AN ABSTRACT OF THE THESIS OF

Joseph Sam Lesmana for the degree of Master of Science in Electrical and Computer Engineering presented on March 2, 2005.

Title: Performance of MC-CDMA with Pilot Code and Blind Equalization Algorithm.

Abstract approved:

Mario E. Magaña

Various methods and techniques have been introduced and applied to advance the state of the art of mobile wireless communications technology. For example, some techniques are applied to overcome the problem of multipath caused by the mobile environment. Multipath produces replicas of the wanted signal which arrive at the receiver with different time delays. If not dealt with properly, this environment will greatly deteriorate the quality of the wanted signal. The so-called multiuser feature of many wireless communication systems will also add some interference to the signal of interest.

This thesis makes an attempt to improve the performance of wireless communication systems that use either pilot-based or blind equalization techniques to obtain channel side information. Specifically, these techniques are concerned with the estimation of multipath parameters in order to improve system performance. By inserting some pre-defined code which is called a pilot code on each pre-defined segment of a data block, we can recover the signal by pilot-based methods. Using an adaptive method, some knowledge of the channel characteristics and input source, we can achieve acceptable error rate using blind equalization as an alternate solution

to pilot-based. Finally, new enhancements are added to blind equalization to improve its performance further.

Performance of MC-CDMA with Pilot Code and Blind Equalization Algorithm

by

Joseph Sam Lesmana

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented March 2, 2005
Commencement June 2005

Master of Science thesis of Joseph Sam Lesmana presented on March 2, 2005

APPROVED:

____

Major Professor, representing Electrical and Computer Engineering

____

Director of the School of the Electrical Engineering and Computer Science

____

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

____

Joseph Sam Lesmana, Author

ACKNOWLEDGEMENTS

The word “Thank You” is the first one that comes to my mind when I finished this thesis. I would to thank “God” for His support and gift through the people in my surroundings. I thank Him for granting my wish.

My parents, Bernadus Djunaedi and Nelly Gunawan are the ones to be thank for their spiritual and material support. Their sacrifices and generosity helps to strengthen me in my education progress. I also thank my Sister, Cicilia Dwi Yuanita, and my Brother, Ignatius Adi Yulianto, for the laughter that cherishes me.

I would like to thank my major advisor, Dr. Mario E. Magaña, for his invaluable educational insight, guidance and support. The criticism and input which made by him helps me grow and see more in scientific world.

I also would like to thank the authors of papers for their quick responses on my email to help my thesis work out. Thomas Zemen, Simone Fiori and Francesco Verde are these people.

TABLE OF CONTENTS

LIST OF FIGURES

PERFORMANCE OF MC-CDMA WITH PILOT CODE AND BLIND EQUALIZATION ALGORITHM.

1. INTRODUCTION

Cellular telephony has undergone a tremendous growth since its inception in 1976. One of the first mobile radio services was established in New York City, and could only accommodate 543 users [36,38]. This is in contrast to today's Personal Communications Systems (PCS), which use digital technology to support a great many users. Today, the so-called Third Generation (3G), which is a mobile wireless communication system that involves internet access and multimedia, is close to being realized. The increasing number of users gives rise to some issues. For example, when many users are present in the system, their signals utilize common shared resources and result in Multiple Access Interference (MAI) and co-channel interference. These issues will bring the bit error rate performance of a communication system down. Moreover, the mobile terrestrial environment produces multipath interference.

In order to resolve the problem of multipath, a good candidate is Orthogonal Frequency-Division Multiplexing (OFDM) technique, which was pioneered by R.W.Chang. Later, he patented it. Given a spectrum bandwidth using a single carrier at a high rate, there is a chance that the signal will suffer Intersymbol Interference (ISI) because the spectrum is bigger than the coherence bandwidth [35,37,43]. If we could separate the signals and transmit them in parallel at lower speeds using multiple carriers, instead of transmitting them in serial mode, we have a chance to make each carrier experience a non-frequency selective channel. That is, the coherence bandwidth is larger than the signal's spectrum. Ove Edfors et al [4] provide some ideas on how OFDM can be implemented to serve this purpose.

To improve performance, we could insert a pilot signal on the user's signal frame [1,13]. The pilot signal will help us to estimate the channel condition. Having pilot symbols on our system lead us to choose a good filter for interpolation, and give us the best estimation of the channel condition. Either the Wiener filter, Minimum Mean Squared Error (MMSE) or other kinds of advanced filters will do well for this purpose.

Although people still try to use it, OFDM technology alone cannot be used efficiently in today's communication market. In order to accommodate the abundance of users, Code Domain Multiple Access (CDMA) is used in many systems. Using CDMA, users theoretically mounted on the same spectrum, can be identified uniquely and are immune to jamming. The combination of the two techniques gives us what is called Multi-Carrier CDMA (MC-CDMA). Lindner, J. [5,7] elaborates on this in his conceptual oriented paper and gives the feasibility of this technology to be used in the new trend. He created the general channel model into what he called the '*channel matrix*'. This channel matrix comprises channel modeling and is a general technique of multiplexing. On this thesis, a generalized form of an OFDM and MC-CDMA matrix is presented. With powerful computing capabilities, this technique could be implemented nowadays, such as in digital audio broadcasting (DAB), asymmetric digital subscriber (ADSL), and wireless personal communication (PCS). However, issues regarding the orthogonality of the carrier become important, since system performance deteriorates rather fast if orthogonality fails.

MC-CDMA falls into the category of systems with a training code. One of the drawbacks of these systems is their lack of power efficiency, since we insert a training pilot code into each of the sub-carriers. Blind equalization [26] offers another alternative to estimate channel impulse response, without the use of a training code. Instead, it estimates the state of the channel based on channel output measurements and the statistical behavior of the input. So far, there have been three blind deconvolution methods developed. They are the Bussgang [18,22] algorithm, the polyspectra and cumulant-based [22] algorithm, and the probabilistic [23,25] algorithm. The Bussgang algorithm is used in this thesis because of its simplicity. Although such systems promise a better efficiency from a power point of view, most of the systems assume a channel impulse response model especially designed for the system. Hence, it might not work for every channel model.

For the purposes of this thesis, a mobile terrestrial environment is assumed. Simulation will be based on multiple transmission signals from mobile users, while a base station with multiple antennae acts as a receptor. The Additive Gaussian White Noise (AWGN) channel model as well as the outdoor Rayleigh fading channel model are used in the simulation. Chapter 2 elaborates on MC-CDMA systems and their application under AWGN and Rayleigh fading. Chapter 3 introduces its counterpart, Blind equalization with Bussgang algorithm. Unlike MC-CDMA simulation, Bussgang algorithm will be simulated under channel impulse response especially designed for the system. Achieving a lower bit error rate for a given energy bit as a reference is the goal of this thesis. Simulations are carried out using the Matlab software package [42].

2. MC-CDMA UNDER FREQUENCY SELECTIVE FADING WITH JOINT CHANNEL ESTIMATION AND ITERATIVE DETECTION

2.1. MC-CDMA in a nutshell

Multiple accesses (MA) arose to accommodate a high number of users, since the demand for mobile communication usage has greatly increased in recent years. One of the popular MA strategies is Coded Division Multiple Access (CDMA). CDMA assigns a unique almost orthogonal code for each user so they can share the same bandwidth and still permit signal recovery at the receiving end. Using CDMA offers the added advantage of protection from jamming. With the user's signal being spread in frequency, a narrow-band interferer signal would partially distort the signal [33,44].

To improve performance in a real-world multipath environment, a multi-channel (MC) capability can be added. This multipath channel enhancement is introduced because of the nature of the mobile transmission medium. A single signal can be reflected by obstructions, which in turn create attenuated and delayed replicas of the original signal. These copies, including the original signal, arrive with different time delays at the receiver. An aggregate of these replicas appears at the receiver's front end to form the resultant signal.

Multipath causes the CDMA signal to experience frequency-selectivity. Here the signal experiences different amounts of attenuation throughout its bandwidth. OFDM can alleviate such condition by separating the signal into several narrowband subchannels. This way they should experience flat fading instead of frequency selective fading. There are several variations of OFDM. The variation which will be used in this thesis is the one that uses a cyclic prefix. A cyclic prefix is a copy of the last part of the OFDM symbol which is added in front of the transmitted symbol. The usage of cyclic prefix in OFDM can alleviate the ISI problem. Figure 2.1 depicts the Discrete-time OFDM system. However, we assume that the cyclic prefix is longer than the channel impulse response and synchronization is perfect.

Now, let **x** be the information source sequence,i.e, **x** = $[\ldots x_{-2}, x_{-1}, x_0, x_1, x_2, \ldots]$, let **s** be the transmitted signal, **n** be Additive White Gaussian Noise (AWGN), **r** the received signal, **g** the channel impulse response and **y** the estimate of the transmitted signal.

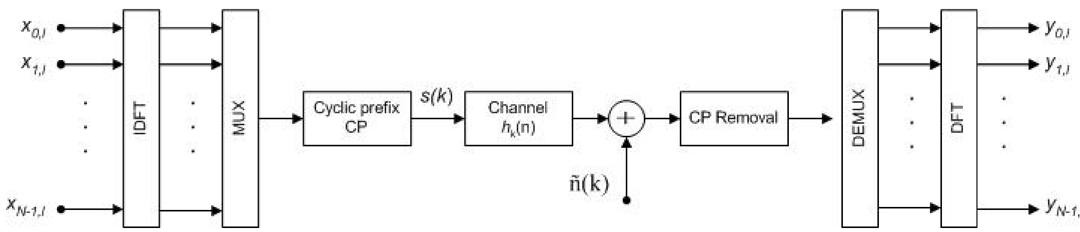


Figure 2.1: Discrete-time OFDM system block diagram

The length ***N*** is chosen as the number of subcarriers. Each of N-bit long data sequences **x** is converted from serial to parallel before applying the Inverse Discrete Fourier Transform (IDFT). Hence, **x** becomes $[\ \ldots\ \mathbf{x}_{-1}, \mathbf{x}_0, \mathbf{x}_1\ \ldots]$, where $\mathbf{x}_l = [x_{0,l} \ldots x_{N-1,l}]$.

Now, $\mathbf{y}_l$ = DFT(IDFT($\mathbf{x}_l$) * g_l) + $\mathbf{n}_l$). However, the DFT does not change the statistical properties of AWGN.

Hence, in the frequency domain,

$$\mathbf{y}_l = \mathrm{DFT}(\mathrm{IDFT}(\mathbf{x}_l) * g_l)) + \mathbf{n}_l.$$

$$\mathbf{y}_l = \mathrm{DFT}(\mathrm{IDFT}(\mathbf{x}_l)\ ) \cdot G_l + \mathbf{n}_l.$$

$$\mathbf{y}_l = \mathbf{x}_l \cdot G_l + \mathbf{n}_l.$$

By adopting the idea of OFDM, MC-CDMA assigns a unique pseudo-code to each user of the system. Just like Direct-Sequence CDMA(DS-CDMA), the code will spread the user's signal. Given the subcarrier of length N, we use a pseudo-code of the same length to spread the signal and then use IDFT to place each spread code to its corresponding subcarrier.

For example, x_t is spread with $\boldsymbol{s} = [s_0,\ldots,s_{N-1}]$, hence, we obtain a vector of $N \times 1 = [x_t.s_0,\ldots, x_t.s_{N-1}]$ elements. Then the IDFT will place the first spread code ($x_t.s_0$) on the first subcarrier ($f_c + \Delta f_0$). The second spread code ($x_t.s_1$) is placed on the second subcarrier ($f_c + \Delta f_1$), and so on. Here, we refer to f_c as the carrier frequency in passband representation and $\Delta f_n = {}^{W}\!/\!_{N}\, n, \quad n = 0,1,..,N-1$, where W represents the signal bandwidth. It is important to understand that application of the IDFT will make subchannels overlap. There are other kinds of subchannels which do not overlap.

2.2. MC-CDMA transmitter block set with pilot code

After briefly describing MC-CDMA, let us look at the details of the proposed transmitter system more thorough. A pilot based MC-CDMA transmitter is shown in figure 2.2 [1]. Without loss of generality, we consider only its baseband representation. The system will use binary phase shift keying (BPSK). The signal of user k (b_k(m)) packs in a block of length M. For the purpose of obtaining pseudo Walsh code (s_k) , the following constraint has to be satisfied:

$$\sum_{n=0}^{N-1} \left| s_k(n) \right|^2 = 1 \qquad \forall k. \tag{2.1}$$

where k represents the index number of users. In order to satisfy the above condition, we need to make sure that our Walsh code remains within the bounded interval $(-{}^{1}\!/\!_{\sqrt{N}}, +{}^{1}\!/\!_{\sqrt{N}})$.

In the block of length M, the first J symbols are dedicated to pilot training codes. These codes could be generated using a Walsh code too, and satisfy formula (2.1). The next M-J symbols are populated with the user's signal which is convolutionally coded, randomly interleaved and originated from (M-J)·R binary information source ($d_k(m)$). R denotes the code rate.

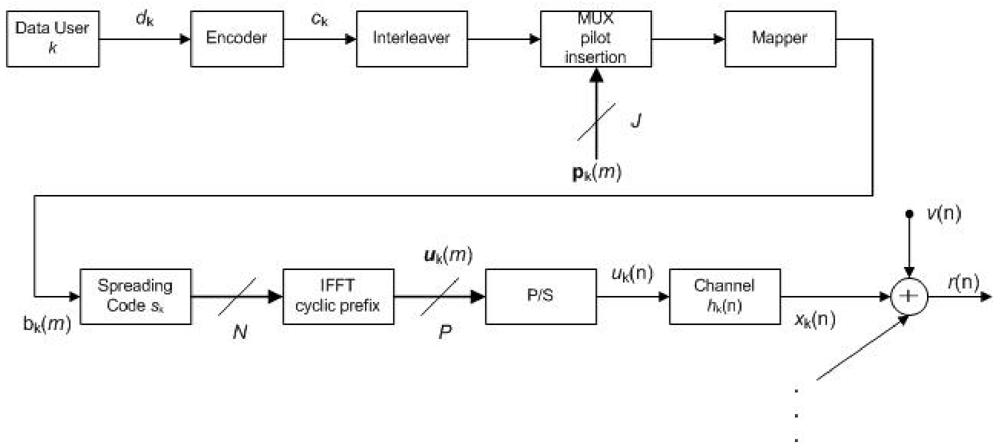


Figure 2.2: MC-CDMA transmitter blockset.

Using a Walsh spreading code (s_k), the block is spread, converted to parallel form and input to the OFDM block. This gives us a vector $\mathbf{s}_k \cdot b_k(\mathrm{m}) a_k$ of N × 1 elements, where a_k is the user's amplitude. Here, we can assume $a_k = 1$ for all k.

Given the length of the channel impulse responses as L, we add a cyclic prefix of length G to the front of the block where G = P – N ≥ L. Now let $\boldsymbol{u}_k(\mathrm{m})$ = [u(mP), u(mP+1), …, u(mP+P-1)]^T, be a vector of P × 1 elements after the cyclic prefix operation.

We can write $\boldsymbol{u}_k(\mathrm{m})$ as

$$\boldsymbol{u}_k(\mathrm{m}) = \boldsymbol{T}_{cp}\boldsymbol{F}^{\boldsymbol{H}}\boldsymbol{s}_k b_k(\mathrm{m}) a_k. \quad (2.2)$$

Matrix $\boldsymbol{F}$ is a unitary $N \times N$ Fourier matrix with elements

$$F_{i,k} = \frac{1}{\sqrt{N}} e^{\frac{-j2\pi ik}{N}}, \qquad \mathrm{i, k = 0 .... N - 1.} \quad (2.3)$$

$\boldsymbol{F}^H$ is the inverse Fourier matrix, and $(.)^H$ denotes the Hermitian transpose. $\boldsymbol{T}_{cp} = [\boldsymbol{I}_{cp}^T, \boldsymbol{I}_N^T]^T$, performs cyclic prefix operation, and $\boldsymbol{I}_{cp}$ gives the last G rows of the identity matrix $\boldsymbol{I}_N$. As soon as $\boldsymbol{u}_k$ is formed, the signal is transmitted through the multipath channel. This channel model is discussed in section 2.3.

2.3. Channel Model

A multipath channel is chosen as the preferred model for our communication system in order to show the effectiveness of MC-CDMA modulation technique. There are two parameters that classify the multipath channel. They are coherence bandwidth $\boldsymbol{BW}_c$ and coherence time $\boldsymbol{T}_c$.

Should the coherence bandwidth be smaller than the signal's bandwidth, the channel will be categorized as frequency selective. On the other hand, if the coherence bandwidth is larger than the signal's bandwidth, then it falls into the frequency non-selective type. Coherence bandwidth can be represented in the time domain, which is called delay spread $\sigma_d = 1/(2\pi\ \boldsymbol{BW}_c)$. Delay spread is the difference in time between the earliest and latest reflected signal that arrives at the receiver front end. This delay spread will depend on the channel impulse response length as we use it in later sections. The number of resolvable paths is directly proportional to the parameter α, which is the fraction of the signal's bandwidth over coherence bandwidth.

Now, if the receiver (R_x) and transmitter (T_x) are mobile units with R_x either moving toward or away from T_x, there will be some frequency shift according to the Doppler Effect. By moving toward or away, the carrier frequency of the signal will experience Doppler shift. Since the original signal is reflected by obstructions, it creates some replicas of itself. The carrier frequency of the replicas that are moving away from Rx will be smaller than that of original signal. On the other hand, the carrier frequency of the replicas that are moving toward to Rx will be larger than that of original signal. The difference between the lowest and highest shifts will be called Doppler Spread. Representation of Doppler Spread ($\boldsymbol{B}_d$) in the time domain is called coherence time $\boldsymbol{T}_c \approx 9/(16\pi\ \boldsymbol{B}_d)$.

Hence, if the signal duration (***T**$_s$*) is smaller than ***T**$_c$*, then the signal will pass through the channel without substantial alteration. On the other hand, should ***T**$_s$* >> ***T**$_c$*, then the signal will undergo some changes and the channel is called the frequency dispersive channel.

We can use Jakes' Model to simulate Rayleigh fading. Here, it is assumed that there is no Line Of Sight (LOS) component. In the case when the LOS is present, then, in lieu of Rayleigh Fading we should use Rician Fading. This technique suggests that a sufficiently large number of sinusoids can add to give an envelope ($|C(t)|$) whose PDF approximates the Rayleigh PDF [39]. The complex envelop (***C***(t)) itself will be the resultant signal of one complex frequency oscillator with frequency of ω_m plus a summation of S_0 complex lower-frequency oscillator with frequency of ω_n. The complex envelope can be expressed as follows:

$$C(t) = \frac{E_0}{\sqrt{2S_0 + 1}}\left[x_c + jx_s\right], \qquad (2.4)$$

where $x_c(t)$ and $x_s(t)$ can be expressed as follows:

$$x_c(t) = 2\sum_{n=1}^{N_0} \cos\phi_n \cos\omega_n t + \sqrt{2}\cos\phi_N \cos\omega_m t \qquad (2.5a)$$

$$x_s(t) = 2\sum_{n=1}^{N_0} \sin\phi_n \cos\omega_n t + \sqrt{2}\sin\phi_N \cos\omega_m t \qquad (2.5b)$$

$$\omega_n = \omega_m \cos(2\pi n / S), n = 1,2,\ldots,S_0$$

$$\omega_m = 2\pi f_d = 2\pi(fc/v)$$

$$S = 2(2S_0+1)$$

$$\phi_N = 0$$

$$\phi_n = \pi n/(S_0+1).$$

where c is the speed of light, v is vehicle speed, $E_o = 1$, $f_d = (fc/v)$ is the maximum Doppler Shift and f is the carrier frequency.

The number of S_0 complex lower-frequency oscillators is chosen as a parameter to create the complex envelope. Jakes suggests that $S_0 = 8$ should satisfy the following statistical properties:

$$E\{X_c^2(t)\} = S_0$$
$$E\{X_s^2(t)\} = S_0 + 1$$
$$E\{|C(t)|^2\} = C_0 = 1.$$

One important property that must be noted in this fading propagation model is

$$\Phi(\Delta t) = E\{C(t)C^*(t+\Delta t)\} = J_0(2\pi f_d \Delta t),$$

where J_0 is a zeroth order Bessel function of the first kind and $()^*$ denotes the complex conjugate.

