C_{(128,4)}$</td>
<td>128</td>
<td>0 dB $^{(1)}$</td>
</tr>
<tr>
<td>user2</td>
<td>$C_{(128,5)}$</td>
<td>128</td>
<td>-2 dB $^{(1)}$</td>
</tr>
<tr>
<td>user3</td>
<td>$C_{(128,6)}$</td>
<td>128</td>
<td>-4 dB $^{(1)}$</td>
</tr>
<tr>
<td>user4</td>
<td>$C_{(128,7)}$</td>
<td>128</td>
<td>-1 dB $^{(1)}$</td>
</tr>
</tbody>
</table>

Powers assigned to the other users, from user1 to user4 as shown in Table 4.1 are relative power values to each other, i.e., after assigning power to CPICH and data, power values for other users are calculated based on these relative power values such that the total power remains one.

$^{(1)}$ The spreading factor and assigned power of the other four users (user1 to user4) are based on the reference table “Table C.13A: OCNS definition for HSDPA receiver testing”, 3GPP TS 25.101 V8.4.0 (2008-09).
4.1.1 Transmit Signal from Node B

Figure 4.2 shows the simulation flow to construct a WCDMA transmit signal at Node B which carries information for all the users/channels mentioned in table 4.1. The information for different users/channels is spread with the use of their respective channelization codes. Scrambling code used at the BS (Node B) is marked as ‘sc1’ which is uniquely assigned to the BS. And any other scrambling code may be employed for transmission from other BS.

CPICH uses all zero bits, whereas, random bits are generated for other users as well as for the desired data user. QPSK is used as the modulation scheme for both for the data users and for CPICH.

Half slot processing is employed, i.e., 1280 chips are processed at one time. Same channelization codes are used each time to spread the different data symbols, while scrambling code is repeated after every 38400 chips. Spreading factor of 16 for the data user implies that 1280/16 = 80 QPSK data symbols can be transmitted through each half slot. Similarly, 5 CPICH symbols and 10 other data user’s symbols are transmitted at one time.

Before scrambling, each user/channel signal is multiplied with its respective amplitude. The signal amplitudes are calculated based on the signal power values mentioned in table 4.1 and the overall signal energy is kept as unity.

Root raised cosine (RRC) filter with roll-off factor = 0.22, over sampling rate = 4, and filter length = 14 chips is used. Up-sampling by a factor 4 is done which provides 5120 signal values after the transmit filter.

![Root Raised Cosine Filter](image)

**Figure 4.1:** Root Raised Cosine Filter; sample rate = 4 per chip, filter length = 14 chips
Figure 4.2: Transmission of WCDMA signal from Node B
4.1.2 Propagation Channel

The propagation channel model used in the simulations is a two path non-fading (static) channel. The power is divided equally between the two paths. Different values of delay spread are simulated; and the results are shown in section 4.4 and onwards. Single path channel model and ITU-Pedestrian B with speed 3km/hr (PB3) which is a slow fading channel; are also used in the simulations.

4.1.3 Signal Reception at UE

The up-sampled signal values are received at UE through a propagation channel. The receiver functions are shown in figure 4.3. WCDMA signal from the desired BS as well as from another nearby BS, which is using a different scrambling code, is received at MS through a wireless channel.

Additive White Gaussian Noise (AWGN) is added to the received signal. Equalizer performances on different values of noise variance are tested by various simulations. Also without AWGN, equalizer performance is observed. Frequency errors are generated at the UE due to any phase offset or synchronization loss between the clocks of the BS and MS. Receiver performance with and without frequency errors are then tested. Analog-to-Digital converter (A/D) is used to digitize the received signal before filtering operation. Equalizers performance with different (A/D) converters are simulated and shown in the results section 4.4.

Received signal is finally fed to the receiver filter which is also a root raised cosine (RRC) filter matched to the transmit filter. Matched filter output is then used for both equalizers.
Figure 4.3: Received Signal at UE via Matched Filtering
4.2 GRAKE Equalizer Receiver

4.2.1 Channel Estimation

Received signal values after the receiver filter are used to detect the channel response to the transmit signal. In figure 4.4, different blocks are shown which are used in the simulator to detect the channel response on different signals received via multiple rake fingers.

\[ \tau = \frac{1}{4} \text{ chip delay;} \]
\[ (2m+1+ds) = \# \text{ of fingers;} \]
\[ ds = \text{delay\_spread} / \tau \]

Figure 4.4: Non-Parametric G-RAKE Receiver Channel Estimation
GRAKE uses \((2m+1+ds)\) fingers which are separated by a fixed time delay between any two rake fingers, i.e., \(\tau\) equals to \(\frac{1}{4}\) chip delay. Different delayed versions of the WCDMA signal are received via these fingers. Depending on the \(R_s\) \& \(T_s\) filter length, the parameter \(m\) is selected as \(20 = 15/\tau\), to collect all the energy stored in the filter taps; while \(ds\) is the delay spread (time difference between LOS signal path and the last significant signal energy path) divided by the finger spacing \((\tau)\). In this thesis \(ds\) or path delay will be referring to the delay (in \# of chips) between two equal energy signal paths. RAKE finger placed at \((0.\tau)\) implies zero delay is introduced and direct despreading is used. For single path test cases, nine fingers with fifth finger at the zero delay position are used in the test simulations (nine fingers are considered to be optimum number in this case).

Downsampling is performed on each rake finger as there is upsampling done at the transmitter side before transmission. Different offset positions are used as shown in the figure 4.3 to simulate the GRAKE equalizer performance for the time offset received data. Descrambling and despreading operations are implemented to recover the original sent symbols. Desired scrambling code ‘sc1’ for the BS scrambling and ‘cc_CPICH’ is used for the CPICH channel. Since the CPICH symbols are known at the UE, dividing with the known symbols return the channels estimate values on the respective rake fingers. Since five CPICH symbols are received in one half slot, averaging filter is implemented to filter the sampled channel estimates. The filtered channel estimate is then used for the estimation of noise covariance matrix and further the G-RAKE equalizer filter weights.

### 4.2.2 Equalizer Filter and SIR

Based on the sampled channel values and the filtered estimate, noise covariance matrix is calculated as shown in figure 4.5. Further, filtered channel estimate and filtered noise covariance matrix are used to determine the equalizer filter weights and an estimate of Signal-to-Interference ratio. Noise covariance matrix inversion is implemented by using standard built-in functions available in IT++ for calculating matrix inverse.
4.2.3 Combiner

Different delayed version of the received WCDMA signal are despread as is done in the GRAKE channel estimation, but now with the desired user channelization code, i.e., cc_data. Filter weights calculated above are now applied on the despread data symbols obtained from different rake fingers. Combiner operations are shown in figure 4.6.

Following the combiner output, QPSK demodulation is performed to recover the sent data bits. Received data bits are then compared with the original sent bits to calculate the bit error rate (BER).
\[ \tau = \frac{3}{4} \text{ chip delay; } \]

\[ 2m+1+ds = \# \text{ of fingers; } \]

\[ ds = (\text{delay\_spread} / \tau) \]

**Figure 4.6:** Non-Parametric G-RAKE Combiner Output and BER
4.3 FFT Equalizer Receiver

4.3.1 Channel Estimation

In the FFT receiver, the received signal is firstly down-sampled to over-sampling 1, that is, one sample per chip, see section 3.3.2. Multiple despreaders are then utilized to calculate the channel sample values as shown in figure 4.7. In FFT case, the despreaders are always

\[ \tau = 1 \text{ chip delay}; \]
\[ 2k+1+ds = \# \text{ of fingers}; \]
\[ ds = \text{ delay spread}. \]

\[ h_0 \]
\[ h_{k+1} \]
\[ h_k \]
\[ h_{k+ds} \]

**Figure 4.7:** FFT Receiver Channel Estimation
separated by one chip delay while descrambling and despreading operations to get the channel sample values are the same as performed in the GRAKE channel estimation.

