BIT-MAPPING AND CHANNEL CODING FOR SPACE MODULATION

A Thesis

Presented to

The Graduate Faculty of The University of Akron

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

Husam Elfadil

December, 2017
Spatial modulation (SM) is a multi-input-multi-output (MIMO) technique that exploits the channel state information (CSI) to transmit information using only one antenna at a time. Space shift keying (SSK) is a special case of SM, in which only the space domain is used for transmission. The combination of SM/SSK with existing performance enhancement techniques has been studied because of its potential to achieve additional performance gains. In this thesis, we investigate the use of the CSI in SM and SSK in two aspects, namely, the combination of SM/SSK with bit-mapping and with trellis coded modulation (TCM).

In the first part of the thesis, we propose a novel low-complexity adaptive bit mapping scheme for SM called the nearest point method (NPM). In this scheme, the SM symbols are first sorted according to a certain sorting order; then, bit sequences that are different only in one bit are assigned to subsequent SM symbols according to that sorting order. To determine the sorting order, NPM starts with the pair of SM symbols that has the minimum distance. From one of the SM symbols in that pair, it identifies the nearest SM symbol and continues to do so until it visits all the SM symbols without visiting the same SM symbol twice. Simulation results show that NPM achieves performance close to the lower bound of the bit error rate (BER)
for SM and space shift keying (SSK) using full CSI at the transmitter (CSIT). It also achieves BER improvement for SSK using partial CSIT. The complexity of the proposed NPM is on the order of $O(K^2)$, where $K$ is the number of symbols, which is remarkably low.

In the second part of the thesis, we investigate a trellis coded space shift keying modulation scheme, called trellis coded space-time shift keying (TC-STSK), which combines trellis coded modulation (TCM) with space shift keying (SSK). By analyzing the rank criteria of the TC-STSK scheme using a graph theoretic approach, we try to maximize the number of unique differences between pairwise codewords as the code design goal. We discuss the code design principles for achieving diversity, and propose a design algorithm that is mostly suitable for rate-$1/n$ codes, where $n$ is the number of the code output bits. We also propose a diversity adaptation mechanism by selecting only the codewords with desired diversity, and we use it to achieve full diversity. We then provide the encoder design and a modified soft Viterbi decoder for such schemes. Simulation results show that the proposed scheme can achieve diversity orders expected from theoretical analysis, and decreasing the bit rate and increasing the number of shift registers and transmit antennas improves the performance of the TC-STSK scheme. The results also show that TC-STSK outperforms the space-time trellis code (STTC) for the same bit rate and number of shift registers.
**LIST OF TABLES**

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Level ( m ) groups and subgroups ( m = 1, 2, 3 ) for rate-1/n code with ( v = 3 ) shift registers</td>
<td>41</td>
</tr>
<tr>
<td>3.2</td>
<td>Inputs, outputs, and end states for each starting state for a rate-1/2 4-state convolutional code in a TC-STS K system with ( N_t = 4 ) and ( T = 3 )</td>
<td>58</td>
</tr>
<tr>
<td>3.3</td>
<td>Selected inputs for each starting state</td>
<td>59</td>
</tr>
<tr>
<td>3.4</td>
<td>Designed codes for the TC-STS K scheme</td>
<td>65</td>
</tr>
<tr>
<td>3.5</td>
<td>Bit rate for codes ( A, B, C, ) and ( D ) for different diversity orders</td>
<td>66</td>
</tr>
</tbody>
</table>
### List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Block diagram of a MIMO transmitter with SM transmission.</td>
<td>8</td>
</tr>
<tr>
<td>2.2</td>
<td>BER performance of SM ((M = 4, N_t = 8, N_r = 1, 2)) for different bit mapping schemes using full CSIT.</td>
<td>18</td>
</tr>
<tr>
<td>2.3</td>
<td>BER performance of SSK ((N_t = 32, N_r = 1, 2)) for different bit mapping schemes using full CSIT.</td>
<td>19</td>
</tr>
<tr>
<td>2.4</td>
<td>BER performance of SM ((K = 128, N_r = 1)) for (\Gamma_c) and NPM using full CSIT for different number of transmit antennas.</td>
<td>21</td>
</tr>
<tr>
<td>2.5</td>
<td>BER performance of uncoded and coded SSK ((N_t = 8, N_r = 1, 2)) with a rate-3/5 convolutional code for different bit mapping schemes using partial CSIT.</td>
<td>22</td>
</tr>
<tr>
<td>2.6</td>
<td>BER performance of MSSK ((N_t = 16, N_r = 1, 2)) for different bit mapping schemes using full CSIT.</td>
<td>23</td>
</tr>
<tr>
<td>3.1</td>
<td>Block diagram of a MIMO system with TC-STSK transmission.</td>
<td>29</td>
</tr>
<tr>
<td>3.2</td>
<td>Graph representation of the incidence matrix in Example 3.2.1.</td>
<td>32</td>
</tr>
<tr>
<td>3.3</td>
<td>A graph that represents the condition of Lemma 3.3.4 for (v = 3) and (k = 1).</td>
<td>42</td>
</tr>
<tr>
<td>3.4</td>
<td>A graph that shows the connection between pairs of states for (v = 3) and (k = 1).</td>
<td>43</td>
</tr>
<tr>
<td>3.5</td>
<td>Trellis diagram of the modified Viterbi decoder in Example 3.5.3.</td>
<td>64</td>
</tr>
<tr>
<td>3.6</td>
<td>BER performance of different rate-1/(n) TC-STSK schemes and uncoded SSK with (N_t = 2).</td>
<td>67</td>
</tr>
<tr>
<td>3.7</td>
<td>BER performance of rate-2/3 TC-STSK and uncoded SSK with (N_t = 4).</td>
<td>68</td>
</tr>
<tr>
<td>Section</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td>------</td>
</tr>
<tr>
<td>3.8</td>
<td>BER performance of rate-1/n codes with $R_b = 1$, $v = 3$ and different number of transmit antennas</td>
<td>70</td>
</tr>
<tr>
<td>3.9</td>
<td>BER performance of rate-$k/4$ codes with $N_t = 16$ and $v = 4$.</td>
<td>71</td>
</tr>
<tr>
<td>3.10</td>
<td>BER performance of rate-1/4 codes with $R_b = 1$ with $N_t = 16$ and different number of shift registers.</td>
<td>72</td>
</tr>
<tr>
<td>3.11</td>
<td>BER performance of four states QPSK-STTC code and full bit rate rate-2/3 TC-STSK code with $N_t = 2$ and $v = 2$.</td>
<td>73</td>
</tr>
</tbody>
</table>
Spatial modulation (SM) is a promising multi-input-multi-output (MIMO) transmission technique. In SM, multiple antennas are used for transmission, but only one antenna is transmitting at a time. By transmitting using only one antenna at each time instant, the index of the transmitting antenna is used to convey information. In SM, the transmitted bit stream consists of two parts: bits that select the amplitude-phase modulation (APM) symbol, and bits that select the transmitting antenna (see e.g. [1–13]. SM has been extensively studied because of its potential advantages such as using only one RF chain and the complete avoidance of inter-antenna synchronization at the transmitter and inter-channel interference at the receiver. As a special case of SM, Space shift keying (SSK) uses only the spatial domain to transmit information (see e.g. [14–20]).