Should we need to build a frequency selective fading channel using this model, we need a tapped-delay line model [38] (figure 2.3).

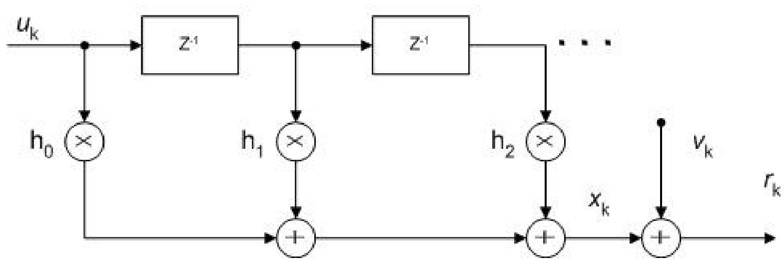


Figure 2.3. Tapped-delay line model for frequency selective channel

The number of delay elements (L) can be chosen as $\left\lfloor \frac{W}{BW_c} \right\rfloor + 1 = \lfloor \alpha \rfloor + 1.$

The signal, after being convolved with the channel impulse response of the discrete model, can be written as

$$x(n) = h(n)_k \otimes u_k(n) = \sum_{l=0}^{L-1} h_k(l) \cdot u_k(n-l). \tag{2.6}$$

Here h(n) denotes channel impulse response and $\otimes$ denotes convolution.

2.4. MC-CDMA receiver block set

Each signal, either the original or a reflected one, will pass through its own frequency selective Rayleigh fading channel. Before the signals arrive at the demodulator, the composite signal will pass through an AWGN channel (figure 2.2). Hence, the composite signal at receiver's front end can be modeled as

$$r(n) = \sum_{k=1}^{K} x_k(n) + v(n), \tag{2.7}$$

where v(n) is AWGN with variance of σ_v^2. Now, r(n) can be extracted by cyclic prefix removal and Fast Fourier Transform (FFT). Borrowing from [1] we can write $\boldsymbol{y}_k$(m) as

$$\mathbf{y}_k = \mathrm{diag}(\mathbf{g}_k)\mathbf{s}_k b_k(m) + \bar{v}(m), \tag{2.8}$$

where $\boldsymbol{g}_k$ is $\boldsymbol{h}_k$ in frequency domain, namely $\mathbf{g}_k = \sqrt{N}\mathbf{F}\mathbf{h}_k$

If we define $\tilde{\boldsymbol{s}}_k = \mathrm{diag}(\boldsymbol{g}_k)\boldsymbol{s}_k$, we can write eq. (2.8) as

$$\mathbf{y}(m) = \tilde{\mathbf{S}}\mathbf{b}(m) + \bar{\mathbf{v}}(m), \tag{2.9}$$

where $\tilde{\boldsymbol{S}} = [\tilde{\boldsymbol{s}}_1, \tilde{\boldsymbol{s}}_2, ..., \tilde{\boldsymbol{s}}_K]$ is called effective spreading matrix. User's data at time m is called $\boldsymbol{b}$(m) = [b_1(m), …, b_K(m)].

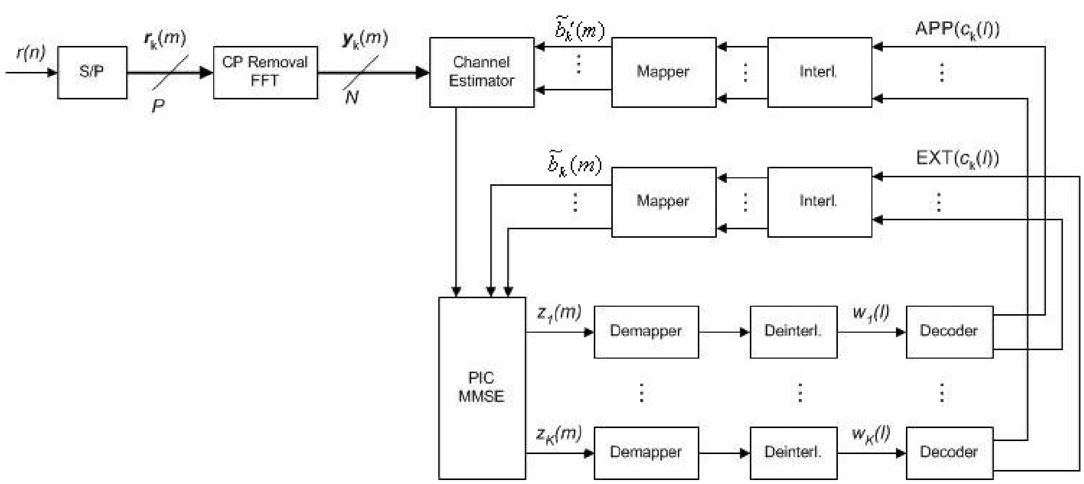


Figure 2.4. MC-CDMA receiver blockset

Next, $\boldsymbol{y}$(m) passes through Parallel Interference Cancellation(PIC), channel estimation and iterative detection as shown in figure 2.4.

Each subsequent sub-section will describe important blocks shown in figure 2.4, namely, the channel estimator, PIC/MMSE, and the channel decoder using BCJR algorithm [2].

2.4.1. Channel Estimator

Having a good channel estimator is crucial because we want the effective spreading matrix ($\tilde{\boldsymbol{S}}$) to be as close as possible to its original value which of course we do not know. Estimation of K user channels ($\hat{\boldsymbol{g}}_k$) in the the frequency domain in the least squares sense can be obtained jointly for all users but individually for every subcarrier q. First, we form matrix $\boldsymbol{P}_q$ to represent the user's pilot codes on each subcarrier q:

$$\mathbf{P}_q = \begin{pmatrix} p_{1,q}(0) & \cdots & p_{K,q}(0) \\ \cdots & \ddots & \vdots \\ p_{1,q}(J-1) & \cdots & p_{K,q}(J-1) \end{pmatrix}. \tag{2.10}$$

$p_{k,q}$(m) represents the pilot spreading coefficient for user k at time index m and subcarrier q. The estimated channel coefficient ($\hat{g}_k(q)$) can be obtain using $\boldsymbol{P}_q$ and y_q(m) for each subcarrier as follows:

$$\begin{pmatrix} \hat{g}_1(q) \\ \hat{g}_2(q) \\ \vdots \\ \hat{g}_K(q) \end{pmatrix} = \mathbf{P}_q^{\#} \begin{pmatrix} y_q(0) \\ y_q(1) \\ \vdots \\ y_q(J-1) \end{pmatrix}, \tag{2.11}$$

where $(.)^{\#}$ denotes pseudo inverse.

Subsequently, after the obtaining channel coefficient ($\hat{\boldsymbol{g}}_k$) for each subcarrier q and user k, we can find the channel impulse response in the time domain ($\boldsymbol{h}_k$) which is comprised of L taps by using the following equation:

$$\hat{\mathbf{h}}_k = \frac{1}{\sqrt{N}} \mathbf{F}^H_{N\times L} \hat{\mathbf{g}}_k. \tag{2.12}$$

This operation also reduces the noise in the channel estimation. $\boldsymbol{F}_{N\times L}$ is a partial Fourier matrix with only the first L columns taken from $\boldsymbol{F}$. Then these estimates are supplied to PIC and MMSE detector.

For the first iteration only eq. (2.12) will be scaled in order to compensate for the missing energy for n = L+1… N-1. Hence,

$$\hat{\mathbf{h}}_k = \frac{1}{\sqrt{N}} \mathbf{F}^H_{N\times L} \hat{\mathbf{g}}_k \sqrt{\frac{N}{L}}. \tag{2.13}$$

For the second iteration, channel estimation will involve a-posteriori probabilities (APP) information from the channel decoder, after being interleaved and mapped. On each iteration i, APP information is mapped as follows:

$$\tilde{b}'_k(m) = \varphi(APP(c_k^{(i)})) = 2 \cdot APP(c_k^{(i)}) - 1. \tag{2.14}$$

The channel estimation becomes:

$$\begin{pmatrix} \hat{g}_1(q) \\ \hat{g}_2(q) \\ \vdots \\ \hat{g}_K(q) \end{pmatrix} = \begin{pmatrix} \mathbf{P}_q \\ \mathbf{Q}_q \end{pmatrix}^{\#} \begin{pmatrix} y_q(0) \\ y_q(1) \\ \vdots \\ y_q(M-1) \end{pmatrix}. \tag{2.15}$$

Matrix $\boldsymbol{Q}_q$ is defined as:

$$\mathbf{Q}_q = \begin{pmatrix} s_1(q)\tilde{b}'_1(J) & \cdots & s_K(q)\tilde{b}'_K(J) \\ \cdots & \ddots & \vdots \\ s_1(q)\tilde{b}'_1(M-1) & \cdots & s_K(q)\tilde{b}'_K(M-1) \end{pmatrix}. \tag{2.16}$$

On the third iteration and so on, each column of $\begin{pmatrix} \boldsymbol{P}_q \\ \boldsymbol{Q}_q \end{pmatrix}$ is normalized by $\sqrt{2M/N}$.

The purpose of normalization is to speed up convergence. During the first iteration, the soft values in matrix $(P_q,Q_q)^T$ can be very small due to strong interference. Using a solution to normalize the matrix, we can achieve a good BER performance for a relatively small number of iterations.

2.4.2. Data Detection

Data detection is performed by calculating the estimate b_k using previously determined values and extrinsic information. But first, before the received sequence is estimated, we need to cancel out the interference of other users from the user of interest, which is called multiple-access interference (MAI). By soft canceling for user *k*, we can eliminate most MAI, i.e.,

$$\tilde{\mathbf{y}}_k^{(i)}(m) = \mathbf{y}(m) + \tilde{\mathbf{s}}_k^{(i)}\tilde{b}_k^{(i)}(m) - \tilde{\mathbf{S}}^{(i)}\tilde{\mathbf{b}}(m). \qquad (2.17)$$

The extrinsic information, was mapped for iteration *i* in order to form

$$\tilde{b}_k^{(i)} = \varphi(EXT(c_k^{(i)})) = 2 \cdot EXT(c_k^{(i)}) - 1. \qquad (2.18)$$

Then using MMSE-filter, we clean $\tilde{\boldsymbol{y}}_k^{(i)}(m)$ from noise and MAI

$$z_k^{(i)}(m) = (\boldsymbol{f}_k^{(i)})^H\, \tilde{\mathbf{y}}_k^{(i)}(m), \qquad (2.19)$$

where z_k(m) gives the estimation of b_k.

For the MMSE-filter itself, we use an unbiased filter estimate (iteration index i, was omitted for simplicity) which has the form

$$\mathbf{f}_k^H = \frac{\tilde{\mathbf{s}}_k^H(\sigma_v^2\mathbf{I} + \tilde{\mathbf{S}}\mathbf{V}\tilde{\mathbf{S}}^H)^{-1}}{\tilde{\mathbf{s}}_k^H(\sigma_v^2\mathbf{I} + \tilde{\mathbf{S}}\mathbf{V}\tilde{\mathbf{S}}^H)^{-1}\tilde{\mathbf{s}}_k}, \qquad (2.20)$$

where $\boldsymbol{V}$ is the error covariance matrix which was constant during iteration *i* and has the diagonal elements of

$$V_{j,j} = E\left\{1 - \left|\tilde{b}_k^{(i)}(m)\right|^2\right\}. \qquad (2.21)$$

The other elements of $\boldsymbol{V}$ are zero. The variance is calculated from all symbols in the block belonging to user *k*, and we call the filter unconditional. By unconditional, we mean that the filter is applied to all symbol instances in a single block for a given user. This has an immediate impact on the assumption that $\boldsymbol{V}$ is ergodic. Hence, we obtain $\boldsymbol{V}$ by averaging as in eq. (2.21) rather than calculate $V(m) = E\{(\boldsymbol{b}(m) - \tilde{\boldsymbol{b}}(m) \cdot (\boldsymbol{b}(m) - \tilde{\boldsymbol{b}}(m)^H\}$, since $\boldsymbol{V}$ does not depend on *m* any longer.

2.4.3. Data Decoder with BCJR algorithm

Data decoding is performed using the BCJR algorithm, named after its creators (Bahl, Cocke, Jelinek and Raviv) [2]. The decoder feeds back soft values of coded bits $c_k(l)$ to improve channel estimation and data detection results. These soft values are provided by a-posteriori probability (APP) and extrinsic information (EXT) from the coded symbols which are interleaved and mapped to BPSK. Here the BCJR decoder acts as a soft-input soft-output (SISO) decoder for binary convolutional codes.

Coded symbol's APP (APP($c_k(l)$)), which is regarded as the probability of $c_k = +1$ when $w_k(l)$ is observed is given by $\mathrm{APP}(c_k(l)) = \Pr[c_k(l) = +1 / w_k(l)]$ (figure 2.4). To get the EXT for coded bit c_k(l), we can link the APP values to the EXT values which are governed by

$$APP(c_k(l)) \propto EXT(c_k(l))\, p(w_k(l) \mid c_k(l) = +1) \qquad (2.22)$$

$$p(w_k(l)/c_k(l) = +1) \propto \exp(-\frac{|w_k(l) - 1|^2}{2\hat{\sigma}^2_{v,k}}),$$

where estimated noise variance is $\hat{\sigma}^2_{v,k} = \frac{1}{2M}\sum_{l=1}^{2M} |w_k(l)|^2 - 1$.

The last part of the right-hand side of equation (2.22) is called the channel transition function, which is formulated as a Gaussian pdf.

To begin our decoding activity, we should see how the APP of the coded symbol is generated. Unlike the Viterbi decoder, which is based on minimization of the probability of word error, the BCJR algorithm was based on minimization of the probability of symbol (bit) error.

We assume that the source is a discrete-time finite-state Markov process, e.g, a binary convolutional encoder. The M distinct states of the source are indexed by integer m, $m = 0,1, \ldots,$ M-1. The state of the Markov process at time t is given by S_t and its output by X_t. A sequence of the source state from time t to t' is denoted by $S_t^{t'} = S_t, S_{t+1}, \ldots, S_{t'}$. Its output sequence is denoted by $X_t^{t'} = X_t, X_{t+1}, \ldots, X_{t'}$.

The state transitions of the Markov source are given by the transition probabilities

$$p_t(m/m') = \Pr\{S_t = m / S_{t-1} = m'\}. \quad (2.23)$$

Similarly, the output is governed by

$$q_t(m/m') = \Pr\{X_t = m / X_{t-1} = m'\}. \quad (2.24)$$

The markov source starts in initial state $S_0 = 0$ and ends in state $S_\tau = 0$. Hence, we might want to give our convolutional encoder initial states 0 and pad zeros to our original source in order to get ending state = 0. Zero padding can be done by adding a sequence of 2^M zeros to the signal sequence.

If the output sequence $\boldsymbol{X}$ is transmitted to some Discrete Memoryless Channel (DMC), we call the output of the channel $\boldsymbol{Y}$. Given $\mathbf{X}_1^\tau$ as the input sequence, it produces $\boldsymbol{Y}_1^\tau = \{Y_1, Y_2, .., Y_\tau\}$. Figure 2.6 shows the trellis diagram for the convolutional encoder with generator $[7\ 5]_8$. The convolutional encoder itself is shown in figure 2.5. Each node on the trellis corresponds to APP $\Pr\{S_r = m / Y_1^\tau\}$ and each branch is represented by APP $\Pr\{S_{t-1} = m'; S_t = m / Y_1^\tau\}$. The objectives of the decoder are to examine $\boldsymbol{Y}_1^\tau$ and to compute these APPs.

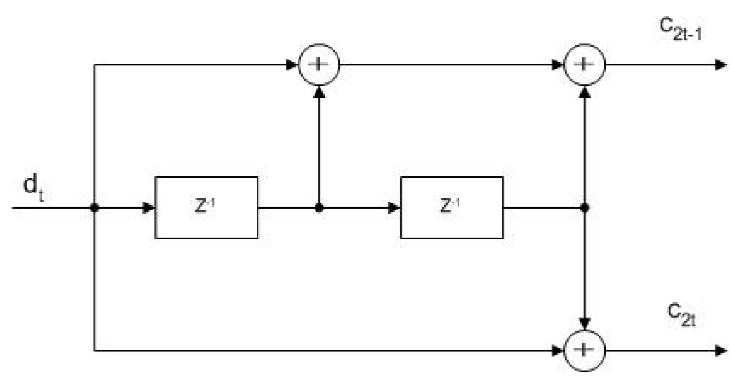


Figure 2.5. Convolutional Encoder, g = [7 5]$_8$, ½ rate

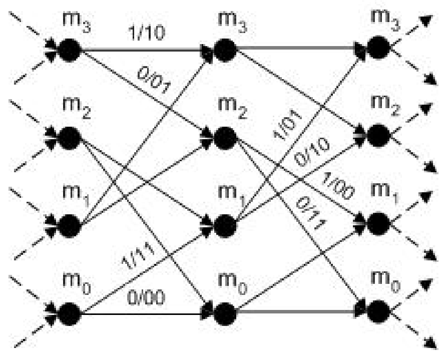


Figure 2.6. Trellis diagram for ½ rate convolutional encoder, g = [7 5]$_8$. The states correspond to the content of delay elements as m_0=(0,0) , m_1=(0,1) , m_2=(1,0) and m_3=(1,1). Each input-output pair shows (d_t/c_1c_2).

Reference [2] provides the complete derivation of the algorithms, here, we only provide a summary of the algorithms which are implemented in the simulations.

Let us define the probability functions:

$$\lambda_t(m) = \Pr\{S_t = m; Y_t^\tau\} \tag{2.25}$$

$$\sigma_t(m', m) = \Pr\{S_{t-1} = m'; S_t = m; Y_t^\tau\} \tag{2.26}$$

$$\alpha_t(m) = \Pr\{S_t = m; Y_t^\tau\} \tag{2.27}$$

$$\beta_t(m) = \Pr\{Y_{t+1}^\tau / S_t = m\} \tag{2.28}$$

$$\gamma_t(m', m) = \Pr\{S_t = m; Y_t / S_{t-1} = m'\}. \tag{2.29}$$

Using the above equations, we can redefine $\lambda_t(m)$ and $\sigma_t(m', m)$ as

$$\lambda_t(m) = \alpha_t(m) \cdot \beta_t(m) \tag{2.30}$$

$$\sigma_t(m', m) = \alpha_{t-1}(m') \cdot \gamma_t(m', m) \cdot \beta_t(m). \tag{2.31}$$

We can calculate $\alpha_t(m)$ for $t = 1,2,\ldots,\tau$

$$\alpha_t(m) = \sum_{m'=0}^{2^{Ld}-1} \alpha_{t-1}(m') \cdot \gamma_t(m', m), \tag{2.32}$$

where *Ld* represents number of memory register (assume 1 bit length).

Initial conditions are needed to set values at t=0. Let $\alpha_0(0) = 1$ and $\alpha_0(m) = 0$ for $m \neq 0$.

The same thing happens for calculating $\beta_t(m)$, t = 1,2,…,τ-1

$$\beta_t(m) = \sum_{m'=0}^{2^{Ld}-1} \beta_{t+1}(m') \cdot \gamma_{t+1}(m,m').$$

The initial conditions used are $\beta_\tau(0) = 1$ and $\beta_\tau(m) = 0$ for $m \neq 0$.

To calculate $\gamma_t(m',m)$, we use:

$$\gamma_t(m',m) = \Pr\{d_t = d_{m'm}\} \cdot \Pr\{c_{2t-1} = c_{1,m',m} / Y_1^\tau\} \cdot \Pr\{c_{2t} = c_{2,m',m} / Y_1^\tau\}, \qquad (2.33)$$

for $(m',m) \in \beta$ and $\gamma_t(m',m) = 0$ for $(m',m) \notin \beta$.

We assumed rate ½ convolutional encoder whose outputs are c_{odd} (the first coded bit in output sequence) and c_{even} (the second one).

