CHAPTER 4

MULTI CARRIER

COMMUNICATION SYSTEM
4.1 Introduction

The task facing the designer of a digital communication system is that of providing a cost-effective facility for transmitting information from one end of the system at a rate and a level of reliability and quality that are acceptable to the user at the other end. The two key system parameters available to the designer are transmitted signal power and channel bandwidth. These two parameters together with the power spectral density of receiver noise determine the signal energy per bit-to-noise power spectral density ratio $E_b/N_0$. This ratio uniquely determines the bit error rate for a particular modulation scheme. Practical considerations usually play a limit on the value that can be assigned to $E_b/N_0$. Accordingly in practice, a modulation scheme is chosen which is not possible to provide acceptable data quality for a fixed $E_b/N_0$. The only practical option available for changing data quality from problematic to acceptable is to use channel encoding schemes.

The Schematic of wireless system is shown in Figure 4.1 and supports interactive videophone calls. The video signal compressed using video compression standard, but resulting bit stream is sensitive to channel errors and is impediment when used in wireless networks [Lajos Hanzo et al, 2000].
Channel encoding schemes for data integrity may be exercised by means of forward error correction. A digital communication system consists of a discrete source that generates information in the form of binary symbols. The channel encoder in the transmitter accepts message bits and adds redundancy according to a prescribed rule thereby producing encoded data at higher bit rate. The channel decoder in the receiver exploits the redundancy to decide which message bits were actually transmitted. The combined goal of the channel encoder and decoder is to minimize the effect of channel noise.

4.2 Channel Code

In a convolution code the encoding operation may be viewed as the discrete time convolution of the input sequence with the impulse response of the encoder. The duration of the impulse response equals the memory of the encoder. Accordingly the encoding of the convolutional code operates on the incoming message sequence using a sliding window equal in duration to its own memory. This in turn means that, in a convolutional code unlike a block code, the channel encoder accepts message bits as a message sequence.
and thereby generates a continuous sequence of encoded bits at a higher rate
[Lajos Hanzo et al, 2001].

The encoders can typically be implemented by the help of linear shift-
register circuitries, an example of which can be seen in Figure 4.2. In general
a k-bit information symbol is entered into the encoder, constituted by K shift-
register stages. In our example of Figure 4.1 the corresponding two shift-
register stages are \( s_1 \) and \( s_2 \). In general, the number of shift-register stages \( K \)
is referred to as the constraint length of the code. An alternative terminology is
to refer to this code as a memory three code, implying that the memory of the
CC is given by \( K+1 \). The current shift-register state \( s_1, s_2 \) plus the incoming bit
\( b_i \) determine the next state of this state machine. The number of outputs bits is
typically denoted by \( n \), while the coding rate by \( R=k/n \), implying that \( R\leq1 \).

Once a specific bit enters the encoder's shift register in Figure 4.1, it
has to traverse through the register, and hence the register's sequence of
state transitions is not arbitrary. Furthermore, the modulo-2 gates impose
additional constraints concerning the output bit-stream because of these
constraints, the legitimate transmitted sequences are restricted to certain bit
patterns, and if there are transmission errors, the decoder will conclude that
such an encoded sequence could not have been generated by the encoder
and that, it must be due to channel errors. In this case, the decoder will
attempt to choose the most resemblent legitimately encoded sequence and
output the corresponding bit-stream as the decoded string. These processes
will be elaborated on in more detail later in the chapter.
Fig. 4.2 Systematic half rate, constraint length two convolution encoder CC(2,1,2).

Because of its high performance, the Viterbi algorithm is commonly used for decoding the convolution coders and is widely used in different communication standards and communication environments. The algorithm uses a metric and tracks this metric for several trellis paths at once. A Viterbi decoder performs a maximum likelihood detection of data bits transmitted over a channel with inter-symbol interference (ISI). The data bits to be transmitted are encoded with an n-bit convolution code in the convolution encoder. Further insight into the details of encoding operation is dealt with in the following section.

4.3 Higher Modulation Schemes

In the model depicted in the Figure 4.1 the operations of channel coding and modulation are performed separately in the transmitter; likewise for the operations of detection and decoding in the receiver. 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, which requires a wider bandwidth. In a bandwidth-limited environment, the use of higher-order modulation schemes can increase efficiency in frequency utilization. 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 to be combined and *Trellis Coded Modulation* (TCM) is one such technology[Proakis,1995].

4.3.1 State and Trellis Transitions:

An often used technique for characterizing the operations of a state machine, such our convolutional encoder, is to refer to the state transition diagram of Figure 4.3. Given that there are two bits in the shift register at any moment, there are four possible states in the state machine and the state transitions are governed by the incoming bit $b_i$. A state transition due to logical zero is indicated by a continuous line in the Figure, while a transition activated by a logical one is represented by a dashed line. The inherent constraints imposed by the encoder manifest themselves here in that from any state there are only two legitimate state transitions, depending on the binary input bit. Similarly, in each state there are two merging paths. It is readily seen from the encoder circuit of Figure 4.2 that, for example, from state $(s_1,s_2) = (1,1)$ a logical one input results in a transition to $(1,1)$ while an input zero leads to state $(0,1)$. The remaining transitions can also be readily checked. A further feature of this Figure is that the associated encoded output bits are also
plotted in the boxes associated with each of the transitions. Hence, this diagram fully describes the operations of the encoder.