The increasing development of channel estimation techniques [21–24] has made it more feasible to assume the knowledge of channel state information at the receiver (CSIR) in many communication systems, which can be used for different purposes such as improving the bit error rate (BER) and lowering the detection complexity [25–27]. SM/SSK exploits the CSIR for transmitted antenna index detection, and, therefore, uses the channel itself as an additional modulation unit [2]. Thus,
making the use of the CSIR a key idea for SM/SSK to achieve higher bit rates and spectral efficiency. The CSI at the transmitter (CSIT) is crucial for multiplexing gains and error improvement in many MIMO systems [26,28–30].

Several works have extensively studied exploiting the use of CSI in SM/SSK to combine the scheme with existing performance improvement techniques, such as power allocation and antenna selection, for additional performance improvements [9,10,16,31–39]. In this thesis, we investigate the combination SM and SSK with two of these enhancement techniques, namely, bit-mapping [40,41] and the trellis coded modulation (TCM) [42–51].

In Chapter 2 of this thesis, we study combining spatial modulation with bit-mapping. Since the way that bits are assigned to symbols affects the BER [52], bit-mapping has been recently studied as an interesting aspect of spatial modulation design. We propose a novel low-complexity bit-mapping scheme for spatial modulation that, while achieving the same BER performance as the existing bit mapping algorithms, it provides a significantly lower computational complexity.

TCM is introduced by Ungerboeck in [42] as a scheme that integrates a convolutional code with a bandwidth efficient modulation scheme such as phase shift keying (PSK) or quadrature amplitude modulation (QAM) to achieve higher coding gains compared to an uncoded system. Traditionally, coding and modulation have been considered separately. However, in TCM coding and modulation are combined to achieve better performance. The output of a convolutional code is used to select a symbol from a constellation. For example, one can use a 2/3 convolutional code
followed by an 8-PSK modulator. In this case, using an 8-PSK modulator instead of a quadrature PSK (QPSK) modulator in the case of the uncoded system increases the error rate because of the reduction in the distance between signal points. However, the TCM compensates for that by providing a coding gain.

The TCM scheme proposed by Ungerboeck in [42] was designed for additive white Gaussian noise (AWGN) channels. In [42], the modulation symbols are divided into groups, and it is shown that the design criterion for TCM in AWGN channels is maximizing the distance between symbols in each group. In [51], the authors proposed a TCM design for single-input-single-output (SISO) fading channels. It is shown that, in addition to the coding gain, TCM in SISO fading channel can achieve time diversity [51]. The time diversity order is equal to the effective length of the convolutional code, which is defined as the minimum number of symbols in error when an error event occurs. Therefore, the design criteria for TCM in SISO fading channels is maximizing the effective length for diversity advantage as well as maximizing the minimum product distance for coding gain advantage. However, the time diversity can be achieved only if the channel is a fast fading channel, in which the channel changes in every time interval. If the fading is not fast, i.e., slow or block fading channel, then there is a little to no time diversity.

In Chapter 3 of this thesis, we investigate the combination of TCM and SSK in a slow fading channel. By combining TCM and SSK, when the channel is changing slowly, it is possible to exploit the space to get some spatial diversity. We analyze the diversity gain criteria of this combination using a graph theoretic
approach. Accordingly, we discuss the design guidelines and we propose a code design algorithm. We further improve performance of the combination scheme, by proposing a diversity adaptation mechanism, in which we only select the codewords with desired diversity, with the appropriate encoding and decoding designs.

The rest of the thesis is organized as follows. Chapter 2 introduces a novel bit mapping algorithm for spatial modulation. The design and performance evaluation of the proposed hybrid TCM and SSK scheme is investigated in Chapter 3 and finally, Chapter 4 concludes the thesis with some suggestions for future work.
CHAPTER II

A NOVEL LOW-COMPLEXITY ADAPTIVE BIT MAPPING SCHEME FOR SPATIAL MODULATION

2.1 Introduction

Bit mapping is the process of assigning bit sequences to different symbols. The choice of bit mapping affects the bit error rate (BER) performance of any modulation scheme [52]. In conventional APM schemes, such as phase-shift keying (PSK) and quadrature amplitude modulation (QAM), a well-known heuristic method for bit mapping is Gray coding, which is proven to be optimal in the case of equally likely and statistically independent APM symbols [52]. In Gray coding, neighboring symbols are assigned bit sequences that are different only in one bit.

While applying Gray coding for conventional APM is straightforward and can be done offline, bit mapping for SM is challenging due to its hybrid nature, in which both APM and space are used creating a more complicated constellation structure. Heuristic bit mapping schemes for SM have been recently proposed [33, 34, 53, 54]. In [33], the authors proposed a bit mapping algorithm, called original binary switch algorithm (OBSA), for SSK in independent Rayleigh fading channel using full channel state information at the transmitter (CSIT). OBSA initially assigns a natural bit
mapping for symbols. Then, for every combination of two symbols, it switches the bit sequences assigned to these two symbols. At every switching, it examines the BER of the resulting bit mapping and then chooses the bit mapping with the best performance. OBSA achieves performance close to the lower bound of the BER at high SNRs. However, a major issue with OBSA is its computational complexity, which is of order $O(K^4)$, where $K$ is the number of symbols. In [34], the authors proposed a bit mapping method for SM assuming the knowledge of the channel distribution at transmitter. This method applies Gray coding to the APM part, and a random bit mapping for the space part. This scheme provides only a slight improvement of less than 1 dB and only when the APM constellation size is less than eight. In [53], the authors proposed a bit-mapper, called brute forth mapper (BFM), for SM. BFM finds the two SM symbols with minimum distance, and assigns bit combinations that are different in only one bit for these two symbols. For the rest of the symbols, BFM uses natural mapping for the space part and Gray coding for the APM part. BFM achieves a performance close to the lower bound of the BER at high SNRs with low complexity order of $O(K^2)$ and low feedback overhead of $2 \log_2 K$ bits. However, a major issue with BFM is that it provides little to no improvement at low SNRs because it depends on minimizing the Hamming distance between the two SM symbols with minimum distance which dominates the BER only at very high SNRs. In [54], the authors proposed a bit mapping scheme for SM with one receive antenna, in which the sorting order of the channel magnitudes and the multiplicity of the corresponding phases are fed back to the transmitter. This method uses a Gray coded bit mapping.
for the space part based on the sorting order of the channel magnitudes. For the APM part, it uses a Gray coding based on the sorting order of the multiplicity of the phases. The method in [54], however, provides less improvement compared to OBSA in [33].

