Performance evaluation of practical G-RAKE equalizers

Dinil Koshy Mathews

Department of Electrical and Information Technology
Faculty of Engineering, LTH, Lund University
SE-221 00 Lund, Sweden

Abstract

In Wideband Code Division Multiple Access (WCDMA), RAKE receivers are used to capture the signals approaching with different delays from different paths. The wider bandwidth in WCDMA provides improved multipath resolution but at cost of higher frequency selectivity (i.e. higher intersymbol interference). The different users in a cell are differentiated by using orthogonal codes. In a highly dispersive environment, the codes no longer remain fully orthogonal. Hence this results in interference. In order to improve interference suppression G-RAKE receivers are used. This includes a RAKE plus a combiner (RACOM). The despreaded signals from each finger of RAKE are combined using weights such that the interference is minimized while maximizing signal power. In this thesis, theoretical derivation for Maximum Likelihood (ML), Minimum Mean Squared Error (MMSE) are studied for SISO system and then extended for MIMO system.

The latter part of thesis involves benchmarking a couple of different practical equalizers to identify potential areas of improvement compared to ideal equalizer. The ideal equalizer is designed with perfect knowledge of propagation channel. The current practical equalizer is evaluated mainly at three points, the number of fingers required, loss in equalizer performance due to fixed point implementation of receiver algorithms and the gap in performance between genie and practical equalizer implementation. The equalizer performance was benchmarked based on the following parameters – throughput achieved in various environments, the SNR at receiver after the RAKE and the Block Error Rate (BLER). From the thesis study, it was found out how the different practical equalizers perform when compared to each other and to genie, proposals were made as to the number of fingers to be used and on how to improve the throughput of practical equalizer. The thesis points out the limitation of the evaluated practical equalizers in a MIMO system. The thesis also describes the loss due to float to fixed point conversion and the performance gap between genie and practical equalizers.

Preface

This thesis was done at Physical Layer Algorithm and Simulation team of ST-Ericsson in Lund from 13-Feb-2012 to 13-Aug-2012. This thesis work is supervised by Dr. Elias Jonsson from ST-Ericsson and Dr. Fredrik Rusek from department of Electrical and Information Technology, LTH, Sweden. ST-Ericsson confidential results have been omitted from academic report.

I would like to take this opportunity to thank all the people who have helped me in completion of this thesis. I thank my supervisor and program coordinator at university, Dr. Fredrik Rusek for the encouragement and guidance he has given me during my thesis as well as my coursework.

I express my sincere gratitude to my supervisor at ST Ericsson, Dr. Elias Jonsson for guiding me through the thesis and helping me develop a proper grasp of the complex theory behind the different equalizer implementations. I thank him for the patience and understanding he has shown and for properly directing me in this thesis which helped me in completing the thesis in timely manner.

I would also like to thank Department manager Daniel Landström and Oskar Drugge for giving me the opportunity to do my thesis with ST Ericsson. I thank Jaroslow Niewczas and Kazuyoshi Uesaka-san for helping me understand the simulation environment.

I also thank my friends and family for the support and encouragement they have given during the duration of my thesis.

TABLE OF CONTENTS

List of Abbreviations

3GPP	Third Generation Partnership Project
WCDMA	Wideband Code Division Multiple Access
UE	User Equipment.
Node B	Base station in 3G is called as Node B.
CPICH	Common Pilot Channel
MIMO	Multiple Input Multiple Output
HSDPA	High-Speed Downlink Packet Access
kmph	Kilometer per hour
FRC	Fixed reference channel
MMSE	Minimum Mean Squared Error
ML	Maximum Likelihood
MUI	Multiple User Interference
ISI	Inter-symbol interference

Equalizer type

G-RAKE	Generalized RAKE.
P-GRAKE	Parametric G-RAKE.
NP-GRAKE	Non Parametric G-RAKE.
RD-CPICH	Equalizer with data correlation matrix estimated from despreaded data symbols and channel estimated from CPICH.
G	Equalizer with genie channel estimates, genie noise covariance matrix and ideal front end. The noise covariance matrix is calculated as in P-GRAKE.
G_15	Equalizer with genie channel estimates and ideal front end. The data correlation matrix is estimated from despreaded data symbols at RAKE finger outputs and averaged over all codes.
G_unused	Equalizer with genie channel estimates and ideal front end. The data correlation matrix is estimated only from unused code (unused code is channelization code where no data is transmitted).
P_15	Equalizer with channel estimates from CPICH and data correlation matrix estimated from all codes. This has non-ideal front end.
P_unused	Equalizer with channel estimates from CPICH and The data correlation matrix is estimated only from unused code. This has non-ideal front end.

Simulation parameters

HSPDSCH_GRAKE	Indicates the physical channel used for transmitting data (HS-PDSCH) and number of antennas at receiver, equal to 1. *HSPDSCH_GRAKE2* indicates two antennas at receiver
pedestrian B 3 kmph	Indicates the type and speed of propagation channel that is being simulated.
CQI_G	Indicates the CQI table to be used in simulation. CQI table maps the

	different CQI values to modulation format, number of channelization codes and transport block size.
max_cfn	Indicates the number of frames to be simulated. Each frame is 15 slots and 10 ms in duration.
hsscch_channel_number	Indicates which HS-SCCH channel is intended for user.
hsscch_channelisation_codes	Channelization codes to be used in HS-SCCH channels.
Ec_Ior_P_SCH_dB	Portion of total transmit power allocated to P-SCH channel.
Ec_Ior_S_SCH_dB	Portion of total transmit power allocated to S-SCH channel.
trueSNR_use	Flag to turn on/off true-SNR calculation.
fix	Flag to switch between fixed point and floating point operations.
isi_canceller_type	The type of equalizer used in receiver.
preset_code_list	Lists the unused codes i.e., channelization codes which are not used to transmit data.
wce_inv_method	Indicates method used for inversion of correlation matrix, value of '3' is for cholesky inversion and '4' for ideal inversion.
max_nrof_fingers_rd_2rx	Maximum number of fingers to be used for receiver with 2 antennas.
max_nrof_fingers_rd_1rx	Maximum number of fingers to be used for receiver with 1 antenna.
max_covariance_matrix_size	Maximum number of rows or columns for covariance matrix.
nrof_combined_fingers	Number of equalizer taps.
total_nrof_despreaders_available	Number of despreaders (correlators) required.
Ec_Ior_HS_PDSCH_dB	Portion of total power allocated to HS-PDSCH channel.
Ec_Ior_HS_SCCH_dB	Portion of total power allocated to HS-SCCH channel.
IorIoc_dB	Ratio of total transmit power to noise power in decibels.
max_nrof_trans_attempts	Maximum number of transmission attempts made by Node B for a corrupted transport block before dropping that block.
delay_dispersive_mode	Set to '1' for dispersive channels and set to '0' for non-dispersive channels.
aCQI_force_table	Set '1' to disable SIR to CQI tuning during simulation. Set '0' to enable SIR to CQI tuning during simulation.
aCQI_table_offset	Defines the offset to be added to SIR values in SIR to CQI mapping defined in aCQI_fixed_table.
aCQI_fixed_table	Fixed table which maps SIR to CQI values.

aCQI_target_BLER	Target BLER for which SIR to CQI values are adaptively tuned during simulation.
aCQI_thr_adjust	Indicates the value in decibels by which the SIR will be adjusted for a particular CQI which is showing very high BLER.
aCQI_BLER_window	Is the tolerance for target BLER for a particular CQI before the SIR value will be moved for that CQI. This means that dynamic CQI threshold will not be adjusted if BLER for particular CQI is between (aCQI_target_BLER - aCQI_BLER_window * aCQI_target_BLER) and (aCQI_target_BLER + aCQI_BLER_window * aCQI_target_BLER).
type_A	Set type_A = 0 for single stream MIMO simulation. Set type_A =1 for dual stream MIMO simulation.
hsdsch_scheduling_mode	To set follow CQI or fixed reference channel simulation.
hsdsch_tfrc_sequence	TFRC (Transport Format and Resource Combination) corresponding to CQI index to be used for transmission from Node B to UE.
hsdsch_scheduling_modeB	To set whether to use preset weights or weights based on feedback from UE for MIMO simulation. The weights are used to scale data streams and combine them before transmitting from each transmit antenna.
mimo_tx_modes	Indicates which combination of preset weights to be used for MIMO simulation.

Mathematical notation

$\mathbf{a}$	$\mathbf{a}$ is a vector (column matrix)
$\mathbf{A}$	$\mathbf{A}$ is a matrix
$(\mathbf{A})_{ij}$	Element at i^{th} row and j^{th} column in matrix $\mathbf{A}$.
$\star$	Convolution operator
$E\{\}$	Expectation operator

1 Introduction

Wideband Code Division Multiple Access (WCDMA) is the transmission standard used in third generation (3G) of mobile telecommunication system. In release 5 of 3GPP specification, a new downlink transport channel, the High Speed Downlink Shared Channel (HS-DSCH) was added to the WCDMA specification [1].

HS-DSCH supports higher order modulation, fast link adaptation and fast Hybrid ARQ with soft combining [1]. HS-DSCH enables transmission in QPSK, 16 QAM or 64 QAM depending on channel conditions. This allows for higher capacity. Fast link adaptation refers to the capability to adjust transmission parameters based on instantaneous radio-channel conditions. The UE measures the instantaneous radio-channel conditions and reports back to Node B. The Node B adjusts the transmission parameters based on UE reports. The simulations in this thesis were done for HS-DSCH.

The large bandwidth of WCDMA systems translates to a smaller chip period which increases multipath resolution [2]. A RAKE receiver is used to tap the signal energy from different paths. Equalizer design for a RAKE receiver refers to the calculation of weights used to combine the despread values produced by RAKE fingers. Interference at each RAKE finger consists of interference from symbols transmitted by other users (MUI), inter-symbol interference (ISI) and filtered AWGN noise. The RAKE receiver is called a G-RAKE (Generalized RAKE) when the combiner weights are designed to maximize signal while minimizing interference. The thesis discusses two methods of equalizer design, MMSE method and ML method.

In this thesis, we study and evaluate the performance of equalizers in HS-DSCH against genie equalizer. Here genie equalizer refers to equalizer with ideal propagation channel knowledge. Extensive simulations were done in a WCDMA simulation bench used for commercial purpose. The performance of equalizer were studied to understand

- impact of more number of RAKE fingers
- loss in performance due to floating point to fixed point conversion and
- gap in performance between genie and practical.

The thesis is arranged as follows, second chapter ‘G-RAKE equalizer’ deals with theoretical derivations of WCDMA equalizer. The expression for equalizer weights are initially derived for non MIMO system and in the second part of second chapter the results are extended for MIMO system. Parametric and non- parametric approach to equalizer design are also included in the derivations.

The third chapter ‘Equalizer evaluation’, contains description on simulation parameters, definition of different equalizer types used for evaluation and the simulation results. The chapter includes results for both non MIMO and MIMO system. The chapter also includes the analysis based on the simulation result and proposal as to improve the throughput performance.

The fourth chapter is the last chapter and it summarizes the results from the thesis study.

2 G-RAKE Equalizer

In this section we analyze the G-RAKE receiver and study in detail the derivations involved in finding the combiner weights. Initially the derivations are done for non MIMO system and later the results are extended for MIMO system.

2.1 Non MIMO system

2.1.1 Communication system

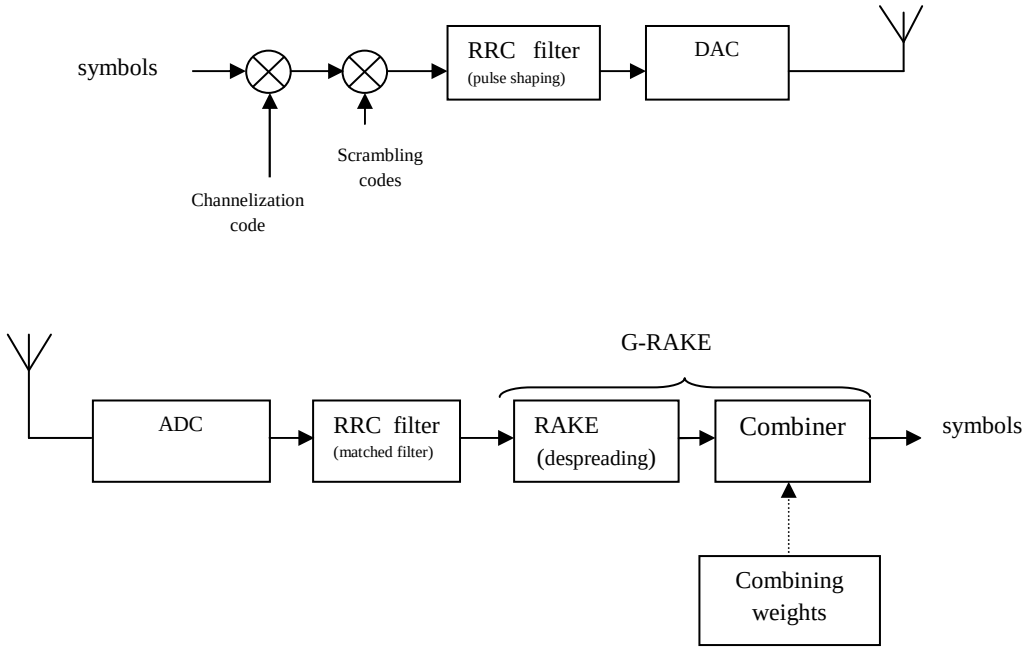


Figure 2.1: Block diagram of WCDMA transmitter (above) and receiver (below)

Figure 2.1 shows a simple block diagram representing the basic transmitter and receiver blocks of a WCDMA system. At the transmitter, the symbol stream from each user is first multiplied with user-specific channelization code and then with base station specific scrambling code to convert to chip stream. The chip stream is then passed through a Root Raised Cosine (RRC) filter to limit bandwidth occupied by the transmit signal. The pulse shaped chip sequence is then interpolated and carrier modulated before transmitting through antenna.

At the receiver side, the RF signal is carrier demodulated and down sampled to retrieve the baseband signal. The baseband signal is then passed through matched filter (RRC filter) which improves the signal to noise ratio. The output of RRC filter is passed through G-RAKE to decode the transmitted symbols.

Figure 2.2 illustrates working of G-RAKE. G-RAKE consists of two blocks – RAKE and Combiner. RAKE taps signal energy from different paths in a multipath channel environment. Each sub-module in the RAKE is called ‘RAKE finger’. Each RAKE finger consists of a despreader unit that correlates the spreading code (spreading code refers to the product of channelization code and scrambling code) to a different delay of the received signal.

The different RAKE finger outputs are weighted and combined in combiner block. In G-RAKE receiver, the weights are calculated such that the interference is minimized while maximizing required signal [2]. There are two important methods of calculating the weights – MMSE and ML method.

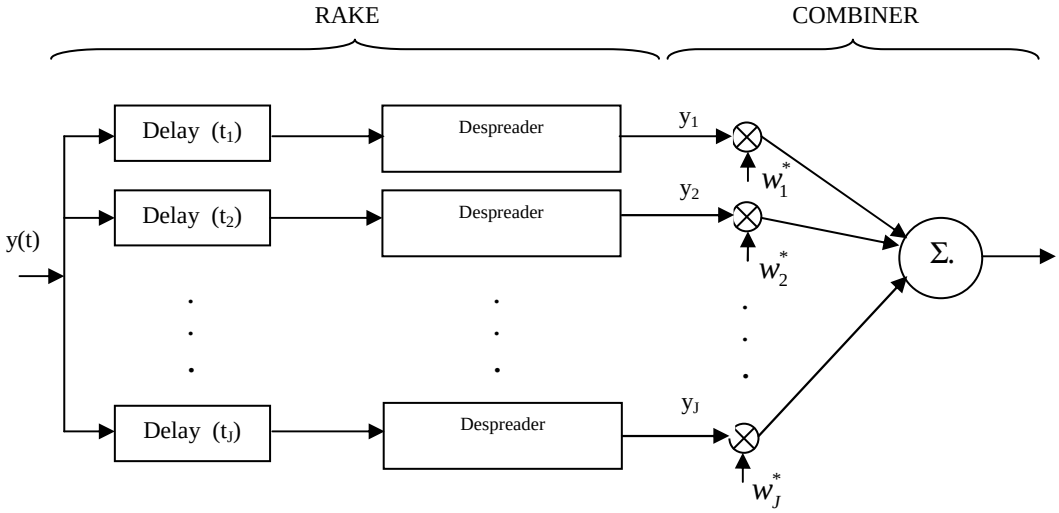


Figure 2.2: G-RAKE block diagram.

2.1.1.1 Channelization code and scrambling code

The channelization codes are user specific and are orthogonal to each other. Channelization codes help in separating symbols from different users at receiver.

$$\sum_{n=0}^{SF-1} c_i^{(ch)}(n)c_j^{(ch)}(n) = \begin{cases} SF \text{ if } i = j \\ 0 \text{ if } i \neq j \end{cases} \quad (1)$$

Here $c_i^{(ch)}$ stands for channelization code for i^{th} user and SF stands for spreading factor. The channelization code entries can take any value in ± 1.