Based on the \( R_s, T_r \) radio filters which are used as standard root raised cosine filters with filter length equals to 14 chips as shown in figure 4.1, the parameter ‘\( k \)’, i.e., the number of despreaders used in the left of the main peak is chosen as 15; while the number of despreaders towards the right of the main peak are chosen as \((15+ds)\). For single path test cases, \( ds = 0 \) is used. In that case, 31 despreaders are used to calculate the channel estimate, 15 despreaders on both sides of the zero delay despreader.

### 4.3.2 Equalizer

Equalization operation is performed in the frequency domain. Both the channel estimates values and received down sampled signal is converted to frequency domain by taking Fast Fourier Transform (FFT). FFT function is used as a standard IT++ library function. Figure 4.8 shows the method used in the FFT equalization operation.

The channel equalization is performed on each half slot received. The channel estimate vector used in the FFT equalizer has 1280 chips with the channel estimate values obtained by \((2k+1+ds)\) despreaders in the centre and rest of the values as zeros.

A parameter tuning block as shown in figure 4.8 is also used in the FFT equalizer method for parameters tuning. An output of the intelligent block, ‘\( \varepsilon \)’ is required in the channel equalization operation. The value of ‘\( \varepsilon \)’ depends on the values of IorIoc, delay spread, and the size of the A/D used. Optimal values of ‘\( \varepsilon \)’ are found for different values of IorIoc and delay spread. The optimal values of ‘\( \varepsilon \)’ needs to be scaled with the same A/D constant (K) used for multiplying the signal before quantization. Different A/D sizes and their respective A/D constant (K) values are used in the simulations.

As the equalization is performed in the frequency domain, so an inverse Fourier Transform (IFFT) is used to convert the equalized signal back to the time domain signal. In the figure, it is also shown that the number of chips that can be used further for despreading and symbol detection is decided by the parameter tuning block’s output ‘\( M \)’. The value of \( M \) depends on the delay spread and target BER. For small delay spread channels, more chips are equalized as compared to large delay spread channels. More details about the number of output chip samples are given in section 4.4.6.1.
4.3.3 Despreeder Output, BER, and SIR Estimate

Two despreaders are shown in figure 4.9, which are used to receive the CPICH symbols and desired data user symbols. Received data symbols are QPSK demodulated to recover the data bits and original sent bits are compared with the received bits to calculate the BER.

As there are five CPICH symbols received in one half slot, received CPICH symbols are filtered by using a first order recursive filter. The filtered CPICH symbol is used to calculate the signal power; while interference is estimated from the filtered CPICH symbol and sampled value of the CPICH symbol. Both these values are used to estimate the SIR.

Two other despreaders are also used in the simulations, though they are not shown in the figure. They are placed at one chip offset on both sides of the main peak CPICH despreader used for CPICH symbols. The estimated SIR of these three despreaders is then used to check if down sampling has been done at some time offset other than zero at the receiver or not.
Figure 4.9: FFT Despreader Output, BER, and SIR Estimate
4.4 Results and Discussion

4.4.1 Parameters Optimization

4.4.1.1 GRAKE Parameters Optimization

Two averaging filters are implemented in the G-RAKE method. The averaging filter which is used to filter the channel estimate results uses $\lambda_h$; and the second filter which is filtering the noise covariance matrix uses $\lambda_{R_{n}}$ as the filtering parameters as shown in (3.11) and (3.12) respectively.

Simulations are performed for different channels (single path and two path non-fading or static channels) and using different Ioorloc values. For single path channel, nine fingers are used to receive the WCDMA signal. And for two path channel, number of fingers used to receive the WCDMA signal is calculated as $(2m+1+ds)$, as discussed in section 4.2. Delay spread ‘ds’ = 5 chip is used for two path channel simulations.

Firstly, for an arbitrary value of $\lambda_h$ equals to 1/200 for our case, multiple simulations are performed for Ioorloc values starting from -4 dB and going up till 28 dB. For each Ioorloc value, a range of values for $\lambda_{R_{n}}$ from 1/25 to 1/500 are simulated. From the simulation results, BER performance of the data user is observed. Based on these results, optimum (largest) value for $\lambda_{R_{n}}$ is selected to be 1/200, for all the Ioorloc values as well as for the two main test cases, i.e., single path and two path channel with ‘ds’ = 5 chip. A large value of $\lambda_{R_{n}}$ is used to give more weightage to the noise samples received in the current WCDMA slot.

Secondly, using the optimized value of $\lambda_{R_{n}} = 1/200$, another set of simulations are performed for a range of values of $\lambda_h$ ranging from 1/25 to 1/500. Though, no significant influence on the BER performance is observed in these simulations for the different values of $\lambda_h$. One optimum value selected for $\lambda_h$ is found as 1/100.

4.4.1.2 FFT Parameters Optimization

The FFT method for the frequency domain equalization is used by first converting the time domain signal to frequency domain and then the equalized signal is converted back to time
domain for despreading. To implement these operations, Fast Fourier Transform (FFT) and Inverse Fast Fourier Transform (IFFT) are used in the simulations. So, possible FFT window size is first determined which can be used for this equalization method.

Multiple simulations are performed with FFT window sizes: 32, 64, 128, 256, 512, 1024, and 2048. Two path static channel model with delay spread = 1, 5, 10, and 70 chips is used. From BER results of the desired data user, noise floor without AWGN is observed. Table 4.2 shows the BER results for a two path channel with delay spread = 5 chip, using different FFT window sizes and for different values of $\varepsilon$. In these simulations, filtering parameter $\lambda_h = 1/200$ is used to filter the channel sample values to get a filtered channel estimate. Different simulation results are obtained using large delay spread values, such as, $ds = 10$ and 70 chip. The obtained results show that FFT window size = 2048 is to be selected, because, with bigger FFT window size larger delay spread can be equalized better as compared to smaller FFT window size.

Later, in all the FFT method simulations $N_{FFT} = 2048$ is opted to use for the FFT equalizer. FFT equalizer performance results with $\lambda_h = 1/200$, $N_{FFT} = 2048$, without AWGN, and for different delay spread values is shown in table 4.3. Without AWGN, optimum values of $\varepsilon$ are selected for different delay spread values, e.g., $\varepsilon$ is chosen as 1/2000 for ‘ds’ = 1 chip and 1/500 for all other delay spread values. $\varepsilon = 0$ is found to be optimum value for single path scenario as it gives BER = 0, if there is no AWGN.

By selecting appropriate values of $\varepsilon$ mentioned above, then optimized values for the filtering parameter $\lambda_h$ is determined for these different propagation channels. Now by using different IorIoc values from -4 dB to 28 dB, the BER performance results are obtained for the same set of delay spread values (1, 5, 10, and 70 chip) as well for the single path channel. It is found that $\lambda_h = 1/200$ is to be used for all the IorIoc values and for all the channels models (single path, two path with different delay spreads). Also, the filtering parameter $\lambda_{CPICH}$, used for the receive CPICH symbols filtering, is assigned the same value as is obtained for $\lambda_h$, i.e., $\lambda_{CPICH} = 1/200$ is used in the simulations estimating SIR values.
Table 4.2: \( N_{\text{FFT}} \) vs BER for two path channel, ‘ds’ = 5 chip, \( \lambda_{\mu} = 1/\mu \), without AWGN