In this chapter, we propose a novel low-complexity adaptive bit mapping scheme for SM called nearest point method (NPM). The scheme first identifies a certain sorting order and then applies a Gray coding for SM symbols sorted according to that order. NPM uses a greedy approach to identify the sorting order. It starts with the pair of SM symbols of minimum distance; then, from one of the SM symbols in that pair it goes to the next SM symbol with minimum distance, and it continues to do that until it visits all the SM symbols without visiting the same SM symbol more than once. It is shown that NPM performs similar to OBSA in a Rayleigh fading environment, and, at high SNRs, its BER performance is close to the lower bound of the BER for both SM and SSK. However, the complexity of the proposed NPM is only of order $\mathcal{O}(K^2)$, which is significantly lower than that of the OBSA ($\mathcal{O}(K^4)$). The feedback overhead of NPM is $(K - 1)\log_2 K$ bits, which is slightly larger than that of BFM ($2\log_2 K$ bits). However, NPM makes up for it by providing better BER performance compared to BFM and the method in [54] as well. We also examine NPM for SSK when only the spatial correlation matrix is available at the transmitter, and we show that it can also provide some performance improvements in this case.

The rest of the chapter is organized as follows. The system model is described
in Section 2.2, and the optimization problem for bit mapping design is formulated in Section 2.3. Section 2.4 describes the design of our proposed NPM. Numerical results and discussions are presented in Section 2.5.

2.2 System Model

We consider an $N_r \times N_t$ MIMO system with SM transmission, where $N_t$ and $N_r$ are the number of transmit and receive antennas, respectively. Then the total number of SM symbols is $K = MN_t$, where $M$ is the APM size. Therefore, $k = \log_2(N_t) + \log_2(M)$ bits are used to select an APM symbol and one of the transmit antennas according to a certain bit mapping scheme, assuming that $N_t$ and $M$ are powers of two (see Fig. 2.1). The received signal vector can be written as

$$y = \sqrt{\rho}Hx_{sm} + n$$  \hspace{1cm} (2.1)
where $H$ is the $N_r \times N_t$ channel matrix, $\rho$ is the signal-to-noise ratio (SNR), $n$ is the $N_r \times 1$ noise vector whose entries are i.i.d. complex Gaussian random variables with zero mean and unit variance, $s_m$ is the $m$th the APM symbol, and $x$ is the $N_t \times 1$ transmit vector with only one non-zero entry. For detecting the signal at receiver, the maximum likelihood (ML) detection can be formulated as [4]

$$\hat{(n, m)} = \arg \min_{(n, m)} \|y - \sqrt{\rho} h_n s_m \| \quad (2.2)$$

where $h_n$ is is the $n$th column of the channel matrix $H$, $\hat{n}$ is the estimated antenna index, and $\hat{m}$ is the estimated APM symbol index.

We consider a correlated Rayleigh Fading channel model such that

$$H = \tilde{H} T^{1/2} \quad (2.3)$$

where $T$ is is the $N_t \times N_t$ transmit spatial correlation matrix satisfying $\text{trace}\{T\} = N_t$, and $\tilde{H}$ is an $N_r \times N_t$ matrix whose elements are i.i.d complex Gaussian random variables with zero mean and unit variance. To model the correlation matrix, we consider a uniform linear array (ULA) spatial correlation model, where the normalized correlation coefficient between the $n$th and $n'$th transmit antennas $T_{n,n'}$ can be written as

$$T_{n,n'} = \frac{e^{j wd(n-n') \sin \phi_0}}{1 + \frac{\sigma_\phi^2}{2}[wd(n-n') \cos \phi_0]^2} \quad (2.4)$$

where $w$ is the wavenumber, $d$ is the antenna separation, i.e. the distance between
two adjacent transmit antennas, $\phi_0$ is the mean angle of arrival (AOA), and $\sigma_\phi$ is the standard deviation of the power azimuth spectrum (PAS) [55].

2.3 Problem Formulation

The upper bound on the BER $P_e^{(r)}$ of SM given a certain bit mapping $\Gamma$ can be written as [8], [13]

$$P_e^{(r)}(\Gamma) \leq \frac{1}{K} \frac{1}{\log_2(K)} \sum_{k=1,k'=1}^{K} D_\Gamma(k, k') P^{(r)}(k, k')$$

(2.5)

where $D_\Gamma(k, k')$ is the hamming distance between the bit assignment of the $k$th SM symbol and the $k'$th SM symbol given the bit mapping scheme $\Gamma$, $P^{(r)}(k, k')$ is the pair-wise error probability between the $k$th SM symbol and the $k'$th SM symbol, and $r$ indicates whether the expression is instantaneous or average, where $r = i$ represents instantaneous, and $r = a$ represents average. Our objective is to find a bit mapping that minimizes the above upper bound.

For the case of full CSIT, we use the instantaneous pair-wise error probability $P^{(i)}(k, k')$, which can be written as

$$P^{(i)}(k, k') = P^{(i)}((n, m), (n', m')) = Q(\sqrt{\frac{P}{2}} \| s_m h_n - s_m h_{n'} \|)$$

(2.6)

If only the channel distribution and $\mathbf{T}$ are known at the transmitter, we use the
average pair-wise error probability $P^{(a)}(k, k')$ that can be written as

$$P^{(a)}(k, k') = P^{(a)}((n, m), (n', m')) = \|s_m t_n - s_{m'} t_{n'}\|^{-2N_r} \quad (2.7)$$

where $t_n$ is the $n$th column of the matrix $T^{1/2}$. The bit mapping optimization problem can then be written as

$$\Gamma^{(r)}_{opt} = \arg \min_{\Gamma} P_e^{(r)}(\Gamma) \quad (2.8)$$

where $\Gamma^{(r)}_{opt}$ is the optimal bit mapping. We define $C^{(r)}(k, k')$ as the distance metric between the $k$th and the $k'$th SM symbols, where $C^{(i)}(k, k') = \|s_m h_n - s_{m'} h_{n'}\|$ and $C^{(a)}(k, k') = \|s_m t_n - s_{m'} t_{n'}\|^{2N_r}$. We also define a symmetric $K \times K$ matrix, $C^{(r)}$, whose entries are the distance metrics. The optimal bit mapping $\Gamma^{(r)}_{opt}$ can be found by exhaustively searching all possible bit mappings, which is computationally complicated. To do the exhaustive search for $K$ symbols, one needs to search over $(K - 1!)/\log_2(K)!$ different bit mappings with unique sets of hamming distances. This is a very large number even for small values of $K$ (e.g. $(16 - 1!)/\log_2(16)! = 5.44 \times 10^{10}$ different bit mappings). Therefore, for practical reasons, a heuristic approach is needed. Due to the lack of constellation structure in SM and its dynamic nature, finding a suitable heuristic bit mapping scheme is challenging in this case.
2.4 Nearest Point Method for Bit Mapping

In this section, we introduce our proposed scheme called NPM as an adaptive bit mapping scheme for SM. A detailed description of the method, the algorithm, an illustrative example, and the computational complexity and a bound on the performance improvement of the proposed scheme are provided in this section.