The scrambling sequences are quasi-orthogonal random sequences of length 38400 chips. Scrambling code is same for different users transmitted from same Node B but it varies between different Node B.

$$\frac{1}{38400}\sum_{n=1}^{38400}\left(c_i^{(sc)}(n)\right)\left(c_i^{(sc)}((n-m) mod\, 38400)\right)^* = \begin{cases} = 1 \text{ if } m = 0 \\ \approx 0 \text{ if } m \neq 0 \end{cases} \quad (2)$$

Scrambling sequence does not spread the chip sequence any further but it will only scramble it by multiplication. Scrambling code can take any value in $\frac{\pm 1 \pm j}{\sqrt{2}}$.

2.1.2 Signal model

In this section, we derive the statistical signal model for output from each finger, y_i. This signal model will be used in later MMSE and ML derivations. Figure 2.3 shows how the transmitter processes the symbols from different users before transmission and how transmitted signal is affected by channel. The i^{th} symbol from k^{th} user $s_k(i)$ is multiplied with spreading code of k^{th} user c_k to obtain chip stream for k^{th} user. Chip stream is scaled by $\sqrt{E_k}$ where E_k is the energy allocated to k^{th} user. Here spreading code c_k is taken as the product of channelization code and scrambling code. The chip streams from all users are added together and pulse shaped and transmitted. Since the channelization codes are orthogonal it is possible to separate the symbols from different users at receiver.

Signal transmitted from base station can be represented as,

$$x(t) = \sum_{k=1}^{K} \sum_{n=-\infty}^{\infty} \sqrt{E_k}\,\tilde{c}_k(n)\,p(t - nT_c)$$

E_k is the symbol energy for k^{th} user.

$\tilde{c}_k$ is the product of the symbol, channelization code and scrambling code for k^{th} user.

$p(t)$ is the RRC pulse.

n is the chip index.

Here the symbol transmitted for k^{th} user is absorbed into the variable $\tilde{c}_k$. This is to simplify later calculations. The 0^{th} symbol for k^{th} user $s_k(0)$ is contained in chips $c_k(0)$ to $c_k(SF-1)$, where SF is the spreading factor.

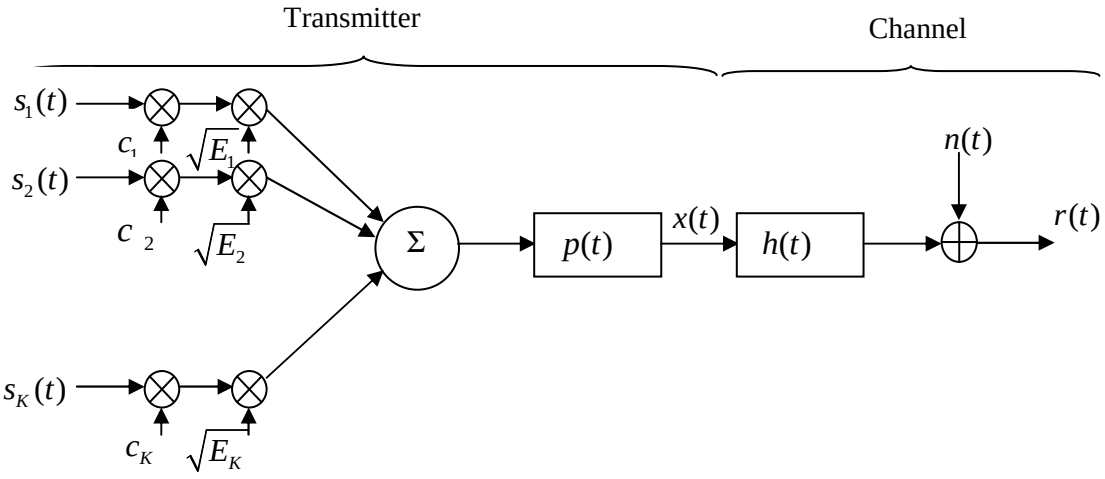


Figure 2.3: At transmitter symbols from K different users $\{s_1(t), s_2(t), \ldots, s_K(t)\}$ is multiplied with orthogonal codes $\{c_1, c_2, \ldots, c_K\}$, scaled by the energy values, summed together and pulse shaped using

RRC filter $p(t)$. The signal at transmitter output $x(t)$ passes through multipath channel $h(t)$ and AWGN noise $n(t)$ gets added to it.

During the derivation we assume that correlation between chips is 1 only when chips are from same user and has same index. In all other cases, it is zero.

$$E\{\tilde{c}_{k_1}(n_1)\tilde{c}^*_{k_2}(n_2)\} = \begin{cases} 1 \text{ if } n_1 = n_2 \text{ and } k_1 = k_2 \\ 0 \text{ otherwise} \end{cases} \quad (3)$$

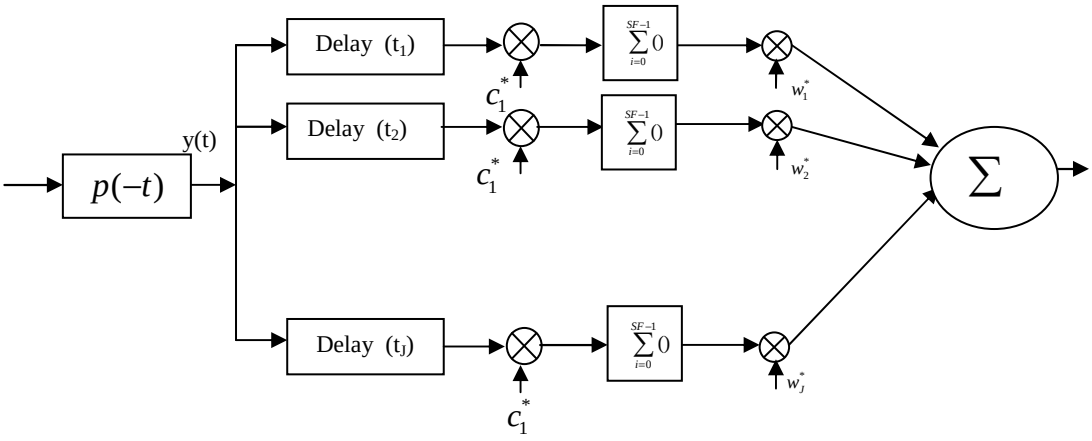


Figure 2.4: At transmitter symbols from K different users $\{s_1(t), s_2(t), \ldots, s_K(t)\}$ is multiplied with spreading codes $\{c_1, c_2, \ldots, c_K\}$, scaled by the energy values, summed together and pulse shaped using RRC filter $p(t)$. The signal at transmitter output $x(t)$ passes through multipath channel $h(t)$ and AWGN noise $n(t)$ is added to it.

This follows from the property of scrambling codes and channelization code. If $n_1 \neq n_2$ then by the property of scrambling codes (scrambling codes are pseudo random sequences) the expectation is equal to zero. When $n_1 = n_2$, and $k_1 \neq k_2$ i.e. same index but different users, we consider the symbols from different users to be uncorrelated with each other and hence the expectation is zero. When we correlate for same chip index and same user, then result is equal to 1.

Figure 2.4 shows the implementation of G-RAKE receiver which is setup to decode the symbols for 1st user. To decode symbols from 1st user despreading is done using c_1. From figure 1.4, the signal received at UE is $r(t)$ is given by,

$$r(t) = x(t) \star h(t) + n(t)$$

where $x(t)$ is the chip sequence transmitted from Node B, $h(t)$ is the multipath channel and $n(t)$ is white Gaussian noise with power spectral density N_0. Multipath channel $h(t)$ can be expressed as

$$h(t) = \sum_{l=1}^{L} h_l \delta(t - \tau_l)$$

where h_l is the channel coefficient in path l and τ_l is the delay introduced by path l. Substituting $h(t)$ in $r(t)$ gives us,

$$r(t) = \sum_{l=1}^{L} h_l x(t - \tau_l) + n(t)$$

$r(t)$ passes through matched filter (assumed symmetric $p(-t) = p(t)$) and at the output of matched filter we get,

$$y(t) = \sum_{k=1}^{K} \sum_{n=-\infty}^{\infty} \sum_{l=1}^{L} \sqrt{E_k} h_l \tilde{c}_k(n) R_p(t - \tau_l - nT_c) + \tilde{n}(t) \tag{4}$$

here $R_p(t) = \int_{-\infty}^{\infty} p(t+\tau) p(\tau) d\tau$ is the autocorrelation of the RRC pulse shape and $\tilde{n}(t) = \int_{-\infty}^{\infty} n(s) p(s-t) ds$ is the noise $n(t)$ after passing through matched filter $p(-t)$.

At the output of fingers we get $\{y_1, y_2, \ldots, y_J\}$ which are the despreaded symbol values at different delays. Let's consider the output at 1st finger which correlates the spreading sequence with the input to RAKE at delay t_1. The output at 1st finger can be expressed as,

$$y_1 = \sum_{m=0}^{SF-1} y(t_1 + mT_c) c_1^*(m)$$

which is RAKE despreading using spreading code (product of channelization code and scrambling code) c_1 starting at time t_1. Substituting expression for $y(t)$ into expression for y_1 and after simplifying we get,

$$\begin{aligned} y_1 = & \sum_{k=1}^{K} \sqrt{E_k} \sum_{l=1}^{L} h_l \sum_{m=0}^{SF-1} \tilde{c}_k(m) c_1^*(m) R_p(t_1 - \tau_l) + && \text{...(term 1)} \\ & \sum_{k=1}^{K} \sqrt{E_k} \sum_{l=1}^{L} h_l \sum_{m=0}^{SF-1} \sum_{\substack{n=-\infty \\ n \neq m}}^{\infty} \tilde{c}_k(n) c_1^*(m) R_p(t_1 - \tau_l + (m-n)T_c) + && \text{...(term 2)} \\ & \sum_{m=0}^{SF-1} \tilde{n}(t_1 + mT_c) c_1^*(m) && \text{...(term 3)} \end{aligned} \tag{5}$$

Here term 1 in equation (5) is the desired signal component $y_1^{(d)}$ and term 2 represents the Intersymbol interference (ISI) and Multi-user interference (MUI) $y_1^{(MUI+ISI)}$. Term 3 is the noise

$y_1^{(noise)}$, which has now been spread due to multiplication with the spreading code. So the output at first finger y_1 can be expressed as,

$$y_1 = y_1^{(d)} + y_1^{(MUI+ISI)} + y_1^{(noise)} \tag{6}$$

In this the desired component can be simplified by the property of channelization code given in equation (1) and of scrambling code given in equation (2). From m=0 to (SF-1), $\tilde{c}_1$ will contain the symbol s_0, which is the symbol transmitted at time $[0,(SF)(T_c)]$. Therefore,

$$\sum_{k=1}^{K}\sum_{m=0}^{SF-1}\tilde{c}_k(m)c_1^*(m) = (SF)(s_o) \tag{7}$$

On substituting equation (7) into equation (5), we get

$$\begin{aligned} y_1 = {} & \sqrt{E_1}(SF)s_0\sum_{l=1}^{L}h_l R_p(t_1-\tau_l) + \\ & \sum_{k=1}^{K}\sqrt{E_k}\sum_{l=1}^{L}h_l\sum_{m=0}^{SF-1}\sum_{\substack{n=-\infty \\ n\neq m}}^{\infty}\tilde{c}_k(n)c_1^*(m)R_p(t_1-\tau_l+(m-n)T_c) + \\ & \sum_{m=0}^{SF-1}\tilde{n}(t_1+mT_c)c_1^*(m) \end{aligned} \tag{8}$$

If we examine the first term, we can find that we have retrieved the symbol sent for user 1 at time $[0,(SF)(T_c)]$. If we choose t_1 such that it is equal to any of the path delays τ_l, then we have

$$y_1^{(d)} = \alpha s_0 h_0 R_p(0) + \alpha s_0 \sum_{\substack{l \\ \tau_l \neq t_1}} h_l\, R_p(t_1-\tau_l)$$

where α is a scaling factor. $R_p(t)$ attains peak value at $t=0$. Let's define the net propagation channel h^{net}, which includes the transmit RRC filter, multipath channel and receive filter.

$$h^{net}(t) = \sum_{l=0}^{L-1}h_l R_p(t-\tau_l) \tag{9}$$

To formulate a generic signal model, the output at finger j can be expressed as

$$y_j = y_j^{(d)} + y_j^{(MUI+ISI)} + y_j^{(noise)} \tag{10}$$

where, $y_j^d = \sqrt{E_j}(SF)h^{net}(t_j)s_o$

$$y_j^{(ISI+MUI)} = \sum_{k=1}^{K} \sqrt{E_k} \sum_{m=0}^{SF-1} \sum_{\substack{n=-\infty \\ n \neq m}}^{\infty} \tilde{c}_k(n) c_1^*(m) h^{net}(t_j + (m-n)T_c)$$

$$y_j^{(noise)} = \sum_{m=0}^{SF-1} \tilde{n}(t_1 + mT_c) c_1^*(m)$$

The expression for output at finger j (equation(10)) can be simplified as follows,

$$y_j = h_j s_0 + n_j \tag{11}$$

where h_j is the product of a scaling factor and the net propagation channel which includes transmit and receive RRC filter,

s_0 is the symbol transmitted at time $[0, (SF)(T_c)]$ and

$n_j = y_j^{(MUI+ISI)} + y_j^{(noise)}$ is the sum of interference due to other symbols transmitted from serving cell (ISI + MUI) and noise.

2.1.3 Non parametric MMSE based equalization

From equation **(11)** we have the statistical signal model; the output at finger j can be expressed as

$$y_j = h_j s_0 + n_j \tag{12}$$

We intend to find the weight w_j, which is the weight applied to the j^{th} finger such that signal to interference ratio is maximized. The MMSE method is to find weight $\boldsymbol{w} = \begin{bmatrix} w_{-M} & \cdots & w_0 & \cdots & w_M \end{bmatrix}^T$ that minimizes mean squared difference between the combined weighted despreaded symbols ($= \mathbf{w}^H \mathbf{y}$) and the transmitted symbol (s_0).

$$\min_{\vec{w}} E\left\{ \left| \sum_{j=-M}^{M} y_j w_j^* - s_0 \right|^2 \right\}$$

On solving the expression (appendix C), we get

$$\sum_{j=-M}^{M} w_j^* E\{y_j y_k^*\} = E\{s_0 y_k^*\} \tag{13}$$

We can further simplify the right hand side of equation (13) by substituting equation (12) as follows,

$$E\{s_0 y_k^*\} = E\left\{ s_0 (h_k s_0 + n_k)^* \right\} = h_k^* E\{s_0 s_0^*\} + E\{s_0 n_k^*\} = h_k^*$$

where we have used results that transmitted symbols are normalized ($s_0 s_0^* = 1$) and that symbols and noise are uncorrelated ($E\{s_0 n_k^*\} = 0$).

Equation (13) reduces to,

$$\sum_{j=-M}^{M} w_j E\{y_j^* y_k\} = h_k \tag{14}$$

where k varies from –M to M. Expressing the equation (14) in matrix format,

$$\boxed{\boldsymbol{w} = (\boldsymbol{R}_y)^{-1}\boldsymbol{h}} \tag{15}$$

where,

$$\boldsymbol{w} = \begin{bmatrix} w_{-M} & \cdots & w_0 & \cdots & w_M \end{bmatrix}^T$$

$$\boldsymbol{R}_y = E\{yy^H\} = \begin{bmatrix} r(0) & \cdots & r^*(M) & \cdots & r^*(2M) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ r(M) & \cdots & r(0) & \cdots & r^*(M) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ r(2M) & \cdots & r(M) & \cdots & r(0) \end{bmatrix}, \text{ where } r(j-k) = E\{y_j y_k^*\} = r^*(k-j)$$

$$\boldsymbol{h} = \begin{bmatrix} h_{-M} & \cdots & h_0 & \cdots & h_M \end{bmatrix}^T$$

Here $\boldsymbol{R}_y$ is the data correlation matrix which is calculated by finding the correlation of despread data symbols observed across fingers. In practical implementation the data correlation matrix is filtered across slots to smooth out noise.

$$\boldsymbol{R}_y^{filt,currentslot} = \lambda \boldsymbol{R}_y^{filt,prevslot} + (1-\lambda)\boldsymbol{R}_y \tag{16}$$

$\boldsymbol{R}_y^{filt,currentslot}$ is the filtered data correlation matrix for current slot used to calculate weights for current slot.

$\boldsymbol{R}_y^{filt,prevslot}$ is the filtered data correlation matrix for previous slot.

$\boldsymbol{R}_y$ is the data correlation matrix calculated for current slot by correlating despread symbol across fingers.

λ is the filtering parameter that depends on Doppler spread.

