CHAPTER 6

MULTILEVEL CODED MODULATION

6.0 INTRODUCTION

The power efficiency of a digital communications system can be increased through the introduction of channel coding. The price is either the increase in bandwidth needed to accommodate the increased symbol rate or a drop in the useful data rate. Gottfried Ungerboeck's important paper in 1982 showed that this problem could be solved by carefully combining the process of channel coding and M-ary modulation; the concept of Coded Modulation. Coding gain can then be achieved without the need to expand bandwidth. This chapter introduces multilevel coded modulation (MLCM) and focuses on one particular technique of bandwidth efficient coding- trellis Coded Modulation. MLCM comprises multiple component codes mapped in parallel on to an M-ary constellation and separate decoding of the component codes. The performance of MLCM will be improved by the use of turbo codes as component codes, to form Turbo Trellis Coded Modulation. The optimization is achieved by using the codes reported in the previous chapters and the choice of generator polynomials [23]. The pseudo random interleavers proposed have been used in all the trellis coded modulation systems to improve the performance and to maintain a low error floor.

6.1 TRELLIS-CODED MODULATION

In a power-limited environment, the desired system performance should be achieved with the smallest possible transmitted power. The use of error-correcting codes can increase power efficiency by adding extra bits to the transmitted symbol sequence. This procedure requires the modulator to operate at a higher data rate, which requires a wider bandwidth. In a bandwidth-limited environment, the use of higher order modulation schemes can increase efficiency in frequency utilization [24]. In this case, a large signal power would be required to maintain the same system bit-error-rate (BER). In order to achieve improved reliability of a digital transmission system without increasing transmitted power or required bandwidth, both coding and modulation are considered in TCM technology; therefore, TCM is a scheme combining error-correcting coding with modulation.

TCM is used for data communication with the purpose of gaining noise immunity over uncoded transmission without changing the data rate. The use of TCM also improves
system reliability without increasing transmitting power and required channel bandwidth. Since channel bandwidth is a function of the signal-to-noise ratio (SNR), larger signal power would be necessary to maintain the same signal separation and the same error probability if more channel bandwidth. Quadrature Amplitude modulation (QAM) and Quaternary Phase Shift Keying (QPSK) are used in TCM to increase data transmission rate [25], [26]. It seems that the Trellis code statement violates the basic power, bandwidth and error probability trade-off principle. Actually this is not true; the reason for this will be explained below. It introduces the concepts of sequence coding.

Trellis coding introduces dependency between every successive transmitting data symbol. The optimum 2-dimensional modulation utilizes the dependency between in-phase and quadrature symbols and the 4-dimensional modulation employs the dependency between symbols of two successive time intervals. Trellis codes and multidimensional modulation are designed to maximize the Euclidean distance between possible sequences of transmitted symbols. Euclidean distance is a straight-line distance between two points in signal constellation. In N dimensions, the Euclidean distance between two points p and q is:

$$\sqrt{\sum_{i=1}^{N} (p_i-q_i)^2}$$  \hspace{1cm} (6.1)

where $p_i$ and $q_i$ are the coordinate of p and q in the dimension i. The minimum Euclidean distance (i.e., the distance between the closest possible sequences) of transmitted symbols in signal space determines the system performance as:

$$P_e \sim e^{d_{\text{min}}/2\sigma^2}$$  \hspace{1cm} (6.2)

where $P_e$ is the message error probability, $d_{\text{min}}$ is the minimum Euclidean distance between signal sequences and $\sigma$ is the noise power. Equation (6.2) indicates that if $d_{\text{min}}$ increases, $P_e$ will decrease. This is one of the reasons why TCM technique does not violate the basic trade-off principle between power, bandwidth, and error probability.

Since Ungerboeck invented TCM in 1976 and had his papers published in the 1980s, numerous researchers have been working on TCM applications in numerous areas: voice band modems, satellite communications, wireless communications trials, digital subscriber
loop, HDTV (high definition television), broadcast channels, CATV (community antenna television) and DBS (direct broadcast satellite) in the 1980's and 1990's. Many innovations in TCM technology have been introduced, such as multidimensional TCM (1984-1985), rotationally invariant TCM with M-PSK (1988), TCM with built-in time diversity (1988-1990), TCM with Tomlinson Precoder (1990-1991), TCM with unequal error protection (1990), multilevel coding with TCM (1992-1993), and concatenated coding with TCM (1993-present).