2.4.1 NPM Algorithm

In the first step of the NPM, the SM symbols are sorted according to a certain sorting order. To identify the sorting order, the NPM finds the pair of SM symbols with the minimum distance. Then, it randomly considers one of them as the first SM symbol and the other as the second SM symbol. It then finds the next SM symbol with the smallest distance to the second SM symbol and sorts it as the third SM symbol. The algorithm continues to sort the rest of the symbols by finding the symbols with the minimum distance to the last SM symbol, excluding those that are already sorted. The sorting is complete when the algorithm visits all the SM symbols without visiting any SM symbol more than once. The second step of the NPM is assigning a Gray coded bit mapping to the SM symbols sorted according to the sorting order found in the previous step as if they are on a line or a circle, similar to Gray coding for PSK symbols. The details of the NPM algorithm are summarized in Algorithm 1.

The BER improvement is due to the fact that the NPM minimizes the Hamming distance between the closest pair of SM symbols, as this is also the case in BFM. However, in addition to that, NPM assigns bit combinations that are different
Algorithm 1 Nearest Point Method

Input: $H$ and $S = \{s_1, s_2, s_3, \ldots, s_M\}$;
Output: $\Gamma_{NPM}$ (bit mappint)

1: $C :=$ all-zeros $K \times K$ matrix.
2: for $k = 1 : 1 : K$ do
3:   for $k' = k : 1 : K$ do
4:     if $k == k'$ then
5:       $C(k, k') = \infty$
6:     else
7:       Calculate $C(k, k')$
8:       $C(k', k) = C(k, k')$
9:     end if
10: end for
11: end for
12: $v :=$ all-zero $1 \times K$ row vector.
13: $v(1) :=$ the column index of the minimum element of the matrix $C$.
14: $v(2) :=$ the row index of the minimum element of the matrix $C$.
15: for $i = 3 : 1 : K$ do
16:   $C(v(i - 1), v(1 : 1 : i - 2)) = \infty$
17:   $y = C(v(i - 1), :)$
18:   $v(i) :=$ the index of the minimum element of $y$.
19: end for

from that assigned to a symbol in only one bit for the nearest two neighbors of that symbol, providing additional BER improvement at low SNRs compared to BFM.

2.4.2 Example

Suppose we have $K = 4$ SM symbols, with the following arbitrary distance matrix

\[
C = \begin{bmatrix}
\infty & 5 & 0.7 & 3.3 \\
5 & \infty & 2 & 1 \\
0.7 & 2 & \infty & 0.8 \\
3.3 & 1 & 0.8 & \infty \\
\end{bmatrix}
\] (2.9)
The minimum distance is 0.7, and it is between the SM symbol #1 and SM symbol #3. So, these two SM symbols #1 are sorted as the first and second symbols. Supposed that symbol #1 is sorted as the first symbol. Then, the symbol with the minimum distance to SM symbol #3 is found which is SM symbol #4, with the distance 0.8. Finally, the SM symbol with the minimum distance to SM symbol #4 from the remaining SM symbol(s) is SM symbol #2. Therefore, we will have the following order vector \( v = (1 \ 3 \ 4 \ 2) \). Now the algorithm will assign the Gray coded bit assignment 00, 01, 11, 10 to the SM symbols sorted in \( v \). For instance, 00 is assigned to symbol #1, 01 is assigned to symbol #3, 11 is assigned to symbol #4, and 10 is assigned to symbol #2.

2.4.3 Computational Complexity and Feedback Overhead

In the first step of the algorithm, \( K(K - 1)/2 \) distance metrics should be calculated. The algorithm also calculates the values of \( s_m \forall m \in \{1, 2, 3, \ldots, M\}, \forall n \in \{1, 2, 3, \ldots, N_t\} \) and stores them. Also, it should calculate \( K \) minimum functions. Therefore, the total number of required calculations \( n_{total} \) can be written as

\[
n_{total} = K(K - 1)/2 \text{ distance metric calculations} \\
+ K \ s_m \text{ calculations} \\
+ K \text{ minimum function calculations}
\]
For each cost function calculation, we need $4N_r - 1$ real additions, and $2N_r$ real multiplications. For each $s_mz_n$ calculation, we need $2N_r$ real additions and $4N_r$ real multiplications; and for each minimum function calculation, we need $K$ comparisons. The total number of required calculations $n_{total}$ can be written as

$$n_{total} = 2N_rK^2 - (K^2 - K)/2 \text{ real additions}$$

$$+ N_r(K^2 + 3K) \text{ real multiplications}$$

$$+ K^2 \text{ comparisons} \tag{2.11}$$

Assuming $K$ is much larger than $N_r$, the complexity order of NPM is in the order of $\mathcal{O}(K^2)$. Therefore, this algorithm is significantly less complicated than the OBSA algorithm proposed in [33], which its complexity if of order $\mathcal{O}(K^4)$, and yet NPM, as we will see later, achieves the same performance improvement in the presence of full CSIT.

The computations of NPM can be done at the receiver, and the sorting order is fed back to the transmitter by sending the indices of $K - 1$ symbols (the index of the last symbol will be known when having the indices of the rest of the symbols in the sorting order). Therefore, NPM needs $(K - 1)\log_2 K$ bits for feedback. NPM makes up for the increased feedback overhead compared to BFM’s overhead of $2\log_2 K$ bits by providing a better BER performance. On the other hand, both NPM and BFM have much lower feedback overhead compared to OBSA ($\lceil \log_2 K! \rceil$ bits).
2.4.4 Performance Bound

Let $P_s$ be the symbol error probability, and $k = \log_2(K)$ be the number of bits per symbol. The lowest possible BER is $P_s/k$, in which case there is only one bit error for every symbol error. So, the bit error probability $P_e$ can be lower bounded as

$$\frac{P_s}{k} \leq P_e$$  \hspace{1cm} (2.12)

To analyze the performance improvement due to bit mapping, we use the average BER of a bit mapping scheme $\Gamma_c$, that applies Gray coding for bit allocation of the APM part, and random bit mapping for the space part as an upper benchmark and compare it to the lower bound of the average BER, i.e. $\bar{P}_s/k$, where $\bar{P}_s$ is the average symbol error rate.

Let $\bar{x}$ be the average number of bits in error when a symbol error occurs. We can write the average BER of benchmark scheme, $\Gamma_c$, as follows

$$\bar{P}_e(\Gamma_c) = \frac{\bar{x}}{k} \bar{P}_s$$  \hspace{1cm} (2.13)

Then, the difference of the lower bound in (2.12) with the performance of $\Gamma_c$ at a high SNR can be written in dB as

$$I = \frac{10 \log_{10}(\bar{P}_e(\Gamma_c)/(P_s/k))}{N_r} = \frac{10 \log_{10}(\bar{x})}{N_r}$$  \hspace{1cm} (2.14)
For SM, $\bar{x}$ is given by equation (16) in [53]. If SSK is used, we can write $\bar{x}$ as [33]

$$\bar{x} = E[x] = \frac{\sum_{l=1}^{k} l \binom{k}{l}}{2^k - 1} = \frac{k 2^{k-1}}{2^k - 1} \quad (2.15)$$

We consider $I$ in (2.14) as the upper bound of the performance gain of a bit mapping scheme. A good bit mapping scheme provides a performance improvement close to this bound.