2.1.4 ML based equalization

Consider $\tilde{\mathbf{y}} = \begin{bmatrix} \tilde{y}_1 & \tilde{y}_2 & \cdots & \tilde{y}_J \end{bmatrix}^T$ to be an observation or realization of set of despread symbols at output of J RAKE fingers. Through maximum likelihood method we intend to find symbol s_0 that maximizes the likelihood of occurrence of $\tilde{\boldsymbol{y}}$. From equation **(11)** we have,

$$y_j = h_j s_0 + n_j$$

To proceed with maximum likelihood estimation, we have to fit $\vec{y}$ to a multi-dimensional probability distribution. A suitable choice is the Gaussian probability distribution which is simpler to manipulate and contains few parameters to estimate. Probability distribution function of $\boldsymbol{y}$ is,

$$p(\mathbf{y} : \mathbf{h}s_0, \mathbf{R}_u) = \frac{1}{(2\pi)^{\frac{J}{2}} \mathbf{R}_u^{\frac{1}{2}}} \exp\left(\frac{-1}{2} (\mathbf{y} - \mathbf{h}s_0)^H \mathbf{R}_u^{-1} (\mathbf{y} - \mathbf{h}s_0) \right)$$

where $\mathbf{R}_u = E\{\vec{n}\vec{n}^H\}$, $\boldsymbol{n} = \begin{bmatrix} n_1 & \cdots & n_J \end{bmatrix}^T$ and $\boldsymbol{h} = \begin{bmatrix} h_1 & \cdots & h_J \end{bmatrix}^T$. The likelihood function can be expressed as,

$$L(s_0 \mid \tilde{\mathbf{y}}) = \frac{1}{(2\pi)^{\frac{J}{2}} \mathbf{R}_u^{\frac{1}{2}}} \exp\left(\frac{-1}{2} (\tilde{\mathbf{y}} - \mathbf{h}s_0)^H \mathbf{R}_u^{-1} (\tilde{\mathbf{y}} - \mathbf{h}s_0) \right)$$

To find s_0 that maximizes the likelihood function, take derivative of logarithm of $L(s_0 \mid \tilde{\vec{y}})$ with respect to s_0 and equate to zero.

$$\frac{\partial}{\partial s_0} log(L(s_0 \mid \tilde{\mathbf{y}})) = 0 \tag{17}$$

$$\begin{aligned}
\frac{\partial}{\partial s_0} log(L(s_0 \mid \tilde{\mathbf{y}})) &= \frac{\partial}{\partial s_0} \left((\tilde{\mathbf{y}} - \mathbf{h}s_0)^H (\mathbf{R}_u^{-1}) (\tilde{\mathbf{y}} - \mathbf{h}s_0) \right) \\
&= \frac{\partial}{\partial s_0} \left(\sum_{i=1}^{J} \sum_{j=1}^{J} (\mathbf{R}_u^{-1})_{ij} (\tilde{y}_i^* - h_i^* s_0^*)(\tilde{y}_j - h_j s_0) \right) \\
&= (-2) \sum_{i=1}^{J} \sum_{j=1}^{J} (\mathbf{R}_u^{-1})_{ij} (\tilde{y}_i^* - h_i^* s_0^*)(h_j) \qquad \left(\because \frac{\partial s_0}{\partial s_0} = 2 \text{ and } \frac{\partial s_0^*}{\partial s_0} = 0 \right) \\
&= (-2)(\tilde{\mathbf{y}} - \mathbf{h}s_0)^H (\mathbf{R}_u^{-1})(\mathbf{h})
\end{aligned} \tag{18}$$

Equating equation (18) to zero we get

$$s_0^* = (\mathbf{h}^H \mathbf{R}_u^{-1} \mathbf{h})^{-1} \tilde{\mathbf{y}}^H \mathbf{R}_u^{-1} \mathbf{h}$$

If we ignore the scaling factor $(\vec{h}^H \mathbf{R}_u^{-1} \vec{h})^{-1}$, we get

$$s_0 = (\mathbf{R}_u^{-1} \mathbf{h})^H \tilde{\mathbf{y}} \tag{19}$$

which is of the form,

$$s_0 = \mathbf{w}^H \tilde{\mathbf{y}} \tag{20}$$

where $\mathbf{w} = \begin{bmatrix} w_1 & \cdots & w_J \end{bmatrix}^T$ is the weight vector used to combine the finger outputs $\tilde{\mathbf{y}} = \begin{bmatrix} \tilde{y}_1 & \tilde{y}_2 & \cdots & \tilde{y}_J \end{bmatrix}^T$.

From equation (19) and equation (20)

$$\boxed{\mathbf{w} = (\mathbf{R}_u)^{-1}\mathbf{h}}. \tag{21}$$

Equation (21) gives us the maximum likelihood formulation of finding the combining weights. On comparing with weights found using MMSE method (equation(15)) we can see that the only difference is in type of correlation matrix used. ML method uses the noise covariance matrix $\mathbf{R}_u$ but MMSE method uses the data correlation matrix $\mathbf{R}_y$.

In ML method, we can find $\mathbf{R}_u$ parametrically (i.e. express $\mathbf{R}_u$ as function of different parameters and we substitute the values of the parameters to find $\mathbf{R}_u$) or non-parametrically (i.e. we estimate $\mathbf{R}_u$ from the despreaded pilot symbols). If $\mathbf{R}_u$ is found parametrically such an equalizer implementation is called Parametric GRAKE (P-GRAKE) and if $\mathbf{R}_u$ is found non-parametrically, the equalizer implementation is called Non Parametric G-RAKE (NP-GRAKE).

2.1.4.1 Parametric G-RAKE

Let's express (j_1, j_2) element in R_u as

$$(\boldsymbol{R}_u)_{j_1 j_2} = E\{n_{j_1} n_{j_2}^*\} \tag{22}$$

where n_{j_1}, n_{j_2} are the noise component n_j (given in equation **(11)**) for fingers j_1 and j_2. Expanding n_{j_1}, n_{j_2} based on equation**(8)**,

$$n_{j_1} = \sum_{k_1=1}^{K} \sqrt{E_{k_1}} \sum_{l_1=1}^{L} h_{l_1} \sum_{m_1=0}^{SF-1} \sum_{\substack{n_1=-\infty \\ n_1 \neq m_1}}^{\infty} \tilde{c}_{k_1}(n_1) c_1^*(m_1) R_p(t_{j_1} - \tau_{l_1} + (m_1 - n_1)T_c) + \sum_{m_1=0}^{SF-1} \tilde{n}(t_{j_1} + m_1 T_c) c_1^*(m_1)$$

$$n_{j_2}^* = \sum_{k_2=1}^{K} \sqrt{E_{k_2}} \sum_{l_2=1}^{L} h_{l_2}^* \sum_{m_2=0}^{SF-1} \sum_{\substack{n_2=-\infty \\ n_2 \neq m_2}}^{\infty} \tilde{c}_{k_2}^*(n_2) c_1(m_2) R_p^*(t_{j_2} - \tau_{l_2} + (m_2 - n_2)T_c) + \sum_{m_2=0}^{SF-1} \tilde{n}^*(t_{j_2} + m_2 T_c) c_1(m_2)$$

Substituting into expressions for n_{j_1}, n_{j_2} into equation (22),

$$(\boldsymbol{R}_u)_{j_1 j_2} = \sum_{k_1,k_2=1}^{K} \sqrt{E_{k_1}}\sqrt{E_{k_2}} \sum_{l_1,l_2=1}^{L} h_{l_1} h_{l_2}^{*} \sum_{m_1,m_2=0}^{SF-1} \sum_{\substack{n_1,n_2=-\infty \\ n_1\neq m_1 \\ n_2\neq m_2}}^{\infty} R_p(t_{j_1} - \tau_{l_1} + (m_1 - n_1)T_c) R_p^{*}(t_{j_2} - \tau_{l_2} + (m_2 - n_2)T_c)$$

$$E\{\tilde{c}_{k_1}(n_1) c_1^{*}(m_1) \tilde{c}_{k_2}^{*}(n_2) c_1(m_2)\} \quad \text{...(interference term)} \tag{23}$$

$$+ \sum_{m_1,m_2=0}^{SF-1} E\{\tilde{n}(t_{j_1} + m_1 T_c)\tilde{n}^{*}(t_{j_2} + m_2 T_c)\} E\{c_1(m_2) c_1^{*}(m_1)\} \quad \text{...(noise term)}$$

where we have omitted terms with expectation of product of symbol and noise since those terms tend to zero as noise and symbol are uncorrelated. Here we make a reasonable assumption that for $n_1 \neq n_2$ and $m_1 \neq m_2$

$$E\{\tilde{c}_{k_1}(n_1) c_1^{*}(m_1) \tilde{c}_{k_2}^{*}(n_2) c_1(m_2)\} = \begin{cases} 1 \text{ if } m_1 = m_2, n_1 = n_2, k_1 = k_2 \\ 0 \text{ otherwise} \end{cases} \tag{24}$$

This simplification does not directly follow from the assumption made in equation (3). However, if we assume that $c_{k_1}(n_1)$ and $c_{k_2}(n_2)$ to be independent if $n_1 \neq n_2$ or $k_1 \neq k_2$ then expression (24) becomes reasonable. So the interference term in equation (23) reduces to,

$$= \sum_{k=1}^{K} E_k \sum_{l_1,l_2=1}^{L} h_{l_1} h_{l_2}^{*} \sum_{m=0}^{SF-1} \sum_{\substack{n=-\infty \\ n\neq m}}^{\infty} R_p(t_{j_1} - \tau_{l_1} + (m-n)T_c) R_p^{*}(t_{j_2} - \tau_{l_2} + (m-n)T_c)$$

$$= (SF) \underbrace{\sum_{k=1}^{K} E_k}_{=I_{or}} \sum_{l_1,l_2=1}^{L} h_{l_1} h_{l_2}^{*} \sum_{\substack{n=-\infty \\ n\neq 0}}^{\infty} R_p(t_{j_1} - \tau_{l_1} + nT_c) R_p^{*}(t_{j_2} - \tau_{l_2} + nT_c) \tag{25}$$

$$= (SF) I_{or} \sum_{l_1,l_2=1}^{L} h_{l_1} h_{l_2}^{*} \sum_{\substack{n=-\infty \\ n\neq 0}}^{\infty} R_p(t_{j_1} - \tau_{l_1} + nT_c) R_p^{*}(t_{j_2} - \tau_{l_2} + nT_c)$$

I_{or} is the total transmit power at Node B. Let's consider the noise term in equation (23). Referring back to equation **(4)** expand $\tilde{n}(t)$ in equation (23),

$$E\{\tilde{n}(t_{j_1}+m_1T_c)\tilde{n}^*(t_{j_2}+m_2T_c)\} = E\left\{\left(\int_{-\infty}^{\infty} n(s)\,p(s-t_{j_1}-m_1T_c)ds\right)\left(\int_{-\infty}^{\infty} n(u)\,p(u-t_{j_2}-m_2T_c)du\right)^*\right\}$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(s-t_{j_1}-m_1T_c)p^*(u-t_{j_2}-m_2T_c)E\{n(s)n^*(u)\}\,ds\,du$$

$$= N_0\int_{-\infty}^{\infty} p^*(u-t_{j_2}-m_2T_c)\,p(u-t_{j_1}-m_1T_c)du \qquad \left(\because E\{n(s)n^*(u)\} = N_0\delta(s-u)\right)$$

$$= N_0R_p((t_{j_2}-t_{j_1})+(m_2-m_1)T_c)$$

$$E\{c_1(m_2)c_1^*(m_1)\} = 1 \text{ only if } m_1 = m_2$$

Noise term simplifies to,

$$\sum_{m_1,m_2=0}^{SF-1} E\{\tilde{n}(t_{j_1}+m_1T_c)\tilde{n}^*(t_{j_2}+m_2T_c)\}E\{c_1(m_2)c_1^*(m_1)\} = \sum_{m=0}^{SF-1} N_0R_p(t_{j_2}-t_{j_1}) = (SF)N_0R_p(t_{j_2}-t_{j_1}) \quad (26)$$

Substituting equation (25) and (26) into equation (23), we get simplified equation for $(\mathbf{R}_u)_{j_1j_2}$

$$(\mathbf{R}_u)_{j_1j_2} = (SF)I_{or}\sum_{l_1,l_2=1}^{L} h_{l_1}h_{l_2}^*\sum_{\substack{n=-\infty\\ n\neq 0}}^{\infty} R_p(t_{j_1}-\tau_{l_1}+nT_c)R_p^*(t_{j_2}-\tau_{l_2}+nT_c)+(SF)N_0R_p(t_{j_2}-t_{j_1})$$

(27)

Writing equation (27) in terms of net propagation channel as given in equation **(9)**,

$$\boxed{(\mathbf{R}_u)_{j_1j_2} = (SF)I_{or}\sum_{\substack{n=-\infty\\ n\neq 0}}^{\infty}\left(h^{net}(t_{j_1}+nT_c)\right)\left(h^{net}(t_{j_2}+nT_c)\right)^* + N_0(SF)R_p(t_{j_2}-t_{j_1})} \quad (28)$$

Equation (28) gives us the parametric expression for $(\mathbf{R}_u)_{j_1j_2}$. In practice, the net channel coefficients (h^{net}) can be estimated at the finger delays and other delays spaced an integral number of chip periods from the finger delays by despreading and smoothing the pilot channel at those delays.

2.1.4.2 Non Parametric G-RAKE

Since pilot symbols are known at receiver, we can use them to estimate noise correlation matrix. We can model the pilot channel as (refer to equation **(11)**),

$$y_j = h_j p_0 + n_j \quad (29)$$

where p_0 is the pilot symbol. From equation (22) we have,

$$\begin{aligned}(\boldsymbol{R}_u)_{j_1 j_2} &= E\{n_{j_1} n_{j_2}^*\} \\ &= (y_{j_1} p_0^* - h_{j_1}^{filt,currentslot})(y_{j_2} p_0^* - h_{j_2}^{filt,currentslot})^*\end{aligned} \tag{30}$$

where $h_j^{filt,currentslot}$ is the filtered version of the samples $y_j p_0^*$ for the current slot

$$h_j^{filt,currentslot} = \lambda_h (y_j p_0^*) + (1-\lambda_h) h_j^{filt,prevslot}$$

where λ_h is a filtering parameter, $h_j^{filt,prevslot}$ is the filtered version of the samples $y_j p_0^*$ for the previous slot.

2.1.5 Comparison between weights obtained using MMSE and ML method

From equation (15) and equation (21) we have

$$\begin{aligned}\boldsymbol{w}_{MMSE} &= (\mathbf{R}_y)^{-1}\boldsymbol{h} \\ \boldsymbol{w}_{ML} &= (\mathbf{R}_u)^{-1}\boldsymbol{h}\end{aligned} \tag{31}$$

We can prove that the weights derived using MMSE method ($\boldsymbol{w}_{MMSE}$) and ML method ($\boldsymbol{w}_{ML}$) are equal up to a scaling constant. Let's first express $\mathbf{R}_y$ in terms of $\mathbf{R}_u$.

$$\begin{aligned}\mathbf{R}_y &= E\{\mathbf{y}\mathbf{y}^H\} \\ &= E\{(\mathbf{h}s_0 + \mathbf{n})((\mathbf{h}^H s_0^* + \mathbf{n}^H)\} \\ &= \mathbf{h}\mathbf{h}^H + \mathbf{h}E\{\mathbf{n}^H s_0\} + E\{\mathbf{n}s_0^*\}\mathbf{h}^H + E\{\mathbf{n}\mathbf{n}^H\} \qquad (\because |s_0|^2 = 1) \\ &= \mathbf{h}\mathbf{h}^H + \mathbf{R}_u \qquad (\because E\{\mathbf{n}^H s_0\} = 0 \text{ and } E\{\mathbf{n}s_0^*\} = 0)\end{aligned}$$

Taking inverse of $\mathbf{R}_y$,

$$\begin{aligned}(\mathbf{R}_y)^{-1} &= (\mathbf{R}_u + \mathbf{h}\mathbf{h}^H)^{-1} \\ &= (\boldsymbol{I} + (\mathbf{R}_u)^{-1}\mathbf{h}\mathbf{h}^H)^{-1}(\mathbf{R}_u)^{-1} \qquad (\because (\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1})\end{aligned}$$

Using property $(\mathbf{I} + \mathbf{a}\mathbf{b}^H)^{-1} = \mathbf{I} - \dfrac{\mathbf{a}\mathbf{b}^H}{1+\mathbf{b}^H\mathbf{a}}$, where $\mathbf{a} = (\mathbf{R}_u)^{-1}\mathbf{h}$ and $\mathbf{b} = \mathbf{h}$ are vectors.

$$(\mathbf{R}_y)^{-1} = (\mathbf{R}_u)^{-1} - \frac{(\mathbf{R}_u)^{-1}\mathbf{h}\mathbf{h}^H(\mathbf{R}_u)^{-1}}{1+\mathbf{h}^H(\mathbf{R}_u)^{-1}\mathbf{h}} \tag{32}$$

Substituting equation (32) in equation (31),

$$\begin{aligned}\mathbf{w}_{MMSE} &= \mathbf{R}_y^{-1}\mathbf{h} \\ &= ((\mathbf{R}_u)^{-1}\mathbf{h}) - \frac{((\mathbf{R}_u)^{-1}\mathbf{h})\mathbf{h}^H((\mathbf{R}_u)^{-1}\mathbf{h})}{1+\mathbf{h}^H((\mathbf{R}_u)^{-1}\mathbf{h})} \\ &= \mathbf{w}_{ML} - \frac{\mathbf{w}_{ML}\mathbf{h}^H\mathbf{w}_{ML}}{1+\mathbf{h}^H\mathbf{w}_{ML}} \\ &= \frac{\mathbf{w}_{ML}}{1+\mathbf{h}^H\mathbf{w}_{ML}}\end{aligned}$$

where $1+\mathbf{h}^H\mathbf{w}_{ML}$ is a scalar quantity. Thus ideally $\mathbf{w}_{MMSE}$ and $\mathbf{w}_{ML}$ are scaled versions of each other. But later in simulation section of this report we can see that the results from MMSE and ML method show considerable difference.