Trellis coded modulation, or TCM, was invented as a method to improve the reliability of a digital transmission system without bandwidth expansion or reduction of data rate. Normal channel codes such as block and convolutional codes improve the performances of the communication system by expanding the bandwidth. The Euclidean distance between the transmitted coded waveforms would be increased by the use of coding, but at the price of increasing the bandwidth [27]. Trellis-coded modulation is a coded modulation scheme that can increase the noise immunity and simultaneously do not increase the bandwidth.

The TCM coding process utilizes signal mapping combining error-correcting coding with modulation. The mapping by the set partitioning technique provides a combination of digital signals used in the modulation. This technique increases the minimum Euclidean distance between the pairs of coded signals; hence the loss from the expansion of the signal set is easily overcome and a significant coding gain is achieved with ECC. This is the reason why TCM technique does not violate the basic trade-off principle between power, bandwidth, and error probability [28]. Therefore, in a bandwidth-limited communication system, without bandwidth enlargement, the redundancy bits are introduced into the signal to achieve good performance of coding gain through the TCM codes. The widely used ECC code in the TCM encoder is the convolutional code.

The innovative aspect of TCM is the concept that convolutional encoding and modulation should not be treated as separated entities, but rather as a unique operation. As a conclusion, the received signal is processed by combining the demodulation and decoding in a single step, instead of being first demodulated and then decoded. The consequence of this is that the parameter governing the performance of the transmission system over the AWGN channel is not the free Hamming distance of the convolutional code but rather the free Euclidean distance between the transmitted signal sequences. Thus, the optimization of the TCM design will be based on the Euclidean distances rather than the Hamming distances, and the choice of the code and of the signal constellation will not be performed in separate steps [29]. Finally, the detection process will involve soft rather than hard decisions. That is,
instead of processing the received signal before making decisions as to which transmitted symbols they correspond to, the demodulator passes metric information to a soft Viterbi decoder directly. The performance of the system depends on the generator polynomial, which is discussed below.

6.2 PRIMITIVE POLYNOMIALS

Codes with symbols from the binary field GF(2) or its extension GF(2^m) are most widely used in digital data transmission and storage systems because information in these systems is universally coded in binary form for practical reasons. Fields with 2^m symbols are called ‘Galois Fields’, GF(2^m). Their arithmetic involves binary additions and subtractions. For two valued variables, (0,1). The binary alphabet (0,1) is called a field of two elements (a binary field and is denoted by GF(2). In binary arithmetic −X=X and X−Y=X⊕Y [19].

Polynomials f(X) with 1 or 0 as the co-efficient can be manipulated using the above relations. The arithmetic of GF(2^m) can be derived using a polynomial of degree ‘m’, with binary co-efficient and using new variable a called the primitive element, such that p(a)=0. When p(x) is irreducible (i.e. it does not have a factor of degree <m and >0, for example X^3 + X^2 + 1, X^3 + X+ 1 etc are irreducible polynomials, whereas f(X) =X^4 + X^3 + X^2 + 1 is not as f(1) =0 and hence has a factor X+1. An irreducible polynomial p(X) of degree m is said to be primitive if the smallest positive integer n for which p(X) divides X^n +1 is n=2^m -1.

A feedback based on primitive polynomial maximizes the output weight for weight-2 information sequences. Also the choice of feedbacks polynomials of the constituent encoders can affect the performance.

6.3 GENERATOR POLYNOMIALS

More generally, the encoder is realizable if all impulse responses g_j(D) are causal and rational, since each response must be (finitely) realizable by itself, and we can obtain a finite realization of all of them by simply realizing each one separately.

A more efficient realization may be obtained as follows. If each g_j(D) is causal and rational, then g_j(D)= n_j(D)/d_j(D) for polynomials n_j(D) and d_j(D) ≠ 0, where by reducing to lowest terms, we may assume that n_j(D) and d_j(D) have no common factors. Then we can write g(D)=(n'_1(D),n'_2(D),...,n'_n(D))/d(D) = n'(D)/d(D) (6.3)

where the common denominator polynomial d(D) is the least common multiple of the denominator polynomials d_j(D), and n'(D) and d(D) have no common factors. In order that g(D) be causal, d(D) cannot be divisible by D; i.e., d_0 =1. This set of n impulse responses
may be realized by a single shift register of length \( v = \max\{\deg n'(D), \deg d(D)\} \) memory elements, with feedback coefficients determined by the common denominator polynomial \( d(D) \), and with the \( n \) outputs formed as \( n \) different linear combinations of the shift register contents [19], [22].