2.5 Simulation Results and Discussions

Throughout simulations, we assume uncorrelated or correlated Rayleigh fading channels with uniform linear array (ULA) spatial correlation model in [55]. We consider $\phi_0 = 0^\circ$ and $\sigma_\phi = 50^\circ$ for the ULA with half a wavelength antenna separation. We compare the BER performance of $\Gamma_c$, NPM, OBSA, BFM, the scheme in [54], and the lower bound of the BER in the case of SM and SSK with full CSIT. In the case of SSK with only the knowledge of the channel distribution and the spatial correlation matrix at the transmitter, we compare the BER performance of $\Gamma_c$, NPM, OBSA when a rate-3/5 convolutional code is used. Note that BFM and the scheme in [54] are not considered because they require full CSIT. We also simulate the performance of bit mapping schemes for SSK with full CSIT when we apply an adaptive modulation scheme called modified space shift keying (MSSK) [9].
2.5.1 Full CSIT

For full CSIT, we simulate the BER for both SM and SSK over an uncorrelated channel. We consider 4-QAM modulation ($M = 4$) and $N_t = 8$ transmit antennas and $N_r = 1$ and $N_r = 2$ receive antennas. The BER performances of SM and SSK are shown in Fig. 2.2 and Fig. 2.3, respectively. We consider $N_t = 32$ transmit antennas.

We consider both $N_r = 1$ and $N_r = 2$ receive antennas (for the scheme in [54], only one receive antenna is considered).

For both SM and SSK in full CSIT, we see that NPM and OBSA provide BER performances close to the lower bound at high SNRs (4 dB for $N_r = 1$ and 2 dB for $N_r = 2$ in this case). These performance improvements are close to the maximum
Figure 2.3: BER performance of SSK ($N_t = 32, N_r = 1, 2$) for different bit mapping schemes using full CSIT.
possible improvement $I$ in (2.14), which equals 4.11 dB and 2.05 dB for $N_r = 1$ and $N_r = 2$, respectively. This confirms that NPM and OBSA perform close to optimal when having full CSIT, with the advantage that NPM has a lower complexity of order $O(K^2)$. Note that both NPM and OBSA outperform BFM and the scheme in [54] in all cases. In Fig. 3.8, we simulate the BER performance of NPM and $\Gamma_c$ mapping for SM with $K = 128$. We simulate two cases with $N_r = 1$: one with $M = 32$ and $N_t = 4$ and the other one with $M = 2$ and $N_t = 64$ to show the effect of the number of transmit antennas on the performance. As seen from Fig. 3.8, the performance improvement of NPM over the generic bit mapping, $\Gamma_c$, increases by about 1dB by increasing the number of transmit antennas showing that NPM is indeed effective in assigning bits in the space domain. Note that the performance increase from $M = 32$ to $M = 2$ (BPSK) is due to a more favorable constellation breakdown in the latter case [8].

2.5.2 Partial CSIT

When only the spatial correlation matrix is known at the transmitter, we simulate the BER of SSK (NPM and OBSA do not provide significant performance improvement for SM in this case). We consider $N_t = 8$ transmit antennas and $N_r = 1$ and 2 receive antennas. To see the performance of a coded system, we also simulate the BER of when a rate-3/5 convolutional code is used. Fig. 2.5 shows the BER performance of the uncoded and coded SSK systems. We notice that, similar to the case of full CSIT, NPM and OBSA provide similar BER improvements. While they are not as effective
Figure 2.4: BER performance of SM ($K = 128$, $N_r = 1$) for $\Gamma_c$ and NPM using full CSIT for different number of transmit antennas.
as the case of full CSIT, NPM and OBSA provide BER improvement in addition to the coding gain provided by the convolutional code.

2.5.3 Combining NPM with MSSK

To further improve the performance using full CSIT, it is possible to combine bit mapping with adaptive modulations and power allocation. In this part, we evaluate the performance of NPM, OBSA, BFM, the scheme in [54] and the lower bound of the BER for SSK when combined with an adaptive modulation and power allocation scheme proposed in [9] called MSSK. We consider $N_t = 16$ and $N_r = 1$ and $N_r = 2$ and an uncorrelated channel. The BER performance results are shown in Fig. 2.6. Both NPM and OBSA provide BER performances close to the lower bound at high
Figure 2.6: BER performance of MSSK \((N_t = 16, N_r = 1, 2)\) for different bit mapping schemes using full CSIT.

SNRs in addition to the gain achieved by adaptive modulation and power allocation, and they outperform both BFM and the scheme in [54]. The results in Fig. 2.6 show that bit mapping can be used to further improve the performance in addition to the link adaptive techniques when full CSIT is available.
3.1 Introduction

Several works in literature have focused on combining TCM with SM and SSK in order to achieve additional performance improvement [56–59]. In [56], the authors proposed a scheme called trellis coded spatial modulation (TCSM) for a spatially correlated MIMO channel. TCSM is similar to SM except that the bits that select the transmit antenna are encoded using a convolutional encoder. In other words, in this scheme, TCM is applied to the space domain while keeping the APM domain uncoded. The scheme divides the transmit antennas into subsets and then maximizes the distance in each subset, which is similar to the design of TCM in AWGN channels. TCSM provides BER performance improvement compared to Vertical-Bell Laboratories Layered Space-Time (V-BLAST) and Alamouti schemes in correlated Rician fading channels. However, this scheme does not provide any performance improvement in uncorrelated fading channels. This is because the code design metric is the same to the one used for TCM is AWGN channels, which is not necessarily suitable for combined TCM with SM or SSK in fading environment.

In [57], the authors proposed another scheme that combines TCM and SM
called spatial modulation with trellis coding (SM-TC). Unlike TCSM, which applies TCM only to space part of SM, in SM-TC, the output of a convolutional encoder is used as input to the SM modulator where the bit are divided into space and APM parts. To get a better code design criteria than TCSM, the upper bound of the unconditional pairwise error probability (UPEP) was derived for SM-TC in uncorrelated quasi-static Rayleigh fading channels. Two design criteria were proposed based on the resulting UPEP upper bound. The degree of freedom (DOF) was defined as the number of unique antenna indices used among the length of a pairwise error event. Assuming that the length of the pairwise error events is equal to the effective length of the convolutional code \( L \), the authors proposed a diversity gain criterion which states that to achieve a diversity order of \( L \), \( DOF \geq L \). The authors also proposed a coding gain criterion that is based on optimizing a certain matrix in the UPEP expression. The upper bound of UPEP was derived for special cases of \( L = 2 \) and \( L = 3 \) using four transmit antennas and M-ary PSK modulation. Whether the symbols in the pairwise error event are transmitted from the same or different antennas affects the resulting UPEP expression. Therefore, the UPEP upper bound was derived for all possible cases of error. Furthermore, based on the two design criteria, octal generator matrices were provided in [57] for convolutional codes of rates 2/4, 3/6, 4/6 using 4, 8, and 16 states. Finally, using simulation results, it was shown that SM-TC outperforms space-time trellis coding (STTC) and coded V-BLAST schemes.