The parameter, β denotes valid branches of index pairs (m',m), meaning that the branch exists. Using figure 2.6 as example, our β becomes

$$\beta = \{(0,0);(0,1);(1,2);(1,3);(2,0);(2,1);(3,3);(3,2)\}.$$

Since we are using a uniformly distributed data source, d_k has alphabet {1,0}. Let us assume that

$$\Pr\{d_t = 0\} = \Pr\{d_t = 1\} = 1/2,$$

and using the Log-Likelihood Ratio (LLR) of $w_k(l)$ (from figure 2.4) we can calculate the probability

$$\Pr\{c_{2t-1} = c_{1,m',m} / Y_1^\tau\} = \frac{\exp(-c_{1,m',m} \cdot w_k(t))}{1+\exp(-w_k(t))}. \qquad (2.34)$$

We can see $w_k(t)$ as $w_k(l) = LLR(c_k(l)/Y_t^{t'}) = \ln \frac{\Pr(c_k(l) = 1/Y_t^{t'})}{\Pr(c_k(l) = 0/Y_t^{t'})}$.

For computation purposes, we create matrix $\boldsymbol{A}_1(x)$ which is defined for $x \in$ {1,0} as follows (according to trellis diagram, e.g. figure 2.6) :

$$\{\mathbf{A}_1(x)\}_{m',m} = \begin{cases} 1 & (\text{m}'\ \text{m}) \text{ is a branch produces } c_{1,m',m} = x \\ 0 & \text{otherwise} \end{cases} \qquad (2.35)$$

The procedure is the same for $\boldsymbol{A}_2(x)$. Using the same procedure we can create matrix $\boldsymbol{A}_2(x)$.

$$\{\mathbf{A}_2(x)\}_{m',m} = \begin{cases} 1 & \text{(m}'\text{ m) is a branch produces } c_{2,m',m} = x \\ 0 & \text{otherwise} \end{cases} \quad (2.36)$$

We also need to generate matrix $\boldsymbol{B}(1)$. This is done in the same manner as $\boldsymbol{A}_1(1)$ to determine the estimated source bit ($\hat{d}_t$).

Now, the estimated source bit ($\hat{d}_t$) can be computed using

$$\Pr\{\hat{d}_t / Y_1^\tau\} = \frac{1}{\lambda_\tau(0)} \sum_{m \in \mathbf{B}(1)} \lambda_t(m). \quad (2.37)$$

The equation above gives the $\text{APP}\{\hat{d}_k(l)\} = \Pr\{\hat{d}_k(l) = +1 / w_k(l)\}$ and $\lambda_t(0)$ is the normalization function. Therefore, the total sum of all states (m) of a given $\lambda_t = 1$. Matrix $\boldsymbol{A}$ helps us to figure out which branch is included in β.

Now, our iterative decoder needs to feedback soft inputs of coded bits, namely, $c_k(l)$. Then to get the APP of the encoder output we use

$$\Pr\{c_{2t-1} = 1; Y_1^\tau\} = \sum_{(m',m) \in \boldsymbol{A}_1(1)} \sigma_t(m', m) \quad (2.38)$$

$$\Pr\{c_{2t} = 1; Y_1^\tau\} = \sum_{(m',m) \in \boldsymbol{A}_2(1)} \sigma_t(m', m). \quad (2.39)$$

We can also normalize the above equations using $\lambda_t(0)$. Then the extrinsic information (EXT) can be computed using eq. (2.22) once the APP is available. When performing the computer simulation, we need to take into account some problems regarding round off error.

2.4.4. Log-MAP Decoder

The MAP decoder using the BCJR algorithm like in section 2.4.3 provides a good estimate in term of symbol by symbol error rate. But, during implementation either in hardware or simulation, the MAP decoder faces most common problems which are expensive to be implemented in floating point arithmetic, thus round off errors occur. Especially in this iterative decoder, the MAP decoder will involve small values which in turn could saturate the system fast (reciprocal of very small values leads to very large numbers which cannot be accommodated by hardware or memory). To work around the problems, we could use a modification of the MAP decoder ,e.g., Max-Log-MAP and Log-MAP algorithms. Should we decide to use the Max-Log-MAP, we use the following approximation:

$$\ln(e^{\delta_1} + e^{\delta_2} + \ldots + e^{\delta_n}) \approx \max_{i \in \{1..n\}} \delta_i. \tag{2.40}$$

However, this Max-Log-MAP approach gives us a sub-optimal solution compared to MAP algorithms. To enhance the performance, we could use the Log-MAP [6,8,9] approach by using following:

$$\ln(e^{\delta_1} + e^{\delta_2}) \approx \max(\delta_1, \delta_2) + \ln(1 + e^{-|\delta_2 - \delta_1|}). \tag{2.41}$$

The second part of the right hand side is called a correction function $f_c(|\delta_2 - \delta_1|)$. The approach can be extended to accommodate more than two elements by doing the calculation iteratively two elements at a time.

For example, in our case, eq. (2.32) becomes

$$\ln(\alpha_t(m)) = \ln(e^{\ln(\alpha_{t-1}(0)\cdot\gamma_t(0,m))} + \ldots e^{\ln(\alpha_{t-1}(2^v)\cdot\gamma_t(2^v,m))}). \tag{2.42}$$

then assume $\Delta = (e^{\ln(\alpha_{t-1}(0)\cdot\gamma_t(0,m))} + e^{\ln(\alpha_{t-1}(1)\cdot\gamma_t(1,m))})$, and we can calculate $\delta = \ln(\Delta)$ using eq. (2.41). But, we can express $\Delta = e^{\delta}$, hence eq. (2.42) becomes

$$\ln(\alpha_t(m)) = \ln(e^{\delta} + e^{\ln(\alpha_{t-1}(2)\cdot\gamma_t(2,m))} + \ldots e^{\ln(\alpha_{t-1}(2^v)\cdot\gamma_t(2^v,m))}). \tag{2.43}$$

2.5. Simulation of MC-CDMA under Multipath

Simulations were carried, K = 16 users. Each user's message code is spread with N = 16 which equals to number of subcarrier. We used a simple convolutional encoder as depicted in figure 2.5. The pilot code length that was inserted in each frame is 8. Frame length is 128. As for the channel impulse response, its delay spread was chosen to be 3 times of the chip symbol period T_c. Cyclic prefix's length (G) is 3, then P = 16+3 = 19. Signal-to-Noise Ratio (SNR) is defined as

$$\frac{E_b}{N_0} = \frac{1 \cdot P \cdot M}{R \cdot \sigma_v^2 \cdot N \cdot (M - J)}. \tag{2.44}$$

The SNR in the above equation will determine the AWGN noise variance. By setting the user's bandwidth to 20KHz, we have a total net of N×W = 32×20KHz = 0.32 MHz bit rate per cell. The result of this simulation is shown in figure 2.7. We can see the performance improvement of the bit error rate (BER) as the number of iterations increases. This BER will reach a limit asymptotically, which means that by increasing the number of iterations we do not get significant improvement after the threshold. This phenomenon can be seen in figure 2.7. BER performance after iteration 7 does not give significant improvement. Hence, it would be futile to increase the number of iteration to 8 or more.

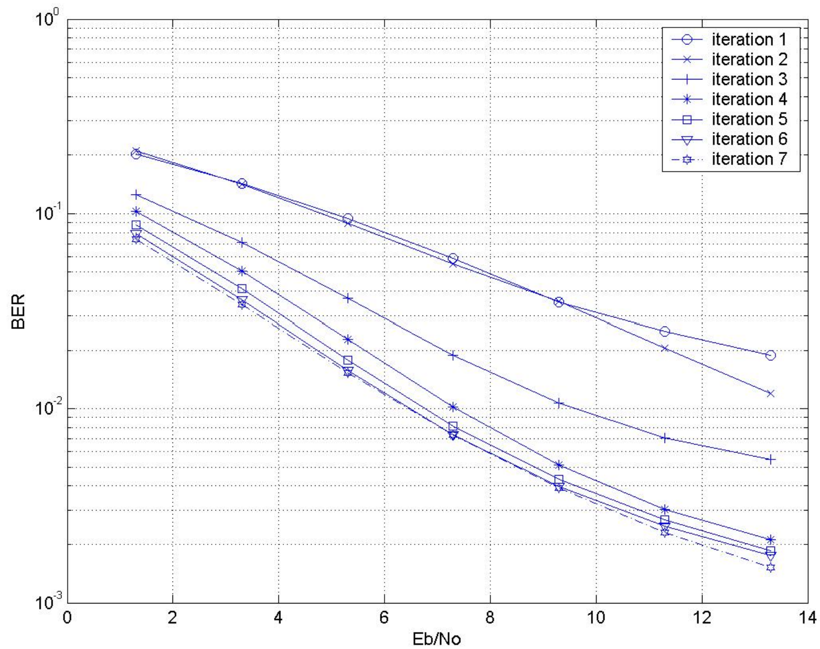


Figure 2.7. BER performance for N=16,K=16,L=3,G=3,M=128 and J=8.

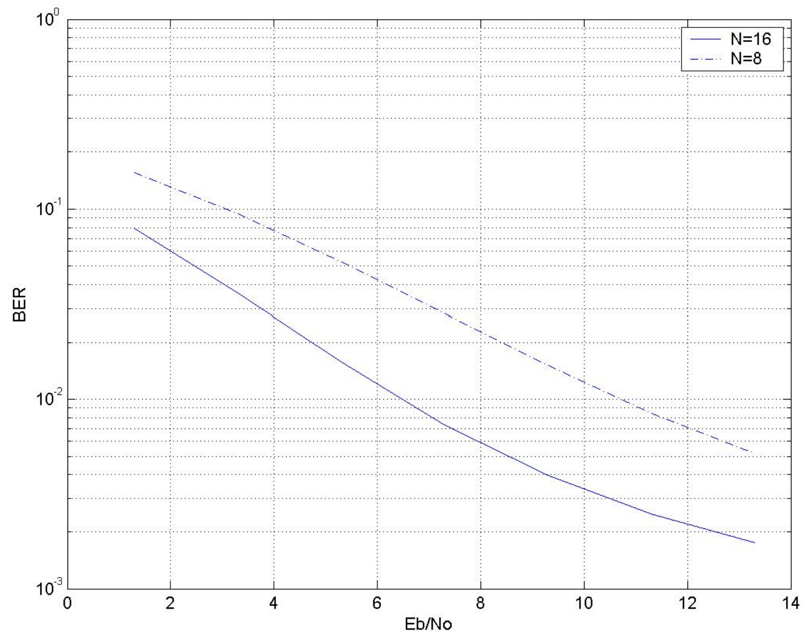


Figure 2.8. The spreading gain effect on fully loaded MC-CDMA systems for L=3,G=3 and J=8.

However, the system would perform better should it have more spreading gain. We might foresee this in equation (2.44). By increasing the number of subcarriers, AWNG noise variance will be lowered too when the other parameters remain the same. Figure 2.8 gives the BER performance of MC-CDMA systems with N = 8 and N = 16 at the 6^{th} iteration.

We could also see the effect of lacking cyclic prefix (*G*) on MC-CDMA system. As we stated before, it is important that each frame should have a long enough cyclic prefix, which is equal or greater than channel impulse response length (*L*). Should the frame still experience some cyclic prefix shortage, we can expect degradation of the system in BER sense. Figure 2.9 shows such situation.

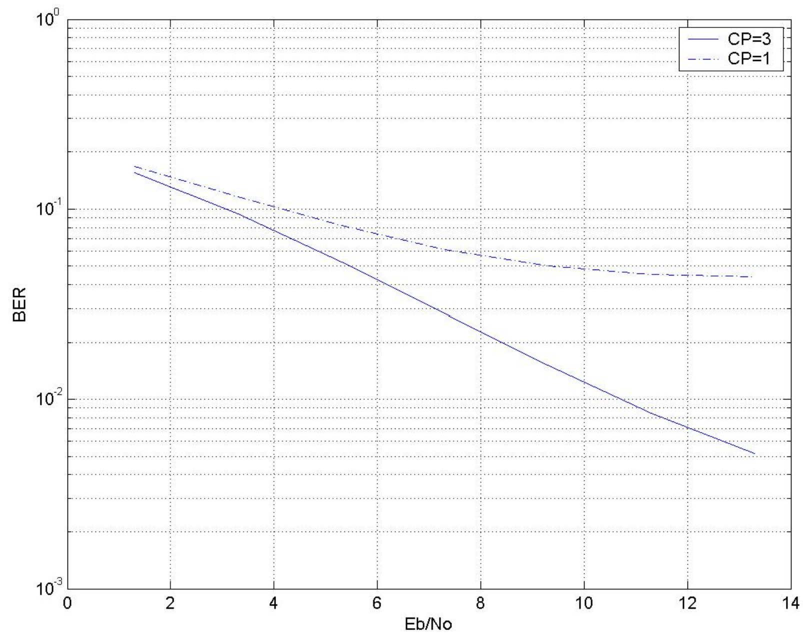


Figure 2.9. MC-CDMA system in case of channel prefix (CP) shortage for N=8,K=8,J=8,L=3 and G=3

Furthermore, we might improve the system by using Turbo codes to encode user's message instead of using conventional convolutional encoder.

In the next chapter, we shall explore the possibility of a system design without any pilot code embedded in the frame. In the absence of a pilot code, the user's bandwidth is used for information only, which in turn enhances the bandwidth efficiency. We still use the uplink scenario as we do in this chapter. The system will also employ the MC-CDMA technique, though with higher bandwidth efficiency. We shall see in chapter 3 that the so-called 'blind equalization' works under certain conditions only.

3. MULTIUSER DETECTION FOR A MC-CDMA SYSTEM USING BLIND EQUALIZATION

3.1. Two Stage CMA as an alternative to pilot based system

In Chapter 2, we described the pilot based MC-CDMA system. The proposed method proves to be robust in a real-world mobile communication medium which experiences multipath propagation. Now, if we remove the pilot signal from a frame, can the user signal be recovered? Recently, several detection methods have been developed based on blind equalization. By blind we mean that the user signal that arrives at the receiver front end will be estimated using the received signal itself and some statistical properties of the input source. Blind methods can be classified into three categories [21], they are:

1. Bussgang algorithms.
2. Polyspectra and cumulant-based algorithms.
3. Probabilistic algorithms.

This thesis will use the first category for its ease of usage compared to the other two. The earliest Bussgang algorithm developed was introduced by Sato [17]. This technique relies on a certain class of cost functions. Later, Godard [27] generalized the cost function of Sato's algorithm. A cost function is a measurement of the amount of ISI introduced by the channel, which does not involve the transmitted symbols. The optimization of the cost function should lead to the minimization of ISI. The name "Bussgang algorithm" itself was created by Bellini et al [18], which is based on the assumptions made about the equalizer and the channel parameters. Sato and Godard algorithms can be classified as special cases of this technique. The Constant Modulus Algorithm (CMA), which is *a* special case of Godard's algorithm, was developed by Treichler [28,29]. In this Two Stage CMA receiver, CMA is employed as the second stage to further clear out ISI. While for the first stage, a pre-filtering is used. The filter in first stage tries to enhance the contribution of the desired-user with respect to the disturbance and eliminate interblock interference (IBI). This method falls into indirect Multiuser Detection (MUD), since the system is not trying to estimate the channel impulse response explicitly. On the other hand, a direct MUD system will estimate the channel impulse response before trying to estimate the input source. There is also a *group-blind* receiver which assumes the knowledge of the spreading codes of all users. However, this receiver requires a costly eigendecomposition operation and oversampling. The two stage receiver, on the other hand, is simpler than this. Hence, it is easier to implement. The Two Stage CMA can work in an asynchronous environment too. In this thesis, we assume synchronous transmission from the users (downlink).

3.2. Transmitter model of blind MC-CDMA system

The basic idea of blind MC-CDMA transmitter system is similar to the one which was presented in chapter 2. Here, we also assume a baseband representation. The transmitter simulates J users with N subcarriers. The user symbol ($d_j(m)$), is produced by user j at time m, which in turn will be convolutionally encoded and modulated using BPSK. The rate ½ convolutional encoder uses the $[7\ 5]_8$ generator matrix, which was also used in previous pilot-based system (fig. 2.5). After channel encoding, the encoded symbols are processed by a spreading block. We can also use a Walsh spreading code to generate our spreading vector ($\boldsymbol{c}_j$) = $\left[c_j^{(0)}, c_j^{(1)}, \ldots, c_j^{(N-1)}\right]$, which satisfies the following constraint:

$$\sum_{n=0}^{N-1} \left|c_j^n\right|^2 = N \qquad \forall \text{ n.}$$

The spreading code itself has the alphabet {-1,1}. As it was shown in chapter 2, this blind system will also use a cyclic prefix to eliminate IBI. Then, with the help of IDFT, we assign each corresponding code ($d_j(m)\, c_j^n$) to its corresponding subchannel (n). Figure 3.1 depicts the blind MC-CDMA transmitter. Now, in order to demonstrate how IBI elimination is achieved, let us mathematically describe how the system works.

At the output of the transmitter,

$$\boldsymbol{u}_j(m) = \boldsymbol{T}_{cp}\boldsymbol{W}_{IDFT}\boldsymbol{c}_j d_j(m), \tag{3.1}$$

where $\mathbf{W}_{IDFT}(m,n) = \frac{1}{N}\exp(j\frac{2\pi}{N}mn),\ \ \mathrm{m,n = 0,1,...N-1}$, is the $N \times N$ IDFT matrix, and $\boldsymbol{T}_{cp} = [\boldsymbol{I}_{cp}^T, \boldsymbol{I}_N]^T$ is the $P{\times}N$ Cyclic Prefix insertion matrix, where $P{=}L{+}N$. $\boldsymbol{I}_{cp}$ denotes an $L{\times}N$ matrix, which is obtained by selecting the last L rows of the $N{\times}N$ identity matrix. Later, we can assume that L is longer than the channel impulse response ,e.g., $L \geq L_j + 1$. L_j denotes the length of the channel impulse response of user j.

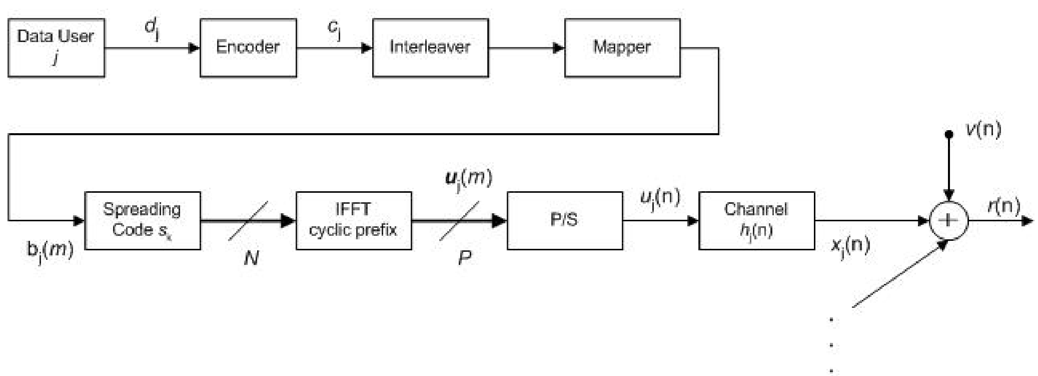


Figure 3.1. MC-CDMA transmitter for blind equalization.

Next, we can use a (Digital-to-Analog) D/A converter after processing the signal by the Parallel-Serial block. The continuous form of $u_j(t)$, which is operated at $1/T_c$, can be expressed as:

$$u_j(t) = \sum_{m=-\infty}^{\infty}\sum_{p=-L}^{N-1} u_j^{(p)}(m)\phi_j(t - pT_c - mT_s - \tau_j), \tag{3.2}$$

where T_c is the chip rate, T_s is the symbol rate, τ_j is the transmission delay, and ϕ_j is the impulse response of the D/A. We can safely assume that the delay between transmitted and receiver signals is zero because of synchronous transmission model.

3.3. Channel Model

For the channel model, we consider two types of channels. The first one is a fictitious discrete-time channel with minimum phase, e.g., its zeros are inside unit circle. Then we use a multipath channel for the second type of channel model, as described in section 2.3.