The component codes are designed using 16-stage shift register with feedbacks from the primitive polynomials as shown in figure. The characteristic polynomials of the feedbacks are \( 1 + X^3 + X^4 + X^9 + X^{16} \) and \( 1 + X^2 + X^4 + X^{15} + X^{16} \). The interesting feature of 16 stage shift register configuration is that feedbacks for all the maximal length configuration have four feedbacks. This is a unique feature and the proposed method generates a trellis encoder having very good structure.

6.4 The Proposed System.

Fig shows the proposed TCM system. It is similar to the conventional system. The input to the mapper function is generated by the circuits which is a 16 stage shift register with a feedback from the primitive polynomials. Two different sets of feedbacks with same number of feedbacks are used to generate the input bits to the mapper.

Connecting the switch to the code terminator can terminate the code. The advantage of this feedback configuration is that it generates quasi-random state in conjunction with the input. These states will increase the Euclidian distance between sequences. The scheme is very attractive in reducing the bit error rate of transmitted message.
Figure 6.2: Performance of 8-PSK, 2 Bits/s/Hz MLCM (64-state) versus TCM over the AWGN Channel

Figure shows the performance of TCM for 8 PSK and 64 states.

### 6.5 Set Partitioning

In TCM, the modulation is an integral part of the encoding process. It is designed in conjunction with the code to increase the minimum Euclidean distance between the pairs of the coded signals. The loss from the expansion of the signal set is easily overcome and a significant coding gain is achieved with relatively simple codes. The key to this integrated modulation and coding approach is to devise an effective method for mapping the coded bits into appropriate signal points so that the minimum Euclidean distance is maximized [30].

Based on the set partitioning, the M-ary constellation is successively partitioned into $2$, $4$, $8$, $2 \log_2 M$ subsets, with size $M/2$, $M/4$, $M/8$, $2 (\log_2 M) - 1$ with progressively larger minimum distances [31]. The set partitioning method follows three Ungerboeck rules:

- **U1**: To parallel transitions are assigned members of the same partition;
- **U2**: To adjacent transitions are assigned members of the next larger partition;
- **U3**: To make all the signals are used equally often.
A translation of a sub-lattice (or a subset) by an element of the original lattice is called a coset. In QAM constellation, each level of partitioning increases the minimum Euclidean distance by 2. The level to which the signal is partitioned depends on the characteristics of the code [32]. A block of the n information bits is separated into two groups, one will be coded and the other remains uncoded. The group of coded bits will be used to select one of the possible subsets in the partitioned signal set, while the uncoded bits are used to select the points in each subset [33], [34], [35].

6.6. **DENOISING METHODS**

The second method of increasing receiver sensitivity is denoising using adaptive thresholding of wavelet coefficients. Denoising algorithms are used for extracting the original signal from noise. Noise suppression methods in signal processing are based on representing the signal in a manner that it is possible to separate the noise component from the
information. The aim of denoising is to remove the noise \( w(n) \) from a signal. The received signal can be modeled as follows,

\[
Y(n) = x(n) + w(n)
\]

Where \( y(n) \) is the received signal, \( x(n) \) is the transmitted signal and \( w(n) \), the noise signal, for example may be a Gaussian noise process, which is statistically independent of \( x(n) \). Denoising can be done by using spectral subtraction, low pass filtering and wavelet transform.

Low pass filtering approaches, which are linear time invariant, can blur the sharp features, in a signal and sometimes it is difficult to separate noise from the signal when there is spectral overlap.

### 6.5.1 Spectral subtraction

The effect of additive noise on the magnitude spectrum of a signal is to increase the mean and variance of the spectrum. The increase in the mean of the signal spectrum can be reduced by the subtraction of an estimate of the noise spectrum from the noisy signal spectrum. This process is known as spectral subtraction.