A similar scheme to SM-TC is proposed in [58] and called trellis coded space-
shift keying modulation (TC-SSKM) and combines TCM with SSK (as a special case of SM). Simulation results show that TC-SSK outperforms uncoded SSK. In [59], convolutionally coded SSK modulation in quasi-static channels is further investigated. Given that the conditional bit error probability of the convolutionally coded SSK is upper bounded by the scalar multidimensional transfer function of the convolutional code, the authors showed techniques that can be employed to derive the error transfer function of the code. Then it was shown, with simulation, that the shortest distance error event in the scalar transfer function can be used to predict the diversity order of the system accurately.

In this chapter, we investigate trellis coded space-time shift keying (TC-STSK), which combines TCM with SSK in quasi-static fading channels, which is a type of space-time shift keying (STSK) [60] with only one active antenna at a time. For designing such a scheme, the diversity gain criterion proposed in [57] is not suitable. For example, while it is correct that for a pairwise error event of length \( L \) with diversity order \( L \), \( DOF \geq L \), the opposite is not true. In other words, having \( DOF \geq L \) for a pairwise error event of length \( L \) does not necessarily mean that the diversity order is \( L \). As we will see later in Section 3.2, because combining TCM and SSK for block fading channels results in a type of STSK, a better diversity gain criterion is the rank criterion used in STSK systems [60].

A major difference between TC-STSK and STSK is that, in TC-STSK, a pairwise error event happens only when two output codewords start from the same state and end in the same state, and this is due to the nature of convolutional coding.
On the other hand, in STSK, a pairwise error event can happen between any pair of space-time matrices. This results in a more relaxed diversity condition in TC-STSK in the sense that to select the output codewords that achieve a certain diversity in TC-STSK, one needs to examine only the mutual pairwise error diversity between codewords that start from the same state and end in the same state. This is different than STSK, in which one needs to examine the mutual pairwise error diversity between all pairs of space-time matrices.

We analyze the rank criterion of the TC-STSK using a graph theoretic approach, and accordingly, we introduce the concept of maximizing the number of unique columns in the code difference matrix of pairwise codewords as a code design goal for the TC-STSK scheme. We discuss the design principles for diversity advantage, and propose and analyze a code design algorithm that is mostly suitable for rate-1/n codes. To generalize to the case of rate-k/n codes, we also propose a diversity adaptation mechanism for the TC-STSK, such that for a given rate-k/n code, we select only the codewords that achieve diversity larger or equal to the desired diversity order. Such a scheme provides a tradeoff between diversity gain and transmission rate. We use the diversity adaptation mechanism to achieve full diversity advantage, and provide the encoder and decoder design with a modified version of the soft Viterbi algorithm that extracts the full diversity advantage offered by the proposed TC-STSK scheme.

The rest of the chapter is organized as follows. The system model is described in Section 3.2, and the code design criteria are discussed in Section 3.3. The proposed
design algorithm is described and analyzed in Section 3.4. The diversity adaptation mechanism with encoder and decoder designs is described in Section 3.5. Simulation results and discussions are included in Section 3.6.

3.2 System Model

We consider a TC-STSK MIMO system with \( N_t \) transmit antennas and \( N_r \) receive antennas as shown in Fig 3.1 where a convolutional encoder of rate \( R = k/n \) where \( n = \log_2(N_t) \) is used. At the transmit side, an input sequence of length \( k \) bits enters the encoder at each time. Assuming a quasi-static channel, \( T \) is the number of time intervals in which the channel remains the same. After \( T \) time intervals, the convolutional encoder outputs a coded bit sequence of length \( nT \) bits. An SSK modulator is then used to convert the \( nT \) bits into a sequence of transmit antenna indices \( z = (z_0, z_1, \ldots, z_{T-1}) \) where \( z_m \in \{0, \ldots, N_t - 1\} \) for \( m = 0, \ldots, T - 1 \) is the index of the active antenna at \( m \)th transmit interval. We consider an \( N_t \times T \) transmission matrix \( X = [x_0 x_1 \ldots x_{T-1}] \) where \( x_m, m = 0, \ldots, T - 1 \), is an \( N_t \times 1 \), where the entries of its \( m \)th column are all zeros except for the \((z_m + 1)\)th element which is one.

We consider an uncorrelated independent Rayleigh fading channel with the \( N_r \times N_t \) MIMO channel \( H \), whose elements are i.i.d complex Gaussian random variables with zero mean and unit variance. We consider a block fading assumption, i.e. \( H \) is fixed during a block of \( T \) transmission intervals and the channel matrix changes independently from one block to another. Under these assumptions, the system model
can be written as

\[ Y = \sqrt{\rho} H X + N \]  \hspace{1cm} (3.1)

where \( \rho \) is the signal to noise ratio (SNR), \( N \) is the \( N_r \times T \) noise matrix whose elements are i.i.d complex Gaussian random variables with zero mean and unit variance, and \( Y \) is the \( N_r \times T \) receive matrix. At the receive side, a soft decoder uses \( Y \) as a soft input and jointly demodulates and decodes the received signal.

A pairwise error event of length \( T \) occurs when the decoder decides in favor of \( \hat{X} \) when \( X \) is transmitted. The conditional pairwise error probability (CPEP) can be written as

\[ Pr(X \rightarrow \hat{X}|H) = Pr(\|Y - \sqrt{\rho} H \hat{X}\|_F^2 > \|Y - \sqrt{\rho} H X\|_F^2) \]  \hspace{1cm} (3.2)
The CPEP can then be simplified as [61]

$$Pr(X \rightarrow \hat{X}|H) = Q(\sqrt{\frac{\rho\|H(X - \hat{X})\|^2}{2}})$$  \hspace{1cm} (3.3)

Using the Chernoff bound, i.e. $Q(x) \leq \frac{1}{2} e^{-\frac{x^2}{2}}$, the CPEP can be upper bounded by

$$Pr(X \rightarrow \hat{X}|H) \leq \frac{1}{2} e^{\rho\|H(X - \hat{X})\|^2}$$  \hspace{1cm} (3.4)

The UPEP can be found by averaging over the channel matrix $H$. For uncorrelated quasi-static Rayleigh fading channel, the UPEP can be written as [57,58,60]

$$Pr(X \rightarrow \hat{X}) \leq \frac{1}{2} \frac{1}{|\rho A \otimes I_{N_r} + I_{N_r} N_r|}$$  \hspace{1cm} (3.5)

where $A = (X - \hat{X})(X - \hat{X})^H$, and $I_K$ is the identity matrix of size $K$. For high SNRs, UPEP can be simplified to [62]

$$Pr(X \rightarrow \hat{X}) \leq \frac{1}{2(\rho^d N_r)} \frac{1}{\prod_{i=1}^d \lambda_i N_r}$$  \hspace{1cm} (3.6)

where $d = \text{rank}(A)$ and $\lambda_i$ is the $i$th non-zeros eigenvalue of $A$. Let $D = X - \hat{X}$ be the code difference matrix. Using the rank property $\text{rank}(A) = \text{rank}(DD^H) = \text{rank}(D)$, $d$ can be written as

$$d = \text{rank}(D) = \text{rank}(X - \hat{X})$$  \hspace{1cm} (3.7)
(3.7) shows that the diversity gain criterion for TC-STSK is indeed the rank criterion used in STSK design. It is well known that in the general case of STSK in which more than one antenna are active at a time, the maximum achievable diversity \(d_{\text{max}}\) is \(\min(T, N_t)\) [60]. It is shown in [63], however, that the maximum achievable diversity \(d_{\text{max}}\) for STSK systems with only one active antenna at a time is \(\min(T, N_t - 1)\). [63] shows that if only one antenna is activated at a time, then it is not possible to have more than \(N_t - 1\) linearly independent columns in the code difference matrix which results in \(d_{\text{max}} \leq \min(T, N_t - 1)\).