<table>
<thead>
<tr>
<th>Sr. #</th>
<th>( 1 / \varepsilon )</th>
<th>BER vs ( N_{\text{FFT}} ) (FFT window size)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>N_{\text{FFT}} = 64</td>
<td>N_{\text{FFT}} = 128</td>
</tr>
<tr>
<td>1</td>
<td>100</td>
<td>1.8E-03</td>
</tr>
<tr>
<td>2</td>
<td>200</td>
<td>9.3E-04</td>
</tr>
<tr>
<td>3</td>
<td>300</td>
<td>1.1E-03</td>
</tr>
<tr>
<td>4</td>
<td>400</td>
<td>1.8E-03</td>
</tr>
<tr>
<td>5</td>
<td><strong>500</strong></td>
<td>3.1E-03</td>
</tr>
<tr>
<td>6</td>
<td>600</td>
<td>4.6E-03</td>
</tr>
<tr>
<td>7</td>
<td>700</td>
<td>6.4E-03</td>
</tr>
<tr>
<td>8</td>
<td>800</td>
<td>8.3E-03</td>
</tr>
<tr>
<td>9</td>
<td>900</td>
<td>1.0E-02</td>
</tr>
<tr>
<td>10</td>
<td>1000</td>
<td>1.3E-02</td>
</tr>
<tr>
<td>11</td>
<td>1100</td>
<td>1.5E-02</td>
</tr>
<tr>
<td>12</td>
<td>1200</td>
<td>1.7E-02</td>
</tr>
<tr>
<td>13</td>
<td>1300</td>
<td>1.9E-02</td>
</tr>
<tr>
<td>14</td>
<td>1400</td>
<td>2.1E-02</td>
</tr>
<tr>
<td>15</td>
<td>1500</td>
<td>2.3E-02</td>
</tr>
<tr>
<td>16</td>
<td>1600</td>
<td>2.5E-02</td>
</tr>
<tr>
<td>17</td>
<td>1700</td>
<td>2.7E-02</td>
</tr>
<tr>
<td>18</td>
<td>1800</td>
<td>2.9E-02</td>
</tr>
<tr>
<td>19</td>
<td>1900</td>
<td>3.1E-02</td>
</tr>
<tr>
<td>20</td>
<td>2000</td>
<td>3.3E-02</td>
</tr>
</tbody>
</table>

One last optimization parameter for FFT method is \( \lambda_{\sigma} \) which is used to filter the interference estimate values. \( \lambda_{\sigma} \) is determined using the same set of IorIoc values (-4dB to 28dB) and using the same single path and two path channel models. For tuning \( \lambda_{\sigma} \), variance of the SIR estimates of different WCDMA half slots (approx. 1200 half slots) is optimized. From the simulation results, i.e., variance of the SIR estimates, \( \lambda_{\sigma} = 1/100 \) is found as the optimized value.
**Table 4.3:** FFT Equalizer BER results for different delay spread values, \( N_{\text{FFT}} = 2048 \), without AWGN, \( \lambda_h = 1/200 \),

<table>
<thead>
<tr>
<th>Sr. #</th>
<th>( 1/\varepsilon )</th>
<th>BER vs Delay Spread</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1 chip</td>
</tr>
<tr>
<td>1</td>
<td>100</td>
<td>0.603%</td>
</tr>
<tr>
<td>2</td>
<td>200</td>
<td>0.312%</td>
</tr>
<tr>
<td>3</td>
<td>300</td>
<td>0.200%</td>
</tr>
<tr>
<td>4</td>
<td>400</td>
<td>0.141%</td>
</tr>
<tr>
<td>5</td>
<td>500</td>
<td>0.101%</td>
</tr>
<tr>
<td>6</td>
<td>600</td>
<td>0.076%</td>
</tr>
<tr>
<td>7</td>
<td>700</td>
<td>0.060%</td>
</tr>
<tr>
<td>8</td>
<td>800</td>
<td>0.047%</td>
</tr>
<tr>
<td>9</td>
<td>900</td>
<td>0.035%</td>
</tr>
<tr>
<td>10</td>
<td>1000</td>
<td>0.028%</td>
</tr>
<tr>
<td>11</td>
<td>1100</td>
<td>0.021%</td>
</tr>
<tr>
<td>12</td>
<td>1200</td>
<td>0.018%</td>
</tr>
<tr>
<td>13</td>
<td>1300</td>
<td>0.015%</td>
</tr>
<tr>
<td>14</td>
<td>1400</td>
<td>0.012%</td>
</tr>
<tr>
<td>15</td>
<td>1500</td>
<td>0.010%</td>
</tr>
<tr>
<td>16</td>
<td>1600</td>
<td>0.008%</td>
</tr>
<tr>
<td>17</td>
<td>1700</td>
<td>0.007%</td>
</tr>
<tr>
<td>18</td>
<td>1800</td>
<td>0.006%</td>
</tr>
<tr>
<td>19</td>
<td>1900</td>
<td>0.006%</td>
</tr>
<tr>
<td>20</td>
<td>2000</td>
<td><strong>0.005%</strong></td>
</tr>
</tbody>
</table>

### 4.4.2 Chip Offset Comparison

Received signal is down sampled to chip rate with different offset positions, i.e., 0 chip, 1/4 chip, and 1/2 chip offset w.r.t the peak sampling point of the matched filter output. Downsampling is performed directly after the matched filtering in case of FFT equalizer and is done after the different delay elements of RAKE fingers in case of GRAKE equalizer.

BER performance results for the two equalization methods for target BER of 10\%, 1\%, and 0.1\% are shown below with different channel models used.

- **Propagation channel:** “Single Path (non-time selective)”
  - FFT Simulation Parameters: \( N_{\text{FFT}}=2048 \), \( \lambda_h = 1/200 \), \( \lambda_\sigma = 1/100 \), optimum value for \( \varepsilon \), and channel estimation despreading = 31.
  - G-RAKE Simulation Parameters: no_fingers = 9, \( \lambda_h = 1/100 \), \( \lambda_{R_\varepsilon} = 1/200 \)
Table 4.4: Chip Offset Comparison: Single Path Channel

<table>
<thead>
<tr>
<th>Chip Offset</th>
<th>Target BER</th>
<th>FFT</th>
<th>GRAKE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>IorIoc [dB]</td>
<td>SIR_l</td>
</tr>
<tr>
<td>0 chip</td>
<td>10%</td>
<td>1.03</td>
<td>0.001</td>
</tr>
<tr>
<td></td>
<td>1%</td>
<td>6.30</td>
<td>0.001</td>
</tr>
<tr>
<td></td>
<td>0.1%</td>
<td>8.78</td>
<td>0.001</td>
</tr>
<tr>
<td>1/4 chip</td>
<td>10%</td>
<td>1.35</td>
<td>0.006</td>
</tr>
<tr>
<td></td>
<td>1%</td>
<td>6.60</td>
<td>0.002</td>
</tr>
<tr>
<td></td>
<td>0.1%</td>
<td>9.05</td>
<td>0.002</td>
</tr>
<tr>
<td>1/2 chip</td>
<td>10%</td>
<td>2.02</td>
<td>0.133</td>
</tr>
<tr>
<td></td>
<td>1%</td>
<td>8.24</td>
<td>0.063</td>
</tr>
<tr>
<td></td>
<td>0.1%</td>
<td>12.38</td>
<td>0.037</td>
</tr>
</tbody>
</table>

In the results, along with the required IorIoc values, SIR estimate (mean) results are also shown for the FFT method. Using three despreaders, one for main peak as shown in figure 4.9 and two others at 1 chip offset in the left and right position of the main despreader are used. SIR_c shows the SIR estimate using the main despreader, where SIR_l and SIR_r show SIR estimates of the left and right despreader respectively. From the SIR estimate results, it can be noted that SIR values close to zero are observed for the left and right despreaders if the downsampling has been done at 0 chip offset.

The above results show that both the methods have almost the same performance for single path channel only if the downsampling has been performed properly after the matched filtering. Also the GRAKE method is independent of any offset induced by the downsampling, but the FFT method performance is deteriorated with different chip offset positions. Furthermore the FFT method gives the worst performance if the offset is 1/2 chip.

