Chapter 7

Techniques to Mitigate Fading Effects

7.1 Introduction

Apart from the better transmitter and receiver technology, mobile communications require signal processing techniques that improve the link performance. Equalization, Diversity and channel coding are channel impairment improvement techniques. Equalization compensates for Inter Symbol Interference (ISI) created by multipath within time dispersive channels. An equalizer within a receiver compensates for the average range of expected channel amplitude and delay characteristics. In other words, an equalizer is a filter at the mobile receiver whose impulse response is inverse of the channel impulse response. As such equalizers find their use in frequency selective fading channels. Diversity is another technique used to compensate fast fading and is usually implemented using two or more receiving antennas. It is usually employed to reduce the depths and duration of the fades experienced by a receiver in a flat fading channel. Channel coding improves mobile communication link performance by adding redundant data bits in the transmitted message.At the baseband portion of the transmitter, a channel coder maps a digital message sequence in to another specific code sequence containing greater number of bits than original contained in the message. Channel Coding is used to correct deep fading or spectral null. We discuss all three of these techniques in this chapter. A general framework of the fading effects and their mitigation techniques is shown in Figure 7.1.

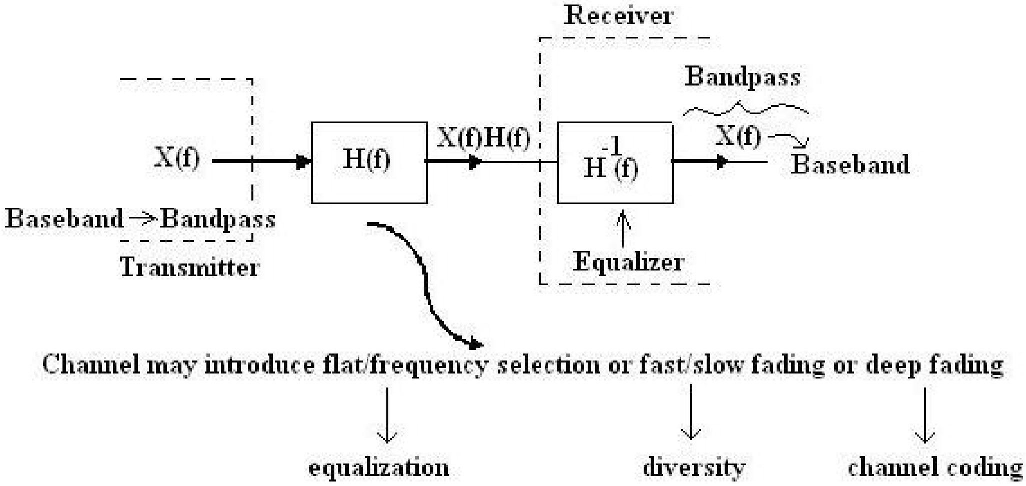

Figure 7.1: A general framework of fading effects and their mitigation techniques.

7.2 Equalization

ISI has been identified as one of the major obstacles to high speed data transmission over mobile radio channels. If the modulation bandwidth exceeds the coherence bandwidth of the radio channel (i.e., frequency selective fading), modulation pulses are spread in time, causing ISI. An equalizer at the front end of a receiver compensates for the average range of expected channel amplitude and delay characteristics. As the mobile fading channels are random and time varying, equalizers must track the time-varying characteristics of the mobile channel and therefore should be time-varying or adaptive. An adaptive equalizer has two phases of operation: training and tracking. These are as follows.

Training Mode:

- Initially a known, fixed length training sequence is sent by the transmitter so that the receiver equalizer may average to a proper setting.
- Training sequence is typically a pseudo-random binary signal or a fixed, of prescribed bit pattern.
- The training sequence is designed to permit an equalizer at the receiver to acquire the proper filter coefficient in the worst possible channel condition. An adaptive filter at the receiver thus uses a recursive algorithm to evaluate

the channel and estimate filter coefficients to compensate for the channel.

Tracking Mode:

- When the training sequence is finished the filter coefficients are near optimal.
- Immediately following the training sequence, user data is sent.
- When the data of the users are received, the adaptive algorithms of the equalizer tracks the changing channel.
- As a result, the adaptive equalizer continuously changes the filter characteristics over time.

7.2.1 A Mathematical Framework

The signal received by the equalizer is given by

$$x(t) = d(t) * h\left(t\right) + n_b\left(t\right) \tag{7.1}$$

where $d(t)$ is the transmitted signal, $h(t)$ is the combined impulse response of the transmitter,channel and the RF/IF section of the receiver and $n_b\left(t\right)$ denotes the baseband noise.

If the impulse response of the equalizer is $h_{eq}\left(t\right)$, the output of the equalizer is

$$\hat{y}\left(t\right) = d\left(t\right) * h\left(t\right) * h_{eq}\left(t\right) + n_b\left(t\right) * h_{eq}\left(t\right) = d\left(t\right) * g\left(t\right) + n_b\left(t\right) * h_{eq}\left(t\right). \tag{7.2}$$

However, the desired output of the equalizer is $d(t)$ which is the original source data. Assuming $n_b\left(t\right)$=0, we can write $y(t) = d(t)$, which in turn stems the following equation:

$$g\left(t\right) = h\left(t\right) * h_{eq}\left(t\right) = \delta\left(t\right) \tag{7.3}$$

The main goal of any equalization process is to satisfy this equation optimally. In frequency domain it can be written as

$$H_{eq}\left(f\right) H\left(f\right) = 1 \tag{7.4}$$

which indicates that an equalizer is actually an inverse filter of the channel. If the channel is frequency selective, the equalizer enhances the frequency components with small amplitudes and attenuates the strong frequencies in the received frequency spectrum in order to provide a flat, composite received frequency response and linear phase response. For a time varying channel, the equalizer is designed to track the channel variations so that the above equation is approximately satisfied.

7.2.2 Zero Forcing Equalization

In a zero forcing equalizer, the equalizer coefficients c_n are chosen to force the samples of the combined channel and equalizer impulse response to zero. When each of the delay elements provide a time delay equal to the symbol duration T, the frequency response $H_{eq}(f)$ of the equalizer is periodic with a period equal to the symbol rate 1/T. The combined response of the channel with the equalizer must satisfy Nyquist's criterion

$$H_{ch}(f)\, H_{eq}(f) = 1, |f| < {}^{1}\!/_{2T} \tag{7.5}$$

where $H_{ch}(f)$ is the folded frequency response of the channel. Thus, an infinite length zero-forcing ISI equalizer is simply an inverse filter which inverts the folded frequency response of the channel.

Disadvantage: Since $H_{eq}(f)$ is inverse of $H_{ch}(f)$ so inverse filter may excessively amplify the noise at frequencies where the folded channel spectrum has high attenuation, so it is rarely used for wireless link except for static channels with high SNR such as local wired telephone. The usual equalizer model follows a time varying or adaptive structure which is given next.

