CHAPTER 3

COMBINED SELECTIVE MAPPING AND TURBO CODES
FOR PAPR REDUCTION

3.1 INTRODUCTION

OFDM is now a day widely used for achieving high data rates as well as combating multipath fading in wireless communications. OFDM presented excellent tolerance in channel fading and noise signals in terms of speed and reliability. OFDM combined with turbo codes contributes an excellent scheme of reducing PAPR in OFDM systems (Pawan sharma and Seema Verma 2011).

In this chapter the concept of Selective Mapping Technique (SLM) is applied to the OFDM symbols to reduce high peak signals. SLM method effectively reduces PAPR without any signal distortion (Gumml et al 1996). The journal survey defines the usage of turbo codes based on Parallel Concatenated Convolutional Codes (PCCC) for PAPR reduction (Abdulla A. Abouda 2004). Turbo codes can operate at 0.1 dB from the Shannon capacity limit outperforming any other coding technique known today and performs good role in the PAPR reduction of OFDM systems. The proposed work is based on the utilization of Hybrid Concatenated Convolutional Codes (HCCC). The HCCC is a combination of parallel and serial turbo codes (Divsalar and Pollara 1997). The trellis encoding procedure is carried out for hybrid turbo codes in OFDM transmitter section. Calculation of PAPR is done
for HCCC. The HCCC PAPR values was compared and analysed with other turbo types like parallel, serial, multiple and tail biting.

3.1.1 Concatenated Code

The turbo code is specified by two or more component code called as concatenated codes. Concatenated is a combination of two or more simple building blocks or component codes to achieve large coding gains (William E.Ryan, 2001) The principle behind the concatenated codes is to fed the output of one encoder to the input of another encoder and so on. The final encoder before the channel is known as inner encoder. The schematic representation of concatenated codes is shown in Figure 3.1.

![Figure 3.1 Block diagram of concatenated code](image)

The building blocks in concatenated codes are referred to as Recursive Systematic Convolutional (RSC) codes. The input sequence is encoded using RSC by applying various methods such as generator polynomial, tree diagram, state diagram and trellis diagram representations. The turbo codes are further classified into parallel, serial, multiple, hybrid and tail biting concatenated convolutional codes.
3.1.2 Convolutional Code

Convolutional codes provide better error correction performance than block codes on wireless communication. Convolutional codes are generated by a tapped shift register with two or more modulo-2 adders wired in a feedback network. The generated output is the convolution of incoming bit stream and the bit sequence that represents the impulse response of the shift register and feedback network. Figure 3.2 shows the simple convolutional encoder using three stage shift register.

![Convolutional Encoder Diagram](image)

**Figure 3.2 A simple convolutional encoder**

The incoming information bits are propagated through the shift register and influences several outgoing bits by spreading the information content of each bit among several adjacent bits. The output data at the right side of Fig 3.2 is different from the input stream at the left; the data bits are not present in the coder output. An error in any one output bit can be overcome at the receiver without any information being lost. The state of the
convolution encoder is defined by the shift register contents that will remain after the next input bits are clocked in. If the shift register is ‘K’ bits long and input bits enter in groups of ‘k’ message bits, then the encoder is $2^{K-k}$ states. Putting in a group of ‘k’ input bits causes the encoder to change states, the change of state is called as state transition. From a given state, a convolutional encoder can go to only $2k$ other states (although one of these $k$ options may be remain in the starting state). Each state transition is associated with a unique sequence of the output bits. The quantity ‘K’, which measures the length of the shift register is called as span or constraint length of the encoder. Constraint means that among of all possible bit sequences only some are allowed, they are possible code words. The code words limit the possible bit sequences. If ‘v’ output bits are transmitted for every ‘k’ input inputs, then the encoding rate is $k/v$. A ratio $k/v = 1/2$ and $1/3$ is widely used. Convolution coding and digital modulation techniques can be combined with trellis coding for the transmission of turbo codes. Trellis coding employs a high level modulation approach and allows only certain sequences of modulation states to be transmitted. The receiver can detect errors in the received data strings if an invalid sequence of modulation states is received. The advantage of trellis coding is that coding gain can be achieved with a smaller increase in bit rate than with conventional convolution encoding.