Another simple way of characterizing the encoder is to portray its trellis diagram, which is depicted in Figure 4.4. At the left of the Figure, the four legitimate encoder states are portrayed. Commencing operations from the all-zero register state (0,0) allow us to mirror the encoder's action seen in Figures 4.2 and 4.3 also in the trellis diagram, using the same input bit-stream. As before, the state transitions are governed by the incoming bits b_i and a state transition due to a logical zero is indicated by a continuous line, while a transition activated by a logical one is represented by a dashed line[ Lazos Hanzo,2001].

Again, the inherent constraints imposed by the encoder manifest themselves here in that from any state there are only two legitimate state transitions depending on the binary input bit and in each state there are two merging paths. Given our specific input bit-stream, it is readily seen from the encoder circuit of Figure 4.2 and the state transition diagram of Figure 4.3 that, for example from state (s_1,s_2)=(0,0) a logical zero inputs bit results in a transition to (0,0), while an input one leads to state (1,0). The remaining transitions shown in the Figure are associated with our specific input bit-stream, which can be readily explored. As before, the associated output bits are indicated in the boxes along each of the transitions. Hence, the trellis diagram gives a similarly unambiguous description of the encoder's operations to the state diagram of Figure 4.3.
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. Quadrature Amplitude modulation (QAM) and Quaternary Phase Shift Keying (QPSK) are used in TCM to increase data transmission rate. 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 signals are required to be transmitted without enlarging channel bandwidth.

The above operations are simulated on a step-by-step basis using Verilog HDL. The Viterbi decoder is partitioned into sub modules. Each sub
module performs a specific function characteristic of the decoder and is implemented as an FPGA or ASIC chip.

4.4 VITERBI ALGORITHM

Andrew Viterbi developed the Viterbi algorithm in 1967. Since then, it has become the standard algorithm to decode the convolution codes. At each time interval, the algorithm compares the actual received codeword with the codeword that might have been generated for each possible memory transition. The algorithm chooses the most likely sequence within a specific time frame based on the metrics of similarity. The maximum likelihood decoding requires less memory than the sequential decoding because unlikely sequences are discarded early, leaving a relatively smaller number of sequences that need to be stored.

Convolution codes are a widely used channel coding techniques in today's digital transmission systems. Because of its high performance, the Viterbi algorithm is commonly used for decoding the convolution coders and is widely used in different communication standards and communication environments. For instance, the GSM standard uses a Viterbi decoder with the constraint length of 5 and a rate of 1/2. The WLAN standard specifies a constraint length of 7, rate of 1/2 for the Viterbi decoder. Thus a flexible low-power and a high-throughput Viterbi decoder design is a key challenge for future portable devices.

The decoder operates by finding the maximum likelihood decoding sequence. The usefulness of the Viterbi decoder is depicted in Figure 4.5. At the source the Viterbi encoder encodes the input stream and transmits it to
the destination via a noisy medium. The encoding is such that the Viterbi decoder can remove potential noise in the incoming stream by decoding it. The characteristics of the decoder are its effectiveness in noise elimination, speed of decoding and cost (hardware utilization).

Fig. 4.5 Viterbi decoder

4.4.1 Viterbi Encoder

A short description of the Viterbi algorithm is given in this section. A small system is chosen to illustrate the Viterbi encoding and decoding process. The algorithm used by the Viterbi Decoder belongs to a class of algorithms known as convolution codes. The rate of the convolution coder is defined as the number of input bits to the output bits. The rate of this encoding is 2/3, i.e. the system encodes 2 input bit to 3 output bits. The encoder is depicted in Figure 4.6. In order to encode bit $x(n)$ from the input stream this encoder creates three bits, namely $Y0(n), Y1(n)$ and $Y2(n)$ using the last 4 input bits, i.e. $x_1(n-1)$, $x_1(n-2)$, $x_1(n)$ and $x_2(n)$.

The encoding bits are produced from the equations:

\[
Y0(n) = x_1(n-1)
\]

\[
Y1(n) = x_1(n) + x_1(n-2) \mod 2
\]

\[
Y2(n) = x_2(n)
\]
The number of bits used for encoding one bit is called constraint length" (in this case the constraint length is 3), and the equations that describe the encoding are called "generator polynomials". At the destination the Viterbi decoder decodes the encoded stream providing the more likelihood decoding sequence. The trellis diagram is used for the decoding. Each node in the trellis diagram denotes one of the four potential pairs \((x(n-1), x(n-2))\) of the last two decoded bits. The trellis diagram can be seen as a flow-control diagram where each node represents a state and transitions happen depending on the input stream. From any node we can make a transition to one of two other nodes corresponding to receiving one of the possible inputs. The way the trellis diagram is constructed depends on the constraint length but not on the generator polynomials. The two numbers shown on every transition in the Figure are the results of the above three generator polynomials[Proakis, 1995]

![Trellis Diagram](image)

**Fig. 4.6 Block diagram of the Viterbi Encoder**

Each stage consisting of four nodes at the same vertical position, in the trellis diagram is associated with one decoded bit and the corresponding two encoded bits. The Viterbi decoder finds the maximum likelihood path through the trellis diagram. Ideally the encoded bits should match with the numbers on...
the transitions of the resulting path. A transition also represents a 2-way communication of information between its two nodes. Each node is an Add Compare Select (ACS) module used during Viterbi decoding in order to perform its two major operations, metric update and backtrack (or trace back).