7.2.3 A Generic Adaptive Equalizer

The basic structure of an adaptive filter is shown in Figure 7.2. This filter is called the transversal filter, and in this case has N delay elements, N+1 taps and N+1 tunable complex multipliers, called weights. These weights are updated continuously by an adaptive algorithm. In the figure the subscript k represents discrete time index. The adaptive algorithm is controlled by the error signal e_k. The error signal is derived by comparing the output of the equalizer, with some signal d_k which is replica of transmitted signal. The adaptive algorithm uses e_k to minimize the cost function and uses the equalizer weights in such a manner that it minimizes the cost function iteratively. Let us denote the received sequence vector at the receiver and

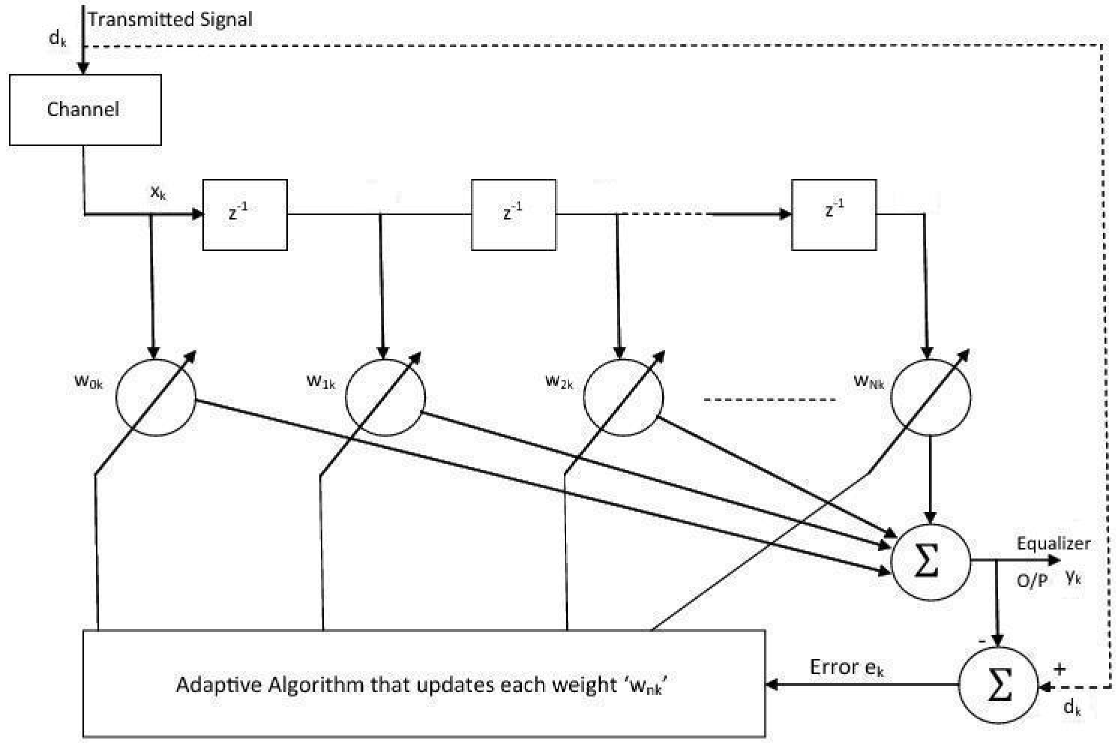

Figure 7.2: A generic adaptive equalizer.

the input to the equalizer as

$$\mathbf{x}_k = [x_k, x_{k-1}, ..., x_{k-N}]^T, \tag{7.6}$$

and the tap coefficient vector as

$$\mathbf{w}_k = [w_k^0, w_k^1, ..., w_k^N]^T. \tag{7.7}$$

Now, the output sequence of the equalizer y_k is the inner product of $\mathbf{x}_k$ and $\mathbf{w}_k$, i.e.,

$$y_k = \langle \mathbf{x}_k, \mathbf{w}_k \rangle = \mathbf{x}_k^T \mathbf{w}_k = \mathbf{w}_k^T \mathbf{x}_k. \tag{7.8}$$

The error signal is defined as

$$e_k = d_k - y_k = d_k - \mathbf{x}_k^T \mathbf{w}_k. \tag{7.9}$$

Assuming d_k and $\mathbf{x}_k$ to be jointly stationary, the Mean Square Error (MSE) is given as

$$\begin{aligned} MSE = E[e_k^2] &= E[(d_k - y_k)^2] \\ &= E[(d_k - \mathbf{x}_k^T \mathbf{w}_k)^2] \\ &= E[d_k^2] + \mathbf{w}_k^T E[\mathbf{x}_k \mathbf{x}_k^T] \mathbf{w}_k - 2E[d_k \mathbf{x}_k^T] \mathbf{w}_k \end{aligned} \tag{7.10}$$

where $\mathbf{w}_k$ is assumed to be an array of optimum values and therefore it has been taken out of the $E()$ operator. The MSE then can be expressed as

$$MSE = \xi = \sigma_k^2 + \mathbf{w}_k^T \mathbf{R} \mathbf{w}_k - 2\mathbf{p}^T \mathbf{w}_k \tag{7.11}$$

where the signal variance $\sigma_d^2 = E[d_k^2]$ and the cross correlation vector $\mathbf{p}$ between the desired response and the input signal is defined as

$$p = E\left[d_k \mathbf{x}_k\right] = E\left[\begin{array}{cccc} d_k x_k & d_k x_{k-1} & d_k x_{k-2} \cdots & d_k x_{k-N} \end{array}\right]. \tag{7.12}$$

The input correlation matrix $\mathbf{R}$ is defined as an $(N+1) \times (N+1)$ square matrix, where

$$R = E\left[\mathbf{x}_k \mathbf{x}_k^T\right] = E\left[\begin{array}{ccccc} x_k^2 & x_k x_{k-1} & x_k x_{k-2} & \cdots & x_k x_{k-N} \\ x_{k-1} x_k & x_{k-1}^2 & x_{k-1} x_{k-2} & \cdots & x_{k-1} x_{k-N} \\ x_{k-2} x_k & x_{k-2} x_{k-1} & x_{k-2}^2 & \cdots & x_{k-2} x_{k-N} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ x_{k-N} x_k & x_{k-N} x_{k-1} & x_{k-N} x_{k-2} & \cdots & x_{k-N}^2 \end{array}\right]. \tag{7.13}$$

Clearly, MSE is a function of $\mathbf{w}_k$. On equating $\frac{\partial \xi}{\partial w_k}$ to 0, we get the condition for minimum MSE (MMSE) which is known as Wiener solution:

$$\mathbf{w}_k = \mathbf{R}^{-1}\mathbf{p}. \tag{7.14}$$

Hence, MMSE is given by the equation

$$MMSE = \xi_{min} = \sigma_d^2 - \mathbf{p}^T \mathbf{w}_k. \tag{7.15}$$

7.2.4 Choice of Algorithms for Adaptive Equalization

Since an adaptive equalizer compensates for an unknown and time varying channel, it requires a specific algorithm to update the equalizer coefficients and track the channel variations. Factors which determine algorithm's performance are:

Rate of convergence: Number of iterations required for an algorithm, in response to a stationary inputs, to converge close enough to optimal solution. A fast rate of convergence allows the algorithm to adapt rapidly to a stationary environment of unknown statistics.

Misadjustment: Provides a quantitative measure of the amount by which the final value of mean square error, averaged over an ensemble of adaptive filters, deviates from an optimal mean square error.

Computational complexity: Number of operations required to make one complete iteration of the algorithm.

Numerical properties: Inaccuracies like round-off noise and representation errors in the computer, which influence the stability of the algorithm.

Three classic equalizer algorithms are primitive for most of today's wireless standards. These include the Zero Forcing Algorithm (ZF), the Least Mean Square Algorithm (LMS), and the Recursive Least Square Algorithm (RLS). Below, we discuss a few of the adaptive algorithms.

Least Mean Square (LMS) Algorithm

LMS algorithm is the simplest algorithm based on minimization of the MSE between the desired equalizer output and the actual equalizer output, as discussed earlier. Here the system error, the MSE and the optimal Wiener solution remain the same as given the adaptive equalization framework.