Here, the discrete-time fictitious channel is described as follows:

1. h_1 = [1 -0.7 0.31 0.183].
2. h_2 = [1 -0.4 0.79 0.272].
3. h_3 = [-1.3 0.1 0.7 1.3].
4. h_4 = [0.8 1.1 0.7 0.9].

The discrete-time fictitious channel will have a length of 4 (L_j = 4). Then, we randomly choose a channel from the above selections for a certain user. The received signal, described in continuous-form, is given by

$$\tilde{r}(t) = \sum_{j=1}^{J} \tilde{r}_j(t) + v(t), \tag{3.3}$$

where $\tilde{r}_j(t) = u_j(t) \otimes h_j(t)$ and $v(t)$ is AWGN. Symbol ($\otimes$) denotes convolution.

3.4. Blind MC-CDMA receiver block set

In order to detect the n^{th} symbol, we sample the received signal at time $t = nT_s + lT_c$. From eq. (3.3), we get:

$$\tilde{r}^{(l)}(n) = \tilde{r}^{(l)}(nT_s + lT_c) = \sum_{j=1}^{J} \tilde{r}_j^{(l)}(n) + v^l(n), \tag{3.4}$$

Each of the samples will be grouped to form a vector $\tilde{\boldsymbol{r}}(n) = [\tilde{r}^{(-L)}(n),...,\tilde{r}^{(N-1)}(n)]$ of P elements. Since we assume zero transmission delay, we can write each sampled received signal as follows:

$$\tilde{r}_j^{(l)}(n) = \sum_{m=-\infty}^{\infty} \sum_{p=-L}^{N-1} u_j^{(p)}(m) h_j[(n-m)P + l - p] \tag{3.5}$$

We refer to (m) as a time index with respect to the n^{th} symbol. The time index when m=n will be referred to as zero.

If we express $h_j^{(k)}(m) \triangleq h_j(mP+k)$, we rewrite eq. (3.5) as:

$$\tilde{r}_j^{(l)}(n) = \sum_{m=-\infty}^{\infty} \sum_{p=-L}^{N-1} h_j^{(p)}(n-m) u_j^{(l-p)}(m). \tag{3.6}$$

Using the matrix form of the received signal $\tilde{\boldsymbol{r}}(n)$, we can rewrite eq. (3.6) in matrix form as follows:

$$\tilde{\boldsymbol{r}}_j(n) = \sum_{m=-\infty}^{\infty} \boldsymbol{H}_j(m) \boldsymbol{u}_j(n-m), \tag{3.7}$$

where $\boldsymbol{H}_j(m)$ is a $P \times P$ Toeplitz matrix whose elements are defined as $\boldsymbol{H}_j(m)_{ab} = h_j^{(a-b)}(m), a,b = 0,1,...,P-1$. As far as the channel impulse response is concerned, we define it as finite impulse response filter, which implies that $h_j(n)$ = 0, for $n \neq [0,1,\ldots,L_j]$. We also assume that the length of cyclic prefix is longer than L_j. Having this in mind, we can readily see that $\boldsymbol{H}_j(m) = \boldsymbol{0}_{P\times P}$ for m ≠ 0,1. Now, without the loss of generality, we select the user of interest as the first user, that is, j = 1. Thus, we can rewrite eq. (3.7) as follows:

$$\tilde{\boldsymbol{r}}_1(n) = \sum_{m=0}^{1} \boldsymbol{H}_1(m) \boldsymbol{u}_1(n-m) + \sum_{m=2}^{J} \tilde{\boldsymbol{r}}_j(n) + \tilde{\boldsymbol{w}}(n). \tag{3.8}$$

The second term on the right hand side of eq. (3.8) is called multiple access interference (MAI). Here, we can see the interference due to other users on the desired user (j = 1). The first term, on the other hand, is IBI, which is produced by the channel impulse response. As long as the length of the channel impulse response is shorter than the cyclic prefix, only the next block will leverage the current block.

Now, the cyclic prefix is removed by multiplying eq. (3.8) by the matrix $\boldsymbol{R}_{cp}$ which has $N \times P$ elements. $\boldsymbol{R}_{cp}$ is denoted as $[\boldsymbol{0}_{N\times P}, \boldsymbol{I}_N]$. Then $\boldsymbol{R}_{cp}\boldsymbol{H}_1(1) = 0$, provided than the length of the CP is greater than the length of L_1. Hence, eq .(3.8) becomes

$$\boldsymbol{r}(n) = \boldsymbol{R}_{cp}\boldsymbol{H}_1(0)\boldsymbol{u}(n) + \boldsymbol{R}_{cp}\sum_{j=2}^{J}\tilde{\boldsymbol{r}}_j(n) + \boldsymbol{R}_{cp}\tilde{\boldsymbol{w}}(n)$$

$$\boldsymbol{r}(n) = \tilde{\boldsymbol{H}}_1\tilde{\boldsymbol{c}}_1 b_1(n) + \boldsymbol{r}_{MAI}(n) + \boldsymbol{w}(n), \tag{3.9}$$

where $\tilde{\boldsymbol{H}}_1 = \boldsymbol{R}_{cp}\boldsymbol{H}_1(0)\boldsymbol{T}_{CP}$ is the $N \times N$ circulant matrix and $\tilde{\boldsymbol{c}}_1 = \boldsymbol{W}_{IDFT}\boldsymbol{c}_1$ is the spreading code in the time domain. The term $\tilde{\boldsymbol{H}}_1\tilde{\boldsymbol{c}}_1$ is referred to as the desired-user composite signature. Using eq. (3.9), we obtain a way to suppress IBI. The next section will describe the *pre-filtering* operation whose purpose is to eliminate MAI.

3.4.1. Pre-filtering for MAI removal

Figure 3.2 depicts the two-stage MC-CDMA receiver. The first stage is pre-filtering, which is used for reducing MAI based on the knowledge of the spreading code of the desired user, even though the knowledge of the desired user channel behavior and timing are unknown.

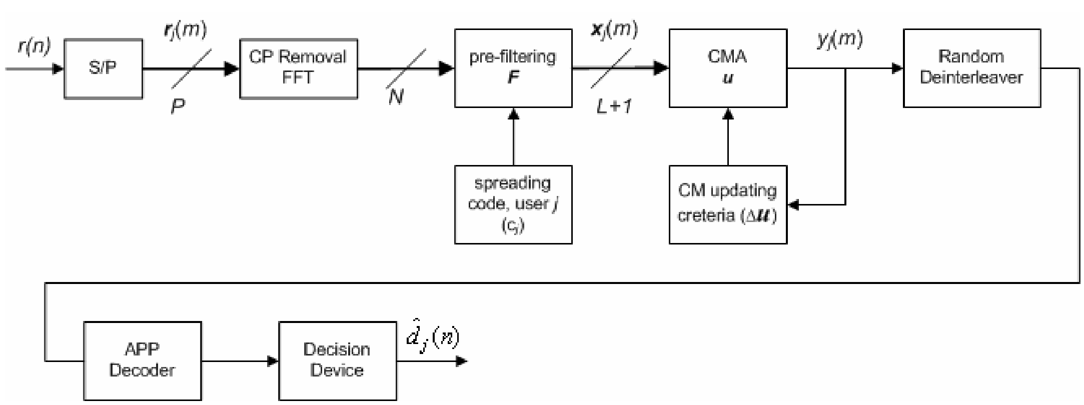


Figure 3.2. The two-stage MC-CDMA receiver.

Using the scheme shown in the above figure, we can get the soft estimate of the desired symbol ($\hat{b}(n)$) through filtering of the received signal. This is given by

$$y(n) = \boldsymbol{f}^H \boldsymbol{r}(n)$$

The filter (***f***) itself is decomposed into two parts $\boldsymbol{f} = \boldsymbol{F}\boldsymbol{u}$, where ***F*** is the pre-filtering part and ***u*** is the CMA filter. The complete derivation can be found in [19]. Here, we use this result in the simulation. Using the Sylvester's structure in the matrix $\boldsymbol{R}_{cp}\boldsymbol{H}_1(0)$ as in [19], we can linearly parameterize the desired-user composite signature in eq. (3.9) as follows:

$$\tilde{\boldsymbol{H}}_1\tilde{\boldsymbol{c}}_1 = \tilde{\boldsymbol{C}}_1\tilde{\boldsymbol{h}}_1, \tag{3.10}$$

where $\tilde{\boldsymbol{C}}_1$ is a $N\times(L+1)$ matrix, whose columns are the permutations of the first column, defined as

$$\tilde{\boldsymbol{C}}_1 = \begin{bmatrix} \tilde{c}_1^{(0)} & \tilde{c}_1^{(N-1)} & \cdots & \tilde{c}_1^{(N-L)} \\ \tilde{c}_1^{(1)} & \tilde{c}_1^{(0)} & \cdots & \tilde{c}_1^{(N-L+1)} \\ \vdots & \vdots & \cdots & \vdots \\ \tilde{c}_1^{(N-1)} & \tilde{c}_1^{(N-2)} & \cdots & \tilde{c}_1^{(N-L-1)} \end{bmatrix} \tag{3.11}$$

Matrix $\tilde{\boldsymbol{h}}_1$ is a $(L+1)$ vector, which represents the channel impulse response. Using eq. (3.10), we can rewrite eq. (3.9) as:

$$\boldsymbol{r}(n) = \tilde{\boldsymbol{C}}_1\tilde{\boldsymbol{h}}_1 b_1(n) + \boldsymbol{r}_{MAI}(n) + \boldsymbol{w}(n), \tag{3.12}$$

From figure 3.2, we have $\boldsymbol{x}(n)$ as the output of the first filter, which can be expressed as:

$$\boldsymbol{x}(n) = \boldsymbol{F}^H \boldsymbol{r}(n) \tag{3.13}$$

By substituting eq. (3.12) into eq. (3.13), we get

$$\boldsymbol{x}(n) = \boldsymbol{F}^H\tilde{\boldsymbol{C}}_1\tilde{\boldsymbol{h}}_1 b_1(n) + \boldsymbol{F}^H\boldsymbol{r}_{MAI}(n) + \boldsymbol{F}^H\boldsymbol{w}(n), \tag{3.14}$$

Borrowing from [19], which states that the first filtering matrix will satisfy $\boldsymbol{F}^H\tilde{\boldsymbol{C}}_1 = \boldsymbol{I}_{L+1}$, the desired symbol ($b_1(n)$) convolved with channel impulse response ($\tilde{\boldsymbol{h}}_1$) will be passed to the CMA stage. The filter (***F***) itself is given by

$$\boldsymbol{F}_{opt} = \boldsymbol{R}_{rr}^{-1}\tilde{\boldsymbol{C}}_1(\tilde{\boldsymbol{C}}_1^H \boldsymbol{R}_{rr}^{-1}\tilde{\boldsymbol{C}}_1)^{-1}, \tag{3.15}$$

where, $\boldsymbol{R}_{rr} = E[\boldsymbol{r}(n)\boldsymbol{r}^H(n)] = \frac{1}{K}\sum_{k=0}^{K-1}\boldsymbol{r}(n+k)\boldsymbol{r}^H(n+k)$. K denotes the chosen sample size. Now that the received signal $\boldsymbol{r}(n)$ has been filtered with $\boldsymbol{F}$, we boost the desired user vector $\boldsymbol{x}(n)$ over the disturbance.

3.4.2. CMA serves as blind equalization block

The second stage involves a Constant Modulus Algorithm block set, which serves the purpose of blind equalization [14,15,20]. CMA is meant to minimize the cost function described by

$$J_{CM}(\boldsymbol{u}) = E[(\gamma - |\boldsymbol{u}^H \boldsymbol{x}(n)|^2)^2], \tag{3.16}$$

where $\gamma = 1$ for BPSK modulation.

Using the filter $\boldsymbol{u}$(n), we will obtain the estimate of the symbol ($b_1(n)$), which is denoted by

$$\hat{b}_1(n) = \boldsymbol{u}^H \boldsymbol{x}(n), \tag{3.17}$$

where $\boldsymbol{u}^H$ has $1\times(L+1)$ dimension and $\boldsymbol{x}(n)$ has dimension $(L+1)\times 1$. The filter $\boldsymbol{u}(n)$ will be updated as follows:

$$\boldsymbol{u}(n+1) = \boldsymbol{u}(n) + \mu y^*(n)(\gamma - |y(n)|^2)\boldsymbol{x}(n). \tag{3.18}$$

where μ is the learning step-size.

3.5. Simulation of blind MC-CDMA under Multipath

This section presents the simulation results of blind MC-CDMA under the proposed channel model, which was described in section 3.4. For the first simulation, we will consider fictitious discrete-time channel model . It is assumed that J = 16 users are present in the system. The source symbol has equiprobable alphabet {0,1}. Without using a convolutional encoder, we spread the source symbol using pseudo-code of length N = 16, which equals the number of subcarrier. Signal-to-Noise Ratio (SNR) is defined as

$$\frac{E_b}{N_0} = \frac{1 \cdot P}{R \cdot \sigma_v^2 \cdot N}. \tag{3.19}$$

The SNR in the above equation will determine the AWGN noise variance. The user's bandwidth is again set to 20 KHz. Cyclic prefix's length (*G*) is L+2=6, thus P = 16+6 = 22, since the length of channel impulse response (*L*) is 4. The result of this simulation is shown in figure 3.3. The channel impulse response is kept the same for the whole simulation. Simulation runs for 10 million bits for each SNR value.

Next, we use the same parameters as in the previous simulations; however, we vary the sample size for calculating the correlation matrix (*K*), while we fix E_b/N_0 = 11.3 dB. The result is shown in figure 3.4.

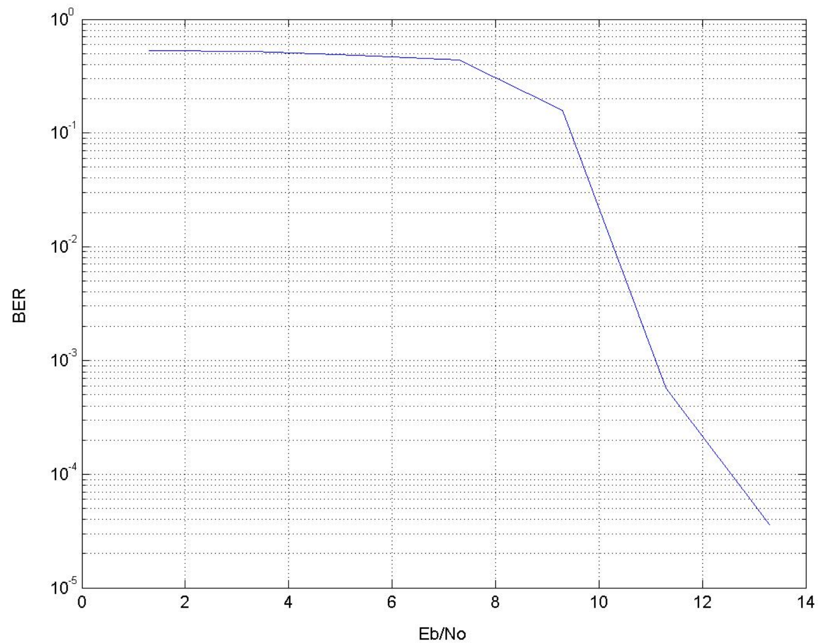


Figure 3.3. BER performance for J=16,L=4,G=6 and K=500.

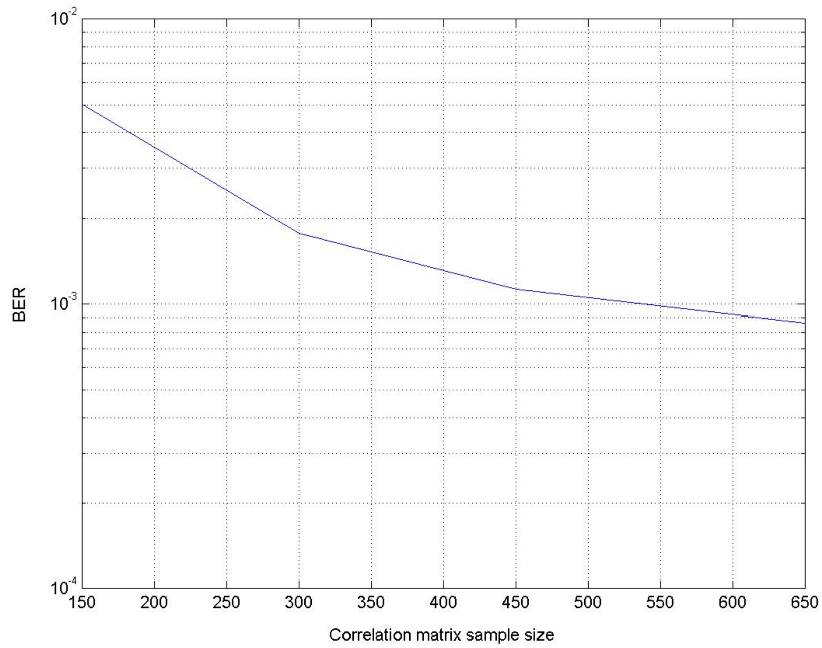


Figure 3.4. The effect of various sample size for the calculation of correlation matrix with parameters set as L=4,G=6,J=16 and E_b/N_0=11.3 dB.

For the second environment, we will use a channel impulse response with 4 paths, where first path is assumed to be deterministic, $L_j^{(1)} = 1$. The remainder part of the channel impulse response will be modeled by the random complex Gaussian variable. We use the MATLAB command as follows:

$\boldsymbol{h}_j$ = [1 randn(1,3)+j*randn(1,3)];

In order to make sure, that channel impulse response will be generated randomly, we use following command on the beginning of simulation series.

randn('state',sum(100*clock));

The result, which is obtained by averaging 100 simulations (each simulates 500000 bits), is shown in figure 3.5. *M* represents block size. We use Viterbi decoder to decode the convolutional code.

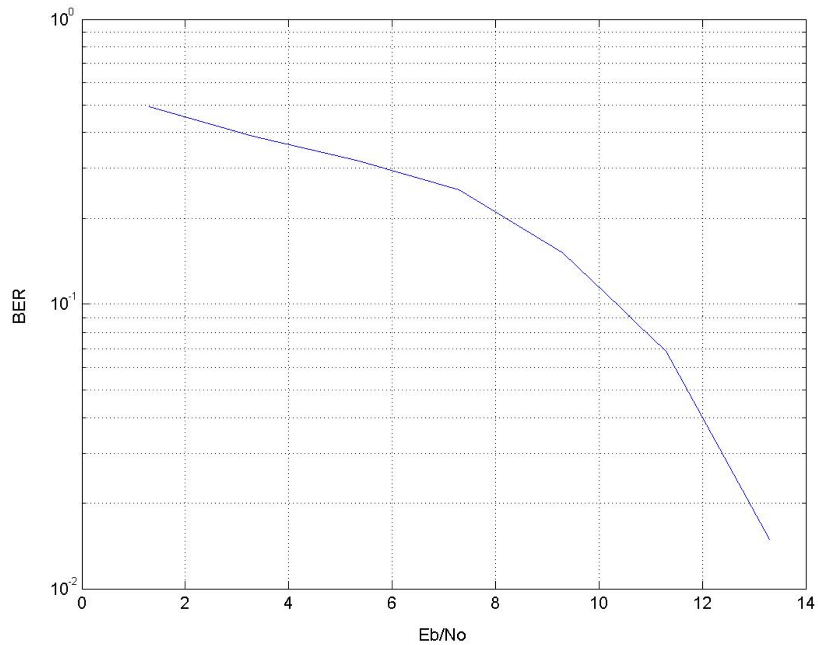


Figure 3.5. Blind MC-CDMA system under a random complex Gaussian variable, using M=8,K=400,J=6,L=4, G=6 and N=16

In this second simulation we perform convolutional encoding on the input source and randomly interleave them. The frame size is 8, with 6 users. Although the BER performance of the blind MC-CDMA system is not as good as that obtained when the pilot was used, the proposed system shows some promise to deliver the user message in an alternative way. It is obvious from the simulation results that the channel impulse response, which can be viewed as filter for the transmitted signal, leverages the BER performance. Should the channel behave nicely, we can expect better results.