**Conclusion:**

When a propagation channel is AWGN like, i.e., having very small (≈ zero) delay spread, the timing of the main peak needs to be determined accurately. This can be accomplished by putting despreader on a potential peak and its ±3/4 chip neighbors. Measuring the estimated SIR values and fitting to the response anticipated for an AWGN channel will locate accurately the peak.
Propagation channel: “Two Path (non-time selective), Delay Spread = 0.75 chip”

For all the two path channel models, the following simulation parameters are used:

FFT Simulation Parameters: \( N_{ FFT } = 2048, \ \lambda_s = 1/200, \ \lambda_c = 1/100, \ \) optimum value for \( \varepsilon \),
channel estimation despreaders = \( 15 + \text{delay} \_\text{spread} + 15 \).

G-RAKE Simulation Parameters: finger spacing ‘\( \tau \)’ = \( 3/4 \), \( h_{\lambda} = 1/100 \), \( \sigma_{\lambda} = 1/200 \),
\( \text{no} \_\text{fingers} = (15 + \text{delay} \_\text{spread} + 15) / \tau \)

Simulation Length: 1000 WCDMA slots

Selecting ‘\( \varepsilon \)’ for FFT method

Table 4.5 shows the simulation results of FFT equalizer using IorIoc = 6 dB and different values of \( \varepsilon \). Propagation channel used is two path (non-fading channel) with delay spread = 0.75 chip. These results show that optimum value of \( \varepsilon \) is needed for the FFT equalizer to minimize the BER at the required IorIoc.

Table 4.5: FFT equalizer BER vs \( \varepsilon \), IorIoc = 6 dB, two path channel ‘ds’ = 0.75 chip

<table>
<thead>
<tr>
<th>Sr. #</th>
<th>1 / ( \varepsilon )</th>
<th>BER vs Chip Offset</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0 chip</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>3.12%</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>2.60%</td>
</tr>
<tr>
<td>3</td>
<td>15</td>
<td>2.39%</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>2.24%</td>
</tr>
<tr>
<td>5</td>
<td>25</td>
<td>2.19%</td>
</tr>
<tr>
<td>6</td>
<td>30</td>
<td>2.13%</td>
</tr>
<tr>
<td>7</td>
<td>35</td>
<td>2.08%</td>
</tr>
<tr>
<td>8</td>
<td>40</td>
<td>2.07%</td>
</tr>
<tr>
<td>9</td>
<td>45</td>
<td>2.09%</td>
</tr>
<tr>
<td>10</td>
<td>50</td>
<td>2.11%</td>
</tr>
<tr>
<td>11</td>
<td>55</td>
<td>2.14%</td>
</tr>
<tr>
<td>12</td>
<td>60</td>
<td>2.14%</td>
</tr>
<tr>
<td>13</td>
<td>65</td>
<td>2.18%</td>
</tr>
<tr>
<td>14</td>
<td>70</td>
<td>2.19%</td>
</tr>
<tr>
<td>15</td>
<td>75</td>
<td>2.22%</td>
</tr>
<tr>
<td>16</td>
<td>80</td>
<td>2.30%</td>
</tr>
<tr>
<td>17</td>
<td>85</td>
<td>2.23%</td>
</tr>
<tr>
<td>18</td>
<td>90</td>
<td>2.21%</td>
</tr>
<tr>
<td>19</td>
<td>95</td>
<td>2.34%</td>
</tr>
<tr>
<td>20</td>
<td>100</td>
<td>2.35%</td>
</tr>
</tbody>
</table>
Simulation results for the target BER of 10%, 1%, and 0.1% using two path channel model with delay spread = 0.75 chip are given in table 4.6.

**Table 4.6: Chip Offset Comparison: Two Path Channel ‘ds’ = 0.75 Chip**

<table>
<thead>
<tr>
<th>Chip Offset</th>
<th>Target BER</th>
<th>FFT</th>
<th>GRAKE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>IorIoc [dB]</td>
<td>SIR_1</td>
</tr>
<tr>
<td>0 chip</td>
<td>10%</td>
<td>0.97</td>
<td>0.727</td>
</tr>
<tr>
<td></td>
<td>1%</td>
<td>7.59</td>
<td>0.228</td>
</tr>
<tr>
<td></td>
<td>0.1%</td>
<td>11.13</td>
<td>0.079</td>
</tr>
<tr>
<td>1/4 chip</td>
<td>10%</td>
<td>0.62</td>
<td>0.373</td>
</tr>
<tr>
<td></td>
<td>1%</td>
<td>6.38</td>
<td>0.084</td>
</tr>
<tr>
<td></td>
<td>0.1%</td>
<td>9.13</td>
<td>0.041</td>
</tr>
<tr>
<td>1/2 chip</td>
<td>10%</td>
<td>0.60</td>
<td>0.427</td>
</tr>
<tr>
<td></td>
<td>1%</td>
<td>6.42</td>
<td>0.073</td>
</tr>
<tr>
<td></td>
<td>0.1%</td>
<td>9.21</td>
<td>0.066</td>
</tr>
</tbody>
</table>

These results show that the GRAKE equalizer has consistent performance with the three different offset positions in downsampling. For the FFT equalizer, downsampling at offset = 1/4 or 1/2 gives the same performance and is also better than the case when offset = 0. The FFT equalizer’s performance for these two cases, i.e., offset = 1/2 & 1/4 is also better than the GRAKE equalizer’s performance.

In FFT equalizer, for offset = 0, all the chip samples of the received signal experience more interference from the second path as compared to the case when offset = 1/4 is used. That is why we have observed better performance with chip offset = 1/4 with FFT equalizer. And for the case when offset = 1/2 is used, the second path appears to be stronger than the first path in the channel estimates; while the interference intensity remains the same as with offset = 1/4 was observed.

**Table 4.7: FFT equalizer BER results at different offset positions using IorIoc = 8 dB**

<table>
<thead>
<tr>
<th>Chip Offset</th>
<th>BER</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.81%</td>
</tr>
<tr>
<td>1/4</td>
<td>0.35%</td>
</tr>
<tr>
<td>1/2</td>
<td>0.34%</td>
</tr>
<tr>
<td>1</td>
<td>0.80%</td>
</tr>
<tr>
<td>5/4</td>
<td>0.35%</td>
</tr>
</tbody>
</table>
Conclusion:

For non-AWGN like channels (having small but non-zero delay spread), it is not obvious that alignment with one of the peaks gives the best BER results. Possible solutions involve computing two equalizer solutions for different offsets and measure SIR or time multiplex the equalizer with different offsets and selecting the best one. In the latter solution, we periodically would have to check different offsets to know which offset is optimal.

➢ Propagation channel: “Two Path (non-time selective), Delay Spread = 1 chip”

Simulation results for the target BER of 10%, 1%, and 0.1% for the two methods (FFT and GRAKE) are given below:

Table 4.8: Chip Offset Comparison: Two Path Channel ‘ds’ = 1 Chip

<table>
<thead>
<tr>
<th>Chip Offset</th>
<th>Target BER</th>
<th>FFT</th>
<th>GRAKE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>IorIoc [dB]</td>
<td>SIR_1</td>
</tr>
<tr>
<td>0 chip</td>
<td>10%</td>
<td>3.08</td>
<td>1.697</td>
</tr>
<tr>
<td></td>
<td>1%</td>
<td>12.93</td>
<td>0.292</td>
</tr>
<tr>
<td></td>
<td>0.1%</td>
<td>20.09</td>
<td>0.101</td>
</tr>
<tr>
<td>1/4 chip</td>
<td>10%</td>
<td>3.23</td>
<td>1.463</td>
</tr>
<tr>
<td></td>
<td>1%</td>
<td>13.67</td>
<td>0.357</td>
</tr>
<tr>
<td></td>
<td>0.1%</td>
<td>22.58</td>
<td>0.055</td>
</tr>
<tr>
<td>1/2 chip</td>
<td>10%</td>
<td>3.25</td>
<td>1.782</td>
</tr>
<tr>
<td></td>
<td>1%</td>
<td>14.95</td>
<td>0.550</td>
</tr>
<tr>
<td></td>
<td>0.1%</td>
<td>28.0</td>
<td>0.089</td>
</tr>
</tbody>
</table>

The above results for two path channel with delay spread = 1 chip show that the BER performance for FFT method is superior to the GRAKE method, if proper downsampling has been done before the equalization. Also comparable performance is observed if the downsampling is performed at offset = 1/4 chip. The FFT method’s performance becomes worse than the GRAKE performance if offset = 1/2 is used. These results also show that the FFT method performs better than the GRAKE in all the cases if the target BER is 10%.
Propagation channel: “Two Path (non-fading), Delay Spread = 4.5 chip”

Simulation results for two path channel with delay spread = 4.5 chip are shown in table 4.9.