In practice, the minimization of the MSE is carried out recursively, and may be performed by use of the stochastic gradient algorithm. It is the simplest equalization algorithm and requires only 2N+1 operations per iteration. The filter weights are updated by the update equation. Letting the variable n denote the sequence of iteration, LMS is computed iteratively by

$$w_k\left(n+1\right) = w_k\left(n\right) + \mu e_k\left(n\right)x\left(n-k\right) \tag{7.16}$$

where the subscript k denotes the kth delay stage in the equalizer and μ is the step size which controls the convergence rate and stability of the algorithm.

The LMS equalizer maximizes the signal to distortion ratio at its output within the constraints of the equalizer filter length. If an input signal has a time dispersion characteristics that is greater than the propagation delay through the equalizer, then the equalizer will be unable to reduce distortion. The convergence rate of the LMS algorithm is slow due to the fact that there is only one parameter, the step size, that controls the adaptation rate. To prevent the adaptation from becoming unstable, the value of μ is chosen from

$$0 < \mu < 2\Bigg/\sum_{i=1}^{N}\lambda_i \tag{7.17}$$

where λ_i is the i-th eigenvalue of the covariance matrix R.

Normalized LMS (NLMS) Algorithm

In the LMS algorithm, the correction that is applied to $w_k(n)$ is proportional to the input sample $x(n-k)$. Therefore when $x(n-k)$ is large, the LMS algorithm experiences gradient noise amplification. With the normalization of the LMS step size by $\|\mathbf{x}(n)\|^2$ in the NLMS algorithm, this problem is eliminated. Only when $x(n-k)$ becomes close to zero, the denominator term $\|\mathbf{x}(n)\|^2$ in the NLMS equation becomes very small and the correction factor may diverge. So, a small positive number ε is added to the denominator term of the correction factor. Here, the step size is time varying and is expressed as

$$\mu(n) = \frac{\beta}{\|\mathbf{x}(n)\|^2 + \varepsilon}. \tag{7.18}$$

Therefore, the NLMS algorithm update equation takes the form of

$$w_k(n+1) = w_k(n) + \frac{\beta}{\|\mathbf{x}(n)\|^2 + \varepsilon} e_k(n)\, x(n-k). \tag{7.19}$$

7.3 Diversity

Diversity is a method used to develop information from several signals transmitted over independent fading paths. It exploits the random nature of radio propagation by finding independent signal paths for communication. It is a very simple concept where if one path undergoes a deep fade, another independent path may have a strong signal. As there is more than one path to select from, both the instantaneous and average SNRs at the receiver may be improved. Usually diversity decisions are made by receiver. Unlike equalization, diversity requires no training overhead as a training sequence is not required by transmitter. Note that if the distance between two receivers is a multiple of $\lambda/2$, there might occur a destructive interference between the two signals. Hence receivers in diversity technique are used in such a way that the signal received by one is independent of the other. Diversity can be of various forms, starting from space diversity to time diversity. We take up the types one by one in the sequel.

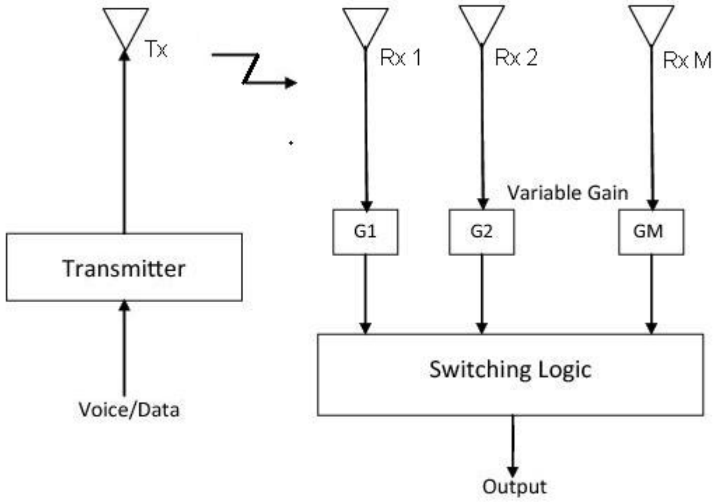

Figure 7.3: Receiver selection diversity, with M receivers.

7.3.1 Different Types of Diversity

Space Diversity

A method of transmission or reception, or both, in which the effects of fading are minimized by the simultaneous use of two or more physically separated antennas, ideally separated by one half or more wavelengths. Signals received from spatially separated antennas have uncorrelated envelopes.

Space diversity reception methods can be classified into four categories: selection, feedback or scanning, maximal ratio combining and equal gain combining.

(a) **Selection Diversity:**

The basic principle of this type of diversity is selecting the best signal among all the signals received from different branches at the receiving end. Selection Diversity is the simplest diversity technique. Figure 7.3 shows a block diagram of this method where 'M' demodulators are used to provide M diversity branches whose gains are adjusted to provide the same average SNR for each branch. The receiver branches having the highest instantaneous SNR is connected to the demodulator.

Let M independent Rayleigh fading channels are available at a receiver. Each channel is called a diversity branch and let each branch has the same average SNR. The signal to noise ratio is defined as

$$SNR = \Gamma = \frac{E_b}{N_0}\overline{\alpha^2} \tag{7.20}$$

where E_b is the average carrier energy, N_0 is the noise PSD, $\overline{\alpha}$ is a random variable used to represent amplitude values of the fading channel.

The instantaneous SNR(γ_i) is usually defined as γ_i = instantaneous signal power per branch/mean noise power per branch. For Rayleigh fading channels, α has a Rayleigh distribution and so α^2 and consequently γ_i have a chi-square distribution with two degrees of freedom. The probability density function for such a channel is

$$p\left(\gamma_i\right) = \frac{1}{\Gamma} e^{\frac{-\gamma_i}{\Gamma}}. \tag{7.21}$$

The probability that any single branch has an instantaneous SNR less than some defined threshold γ is

$$\Pr\left[\gamma_i \leq \gamma\right] = \int_0^\gamma p\left(\gamma_i\right) d\gamma_i = \int_0^\gamma \frac{1}{\Gamma} e^{\frac{-\gamma_i}{\Gamma}} d\gamma_i = 1 - e^{\frac{-\gamma}{\Gamma}} = P(\Gamma). \tag{7.22}$$

Similarly, the probability that all M independent diversity branches receive signals which are simultaneously less than some specific SNR threshold γ is

$$\Pr\left[\gamma_1, \gamma_2, \ldots, \gamma_M \leq \gamma\right] = \left(1 - e^{\frac{-\gamma}{\Gamma}}\right)^M = P_M\left(\gamma\right) \tag{7.23}$$

where $P_M\left(\gamma\right)$ is the probability of all branches failing to achieve an instantaneous SNR $= \gamma$. Quite clearly, $P_M(\Gamma) < P(\Gamma)$. If a single branch achieves SNR $> \gamma$, then the probability that SNR $> \gamma$ for one or more branches is given by

$$\Pr\left[\gamma_i > \gamma\right] = 1 - P_M\left(\gamma\right) = 1 - \left(1 - e^{\frac{-\gamma}{\Gamma}}\right)^M \tag{7.24}$$

which is more than the required SNR for a single branch receiver. This expression shows the advantage when a selection diversity is used.