We, here, provide an alternative explanation of the result in [63] using some proven results from graph theory. Since the code difference matrix \(D\) is an \(N_t \times T\) matrix in which each column contains one 1, one -1, and \(N_t - 2\) zeros, matrix \(D\) represents an incidence matrix for a directed graph \(G\) with \(V = N_t\) vertices and \(E = T\) edges; Each column in the incidence matrix specifies an edge that connects the two vertices associated with 1 and -1 entries. However, because switching these two entries in a column does not change the rank of the matrix, we can model the graph described by incidence matrix \(D\) as an undirected graph.

**Example 3.2.1.** Let \(N_t = 4\) and \(T = 4\). Suppose we have two codewords with
transmit antenna indices $\mathbf{x} = (0 \ 1 \ 2 \ 1)$ and $\hat{\mathbf{x}} = (1 \ 2 \ 3 \ 3)$. Then, we have

$$D = \begin{bmatrix}
1 & 0 & 0 & 0 \\
-1 & 1 & 0 & 1 \\
0 & -1 & 1 & 0 \\
0 & 0 & -1 & -1
\end{bmatrix} \quad (3.8)$$

Fig. 3.2 shows a graphical representation of the incidence matrix $D$ with $V = N_t = 4$ vertices and $E = T = 4$ edges.

Our aim is to study the properties of the rank of the incidence matrix $D$. To do this, the following Lemma 3.2.2 and Theorem 3.2.3 from [64] are useful.

**Lemma 3.2.2.** Let $G$ be a graph on $N_t$ vertices and let $D(G)$ be the incidence matrix of $G$. Then $\text{rank}(D(G)) \leq N_t - 1$ with equality if and only if $G$ is connected.

**Theorem 3.2.3.** If $G$ is a graph with $N_t$ vertices and has $k$ connected components, then $\text{rank}(D(G)) = N_t - k$.

Lemma 3.2.2 shows that the maximum rank of the incidence matrix of a graph with $N_t$ vertices is $N_t - 1$, and is only achievable when the incidence matrix forms
a connected graph on the $N_t$ vertices. A graph is connected when there is a path between every pair of vertices. Theorem 3.2.3 shows that the rank of the incidence matrix is, in fact, $N_t - k$, where $k$ is the number of connected components. The maximum rank of $N_t - 1$ is a special case of Theorem 3.2.3 with only one connected component since all the $N_t$ vertices of the graph are connected.

**Definition 3.2.4.** The set of unique column differences in the incidence matrix $D$ (unique edges of the graph), is the set of columns in which each column connects two different vertices than any other column in the set.

**Example 3.2.5.** Let the transmitted antenna sequences of two pairwise codewords be $x = (0 \ 2 \ 1 \ 3)$ and $\hat{x} = (1 \ 3 \ 0 \ 1)$. The set of unique edges is $(0,1), (2,3), (3,1)$. The number of unique differences is three instead of four because there are two edges that connect the same 0 and 1 vertices.

**Lemma 3.2.6.** If $G$ is a graph with $N_t$ vertices, the number of unique edges $N_u$ (the number of unique columns in $D(G)$) for having rank($D(G)$) = $d$, where $d \leq N_t - 1$, is bounded by

$$d \leq N_u \leq \binom{d}{2} + 1 \quad (3.9)$$

**Proof.** According to Theorem 3.2.3, we need to have exactly $k = N_t - d$ connected components to have rank($D(G)$) = $d$. The total number of $N_t$ vertices is divided into $k$ disjoint sets, such that $N_t = \sum_{i=1}^{k} N_i$, where $N_i$ is the number of vertices in the $i$th set. Let $N_{ui}$ be the number of unique edges needed to connect the vertices in the $i$th
set $i = 1, 2, \ldots, k$. $N_u_i$ is given by

$$
\begin{cases}
N_u_i = 0 & \text{if } N_i = 1 \\
N_i - 1 \leq N_u_i \leq \binom{N_i - 1}{2} + 1 & \text{if } N_i \geq 2
\end{cases}
$$

(3.10)

Let $d_i = N_i - 1$, then $\sum_{i=1}^{k} d_i = N_t - k = d$. The total number of unique edges needed to get diversity $d$ is the sum of the number of edges that needs to connect each component, i.e. $N_u = \sum_{i=1}^{k} N_u_i$. We have

$$
N_u = \sum_{i=1}^{k} N_u_i \geq \sum_{i=1}^{k} d_i = d
$$

(3.11)

This proves that $N_u \geq d$. For the upper bound, we have

$$
N_u = \sum_{i=1}^{k} N_u_i \leq \sum_{i=1}^{k'} \left( \binom{d_i}{2} + 1 \right)
$$

(3.12)

where $k' \leq k$ is the number of sets in which there are more than one vertex. Since $d_i = 0$, if the $i$th set has one vertex, then $d = \sum_{i=1}^{k} d_i = \sum_{i=1}^{k'} d_i$. Now we prove the following inequality

$$
\sum_{i=1}^{k'} \left( \binom{d_i}{2} + 1 \right) \leq \left( \sum_{i=1}^{k'} d_i \right) + 1 = \left( \frac{d}{2} \right) + 1
$$

(3.13)

If $k' = 1$, then the inequality holds true. By induction, we show that if the inequality
is true for \( k' = m \), then it also holds true for \( k' = m + 1 \). We have

\[
\sum_{i=1}^{m+1} \left( \frac{d_i}{2} \right) + 1 \leq \left( \frac{\sum_{i=1}^{m+1} d_i}{2} \right) + 1
\]  

(3.14)

This can be written as

\[
\sum_{i=1}^{m} \left( \frac{d_i}{2} \right) + 1 + \frac{d_{m+1}}{2} + 1 \leq \left( \frac{\sum_{i=1}^{m} d_i + d_{m+1}}{2} \right) + 1
\]  

(3.15)

which is simplified to

\[
2 \sum_{i=1}^{m} \left( \frac{d_i}{2} \right) + 1 + d_{m+1}^2 - d_{m+1} + 2 \leq
\]

\[
\left( \sum_{i=1}^{m} d_i \right)^2 - \sum_{i=1}^{m} d_i + 2 + d_{m+1}^2 - d_{m+1} + 2d_{m+1} \sum_{i=1}^{m} d_i
\]  

(3.16)