**Table 4.9: Chip Offset Comparison: Two Path Channel ‘ds’ = 4.5 Chip**

<table>
<thead>
<tr>
<th>Chip Offset</th>
<th>Target BER</th>
<th>FFT</th>
<th>GRAKE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>IorIoc [dB]</td>
<td>SIR_c</td>
</tr>
<tr>
<td>0 chip</td>
<td>10%</td>
<td>3.01</td>
<td>0.005</td>
</tr>
<tr>
<td></td>
<td>1%</td>
<td>11.28</td>
<td>0.007</td>
</tr>
<tr>
<td></td>
<td>0.1%</td>
<td>16.64</td>
<td>0.004</td>
</tr>
<tr>
<td>1/4 chip</td>
<td>10%</td>
<td>2.89</td>
<td>0.014</td>
</tr>
<tr>
<td></td>
<td>1%</td>
<td>11.08</td>
<td>0.009</td>
</tr>
<tr>
<td></td>
<td>0.1%</td>
<td>16.53</td>
<td>0.004</td>
</tr>
<tr>
<td>1/2 chip</td>
<td>10%</td>
<td>3.07</td>
<td>0.006</td>
</tr>
<tr>
<td></td>
<td>1%</td>
<td>11.24</td>
<td>0.005</td>
</tr>
<tr>
<td></td>
<td>0.1%</td>
<td>16.60</td>
<td>0.003</td>
</tr>
</tbody>
</table>

The above results show that no significant performance degradation is observed at different chip offset positions in the downsampling for the two methods and consistent results are obtained. Also, the results illustrate that the FFT method has performed much better than the GRAKE equalizer for the channel model used. Furthermore, based on the above results it is clear that to achieve a target BER of 0.1%, FFT method requires about an IorIoc = 16.6 dB. On the other hand, IorIoc value more than 28 dB is required for the GRAKE equalizer.
Propagation channel: "Two Path (non-fading), Delay Spread = 5 chip"

Table 4.10: Chip Offset Comparison: Two Path Channel ‘ds’ = 5 Chip

<table>
<thead>
<tr>
<th>Chip Offset</th>
<th>Target BER</th>
<th>FFT</th>
<th>GRAKE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>IorIoc [dB]</td>
<td>SIR_c</td>
</tr>
<tr>
<td>0 chip</td>
<td></td>
<td>3.21</td>
<td>0.001</td>
</tr>
<tr>
<td></td>
<td>10%</td>
<td>11.89</td>
<td>0.002</td>
</tr>
<tr>
<td></td>
<td>1%</td>
<td>17.22</td>
<td>0.001</td>
</tr>
<tr>
<td>1/4 chip</td>
<td></td>
<td>3.46</td>
<td>0.008</td>
</tr>
<tr>
<td></td>
<td>10%</td>
<td>12.32</td>
<td>0.004</td>
</tr>
<tr>
<td></td>
<td>1%</td>
<td>17.65</td>
<td>0.003</td>
</tr>
<tr>
<td>1/2 chip</td>
<td></td>
<td>3.92</td>
<td>0.042</td>
</tr>
<tr>
<td></td>
<td>10%</td>
<td>13.91</td>
<td>0.031</td>
</tr>
<tr>
<td></td>
<td>1%</td>
<td>21.16</td>
<td>0.022</td>
</tr>
</tbody>
</table>

For a delay spread = 5 chip, simulation results using the FFT method and NP-GRAKE method are presented above in table 4.10. These results illustrate that the FFT equalizer can provide better performance than the GRAKE equalizer. Again, it is shown in the results that the best performance for FFT equalizer is only achieved if the downsampling has been performed appropriately at the receiver after the matched filtering.
4.4.3 Frequency Error Comparison

If the clocks of UE and Node B are not synchronized, then there will be a phase offset generated in the UE clock. This residual phase offset accumulates with time and causes phase rotation in the received signal. To simulate this problem, and to observe the performance of the time domain (NP-GRAKE) and frequency domain (FFT) equalizers, simulations are performed for different frequency errors $= 5$, $10$, $15$, and $20$ Hz. BER performance for the case with frequency error $= 0$ Hz is also shown for reference.

For the case of frequency errors at the UE, fast filtering, i.e., higher value of $\lambda_h$ is required for the filtered channel estimate as used in (3.11) and (3.23). Multiple simulations are performed and on the basis of obtained results, it is found that $\lambda_h = 1/40$ (i.e., about 4 WCDMA slots) could be an appropriate value to filter the channel estimates, in case of frequency errors. Other simulation parameters used in the simulations are the same. Simulations results for different channel models using $\lambda_h = 1/40$ and $\lambda_h = 1/200$ are given below for the above mentioned set of frequency errors.

➢ **Propagation channel: “Single Path (non-fading), Offset = 0 chip”**

Table 4.11: Frequency Error Comparison: Single Path Channel, Offset = 0 Chip

<table>
<thead>
<tr>
<th>Frequency Errors</th>
<th>Target BER</th>
<th>$\lambda_h = 1/200$</th>
<th>$\lambda_h = 1/40$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>FFT</td>
<td>IorIoc [dB]</td>
</tr>
<tr>
<td>0 Hz</td>
<td>10%</td>
<td>1.06</td>
<td>1.11</td>
</tr>
<tr>
<td></td>
<td>1%</td>
<td>6.29</td>
<td>6.27</td>
</tr>
<tr>
<td></td>
<td>0.1%</td>
<td>8.84</td>
<td>8.76</td>
</tr>
<tr>
<td>5 Hz</td>
<td>10%</td>
<td>3.33</td>
<td>3.84</td>
</tr>
<tr>
<td></td>
<td>1%</td>
<td>10.62</td>
<td>13.75</td>
</tr>
<tr>
<td></td>
<td>0.1%</td>
<td>13.80</td>
<td>-</td>
</tr>
<tr>
<td>10 Hz</td>
<td>10%</td>
<td>15.06</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>1%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>0.1%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>20 Hz</td>
<td>10%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>1%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>0.1%</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
For the given single path channel at offset = 0 chip, simulation results show that both the methods can perform sufficiently well up till 10 Hz with a slightly better performance results from GRAKE as compared to the FFT method. For 20 Hz case, FFT method performs better than the GRAKE method. Also the FFT method works in a reasonable IorLoc limits for a target BER of 0.1% while the required IorLoc becomes too high for the GRAKE method.