![Figure 6.4 Spectral subtraction](image)

### 6.5.2 Processing distortions

The main problem in spectral subtraction is the processing distortions caused by the random variations of the noise spectrum and the constraint that the magnitude spectrum must be non-negative. Three sources of distortions of the instantaneous estimate of the magnitude or power spectrum are,

(a) The variation of the instantaneous noise power spectrum about the mean
(b) Distortion due to cross product terms
(c) The nonlinear mapping of the spectral estimate that fall below the threshold.

6.7. WAVELET TRANSFORM METHOD

The wavelet transform is a new and powerful method for representing oscillatory, discontinuous signals or complex combination of these. In this method the basis functions have localized oscillatory form and unlike sinusoids in Fourier Transform, they decay. The main advantage of wavelet lies in the additional "spatial resolution" of the transformed signal. In contrast to the Fourier transform method, the signal is decomposed into waves of finite length, i.e. waves, which are spatially, localized hence the name wavelets. The wavelet transform of a one-dimensional signal has 2 independent variables – frequency and spatial location variable. It leads to a decomposition of say a spectrum into a series of spectra at finer and coarser resolutions.

The wavelet transform is a powerful tool for analyzing the dependence of a signal on scale, position and angle. The wavelet transform coefficient is a measure of change in the value of the signal in a time frequency tile, defined by the wavelet function. For wavelets the amplitude instead of the locations of the Fourier spectra differs from that of noise. If a signal has energy concentrated in a small number of wavelet coefficients, their values will be large in comparison to the noise that has its energy spread over a large number of coefficients. These localizing properties allow filtering of noise from a signal to be very effective. While linear methods tradeoff suppression of noise for broadening of signal features, whereas noise reduction using wavelet transform allows features in the original signal to remain sharp.

Most of the wavelet coefficients of the transform of noisy signals are close to zero. Therefore most obvious way of filtering in the wavelet domain is to identify those wavelet coefficients that are significantly nonzero against the noisy background. This motivates the method of soft thresholding and hard thresholding. Before explaining these methods we can have some basic knowledge on wavelet transform.

6.7.1 Definition of the CWT

Let \( f(t) \) be any square integrable function. The CWT or continuous – time wavelet transform of \( f(t) \) with respect to a wavelet \( \varphi(t) \) is defined as,
\[ W(a,b) = \int_{-\infty}^{\infty} f(t) \frac{\Psi^* \left( \frac{t-b}{a} \right)}{\sqrt{|a|}} \, dt \]

where \(a\) and \(b\) are real and * denotes complex conjugation. Thus the wavelet transform is a function of two variables, \(f(t)\) and \(\psi(t)\) belong to \(L^2(\mathbb{R})\), the set of square integrable functions, also called set of energy signals. A more compact form can be obtained by \(\psi_{a,b}(t)\) as,

\[ \psi_{a,b}(t) = \frac{1}{\sqrt{|a|}} \psi \left( \frac{t-b}{a} \right) \]

Then,

\[ W_{a,b} = \int_{-\infty}^{\infty} f(t) \psi^*_{a,b}(t) \, dt \]

Note that, \(\psi_{1,0}(t) = \psi(t)\)

The normalizing factor of \(1/\sqrt{|a|}\) ensures that the energy stays the same for all \(a\) and \(b\) that is,

\[ \int_{-\infty}^{\infty} |\psi_{a,b}(t)|^2 \, dt = \int_{-\infty}^{\infty} |\psi(t)|^2 \, dt \]

for all \(a\) and \(b\). For a given value of \(a\), the \(\psi_{a,b}(t)\) is a shift of \(\psi_{a,0}(t)\) by an amount \(b\) along the time axis. Thus, the variable \(b\) represents the time shift or translation.

\[ \psi_{a,0}(t) = \frac{1}{\sqrt{|a|}} \psi \left( \frac{t}{a} \right) \]

it follows that \(\psi_{a,0}(t)\) is a time-scaled and amplitude-scaled version of \(\psi(t)\). Since \(a\) determines the amount of time scaling or dilation, it is referred to as the scale or dilation variable. If \(a>1\), there is a stretching of \(\psi(t)\) along the time axis, whereas if \(0<a<1\), there is a contraction of \(\psi(t)\). Negative values result in a time reversal in combination with dilation.