To determine of average signal to noise ratio, we first find out the pdf of γ as

$$p_M\left(\gamma\right) = \frac{d}{d\gamma} P_M\left(\gamma\right) = \frac{M}{\Gamma}\left(1 - e^{-\gamma/\Gamma}\right)^{M-1} e^{-\gamma/\Gamma}. \tag{7.25}$$

The average SNR, $\bar{\gamma}$, can be then expressed as

$$\bar{\gamma} = \int_0^\infty \gamma p_M\left(\gamma\right) d\gamma = \Gamma \int_0^\infty Mx\left(1 - e^{-x}\right)^{M-1} e^{-x} dx \tag{7.26}$$

where $x = \gamma/\Gamma$ and Γ is the average SNR for a single branch, when no diversity is used.

This equation shows an average improvement in the link margin without requiring extra transmitter power or complex circuitry, and it is easy to implement as it needed a monitoring station and an antenna switch at the receiver. It is not an optimal diversity technique as it doesn't use all the possible branches simultaneously.

(b) **Feedback or Scanning Diversity:**

Scanning all the signals in a fixed sequence until the one with SNR more than a predetermined threshold is identified. Feedback or scanning diversity is very similar to selection diversity except that instead of always using the best of N signals, the N signals are scanned in a fixed sequence until one is found to be above a predetermined threshold. This signal is then received until it falls below threshold and the scanning process is again initiated. The resulting fading statistics are somewhat inferior, but the advantage is that it is very simple to implement(only one receiver is required).

(c) **Maximal Ratio Combining:**

Signals from all of the m branches are weighted according to their individual signal voltage to noise power ratios and then summed. Individual signals must be cophased before being summed, which generally requires an individual receiver and phasing circuit for each antenna element. Produces an output SNR equal to the sum of all individual SNR. Advantage of producing an output with an acceptable SNR even when none of the individual signals are themselves acceptable. Modern DSP techniques and digital receivers are now making this optimal form, as it gives the best statistical reduction of fading of any known linear diversity combiner. In terms of voltage signal,

$$r_m = \sum_{i=1}^{m} G_i r_i \tag{7.27}$$

where G_i is the gain and r_i is the voltage signal from each branch.

(d) **Equal Gain Combining:**

In some cases it is not convenient to provide for the variable weighting capability required for true maximal ratio combining. In such cases, the branch weights are all set unity, but the signals from each branch are co-phased to provide equal gain combining diversity. It allows the receiver to exploit signals that are simultaneously received on each branch. Performance of this method is marginally inferior to maximal ratio combining and superior to Selection diversity. Assuming all the G_i to be

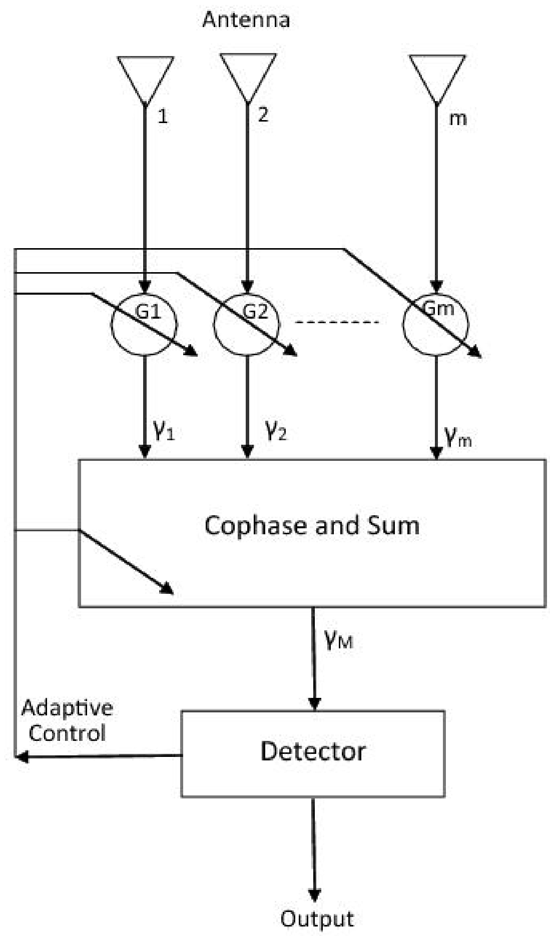

Figure 7.4: Maximal ratio combining technique.

unity, here,

$$r_m = \sum_{i=1}^{m} r_i. \tag{7.28}$$

Polarization Diversity

Polarization Diversity relies on the decorrelation of the two receive ports to achieve diversity gain. The two receiver ports must remain cross-polarized. Polarization Diversity at a base station does not require antenna spacing. Polarization diversity combines pairs of antennas with orthogonal polarizations (i.e. horizontal/vertical, $\pm$ slant 45^o, Left-hand/Right-hand CP etc). Reflected signals can undergo polarization changes depending on the channel. Pairing two complementary polarizations, this scheme can immunize a system from polarization mismatches that would otherwise cause signal fade. Polarization diversity has prove valuable at radio and mobile communication base stations since it is less susceptible to the near random orientations of transmitting antennas.

Frequency Diversity

In Frequency Diversity, the same information signal is transmitted and received simultaneously on two or more independent fading carrier frequencies. Rationale behind this technique is that frequencies separated by more than the coherence bandwidth of the channel will be uncorrelated and will thus not experience the same fades. The probability of simultaneous fading will be the product of the individual fading probabilities. This method is employed in microwave LoS links which carry several channels in a frequency division multiplex mode (FDM). Main disadvantage is that it requires spare bandwidth also as many receivers as there are channels used for the frequency diversity.

Time Diversity

In time diversity, the signal representing the same information are sent over the same channel at different times. Time diversity repeatedly transmits information at time spacings that exceeds the coherence time of the channel. Multiple repetition of the signal will be received with independent fading conditions, thereby providing for diversity. A modern implementation of time diversity involves the use of RAKE receiver for spread spectrum CDMA, where the multipath channel provides redundancy in the transmitted message. Disadvantage is that it requires spare bandwidth also as many receivers as there are channels used for the frequency diversity. Two important types of time diversity application is discussed below.

Application 1: RAKE Receiver

In CDMA spread spectrum systems, CDMA spreading codes are designed to provide very low correlation between successive chips, propagation delay spread in the radio channel provides multiple version of the transmitted signal at the receiver. Delaying multipath components by more than a chip duration, will appear like uncorrelated noise at a CDMA receiver. CDMA receiver may combine the time delayed versions of the original signal to improve the signal to noise ratio at the receiver. RAKE

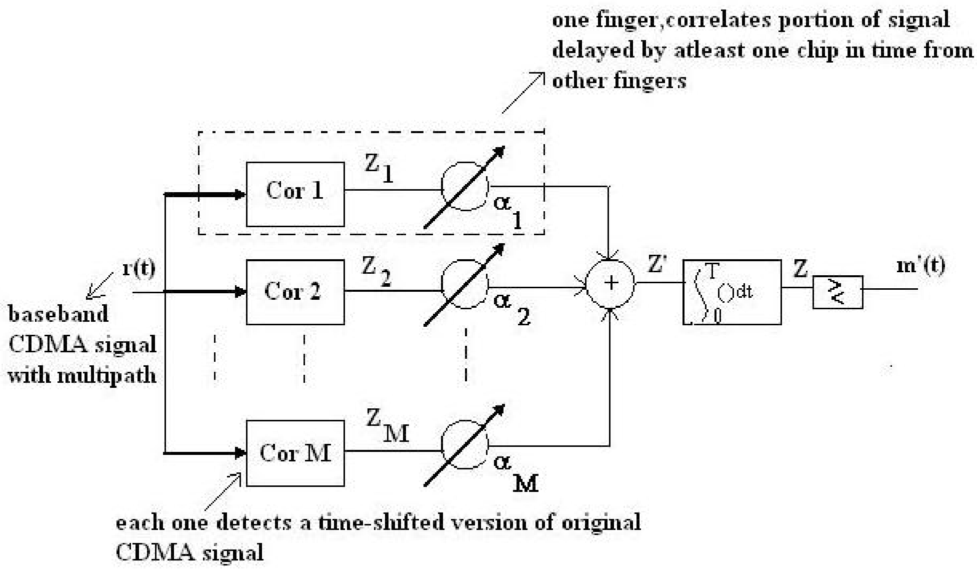

Figure 7.5: RAKE receiver.

receiver collect the time shifted versions of the original signal by providing a separate correlation receiver for M strongest multipath components. Outputs of each correlator are weighted to provide a better estimate of the transmitted signal than provided by a single component. Demodulation and bit decisions are based on the weighted output of the correlators. Schematic of a RAKE receiver is shown in Figure 7.5.