Next, we use an A-Posteriori Probability Decoder to decode the soft value which is provided by the CMA blockset. The parameters which were used in the second simulation are used again in the third simulation, except that the Viterbi decoder is replaced by an APP decoder. The result is shown in figure 3.6. We can also see the original version which is developed in [19] for comparison purpose. In the paper, author used a sample-based model and using direct decision device which is unable to use the advantage of having soft values provided by CMA equalization. By inserting an interleaver, the modified system eliminates burst errors which are introduced by the channel impulse response.

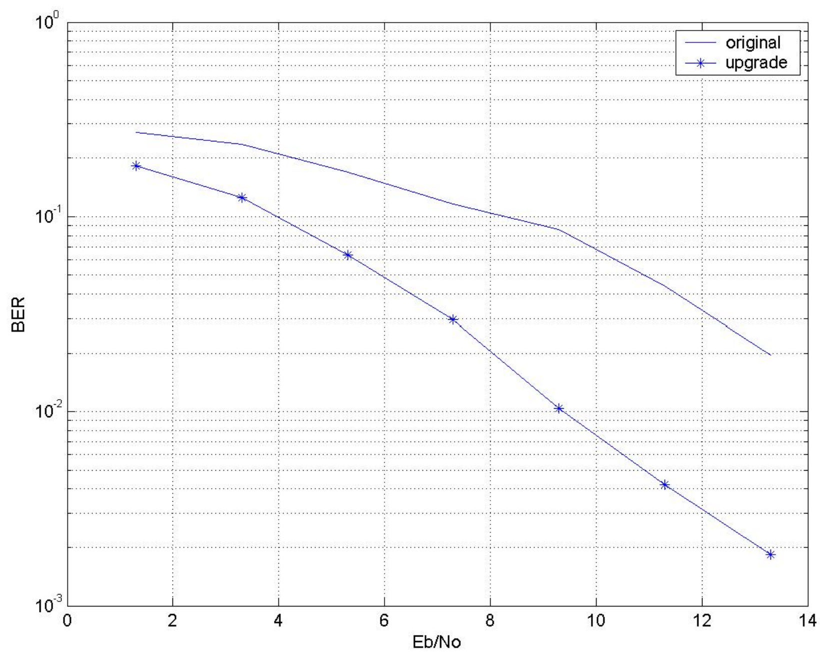


Figure 3.6. APP decoder decodes the soft values provided by CMA blockset.

The one thing that should be pointed out is that blind equalization or blind deconvolution heavily depends on how we model the mobile channel as a filter and how the CMA filter is initialized. This channel model is usually obtained from statistical measurements of the actual communication medium. Then, we formulate it as a filter. Reference [20,24] contains a good example of the wired telephone channel model. From that, we can initialize the CMA filter coefficient based on that channel model. This initialization part should make the energy of the CMA filter be positive. Should the initialization value sway the CMA filter to be negative then the system will have difficulties.

4. CONCLUSION

Let us summarize the contribution made by this research work. We start by stating that the objective of this thesis was to study the channel estimation using either pilot-based techniques or blind equalization in MC-CDMA. Before these methods were introduced, however, we described the concept of MC-CDMA. MC-CDMA alleviates the problem of frequency selectivity of the channel, which attenuates the signal spectrum differently throughout the signal bandwidth. By using a sufficiently large number of sub-carriers (narrower channels), the spectrum experiences flat instead of frequency selective fading. This idea is implemented by using the IFFT instead of a bank of modulators in order to minimize the MC-CDMA complexity in hardware. To avoid Interblock Interference (IBI), we used a cyclic prefix to pad the front of the frame. By assuring that the length of the channel impulse response is smaller than that of cyclic prefix, IBI is suppressed completely.

Next, we modeled the transmission environment that was assumed in this thesis. We used a multipath fading channel model which is assumed to be constant over one frame period. This channel model is realized by a tapped delay line model or transversal FIR filter. Each of the coefficients that are used as weights is generated by Jakes' model. The magnitude of each tap coefficient has a Rayleigh probability distribution.

After describing the channel model, we addressed the pilot based system. As shown in Chapter 2, a pilot code of length J was injected to each frame. This code was generated using a Walsh code matrix. With the aid of this pilot code, we were able to track down the behavior of the channel assigned to each sub-carrier. The channel estimator block estimated the frequency response of the channel impulse response on an iterative basis. On the first iteration, we scaled up the channel impulse response, in order to compensate for the missing energy. This was done because we have a length N channel frequency response and a length L channel impulse response. We then used the estimated channel to clean out the received vector from MAI. This process is called Parallel Interference Cancellation. We also used MMSE filter to further reduce the effect of noise on the received signal. The PIC and channel estimation used the intrinsic and extrinsic values which were supplied by the APP decoder. However, only extrinsic values were fed back to the MMSE filter. The APP decoder itself was realized by using the BCJR algorithm which optimized the estimation process on a symbol by symbol basis. Then, we simulated the pilot-based system using Simulink. The results showed a promising BER performance. However, a saturation effect was observed. Upon reaching saturation, we did not get a significant improvement of BER performance by increasing the number of iterations. We also demonstrated that by using a larger number of sub-carriers we can obtain better performance. Although a system with a large number of sub-carriers can estimate the channel better than its counterpart, we need to keep in mind that these sub-carriers need to be orthogonal to each other.

In Chapter 3, we discussed the channel estimation without a pilot code. To do this, receiver needed to assume some knowledge about the transmitted signal. Namely, the statistical behavior of the transmitted signal. Having this information, the receiver could estimate the transmitted signal. The receiver itself was realized by two filters. The first filter was used to eliminate MAI and enhance the contribution of the transmitted symbol of the desired used over MAI and interference. After this, we applied CMA to obtain the estimate of the transmitted symbol. It was learned in the process that it is imperative for a blind system to ascertain that the initialization of the CMA's coefficients will produce the positive value for the cross correlation between the CMA filter and the channel impulse response. Should this requirement not be fulfilled, we will have phase ambiguity at the output of the CMA filter. To improve performance of the blind system, we operated the system on a frame basis, applied an APP decoder and an interleaver on the receiver end. The interleaver helped the system to deal with burst errors, while the APP decoder enhanced the system's immunity in AWGN. We showed via simulation that the system performs well under a fictitious channel with multipath.

For future research, we should investigate the possibility of blind channel estimation without the assumption of positive energy. This way, the blind system can be implemented in arbitrary environments. However, we must take account the complexity of the blind system. If the system is too complex, we cannot implement it in practice, simply because of its prohibitive cost.

BIBLIOGRAPHY

[1] T. Zemen, J. Wehinger, C. F. Mecklenbräuker and R. Müller, "Iterative Detection and Channel Estimation for MC-CDMA," *Proc. IEEE ICC*, vol. 5, Anchorage (AK), May 2003, pp. 3462-3466.

[2] L. Bahl, J. Cocke, F. Jelinek and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. 20, March 1974, pp. 284-287.

[3] T. Zemen, M. Loncar, J. Wehinger, C. F. Mecklenbräuker and R. Müller, "Improved Channel Estimation for Iterative Receiver," *IEEE GLOBECOM*, vol. 1, San Francisco (CA), December 2003, pp. 257-261.

[4] O. Edfors, M. Sandell, J.-J. van de Beek, D. Landström, and F. Sjöberg, "An introduction to orthogonal frequency-division multiplexing**,"** *Research Report TULEA 1996:16., Div. of Signal Processing*. Luleå University of Technology, September 1996.

[5] J. Lindner, "MC-CDMA and its Relational to General Multiuser/Multisubchannel Transmission Systems," in *IEEE 4th International Symposium*, vol. 1, Mainz (Germany), September 1996, pp. 115 – 121.

[6] R. Koetter, A. C. Singer and M. Tüchler, "Turbo Equalization," *IEEE Signal Processing Magazine*, vol. 21, January 2004, pp. 67-80.

[7] J. Lindner, M. Nold, W. G. Teich and M. Schreiner, "MC-CDMA and OFDMA for Indoor Communications: The Influence of Multiple Receiving Antenne," *Proc. IEEE 5th International Symposium*, vol. 1, Sun City (South Africa), September 1998, pp. 189-194.

[8] P. Robertson, E. Villebrun and P. Hoeher, "A comparison of Optimal and Sub-Optimal MAP Decoding Algorithms Operating in the Log Domain," *IEEE International Conference on Communication*, vol. 2, Seattle(WA), June 1995, pp. 1009 – 1013.

[9] L. Papke, P. Robertson and E. Villebrun, "Improved decoding with the SOVA in a parallel concatenated (Turbo-code) scheme," *IEEE International Conference on Communication*, vol. 1, Dallas(TX), June 1996, pp. 102 – 106.

[10] A. C. McCormick and E. A. Al-Susa, "Multicarrier CDMA for future generation mobile communication," *Electrical & Communication Engineering Journal*, vol. 14, no. 2, pp. 52-60, April 2002.

[11] T. Zemen, C. F. Mecklenbräuker and R. R. Müller, "Time Variant Channel Equalization for MC-CDMA via Fourier Basis Functions," *in MC-SS Workshop 2003*, Oberpaffenhofen (Germany), September 2003, pp. 451-458.

[12] M. Kobayashi, J. Boutros and G. Caire, "Successive Interference Cancellation with SISO Decoding and EM Channel Estimation," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 8, August 2001.

[13] T. A. Kadous and A. M. Sayeed, "A new scheme for MC-CDMA systems in the presence of imperfections," *34th Asilomar Conference*, 2000.

[14] C. R. Johnson, et al, "Blind Equalization Using the Constant Modulus Criterion: A Review," *Proc. IEEE*, vol. 86, no. 10, October 1998, pp. 1927 – 1950.

[15] R. E. Kamel and Y. Bar-Ness, "Anchored Blind Equalization Using the Constant Modulus Algorithm," *IEEE Transaction on Circuits and System II*, vol. 44, no. 5, May 1997, pp. 397 – 403.

[16] O. Shalvi and E. Weinstein, "New Criteria for Blind Deconvolution of Nonminimum Phase Systems (Channels)," *IEEE Transaction on Information Theory*, vol. 36, no. 2, March 1990, pp. 312 – 321.

[17] Y. Sato, "A Method of Self-Recovering Equalization for Multilevel Amplitude-Modulation Systems," *IEEE Transaction on Communications*, vol. 23, no. 6, June 1975, pp. 679 – 682.

[18] S. Bellini, "Bussgang Techniques for Blind Equalization," *IEEE Global Telecommunication Conf. Rec.*, December 1986, pp. 1634 – 1640.

[19] D. Darsena, G. Gelli, L. Paura and F. Verde, "Blind Multiuser Detection for MC-CDMA Systems," *36th Asilomar*, vol. 2, November 2002, pp. 1419 – 1423.

[20] S. Fiori and G. Maiolini, "Weighted least-squares blind deconvolution of non-minimum phase systems," *IEE Proceedings-Vision, Image and Signal Processing*, vol. 147, no. 6, December 2000, pp. 557 – 563.

[21] C. A. Haro, J. A. R. Fonollosa and J. R. Fonollosa, " Blind Channel Estimation and Data Detection Using Hidden Markov Models," *IEEE Transactions on Signal Processing*, vol. 45, no. 1, January 1997, pp. 241 – 247.

[22] J. G. Proakis and C. L. Nikias, "Blind Equalization," in *Proc. SPIE Adaptive Signal Processing*, vol. 1565, 1991, pp. 76 – 87.

[23] R. Vallet, "Symbol by symbol MAP detection and the Baum-Welch identification algorithm in digital communications," in *Proc. Europ. Signal Processing Conf.*, September 1992, pp. 131 – 134.

[24] A. Benveniste, "Blind Equalizer," *IEEE Transaction on Communications*, vol. 32, no. 8, August 1984, pp. 871 – 883.

[25] G. K. Lee and S. B. Gelfand, "Bayesian Techniques for Blind Deconvolution," *IEEE Transaction on Communications*, vol. 44, no. 7, July 1996, pp. 826 – 835.

[26] G. Gelli, L. Paura and F. Verde, "A Two-Stage CMA-Based Receiver for Blind Joint Equalization and Multiuser Detection in High Data-Rate DS-CDMA Systems," *IEEE Transaction on Wireless Communications*, vol. 3, no. 4, July 2004, pp. 1209 – 1223.

[27] D.N. Godard, "Self – recovering equalization and carrier tracking in two-dimensional data communications systems," *IEEE Trans. Commun.*, vol. COM-28, November 1980, pp. 1967 – 1875.

[28] J. R. Treichler and M. G. Agee, "A new approach to multipath correction of constant modulus signals," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 31, April 1983, pp. 349 – 472.

[29] J. R. Treichler and M. G. Larimore, "New processing techniques based on the constant modulus adaptive algorithm," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 33, April 1985, pp. 420 – 431.

[30] B. Baykal, "Blind Channel Estimation via Combining Autocorrelation and Blind Phase Estimation," *IEEE Transaction on Circuits and Systems*, vol. 51, no. 6, June 2004, pp. 1125 – 1131.

[31] V. Koivunen, J. Laurila and E. Bonek, "Blind Methods for Wireless Communication Receivers," *Review of Radio Science 1999-2002*. New York, NY: John Wiley & Sons, pp. 247 – 273.

[32] P. Zong, "Signal Processing Topics in Multicarrier Modulation: Frequency Offset Correction for OFDM Multiuser Interface Cancellation for MC-CDMA," *Master Thesis*. New Jersey: NJIT, May 1998.

[33] M. K. Simon, J. K. Omura, R. A. Scholtz and B. K. Levitt, *Spread Spectrum Communications Handbook*. New York, NY: McGraw-Hill, 1994.

[34] S. Sheng and R. Brodersen, *Low-Power CMOS Wireless Communications: A Wideband CDMA System Design*. Norwell, MA: Kluwer Academic Publishers, 1998.

[35] J.G. Proakis, *Digital Communications*. 4th ed. New York, NY: McGraw-Hill, 2000.

[36] D. M. Pozar, *Microwave and RF Design of Wireless Systems*. New York, NY: John Wiley & Sons, Inc, 2000

[37] W. C. Jakes, *Microwave Mobile Communications*. New York, NY: John Wiley & Sons, 1974.

[38] M. Schwartz, W. R. Bennett and S. Stein, *Communication Systems and Techniques*. New York, NY: McGraw-Hill, Inc, 1966.

[39] K. Pahlavan and A. H. Levesque, *Wireless Information Networks*. New York, NY: John Wiley & Sons, Inc, 1995.

[40] S. Haykin, *Adaptive Filter Theory*. 2nd ed. Englewood Cliffs, NJ: Prentice Hall, 1991.

[41] S. Haykin, *Introduction to Adaptive Filters*. New York, NY: Macmillan Publishing Company, 1984.

[42] "External Interfaces Reference," *Mathworks*. 2004. <http://www.mathworks.com/access/helpdesk/help/techdoc/apiref/apiref.html>

[43] "Frequently Ask Questions", *Digital Radio Mondiale*. <http://www.drm.org/newsevents/faqs>

[44] J. Meel, "Spread Spectrum: Introduction," *Spread Spectrum Scene*., October 1999 <http://www.sss-mag.com/pncodes.html>

[45] D. Smalley, "Equalization Concepts: A Tutorial,", October 1994 <http://www.ek.isy.liu.se/equalization/1999NOV26_RFD_DSP_AN.pdf>

APPENDICES

These appendices will list the model that is used in Simulink. The External Interface listing codes using C are also enlisted.

We are going to start with the pilot-based model system then the blind equalization system.

A. Pilot-based System

Following is the important parameter setting in startup stage for the pilot-based model system:

M = frame size (number of chips per frame).

N = subcarrier length.

J = pilot code length per frame.

R = convolutional encoder rate.

L = channel impulse response length.

K = number of users which are served in the system.

awgnvar = variance of the additive white Gaussian noise

source_sample_time = sample time of input source for each user.

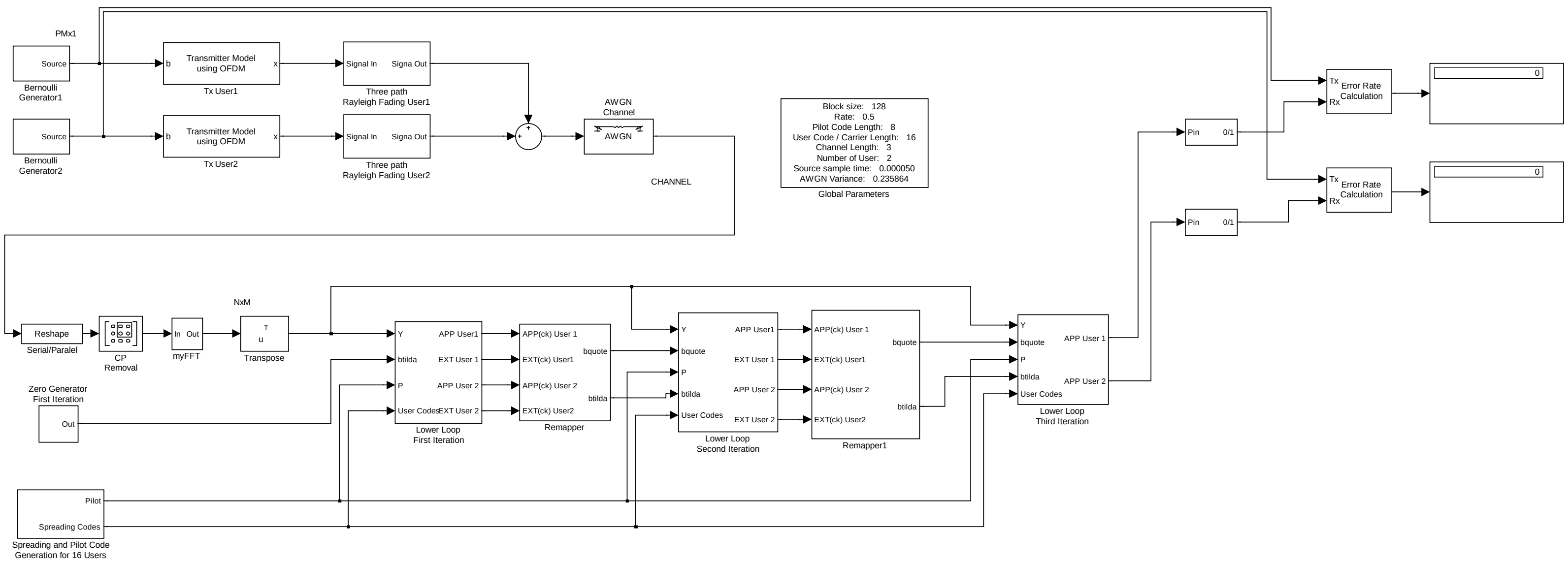
These parameters were set by using of file "my_setup_multiuser.m". The "Global Parameter" block set does this for us. We use the following S-Functions (C-MEX file):

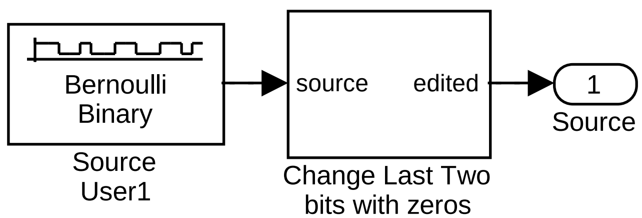
1. ColumnNorm.c -> Normalizes the column of a given matrix to a given value.
2. myDiagonalize.c -> Makes the vector input matrix to its corresponding diagonal matrix
3. LAPPDecoderBCJRbit.c -> Outputs the decoded A-Posteriori Probability of a given convolutionally encoded input source.
4. LEXTDecoderBCJRcoded.c -> Outputs the encoded A-Posteriori Probability and its Extrinsic value of a given convolutionally encoded input source.