**Propagation channel: “Single Path (non-time selective), Offset = 1/2 chip”**

Observed results for different frequency errors using the single path channel, and downsampling performed at chip offset = 1/2 chip are shown below:

**Table 4.12: Frequency Error Comparison: Single Path Channel, Offset = 1/2 Chip**

<table>
<thead>
<tr>
<th>Frequency Errors</th>
<th>Target BER</th>
<th>( \lambda_n = 1/200 )</th>
<th>( \lambda_n = 1/40 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FFT</td>
<td>GRAKE</td>
<td>FFT</td>
</tr>
<tr>
<td></td>
<td>IorLoc [dB]</td>
<td>IorLoc [dB]</td>
<td>IorLoc [dB]</td>
</tr>
<tr>
<td>0 Hz</td>
<td>10%</td>
<td>2.08</td>
<td>1.11</td>
</tr>
<tr>
<td></td>
<td>1%</td>
<td>8.20</td>
<td>6.33</td>
</tr>
<tr>
<td></td>
<td>0.1%</td>
<td>12.30</td>
<td>8.77</td>
</tr>
<tr>
<td>5 Hz</td>
<td>10%</td>
<td>4.54</td>
<td>3.90</td>
</tr>
<tr>
<td></td>
<td>1%</td>
<td>13.77</td>
<td>13.88</td>
</tr>
<tr>
<td></td>
<td>0.1%</td>
<td>21.31</td>
<td>-</td>
</tr>
<tr>
<td>10 Hz</td>
<td>10%</td>
<td>18.92</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>1%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>0.1%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>20 Hz</td>
<td>10%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>1%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>0.1%</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

These results show that the performance of the GRAKE equalizer has not been changed while the FFT equalizer’s performance has become worse because the downsampling is now performed at time offset = 1/2 chip. In these results the GRAKE performance is always better than the FFT method, yet the performance of both the methods is robust to frequency errors up to 10 Hz.
Propagation channel: “Two Path (non-fading), Delay Spread = 5 chip, Offset = 0 chip”

Table 4.13: Frequency Error Comparison: Two Path Channel ‘ds’ = 5 Chip, Offset = 0 Chip

<table>
<thead>
<tr>
<th>Frequency Errors</th>
<th>Target BER</th>
<th>( \lambda_n = 1/200 )</th>
<th>( \lambda_n = 1/40 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>FFT</td>
<td>GRAKE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IorIoc [dB]</td>
<td>IorIoc [dB]</td>
</tr>
<tr>
<td>0 Hz</td>
<td>10%</td>
<td>3.32</td>
<td>4.01</td>
</tr>
<tr>
<td></td>
<td>1%</td>
<td>17.49</td>
<td>-</td>
</tr>
<tr>
<td>5 Hz</td>
<td>10%</td>
<td>6.57</td>
<td>24.88</td>
</tr>
<tr>
<td></td>
<td>1%</td>
<td>19.88</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>0.1%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>10 Hz</td>
<td>10%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>1%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>0.1%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>20 Hz</td>
<td>10%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>1%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>0.1%</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

For two path channel with delay spread = 5 chip, simulations results of the two methods are shown above in table 4.13. These results show that the FFT method has better results than the NP-GRAKE equalizer. Also the FFT method performs suitably well for frequency errors up till 10 Hz for the given channel conditions.

Conclusion:

FFT method is robust for frequency errors up to 10Hz, similarly to G-RAKE.
4.4.4 Colored Noise Comparison

The performance of the two equalization schemes (NP-GRAKE and FFT) has been compared for the colored noise scenario, i.e., another base station (Node B) causing interference in the WCDMA signal from the desired base station (Node B). No Additive White Gaussian Noise (AWGN) is added and only pure colored noise is used as interference in the simulations.

Simulation results of the two equalizers for a target BER of 10%, 1%, and 0.1% are shown below by using different channel models in the simulations.

**Table 4.14: Colored Noise Comparison**

<table>
<thead>
<tr>
<th>Target BER</th>
<th>Single Path, Offset 0</th>
<th>Single Path, Offset 2</th>
<th>Two Path (ds = 5 chip), Offset 0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FFT</td>
<td>GRAKE</td>
<td>FFT</td>
</tr>
<tr>
<td></td>
<td>IorIoc [dB]</td>
<td>IorIoc [dB]</td>
<td>IorIoc [dB]</td>
</tr>
<tr>
<td>10%</td>
<td>0.21</td>
<td>0.45</td>
<td>1.21</td>
</tr>
<tr>
<td>1%</td>
<td>7.38</td>
<td>7.49</td>
<td>8.39</td>
</tr>
<tr>
<td>0.1%</td>
<td>10.83</td>
<td>10.86</td>
<td>11.84</td>
</tr>
</tbody>
</table>

The above results show that FFT equalizer performs better than the GRAKE for single path at offset = 0 as well as for the two path channel. Also it is important to note that for two path channel with ds = 5 chip, a target BER of 0.1% using GRAKE method requires more than 28 dB of IorIoc while the FFT method requires only IorIoc = 14.32 dB, whereas, the GRAKE method requires lesser IorIoc than FFT for a target BER of 10%.

**Conclusion:**

FFT method is as robust as the G-RAKE method to colored noise, even though the used FFT method does not model the colored noise in its algorithm.
4.4.5 Analog-to-Digital Converter Comparison

Analog-to-Digital converter (A/D) is used before the Rx filter to convert the floating point signal into 8-bits, 10-bits, or 12-bits signed int values w.r.t the A/D used. Simulation results are given below for the two equalization methods (NP-GRAKE and FFT). Also for comparison to the case where no A/D is used, simulation results are also shown. In the simulation results, value of the A/D constant (K) is also shown that is used in the simulations to scale the signal for integer representation.

- Propagation channel “Single Path (non-fading), Chip Offset = 0 chip”

Table 4.15: A/D Conversion Comparison: Single Path, Offset = 0 Chip

<table>
<thead>
<tr>
<th>Target BER</th>
<th>No A/D</th>
<th>8-bit (K=50)</th>
<th>10-bit (K=150)</th>
<th>12-bit (K=600)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FFT</td>
<td>GRAKE</td>
<td>FFT</td>
<td>GRAKE</td>
</tr>
<tr>
<td>10%</td>
<td>1.03</td>
<td>1.14</td>
<td>1.04</td>
<td>1.12</td>
</tr>
<tr>
<td>1%</td>
<td>6.30</td>
<td>6.31</td>
<td>6.27</td>
<td>6.32</td>
</tr>
<tr>
<td>0.1%</td>
<td>8.78</td>
<td>8.75</td>
<td>8.79</td>
<td>8.72</td>
</tr>
</tbody>
</table>

- Propagation channel “Single Path (non-fading), Chip Offset = 1/2 chip”

Table 4.16: A/D Conversion Comparison: Single Path, Offset = 1/2 Chip

<table>
<thead>
<tr>
<th>Target BER</th>
<th>No A/D</th>
<th>8-bit (K=50)</th>
<th>10-bit (K=150)</th>
<th>12-bit (K=600)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FFT</td>
<td>GRAKE</td>
<td>FFT</td>
<td>GRAKE</td>
</tr>
<tr>
<td>10%</td>
<td>2.02</td>
<td>1.08</td>
<td>2.11</td>
<td>1.09</td>
</tr>
<tr>
<td>1%</td>
<td>8.24</td>
<td>6.35</td>
<td>8.22</td>
<td>6.30</td>
</tr>
<tr>
<td>0.1%</td>
<td>12.38</td>
<td>8.75</td>
<td>12.37</td>
<td>8.72</td>
</tr>
</tbody>
</table>
Propagation channel “Two Path (non-fading), Delay Spread = 5 chip, Chip Offset = 0 chip”

Table 4.17: A/D Conversion Comparison: Two Path ‘ds’ = 5 Chip, Offset = 0 Chip

<table>
<thead>
<tr>
<th>Target BER</th>
<th>No ADC</th>
<th>8-bit (K=50)</th>
<th>10-bit (K=150)</th>
<th>12-bit (K=600)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FFT</td>
<td>GRAKE</td>
<td>FFT</td>
<td>GRAKE</td>
</tr>
<tr>
<td>10%</td>
<td>3.21</td>
<td>3.98</td>
<td>3.31</td>
<td>3.98</td>
</tr>
<tr>
<td>1%</td>
<td>11.89</td>
<td>14.18</td>
<td>11.75</td>
<td>14.50</td>
</tr>
<tr>
<td>0.1%</td>
<td>17.22</td>
<td>-</td>
<td>17.14</td>
<td>-</td>
</tr>
</tbody>
</table>

The above simulation results show that both equalizers can be used with 8-bit implementation. The value of the A/D constant (K) which is used to multiply the input WCDMA signal is also mentioned in the tables for each A/D. From the above simulation results, it can be noted that both equalizers using the same A/D constant for their implementation. For FFT method, w.r.t the A/D constant (K), the value of ‘ε’ has to be scaled up by the square of A/D constant, i.e., K².
4.4.6 Complexity Comparison

4.4.6.1 FFT Equalizer Output Samples

The output from the FFT equalizer ‘x’ as shown in figure 4.8 represents the equalized chips after cancelling the channel effect on the WCDMA transmit signal. Total number of chips that can be equalized by the channel cancellation operation depends on the channel profile (delay spread).