4.5 DECODER USING VITERBI ALGORITHM

The principle of decoding is referred to as an algorithm since it can be implemented in software. The equivalence between maximum likelihood decoding and minimum distance decoding for a binary symmetric channel implies that a convolution code may be decoded by choosing a path in the trellis whose coded sequence differs from the received sequence in the fewest number of places.

The algorithm operates by computing a metric or discrepancy for every possible path in the trellis. The metric of a particular path is defined as the Euclidean distance between the coded sequences represented by that path and the received sequence. Thus for each node in the trellis the algorithm compares the two paths entering the node. The path with the lower metric is retained, and the other path is discarded. This computation is repeated for every level $j$ of the trellis in the range $M \leq j \leq L$, where $M=K-1$ is the encoder's memory and $L$ is the length of the incoming message sequence. The paths that are retained by the algorithm are called survivor or active paths.

4.5.1 Decoding Algorithm

The operations performed by the algorithm could be summarized as follows:
Initialization:

Label the left-most state of the trellis (i.e. the all zero state at level 0) as 0, since there is no discrepancy at this point of computation.

Computation step $j+1$:

Let $j=0,1,2$ etc. and suppose that at the previous step $j$ we have done two things:

- All survivor paths are identified.
- The survivor path and its metric for each of the state of the trellis are stored.

Then, at level $j+1$, compute the metric for all the paths entering each state of the trellis by adding the metric of the incoming branches to the metric of the connecting survivor path from level $j$. Hence for each state, identify the path with the lowest metric as the survivor of step $j+1$, thereby updating the computation.

FINAL STEP:

Continue the computation until the algorithm completes its forward search through the trellis and therefore reaches the termination node at which time it makes a decision on the maximum likelihood path. Then the sequence of symbols associated with that path is released to the destination as the decoded version of the received sequence. In this sense, it is therefore more correct to refer to the Viterbi algorithm as a Maximum likelihood sequence estimator.
4.6 FUNCTIONAL EQUIVALENT OF THE VITERBI DECODER

The Viterbi Decoder consists of many ACS blocks. Each one includes two 8-bit adders and one comparator. The ACS logic computes and transmits two metrics using the smallest metric of the two received metrics and one encoded bit of the input stream. An ACS is called "state" in the diagram. A "stage" of an encoded bit consists of all the states that receive the encoded bit. Finally stages are grouped up into "blocks". The input encoded bits arrive one after the other and the metrics of the stages are computed in the same order. In the Figure 4.8 the input bits are received at the top, and the decoder gives the output-decoded bits at the bottom. The dashed lines correspond to 2-way communications.

A state sends the 8-bit metric that it computes to the states connected at its right during the metric update process, and can receive a 1-bit signal from one of its states connected at its right during the backtracking process.
Fig. 4.8 Implementation of Viterbi decoder using registers

The metric update process works as follows. A stage waits for the metrics from the stage at its left and for the encoded bit from the input stream to arrive. When both arrive all its states will compute their metrics and send them to the states they're connected to at their right. Thus, each state of the stage at the right will receive two metrics. In order to compute its own metric it will compare the two metrics that it received and keep the smaller of the two. The results from these comparisons, generated during the metric update process, are needed in order to generate the decoded bits. It turns out that in order to generate the decoded bits for Block N, we need to have the results from the metric comparisons of Blocks N and N+1. The process of generating these decoded bits from the comparison results is called backtracking since the states are visited in reverse order, i.e. starting from the rightmost stage of Block N+1 and finishing at the leftmost stage of Block N. In contrast to the metric update process where all states computed their metrics, in the backtracking process only one state of each stage is activated. The decoded bits of Block N correspond to the comparison results of the states of this Block that were activated during the backtracking process. As we see whenever the
metric update process computes the metrics of one Block (N+1 in this case),
the backtracking process has to backtrack two Blocks (N+1 and N).

In the Figure, each step at the top represents the computation of the
metrics of a block while each step at the bottom represents a backtracking
step of two blocks. For example, in the first step the decoder computes the
metrics of the first block. In the second step, the decoder computes the
metrics of the second block. In the third step, the decoder backtracks the first
two blocks in order to find the decoded bits of the first block and at the same
time computes the metrics of the third block. It continues in the same way in
the next steps. The resulting operation of the Viterbi decoder is that at each
step it computes the metrics of the next block and backtracks the last two
blocks.[Smith,1999].

Figure 4.9 shows a simple example illustrating the Viterbi algorithm.
The branch metrics, the survivor paths at each node, and the partial path
metric of each surviving path are illustrated in this Figure. The two-state (S0,
S1) trellis with the branch metrics of he transitions is marked and the Viterbi
algorithm is illustrated. The Viterbi algorithm finds iteratively the path with the
minimum path metric of 0.5.