### 3.1.3 Recursive Systematic Convolutional (RSC) Encoder

An encoder is said to be systematic, if the input information sequence appears unchanged at one of its outputs. Furthermore, the encoder is called recursive, when the input does not directly affect the memory state of the encoder, due to the presence of a feedback loop. The schematic of a rate $1/2$ Recursive Systematic Convolutional (RSC) is shown in Figure 3.3.
An element of a generator vector is set to ‘1’ if there is a connection between the encoder input or the output of the corresponding shift register and the modulo-2 adder, otherwise it is set to ‘0’. In the case of the RSC encoder of Figure 3.3, we need a generator vector $G_R$ to describe the connections of the feedback loop and a generator vector $G_F$ to describe the connections of the feed-forward path. The notation used for describing the RSC encoder is RSC $(1, G_F / G_R)$, where ‘1’ corresponds to the systematic output of the RSC encoder. Note that the generator vectors are more conveniently represented as octal numbers. A more compact representation of RSC encoder is RSC $(1, 5/7)$ respectively.

Other significant parameters that characterize a convolutional code are the memory size or memory order, which is equal to the number of shift registers ‘v’ used by the encoder, and the constraint length which refers to the total number of bits involved in the encoding operation at each time step. In the case of rate 1/2 convolutional codes, the constraint length is equal to $v + 1$. 

Figure 3.3 Schematic for a Recursive Systematic Convolutional Encoder
3.1.4 Turbo Encoder

A turbo encoder is the parallel concatenation of RSC codes, separated by an interleaver, as shown in Figure 3.4. The information are applied to the first elementary RSC encoder, and after interleaving, it feeds a second elementary RSC encoder. The input stream is also systematically transmitted and the redundancies produced by encoders 1 and 2 are transmitted as parity outputs. For turbo codes, the main reason of using RSC encoders as constituent encoders instead of the traditional Non-Recursive Non Systematic Convolutional (NR-NSC) codes is to use their recursive nature and not the fact that they are systematic.

![Figure 3.4 A Fundamental Turbo Encoder](image)

To achieve a higher rate, the parity outputs can be punctured. In order to obtain a rate half encoder, the parity bits at the encoder outputs are selected alternately, i.e., one bit from encoder 1 and other bit from encoder 2, as illustrated in Figure 3.5.
The interleaver is an important design parameter in a turbo code. It takes a particular stream at its input and produces a different sequence as output. Its main purpose at the encoder side is to increase the free distance of the turbo code, hence improving its error-correction performance. Some different types of interleavers are block, pseudo-random and odd even interleavers. They differ in the way they shuffle the input symbols. As an example, the block interleaver is explained below. A sequence of k bits is written into an N×M matrix row by row starting from the first row of the matrix. Block interleaving then consists of reading the matrix elements column by column starting from the first one. The resulting sequence is written to an array of length as shown in Table 3.1.
Table 3.1 Conceptual representation of Block Interleaver

<table>
<thead>
<tr>
<th>M</th>
<th>N</th>
<th>N_1</th>
<th>N_2</th>
<th>N_3</th>
<th>N_4</th>
</tr>
</thead>
<tbody>
<tr>
<td>M_1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>M_2</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>M_3</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>M_4</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td></td>
</tr>
</tbody>
</table>

Input sequence : 0, 1, 2, 3,…………12, 13, 14, 15
Output sequence : 0, 4, 8, 12,………..3, 7, 11, 15

3.1.5 Selective Mapping Method

Selective mapping (SLM) is a specific scheme for PAPR reduction. SLM takes advantage of the fact that the PAPR of an OFDM signal is very sensitive to phase shifts in the frequency-domain data (Abdulla and Abouda, 2004)

![Figure 3.6 Block diagram of general SLM technique](image)

Figure 3.6 Block diagram of general SLM technique
In selected mapping (SLM) method a whole set of candidate signals is generated representing the same information, and then the most favorable signal as regards to PAPR is chosen and transmitted. SLM scheme is one of the initial probabilistic approaches for reducing the PAPR problem, with a goal of making occurrence of the peaks less frequent, not to eliminate the peaks. The scheme can handle any number of subcarriers and drawback associated with the scheme is the overhead of side information that needs to be transmitted to the receiver (Chin-Liang Wang et al 2005). Figure 3.6 represents the block diagram of the general SLM technique. The information bits from the data source are applied to the serial to parallel converter. PAPR reduction is achieved by multiplying independent phase sequences \( (P_1, P_2, \ldots, P_u) \) to the original data \( (X_1, X_2, \ldots, X_u) \) and IFFT are applied to all the paths (Irukulapati et al 2009). Then the PAPR of each sequence was determined. The combination with the lowest PAPR is transmitted.