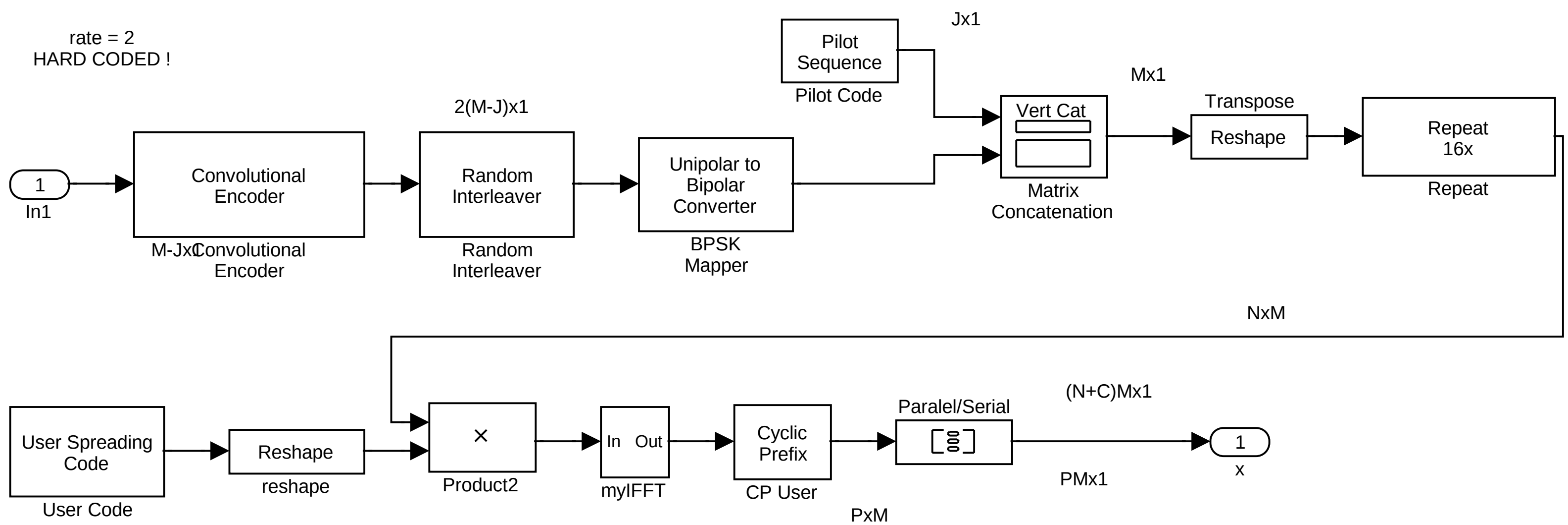
Here, we list the "my_setup_multiuser.m" and the detail of pilot-based model.

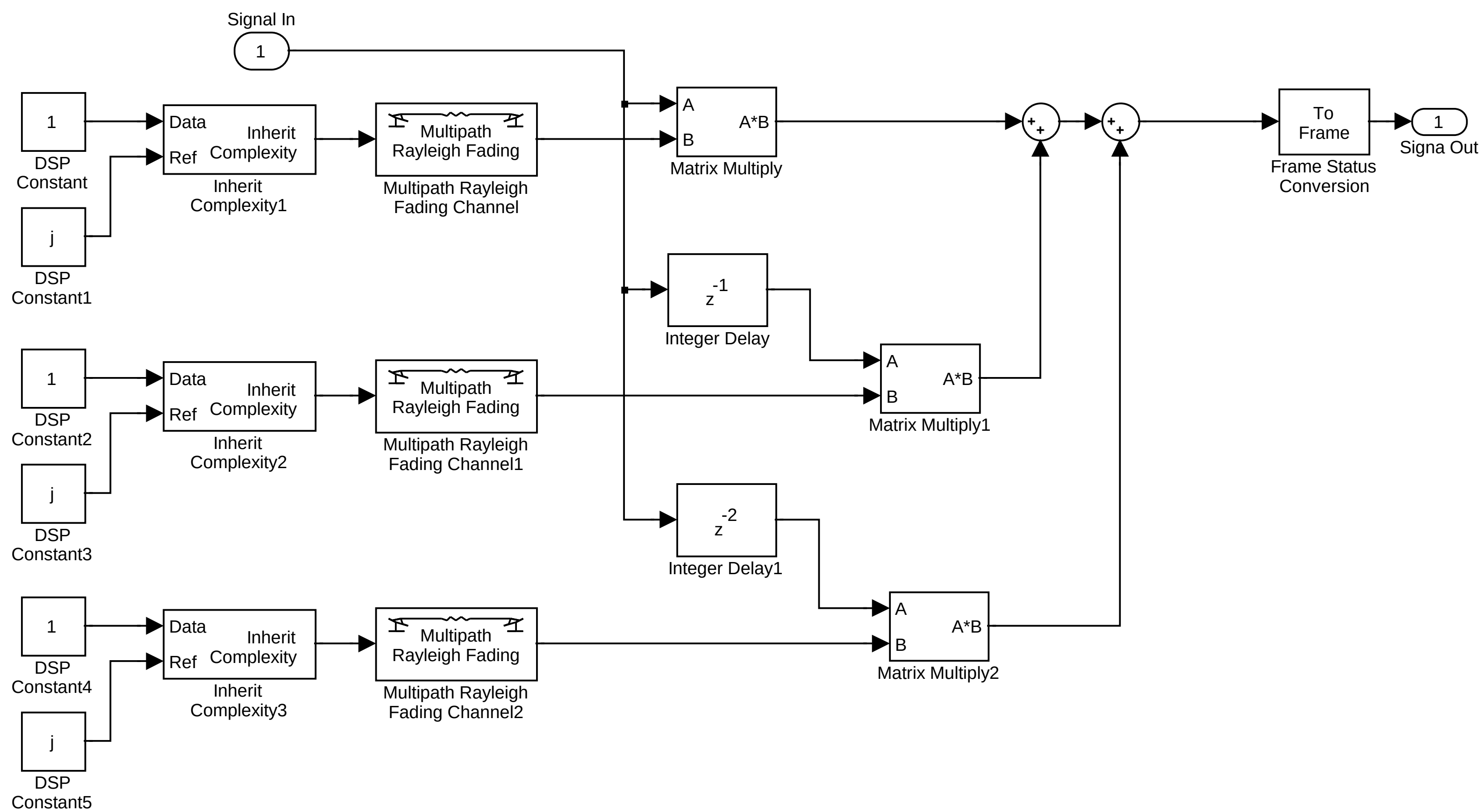
Function my_setup_multiuser

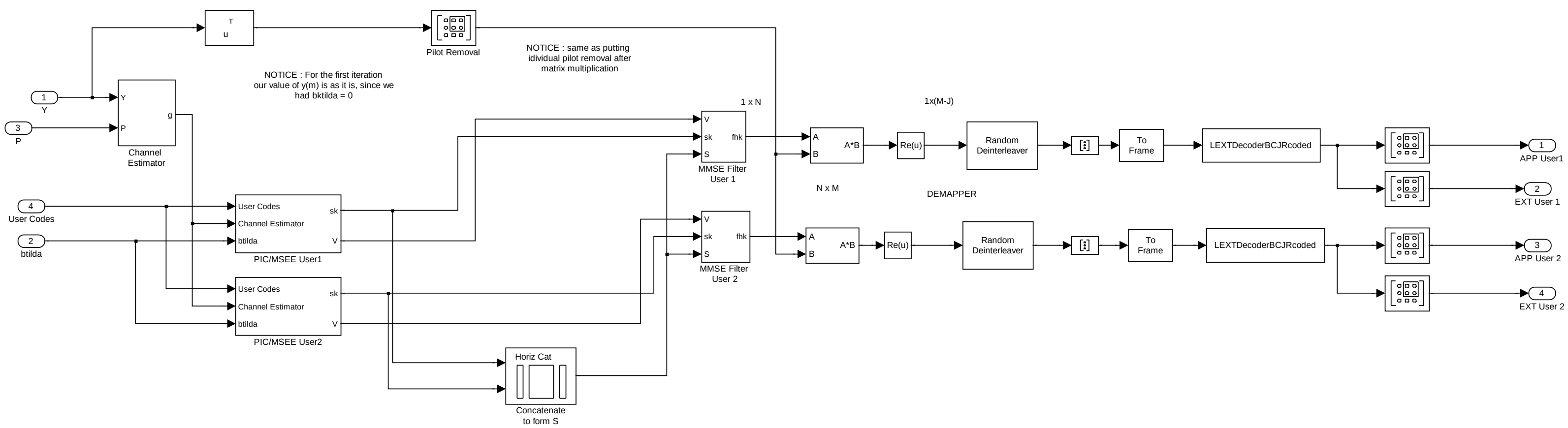
```
% Get parameter names and values from mask
%Hard coded as if the K is the number of user in the system
mask_ws_vars = get_param([gcs '/Global Parameters'],'maskwsvariables');
if ~isempty(mask_ws_vars)
    for i = 1:length(mask_ws_vars),
        curr_var = mask_ws_vars(i).Name;
          if(curr_var == 'K')
              num_user = mask_ws_vars(i).Value;
          end
        evalin('base',[curr_var ' = ' num2str(mask_ws_vars(i).Value) ';']);
    end
    evalin('base','input_samples_per_frame = (M-J)/(1/R);');
    evalin('base','frame_period = input_samples_per_frame*source_sample_time;');
else
end
%Filling the random interleaver seeding
for i = 1:num_user
  evalin('base',['rnd_inter_seed(' num2str(i) ') = ' num2str(get_param([gcs '/Tx User'
            num2str(i)],'rand_interleave_seed')) ';']);
end
```


PMx1
Source
Bernoulli Generator1
Source
Bernoulli Generator2
Transmitter Model using OFDM
Tx User1
Transmitter Model using OFDM
Tx User2
Signal In Signa Out
Three path Rayleigh Fading User1
Signal In Signa Out
Three path Rayleigh Fading User2
AWGN Channel
AWGN
CHANNEL
Block size: 128
Rate: 0.5
Pilot Code Length: 8
User Code / Carrier Length: 16
Channel Length: 3
Number of User: 2
Source sample time: 0.000050
AWGN Variance: 0.235864
Global Parameters
Error Rate Calculation
Error Rate Calculation
Pin 0/1
Pin 0/1
Reshape
Serial/Paralel
CP Removal
In Out
myFFT
NxM
Transpose
Zero Generator First Iteration
Out
Y
btilda
P
User Codes
APP User1
EXT User 1
APP User 2
EXT User 2
Lower Loop First Iteration
APP(ck) User 1
EXT(ck) User1
APP(ck) User 2
EXT(ck) User2
bquote
btilda
Remapper
Lower Loop Second Iteration
Remapper1
APP User 1
APP User 2
Lower Loop Third Iteration
Pilot
Spreading Codes
Spreading and Pilot Code Generation for 16 Users

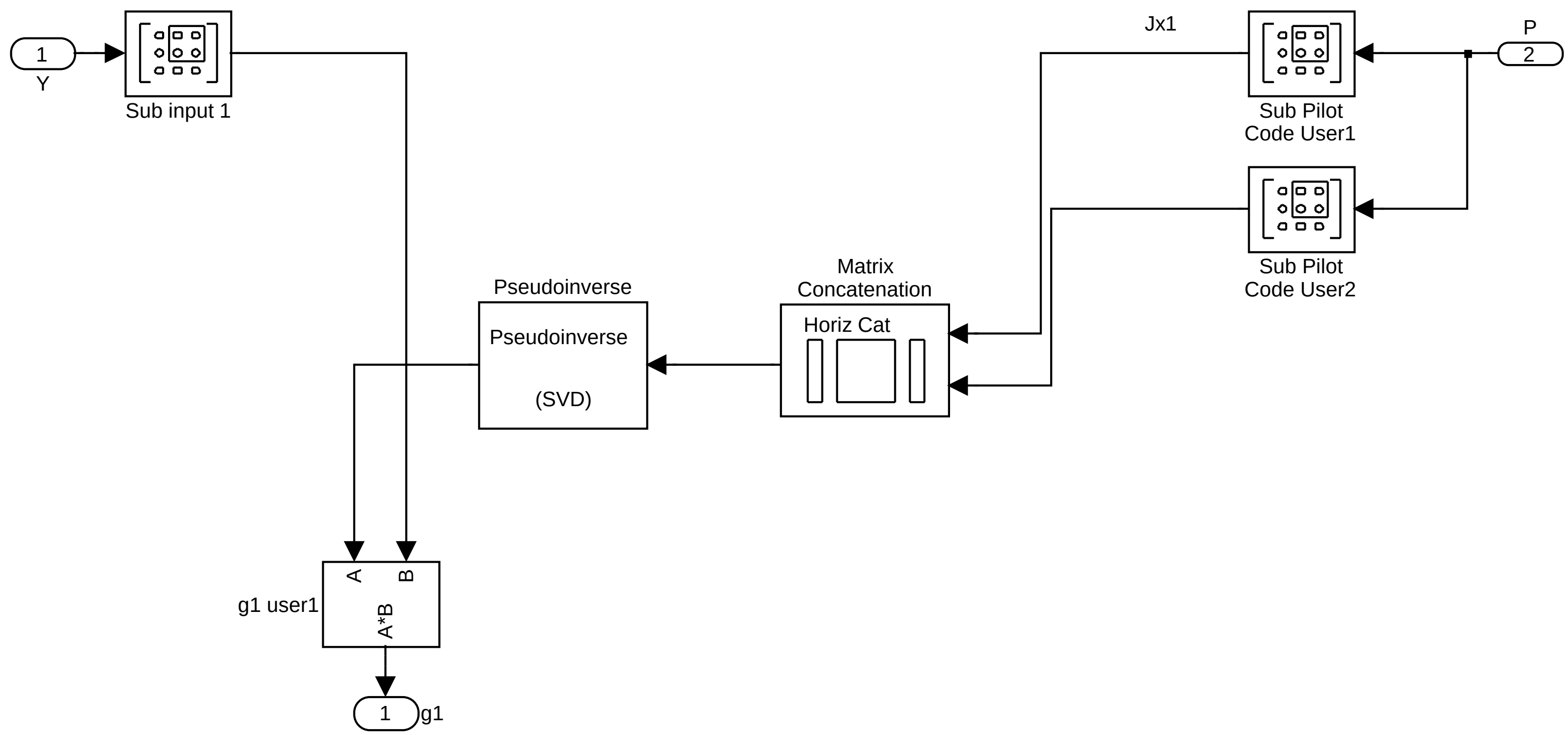

Bernoulli
Binary
Source
User1
source
edited
Change Last Two
bits with zeros
1
Source


rate = 2
HARD CODED !
1
In1
Convolutional Encoder
M-Jx1
Convolutional Encoder
2(M-J)x1
Random Interleaver
Random Interleaver
Unipolar to Bipolar Converter
BPSK Mapper
Pilot Sequence
Pilot Code
Jx1
Vert Cat
Matrix Concatenation
Mx1
Transpose
Reshape
Repeat 16x
Repeat
NxM
User Spreading Code
User Code
Reshape
reshape
×
Product2
In Out
myIFFT
Cyclic Prefix
CP User
PxM
Paralel/Serial
(N+C)Mx1
PMx1
1
x

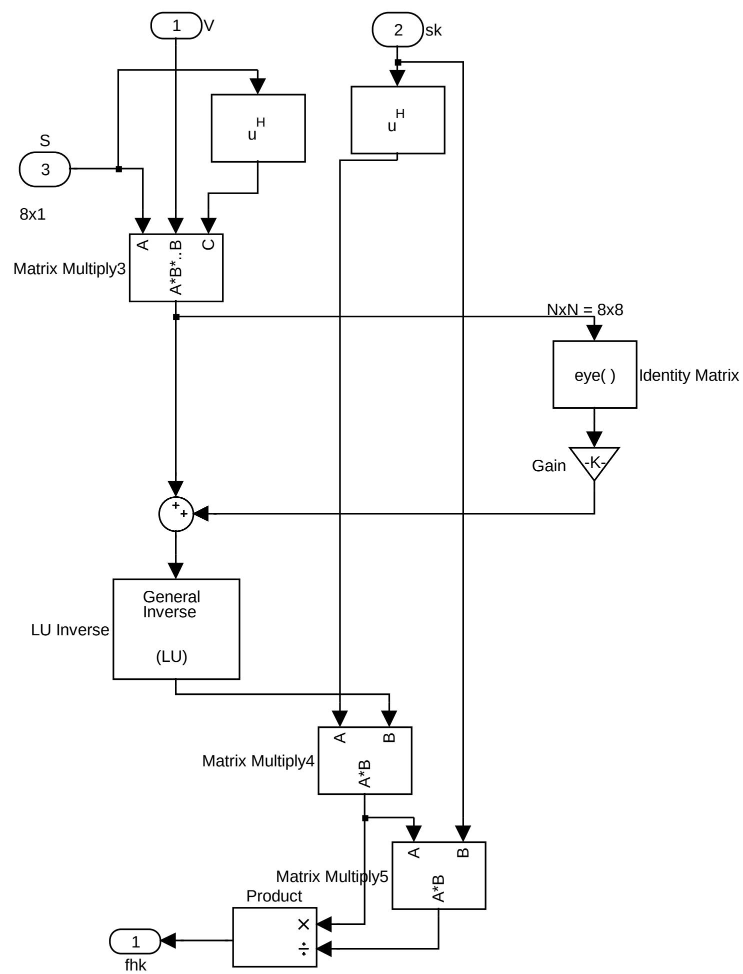

Signal In
1
1
DSP Constant
j
DSP Constant1
Data
Ref
Inherit Complexity
Inherit Complexity1
Multipath Rayleigh Fading
Multipath Rayleigh Fading Channel
A
B
A*B
Matrix Multiply
To Frame
Frame Status Conversion
1
Signa Out
z-1
Integer Delay
A
B
A*B
Matrix Multiply1
1
DSP Constant2
j
DSP Constant3
Data
Ref
Inherit Complexity
Inherit Complexity2
Multipath Rayleigh Fading
Multipath Rayleigh Fading Channel1
z-2
Integer Delay1
A
B
A*B
Matrix Multiply2
1
DSP Constant4
j
DSP Constant5
Data
Ref
Inherit Complexity
Inherit Complexity3
Multipath Rayleigh Fading
Multipath Rayleigh Fading Channel2