In Figure 4.9 pattern (1) shows the branch metric values from S0 to S1,
or from S1 to S0 at every time interval k=0 to k=4. Pattern (2) assumes
starting state is S0 at time interval k=0, the forward states will be S0 and S1.
The cost for S0-S0 branch is 0.0, for S0-S1 is 0.8. Because 0.0<0.8, the
survival branch is S0-S0, and the accumulate cost is 0.0 at k=1. Pattern (3)
shows starting state is S0 at k=1 and the forward states will be S0 and S1. The cost for S0-S0 branch is 0.3, for S0-S1 is 0.0. At k=2, accumulate cost for S0 is 0.3 (i.e., 0.0 plus 0.3), for S1 is 0.0 (i.e., 0.0 plus 0.0). Because 0.0<0.3, the survival branch is S0-S1. Same process works on patterns (4) and (5). Finally, at k=4, the minimum accumulate cost is 0.5 for branch S0-S0. After tracing back, the survival path is

Fig. 4.9 Viterbi algorithm illustration based on two state trellis
S0-S0-S1-S0-S0; this path is shown in the bold line in Fig. 4.6.

4.7 DESIGN PARAMETERS OF THE VITERBI DECODER

The Viterbi algorithm consists of several parameters. The most significant are:
The way the data stream is received. The encoded bits are transmitted as signed antipodal signals (i.e. 0 is transmitted with a positive voltage and 1 is transmitted with a negative voltage). They are received at the decoder and quantized with an n-bit quantizer. In the project the value of n is chosen to be 3. The quantized number is represented in 2's complement with a range of -4 to 3. The minimum value i.e. -4 represents bit value 0 while the maximum value i.e. 3 represents bit value 1. The decoder can get intermediate values due to noise. This scheme is called 3-level soft decision and used by many Viterbi decoders.

The encoding rate which is the ratio of the number of the decoded bits of the data stream over the number of the encoded bits. The encoding rate used is 2/3, which is used by the majority of the Viterbi decoders.

The constraint length, which is the number of bits that are used for encoding one bit. This is the number of the preceding bits in the input that are used for encoding the bit plus one (the bit being encoded). This parameter is called K in the report and the number of states in a stage must be 2K-1. By increasing the constraint length or the number of states per block the Viterbi algorithm has better noise immunity. The generator polynomials, which define how an input bit at the Viterbi encoder is encoded. The number of states per block denoted as B.

4.8 ARCHITECTURAL IMPLEMENTATION

The trellis diagram could be implemented in hardware. A cost effective solution could be to implement in hardware a smaller number of stages (for example the stages included in one block) and reuse them in every clock
cycle. This number is called H. The output metrics of the last stage can be stored in a register in order to be used by the first stage (of the next block) at the next cycle. The comparisons results of the states can also be held in registers that are used by the backtracking logic. The forwarding logic includes the states of one block that compute the metrics of that block. In every cycle the metrics of one block are computed and the results from the metric comparisons are shifted in the register. This shift register is called S1, while the one used by the backtracking logic is called S2. The shift register has to be large enough to hold the comparison results of the states of two blocks, since the back track logic needs the results of two consecutive blocks in order to find the decoded bits of the first one. In every cycle the forwarding logic computes the metrics and comparison results of one block while the backtracking logic backtracks two blocks finding the decoded bits of the first one.

4.9 TRELLIS CODED MODULATION

In the traditional approach to channel coding, encoding is performed separately from modulation in the transmitter; likewise for decoding and detection in the receiver. Moreover, error control is provided by transmitting additional redundant bits in the code, which has the effect of lowering the information bit rate per channel bandwidth. That is, bandwidth efficiency is traded for increased power efficiency.

To attain a more effective utilization of the available bandwidth and power, coding and modulation have to be treated as a single entity. We may deal with this new situation by redefining coding as the process of imposing certain patterns on the transmitted signal. Indeed, this definition includes the
traditional idea of parity coding. Trellis codes for band-limited channels result from the treatment of modulation and coding as a combined entity rather than two separate operations. The combination itself is referred to as trellis-coded modulation (TCM). This form of signaling has three basic features:

The number of signal points in the constellation used is larger than what is required for the modulation format of interest with the same data rate; the additional point as allow redundancy for forward error-control coding without sacrificing for bandwidth. Convolution coding is used to introduce a certain dependency between successive signal points such that only certain patterns or sequence of signal points are permitted. Soft-decision decoding is performed in the receiver, in which the permissible sequence of signals is modeled as a trellis structure; hence, the name "trellis codes" [Syman Hakins, 2003].

In the two major classes of codes studied so far, i.e., block and convolution codes, an improvement in the performance of the communication system is achieved by expanding bandwidth. In both cases the Euclidian distance between the transmitted code waveforms is increased by use of coding, but at the same time the bandwidth is increased by a factor of $n/k = 1/R_c$. These types of codes have wide applications in cases where there exists enough bandwidth and the communication system designer is not under tight bandwidth constraints. Examples of such cases are deep-space communication systems and storage on high-density media. However, in many practical applications, communication channels with strict bandwidth constraints are dealt with, and the bandwidth expansion due to coding may not be acceptable. For example, in transmission of digital data over telephone
channel (modem design) a channel that has a restricted bandwidth is considered, and the overhead due to coding imposes a major restriction on the transmission rate. Hence, in practical applications the channels involved are bandwidth constrained, an integral coding and modulation scheme called trellis coded modulation is employed.