### 3.2 PROPOSED SLM TRANSMITTER

Figure 3.7 represents the block diagram of the proposed system. Input bits are applied randomly and passed to the label inserter. Labels are used to generate the same data in different form in the selective mapping technique.

![Figure 3.7 Block diagram of the proposed SLM transmitter](image-url)
The number of labels depends on the ‘U’ value, where ‘U’ represents the number of SLM paths. The SLM path may vary from 1, 2, 4, 8, 16……U where U=2^m different sequences and ‘m’ defines the length of the inserted bits (Peng Cheng et al 2007). The arrangement of U value can be represented as:

- For U=1, only one path and hence there is no SLM. The input bits are passed directly.
- For U=8, there are 8 SLM paths and hence there are 8 labels representing the data. Labels (8)={000; 001; 010; 011; 100; 101; 110; 111}
- For U=16, 16 SLM paths and hence 16 labels. Labels(16)={0000; 0001; 0010; 0011; 0100; 0101; 0110; 0111; 1000; 1001; 1010; 1011; 1100; 1101; 1110; 1111}
- For U=32, 32 SLM paths and hence 32 labels. Labels (32) = {00000; 00001; 00010; 00011; 00100; 00101; 00110; 00111; 01000; 01001; 01010; 01011; 01100; 01101; 01110; 01111; 10000; 10001; 10010; 10011; 10100; 10101; 10110; 10111; 11000; 11001; 11010; 11011; 11100; 11101; 11110; 11111 }

It must be noted that the length of the labels depends on the U value.

3.2.1 Label Inserter

Label inserter is used to insert the labels at the beginning of the data. Insertion is nothing but the concatenation of each of the labels to the same data. For example, let us consider for U=8, there are 8 labels of length with 3 bits each, 4N bit data and hence 8 SLM paths, such that after label
insertion the length becomes 4N+3(labels). Thus an array of 8 x (4N+3) is formed.

3.2.2  HCCC Turbo Encoder

![Diagram](image)

Figure 3.8  HCCC-Turbo Encoder

Figure 3.8 shows the HCCC Encoder. The parallel concatenation and serial concatenation turbo encoders are combined to form HCCC. The encoding procedure was influenced by four main factors, the number of iterations, constraint length, interleaver design and puncturing. The block diagram of HCCC consists of two interleaver, three RSC (inner, outer and parallel RSC), multiplexer and puncturing unit. Interleaving is basically a way to arrange the input data in order to protect against burst errors and to increase the latency of the system. A hybrid concatenated convolution code with two interleaver is the parallel concatenation of an encoder which accepts the permuted version of the information sequence as its input and with serially concatenated code accepts the unpermuted information sequence (Divsalar and Pollara, 1997). The serially concatenated code consists of an outer encoder, an interleaver and an inner encoder. A RSC codes produces a low weight output with low probability however some inputs still causes a low weight outputs. The code description and generator polynomial selected for hybrid concatenated convolutional codes is shown Table 3.2.
Table 3.2 Code description and generator polynomial for HCCC

<table>
<thead>
<tr>
<th>Code Description</th>
<th>Generator Polynomial</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rate 1/2 Recursive Parallel</td>
<td>[1,5/7]</td>
</tr>
<tr>
<td>Rate 1/2 Non Recursive outer</td>
<td>[7,5]</td>
</tr>
<tr>
<td>Rate 2/3 Recursive Inner</td>
<td>[1,0,5/7],[0,1,6/7]</td>
</tr>
</tbody>
</table>

The concept of multiplexing denotes the reordering of output bits of an encoder into a composite form. The output from the multiplexing unit is in the form of matrix which contains systematic bits in first row, Parity bits I in the second row and parity bits II in the third row. The outputs from the multiplexing unit represents the coding rate 1/3. In order to reduce the coding rate from 1/3 to 1/2, the concept of puncturing is done. Without puncturing code rate will be high and hence PAPR will be high. In error correction codes puncturing is done to remove some of the parity bits after encoding the data. Puncturing increases the flexibility of the system without increasing its complexity (Maria Kovacil et al 2007). The main objective of puncturing in this work is to minimize the systematic bits and to provide approximately equal puncturing of parity bits for the two encoders. The general puncturing algorithm can be specified as follows

\[
\text{Mux o/p} = [ S_1 \quad S_2 \ldots \quad S_N \\
P_{11} \quad P_{12} \ldots \quad P_{N} \quad (\text{RSC 1}) \\
P_{21} \quad P_{22} \ldots \quad P_{2N} \quad (\text{RSC 2}) ]
\]