2.2 MIMO

In MIMO systems we use multiple transmit and receive antennas to increase data rate. HSDPA uses 2x2 MIMO. MIMO system can be configured to beam-form data, increase diversity or spatially multiplex data depending on SNR.

For a typical channel capacity plot, $C = \log_2(1+SNR)$ capacity slope is steeper at low SNR and the curve flattens at higher SNR. So an increase in received SNR at lower SNR region will translate to higher increase in achievable data rates. Received SNR at low SNR region can be increased by beam-forming (same data transmitted from both antennas but with phase offset) or diversity (Space time coding) techniques. At higher SNR region, capacity curve is almost flat, so increase in SNR gives only smaller increase in capacity. At this region, we can utilize multi-stream transmission since it will divide the available SNR into several data streams increasing the data without bandwidth expansion. In medium SNR region which is the SNR in many common deployments e.g. urban and sub-urban macro cells, a mixture of beam forming and spatial multiplexing is used.

For MIMO operation HS-DSCH channel, either single stream or dual stream transmission can be applied. In single stream transmission, the same data content is weighted differently and transmitted from both transmit antennas. In dual stream transmission, which is a mixture of beam forming and spatial multiplexing, two parallel streams of data are transmitted simultaneously from transmit antennas. Figure 2.5 shows how MIMO is implemented in HS-DSCH. Other down link channels may be transmitted using transmit diversity, e.g. STTD (Space Time Transmit Diversity) or TSTD (Time Switched Transmit Diversity).

UE determines preferred precoding vector (w_1, w_2) and signals it to Node B [5]. The weights are determined as follows (Section 9, [5])

$$w_3 = w_1 = \frac{1}{\sqrt{2}} \text{ and } w_4 = -w_2 \text{ where } w_2 \in \left\{\frac{1+j}{2}, \frac{1-j}{2}, \frac{-1+j}{2}, \frac{-1-j}{2}\right\}$$

The weights make the streams transmitted from antennas orthogonal to each other. UE determines CQI and feeds back to Node B.

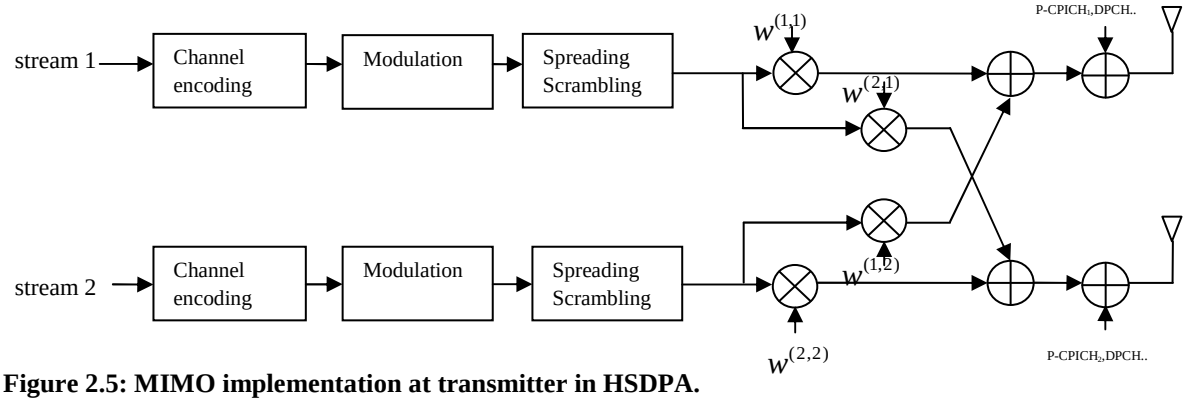


Figure 2.5: MIMO implementation at transmitter in HSDPA.

In this section GRAKE derivations are extended for a Multiple Input and Multiple Output system.

2.2.1 Signal model

We model the different data streams as different physical channels. Different modes considered in this derivation are,

- Single stream: Data from user is spread using user-specific spreading code and then sent with different weights from two transmit antennas.
- Dual stream: Data from same user split into two streams, spread using same user-specific spreading code, weighted and added together before sending simultaneously from the two transmit antennas as shown in Figure 2.5.

The data transmitted from transmitter can be expressed as,

$$x^{(q)}(t) = \sum_{k=1}^{K} \sum_{n=-\infty}^{\infty} \sum_{s=0}^{S(k)-1} \sqrt{E_k}\, w_k^{(q,s)} \tilde{c}_k^{(q,s)}(n)\, p(t - nT_c)$$

q transmit antenna index, k is user index, n is chip index and s is stream index.

$S(k)$ is the number of streams for k^{th} user . For single and dual stream transmission, $S(k)$ is 1 and 2 respectively.

$\tilde{c}_k^{(q,s)}$ is the product of symbol, channelization code and scrambling code for s stream of k^{th} user is transmitted through q^{th} antenna. Let's denote c_k as the product of channelization code and scrambling code and this is the same for both streams. $\tilde{c}_k^{(0)}$ contains symbol $s_k^{(0)}$ (symbol sent by k^{th} user in stream 0 in time $[0, SF * T_c]$) and $\tilde{c}_k^{(1)}$ contains symbol $s_k^{(1)}$ (symbol sent by k^{th} user in stream 1 in time $[0, SF * T_c]$).

E_k is the energy assigned to k^{th} user.

$p(t)$ is the transmit pulse.

$w_k^{(q,s)}$ is the weight applied to s stream transmitted from q antenna for k^{th} user.

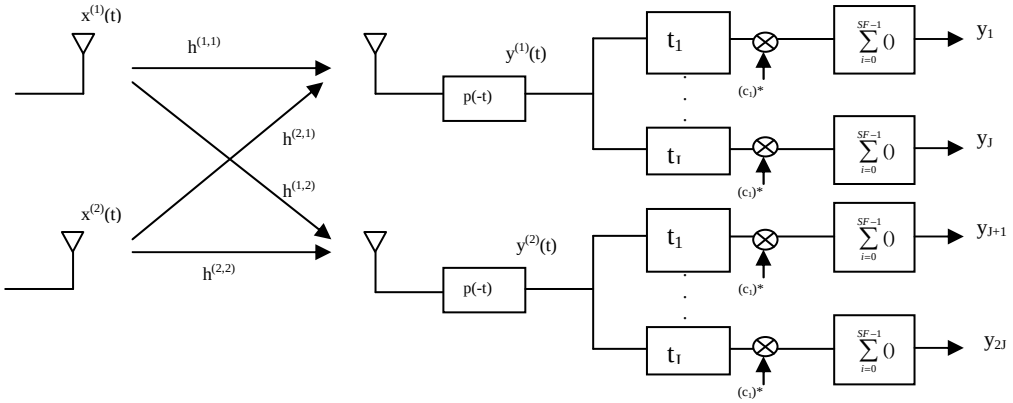


Figure 2.6: Transmission of data through MIMO channel and RAKE despreading at the receiver across two receiver antennas to recover symbols transmitted by 1st user.

Figure 2.6 shows the RAKE despreading for MIMO channel to decode symbols sent from 1st user. We can derive an expression for output from each RAKE finger output as we did for non-MIMO channel. The output at jth finger can be expressed as,

$$y_j = h_j^{(0)} s_1^{(0)} + h_j^{(1)} s_1^{(1)} + n_j \tag{33}$$

where,

$h_j^{(0)} = (SF)\sum_{q=1}^{2} \sqrt{E_1} h_{net}^{(r,q)}(t_j) w_1^{(q,0)}$, r is index of receive antenna

$$h_j^{(1)} = (SF)\sum_{q=1}^{2} \sqrt{E_1} h_{net}^{(r,q)}(t_j) w_1^{(q,1)}$$

$$n_j = \sum_{q=1}^{2}\sum_{k=1}^{K} \sqrt{E_k} \sum_{m=0}^{SF-1} \sum_{\substack{n=-\infty \\ n\neq m}}^{\infty} \sum_{s=0}^{S(k)} \tilde{c}_k^{(q,s)}(n) w_k^{(q,s)} h_{net}^{(r,q)}(t_j + (m-n)T_c) c_1^*(m) + \sum_{m=0}^{SF-1} \tilde{n}(t_1 + mT_c) c_1^*(m)$$

$s_k^{(0)}$ symbol sent by k^{th} user in stream 0 in time $[0, SF * T_c]$

$s_k^{(1)}$ (symbol sent by k^{th} user in stream 0 in time $[0, SF * T_c]$

The MIMO signal model is essentially same as Non MIMO signal model. If we are decoding symbol sent in first stream $s_1^{(0)}$ then symbol in second stream is considered as random variable and is added to the MUI and ISI terms, which gives us the following signal model,

$$y_j = h_j^{(0)} s_1^{(0)} + \widetilde{n_j} \tag{34}$$

Equation (34) is same as equation (12) for the non MIMO signal model. So MMSE and ML formulations retain the same expression as Non MIMO for MIMO.

2.2.2 MMSE

Weights derived using MMSE method to decode first stream $s_1^{(0)}$ is given by

$$\boldsymbol{w}^{(0)} = (\boldsymbol{R}_y)^{-1}\boldsymbol{h}_1^{(0)} \tag{35}$$

and weights to decode second stream $s_1^{(1)}$ is given by

$$\boldsymbol{w}^{(1)} = (\boldsymbol{R}_y)^{-1}\boldsymbol{h}_1^{(1)} \tag{36}$$

where $\mathbf{w}^{(s)} = [w_{-2J}^{(s)} \cdots w_0^{(s)} \cdots w_{2J}^{(s)}]$ is the weight vector used to combine finger outputs $\mathbf{y} = [y_{-2J} \cdots y_0 \cdots y_{2J}]$ in order to decode s stream,

$\mathbf{R}_y = E\{yy^H\}$ is the autocorrelation matrix of despreaded finger outputs $\mathbf{y}$, and $\mathbf{h}^{(s)} = [h_{-2J}^{(s)} \cdots h_0^{(s)} \cdots h_{2J}^{(s)}]$ is the distortion factor of symbol $s_1^{(s)}$ as given in equation (33), due to the propagation channel.

2.2.3 Maximum Likelihood

Weights derived using ML method to decode first stream $s_1^{(0)}$ is given by

$$\mathbf{w}^{(0)} = \underbrace{(\mathbf{R}_n + \mathbf{h}_1^{(1)}(\mathbf{h}_1^{(1)})^H)}_{R_u^{(0)}}{}^{-1}(\mathbf{h}_1^{(0)})^* \tag{37}$$

Similarly to decode second stream $s_1^{(1)}$

$$\mathbf{w}^{(1)} = \underbrace{(\mathbf{R}_n + \mathbf{h}_1^{(0)}(\mathbf{h}_1^{(0)})^H)}_{R_u^{(1)}}{}^{-1}(\mathbf{h}_1^{(1)})^* \tag{38}$$

$\mathbf{n} = [n_1 \cdots n_{2J}]$ where n_j is the sum of MUI and ISI at each finger as given in equation (33).

$\mathbf{R}_n = E\{\mathbf{n}\mathbf{n}^H\}$,

$\mathbf{h}^{(s)} = [h_1^{(s)} \cdots h_{2J}^{(s)}]$ is the distortion factor of symbol $s_1^{(s)}$ as given in equation (33), due to the propagation channel.

2.2.3.1 Parametric G-RAKE

We can express the noise at each finger j as follows,

$$\tilde{n}_j = \sqrt{E_1}\sum_{q=1}^{2}\sum_{m=0}^{SF-1}\sum_{\substack{n=-\infty \\ n\neq m}}^{\infty}\tilde{c}_1^{(q,0)}(n)w_1^{(q,0)}h_{net}^{(r,q)}(t_j+(m-n)T_c)c_1^*(m) \quad \text{...(user 1,data stream 0)}$$

$$+\sqrt{E_1}\sum_{q=1}^{2}\sum_{m=0}^{SF-1}\sum_{\substack{n=-\infty \\ n\neq m}}^{\infty}\tilde{c}_1^{(q,1)}(n)w_1^{(q,1)}h_{net}^{(r,q)}(t_j+(m-n)T_c)c_1^*(m) \quad \text{...(user 1,data stream 1)}$$

$$+\sum_{q=1}^{2}\sum_{k=2}^{K}\sqrt{E_k}\sum_{m=0}^{SF-1}\sum_{\substack{n=-\infty \\ n\neq m}}^{\infty}\sum_{s=0}^{S(k)}\tilde{c}_k^{(q,s)}(n)w_k^{(q,s)}h_{net}^{(r,q)}(t_j+(m-n)T_c)c_1^*(m) \quad \text{..(data streams from other users)}$$

$$+\sum_{m=0}^{SF-1}\tilde{n}(t_1+mT_c)c_1^*(m) \quad \text{..(noise)}$$

Using following assumption, which does not assume STTD encoding,

$$E\{\tilde{c}_{k_1}^{(q_1,s_1)}(n_1)c_1^*(m_1)(\tilde{c}_{k_2}^{(q_2,s_2)}(n_2))^*c_1(m_2)\} = \begin{cases}1, & \text{if } m_1=m_2, n_1=n_2, k_1=k_2, s_1=s_2\\ 0, & \text{otherwise}\end{cases}$$

the noise covariance matrix to find symbol $s_1^{(0)}$ can be expressed as,

$$\begin{aligned}(\boldsymbol{R}_u^{(0)})_{j_1j_2} &= E\{\tilde{n}_{j_1}\tilde{n}_{j_2}^*\}\\ &= E_1(SF)\sum_{q_1,q_2=1}^{2}\sum_{n\neq 0}(w_1^{(q_1,0)})(w_1^{(q_2,0)})^*h_{net}^{(r,q)}(t_{j_1}+nT_c)h_{net}^{(r,q)}(t_{j_2}+nT_c)\\ &+E_1(SF)\sum_{q_1,q_2=1}^{2}\sum_{n=-\infty}^{\infty}(w_1^{(q_1,1)})(w_1^{(q_2,1)})^*h_{net}^{(r,q)}(t_{j_1}+nT_c)h_{net}^{(r,q)}(t_{j_2}+nT_c)\\ &+(SF)\sum_{k=2}^{K}E_k\sum_{q_1,q_2=1}^{2}\sum_{n=-\infty}^{\infty}\sum_{s=0}^{S(k)}(w_k^{(q_1,s)})(w_k^{(q_2,s)})^*h_{net}^{(r,q)}(t_{j_1}+nT_c)h_{net}^{(r,q)}(t_{j_2}+nT_c)\\ &+(SF)N_0R_p(t_{j_2}-t_{j_1})\end{aligned} \quad (39)$$

2.2.3.2 Non parametric G-RAKE

Non parametric G-RAKE as given in equation (30) for non MIMO can be applied for MIMO. The noise covariance matrix to decode stream 0, is given by

$$\begin{aligned}(\boldsymbol{R}_u^{(0)})_{j_1j_2} &= E\{n_{j_1}n_{j_2}^*\}\\ &= (y_{j_1}(s_1^{(0)})^* - h_{j_1}^{(0)})(y_{j_2}(s_1^{(0)})^* - h_{j_2}^{(0)})^*\end{aligned}$$

where $h_{j_1}^{(0)}$ is the mean of $y_{j_1}(s_1^{(0)})^*$ and $h_{j_2}^{(0)}$ is the mean of $y_{j_2}(s_1^{(0)})^*$.

3 Equalizer evaluation

3.1 Simulation Setup

Simulations were done in STE's advanced in-house simulator which is implemented in C++ and allows us to simulate equalizer performance in UE for different base station parameters, channel and equalizer settings. In the simulation environment, we can choose the type (dispersive/ non-dispersive) and speed of channel. The results from the simulation can be analyzed to find throughput, SNR and BLER which helps in quantifying the performance of receiver.

The simulation environment takes its input from a control file, which is a Perl script used to configure the simulation parameters. In the control file, each test case is identified by a simulation index and it contains the simulation parameters describing the channel condition, modem settings and output parameters. Test case refers to simulation with data being sent to and fro from Node B to User Equipment for a number of frames for the propagation channel defined in the test case. For example, a particular test case will be to run the simulation for Pedestrian B channel at 3 kilometer per hour with IorIoc 20 (PB3-20) for practical equalizer for 2000 frames. Once the simulation is finished we analyze the output parameters to compare the equalizer performance. The various simulation parameters are discussed in detail later in this section.

3.2 Understanding Genie and practical equalizers

Genie equalizer is assumed to have perfect knowledge of propagation channel. The genie noise covariance matrix ($\mathbf{R}_u$) is calculated parametrically as shown in equation(28),

$$(\mathbf{R}_u)_{j_1 j_2} = (SF) I_{or} \sum_{\substack{n=-\infty \\ n\neq 0}}^{\infty} \left(h^{net}(t_{j_1} + nT_c)\right)\left(h^{net}(t_{j_2} + nT_c)\right)^* + (SF) N_0 R_p (t_{j_2} - t_{j_1})$$

The net channel $h^{net}(t)$ used for simulation is known at the receiver in genie mode and this exact value is used for calculating noise covariance matrix. Thus we have perfect knowledge of propagation channel while calculating $\mathbf{R}_u$. The combiner weights are calculated using equation(21), $\mathbf{w} = (\mathbf{R}_u)^{-1}\mathbf{h}$,where $\mathbf{h} = [h_1 \quad \cdots \quad h_J]^T$ can be calculated from $h^{net}(t)$.

$$h_j = (SF)\sqrt{E_j}\, h^{net}(t_j)$$

$\mathbf{R}_u$ and $\mathbf{h}$ calculated from ideal $h^{net}(t)$ knowledge are called genie covariance matrix and genie channel estimate respectively. Genie equalizer has genie covariance matrix and genie channel estimate.