Since the CWT is generated using stretches and translates of the single function \(\psi(t)\), the wavelet, for transform is referred to as mother wavelet. Dialated and time shifted versions of mother wavelet are called daughter wavelets,
The mother wavelet or the wavelet function $\Psi(t)$ should satisfy the following 3 conditions.

1. The function integrates to zero;

$$\int_{-\infty}^{\infty} \Psi(t) dt = 0.$$

2. It is square integrable or equivalently has finite energy;

$$\int_{-\infty}^{\infty} |\Psi(t)|^2 dt < \infty.$$

3. Admissibility condition.

$$C \equiv \int_{-\infty}^{\infty} \frac{|\Psi(\omega)|^2}{|\omega|} d\omega, \ 0 < C < \alpha.$$

**6.7.2 Hard Thresholding**

In this scheme all wavelet coefficients with magnitude below a predefined threshold are simply discarded (i.e. set to zero). In this scheme such coefficients are classified purely as noise, while coefficients above threshold are classified purely as signal. Note that such a scheme may be employed to effect compression as well as noise removal. This is sometimes referred to as the “keep or kill” method.

![Fig.6.5. Hard thresholding-transfer characteristics](image)

$$W_m = 0 \quad \text{if} \ |w_m| < C \quad 6.8$$

$$W_m = w_m \quad \text{if} \ |w_m| \geq C$$

Where $w_m$ is the wavelet coefficients of noisy signal and $W_m$ are the filtered wavelet coefficients.
6.7.3 Soft Thresholding

In this model wavelet coefficients above threshold are classified as signal, whereas those below the threshold are considered to be "signal plus noise", consequently the signals below the threshold are attenuated by a constant amount to reduce the noise power.

\[ W_m = 0 \quad \text{if } |w_m| < \epsilon \]
\[ W_m = w_m (w_m - \epsilon) \quad \text{if } |w_m| \geq \epsilon \]

The choice of thresholding parameter is important. If it is too low, then there will not be any efficient cancellation of noise. If it is too large then there will be loss of information. A general method of threshold parameter selection is based on median of \( W_m \) given by equation (6.9). The performance of the system depends on the selection of the threshold parameter and the signal characteristics.

In case where the standard deviation of the noisy signal, one usually estimate it as the median of wavelet coefficient divided by 0.6745.

\[ \sigma = \frac{\text{median}(W_m)}{0.6745} \]  
6.9

6.8 Adaptive Thresholding - The Proposed Method

Wavelet transform is unique and efficient noise reduction technique. The effectiveness and efficiency of the method relies on the proper selection of the threshold value for the case under study. Hence the value of the threshold parameter has to be adaptively varied to annihilate the noise. Any noise reduction should be adaptive, since the noise content can
change under different environmental conditions. Even though noise is random, the spectrum components are evenly distributed throughout the range.

Figure 6.7 gives the basic block diagram of the proposed noise reduction system. First module converts the received signal into wavelet transform. The function of the second module is to select those components that carry signal information only and reject the noise component. The value of threshold function and thresholding parameter is computed using the equation. The last module converts the signal into time-domain function, which is the denoised signal.

![Block diagram of adaptive thresholding](image)

**6.8.1 Threshold function**

The threshold function is zero for \(|x| < \lambda\). That is these coefficients are removed. Note that as \(|\lambda|\) increases the slope of the function increases.

\[
W_m = \begin{cases} 
\lambda (x-\lambda) & x > \lambda \\
0 & |x| < \lambda \\
\lambda (x+\lambda) & x < -\lambda 
\end{cases}
\]

Where \(W_m\) is the coefficient after thresholding is the input to wavelet transform module and \(\lambda\) is the threshold parameter. Here for positive signal values we have choose the threshold as, \(\lambda (x-\lambda)\). If we choose threshold as \(\lambda (x+\lambda)\), then if any noise coefficients having value slightly greater than the threshold will be amplified. Here we propose a \(\lambda\), which reduces the cost function based on energy content of the signal.

\[
\sum f = E_t - E_e, \quad E_t \text{ is the transmitted signal energy and } E_e \text{ is the received signal energy.}
\]
6.8.1 Threshold parameter calculation

The threshold parameter is computed from the following equation.

\[ \lambda = \alpha (\mu - x_p) \left( \frac{\sigma_N}{\sigma_Y} \right) \]

\( \alpha \) = scaling factor which depends on the signal spread above and below the mean and it is taken as the square root of standard deviation.