First row denotes systematic bits; second row denotes parity bits of RSC1; third row denotes parity bits of RSC2.
3.2.3 Subcarrier Mapping

The coded bit stream is modulated into symbols to increase the efficiency of the communication systems. Modulation of the signal changes the amplitude, phase and frequency of that signal. In OFDM, only the phase and amplitude is varied. The frequency is left constant to ensure the orthogonal aspect of the sub-carriers. The modulation schemes used in this work is QAM (Quadrature Amplitude Modulation). The encoded data is mapped with QAM 16 which is best suited for the SLM technique for the OFDM symbol. QAM conveys data by changing some aspect of a carrier signal, in response to a data signal. In the case of QAM, the amplitude of two waves with 90 degrees out of phase with each other is modulated to represent the data signal (Asma Latif and Nasi D. Gohar, 2008). A variety of QAM are available and some of the more common types include 16 QAM and 64 QAM. 16-QAM has four bits equating to a symbol where 64-QAM has six bits per symbol. The purpose of avoiding higher order modulation format is that more constellation points are formed while mapping and it is necessary to transmits more bits per symbol. However the constellation points are closer together which extends to more noise and data errors in the system.

3.2.4 Inverse Fast Fourier Transform

After mapping, N point IFFT is performed. The IFFT process the ‘N’ symbols at a time where ‘N’ is the number of subcarriers in the system. Each of the ‘N’ input symbols has a symbol period of ‘T’ seconds which acts like a complex weight for the corresponding sinusoidal basis function. Since the input symbols are complex, the value of the symbol determines both the amplitude and phase of the sinusoidal signals. The IFFT converts the frequency domain data stream into the corresponding time domain data. To enable the most efficient use of the IFFT function, the sub-carriers is kept to a power of two. After performing IFFT cyclic prefix is added. The
Orthogonality between sub-carriers is mainly achieved by cyclic-prefixing of the input before transmitting. Cyclic-Prefixing (CP) of the input forces the channel to produce results as that of circular convolution instead of linear convolution. Removal of cyclic prefixing is carried out at the receiver side to perform optimum detection of the transmitted bits. The signal from the CP is fed to the selector. Then PAPR is calculated and the selector selects the path with lowest PAPR that is assigned for transmitting the OFDM signal.

3.3 TYPES OF TURBO CODES

The different types of turbo codes compared with HCCC in this work is given below

1. Parallel Concatenated Convolutional Codes (PCCC)
2. Serial Concatenated Convolutional Codes (SCCC)
3. Multiple Turbo Codes (MTC)
4. Tail biting Turbo Codes (TBTC)

3.3.1 Parallel Concatenated Convolutional Codes

![Figure 3.9 Turbo - PCCC Encoder](image)

The Figure 3.9 represents the block diagram of PCCC. The PCCC-Turbo encoder consists of two RSC blocks with an interleaver in between.
them. Since the RSC blocks are connected in parallel they are called as PCCC. The information bits are first encoded by first RSC block and then passing the information bits through an interleaver which are encoded by a second RSC block. Multiplexing & puncturing is done to reduce the code rate from 1/3 to 1/2. Here the systematic bits are transmitted only once. The constraint length and generator polynomial of octal numbers chosen for PCCC is shown in Table 3.3.

**Table 3.3 Generator polynomial description of PCCC**

<table>
<thead>
<tr>
<th>Code Description</th>
<th>G(D)</th>
<th>Octal no</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSC1</td>
<td>[ 1+D^2 / 1+D+D^2, 1 ]</td>
<td>15/17, 1</td>
</tr>
<tr>
<td>RSC2</td>
<td>[ 1+D^2 / 1+D+D^2, 1 ]</td>
<td>15/17, 1</td>
</tr>
</tbody>
</table>

Constraint Length of RSC Block = 4

### 3.3.2 Serial Concatenated Convolutional Codes

The SCCC encoder consists of an outer convolutional encoder of R=1/2 and inner RSC coder of R=2/3 joined serially by an interleaver in between as shown in Figure 3.10.