Practical equalizers considered are combination of data correlation matrix and genie or practical channel estimates. Practical equalizer implementation is based on MMSE method as given in equation (15)

$$\boldsymbol{w} = (\boldsymbol{R}_y)^{-1}\boldsymbol{h}$$

There are two different methods of calculating $\mathbf{R}_y$,

1. Data correlation matrix calculated from all codes – In HS-DSCH, 15 channelization codes are shared by the users. Irrespective of the channelization code assigned to the user, the user can generate 15 realizations of data correlation matrix corresponding to each user. The final data correlation is found by averaging the 15 realizations as follows,

$$\boldsymbol{R}_y = \frac{1}{15}\sum_{i=1}^{15} \boldsymbol{R}_y^{(i)}$$

$\boldsymbol{R}_y^{(i)}$ is the data correlation matrix obtained by dispreading using i^{th} channelization code. Data correlation matrix calculated from all codes is called RD-ALL.

2. Data correlation matrix calculated from unused code – In this method, data correlation matrix is generated using channelization code in which no data is transmitted. Data correlation matrix calculated from unused code is called RD-Unused.

While calculating weights we can use either genie or practical channel estimate ($\mathbf{h}$). In practical channel estimation, a net channel estimate for each slot is calculated by averaging the 10 channel samples obtained by despreading and correlating the CPICH symbols.

$$h_{(i,j)}^{samp} = y_j p_i^*$$

$$h_{(current\_slot,j)}^{est} = \frac{1}{10}\sum_{i=1}^{10} h_{(i,j)}^{samp}$$

y_j is the depsreaded output at j^{th} finger of CPICH despreader.

p_i is the i^{th} pilot symbol in a slot.

$h_{(i,j)}^{samp}$ is the channel sample found by correlating with i^{th} pilot symbol in a slot at j^{th} finger.

$h_{(current\_slot,j)}^{est}$ is the channel estimate for current slot for j^{th} finger found by averaging the channel samples in a slot. The finger index is dropped from later calculations to improve readability.

Following equalizer configurations were considered for further analysis

1) G – Genie covariance matrix and genie channel estimates
2) G_15 – RD-ALL and genie channel estimates
3) G_unused – RD-Unused and genie channel estimates
4) P_15 – RD-ALL and practical channel estimates
5) P_unused – RD-Unused and practical channel estimates

3.3 Simulation parameters

In this section, the different simulation parameters are discussed in further detail. Basically the simulation parameters help us to define the Node B, Channel and UE characteristics and to analyze the results once simulation is finished.

3.3.1 Input parameters

Here we discuss the parameters used to change Node B, propagation channel and User Equipment settings.

3.3.1.1 Propagation channel specific parameters

We can specify the type and speed of channel in the simulation environment. The different types of channels which are relevant to equalizer benchmarking are AWGN, Pedestrian A (PA), Pedestrian B (PB) and Vehicular (VA) channels. The power delay profile for PA, PB and VA are specified in Table 3.1. AWGN is channel with additive white Gaussian noise and has no fading. PA, PB and VA

channels are dispersive fading channels. Pedestrian B channel has the highest delay spread. So it is prudent to use higher number of fingers in Pedestrian B than in Pedestrian A. We can also set the speed of channel and it is specified in kilometers per hour (kmph). Higher speed indicates faster channel variations and increased error rate.

Pedestrian A (3km/h)		Pedestrian B (3 km/h)		Vehicular A (30 km/h)	
Relative Delay [ns]	Relative Mean Power [Db]	Relative Delay [ns]	Relative Mean Power [dB]	Relative Delay [ns]	Relative Mean Power [dB]
0	0	0	0	0	0
110	-9.7	200	-0.9	310	-1
190	-19.2	800	-4.9	710	-9
410	-22.8	1200	-8	1090	-10
		2300	-7.8	1730	-15
		3700	-23.9	2510	-20

Table 3.1: The propagation conditions for PA, PB and VA as specified in [4].

3.3.1.2 Node B specific parameters

3.3.1.2.1 Transmit power

Transmit power at Node B (base station) is determined by Ior/Ioc parameter. This is the ratio of Node B power (Ior) to noise power (Ioc), expressed in decibels. Here noise refers to interference from other cells and thermal noise. If Ior/Ioc is 0 dB, then the transmit power and noise power are equal. The ratio Ior/Ioc is also referred to as geometry. Higher geometry refers to less noisy conditions. In this report, geometry is indicated by *IorIoc*.

The power allocated to different channels (HS-PDSCH, CPICH etc.) can also be specified through simulation parameters. The ratio of power allocated to a particular channel is written as Ec/Ior, which is the ratio of transmit energy per chip (Ec) for that channel to total transmit power at base station (Ior). If Ec/Ior for HS-PDSCH channel is –3 dB, then it means that 50% of the total transmit power is allocated to HS-PDSCH channel. For the simulations done as part of this thesis, no power was allocated to the control channels like Primary synchronization channel (P-SCH) and Secondary synchronization channel (S-SCH). This is to avoid interference due to these channels at the receiver since they are non-orthogonal to the other channels. Instead, the receiver was setup to have prior knowledge of (genie) information communicated by these control channels, like frame synchronization and downlink scrambling code.

3.3.1.2.2 Transport Format and Resource Combination (TFRC)

TFRC refers to the modulation format (QPSK, 16 QAM, 64 QAM), number of channelization codes and transport block size used to send data in downlink in HS-PDSCH. Two different modes of TFRC selection are as follows,

- Follow CQI – CQI stands for Channel Quality Indicator and it is an indication as to how good the channel propagation condition is. Higher CQI indicates better channel propagation condition. The UE calculates the SIR (Signal to interference ratio) and maps the SIR values to CQI values. If the UE detects higher SIR it will report higher CQI. The mapping from SIR to CQI needs to be adjusted to achieve target BLER. Node B will choose the TFRC based on the CQI reports it gets from the UE. This link adaptation scheme in which Node B changes TFRC based on CQI reports is known as Follow CQI, figure 3.1.

TFRC corresponding to each CQI is defined in CQI table. CQI_G table defined in 3GPP spec and CQI_Normal_G was used for Non MIMO simulation. These tables are defined in Appendix A.

- Adaptive CQI tuning – For the simulations, adaptive CQI tuning was turned on for the Follow CQI simulations. Adaptive CQI tuning basically tunes the mapping from SIR to CQI so as to achieve target BLER.

- Fixed TFRC – In this simulation mode the TFRC to be used in HS-DSCH channel is fixed. Although the UE reports CQI to Node B, Node B does not adjust TFRC based on CQI reports. In the simulation environment, we can specify the TFRC combination to be used by giving the corresponding index in the CQI table. For e.g. if CQI_G28 is specified, then TFRC corresponding to index 28 in CQI table (transport block size of 32264 bytes, 14 channelization codes, and 64 QAM modulation) will be used throughout simulation.

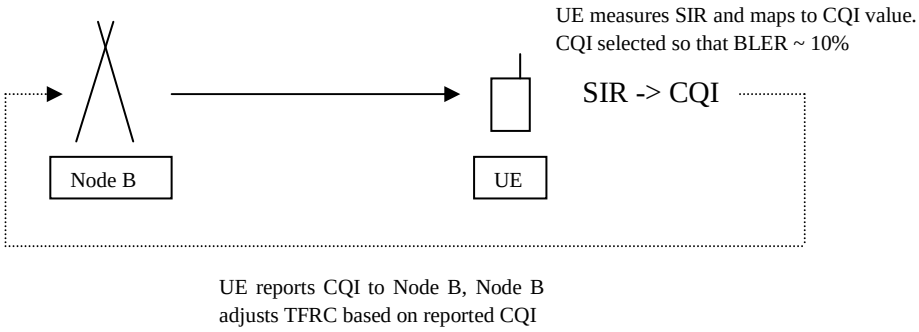


Figure 3.1: Follow CQI mode of operation.

3.3.1.2.3 Maximum number of retransmission attempts

If the received packet is corrupted (which can be verified by CRC check) then the UE can send a NACK back to Node B and Node B may retransmit the packet again. Through this parameter we can specify the maximum number of retransmission attempts the base station makes before it drops (or ignores) the packet.

3.3.1.3 UE specific parameters

3.3.1.3.1 Equalizer type

Equalizer type to be used is set by the 'isi_canceller_type' parameter. The different equalizer options are 'RD_CPICH', 'P_GRAKE' and 'NP_GRAKE'. The equalizer weights **w** can be expressed as ratio of channel estimates **h** to correlation matrix **R**.

$$\boldsymbol{w} = (\boldsymbol{R})^{-1}\boldsymbol{h}$$

The different equalizer methods differ in how it calculates the correlation matrix and channel estimates. If we set 'isi_canceller_type' to RD_CPICH then **R** will be calculated by autocorrelation of depsreaded symbols of RAKE fingers (MMSE method, equation(15)) and **h** will be calculated by despreading using CPICH symbols (as explained in section 3.2).
If 'isi_canceller_type' is set to P_GRAKE then **R** is noise covariance matrix calculated parametrically (ML method, equation(28) and **h** estimated from CPICH symbols.
In NP_GRAKE method, R is noise covariance matrix calculated non-parametrically (ML method, equation (30)) and **h** estimated from CPICH symbols.

3.3.1.3.2 Number of receiver antennas

The number of receiver antennas can be switched between 1 and 2 using parameters 'HSPDSCH_GRAKE' and 'HSPDSCH_GRAKE2' respectively.

3.3.1.3.3 Float or fixed point implementation

The receiver algorithms are implemented in fixed point for it to with limited bit widths. Float to fixed point conversion involves quantization to convert floating point numbers to fixed point numbers and sometimes entirely new algorithm has to be implemented to do a particular operation in fixed point. By using a parameter *'fix'*, we can turn on/off fixed point operations. This helps in understanding the precision loss by implementing a particular algorithm in fixed point.

3.3.1.3.4 Correlation matrix inversion method

We can control the algorithm used to invert the correlation matrix using parameter '*wce_inv_method*'. *wce_inv_method = [5]* stands for inversion using built in matrix inverse of IT++, *wce_inv_method = [4]* stands for cholesky inversion.

3.3.1.3.5 Adaptive delta scaling factor

Before fixed point inversion of correlation matrix, a constant value is added to the diagonal elements of correlation matrix.

$$R = R + \delta * I$$

Where R is the correlation matrix, δ is adaptive delta scaling factor (scalar constant) and I is identity matrix. The adaptive delta scaling factor can be changed using 'adaptive_delta_scaling_factor_rd' for RD equalizer.

3.3.1.3.6 Number of fingers

The number of fingers in RAKE receiver can be changed using parameter 'max_nrof_fingers_rd_2rx' for 2 receiver antenna configuration and using parameter 'max_nrof_fingers_rd_1rx' for 1 receiver antenna configuration. The parameters to change along with above mentioned ones are "max_covariance_matrix_size", "nrof_combined_fingers" and "total_nrof_despreaders_available".

3.3.1.3.7 Genie mode

Genie mode can be turned on using the parameters,
"rf_asic" => ["MOA"], "front_end_settings" => [3], "front_end_receiver_mode" => [0], - this sets up idealized front-end mode, no impairments
"tx_rrc_length_chips" => [30], length of RRC filter in number of chips.
"use_genie_chest" => [1], Use genie channel estimates
"use_genie_cov_matrix" => [1], Use genie correlation matrix

3.3.1.4 Miscellaneous input parameters

3.3.1.4.1 Number of frames

This sets the number of frames to be simulated. Each frame consists of 15 slots and is 10 ms in duration. Number of frames to be simulated is set by parameter '*max_cfn*'.

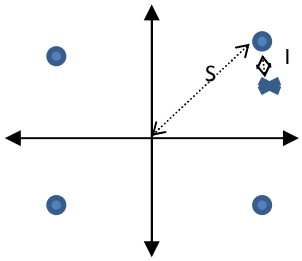
3.3.2 Output parameters

3.3.2.1 True SNR

For true SNR calculation, it is assumed that transmitted constellation is known at the receiver. True SNR is calculated as follows,

$$\text{True SNR} = \frac{\text{Transmitted constellation power}}{\text{Noise power}}$$

Here noise power is the square of distance between received symbol and actual transmitted symbol. Transmitted constellation power is the square of distance of symbol from the origin in the ideal constellation.



$$trueSNR = \frac{(S)^2}{(I)^2}$$

Figure 3.2: Calculation of true SNR. The blue circles represent the transmit constellation. The blue cross mark represents the decoded symbol.

3.3.2.2 Throughput

Throughput achieved by considering the total bits transmitted successfully over the number of frames. Throughput is printed in kbps.

$$\text{Throughput} = \frac{\text{total bits transmitted successfully}}{\text{total frames*10 ms*1000}} \text{ (kbps)}$$

3.3.2.3 BLER

BLER stands for Block Error Rate. For HS-DSCH there is at most one transport block of dynamic size per Transmission Time Interval (TTI). Each transport block has one CRC attachment and receiver performs a CRC check after turbo decoder to verify whether the transport block is corrupted or not. BLER is ratio of number of corrupted transport blocks to total transport blocks transmitted.

3.4 Simulations

Simulations were done for both non-MIMO and MIMO configurations. For each of the configurations, following points were evaluated,

1. Number of fingers - The purpose of this set of simulations was to figure out how much improvement in throughput can be achieved by increasing the number of RAKE fingers or equivalently the number of equalizer taps.
2. Practical equalizer evaluation – Compare the performance of different practical equalizer configurations with respect to genie equalizer configuration.
3. Fixed point design – The purpose of this set of simulations was to find out the throughput loss caused by fixed point implementation of various receiver algorithms. A broader set of test cases covering different speeds, geometry and dispersion was run to evaluate the float to fixed point gap.

4. Practical to genie gap – Here practical refers to fixed point equalizer implemented in STE simulator. Genie refers to floating point implementation with ideal front-end, ideal channel knowledge and parametrically found covariance matrix. Simulations were done for broader set of test cases as done for fixed point design. Through the simulations, those environments and geometries were identified which showed large genie to practical gap.

3.4.1 Non MIMO

3.4.1.1 Number of Fingers

3.4.1.1.1 Description

The purpose of this set of simulations was to understand the increased benefits of using more fingers in the equalizer. Simulations were done in Pedestrian B channel at 3 kmph for high geometries. Higher geometries, which mean less noisy environments, enable us to better exploit the degrees of freedom offered by more number of fingers. Pedestrian B channel was chosen since it has largest delay spread. Larger delay spread means that there is more energy to be captured by increasing number of fingers. (For PA channel, which is least dispersive, we will not get higher throughput gains by increasing number of fingers and the throughput saturates for less number of fingers). The test cases that were simulated for finding the number of fingers are provided in table 3.2,

Test	Channel	Geometry	Antennas	Receiver
1	PB3	noiseless	1	Genie
2	PB3	noiseless	1	Practical
3	PB3	noiseless	2	Genie
4	PB3	noiseless	2	Practical
5	PB3	20	1	Genie
6	PB3	20	1	Practical
7	PB3	20	2	Genie
8	PB3	20	2	Practical

Table 3.2: Test cases for number of fingers dimensioning

3.4.1.1.2 Simulation setup

Simulations were done for G,G_15, G_unused,P_15 and P_unused equalizer. "*Pedestrian B 3 kmph*" channel was defined in all the test cases. The number of receiver antennas was switched using parameters "*HSPDSCH_GRAKE*" and "*HSPDSCH_GRAKE2*". Geometry was set to 20 and 40dB using *IorIoc_dB* parameter. Test cases were run with 24, 48, 64, 72, 80, 96, 144 and 192 fingers

Follow CQI with adaptive CQI tuning
For low speed test cases such as 3 kmph, simulations were done with follow CQI with adaptive CQI tuning. In Follow CQI mode, Node B adjusts the transport format based on CQI reports from UE. Adaptive CQI tuning is turned on to tune the mapping from SIR to CQI in UE, so as to achieve target BLER. The CQI table was set to 3GPP CQI table and target BLER rate was set to 15%.

For adaptive CQI tuning, a fixed table lists the SIR values corresponding to each CQI value. If the BLER based on this default list is far from the target BLER, then the adaptive CQI algorithm (which dynamically adjusts the SIR to CQI mapping) will take a long time to tune the SIR to CQI mapping so

that we get target BLER. To aid this tuning process, we can manually specify a table offset which can be considered as a constant value that is added to all the SIR values in fixed table. So for a positive table offset, we are effectively increasing the SIR value required to switch CQI. For example, if SIR = 4.8 corresponds to CQI 0 and SIR = 5.8 corresponds to CQI 1 then once the SIR at receiver reaches 5.8, the UE will report CQI 1. If follow CQI is enabled, the base station will send the transport format combination corresponding to CQI 1. If we specify table offset as 2, then SIR = 6.8 will correspond to CQI 0 and SIR = 7.8 will correspond to CQI 1, therefore once SIR reaches 5.8 the UE will report CQI 0. So we effectively reduce BLER with higher table offset. In order to initially adjust the table offset, we run the simulations for less number of frames and if the BLER achieved is much higher than target BLER, we lower table offset and vice versa.