\( \mu \) = mean of the wavelet transformed signal; \( \sigma_N^2 \) = noise variance;

\( \sigma_Y^2 \) = signal variance wavelet transformed. \( x_p \) = peak value of wavelet transformed signal;

The proposed adaptive thresholding, has superior noise performance, compared to other conventional method. The slope of the linear threshold will automatically adjust to accommodate the higher weight components, as more and lower weight components are discarded. To obtain optimum performance the value of \( \alpha \) is iterated, so that the cost function is optimized. The signal characteristics and channel characteristics are the prerequisites for the adaptive thresholding to function satisfactory.

6.9. MATLAB Simulation Results

Example 1 \( N_p = 0.02; \lambda = 0.8733; \lambda_D = 0.8909; \)
First figure represents the figure for transmission. Second figure represents the noisy received signal. Third figure represents the wavelet coefficients before and after thresholding. Last figure represents the recovered signal after inverse wavelet transform.

2) Example.2 \( N_p = 0.025; \lambda = 0.9760; \lambda_D = 1.0043; \)
3) Example. $N_p = 0.03; \lambda = 1.0688; \lambda_D = 1.1090$;
The values of all parameters and simulation results are shown in the table below.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Np</th>
<th>M1</th>
<th>Xp</th>
<th>σn</th>
<th>σy</th>
<th>λ</th>
<th>M1d</th>
<th>Xpd</th>
<th>σnd</th>
<th>σyd</th>
<th>ad</th>
<th>λd</th>
<th>error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.02</td>
<td>0.352</td>
<td>7.046</td>
<td>0.017</td>
<td>1.212</td>
<td>1.099</td>
<td>0.354</td>
<td>3.644</td>
<td>0.004</td>
<td>1.123</td>
<td>4.397</td>
<td>0.890</td>
<td>718.5</td>
</tr>
<tr>
<td></td>
<td>0.025</td>
<td>0.349</td>
<td>7.039</td>
<td>0.021</td>
<td>1.213</td>
<td>1.099</td>
<td>0.354</td>
<td>3.671</td>
<td>0.005</td>
<td>1.124</td>
<td>4.399</td>
<td>1.004</td>
<td>630.98</td>
</tr>
<tr>
<td></td>
<td>0.03</td>
<td>0.345</td>
<td>7.033</td>
<td>0.025</td>
<td>1.215</td>
<td>1.180</td>
<td>0.354</td>
<td>3.699</td>
<td>0.006</td>
<td>1.126</td>
<td>4.402</td>
<td>1.109</td>
<td>556.72</td>
</tr>
<tr>
<td></td>
<td>0.035</td>
<td>0.342</td>
<td>7.027</td>
<td>0.029</td>
<td>1.217</td>
<td>1.101</td>
<td>0.354</td>
<td>3.726</td>
<td>0.007</td>
<td>1.128</td>
<td>4.406</td>
<td>1.207</td>
<td>494.14</td>
</tr>
<tr>
<td></td>
<td>0.04</td>
<td>0.338</td>
<td>7.021</td>
<td>0.034</td>
<td>1.219</td>
<td>1.102</td>
<td>0.355</td>
<td>3.753</td>
<td>0.008</td>
<td>1.130</td>
<td>4.411</td>
<td>1.301</td>
<td>442.94</td>
</tr>
<tr>
<td></td>
<td>0.05</td>
<td>0.331</td>
<td>7.008</td>
<td>0.042</td>
<td>1.226</td>
<td>1.104</td>
<td>0.355</td>
<td>3.807</td>
<td>0.010</td>
<td>1.137</td>
<td>4.424</td>
<td>1.475</td>
<td>375.28</td>
</tr>
<tr>
<td></td>
<td>0.06</td>
<td>0.324</td>
<td>6.996</td>
<td>0.051</td>
<td>1.235</td>
<td>1.110</td>
<td>0.356</td>
<td>3.862</td>
<td></td>
<td></td>
<td>4.491</td>
<td>1.643</td>
<td>338.28</td>
</tr>
</tbody>
</table>

Table 6.1 parameter values and simulation results
6.9.1 Mathematical model for noise signal used in the program