Application 2: Interleaver

In the encoded data bits, some source bits are more important than others, and must be protected from errors. Many speech coder produce several important bits in succession. Interleaver spread these bit out in time so that if there is a deep fade or noise burst, the important bits from a block of source data are not corrupted at the same time. Spreading source bits over time, it becomes possible to make use of error control coding. Interleaver can be of two forms, a block structure or a convolutional structure.

A block interleaver formats the encoded data into a rectangular array of m rows and n columns, and interleaves nm bits at a time. Each row contains a word of source data having n bits. an interleaver of degree m consists of m rows. source bits are placed into the interleaver by sequentially increasing the row number for each successive bit, and forming the columns. The interleaved source data is then read out row-wise and transmitted over the channel. This has the effect of separating the original source bits by m bit periods. At the receiver, de-interleaver stores the received data by sequentially increasing the row number of each successive bit, and then clocks out the data row-wise, one word at a time. Convolutional interleavers are ideally suited for use with convolutional codes.

7.4 Channel Coding

In channel coding, redundant data bits are added in the transmitted message so that if an instantaneous fade occurs in the channel, the data may still be recovered at the receiver without the request of retransmission. A channel coder maps the transmitted message into another specific code sequence containing more bits. Coded message is then modulated for transmission in the wireless channel. Channel Coding is used by the receiver to detect or correct errors introduced by the channel. Codes that used to detect errors, are error detection codes. Error correction codes can detect and correct errors.

7.4.1 Shannon's Channel Capacity Theorem

In 1948, Shannon showed that by proper encoding of the information, errors induced by a noise channel can be reduced to any desired level without sacrificing the rate of information transfer. Shannon's channel capacity formula is applicable to the AWGN channel and is given by:

$$C = B\log_2\left(1+\frac{S}{N}\right) = B\log_2\left(1+\frac{P}{N_0 B}\right) = B\log_2\left(1+\frac{E_b R_b}{N_0 B}\right) \tag{7.29}$$

where C is the channel capacity (bit/s), B is the channel bandwidth (Hz), P is the received signal power (W), N_0 is the single sided noise power density (W/Hz), E_b is the average bit energy and R_b is transmission bit rate.

Equation (7.29) can be normalized by the bandwidth B and is given as

$$\frac{C}{B} = \log_2\left(1+\frac{E_b R_b}{N_0 B}\right) \tag{7.30}$$

and the ratio C/B is denoted as bandwidth efficiency. Introduction of redundant bits increases the transmission bit rate and hence it increases the bandwidth requirement, which reduces the bandwidth efficiency of the link in high SNR conditions, but provides excellent BER performance at low SNR values. This leads to the following two inferences.

Corollary 1: While dealing within maximum channel capacity, introduction of redundant bits increase the transmitter rate and hence bandwidth requirement also increases, while decreasing the bandwidth efficiency, but it also decreases the BER.

Corollary 2: If data redundancy is not introduced in a wideband noisy environment, error free performance in not possible (for example, CDMA communication in 3G mobile phones).

A channel coder operates on digital message (or source) data by encoding the source information into a code sequence for transmission through the channel. The error correction and detection codes are classified into three groups based on their structure.

1. Block Code
2. Convolution Code
3. Concatenated Code.

7.4.2 Block Codes

Block codes are *forward error correction* (FEC) codes that enable a limited number of errors to be detected and corrected without retransmission. Block codes can be used to improve the performance of a communications system when other means of improvement (such as increasing transmitter power or using a more sophisticated demodulator) are impractical.

In block codes, parity bits are added to blocks of message bits to make codewords or code blocks. In a block encoder, k information bits are encoded into n code bits. A total of $n-k$ redundant bits are added to the k information bits for the purpose of detecting and correcting errors. The block code is referred to as an (n, k) code, and the rate of the code is defined as $R_c = k/n$ and is equal to the rate of information divided by the raw channel rate.

Parameters in Block Code

(a) Code Rate (R_c): As defined above, $R_c = k/n$.

(b) Code Distance (d): Distance between two codewords is the number of elements in which two codewords C_i and C_j differs denoted by $d(C_i, C_j)$. If the code used is binary, the distance is known as 'Hamming distance'. For example d(10110, 11011) is 3. If the code 'C' consists of the set of codewords, then the minimum distance of the code is given by $d_{\text{min}} = min\{d(C_i, C_j)\}$.

(c) Code Weight (w): Weight of a codeword is given by the number of nonzero elements in the codeword. For a binary code, the weight is basically the number of 1s in the codeword. For example weight of a code 101101 is 4.

Ex 1: The block code C = 00000, 10100, 11110, 11001 can be used to represent two bit binary numbers as:

- 00 – 00000
- 01 – 10100
- 10 – 11110
- 11 – 11001

Here number of codewords is 4, k = 2, and n = 5.
To encode a bit stream 1001010011

- First step is to break the sequence in groups of two bits, i.e., 10 01 01 00 11
- Next step is to replace each block by its corresponding codeword, i.e.,

  11110 10100 10100 00000 11001

Quite clearly, here, $d_{\text{min}} = min\{d(C_i, C_j)\} = 2$.

Properties of Block Codes

(a) Linearity: Suppose C_i and C_j are two code words in an (n, k) block code. Let α_1 and α_2 be any two elements selected from the alphabet. Then the code is said to be linear if and only if $\alpha_1 C_1 + \alpha_2 C_2$ is also a code word. A linear code must contain the all-zero code word.

(b) Systematic: A systematic code is one in which the parity bits are appended to the end of the information bits. For an (n, k) code, the first k bits are identical to the information bits, and the remaining $n - k$ bits of each code word are linear combinations of the k information bits.

(c) Cyclic: Cyclic codes are a subset of the class of linear codes which satisfy the following cyclic shift property: If $C = [C_{n-1}, C_{n-2}, ..., C_0]$ is a code word of a cyclic code, then $[C_{n-2}, C_{n-3}, ..., C_0, C_{n-1}]$, obtained by a cyclic shift of the elements of C, is also a code word. That is, all cyclic shifts of C are code words.

In this context, it is important to know about **Finite Field or Galois Field**. Let F be a finite set of elements on which two binary operations – addition (+) and multiplication (.) are defined. The set F together with the two binary operations is called a *field* if the following conditions are satisfied:

1. F is a commutative group under addition.
2. The set of nonzero elements in F is a commutative group under multiplication.
3. Multiplication is distributive over addition; that is, for any three elements a, b, and c in F, $a(b + c) = ab + ac$
4. Identity elements 0 and 1 must exist in F satisfying $a + 0 = a$ and $a.1 = a$.
5. For any a in F, there exists an additive inverse $(-a)$ such that $a + (-a) = 0$.
6. For any a in F, there exists an multiplicative inverse a^{-1} such that $a.a^{-1} = 1$.