*Figure 3.10 Turbo - SCCC Encoder*
The inner encoder is required to be an RSC code, whereas the outer encoder need not be recursive. However, RSC inner and outer encoders are often preferred since it is convenient to puncture only parity bits to obtain high code rates. (Divsalar and Pollara, 1998). The constraint length and generator polynomial of octal numbers chosen for SCCC is shown in Table 3.4.

Table 3.4 Generator polynomial description of SCCC

<table>
<thead>
<tr>
<th>Code Description</th>
<th>G(D)</th>
<th>Octal no</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSC1</td>
<td>[1, 0, 1+D^2/1+D+D^2]</td>
<td>[1,0,5/7]</td>
</tr>
<tr>
<td>RSC2</td>
<td>[0, 1, 1+D /1+D+D^2 ]</td>
<td>[0,1,6/7]</td>
</tr>
<tr>
<td>Constraint Length of RSC Block = 3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3.3.3 Multiple Turbo Codes

Multiple turbo codes are a class of parallel concatenated codes with three or more constituent encoders separated by multiple interleavers. A multiple turbo code gives us the flexibility to design codes with low state complexity (Divsalar and Pollara, 1995) The Figure 3.11 represents the block diagram of multiple turbo codes (MTC) encoder. The $X_{1S}$ represents the systematic bits, $X_{2P}$, $X_{3P}$ and $X_{4P}$ denotes the parity bits for RSC1, RSC2 and RSC3 respectively.
The constraint length and generator polynomial of octal numbers chosen for MTC is shown in Table 3.5.

**Table 3.5 Generator polynomial description of turbo- MTC encoder**

<table>
<thead>
<tr>
<th>Code Description</th>
<th>G(D)</th>
<th>Octal no</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSC1</td>
<td>[1+D^2 / 1+D+D^2 , 1 ]</td>
<td>15/17 , 1</td>
</tr>
<tr>
<td>RSC2</td>
<td>[1+D^2 / 1+D+D^2 , 1 ]</td>
<td>15/17,1</td>
</tr>
<tr>
<td>RSC 3</td>
<td>[1+D^2 / 1+D+D^2 , 1 ]</td>
<td>15/17,1</td>
</tr>
</tbody>
</table>

Constraint Length of RSC Block = 4

The puncturing Pattern for multiple turbo codes is different from other types and the algorithm is given below.

\[ \text{Mux o/p = [ } s_1 \ s_2 \ldots \ s_N \] 

\[ \begin{array}{c c c c}
P_{11} & P_{12} \ldots & P_N \ (\text{RSC 1}) \\
P_{21} & P_{22} \ldots & P_{2N} \ (\text{RSC 2}) \\
P_{31} & P_{32} \ldots & P_{3N} \ (\text{RSC 3}) \\
\end{array} \]
The first row denotes systematic bits, second row denotes parity bits of RSC1, third row denotes parity bits of RSC2 and fourth row denotes parity bits of RSC3.

### 3.3.4 Tail Biting Turbo Codes

A problem that was recognized early in the design of turbo-codes is the termination of the trellis (Crozier et al 1998). As with conventional block oriented non-recursive convolutional codes, it is desirable to have the encoder start and stop in a known state. This is achieved for non-recursive convolutional codes by starting the encoder in a known state (usually the zero state) and then using known flush bits to terminate the encoder. The flush bits are usually zeros to force the encoder back into the zero state (zero flush). This result in a simple decoder, but the flush bits take energy away from the information bits and also lower the code rate or bandwidth efficiency of the code. One solution to this problem is to perform tail-biting, where the constraint length encoder is initialized with the first $k-1$ information bits and then flushed at the end of the block with the same $k-1$ information bits. This eliminates the overhead associated with the flush bits. In this case the starting and ending states are the same, but unknown (Yung-Chih Tsai and Yeong-Luh Ueng, 2007).

The termination problem is more complicated for turbo codes. This is because there are two trellises to be terminated and the input to the second encoder is an interleaved version of the input to the first encoder. The most common approach is to add termination bits to the end of the first uninterleaved block, resulting in termination of the first trellis, but leave the second trellis un terminated. The first encoder cannot be simply flushed with zero bits because the encoder is recursive. To determine the termination bits required to force the encoder into a given state such as the zero state, the encoder and decoder are the same as that of the PCCC. The only difference is
the termination of the trellis. The concept of tail biting is shown in the Figure 3.12.