Theoretical maximum throughput – Based on the maximum allowed CQI (CQI 29 or CQI 30) different maximum throughput values can be reached. For CQI_G table, if CQI 30 is allowed, then up to transport block size of 38576 bits can be transmitted in one TTI (transmission time interval – 2ms) which yields the theoretical maximum throughput of 19.28 Mbps. If use of CQI 30 is not allowed, then theoretical maximum value is 16.132 Mbps which corresponds to throughput achievable by using CQI 29. In simulations done to evaluate performance of equalizers at different number of fingers, CQI 30 was not allowed. Hence maximum throughput achievable is 16.132 Mbps.

3.4.1.1.3 Results

The results were analyzed using true SNR values and throughput values. An increase in true SNR values indicates a corresponding increase in throughput. The results can be analyzed for following cases:

1) High dispersive channel (PB3)
   a) Two receiver antennas, IorIoc 20
      The throughput and true SNR plots are shown in figure 3.3.

- The full genie configuration (‘G’) shows increase in performance up to 144 fingers leveling off from 144 to 192 fingers.
- RD-unused method with genie (G_unused) and practical channel estimates (P_unused) exhibit similar performance, peaking at 96 fingers and then after 96 fingers performance starts decreasing. Inaccuracy of practical channel estimates when compared to genie channel estimates seems not to affect the performance for RD-unused method.
- RD-ALL method shows considerable difference in performance between genie (G_15) and practical (P_15) channel estimates. G_15 performance levels off at 48 fingers but P_15 shows decreasing performance from 24 fingers. This shows that there is lot of performance to gain in P_15 method with better channel estimation techniques.
- Another important observation is that P_unused is better than G_15, which shows that however better channel estimation technique we use in RD-ALL method it will not catch up with performance of RD-unused method. RD-unused method is clearly better than RD-ALL method while calculating equalizer weights.
- For genie equalizer configuration at IorIoc 20, as the channel condition worsens (low SIR) CQI drops from CQI 29 to maintain BLER, hence throughput achieved is less than theoretical maximum.

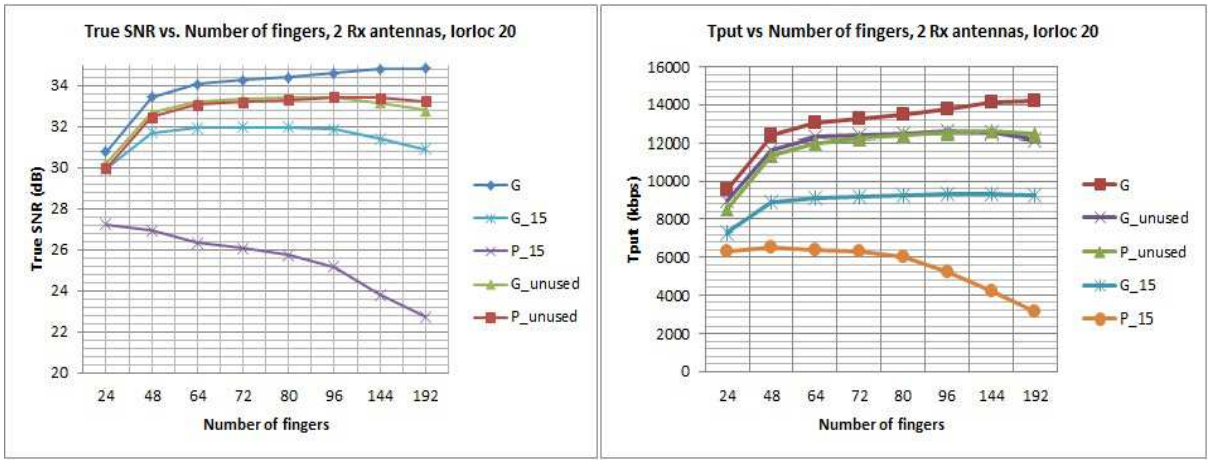


Figure 3.3: Simulated true SNR and throughput performance of UE with two receiver antennas as we increase number of fingers for Pedestrian B channel at 3 kmph and geometry IorIoc 20.

b) Single receiver antenna, IorIoc 20
The throughput and true SNR performance is shown in figure 3.4.

- For genie equalizer, throughput obtained from single receiver antenna is nearly half of the throughout obtained from using 2 receiver antennas. This shows the advantage of using two receiver antennas at receiver terminal. By using two receiver antennas there is more chance of seeing a good channel in at least one antenna.
- Apart from the lower throughput, the performance of different equalizer configurations with increasing fingers remains similar to two receiver antennas, at same geometry. A notable difference is that G_15 performance catches up with G_unused with one receiver antenna.

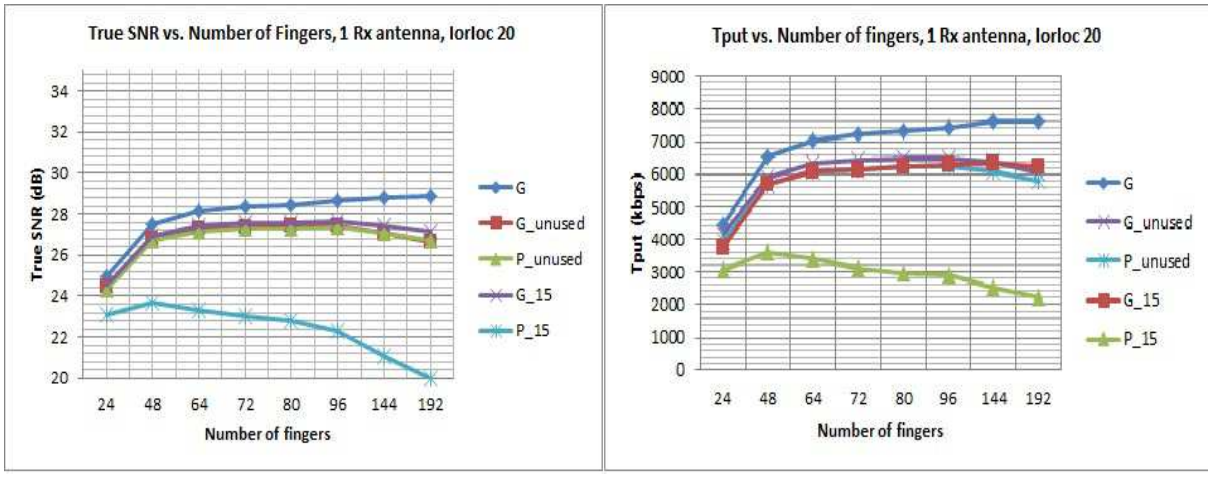


Figure 3.4: Simulated true SNR and throughput performance of UE with single receiver antenna with increasing fingers for Pedestrian B channel at 3 kmph and IorIoc 20

c) Two receiver antennas, IorIoc 40
Throughput and true SNR plots are shown in figure 3.5.

- Much higher throughput than two receiver antennas for noiseless channel. Full genie throughput levels off at nearly 16 Mbps at 48 fingers. Although true SNR increases till 144

fingers but this does not translate to growth in throughput performance since it reaches maximum achievable throughput earlier at 48 fingers itself.

- With noiseless channel, RD-unused (G_unused & P_unused) catches up in throughput performance with full genie (G) at 144 fingers.
- G_15 performance nearly levels off at 48 fingers, but holds on till 96 fingers and after that performance starts falling. P_15 shows decreasing performance with higher number of fingers.
- We can see that in noiseless case (discussed later in this section) performance of genie equalizer reaches theoretical maximum value of 16 Mbps. In noiseless case, CQI 29 is selected throughout the simulation duration.

d) Single receiver antenna, IorIoc 40

Throughput and true SNR plots are shown in figure 3.6.

- At IorIoc 40, throughput for 1 receiver antenna catches up with that of 2 receiver antenna for genie mode with 192 fingers. From this we can infer that the receiver works well even in fading dips when there is no noise in channel. For 2 receiver antenna configuration, we assume that receiver sees 2 independent channel realizations. So the probability of having poor channel (fading dip) at both antennas decreases and this is the reason for better throughput with 2 receiver antenna. But in noiseless condition, the receiver with 1 antenna and 192 fingers starts decoding content even in fading dips, making the second receiver antenna redundant.
- Equalizer performance is similar to two receiver antenna at IorIoc 40, notable difference being G_15 catches up with RD_unused method.

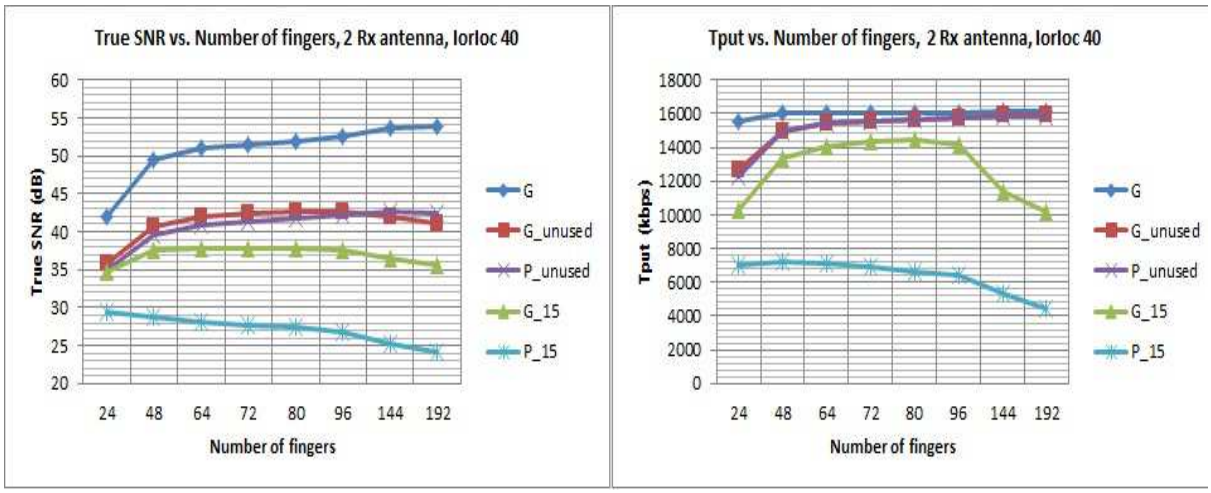


Figure 3.5: Simulated true SNR and throughput performance of UE with 2 receiver antennas as we increase number of fingers for Pedestrian B channel at 3 kmph and noiseless condition (IorIoc 40).

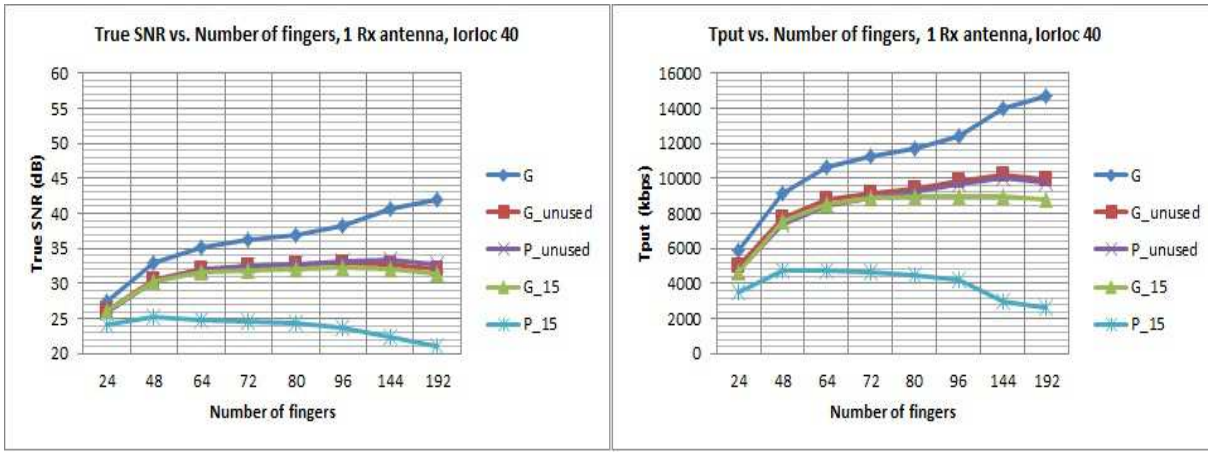


Figure 3.6: Simulated true SNR and throughput performance of UE with single receiver antenna with increasing fingers for Pedestrian B channel at 3 kmph and noiseless channel (IorIoc 40).

2) Low dispersive channel (PA3)
   a) 2 receiver antennas, IorIoc 20
   Throughput and true SNR plots are shown in figure 3.7

- Additional simulations were done in PA3 with FRC to understand the equalizer performance in less dispersive channel.
- The genie equalizer (G) does not show any performance improvement with higher number of fingers since this is less dispersive channel. All the other equalizers show decreasing throughput and true SNR with more fingers.
- P_unused is better that G_15, which shows that RD_unused method is better than RD_ALL at low dispersive channel as well.

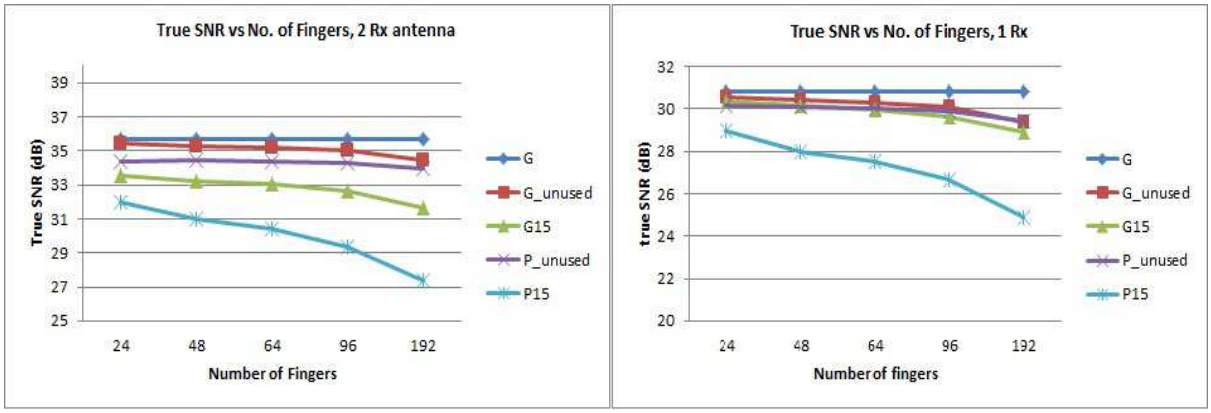


Figure 3.7: Simulated true SNR and throughput performance of UE with two receiver antenna and single receiver antenna with increasing fingers for Pedestrian A channel at 3 kmph and noiseless channel (IorIoc 40).

The number of floating point operations (flops) required for cholesky inversion for n-finger RAKE is around $n^3/3$ [6]. So when we double the finger count, the number of floating point operations per second goes up by 8 times.

3.4.1.1.4 Throughput vs. target BLER for different geometries

The purpose of this analysis was to see how the throughput varies when we change the adaptive BLER target in different geometries. The simulations were done for full genie ('G) mode for 96 fingers for Pedestrian B channel at 3 kmph. The simulation setup was same as that for analyzing number of fingers.

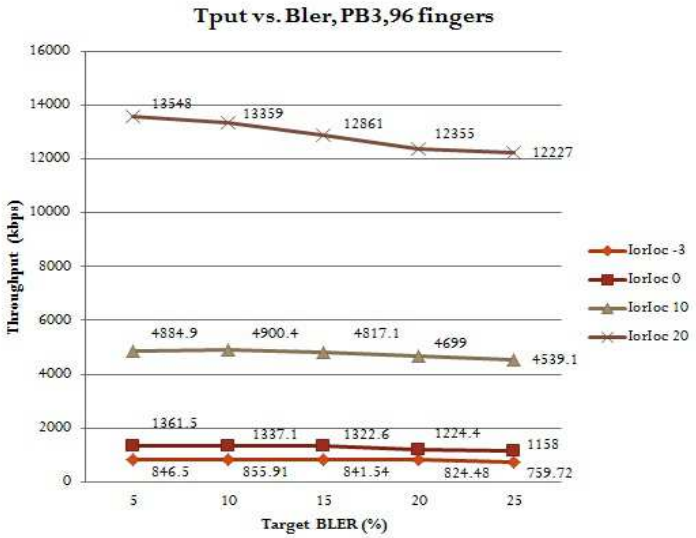


Figure 3.8: Variation in throughput with different BLER.

In a UE in real environment, the SIR to CQI mapping is pre-defined for different environments i.e. there is no adaptive CQI tuning in real implementations. To simulate this behavior the test case was run with follow CQI with adaptive CQI tuning till it converges to the target BLER. The SIR to CQI mapping values at the end of the simulation will form the starting point for next set of simulations which will be run with follow CQI but with no adaptive CQI tuning. To run simulation with a predefined SIR to CQI mapping, set "*aCQI_force_table*" to 1 and the tuned SIR values should be given in "*aCQI_fixed_table"*.

The results are shown in figure 3.8. We get maximum throughput when the target BLER is set at 5-10% for 'G' equalizer with 96 fingers.