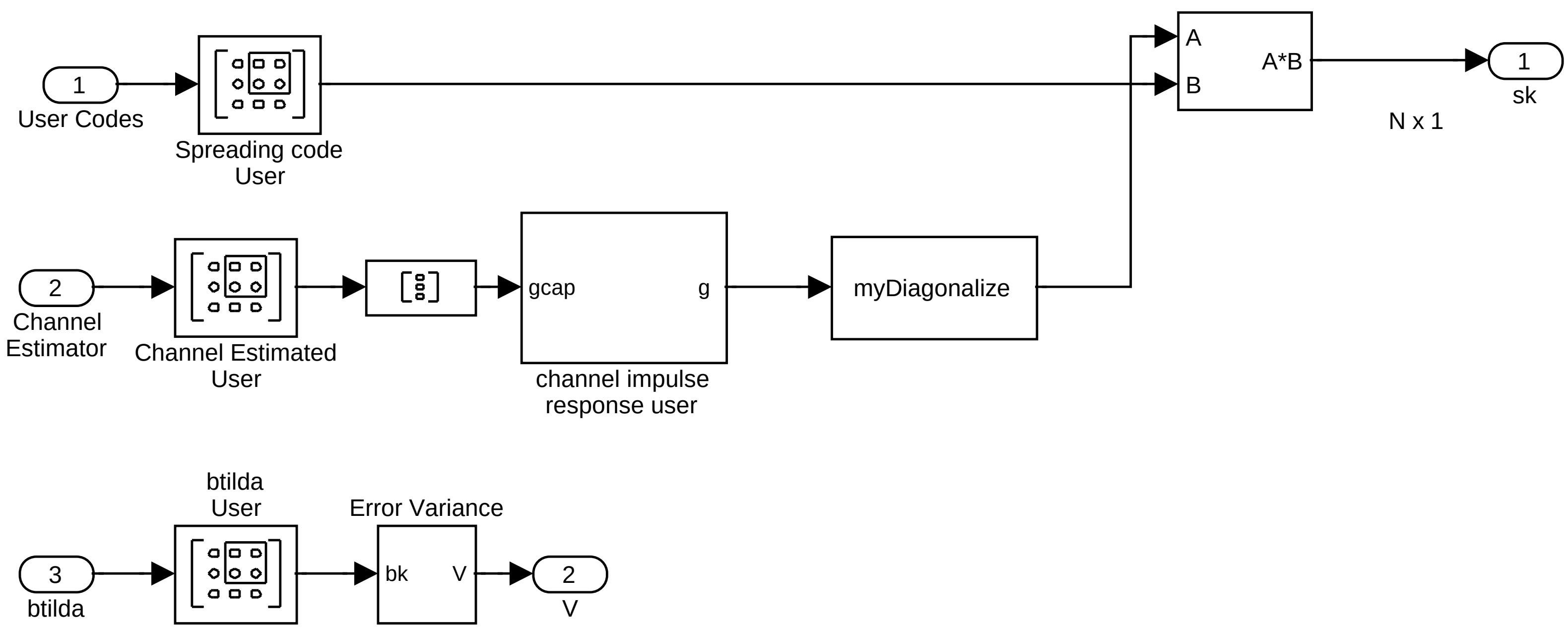


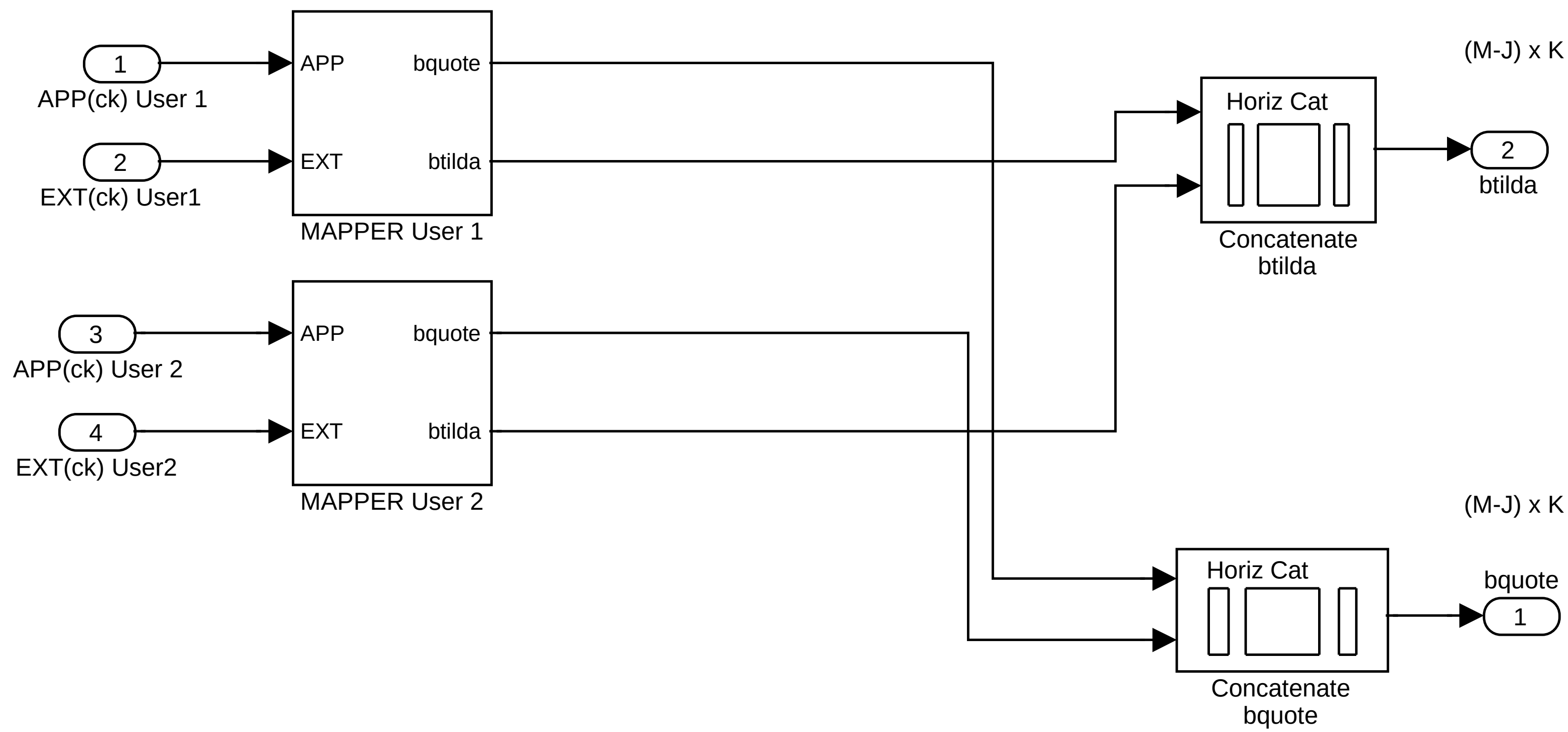
Pilot Removal
NOTICE : same as putting idividual pilot removal after matrix multiplication
NOTICE : For the first iteration our value of y(m) is as it is, since we had bktilda = 0
1 Y
3 P
Channel Estimator
4 User Codes
2 btilda
User Codes
Channel Estimator
btilda
PIC/MSEE User1
PIC/MSEE User2
MMSE Filter User 1
MMSE Filter User 2
Horiz Cat
Concatenate to form S
1 x N
N x M
1x(M-J)
A*B
Re(u)
Random Deinterleaver
To Frame
DEMAPPER
LEXTDecoderBCJRcoded
1 APP User1
2 EXT User 1
3 APP User 2
4 EXT User 2

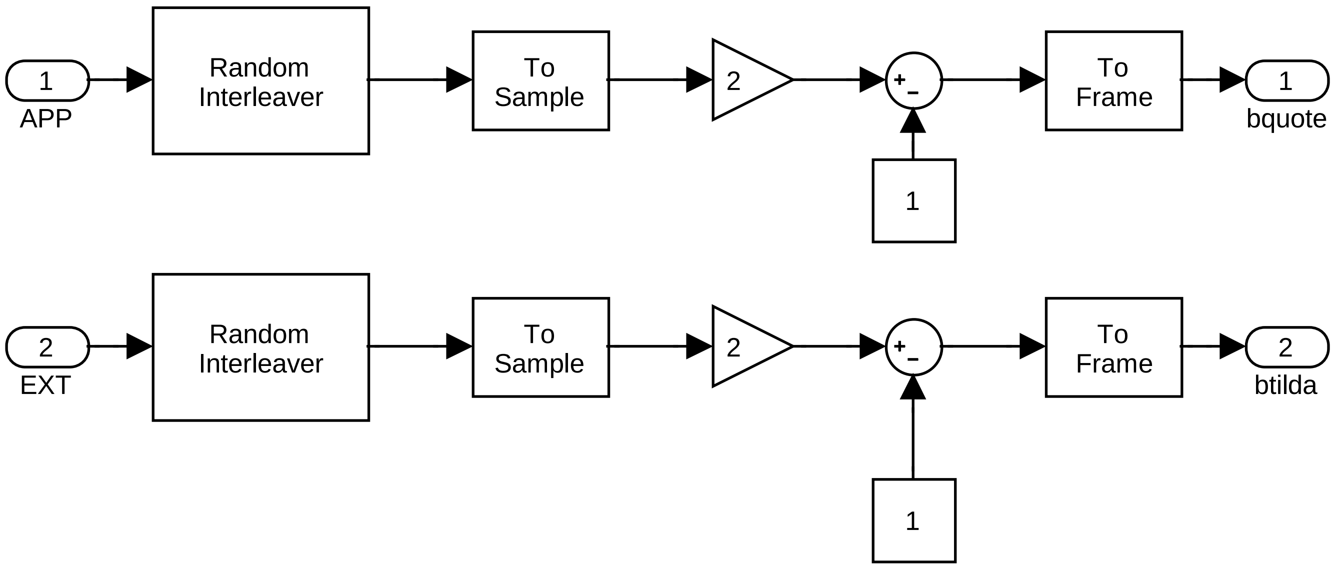


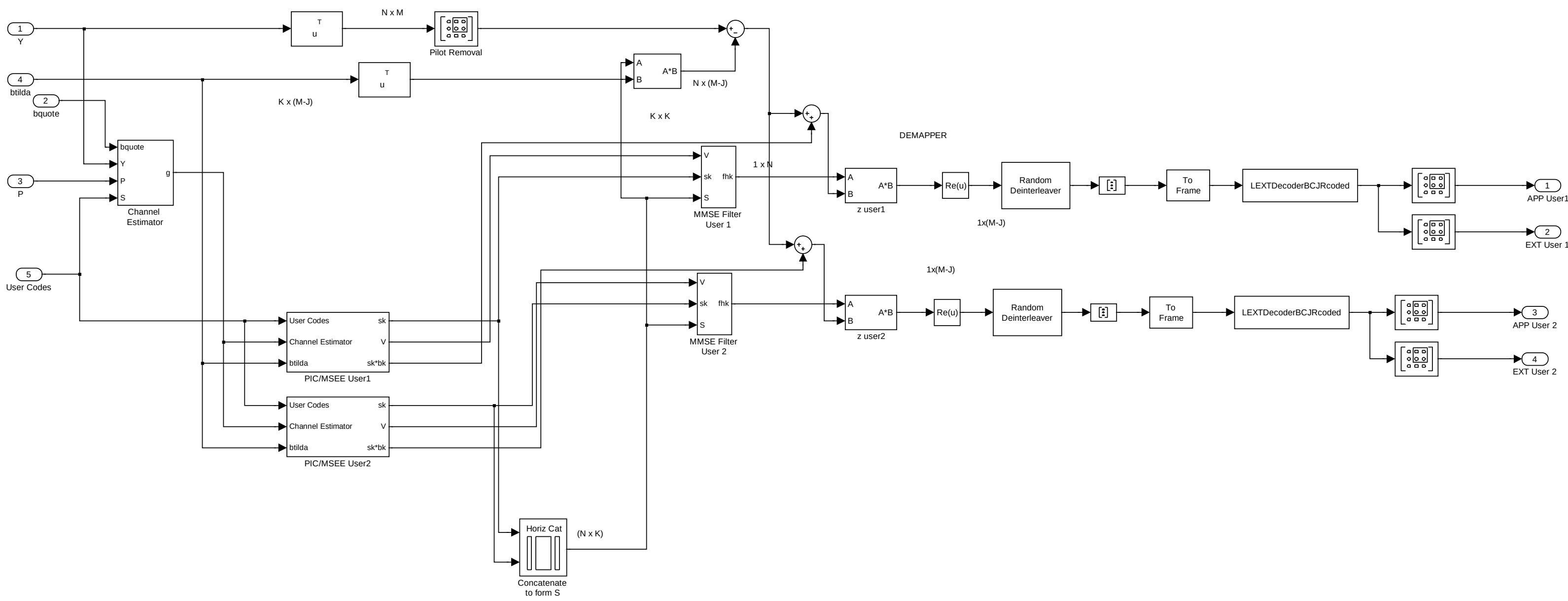
1
Y
Sub input 1
Jx1
P
2
Sub Pilot
Code User1
Sub Pilot
Code User2
Matrix
Concatenation
Horiz Cat
Pseudoinverse
Pseudoinverse
(SVD)
A
B
A*B
g1 user1
1
g1


1
V
2
sk
S
3
8x1
u^H
u^H
A
A*B*..B
C
Matrix Multiply3
NxN = 8x8
eye( )
Identity Matrix
Gain
-K-
+
+
General
Inverse
(LU)
LU Inverse
A
B
A*B
Matrix Multiply4
A
B
A*B
Matrix Multiply5
Product
×
÷
1
fhk

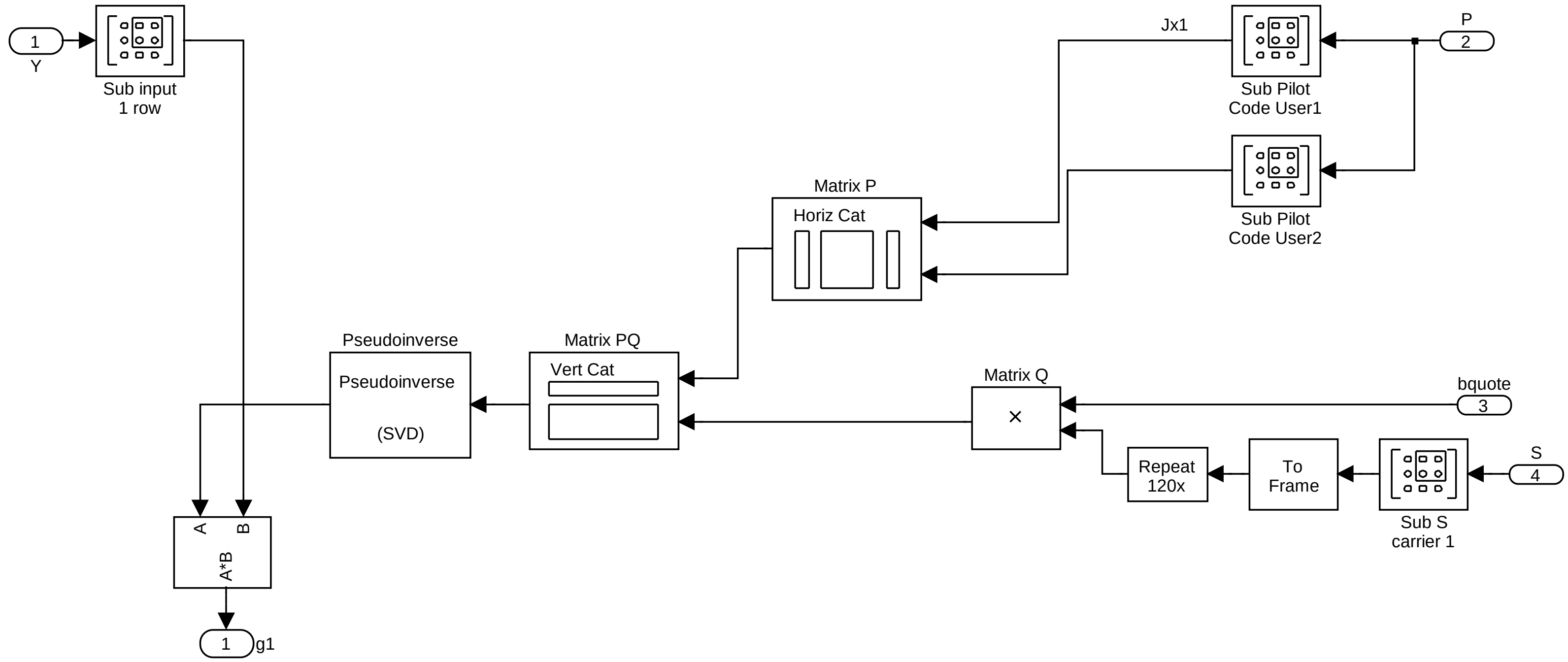

1
User Codes
Spreading code
User
A
B
A*B
1
sk
N x 1
2
Channel
Estimator
Channel Estimated
User
gcap
g
channel impulse
response user
myDiagonalize
btilda
User
Error Variance
3
btilda
bk
V
2
V

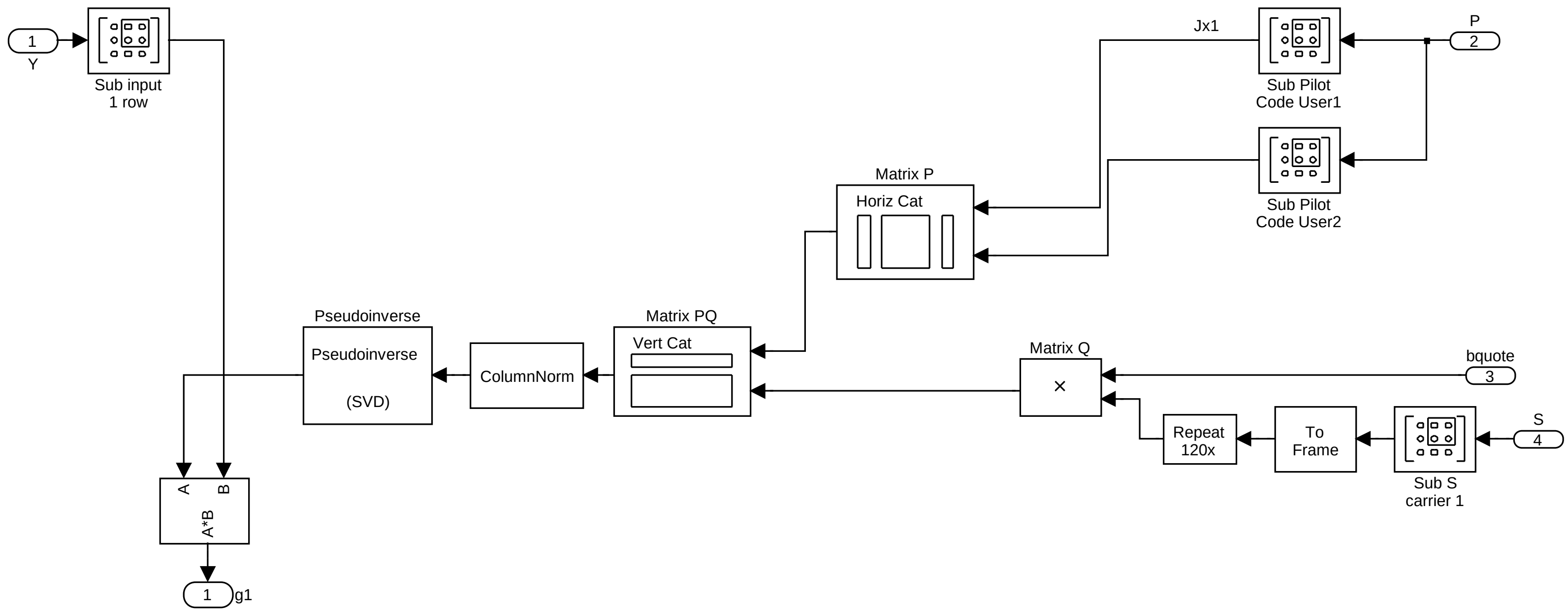

1
APP(ck) User 1
2
EXT(ck) User1
APP
EXT
bquote
btilda
MAPPER User 1
3
APP(ck) User 2
4
EXT(ck) User2
APP
EXT
bquote
btilda
MAPPER User 2
(M-J) x K
Horiz Cat
Concatenate
btilda
2
btilda
(M-J) x K
Horiz Cat
Concatenate
bquote
bquote
1


1
APP
Random
Interleaver
To
Sample
2
+
-
1
To
Frame
1
bquote
2
EXT
Random
Interleaver
To
Sample
2
+
-
1
To
Frame
2
btilda



Y
btilda
bquote
P
User Codes
N x M
Pilot Removal
K x (M-J)
N x (M-J)
K x K
A*B
Channel Estimator
MMSE Filter User 1
MMSE Filter User 2
1 x N
DEMAPPER
z user1
z user2
Re(u)
Random Deinterleaver
To Frame
LEXTDecoderBCJRcoded
1x(M-J)
APP User1
EXT User 1
APP User 2
EXT User 2
User Codes
Channel Estimator
btilda
sk
V
sk*bk
PIC/MSEE User1
PIC/MSEE User2
Horiz Cat
(N x K)
Concatenate to form S


1
Y
Sub input
1 row
Jx1
P
2
Sub Pilot
Code User1
Sub Pilot
Code User2
Matrix P
Horiz Cat
Pseudoinverse
Pseudoinverse
(SVD)
Matrix PQ
Vert Cat
Matrix Q
×
bquote
3
Repeat
120x
To
Frame
Sub S
carrier 1
S
4
A
B
A*B
1
g1


1
Y
Sub input
1 row
Jx1
P
2
Sub Pilot
Code User1
Sub Pilot
Code User2
Matrix P
Horiz Cat
Pseudoinverse
Pseudoinverse
(SVD)
ColumnNorm
Matrix PQ
Vert Cat
Matrix Q
×
bquote
3
Repeat
120x
To
Frame
Sub S
carrier 1
S
4
A
B
A*B
1
g1

B. Blind Equalization System

Following is the important parameter setting in startup stage for the pilot-based model system:

M = frame size (number of chips per frame).

N = subcarrier length.

J = pilot code length per frame.