![Figure 3.12 Concept of Tail Biting Turbo codes](image)

### 3.4 GENERAL ALGORITHM

The general algorithm for the proposed work is explained as follows

**Step 1:** Get the No. of Symbols (N)

**Step 2:** Get the No. of SLM paths (U)

**Step 3:** Enter 0 for puncturing and 1 for no-puncturing. If puncturing data rate of the encoder is 1/2 else data rate is 1/3.

**Step 4:** Initialize the threshold and count values for plotting $\text{PAPR}_0$ vs PAPR

**Step 5:** Generate the input bits of N symbols of length 4 bits each (4 * N bits) such that 0’s and 1’s are of equal probability.

**Step 6:** From the ‘U’ value, calculate the length of the labels and initialize the labels.

**Step 7:** Turbo encoding function which is to be performed is called. Determine the parameters N, U, labels, data and punctured outputs. The output of the encoder is returned from the function.
Step 8: Once the bits are encoded, mapping is done. Before mapping, check for zero padding.

Step 9: Re-assemble the zero padded bits symbol-wise, so that mapping becomes easier. Map the zero-padded bits using QAM-16.

Step 10: Compute N point IFFT of the mapped sequence and add cyclic prefix.

Step 11: Calculate the PAPR value.

Step 12: Calculate minimum PAPR for each iteration. Calculate the Probability of PAPR valued that exceeds the threshold values in the case of U=1, 8, 16 and 32.

Step 13: Plot the CCDF graph between threshold and probability values (Threshold on x-axis and Probability (PAPR>PAPR₀) on y-axis

Step 14: Compare the output results for various turbo does.

Step 15: Stop the program.

3.5 ALGORITHM FOR HYBRID CONCATENATED CONVOLUTIONAL CODES

Step 1: Get the parameters N and U value .Initialize the label and data. Concatenate the data and labels such that labels are inserted before data. Therefore ‘U’ SLM paths are created.

Step 2: Define the trellis structure for inner encoder 1 with its respective constraint length & generator polynomial.

Step 3: Convolutionally encode (RSC 1) the concatenated bits according to the trellis structure defined. Feedback is given to the convolutional
encoder such that it becomes recursive systematic convolutional
encoder.

**Step 4:** Generate a random interleaved pattern. Interleave the convolutional encoder RSC1 output (parity bits) using the generated interleaved pattern

**Step 5:** Define the trellis structure for outer encoder 1 with its respective constraint length & generator polynomial. Convolutionally encode the interleaved data (RSC2), according to the new trellis structure defined.

**Step 6:** Generate a random interleaved pattern. Define the trellis structure for parallel encoder with its respective constraint length & generator polynomial. Convolutionally encode the interleaved data, according to the new trellis structure defined.

**Step 7:** The output of the serial inner and outer encoders are converted into a rate $\frac{1}{2}$ encoder output using the general puncturing pattern.

**Step 8:** Re-organize the output bits from the two encoders (serial and parallel). Separate the systematic & parity bits from serial output and parallel output. The systematic bits of parallel encoder can be left since it is an interleaved version input/systematic bits.

**Step 9:** Re-organize the systematic bits, parity bits (serial RSC), parity bits (parallel RSC) into a matrix form (multiplexed output) forming row 1, row 2, row 3 respectively.

**Step 10:** If puncture='0' go to step 10. If puncture='1' go to step 11.

**Step 11:** Puncture even parity bits for odd systematic bits and odd parity bits for even systematic bits and the output is returned
3.6 FLOWCHART FOR HCCC

Figure 3.13 Flow Chart for HCCC Encoder
3.7 SIMULATION AND RESULTS

SLM combined with turbo encoding was carried out for serial, parallel, multiple, tail biting and Hybrid turbo codes. PAPR is calculated for all types of turbo codes applied in this work and comparison was done using CCDF plot. Graph is plotted among threshold and CCDF values. Simulations were carried out by using MATLAB software. The OFDM signal of SLM with N=128 subcarriers, 16 QAM mapping, coding rate 1/2 and 1/3 was considered. The SLM paths selected in this work is U=1, 8, 16 and 32. The CCDF plot representing PAPR of turbo codes are shown as follows.