3.4.1.2 Smoothing of channel estimates

From the analysis of different equalizer configurations at different fingers, we have seen that P_15 lags well behind performance of G_15. This section covers a recommendation as to how to close the gap between P_15 and G_15 (or P_unused and G_unused).

By default the net channel estimates which are calculated per slot (explained in section 3.2) are not filtered across slots. But we can enable filtering of channel estimates across slots to understand the impact of smoothing of channel estimates on performance of P_15 and P_unused. Equation (40) shows the implementation of filtering (smoothing) of channel estimates across slots.

$$h_{current\_slot}^{filt,est} = \lambda h_{current\_slot}^{est} + (1-\lambda) h_{previous\_slot}^{filt,est} \quad (40)$$

λ is the smoothing filtering parameter.

$h_{current\_slot}^{filt,est}$ is the filtered channel estimate for the current slot which will be used in equalizer weight calculation.

$h_{current\_slot}^{est}$ is the channel estimate for each slot calculated by averaging the 10 channel estimates obtained by despreading and correlating the CPICH symbols. $h_{current\_slot}^{est}$ is the unfiltered channel estimate for the current slot.

$h_{previous\_slot}^{filt,est}$ is the filtered channel estimate for the previous slot.

3.4.1.2.1 Simulation setup

Five different values were tried for smoothing filtering parameter $2^0, 2^{-1}, 2^{-2}, 2^{-4}, 2^{-6}$. $\lambda = 1$ indicates no filtering. Simulations were done for PB3 channel for 48 and 96 fingers at noisy (IorIoc 0) and less noisy (IorIoc 20) environments. Initially the test cases were run with follow CQI with adaptive CQI tuning to obtain optimal mapping from SIR to CQI values for the specified target BLER. Once we had the optimal SIR values, adaptive CQI tuning was turned off and simulations were run with follow CQI using the optimal SIR values.

3.4.1.2.2 Results

Table 3.3 shows the throughput and true SNR for G_15 and for P_15 at different values of λ. We can see that P_15 is closest to G_15 when $\lambda = 0.25$ .We can see considerable increase in performance of P_15 with filtering turned on. With no filtering (λ =1), performance declines for P_15 from 48 to 96 fingers. At λ =0.25, the performance for P_15 levels off from 48 to 96 fingers.

Similarly, table 3.4 shows the throughput and true SNR for G_unused and for P_unused at different values of λ. Performance of P_unused is close to that of G_unused with no filtering but still we can see an improvement in true SNR and throughput of P_unused once we turn on filtering.

Channel	IorIoc	Fingers	Equalizer	smoothing_filt_param	Tput	true_snr	bler
PB3	0	48	G_15		**1399.2**	**15.862**	**15.958**
PB3	0	48	P_15	0.015625	394.31	10.092	21.23
				0.0625	1120.6	15.071	14.681
				0.25	**1323.3**	**15.617**	**13.525**
				0.5	1289.3	15.371	13.302
				1	1116.6	14.536	15.309
PB3	20	48	G_15		**10793**	**31.824**	**17.783**
PB3	20	48	P_15	0.015625	52.261	10.544	99.27
				0.0625	2868.3	21.319	42.785
				0.25	**9163.1**	**30.33**	**15.877**
				0.5	9045.6	29.959	16.972
				1	6306.6	27.399	19.648
PB3	0	96	G_15		**1394.1**	**15.873**	**15.999**
PB3	0	96	P_15	0.015625	431.91	10.137	24.069
				0.0625	1113.4	15.033	14.985
				0.25	**1285.3**	**15.431**	**12.146**
				0.5	1204.3	14.96	13.728

				1	927.33	13.564	18.506
PB3	20	96	G_15		**11159**	**31.933**	**15.816**
PB3	20	96	P_15	0.015625	23.511	10.513	99.635
				0.0625	2819.4	21.199	43.292
				0.25	**9176.4**	**30.019**	**15.715**
				0.5	7988.6	29.055	14.579
				1	6134.7	25.746	15.046

Table 3.3: Throughput and true SNR for G_15 and for P_15 at different values of λ.

Channel	IorIoc	Fingers	Equalizer	Smoothing_filt_param	Tput	true_snr	bler
PB3	0	48	G_unused		**1323.3**	**15.517**	**14.478**
PB3	0	48	P_unused	0.015625	596.35	12.258	15.999
				0.0625	1186.3	15.201	13.809
				0.25	**1303.8**	**15.462**	**13.18**
				0.5	1299.4	15.38	12.835
				1	1216.1	15.061	18.385
PB3	20	48	G_unused		**11868**	**32.831**	**15.086**
PB3	20	48	P_unused	0.015625	4214	25.225	22.366
				0.0625	9926.9	31.392	15.025
				0.25	11356	32.501	15.735
				0.5	**11459**	**32.551**	**13.626**
				1	11508	32.53	16.1
PB3	0	96	G_unused		1137.5	15.19	31.247
PB3	0	96	P_unused	0.015625	608.32	12.051	18.331
				0.0625	1130.4	14.917	23.116
				0.25	**1156.4**	**15.117**	**28.631**
				0.5	1100.7	14.949	32.788
				1	940.63	14.372	45.38
PB3	20	96	G_unused		12858	33.543	15.309
PB3	20	96	P_unused	0.015625	4564.9	25.941	21.25
				0.0625	10906	32.357	13.545
				0.25	12659	33.541	13.221
				0.5	**12768**	**33.585**	**12.126**
				1	12696	33.541	14.66

Table 3.4: Throughput and true SNR for G_unused and for P_unused at different values of λ

3.4.2 MIMO

In this section, the results for simulation of genie and practical equalizer configurations are discussed for a 2x2 MIMO system. Simulations were done in both single stream and dual stream mode to understand how the equalizers perform in each mode.

For MIMO simulations we use CQI_K table defined in 3GPP spec in addition to CQI_G table. In the simulator, CQI_G (Non MIMO) and CQI_K (MIMO) tables are combined into CQI_G_K table. In CQI_G_K table first 31 entries are same as CQI_G and next 15 entries are from CQI_K table. CQI_G_K is given in appendix A. For single stream MIMO operation, we need CQI_G table and for dual stream MIMO operation, we need CQI_K table. The different MIMO simulation parameters and their required values are given in Table 3.5.

Mode	Open/ Closed loop	CQI_table	hsdsch_scheduling_mode	hsdsch_tfrc_sequence	hsdsch_scheduling _modeB	mimo_tx _modes	type_A
Single stream	FRC	CQI_G_K	["Fixed TFRCI"]	["tbl_index1; tbl_index2"] tbl_index1 and tbl_index2 can take any value from 1 to 30. tbl_index1 =tbl_index2.	Prefered mode	0	0
	Follow CQI	CQI_G_K	["CQI based"]	["1 ;1"]	Prefered mode	0	0
Dual stream	FRC	CQI_G_K	["Fixed TFRCI"]	["tbl_index1; tbl_index2"] tbl_index1 and tbl_index2 can take any value from 31 to 45. tbl_index1 >=tbl_index2.	Prefered mode	5	1
	Follow CQI	CQI_G_K	["CQI based"]	["31;31"]	Prefered mode	5	1

Table 3.5: Simulation parameters and required values for different modes of operation of MIMO.

3.4.2.1 Number of fingers

Throughput and true SNR performance for increasing number of fingers were studied separately for single stream and dual stream mode for more dispersive PB3 channel and less dispersive VA3 channel. The performance was studied across five different equalizer configurations as for Non MIMO simulation G, G_15, P_15, G_unused and P_unused. Simulations were done with follow CQI with adaptive CQI tuning. Smoothing of channel estimates for P_15 and P_unused was turned on for MIMO simulations. Smoothing filter parameter was set to 0.25 since this value showed maximum gain in performance for Non MIMO.

Theoretical maximum throughput – Single stream MIMO operation uses the same CQI entries as non MIMO simulation. Hence achievable maximum throughput is same as that for non MIMO. Since CQI 30 is reserved, maximum achievable throughput is 16.1 Mbps for single stream MIMO. For dual stream MIMO operation, entries 31 to 45 in CQI_G_K table (Appendix-A, Table A-3) are used. Hence maximum achievable throughput corresponds to entry 45, which has transport block size of 42192 bits. This translates to achievable maximum throughput of 21 Mbps per data stream. Since in dual stream MIMO, we can transmit two data streams simultaneously, theoretical maximum throughput is 42 Mbps.

3.4.2.1.1 Results

3.4.2.1.1.1 Single stream MIMO

Results can be grouped into those for PB3 channel and for VA3 channel. For PB3 channel, results are as follows,

1) IorIoc 20
a. PB3

The results are shown in figure 3.9. The full genie equalizer 'G', shows gain in throughput and true SNR up to 192 fingers. For single stream MIMO also the RD_unused method outperforms RD_ALL method. G_unused, G_15 and P_unused all peaks at 96 fingers while for P_15 performance levels off from 48 to 96 fingers. G_unused performs better than G at lower fingers. This is due to the fact that genie covariance matrix used in G equalizer for MIMO is an approximation in the sense that it does not model all the terms in equation (39) i.e. cross antenna terms are missing.

Figure 3.9: True SNR and throughput results for Single stream MIMO for Pedestrian B channel at 3 kmph at IorIoc 20.

b. VA3

The results are shown in figure 3.10. In this case genie equalizer 'G' shows gain up to 96 fingers and then levels off. This is expected since VA3 is less dispersive channel than PB3. RD_unused outperforms RD_ALL for VA3 as well. This is similar to the results we saw in Non MIMO simulation where RD_unused outperforms RD_ALL for dispersive as well as less dispersive channel. G_unused, P_unused and G_15 all levels off at 48 fingers and P_15 shows dropping performance after 48 fingers. Here also we can see that G_unused equalizer outperforms G equalizer since the genie noise covariance matrix used in G equalizer is an approximation.

Figure 3.10: True SNR and throughput results for Single stream MIMO for Vehicular A channel at 3 kmph at IorIoc 20.

2) IorIoc 40 (noiseless condition)

a. PB3

The throughput and true SNR results are shown in figure 3.11. The full genie equalizer G shows steady increase in throughput up to 192 fingers. G_unused, P_unused and G_15 shows increase up to 96 fingers and then levels off. P_15 levels off at 48 fingers.

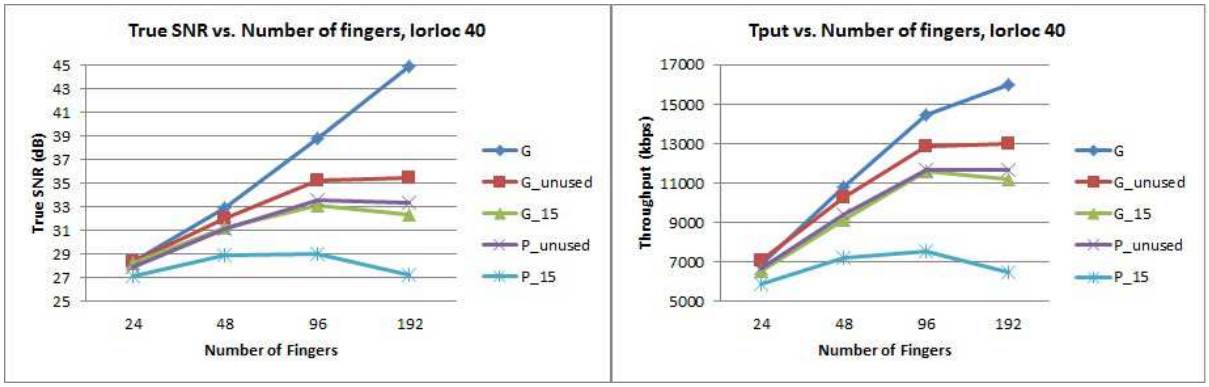


Figure 3.11: True SNR and throughput results for Single stream MIMO for Pedestrian B channel at 3 kmph at IorIoc 40.

b. VA3

The throughput and true SNR results are shown in figure 3.12. Notable difference with IorIoc 20 is the higher gain at IorIoc 40 from 96 to 192 fingers for G equalizer. G_unused, G_15 and P_unused peaks at 96 fingers while P_15 peaks at 48 fingers. G equalizer achieves maximum throughput of 16 Mbps at 192 fingers.

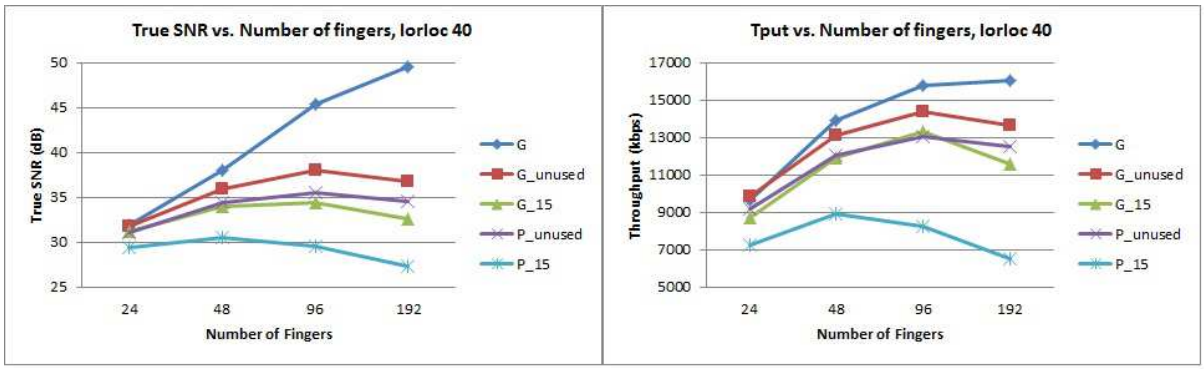


Figure 3.12: True SNR and throughput results for Single stream MIMO for Vehicular A channel at 3 kmph at IorIoc 40.

3.4.2.1.1.2 Dual stream MIMO

For dual stream MIMO operation, CQI_K table is used. All the CQI indexes in CQI_K table allocated all 15 codes for transmission. It makes sense since dual stream transmission is used only when the channel is really good. So for dual stream MIMO, we cannot consider G_unused and P_unused.

1) IorIoc 20

a. PB3

Here also we can see that G_15 outperforms G equalizer at lower fingers. G shows increase in performance up to 192 fingers. G_15 and P_15 peaks at 96 fingers. It is interesting to look at

the throughput gain for P_15 as we increase fingers from 48 to 96 fingers. Increasing the finger count is a good option for dual stream MIMO.

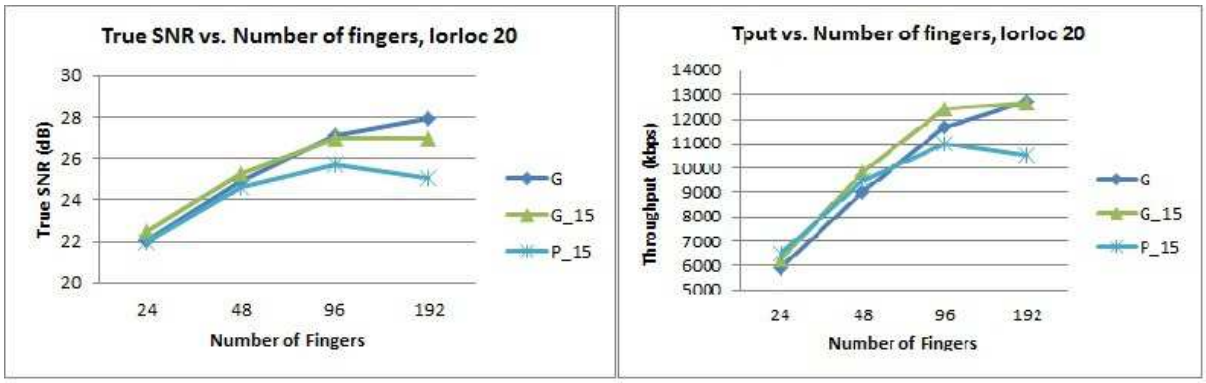


Figure 3.13: True SNR and throughput results for Dual stream MIMO for Pedestrian B channel at 3 kmph at IorIoc 20.

b. VA3

Performance of equalizers is similar but lower than PB3 at higher fingers.

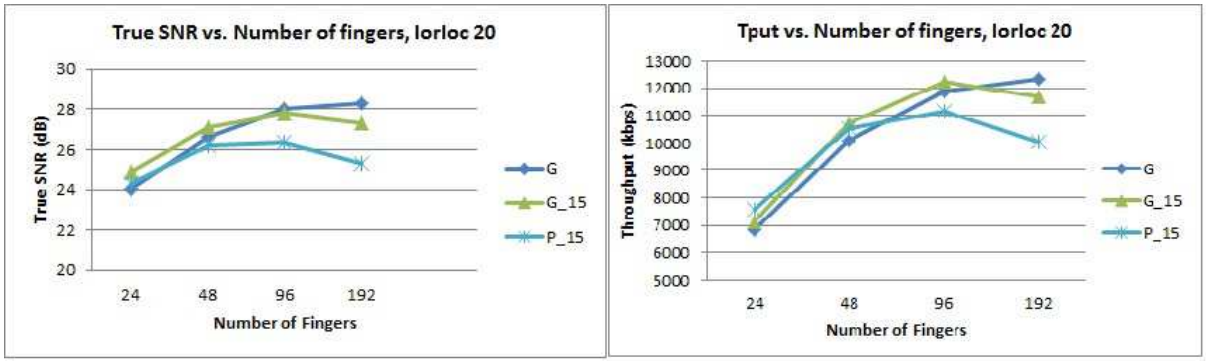


Figure 3.14: True SNR and throughput results for Dual stream MIMO for Vehicular A channel at 3 kmph at IorIoc 20

2) IorIoc 40 (noiseless condition)

a. PB3

G equalizer shows steep increase up to 192 fingers. G_15 also shows increase up to 192 fingers but P_15 levels off at 96 fingers.