Noise fluctuations within a finite bandwidth $\delta f$,

$$V_n(t) = \sum_k N_p \cos[(\omega k - \omega c)t + \theta t + \omega ct]$$

$N_p$ is the noise power in the bandwidth $\delta f$. $\theta t$ = phase

$$V_n(t) = \sum_k N_p \cos[(\omega k - \omega c)t + \theta t] \cos[\omega ct - \sin[(\omega k - \omega c)t + \theta t] \sin \omega ct]$$

$$= X(t) \cos \omega ct + Y(t) \sin \omega ct$$

$$X(t) = \sum_k N_p \cos[(\omega k - \omega c)t + \theta t]$$

$$Y(t) = -\sum_k N_p \sin[(\omega k - \omega c)t + \theta t]$$

In the program we have taken $\omega c = 45$ KHz and since we can not take $k = a$ in the program we take it in the range from zero to ten thousand.

Other parameters are calculated as follows;

CA = Approximation coefficients; CD = detailed coefficient

$M_1$ = mean (CA); $X_p$ = peak value of CA

$$\sigma_n^2 = \frac{\text{median(CA)}}{0.6745} ; \sigma_y^2 = \text{variance (CA)}; \alpha = \text{standard deviation (xin)};$$

$$\lambda = \left| \alpha (M_1 - X_p) \frac{\sigma_n}{\sigma_y} \right|$$

$M_{1d}$ = mean (CD); $X_{pd}$ = peak value of CD

$$\sigma_{nd}^2 = \frac{\text{median(CD)}}{0.6745} ; \sigma_{yd}^2 = \text{variance (CD)}; \alpha_d = 4x\text{standard deviation (xin)};$$

$$\lambda_d = \left| \alpha_d (M_{1d} - X_{pd}) \frac{\sigma_{nd}}{\sigma_{yd}} \right|$$

Error = $\sum |Y_{out}^2 - X^2|$

$Y_{out}$ = signal after inverse wavelet transform and $X$ is the transmitted signal.
**6.10 Comparison between Hard Thresholding and Adaptive Thresholding**

Following tables show various results of hard thresholding.

**Table 6.2**

<table>
<thead>
<tr>
<th>Np</th>
<th>( \Lambda )</th>
<th>( \lambda_d )</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.02</td>
<td>0.2</td>
<td>0.2</td>
<td>1230</td>
</tr>
<tr>
<td>0.025</td>
<td>0.2</td>
<td>0.2</td>
<td>1230.1</td>
</tr>
<tr>
<td>0.03</td>
<td>0.2</td>
<td>0.2</td>
<td>1230.2</td>
</tr>
<tr>
<td>0.035</td>
<td>0.2</td>
<td>0.2</td>
<td>1230.4</td>
</tr>
<tr>
<td>0.04</td>
<td>0.2</td>
<td>0.2</td>
<td>1230.5</td>
</tr>
</tbody>
</table>

**Table 6.3**

<table>
<thead>
<tr>
<th>Np</th>
<th>( \Lambda )</th>
<th>( \lambda_d )</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.02</td>
<td>0.4</td>
<td>0.4</td>
<td>1115.6</td>
</tr>
<tr>
<td>0.025</td>
<td>0.4</td>
<td>0.4</td>
<td>1116</td>
</tr>
<tr>
<td>0.03</td>
<td>0.4</td>
<td>0.4</td>
<td>1116.4</td>
</tr>
<tr>
<td>0.035</td>
<td>0.4</td>
<td>0.4</td>
<td>1116.9</td>
</tr>
<tr>
<td>0.04</td>
<td>0.4</td>
<td>0.4</td>
<td>1117.4</td>
</tr>
</tbody>
</table>

**Table 6.4**

<table>
<thead>
<tr>
<th>Np</th>
<th>( \Lambda )</th>
<th>( \lambda_d )</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.02</td>
<td>0.6</td>
<td>0.6</td>
<td>960.3040</td>
</tr>
<tr>
<td>0.025</td>
<td>0.6</td>
<td>0.6</td>
<td>961.02</td>
</tr>
<tr>
<td>0.03</td>
<td>0.6</td>
<td>0.6</td>
<td>961.78</td>
</tr>
<tr>
<td>0.035</td>
<td>0.6</td>
<td>0.6</td>
<td>962.59</td>
</tr>
<tr>
<td>0.04</td>
<td>0.6</td>
<td>0.6</td>
<td>963.47</td>
</tr>
</tbody>
</table>
### Table 6.5