Trellis-coded modulation, or TCM, is a simple method for designing coded modulation schemes that can achieve good overall performance. This coding-modulation scheme is based on the concept of mapping by set partitioning developed by Ungerboeck. Mapping by set partitioning can be used in conjunction with both block as well as convolution codes, but due to the existence of a simple optimal soft decision-decoding algorithm for convolution codes (the VITERBI algorithm), it has been used with convolution codes. When used with convolution codes, the resulting coded modulation scheme is called Trellis Coded Modulation.

The following example is an illustration of TCM. Consider an 8-PSK-modulation system with the 8 phases being denoted a, b, c, d, e, f, g, and h as shown in Figure 4.10 below. Two different methods of assigning binary digits to the 8 phases are also shown in Fig 4.10 b & c.

![Figure 4.10 8-PSK Modulation](image)

4.7a 4.7b 4.7c

**Fig. 4.10 8-PSK Modulation**
For the moment, assume that signal phase A (0°) was transmitted and that no coding is used. In a high signal-to-noise ratio AWGN channel, the receiver might make an error but if it did, it would probably choose phases B or H. Next most likely for the choices of the detector are the phases C and G and after that comes the pair D and F. The most unlikely choice for the decision by the detector (given that phase A was transmitted) would be phase E. The probability of mistaking one phase decreases exponentially with the squared Euclidean distance between the two phases. Assuming the points are on a circle of radius 1, the squared Euclidean distances between all pairs of phases is listed in the table below. The key point to be observed is how the spacing of these 8 points in itself protects against certain types of errors.

Trellis coding introduces dependency between every successive transmitting data symbol. Trellis codes and multi-dimensional modulation are designed to maximize the Euclidean distance between possible sequences of transmitted symbols. In N dimensions, the Euclidean distance between two points p and q is:

$$\sqrt{\Sigma(p_i - q_i)^2}$$

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.

$$P_e = e^{-\frac{d_{min}^2}{2\sigma^2}}$$

where $P_e$ is message error probability, $d_{min}$ is the minimum Euclidean distance between signal sequences and $\sigma$ is the noise power. Equation (4.1)
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.

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>.586</td>
<td>2</td>
<td>3.414</td>
<td>4</td>
<td>3.414</td>
<td>2</td>
<td>.586</td>
</tr>
<tr>
<td>B</td>
<td>.586</td>
<td>0</td>
<td>.586</td>
<td>2</td>
<td>3.414</td>
<td>4</td>
<td>3.414</td>
<td>2</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>.586</td>
<td>0</td>
<td>.586</td>
<td>2</td>
<td>3.414</td>
<td>4</td>
<td>3.414</td>
</tr>
<tr>
<td>D</td>
<td>3.414</td>
<td>2</td>
<td>.586</td>
<td>0</td>
<td>.586</td>
<td>2</td>
<td>3.414</td>
<td>4</td>
</tr>
<tr>
<td>E</td>
<td>4</td>
<td>3.414</td>
<td>2</td>
<td>.586</td>
<td>0</td>
<td>.586</td>
<td>2</td>
<td>3.414</td>
</tr>
<tr>
<td>F</td>
<td>3.414</td>
<td>4</td>
<td>3.414</td>
<td>2</td>
<td>.586</td>
<td>0</td>
<td>.586</td>
<td>2</td>
</tr>
<tr>
<td>G</td>
<td>2</td>
<td>3.414</td>
<td>4</td>
<td>3.414</td>
<td>2</td>
<td>.586</td>
<td>0</td>
<td>.586</td>
</tr>
<tr>
<td>H</td>
<td>.586</td>
<td>2</td>
<td>3.414</td>
<td>4</td>
<td>3.414</td>
<td>2</td>
<td>.586</td>
<td>0</td>
</tr>
</tbody>
</table>

4.10 Turbo Equalization

Structure of original turbo equalizer Turbo equalization is proposed for serially concatenated rate $R=1/2$, convolution code for BPSK system. The turbo could mitigate the effects of intersymbol interference provided that the channel impulse response (CIR) is known Better performance can achieved by considering the channel’s memory, when performing joint equalization and decoding iteratively. The basic philosophy of the original turbo-equalization technique derives from the iterative turbo decoding algorithm consisting of two Soft-In/soft-out(SISO) decoders The term priori, Information associated with a bit $v_m$ is the information known before equalization or decoding commences. The Extrinsic information associated with a bit $v_m$ is the information provided...
by the equalizer or decoder based on the received sequence. Posteriori is the term for information associated with a bit is the information that SISO algorithm provides taking into account all available source of information about the bit $u_k$. The turbo equalizer of Figure 4.11 consists of SISO equalizer and a SISO decoder. The SISO equalizer generates the posteriori probability upon receiving the corrupted transmitted signal sequence and the a priori probability provided by the SISO decoder [Lajos Hanzo et al, 2000].