Depending upon the number of elements in it, a field is called either a finite or an infinite field. The examples of infinite field include Q (set of all rational numbers), R (set of all real numbers), C (set of all complex numbers) etc. A field with a finite number of elements (say q) is called a 'Galois Field' and is denoted by GF(q). A finite field entity p(x), called a polynomial, is introduced to map all symbols (with several bits) to the element of the finite field. A polynomial is a mathematical expression

$$p(x) = p_0 + p_1 x + ... + p_m x_m \tag{7.31}$$

where the symbol x is called the indeterminate and the coefficients $p_0, p_1, ..., p_m$ are the elements of GF(q). The coefficient p_m is called the leading coefficient. If p_m is not equal to zero, then m is called the degree of the polynomial, denoted as deg p(x). A polynomial is called monic if its leading coefficient is unity. The division algorithm states that for every pair of polynomials a(x) and b(x) in F(x), there exists a unique pair of polynomials q(x), the quotient, and r(x), the remainder, such that $a(x) = q(x)b(x) + r(x)$, where deg r(x)¡deg b(x). A polynomial p(x) in F(x) is said to be reducible if p(x)=a(x)b(x), otherwise it is called irreducible. A monic irreducible polynomial of degree at least one is called a prime polynomial.

An irreducible polynomial $p(x)$ of degree 'm' is said to be primitive if the smallest integer 'n' for which $p(x)$ divides x^n+1 is $n = 2^m-1$. A typical primitive polynomial is given by $p(x) = x^m + x + 1$.

A specific type of code which obeys both the cyclic property as well as polynomial operation is cyclic codes. Cyclic codes are a subset of the class of linear codes which satisfy the cyclic property. These codes possess a considerable amount of structure which can be exploited. A cyclic code can be generated by using a generator polynomial g(p) of degree (n-k). The generator polynomial of an (n,k) cyclic code is a factor of $p^n + 1$ and has the form

$$g\left(p\right) = p^{n-k} + g_{n-k-1}p^{n-k-1} + \cdots + g_1 p + 1. \tag{7.32}$$

A message polynomial x(p) can also be defined as

$$x\left(p\right) = x_{k-1}p^{k-1} + \cdots + x_1 p + x_0 \tag{7.33}$$

where $(x_{k-1}, \ldots, x_0)$ represents the k information bits. The resultant codeword c(p) can be written as

$$c\left(p\right) = x\left(p\right) g\left(p\right) \tag{7.34}$$

where c(p) is a polynomial of degree less than n. We would see an application of such codes in Reed-Solomon codes.

Examples of Block Codes

(a) Single Parity Check Code: In single parity check codes (example: ASCII code), an overall single parity check bit is appended to 'k' information bits. Let the information bit word be: $(b_1, b_2, ..., b_k)$, then parity check bit: $p = b_1 + b_2 + ... + b_k$ modulo 2 is appended at the (k+1)th position, making the overall codeword: $C = (b_1, b_2, ..., b_k, p)$. The parity bit may follow an even parity or an odd parity pattern. All error patterns that change an odd number of bits are detectable, and all even numbered error patterns are not detectable. However, such codes can only detect the error, it cannot correct the error.

Ex. 2: Consider a (8,7) ASCII code with information codeword (0, 1, 0, 1, 1, 0, 0) and encoded with overall even parity pattern. Thus the overall codeword is (0, 1, 0, 1, 1, 0, 0, 1) where the last bit is the parity bit. If there is a single error in bit 3: (0, 1, **1**, 1, 1, 0, 0, 1), then it can be easily checked by the receiver that now there are odd number of 1's in the codeword and hence there is an error. On the other hand, if there are two errors, say, errors in bit 3 and 5: (0, 1, **1**, 1, **0**, 0, 0, 1), then error will not be detected.

After decoding a received codeword, let p_c be the probability that the decoder gives correct codeword C, p_e is the probability that the decoder gives incorrect codeword $C' \neq C$, and p_f is the probability that the decoder fails to give a codeword. In this case, we can write $p_c + p_e + p_f = 1$.

If in an n-bit codeword, there are j errors and p is the bit error probability, then the probability of obtaining j errors in this codeword is $P_j = {}^nC_j p^j \left(1-p\right)^{n-j}$. Using this formula, for any $(n, n-1)$ single parity check block code, we get

- $p_c = P_0$,
- $p_e = P_2 + P_4 + ... + P_n'$ ($n' = n$ if n is even, otherwise $n' = n - 1$),
- $p_f = P_1 + P_3 + ... + P_n'$ ($n' = n - 1$ if n is even, otherwise $n' = n$).

As an example, for a (5,4) single parity check block code, $p_c = P_0$, $p_e = P_2 + P_4$, and $p_f = P_1 + P_3 + P_5$.

(b) Product Codes: Product codes are a class of linear block codes which provide error detection capability using product of two block codes. Consider that nine information bits $(1, 0, 1, 0, 0, 1, 1, 1, 0)$ are to be transmitted. These 9 bits can be divided into groups of three information bits and (4,3) single parity check codeword can be formed with even parity. After forming three codewords, those can be appended with a vertical parity bit which will form the fourth codeword. Thus the following codewords are transmitted:

$$C1 = [1 \quad 0 \quad 1 \quad 0]$$
$$C2 = [0 \quad 0 \quad 1 \quad 1]$$
$$C3 = [1 \quad 1 \quad 0 \quad 0]$$
$$C4 = [0 \quad 1 \quad 0 \quad 1].$$

Now if an error occurs in the second bit of the second codeword, the received codewords at the receiver would then be

$$C1 = [1 \quad 0 \quad 1 \quad 0]$$

$$C2 = [0 \quad 1 \quad 1 \quad 1] \leftarrow$$

$$C3 = [1 \quad 1 \quad 0 \quad 0]$$

$$C4 = [0 \quad 1 \quad 0 \quad 1]$$

$$\uparrow$$

and these would indicate the corresponding row and column position of the erroneous bit with vertical and horizontal parity check. Thus the bit can be corrected. Here we get a horizontal (4, 3) codeword and a vertical (4, 3) codeword and concatenating them we get a (16, 9) product code. In general, a product code can be formed as (n_1, k_1) & $(n_2, k_2) \rightarrow (n_1 n_2, k_1 k_2)$.

(c) Repetition Codes: In a (n,1) repetition code each information bit is repeated n times (n should be odd) and transmitted. At the receiver, the majority decoding principle is used to obtain the information bit. Accordingly, if in a group of n received bit, 1 occurs a higher number of times than 0, the information bit is decoded as 1. Such majority scheme works properly only if the noise affects less than n/2 number of bits.

Ex 3: Consider a (3,1) binary repetition code.

- For input bit 0, the codeword is (0 0 0) and for input bit 1, the codeword is (1 1 1).
- If the received codeword is (0 0 0), i.e. no error, it is decoded as 0.
- Similarly, if the received codeword is (1 1 1), i.e. no error, it is decoded as 1.
- If the received codeword is (0 0 1) or (0 1 0) or (1 0 0), then error is detected and it is decoded as 0 with majority decoding principle.
- If the received codeword is (0 1 1) or (1 1 0) or (1 0 1), once again error is detected and it is decoded as 1 with majority decoding principle.