<table>
<thead>
<tr>
<th>Np</th>
<th>$\Lambda$</th>
<th>$\lambda_D$</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.02</td>
<td>0.8733</td>
<td>0.8733</td>
<td>725.67</td>
</tr>
<tr>
<td>0.025</td>
<td>0.8733</td>
<td>0.8733</td>
<td>726.97</td>
</tr>
<tr>
<td>0.03</td>
<td>0.8733</td>
<td>0.8733</td>
<td>728.22</td>
</tr>
<tr>
<td>0.035</td>
<td>0.8733</td>
<td>0.8733</td>
<td>729.49</td>
</tr>
<tr>
<td>0.04</td>
<td>0.8733</td>
<td>0.8733</td>
<td>730.83</td>
</tr>
</tbody>
</table>

### Table 6.6

<table>
<thead>
<tr>
<th>Np</th>
<th>$\Lambda$</th>
<th>$\lambda_D$</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.02</td>
<td>0.8909</td>
<td>0.8909</td>
<td>710.76</td>
</tr>
<tr>
<td>0.025</td>
<td>0.8909</td>
<td>0.8909</td>
<td>712.11</td>
</tr>
<tr>
<td>0.03</td>
<td>0.8909</td>
<td>0.8909</td>
<td>713.39</td>
</tr>
<tr>
<td>0.035</td>
<td>0.8909</td>
<td>0.8909</td>
<td>714.68</td>
</tr>
<tr>
<td>0.04</td>
<td>0.8909</td>
<td>0.8909</td>
<td>716.05</td>
</tr>
</tbody>
</table>

### Table 6.7

<table>
<thead>
<tr>
<th>Np</th>
<th>$\Lambda$</th>
<th>$\lambda_D$</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.02</td>
<td>0.9760</td>
<td>0.9760</td>
<td>640.23</td>
</tr>
<tr>
<td>0.025</td>
<td>0.9760</td>
<td>0.9760</td>
<td>641.81</td>
</tr>
<tr>
<td>0.03</td>
<td>0.9760</td>
<td>0.9760</td>
<td>643.28</td>
</tr>
<tr>
<td>0.035</td>
<td>0.9760</td>
<td>0.9760</td>
<td>644.70</td>
</tr>
<tr>
<td>0.04</td>
<td>0.9760</td>
<td>0.9760</td>
<td>646.6905</td>
</tr>
</tbody>
</table>
Table 6.8

<table>
<thead>
<tr>
<th>Np</th>
<th>Λ</th>
<th>$\lambda_D$</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.02</td>
<td>1.0043</td>
<td>1.0043</td>
<td>617.53</td>
</tr>
<tr>
<td>0.025</td>
<td>1.0043</td>
<td>1.0043</td>
<td>619.15</td>
</tr>
<tr>
<td>0.03</td>
<td>1.0043</td>
<td>1.0043</td>
<td>620.16</td>
</tr>
<tr>
<td>0.035</td>
<td>1.0043</td>
<td>1.0043</td>
<td>622.16</td>
</tr>
<tr>
<td>0.04</td>
<td>1.0043</td>
<td>1.0043</td>
<td>623.67</td>
</tr>
</tbody>
</table>

Table 6.9

<table>
<thead>
<tr>
<th>Np</th>
<th>Λ</th>
<th>$\lambda_D$</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.02</td>
<td>1.0688</td>
<td>1.0688</td>
<td>567.4</td>
</tr>
<tr>
<td>0.025</td>
<td>1.0688</td>
<td>1.0688</td>
<td>569.19</td>
</tr>
<tr>
<td>0.03</td>
<td>1.0688</td>
<td>1.0688</td>
<td>570.88</td>
</tr>
<tr>
<td>0.035</td>
<td>1.0688</td>
<td>1.0688</td>
<td>572.47</td>
</tr>
<tr>
<td>0.04</td>
<td>1.0688</td>
<td>1.0688</td>
<td>574.04</td>
</tr>
</tbody>
</table>

Bold numbers shows better result compared to adaptive thresholding. From the result it can be noted that for some example the hard thresholding shows better results than adaptive thresholding. But the performance of adaptive thresholding is better under excessive noise conditions.