Fig. 4.11 SISO Equalizer and SISO Decoder

4.11 Bit Interleaving in TCM

TCM has been applied to channels other than the AWGN channel. One important class of channel models is that of fading channels. There are many such models, one of which is slow fading where the S/N varies slowly over time. This variation in S/N ratio can be thought of as if the signal points in the constellation had a scale that contracted or expanded with time. For example, with PSK modulation, consider that the radius of the circle was described by a random variable $G$ ($G$ for gain) which varied slowly with time. The result would be that if the gain $G$ remained constant over the amount of time corresponding to the path length of the free Euclidean distance error event, then this free Euclidean distance would be multiplied by a factor $G$ (which is a random variable). It would be much better if the fading values could be.
averaged over by multiplying the free Euclidean distance by the average of G. Bit interleaving is one way of accomplishing this averaging. In this approach, a rate $\frac{1}{2}$ convolution code is punctured to obtain a high rate code, the output of the punctured encoder is put through an interleaver and then to a signal mapper. The resulting trellis does not have parallel branches. Parallel branches are bad in TCM codes for fading channels if the mapping is such that the parallel branches can fade with little or no diversity protection.

4.12 Implementation of VITERBI Encoder and Decoder

In this section, an encoder in which every two bits of a data stream are encoded into three bits for transmission is described. The ratio of input to output information in an encoder is defined as the rate of the encoder; and in this work a rate $2/3$ encoder is considered.

4.12.1 Encoder

The following equations relate the three encoder output bits ($Y_n^2$, $Y_n^1$, and $Y_n^0$) to the two encoder input bits ($X_n^2$ and $X_n^1$) at a time $N_t$.

$$
Y_n^2 = X_n^2 \\
Y_n^1 = X_n^1 \text{xor} X_{n-2}^1 \\
Y_n^0 = X_{n-1}^1
$$

The input bits can be represented as a single number. For example, if $X_n^2 = 1$ and $X_n^2 = 0$, then $X_n = 2$ can be written. Equation 4.1 defines a state machine with two memory elements for the two last input values for $X_n^1$, $X_{n-1}^1$ and $X_{n-2}^1$. These two state variables define four states: $\{X_{n-1}^1, X_{n-2}^1\}$, with $S_0 = \{0, 0\}$, $S_1 = \{1, 0\}$, $S_2 = \{0, 1\}$ and $S_3 = \{1, 1\}$. The 3-bit output $Y_n$ is a function...
of the state and current 2-bit input \(X_n\). The state diagram depicting the above mentioned encoder is shown in Figure 4.12.

The eight possible encoder outputs \((Y_n = 0-7)\) for the signals that are transmitted over a noisy communications channel (perhaps a microwave signal to a satellite) using the signal constellation is shown in Figure 4.13. Typically this is done using phase-shift keying (PSK) with each signal position corresponding to a different phase shift in the transmitted carrier signal.

![State diagram](image)

Figure 4.12 A state diagram for the rate 2/3 Viterbi encoder. The inputs and outputs are shown in binary as \(X_{n-1} X_{n-2} Y_n^0 Y_n^1 Y_n^2\), and in decimal as \(X_n/ Y_n\).

![Signal constellation](image)

Fig. 4.13 The signal constellation for an 8 PSK (phase-shift keyed) code.
4.12.2 Receiver

The noisy signal enters the receiver. The receiver has to discover which of the eight possible signals were transmitted at each time step. First, the distance of each received signal from each of the known eight positions in the signal constellation needs to be calculated. Table 4.2 shows the distances between signals in the 8-PSK constellation. The distances in Table 4.2 represent the possible distance measures of the received signal from the 8-PSK signals.

<table>
<thead>
<tr>
<th>Signal</th>
<th>Algebraic distance from signal '0'</th>
<th>$X = \text{Distance from signal '0'}$</th>
<th>$E = X^2$</th>
<th>$B = \text{binary quantized value of } E$</th>
<th>$D = \text{decimal value of } B$</th>
<th>Quantization error $Q = D - 1.75 \frac{E}{E}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$2 \sin (0 \pi / 8)$</td>
<td>0.00</td>
<td>0.00</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>$2 \sin (1 \pi / 8)$</td>
<td>0.77</td>
<td>0.59</td>
<td>001</td>
<td>1</td>
<td>-0.0325</td>
</tr>
<tr>
<td>2</td>
<td>$2 \sin (2 \pi / 8)$</td>
<td>1.41</td>
<td>2.00</td>
<td>100</td>
<td>4</td>
<td>0.5</td>
</tr>
<tr>
<td>3</td>
<td>$2 \sin (3 \pi / 8)$</td>
<td>1.85</td>
<td>3.41</td>
<td>110</td>
<td>6</td>
<td>0.0325</td>
</tr>
<tr>
<td>4</td>
<td>$2 \sin (4 \pi / 8)$</td>
<td>2.00</td>
<td>4.00</td>
<td>111</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>$2 \sin (5 \pi / 8)$</td>
<td>1.85</td>
<td>3.41</td>
<td>110</td>
<td>6</td>
<td>0.0325</td>
</tr>
<tr>
<td>6</td>
<td>$2 \sin (6 \pi / 8)$</td>
<td>1.41</td>
<td>2.00</td>
<td>100</td>
<td>4</td>
<td>0.5</td>
</tr>
<tr>
<td>7</td>
<td>$2 \sin (7 \pi / 8)$</td>
<td>0.77</td>
<td>0.59</td>
<td>001</td>
<td>1</td>
<td>-0.0325</td>
</tr>
</tbody>
</table>
The distances, \( X \), in the first column of Table 4.2 are the geometric or algebraic distances. We measure the Euclidean distance, \( E = X^2 \) shown as \( B \) (the binary quantized value of \( E \)) in Table 4.2. The rounding errors that result from conversion to fixed-width binary are quantization errors and are important in any practical implementation of the Viterbi decoder. The effect of the quantization error is to add a form of noise to the received signal.