For such a (3,1) repetition code, $p_c = P_0 + P_1$, $p_e = P_2 + P_3$, and $p_f = 0$.

(d) Hamming Codes: A binary Hamming code has the property that

$$(n, k) = (2^m - 1, 2^m - 1 - m) \tag{7.35}$$

where k is the number of information bits used to form a n bit codeword, and m is any positive integer. The number of parity symbols are $n - k = m$. Thus, a codeword is represented by $C = [i_1, ...i_n, p_1, ..., p_{n-k}]$. This is quite a useful code in communication which is illustrated via the following example.

Ex 4: Consider a (7, 4) Hamming code. With three parity bits we can correct exactly 1 error. The parity bits may follow such a modulo 2 arithmetic:

$$\begin{aligned} p_1 &= i_1 + i_2 + i_3, \\ p_2 &= i_2 + i_3 + i_4, \\ p_3 &= i_1 + i_3 + i_4, \end{aligned}$$

which is same as,

$$\begin{aligned} p_1 + i_1 + i_2 + i_3 &= 0 \\ p_2 + i_2 + i_3 + i_4 &= 0 \\ p_3 + i_1 + i_3 + i_4 &= 0. \end{aligned}$$

The transmitted codeword is then $C = [i_1, i_2, ..., i_4, p_1, p_2, p_3]$.
Syndrome Decoding: For this Hamming code, let the received codeword be $V = [v_1, v_2, ..., v_4, v_5, v_6, v_7]$. We define a syndrome vector S as

$$\begin{aligned} S &= [S_1\ S_2\ S_3] \\ S1 &= v_1 + v_2 + v_3 + v_5 \\ S2 &= v_2 + v_3 + v_4 + v_6 \\ S3 &= v_1 + v_2 + v4 + v7 \end{aligned}$$

It is obvious that in case of no error, the syndrome vector is equal to zero. Corresponding to this syndrome vector, there is an error vector e which can be obtained from a syndrome table and finally the required codeword is taken as $C = V + e$. In a nutshell, to obtain the required codeword, we perform the following steps:
1. Calculate S from decoder input V.
2. From syndrome table, obtain e corresponding to S.
3. The required codeword is then $C = V + e$.

A few cases are given below to illustrate the syndrome decoding.
1. Let C = [0 1 1 1 0 1 0] and V = [0 1 1 1 0 1 0]. This implies S = [0 0 0], and it corresponds to e = [0 0 0 0 0 0 0]. Thus, C = V + e = [0 1 1 1 0 1 0].

2. Let C = [1 1 0 0 0 1 0] and V = [1 1 0 1 0 1 0]. This means S = [0 1 1], from which we get e = [0 0 0 1 0 0 0] which means a single bit error is there in the received bit v_4. This will be corrected by performing the operation C = V + e.
3. Another interesting case is, let C = [0 1 0 1 1 0 0] and V = [0 0 1 1 1 0 1] (two errors at second and third bits). This makes S = [0 0 0] and as a result, e = [0 0 0 0 0 0 0]. However, $C \neq V$, and $C = V + e$ implies the double error cannot be corrected. Therefore a (7,4) Hamming code can correct only single bit error.

(e) Golay Codes: Golay codes are linear binary (23,12) codes with a minimum distance of seven and a error correction capability of three bits. This is a special, one of a kind code in that this is the only nontrivial example of a perfect code. Every codeword lies within distance three of any codeword, thus making maximum likelihood decoding possible.

(f) BCH Codes: BCH code is one of the most powerful known class of linear cyclic block codes, known for their multiple error correcting ability, and the ease of encoding and decoding. It's block length is $n = 2^m - 1$ for $m \geq 3$ and number of errors that they can correct is bounded by $t < (2^m - 1)/2$. Binary BCH codes can be generalized to create classes of non binary codes which use m bits per code symbol.

(g) Reed Solomon (RS) Codes: Reed-Solomon code is an important subset of the BCH codes with a wide range of applications in digital communication and data storage. Typical application areas are storage devices (CD, DVD etc.), wireless communications, digital TV, high speed modems. It's coding system is based on groups of bits, such as bytes, rather than individual 0 and 1. This feature makes it particularly good at dealing with burst of errors: six consecutive bit errors. Block length of these codes is $n = 2^m - 1$, and can be extended to 2^m or $2^m + 1$. Number of parity symbols that must be used to correct e errors is $n - k = 2e$. Minimum distance $d_{min} = 2e + 1$, and it achieves the largest possible d_{min} of any linear code.

For US-CDPD, the RS code is used with $m = 6$. So each of the 64 field elements is represented by a 6 bit symbol. For this case, we get the primitive polynomial as $p(x) = x^6 + x + 1$. Equating $p(x)$ to 0 implies $x^6 = x + 1$.
The 6 bit representation of the finite field elements is given in Table 7.1. The table elements continue up to α^{62}. However, to follow linearity property there should be

Table 7.1: Finite field elements for US-CDPD

	α^5	α^4	α^3	α^2	α^1	α^0
1	0	0	0	0	0	1
α^1	0	0	0	1	0	0
α^2	0	0	1	0	0	0
.	.	.	.	.	.	.
.	.	.	.	.	.	.
$\alpha^6 = \alpha + 1$	0	0	0	0	1	1
.	.	.	.	.	.	.
.	.	.	.	.	.	.

a zero codeword, hence α^{63} is assigned zero.

The encoding part of the RS polynomial is done as follows:
Information polynomial: $d(x) = C_{n-1}x^{n-1} + C_{n-2}x^{n-2} + ... + C_{2t}x^{2t}$,
Parity polynomial: $p(x) = C_{2t-1}x^{2t-1} + ... + C_0$,
Codeword polynomial: $c(x) = d(x) + p(x)$.
Since generating an information polynomial is difficult, so a generating polynomial is used instead. Information polynomial is then the multiple of generating polynomial. This process is given below.
Since this kind of codes are cyclic codes, we take a generating polynomial $g(x)$ such that $d(x) = g(x)q(x) + r(x)$ where $q(x)$ is the quotient polynomial and $r(x)$ is the remainder polynomial. The codeword polynomial would then be given as: $c(x) = g(x)q(x) + r(x) = p(x)$. If we assign a parity polynomial $p(x) = r(x)$, then the codeword polynomial $c(x) = g(x)p(x)$ and the entire process becomes easier.

On the decoder side one has to find a specific $r(x) = p(x)$ or vice-versa, but due to its complexity, it is mainly done using syndrome calculation. The details of such a syndrome calculation can be found in [1].