Figure 3.14 CCDF plot of the SLM for N=128; U=1, 8, 16, 32; Punctured (rate=1/2), PCCC
Figure 3.15 CCDF plot of the SLM for N=128; U=1, 8, 16, 32; un-punctured (rate =1/3) PCCC

Figure 3.16 CCDF Plot of the SLM for N=128; U=1,8,16,32, Punctured (rate = 1/2), SCCC
Figure 3.17 CCDF Plot of the SLM for N=128; U=1,8,16,32, unpunctured (rate = 1/3), SCCC

Figure 3.18 CCDF plot of the SLM for N=128; U=1, 8, 16, 32; Punctured (rate=1/2), MTC
Figure 3.19  CCDF plot of the SLM for N=128; U=1, 8, 16, 32; unpunctured (rate =1/3) MTC

Figure 3.20  CCDF plot of the SLM for N=128; U=1, 8, 16, 32; Punctured (rate=1/2), TBTC
Figure 3.21 CCDF plot of the SLM for $N=128$; $U=1, 8, 16, 32$; un-punctured (rate $=1/3$) TBTC

Figure 3.22 CCDF plot of the SLM for $N=128$; $U=1, 8, 16, 32$; Punctured (rate=$1/2$), HCCC
Figure 3.23 CCDF plot of the SLM for $N=128$; $U=1, 8, 16, 32$; un-punctured (rate = 1/3) HCCC

The Figure 3.14 to 3.23 shows the CCDF plot for punctured and un-punctured performance of serial, parallel, Hybrid, multiple and tail biting turbo codes. The coding rate of the Encoder is 1/2 and 1/3 with $N=128$ for different SLM paths by taking threshold along x-axis ($\text{PAPR}_0$) and probability ($\text{PAPR} > \text{PAPR}_0$) along y-axis. Threshold values are chosen from 0 to 25. Simulation results from Figure 3.14 to 3.23 shows that if $U$ SLM path increases the PAPR decreases. Similarly if the coding rate increases the PAPR also increases. The HCCC shows a better PAPR reduction when compared to other types. For 1/2 coding rate HCCC, with $U=1$ the PAPR value is 21.7 dB. There is 0.5 dB difference between $U=8$ and 16, nearly 1 dB difference between $U=16$ and 32.

Figure 3.23 shows the CCDF plot for HCCC, $N=128$ (rate = 1/3) for the SLM paths $U=1, 8, 16, 32$. For $U=1$ the PAPR value is 23.8 dB and for $U=8$ to 32 there is an average of 0.7 dB difference of PAPR reduction is
attained. Simulations results from Figure 3.22 and Figure 3.23 defines that the PAPR value for 1/2 coding rate is around 21 dB for U=1 and for 1/3 coding rate the PAPR value is 24 dB.

Figure 3.24 CCDF plot of the SLM for N=128; U=1; Punctured (rate=1/2), Turbo codes (parallel, serial, hybrid, multiple and tail biting)

Figure 3.24 shows the CCDF plot for N=128 for U=1 (1/2 rate). The HCCC attains a good PAPR reduction when compared to serial, parallel, tail biting and multiple turbo codes. There is a 1.6 dB difference of PAPR values between hybrid and parallel and 0.98 dB from serial concatenated convolutional codes.
Figure 3.25 shows the CCDF plot for N=128 for U=1 (1/3 rate). There is 2dB PAPR difference is maintained between serial to hybrid and 0.5 dB differences for parallel, tail biting and multiple turbo codes.

![CCDF plot for N=128, U=1, un-punctured (rate=1/3), turbo codes (parallel, serial, hybrid, multiple and tail biting)](image)

**Figure 3.25** CCDF plot of the SLM for N=128; U=1; un-punctured (rate=1/3), turbo codes (parallel, serial, hybrid, multiple and tail Biting)

Figure 3.26 shows the CCDF plot for N=128, U=32(rate= 1/2) .The HCCC shows a good performance in PAPR reduction .There is a 4dB difference between multiple and hybrid turbo codes. The HCCC attains a 22% of PAPR reduction when compared to multiple turbo codes, 15% from tail biting, 13% from parallel and 10% from SCCC.
Figure 3.26 CCDF plot of the SLM for N=128; U=32; Punctured (rate=1/2), Turbo codes (parallel, serial, hybrid, multiple and tail Biting)