4.13 Reconfigurable Channel Coder Algorithm

The Viterbi algorithm is the most popular decoding approach for convolution codes and determines a minimum distance path with regards to Hamming distances applied to each received symbol. The prime limiting factor in the implementation of this efficient algorithm is the need to preserve paths at all \( 2^{k-1} \) trellis states for each received symbol. To minimize this limitation the reconfigurable algorithm is proposed to reduce the average computation and path storage required by the Viterbi algorithm. In a static Viterbi algorithm, all the possible \( 2^{k-1} \) paths are retained (where \( K \) is the constraint length). But, it is advantageous to retain only those paths that satisfy certain cost conditions for each received symbol at each state node. In this context, path retention can be based on the following criteria:

1. Let \( d_{\text{min}} \) be the minimum cost among all surviving paths in the previous trellis stage. Let \( 'T' \) be a threshold value which determines whether a path is retained or not. In the reconfigurable decoder algorithm, a path in the present trellis stage is to be retained if its path metric is less than \( d_{\text{min}} \) of the previous trellis stage + \( T \). By this all the high-cost paths that do
not represent the transmitted data are eliminated from scanning early in the decoding process.

2. Let $N_{max}$ be the total number of surviving paths to be allowed for each trellis stage. This value is set apriori to the start of communication. By this many paths with similar cost are not retained and hence, the number of paths is restricted.

To compare the reconfigurable algorithm a (2,1,3)[input v, no. of input bits occupied at a time b, constant length k] convolution encoder and its state diagram are shown in Figures 4.14 and 4.15. The trellis diagram for the chosen convolution encoder is shown in Figure 4.16. The trellis diagram obtained with the above two changes introduced into the static Viterbi algorithm i.e. reconfigurable Viterbi algorithm Sriram Swaminathan et al., Feb, 2002].

![Fig. 4.14 A (2, 1, 3) Convolutional encoder](image-url)
Fig. 4.15 State diagram for the convolutional encoder.

The states are allocated at different time instants as follows:

a) Initially, at $t=0$, the decoder state is set to \('00'\).

Figure 4.16 is obtained by pruning the number of survivor paths at each stage by calculating the minimum cost (path metric) $d_{\text{min}}$ of the previous stage and using the threshold value \('T'\) and the allowed maximum survivors $N_{\text{max}}$. The states allocated at different time instants are explained as follows:

a) Initially, at $t=0$, the decoder state is set to \('00'\).
b) Two branches emanate from state '00' to states '00' and '10' at t=1 representing encoded transmission 0 and 1 values respectively by the encoder.

c) If the received value at t=0' is '00' as shown in the trellis, it is more likely that a b='0', v='00' was transmitted rather than a b='1', v='11' value since both bits of the latter v would have been corrupted by noise. Hence, the path metric of the top branch is '0' and the bottom branch is '2'. These values on the trellis represent the Hamming distances between the received and expected values.

d) Since state '00' is the only state at t=0', d_{min} is the path metric of state '00', which is '0'. As a result d_{min} + T is '1'. At t=1, the path leading to state '10' does not survive since 2, the current path metric of state '10' is greater than the present value of d_{min} + T. As a result only one branch the branch leading to state '00' survives at t=1'.

e) The new d_{min} to be used at t=2 is the minimum among metrics of all surviving paths at t=1. Since only one path survives at t=1, d_{min} =0, the path metric of state '00'.

f) At each stage the process is repeated until the truncation length is reached and the least error path can be identified.

The proposed reconfigurable algorithm is influenced strongly by the choice of 'T' and N_{max} and this is shown in Figure 4.17. From Figure it can be inferred that if 'T' is small, the average number of paths retained at each trellis stage is reduced. This can increase the BER since the decision on the most likely path has to be taken from a reduced number of possible paths. Alternately, if 'T' is large the average number of survivor paths increases and results in a reduced
BER. Thus increased decoder accuracy comes at the expense of an additional computation and a larger path storage memory. Similarly if $N_{\text{max}}$ is small, paths which satisfy the threshold condition may be discarded, potentially resulting in an increased BER. Alternately, if $N_{\text{max}}$ is increased, extra computation and memory are required, potentially with a little benefit to BER reduction. Optimally, the values of $T$ and $N_{\text{max}}$ are selected so that the BER is within allowable limits and also the resource capability of the hardware is matched.