7.4.3 Convolutional Codes

A continuous sequence of information bits is mapped into a continuous sequence of encoder output bits. A convolutional code is generated by passing the information sequence through a finite state shift register. Shift register contains 'N' k-bit stages

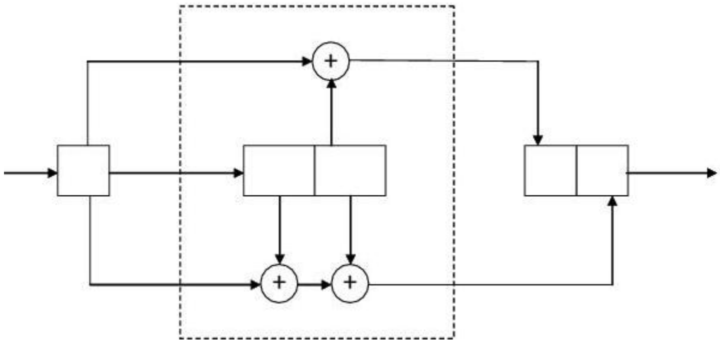

Figure 7.6: A convolutional encoder with n=2 and k=1.

and m linear algebraic function generators based on the generator polynomials. Input data is shifted into and along the shift register, k-bits at a time. Number of output bits for each k-bit user input data sequence is n bits, so the code rate $R_c = k/n$. The shift register of the encoder is initialized to all-zero-state before

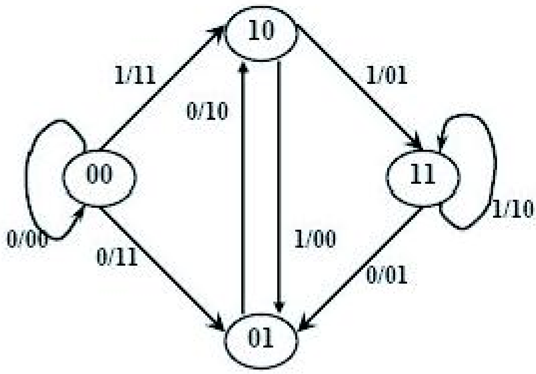

Figure 7.7: State diagram representation of a convolutional encoder.

encoding operation starts. It is easy to verify that encoded sequence is 00 11 10 00 01 ... for an input message sequence of 01011 .... Convolution codes may be represented in various ways as given below.

State Diagram:

Since the output of the encoder is determined by the input and the current state of the encoder, a state diagram can be used to represent the encoding process. The state diagram is simply a graph of the possible states of the encoder and the possible transitions from one state to another. The path information between the states, denoted as b/c_1c_2, represents input information bit 'b' and the corresponding

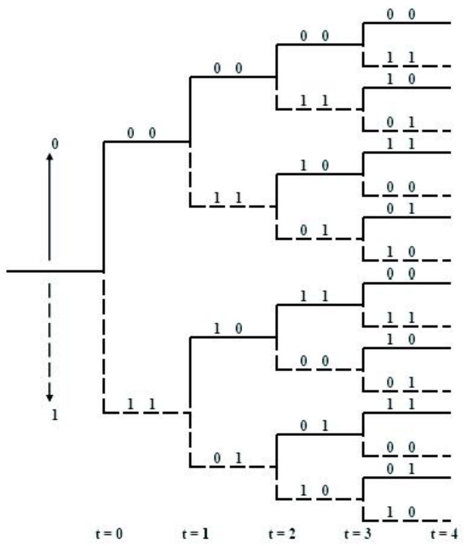

Figure 7.8: Tree diagram representation of a convolutional encoder.

output bits (c_1c_2). Again, it is not difficult to verify from the state diagram that an input information sequence b = (1011) generates an encoded sequence c = (11, 10, 00, 01).

Tree Diagram:

The tree diagram shows the structure of the encoder in the form of a tree with the branches representing the various states and the outputs of the coder. The encoded bits are labeled on the branches of the tree. Given an input sequence, the encoded sequence can be directly read from the tree. As an example, an input sequence (1011) results in the encoded sequence (11, 10, 00, 01).

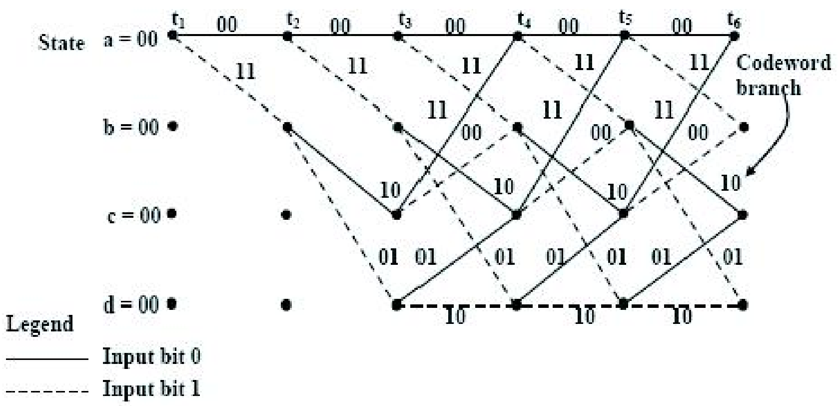

Figure 7.9: Trellis diagram of a convolutional encoder.

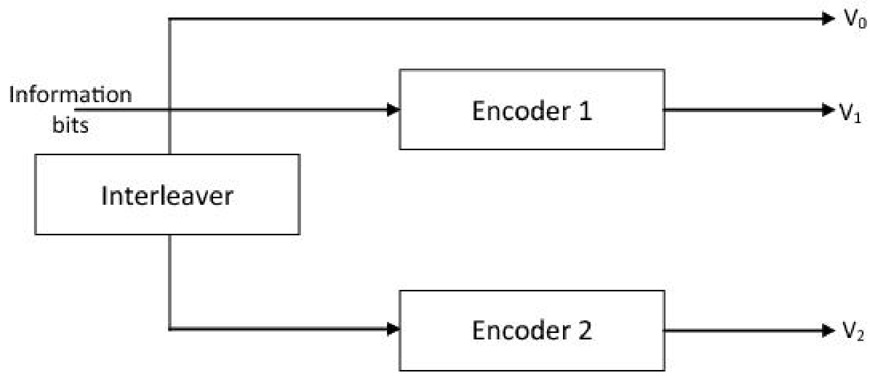

Figure 7.10: Block diagram of a turbo encoder.

Trellis Diagram:

Tree reveals that the structure repeats itself once the number of stages is greater than the constraint length. It is observed that all branches emanating from two nodes having the same state are identical in the sense that they generate identical output sequences. This means that the two nodes having the same label can be merged. By doing this throughout the tree diagram, we obtain another diagram called a Trellis Diagram which is more compact representation.

7.4.4 Concatenated Codes

Concatenated codes are basically concatenation of block and convolutional codes. It can be of two types: serial and parallel codes. Below, we discuss a popular parallel concatenated code, namely, turbo code.

Turbo Codes: A turbo encoder is built using two identical convolutional codes of special type with parallel concatenation. An individual encoder is termed a component encoder. An interleaver separates the two component encoders. The interleaver is a device that permutes the data sequence in some predetermined manner. Only one of the systematic outputs from the two component encoders is used to form a codeword, as the systematic output from the other component encoder is only a permuted version of the chosen systematic output. Figure 7.10 shows the block diagram of a turbo encoder using two identical encoders. The first encoder outputs the systematic V_0 and recursive convolutional V_1 sequences while the second encoder discards its systematic sequence and only outputs the recursive convolutional V_2 sequence. Depending on the number of input bits to a component encoder it may be binary or m-binary encoder. Encoders are also categorized as systematic or non-systematic. If the component encoders are not identical then it is called an asymmetric turbo code.

7.5 Conclusion

Although a lot of advanced powerful techniques for mitigating the fading effects such as space diversity in MIMO systems, space-time block coding scheme, MIMO equalization, BLAST architectures etc. have taken place in modern wireless communication, nevertheless, the discussed topics in this chapter are the basic building blocks for all such techniques and that stems the necessity for all these discussions. The effectiveness of the discussed topics would be more clear in the next chapter in the context of different multiple access techniques.

7.6 References

1. T. S. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed. Singapore: Pearson Education, Inc., 2002.

2. J. R. Treichler, C. R. Johnson (Jr.) and M. G. Larimore, *Theory and Design of Adaptive Filters.* New Delhi: PHI, 2002.

3. S. Gravano, *Introduction to Error Control Codes.* NY: Oxford University Press, 2001.