Figure 3.27 shows the CCDF plot for N=128, U=32 (rate=1/3). The HCCC attains a 20% of PAPR Reduction when compared to multiple turbo codes and 8.6% from tail biting and nearly 9% from serial and parallel convolutional concatenated codes. Figure 3.28 and 3.29 shows the CCDF plot for two SLM paths i.e U=8 & 32 with N=128. The HCCC dominates all types of turbo codes used in this study. The simulation results of HCCC analysed in this chapter with different paths (‘U’ value) states that the PAPR performance for 1/2 coding rate is better when compared to 1/3 coding rate.
Figure 3.27 CCDF plot of the SLM for N=128; U=32; un-punctured (rate=1/3), turbo codes (parallel, serial, hybrid, multiple and tail Biting)

Figure 3.28 CCDF plot of the SLM for N=128; U=8&32; punctured (rate=1/2), turbo codes (parallel, serial, hybrid, multiple and tail Biting)
Figure 3.29 CCDF plot of the SLM for \( N=128; U=8\&32 \) un-punctured (rate=1/3), turbo codes (parallel, serial, hybrid, multiple and tail Biting).

The PAPR comparison table for different types of turbo codes with coding rates 1/2 and 1/3 is given in Table 3.6.

### Table 3.6 Comparison Table for family of Turbo Codes with PAPR in dB

<table>
<thead>
<tr>
<th>Turbo codes</th>
<th>PAPR(dB) for coding rate '1/2' (Punctured)</th>
<th>PAPR(dB) for coding rate '1/3' (Un-Punctured)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No of Paths</td>
<td></td>
</tr>
<tr>
<td></td>
<td>U=1</td>
<td>U=8</td>
</tr>
<tr>
<td>Multiple</td>
<td>23.1</td>
<td>20</td>
</tr>
<tr>
<td>Tail Biting</td>
<td>23</td>
<td>18.5</td>
</tr>
<tr>
<td>Parallel</td>
<td>22.90</td>
<td>18.2</td>
</tr>
<tr>
<td>Serial</td>
<td>22.28</td>
<td>17.5</td>
</tr>
<tr>
<td>Hybrid</td>
<td>21.7</td>
<td>15.9</td>
</tr>
</tbody>
</table>

The table 3.6 states the analyses of different SLM paths for multiple, tail biting, parallel, serial and hybrid turbo codes. For all the SLM paths the HCCC attains a better PAPR reduction when compared to other types of turbo codes applied in this study.
3.8 BAR CHART ANALYSIS

The bar chart analysis from Figure 3.30 to Figure 3.39 clearly briefs the comparison of HCCC with all types of turbo codes used in this study. The PAPR comparison of coding rates and different SLM paths are also analysed in the bar chart analysis.

Figure 3.30 Comparison of PAPR values for hybrid (vs) multiple turbo codes (1/2 coding rate)

Figure 3.31 Comparison of PAPR values for hybrid (vs) tail biting turbo codes (1/2 coding rate)
Figure 3.32 Comparison of PAPR values for hybrid (vs) parallel concatenated convolutional codes (1/2 coding rate)

Figure 3.33 Comparison of PAPR values for hybrid (vs) serial concatenated convolutional codes (1/2 coding rate)
Figure 3.34 Comparison of PAPR values for hybrid (vs) multiple turbo codes (1/3 coding rate)

Figure 3.35 Comparison of PAPR values for hybrid (vs) tail Biting turbo codes (1/3 coding rate)
Figure 3.36 Comparison of PAPR values for hybrid (vs) parallel concatenated convolutional codes (1/3 coding rate)

Figure 3.37 Comparison of PAPR values for hybrid (vs) serial concatenated convolutional codes (1/3 coding rate)
Figure 3.38  Comparison of PAPR values for all types of turbo codes used in this study (1/2 and 1/3 coding rate)

Figure 3.39  Comparison of PAPR values for all types of turbo codes with SLM paths U= 1,8,16 and 32
3.9 SUMMARY

The comparison table and bar chart defines that the usage of multiple turbo codes shows the high PAPR values when compared to tail biting and all other codes in this study. The PAPR values of PCCC attains very closer to serial and shows a good improvement when compared to tail biting and multiple turbo codes. For the SLM paths U=1,8,16 and 32 and N=128, the utilization of HCCC gives a good PAPR reduction results when compared to serial, parallel, tail biting and multiple turbo codes. The work is designed for any number of SLM paths, coding rates and different ‘N’ values. For all the above parameters the HCCC performs a good role in the PAPR reduction.