R = convolutional encoder rate.

L = channel impulse response length.

K = number of users which are served in the system.

awgnvar = variance of the additive white Gaussian noise

source_sample_time = sample time of input source for each user.

correlation sample size = sample size of correlation calculation.

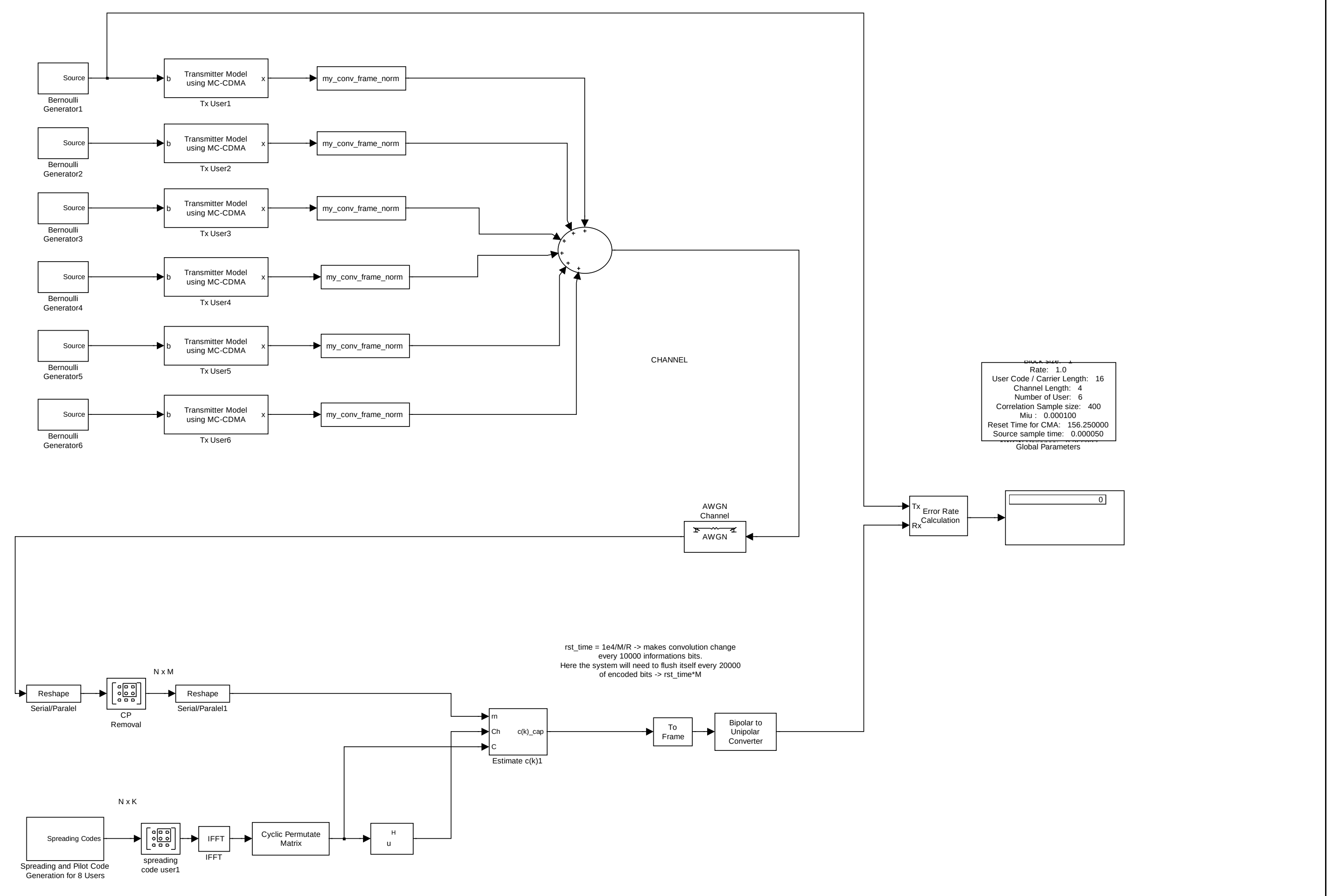
These parameters were set by using of file “setup_mucma.m”. The “Global Parameter” block set does this for us. We use the following S-Functions (C-MEX file):

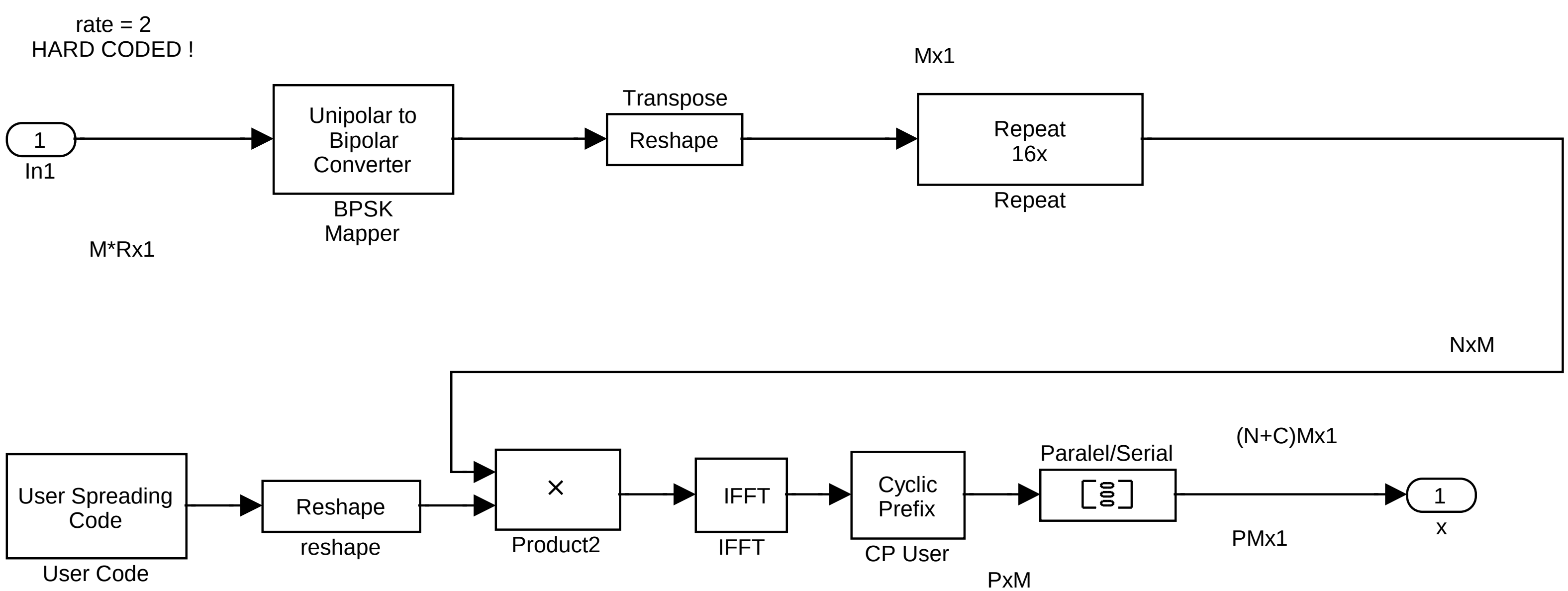
1. my_conv_frame_norm.c -> convolute the input sequence with a given channel impulse response in the field. The channel impulse response will be normalized first before the result is calculated.
2. my_cyclic_permutation_matrix.c -> outputs a permutated matrix version of the given input vector.
3. Discard_N_first_frame.c -> neglect the N first input from a given sequence. The outputs are exactly the same, thereafter.
4. CMA_frame.c -> operate the Constant Modulus Algorithm for a given input sequence and a given initial value of filter.

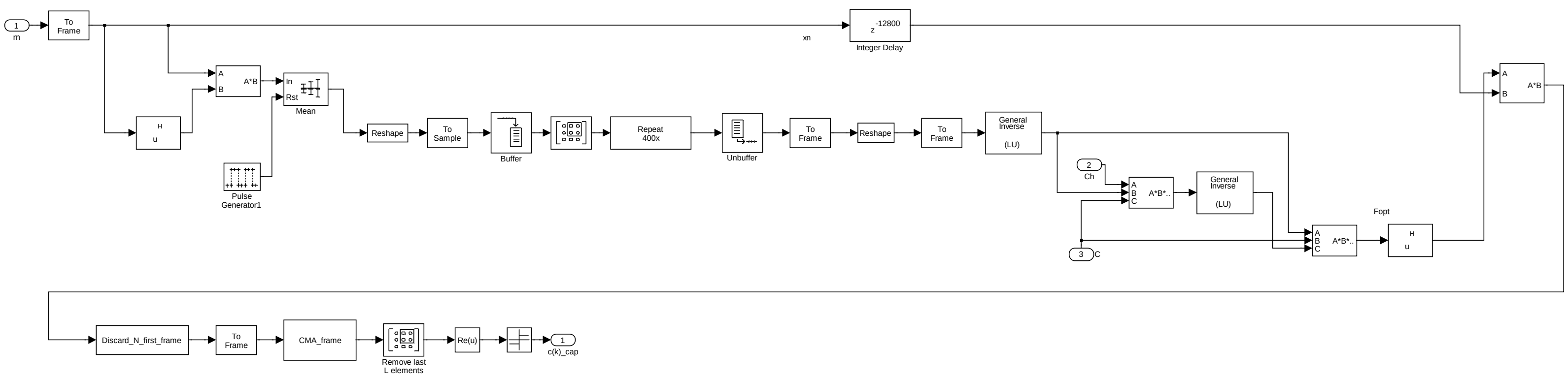
Here, we list the “setup_mucma.m” and the detail of blind-equalization model.

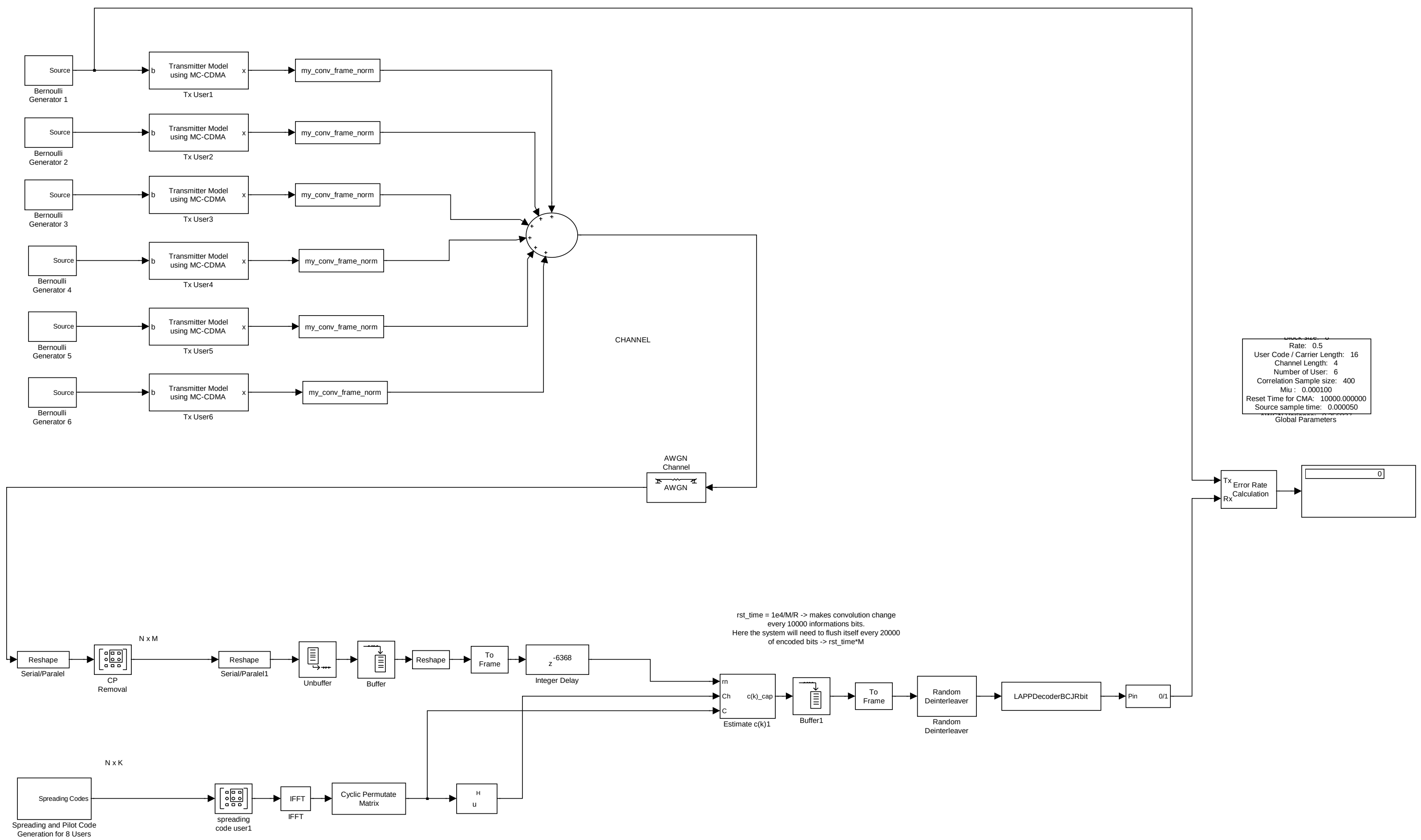
Function setup_mucma

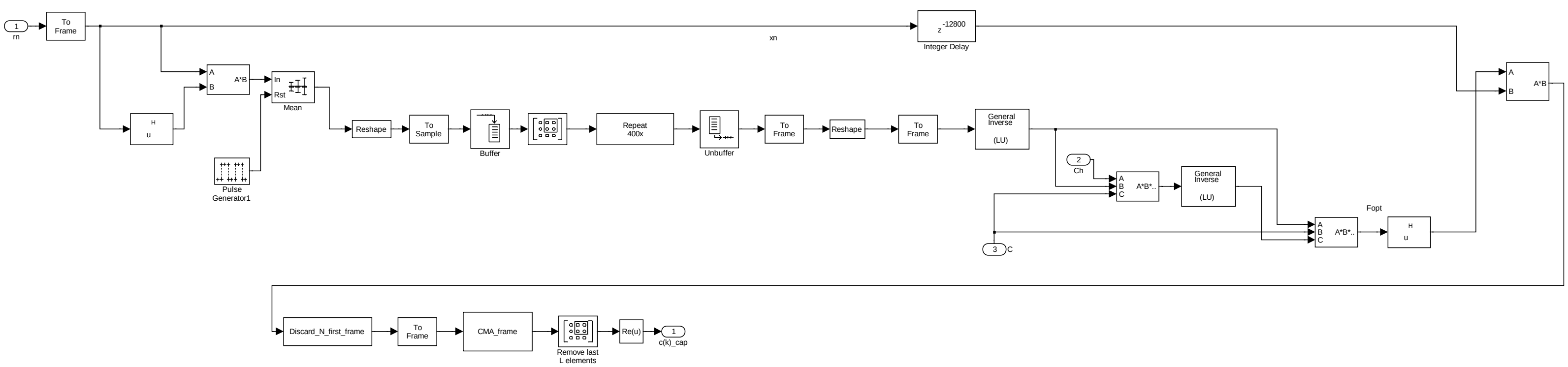
```
% Get parameter names and values from mask
%Hard coded as if the K is the number of user in the system
mask_ws_vars = get_param([gcs '/Global Parameters'],'maskwsvariables');

if ~isempty(mask_ws_vars)
   for i = 1:length(mask_ws_vars),
      curr_var = mask_ws_vars(i).Name;
        if(curr_var == 'K')
           num_user = mask_ws_vars(i).Value;
        end
      evalin('base',[curr_var ' = ' num2str(mask_ws_vars(i).Value) ';']);
   end
   evalin('base','input_samples_per_frame = (M)/(1/R);');
   evalin('base','frame_period = input_samples_per_frame*source_sample_time;');
else
end

%Filling the random interleaver seeding
% for i = 1:num_user
%   evalin('base',['rnd_inter_seed(' num2str(i) ') = ' num2str(get_param([gcs '/Tx User'
         num2str(i)],'rand_interleave_seed')) ';']);
% end
```


Source
Bernoulli Generator1
Source
Bernoulli Generator2
Source
Bernoulli Generator3
Source
Bernoulli Generator4
Source
Bernoulli Generator5
Source
Bernoulli Generator6
Transmitter Model using MC-CDMA
Tx User1
Tx User2
Tx User3
Tx User4
Tx User5
Tx User6
my_conv_frame_norm
CHANNEL
AWGN Channel
AWGN
Rate: 1.0
User Code / Carrier Length: 16
Channel Length: 4
Number of User: 6
Correlation Sample size: 400
Miu : 0.000100
Reset Time for CMA: 156.250000
Source sample time: 0.000050
Global Parameters
Error Rate Calculation
Tx
Rx
0
rst_time = 1e4/M/R -> makes convolution change every 10000 informations bits.
Here the system will need to flush itself every 20000 of encoded bits -> rst_time*M
Reshape
Serial/Paralel
CP Removal
N x M
Reshape
Serial/Paralel1
m
Ch
C
c(k)_cap
Estimate c(k)1
To Frame
Bipolar to Unipolar Converter
N x K
Spreading Codes
Spreading and Pilot Code Generation for 8 Users
spreading code user1
IFFT
IFFT
Cyclic Permutate Matrix
H
u


rate = 2
HARD CODED !
1
In1
Unipolar to Bipolar Converter
BPSK Mapper
M*Rx1
Transpose
Reshape
Mx1
Repeat 16x
Repeat
NxM
User Spreading Code
User Code
Reshape
reshape
×
Product2
IFFT
IFFT
Cyclic Prefix
CP User
PxM
Paralel/Serial
(N+C)Mx1
PMx1
1
x


1
rn
To Frame
H u
A B A*B
In Rst
Mean
Pulse Generator1
Reshape
To Sample
Buffer
Repeat 400x
Unbuffer
To Frame
Reshape
To Frame
General Inverse (LU)
xn
z-12800
Integer Delay
2
Ch
A B C A*B*..
General Inverse (LU)
3
C
A B C A*B*..
Fopt
H u
A B A*B
Discard_N_first_frame
To Frame
CMA_frame
Remove last L elements
Re(u)
1
c(k)_cap


Source
Bernoulli Generator 1
Transmitter Model using MC-CDMA
Tx User1
my_conv_frame_norm
Bernoulli Generator 2
Tx User2
Bernoulli Generator 3
Tx User3
Bernoulli Generator 4
Tx User4
Bernoulli Generator 5
Tx User5
Bernoulli Generator 6
Tx User6
CHANNEL
Rate: 0.5
User Code / Carrier Length: 16
Channel Length: 4
Number of User: 6
Correlation Sample size: 400
Miu : 0.000100
Reset Time for CMA: 10000.000000
Source sample time: 0.000050
Global Parameters
AWGN Channel
AWGN
Tx
Rx
Error Rate Calculation
0
rst_time = 1e4/M/R -> makes convolution change
every 10000 informations bits.
Here the system will need to flush itself every 20000
of encoded bits -> rst_time*M
N x M
Reshape
Serial/Paralel
CP Removal
Serial/Paralel1
Unbuffer
Buffer
To Frame
z-6368
Integer Delay
rn
Ch
C
c(k)_cap
Estimate c(k)1
Buffer1
Random Deinterleaver
LAPPDecoderBCJRbit
Pin
0/1
N x K
Spreading Codes
Spreading and Pilot Code Generation for 8 Users
spreading code user1
IFFT
Cyclic Permutate Matrix
H
u



1
m
To Frame
xn
z-12800
Integer Delay
A
B
A*B
In
Rst
Mean
H
u
Pulse Generator1
Reshape
To Sample
Buffer
Repeat 400x
Unbuffer
To Frame
Reshape
To Frame
General Inverse
(LU)
2
Ch
A
B
C
A*B*..
General Inverse
(LU)
3
C
Fopt
A
B
C
A*B*..
H
u
A
B
A*B
Discard_N_first_frame
To Frame
CMA_frame
Remove last L elements
Re(u)
1
c(k)_cap