For example, using N_{FFT} = 2048, FFT equalizer can perform channel cancellation operation on 2048 chips at one time. In the simulation results it is found that if all the output chips from the FFT equalizer are used for further despreading and data symbol detection, higher BER is observed. But, if we discard some chips in the end out of these 2048 chips, the BER results are improved. To illustrate more about the number of chips that needs to be discarded, simulation results for two path channel model with ‘ds’ = 5 chip and 70 chip are shown in figure 4.1.

**Figure 4.10:** FFT Equalizer Output Chip Samples vs Delay Spread
The simulation results show that, for FFT equalizer, the number of chips that are equalized in a WCDMA slot depends on the channel profile (delay spread). Also, the simulations results show that depending on the target BER output chip samples that can be retained are different. For a target BER of 10%, even with a delay spread of 70 chips, all the 2560 output chips can be used for despreading. But for a target BER of 0.1%, and delay spread = 70 chips, number of chip samples to take are about half of the output samples, i.e., 1280 chips. Similarly the above results show that, for a delay spread = 5 chip and target BER of 1% and 0.1%, about 80 chips can be discarded out of 2560 chips while others can be used for despreading and symbol detection.

Now, by using a two path channel with delay spread = 5 chip, in order to process one WCDMA slot complexity comparison for the two equalization schemes is done based on the following computations:

- CPICH channel generation
- Equalizer weights generation for NP-GRAKE or FFT method
- FFT and IFFT
- Data demodulation

Assuming that the number of despreaders used for the CPICH channel estimates are same for the two equalization methods. For a two path channel with delay spread = 5 chip, the number of despreaders required for channels estimate = 15+delay_spread+15 = 35 despreaders.

The complexity comparison will be based on the total number of complex operations required for the two equalizers. In general, we will refer to the different arithmetic operations such as multiplication, division, addition, or subtraction as a complex operation.

4.4.6.2 Computations For FFT

Total complex operations required for CPICH channel generation using FFT method is given as

$$35 \times (256 + 256 + 256 + 1) \times 10 + (35 + 35 + 35) \times 10 = 270,200$$

where first 256 operations represent the scrambling code multiplications, second 256 operations are for channelization code and third 256 operations are for the summation operation following these multiplications. One division operation is added for CPICH pilot symbol division. Total 35 despreaders are used for channel values estimation. Then the other
operations are used for the filtering operation which is performed on the current channel values and filtered channel values.

Total complex operations for an FFT are calculated using the classic Cooley-Tukey method. For a radix-2 algorithm, total complex multiplication required are $\sim (N_{FFT}/2) \log_2(N_{FFT})$ and total complex additions are $(N_{FFT}) \log_2(N_{FFT})$, where $N_{FFT}$ is the size of the FFT used [9]. After adding other operations to compute the FFT equalizer numerator and denominator values, using $N_{FFT} = 2048$, total operations required to calculate the FFT equalizer filter are

$$(N_{FFT}+ N_{FFT}/2) \log_2(N_{FFT}) + (N_{FFT}+ N_{FFT}+ N_{FFT}) = 37,888$$

Computations done for taking the FFT of the input signal and the IFFT of the equalized signal are given below

$$(N_{FFT}+ N_{FFT}/2) \log_2(N_{FFT}) + N_{FFT} + (N_{FFT}+ N_{FFT}/2) \log_2(N_{FFT}) = 69,632$$

‘M’ Equalized chips are taken out based on the decision output from the intelligent block of FFT equalizer. For a delay spread = 5 chip, using $N_{FFT} = 2048$, ‘M’ = 1984 chips out of 2048 can be retained after equalization. The computations required to detect the data symbols using the descrambling and despreading operations are then

$$M+M+M = 5,952$$

### 4.4.6.3 Computations for NP-GRAKE

Total complex operations required for CPICH channel generation using NP-GRAKE method are given below, which are the same as for the FFT method

$$35*(256+256+256+1)*10 + (35+35+35)*10 = 270,200$$

The computations required for calculating the noise covariance matrix and further the matrix inversion are determined using the Cholesky based algorithm [10]. Matrix inversion operation is performed once per WCDMA slot and noise covariance matrix is computed 10 times based on the sample channel estimate values. As the covariance matrix is symmetric, total operations are also divided by 2 in the following expression. By including the filtering operation, the number of computations required to get the equalizer filter weights for NP-GRAKE equalizer are:

$$\text{no\_fing}^3 / 3 + 2\text{no\_fing}^2 + (\text{no\_fing}^2+\text{no\_fing}^2+\text{no\_fing}^2+\text{no\_fing}^2)*(10/2) = 41,242$$
The received signal received at different rake fingers is given the associated weight and the outputs are then combined (MRC). Total complex computations per received data symbol are then:

\[ 35 \times (2560+2560+2560) + (35+35) \times 80 = 274,400 \]

A complexity comparison for the two equalizers based on the above mentioned results is given in table 4.34.

FFT method total computations = 383,672

G-RAKE method total computations = 585,842

Table 4.18: Complexity Comparison between FFT and G-RAKE

<table>
<thead>
<tr>
<th>Equalizer method</th>
<th>Total complex operations required / data symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP-GRAKE</td>
<td>3,662</td>
</tr>
<tr>
<td>FFT</td>
<td>3,094</td>
</tr>
</tbody>
</table>
4.4.7 ITU-Pedestrian B Channel Comparison

For the two equalizers, FFT and NP-GRAKE, simulation results for ITU Pedestrian-B channel (speed = 3km/hr) without AWGN are shown in Figure 4.2. Simulation parameters used are

FFT Simulation Parameters: \( N_{\text{FFT}} = 2048, \quad \lambda_b = 1/30, \quad \varepsilon = 1/50, \)
\[
\text{channel estimation despreaders} = 15 + \text{delay spread} + 15
\]

G-RAKE Simulation Parameters: finger spacing ‘\( \tau \)’ = 3/4, \( \lambda_n = 1/30, \quad \lambda_R = 1/160, \)
\[
\text{no. fingers} = (15 + \text{delay spread} + 15) / \tau
\]

Simulation Length: 30,000 slots

Using different EcIor values, BER results are plotted for the two equalizers. Where CPICH always uses the same power and power values assigned to the other users are updated corresponding to the EcIor, such that total power is always one.

![Figure 4.11: BER vs EcIor for FFT and GRAKE equalizer using ITU-PB3 Channel without AWGN](image)

From the obtained results, it is observed that the FFT equalizer with offset = 0 chip always perform better than the GRAKE equalizer. Also the FFT equalizer with offset = 1/2 chip
performs better than the GRAKE equalizer if the desired user have more power assigned. From the results it can be noted that if $E_{cIor}$ is in the range of -12 to -6 dB, then the FFT equalizer is better than the G-RAKE equalizer even if the chip offset = 1/2 chip.
Chapter 5

5 Conclusion

In this thesis, we have investigated two equalization schemes for WCDMA. One is time domain equalization using ML-based Non-parametric G-RAKE method and the other is frequency domain equalization using FFT method. A simulator is made in IT++ and uncoded BER results are used to compare the performance of these two equalizers.