Fig. 4.17  Trellis diagram for a hard-decision adaptive Viterbi decoder with $T=1$ and $N_{\text{max}}=3$
4.14 FFT Processor Using FPGA

The discrete fourier transform (DFT) of $N$ complex samples $f(k)$, $k=1,...,N-1$

Is defined as: 

$$
(r) = \sum_{k=0}^{N-1} f(k) W^{kr} \quad r=0,1,...,N-1
$$

... (4.2)

Where $W = \exp(-2\pi j / N)$

In the FFT operation employing decimation in frequency algorithm the discrete data samples $f(k)$ are divided into two equal parts $k$ and $h(k)$ with first part having the $N/2$ data samples such as

![Signal flow graph for FFT computation](image)

Fig. 4.18 Signal flow graph for FFT computation

![Basic Butterfly computation](image)

Fig. 4.19 Basic Butterfly computation
DFT of length $N$ is computed through two $N/2$ point DFTs, each of which may again be computed through two $N/4$ point DFTs and so on. This is shown in the signal follow graph as an example for $N=8$ in Figure 4.18. The two point DFT butterfly is depicted in Figure 4.19 [Neil H.E. Weste et al, 2006]

\[
R = P + Q
\]

\[
S = (P - Q)w^k
\]  \hspace{1cm} \cdots \hspace{1cm} (4.3)

The equations can be written in terms of real and imaginary parts of the signal as

\[
R_{re} = P_{re} + Q_{re}
\]

\[
R_{im} = P_{im} + Q_{im}
\] \hspace{1cm} \cdots \hspace{1cm} (4.4)

\[
S_{re} = (P_{re} - Q_{re})\cos(k\theta) + (P_{im} - Q_{im})\sin(k\theta)
\]

\[
S_{im} = -(P_{re} - Q_{re})\sin(k\theta) + (P_{im} - Q_{im})\cos(k\theta)
\] \hspace{1cm} \cdots \hspace{1cm} (4.5)

The last two in the set of equations 4.5 essentially represent a plane rotation operation which is computed by applying the CORDIC algorithm. In this technique, the plane rotation through an angle $\alpha$ is achieved by decomposing the target angle into several elementary angles and carrying out rotations through each of these angles as follows

\[
\alpha = \sum_{i=0}^{M-1} \delta_i \theta_i \quad \text{where} \quad \theta_i = \tan^{-1}(2^{-i})
\] \hspace{1cm} \cdots \hspace{1cm} (4.6)

with $M$ being the word length. And $\delta_i = +1$ or $-1$.

The elementary plane rotation in two dimension and is shown in Figure 4.20

\[
x_{r+1} = x_r \cos(\theta_i) + \delta_i y_r \sin(\theta_i)
\]

\[
y_{r+1} = -x_r \delta_i \sin(\theta_i) + y_r \cos(\theta_i)
\] \hspace{1cm} \cdots \hspace{1cm} (4.7)
Fig. 4.20 Elementary plane rotation of CORDIC operation

With the value of $\delta$ deciding direction of the rotation. Applying the condition $\tan(\theta_i) = 2^{-i}$ the above set can be simplified as

$$x_{i+1} = \cos(\theta_i)(x_i + \delta_i y_i 2^{-i})$$

$$y_{i+1} = \cos(\theta_i)(-\delta_i x_i 2^{-i} + y_i) \quad \ldots \quad (4.8)$$

The end results provided by above equations are

$$x_{i+1} = x_i + \delta_i y_i 2^{-i}$$

$$y_{i+1} = -\delta_i x_i 2^{-i} + y_i \quad \ldots \quad (4.9)$$

$$x_{i+1} = \gamma_M x_M$$

$$y_{i+1} = \gamma_M y_M \quad \ldots \quad (4.10)$$

where scaling factor $\gamma_M = \frac{1}{\prod_{i=0}^{M-1} \cos(\theta_i)}$ depends on word length. The iterative equation depicted in equation 4.9 will have easy digital hardware realization since multiplication with the term $2^{-i}$ is same as i-bit right shifting of the operand. So plane rotation by an arbitrary angle is accomplished by to and fro elementary rotations where the direction of next elementary rotation is determined by the sign of the present error as
\[ \delta_i = \text{Sign}(\varepsilon_i) \text{ where } \varepsilon_{i+1} = \varepsilon_i - \delta_i \text{ with } \varepsilon_0 = \alpha \]

The binary format for the representation of the error angle \( \varepsilon \) offers two special advantages.

* the ease of extending the range of rotation from \( \pm \pi/2 \) to \( \pm \pi \) covering the all quadrants completely

* The scope of representing the rotation angle \( k\theta \) for any butterfly stage in terms of the multiplier \( k \) on it, thereby requiring no multiplier. Direction of rotation during vectoring operation is shown in Figure 4.21 and Quadrant transformation in CORDIC For Maintaining \( -\pi/2 \leq T \leq \pi/2 \) is shown in Figure 4.22. The FFT based OFDM modem is shown in Figure 4.23.

<table>
<thead>
<tr>
<th>Sign(x)</th>
<th>Sign(y)</th>
<th>Rotation</th>
<th>Sign Change At o/p</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Clockwise</td>
<td>No</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Anticlockwise</td>
<td>No</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Anticlockwise</td>
<td>Yes</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Clockwise</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Figure 4.21 Direction of rotation during vectoring operation

Fig. 4.22 Quadrant transformation in CORDIC For Maintaining \( -\pi/2 \leq T \leq \pi/2 \)
Fig. 4.23 FFT based OFDM modem schematic [Lajos Hanzo et al, 2001]