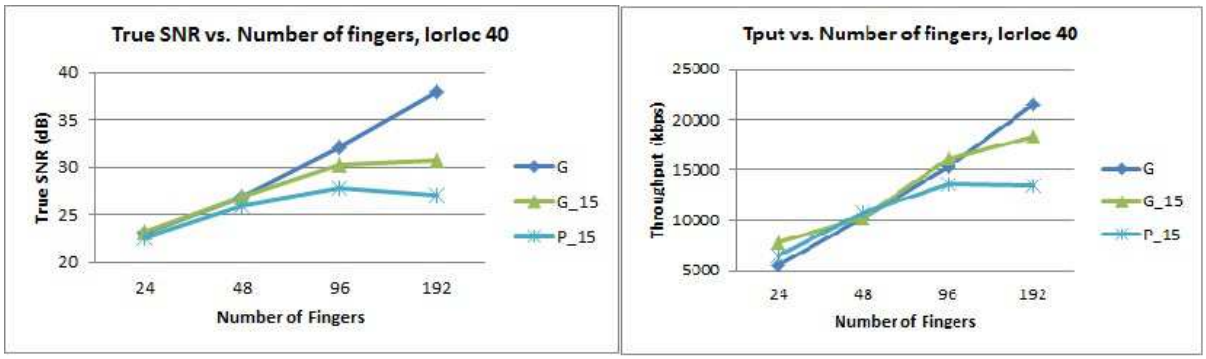


Figure 3.15: True SNR and throughput results for Dual stream MIMO for Pedestrian B channel at 3 kmph at IorIoc 40.

b. VA3

G equalizer shows increase up to 192 fingers. G_15 and P_15 performance peaks at 96 fingers.

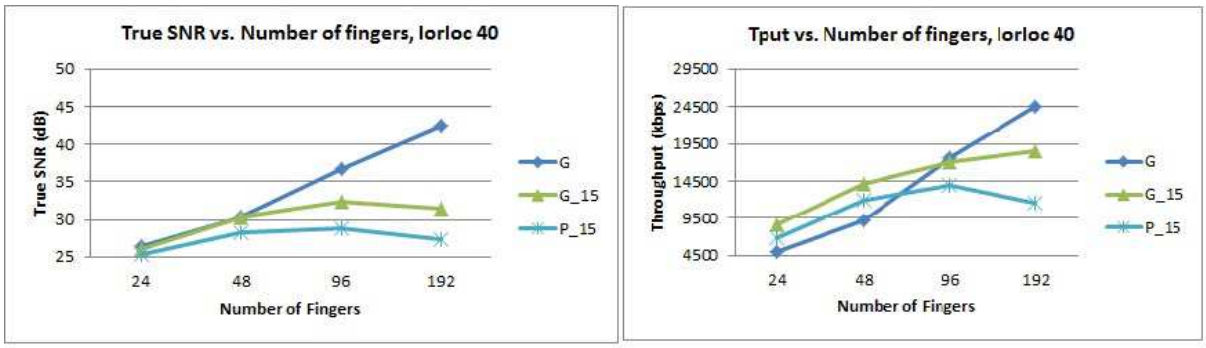


Figure 3.16: True SNR and throughput results for Dual stream MIMO for Vehicular A channel at 3 kmph at IorIoc 40.

3.4.2.2 Comparison of MIMO performance with Non MIMO

In this section we compare the performance of single stream MIMO (transmit diversity and receiver diversity) and non MIMO with two receiver antennas (no transmit diversity but with receiver diversity). We expect the MIMO throughput to be higher but in reality it depends on how dispersive the channel is. Here SIMO refers to non MIMO channel with two receiver antennas and SS_MIMO refers to single stream MIMO.

The results for simulation are shown in table 3.6. The simulations were done initially for PB3 channel. The throughput for SS_MIMO was less than SIMO for PB3 channel. Simulations were then done for less dispersive channels like PA3 and for non-dispersive single path channel. We can see that P_15 equalizer configured in SS_MIMO for PA3 and Single Path (3 kmph) outperforms P_15 equalizer in SIMO environment.

So for SS_MIMO we need better equalizers to improve performance in high dispersive channel. The current equalizer cannot cope with the additional interference due to transmit diversity in addition to the interference created by the high dispersive channel.

Channel	Equalizer	MIMO / Non MIMO	Fingers	Tput	True SNR	BLER
Single Path	G	SS_MIMO	16	17362	38.083	9.9278
	G	SIMO	16	14984	35.951	21.416
	P_15	SS_MIMO	16	11999	33.614	18.551
	P_15	SIMO	16	11783	32.943	15.712
PA	G	SS_MIMO	16	15644	36.654	18.476
	G	SIMO	16	14706	35.598	22.776
	P_15	SS_MIMO	16	11167	32.792	16.415
	P_15	SIMO	16	10978	32.521	30.914
PB3	G	SS_MIMO	48	7905.2	29.711	13.965
	G	SIMO	48	12212	33.48	20.561
	P_15	SS_MIMO	48	6401.6	27.739	16.553
	P_15	SIMO	48	9228	30.421	19.171

Table 3.6: Throughput figures in kbps for single stream MIMO (SS-MIMO) and non MIMO with two receiver antennas (SIMO) for different channels for 'G' and 'P_15' equalizer configurations.

3.4.2.3 When to increment number of RAKE fingers?

In this section we try to understand when it more prudent to use more RAKE fingers for P_15 equalizer. We have seen that as we increase the number of RAKE fingers from 48 to 96 for PB3 channel, throughput might go up or down. A good way to understand when to use more fingers for P_15 is to study the throughput gain shown by full genie 'G' equalizer as we increase more fingers. If 'G' equalizer shows sufficiently good slope (in throughput) as we go from 48 to 96 fingers then P_15 will also show performance gain from 48 to 96 fingers.

Figure 3.17 shows the throughput performance of different equalizer configurations at increasing finger counts. Smoothing for channel estimates is turned on for P_15 equalizer for MIMO and non MIMO. The slope for genie 'G' equalizer for non MIMO decreases as we progress from 48 to 96 fingers. We can see that P_15 equalizer shows decrease in throughput as we increase fingers from 48 to 96. For single stream MIMO, the slope for 'G' equalizer from 48 to 96 is quite at the same level as from 24 to 48. P_15 equalizer performance levels off at 48 fingers. Dual stream shows the highest slope when compared to single stream MIMO and non MIMO for 'G' equalizer. For dual stream MIMO, P_15 equalizer shows increase in throughput as we go from 48 to 96 fingers.

So a good rule of thumb is that if full genie 'G' equalizer shows sufficiently good slope as we move to higher fingers then it is beneficial to use higher number of fingers at P_15.

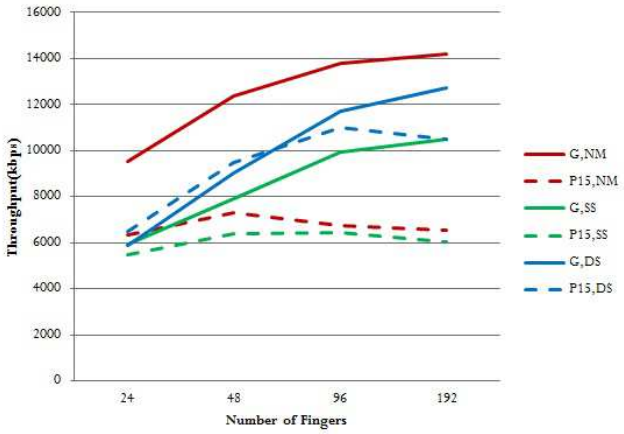


Figure 3.17: Throughput vs. number of fingers for G and P_15 for non MIMO and MIMO implementation. In legend 'N' stands for non MIMO, 'SS' stands for single stream MIMO and 'DS' stands for dual stream MIMO. Simulation results are for PB3 channel at IorIoc 20 and 2 receiver antennas.

4 Conclusion

In this thesis we have evaluated the performance of commercial practical equalizer implementation by comparing against ideal equalizer implementation. The simulations were done in ST-Ericsson's WCDMA simulation bench. Practical equalizers can be based on two different methods of correlation matrix calculation – RD_unused and RD_ALL. In both methods the correlation matrix is calculated from depsreaded data symbols. For RD_unused method, data correlation matrix is generated from an unused code and for RD_ALL method data correlation matrix is generated using all codes. The equalizers were evaluated based on the true SNR and throughput performance.

First part of thesis comprised of analyzing the performance of different equalizers at increased number of RAKE fingers. RD_unused method was found to outperform RD_ALL method. Even with genie channel estimates RD_ALL method could not catch up with performance of RD_unused for test cases with 2 receiver antennas. For test cases with one receiver antenna, RD_ALL with genie channel estimates reaches performance of RD_unused.

An important result from the thesis study was that by smoothing of channel estimates the performance of equalizer method with practical channel estimates can be brought closer to the performance of same equalizer method with genie channel estimates. Smoothing of channel estimates across slots was found to be an effective method to improve the throughput performance for practical equalizers based on RD_unused (P_unused) or RD_ALL (P_15).

For highly dispersive channels, full genie equalizer ('G') showed increase in throughput performance even up till 192 fingers. But for practical equalizer implementation based on RD_unused and RD_ALL method, the peak performance occurred earlier than 192 fingers. Practical equalizer based on RD_unused method (P_unused) reached peak performance at 96 fingers. Practical equalizer based on RD_ALL method (P_15) reached peak performance at 48 fingers or earlier except for dual stream MIMO system for which there was gain up to 96 fingers. Low dispersive channels like PA showed reduction in throughput with more fingers. A good rule of thumb to find whether to use higher number of fingers for practical implementation is to check the gain in throughput for genie equalizer ('G') at more number of fingers. A steady increase in throughput for full genie equalizer 'G' with more fingers will suggest increase in throughput for practical equalizer as well.

It was found that the genie and practical equalizer methods suffer at highly dispersive channel for a MIMO channel while compared to non MIMO channel. Simulation in a PB3 environment for single stream MIMO showed less throughput when compared to same simulation done for non MIMO. But for less dispersive channels like PA and single path single stream MIMO delivered better throughput than non MIMO as expected. This result suggests that we need better equalizers to tap the benefit of MIMO channel in a highly dispersive environment.

Appendix – A

CQI tables

CQI Index	Transport block size (bits)	Number of codes	Modulation
0	136	0	4
1	136	1	4
2	176	1	4
3	232	1	4
4	320	1	4
5	376	1	4
6	464	1	4
7	648	2	4
8	792	2	4
9	928	2	4
10	1264	3	4
11	1488	3	4
12	1744	3	4
13	2288	4	4
14	2592	4	4
15	3328	5	4
16	3576	5	16
17	4200	5	16
18	4696	5	16
19	5296	5	16
20	5896	5	16
21	6568	5	16
22	7184	5	16
23	9736	7	16
24	11432	8	16
25	14424	10	16
26	15776	10	64
27	21768	12	64
28	26504	13	64
29	32264	14	64
30	38576	15	64

Table A-1: CQI_G table defined in 3GPP 25.214

CQI Index	Transport block size (bits)	Number of codes	Modulation
0	136	0	4
1	136	1	4
2	176	1	4
3	232	1	4
4	320	1	4
5	376	1	4
6	464	1	4
7	640	2	4
8	776	2	4
9	1096	3	4
10	1136	4	4
11	1744	5	4
12	2288	7	4
13	2832	9	4
14	3640	12	4
15	4592	15	4
16	5392	15	4
17	6336	15	4
18	7440	15	4
19	8744	15	4
20	9736	15	4
21	10840	15	4
22	12064	15	16
23	14168	15	16
24	15776	15	16
25	17568	15	16
26	21768	15	16
27	25120	15	64
28	29504	15	64
29	34656	15	64
30	38576	15	64

Table A-2: CQI_Normal_G table

CQI index	Transport block size (bits)	first channel code	Number of codes	Modulation format	Ec_hsdsch_offset_dB
0	136	10	0	4	0
1	136	10	1	4	0
2	176	10	1	4	0
3	232	10	1	4	0

4	320	10	1	4	0
5	376	10	1	4	0
6	464	10	1	4	0
7	648	10	2	4	0
8	792	10	2	4	0
9	928	10	2	4	0
10	1264	10	3	4	0
11	1488	10	3	4	0
12	1744	10	3	4	0
13	2288	10	4	4	0
14	2592	10	4	4	0
15	3328	10	5	4	0
16	3576	10	5	16	0
17	4200	10	5	16	0
18	4672	10	5	16	0
19	5296	10	5	16	0
20	5896	10	5	16	0
21	6568	10	5	16	0
22	7184	10	5	16	0
23	9736	8	7	16	0
24	11432	7	8	16	0
25	14424	5	10	16	0
26	15776	5	10	64	0
27	21768	3	12	64	0
28	26504	2	13	64	0
29	32264	1	14	64	0
30	38576	1	15	64	0
31	4592	1	15	4	-3
32	4592	1	15	4	-1
33	5296	1	15	4	0
34	7312	1	15	4	0
35	9392	1	15	4	0
36	11032	1	15	4	0
37	14952	1	15	16	0
38	17880	1	15	16	0
39	21384	1	15	16	0
40	24232	1	15	16	0
41	27960	1	15	64	0
42	32264	1	15	64	0
43	36568	1	15	64	0
44	39984	1	15	64	0
45	42192	1	15	64	0

Table A-3: CQI_G_K table

Appendix - B

Find weight $\boldsymbol{w} = \begin{bmatrix} w_{-M} & \cdots & w_0 & \cdots & w_M \end{bmatrix}^T$ that $\min_{\bar{w}} E\left\{ \left| \sum_{j=-M}^{M} y_j w_j^* - s_0 \right|^2 \right\}$

To solve this expression, we have to solve

$$\frac{\partial}{\partial w_k} E\left\{ \left| \sum_{j=-M}^{M} y_j w_j^* - s_0 \right|^2 \right\} = 0 \tag{41}$$

where k varies from $-M$ to M .

Let's define error e as,

$$e = \sum_{j=-M}^{M} y_j w_j^* - s_0$$

Equation (41) reduces to,

$$\frac{\partial}{\partial w_k} E\{ee^*\} = 0$$

$$E\left\{ e\frac{\partial e^*}{\partial w_k} + e^* \frac{\partial e}{\partial w_k} \right\} = 0 \tag{42}$$

If $w_k = a_k + ib_k$ then $\frac{\partial}{\partial w_k} \triangleq \frac{\partial}{\partial a_k} - i\frac{\partial}{\partial b_k} \therefore \frac{\partial w_k}{\partial w_k} = 2$ and $\frac{\partial w_k^*}{\partial w_k} = 0$ (43)

$$\begin{aligned} \frac{\partial e^*}{\partial w_k} &= \frac{\partial}{\partial w_k}\left(\sum_{j=-M}^{M} y_j^* w_j - s_0^* \right) \\ &= \sum_{j=-M}^{M} y_j^* \frac{\partial}{\partial w_k} w_j - \frac{\partial}{\partial w_k} s_0^* \\ &= 2y_k^* \qquad \left(\because \frac{\partial w_k}{\partial w_k} = 2 \right) \\ \frac{\partial e}{\partial w_k} &= \frac{\partial}{\partial w_k}\left(\sum_{j=-M}^{M} y_j w_j^* - s_0 \right) \\ &= \sum_{j=-M}^{M} y_j \frac{\partial}{\partial w_k} w_j^* - \frac{\partial}{\partial w_k} s_0 \\ &= 0 \qquad \left(\because \frac{\partial w_k^*}{\partial w_k} = 0 \right) \end{aligned}$$

$$E\left\{e\frac{\partial e^*}{\partial w_k}+e^*\frac{\partial e}{\partial w_k}\right\}=2E\left\{e\ y_k^*\right\}$$

$$=2\sum_{j=-M}^{M}w_j^*E\{y_j y_k^*\}-2E\{s_0 y_k^*\}$$

Applying equation (42) we get,

$$\sum_{j=-M}^{M}w_j^*E\{y_j y_k^*\}=E\{s_0 y_k^*\}$$

References

[1] WCDMA High-Speed Downlink Shared Channel Technical Description, Per Beming, Eric Dahlman et al.

[2] Bottomley, G. E., Ottosson, T., and Wang,Y.-P. E.: A generalized RAKE receiver for interference suppression. IEEE Journal on Selected Areas in Communications, Vol. 18 (2000):8, pp. 1536-1545

[3] WCDMA FOR UMTS Third Edition, Harri Holma and Antti Toskala.

[4] 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; User Equipment (UE) radio transmission and reception (FDD) (Release 9); 3GPP TS 25.101 V9.4.0 (2010-06)

[5] Universal Mobile Telecommunications System (UMTS); Physical layer procedures (FDD) (3GPP TS 25.214 version 9.8.0 Release 9)

[6] An introduction to numerical analysis, Kendall Atkinson