BER performance results for different time offset (0, ¼, ½ chip) received signals are compared. From the observed results, it is found that the performance of FFT based equalizer is better than the G-RAKE equalizer for large delay spread channels. Also, the FFT equalizer has comparable performance to the G-RAKE equalizer for small delay spread channels. It is found that, the G-RAKE equalizer performance is not affected by the different time offset received signals, but, the FFT equalizer has varied performance results. For FFT method, the timing of the main peak needs to be determined accurately for small delay spread channels; and for larger delay spread channels, two equalizer solution might be required by measuring the SIR estimate for two different offsets and selecting the one with highest SIR estimate.

The performance of the two equalizer methods is also compared for different frequency errors generated at the UE. From the obtained results, it is found that both the equalizers are robust to frequency errors at UE, up-to 10 Hz. Simulation results for colored noise (interference from another Node B) show that the FFT equalizer is as robust to colored noise as the G-RAKE method even though the used FFT method does not model colored noise in its algorithm. Also, for analog-to-digital conversion, quantization issues are observed for different fixed point implementations and it is concluded that the FFT method can be used with 8-bit ADC implementation. Total complex operations required per data symbol for FFT and G-RAKE method are determined and it is found that the computations required for FFT method are lesser as compared to the G-RAKE method.

Also, in this thesis, a comparison is shown between FFT and G-RAKE method using ITU-Pedestrian B channel with speed = 3km/hr and frequency = 2.0 GHz. Simulations are
performed using different EcIoR values and without AWGN. From the observed results, it is found that the FFT method has better performance than the G-RAKE equalizer for fading and multi-path channel conditions.

In the simulation results it is also observed that, for FFT equalizer, the number of chips output from the equalizer depend on the delay spread. For small delay spread channels almost all the chips in the FFT window are equalized but for medium to larger delay spread channels, lesser chips can be retained at the equalizer output for subsequent despreading and data symbol detection.

Future studies can be performed to optimally place the FFT grid on the input signal to get higher throughput with minimum BER. In the future work, FFT equalizer can be implemented with fixed point implementation for propagation channels with higher Doppler frequencies. Smaller FFT window size will be required at higher velocities as the propagation channels needs to be constant over an FFT window.
Appendix

A. Parametric Equalizer

Assume $c_n$ is the sent chip sequence, and $h$ is a propagation channel including the effects of $R_x$ and $T_x$ radio filters, then we have

\[
y_n = \sum m h_m c_{n-m}
\]

where $y_n$ are sampled at the rate of produced chips $c_n$. This sampled output is then convolved with a filter $g$ at the receiver and let the final output be denoted by $z_n$, then

\[
z_n = \sum m \sum m h_m c_{m+\hat{m}-m} g_m = \{k = \hat{m} - m\} = \sum k \sum m h_m g_{m+k} c_{n+k}
\]

\[
\approx \left( \sum m h_m g_m \right) c_n \quad \text{if} \quad \sum h_m g_{m+k} = \delta_k \quad \Leftrightarrow \quad h \otimes g = \vec{e}
\]

Here, $\vec{e}$ is a unit vector containing a single 1 and the rest of elements are zero. The task is to find a filter $g$ such that the following expression is minimized

\[
\|h \otimes g - \vec{e}\|^2 + \alpha \|g\|^2
\]

where $\alpha \|g\|^2$ is a regularization term.

Let $h = \sum_{i=1}^L h_i z^{-r(i)}$ and $g = \sum_{k=1}^K g_k z^{r(k)}$. Then

\[
h \otimes g = \sum_{i=1}^L \sum_k h_i g_k z^{-r(i)+r(k)}
\]
\[
\sum_{k=1}^{K} \left( \sum_{m=1}^{L} h_{m} z^{-\tau(l)+\rho(k)} \right) g_k = Hg
\]

Here \( H \) is a \( N \times K \) complex matrix. And \( N = -\min \tau + \max \rho + \max \tau - \min \rho \)

Let the vector elements represent the delays given by \( -\tau(l) + \rho(k) \), \( l = 1, \cdots, L, k = 1, \cdots, K \)
a total of \( \leq LK \) elements. Since \( -\tau(l) + \rho(k) \) could take on the same value for different pairs \((l_1, k_1), (l_2, k_2)\).

\[
H \tilde{g} = \tilde{e}
\]

Minimize with respect to \( g \).

\[
\|Hg - \tilde{e}\|^2 + \alpha\|g\|^2
\]

The second term favors a filter \( g \) that smoothes. The factor \( \alpha \) determines the amount of smoothing to be used.

Solving for \( g \) in \( \frac{\partial}{\partial g} \|Hg - \tilde{e}\|^2 + \alpha\|g\|^2 = 0 \)

\[
\Rightarrow g = \left((H^*)^\dagger H + \alpha I\right)^{-1} (H^*)^\dagger \tilde{e}, \text{ see Appendix B for details.}
\]
B. Differentiate $\|H - e\|^2 \text{ w.r.t } g$

Let $g = g + \varepsilon \tilde{g}, \quad \varepsilon \in \mathbb{R}$

$(x, y) = x'y, \quad x, y \text{ column vectors}$

Now the directional derivate is

\[
\begin{align*}
(A(g + \varepsilon \tilde{g}) - e, H^*(g^* + \varepsilon \tilde{g}^*) - e^*) - 
& (Hg - e, H^*g^* - e^*) \\
= (Hg - e, H^*g^* - e^*) + (H \varepsilon \tilde{g}, H^*(g^* + \varepsilon \tilde{g}^*) - e^*) \\
& + (Hg - e, H^* \varepsilon \tilde{g}^*) - (Hg - e, H^*g^* - e^*) \\
= \varepsilon (H \tilde{g}, H^*g^* - e^*) + \varepsilon (Hg - e, H^* \tilde{g}^*) + O(\varepsilon^2) \\
= \varepsilon (\tilde{g}, H'H^*g^* - H'e^*) + \varepsilon [(H^*)'Hg - (H^*)'e, \tilde{g}^*]
\end{align*}
\]

The directional derivate equals (divide above by $\varepsilon$ and let $\varepsilon \to 0$)

\[
\begin{align*}
(\tilde{g}, H'H^*g^* - H'e^*) + [(H^*)'Hg - (H^*)'e, \tilde{g}^*]
\end{align*}
\]

Minimize an expression of the type

\[
\begin{align*}
\arg \min_g \alpha_1 \|H_1g - e_1\|^2 + \alpha_2 \|H_2g - e_2\|^2
\end{align*}
\]

$\Rightarrow$ all directional derivates should equal 0

$\Rightarrow$

\[
\begin{align*}
(\tilde{g}, \sum_{i=1}^{2} \alpha_i (H_i'H_i^*g^* - H_i'e_i^*)) + \\
(\tilde{g}^*, \sum_{i=1}^{2} \alpha_i [(H_i^*)'H_ig - (H_i^*)'e_i]) = 0 \quad \forall \tilde{g} \in \mathbb{C}^n
\end{align*}
\]

$\Leftrightarrow$

\[
\begin{align*}
\text{Re}(\tilde{g}, f(g)) = 0 \quad \forall \tilde{g} \in \mathbb{C}^n \\
f(g) = \sum_{i=1}^{2} \alpha_i (H_i'H_i^*g^* - H_i'e_i^*)
\end{align*}
\]

Assume $f(g) \neq 0$, take $\tilde{g} = f(g)^*$ $\Rightarrow$ contradiction.

Thus

\[
\begin{align*}
f(g) = 0 \quad \text{at a minimum} \quad \text{or}
\end{align*}
\]
$$\sum_{i=1}^{2} a_i \left( H_i^* \right) H_i g - \left( H_i^* \right) e_i = 0 \iff$$

$$g = \left( \sum_{i=1}^{2} a_i \left( H_i^* \right) H_i \right)^{-1} \left( \sum_{i=1}^{2} \left( H_i^* \right) e_i \right)$$
References