This can be reduced to

\[
\sum_{i=1}^{m} \left( \frac{d_i}{2} \right) + 1 \leq \left( \frac{\sum_{i=1}^{m} d_i}{2} \right) + 1 + d_{m+1} \sum_{i=1}^{m} d_i
\]  

(3.17)

Since the summation is over the sets with more than one vertex, \( d_i (i = 1, 2, \ldots, k') \) is a positive integer. Therefore, \( 1 \leq d_{m+1} \sum_{i=1}^{m} d_i \), and this proves that inequality (3.13) holds true for \( k' = m + 1 \) if it is true for \( k' = m \). Since \( N_u \) is upper bounded by the left hand side of inequality (3.13), it is also upper bounded by the right hand side. This proves that \( N_u \leq \binom{d}{2} + 1 \).
We use the term pairwise codewords for a pair of codewords that start from the same state, and end in the same state. In the context of designing convolutional codes for TC-STSK, one should design the code such that the code maximizes the number of unique incidence matrix columns between pairwise codewords. This is because having unique edges would serve to increase the connectedness of the incidence matrix graph, and thus increasing its rank. We provide the guidelines to design such codes in the next section. In the following, we study each of these steps.

3.3 Design Principles of Convolutional Codes for TC-STSK

To design convolutional codes for diversity advantage that achieve one needs to maximize the number of unique differences between pairwise codewords. This design goal is achieved through the following steps: 1) Maximizing the branch distance; 2) Maximizing the effective length; and 3) Maximizing the number of unique differences.

3.3.1 Maximizing the branch distance

The branch distance is the minimum length in which two codewords diverge from a state and remerge at another state. In [51], the maximum branch distance $L_b$ for a rate-$k/n$ code is given by

$$L_b = \lfloor v/k \rfloor + 1 \quad (3.18)$$
In [65] and [66], it has been shown that the following trellis maximizes the branch distance

\[ S_m = (B S_{m-1} + b_{m-1}) \mod 2^v \]  

where \( S_m \) is the state at time \( m = 1, \ldots, L_b \), \( B = 2^k \) is the input size, and \( b_m \in \{0, \ldots, B-1\} \) is the input at time \( m \). Therefore, we use this trellis throughout the rest of this chapter.

### 3.3.2 Maximizing the effective length

The effective length of a convolutional code is the minimum number of output differences between two pairwise codewords. The effective length is less than or equal to the branch distance [51]. Therefore, the maximum effective length \( L_{max} \) for a rate-\( k/n \) code is equal to the maximum branch distance; i.e. \( L_{max} = L_b = \lfloor v/k \rfloor + 1 \). To get an understanding of how to maximize the effective length, we first need to briefly describe the encoding process. The encoder goes through the following states [65,66]

\[
S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow \ldots \rightarrow S_{L_b-1} \rightarrow S_{L_b} \tag{3.20}
\]

The encoder starts in \( S_0 \), takes the first input \( b_0 \), moves to \( S_1 \) and so on. The decoder, due to decoding errors, goes through a different sequence of states

\[
S'_0 \rightarrow S'_1 \rightarrow S'_2 \rightarrow \ldots \rightarrow S'_{L_b-1} \rightarrow S'_{L_b} \tag{3.21}
\]

We use the following definitions used in the code design for STTCs [65–67]
**Definition 3.3.1.** A level $m$ group is a collection of all destination states that can be reached at state transition $m$ from a given $S_0$ starting state through all possible $b_0, b_1, \ldots, b_{m-1}$ input sequences.

**Definition 3.3.2.** A subgroup of a level $m$ group is a collection of all destination states that can be reached at state transition $m$ from a given $S_0$ starting state and a given $b_0$ starting branch through all possible $b_1, b_2, \ldots, b_{m-1}$ input sequences.

It is shown in [65–67] that the above definitions have the following properties for $m = 1, \ldots, L_b - 1$:

1. Any level $m$ group starts at state $i$ such that $i \mod B^m = 0$ and consists of $B^m$ consecutive states.

2. Any subgroup of a level $m$ group starts at state $i$ such that $i \mod B^{m-1} = 0$ and consists of $B^{m-1}$ consecutive states.

3. Every level $m$ group consists of $B$ disjoint subgroups.

4. Because both the correct and the erroneous paths start from the same state $(S_0 = S'_0)$, $S_m$ and $S'_m$ belong to the same level $m$ group.

5. For some $j \neq l$ ($j, l \in \{0, 1, \ldots, B - 1\}$), $S_m$ belongs to the $j$th and $S'_m$ belongs to the $l$th subgroup of the same level $m$ group.

In a convolutional code state diagram, there are $B$ branches exiting each state, and each one of those branches is the output that corresponds to one of the $B$ possible inputs. In the TC-STSK, in which we have $N_t$ output symbols, the output
vector $\mathbf{M}(S_m)$ that represents the $B$ output symbols from a given state $S_m$ can be given by

$$
\mathbf{M}(S_m) = [z_0, z_1, \ldots, z_{B-1}]
$$

(3.22)

where $z_j \in \{0, \ldots, N_t - 1\}$ is the output symbol that corresponds to the input $j \in \{0, \ldots, B - 1\}$.

It is a common practice in convolutional code design that the output vectors that are assigned to the existing branches are disjoint [42]. Therefore, we divide the $N_t$ output symbols into $N_t/B$ disjoint output vectors of length $B$, and for each one of these disjoint vectors, we have $B$ permutations such that all the permutations have different output symbols in each of the $B$ elements of the output vector. Therefore, the total number of output vectors that can be used for a code is $N_t/B \times B = N_t$.

**Example 3.3.3.** Let $k = 1$ ($B = 2$) and $N_t = 8$. We have $N_t/B = 4$ disjoint vectors and let those vectors be $[0\ 1]$, $[2\ 3]$, $[4\ 5]$, $[6\ 7]$. For each vector we have $B = 2$ permutations. So the output vectors that can be assigned to states are $[0\ 1]$, $[2\ 3]$, $[4\ 5]$, $[6\ 7]$, $[1\ 0]$, $[3\ 2]$, $[5\ 4]$, $[7\ 6]$.

**Lemma 3.3.4.** The effective length of rate-$k/n$ code is maximized to $\lfloor v/k \rfloor + 1$ if and only if the output vectors that are assigned to $S_m$ and $S'_m$ are different for all level $m = 1, 2, \ldots, \lfloor v/k \rfloor$ groups.

**Proof.** From property 5, we know that $S_m$ and $S'_m$ are in different subgroups of the same level $m$ group. In fact, for some $j \neq l$ ($j, l \in \{0, 1, \ldots, B - 1\}$) and $i \in \{0, 1, \ldots, B^{m-1} - 1\}$, where $m$ is the level index, $S_m$ is the $i$th element of the $j$th...
subgroup and \(S'_m\) is \(i\)th element of the \(l\)th subgroup of the same level \(m\) group. If we assign the same output vector to \(S_m\) and \(S'_m\), then both the correct and the erroneous paths will have the same output symbol at time \(m\), and thus reducing the effective length. On the other hand, because different output vectors are either disjoint or they are different permutations that are different in each of the \(B\) elements of the output vector, if \(S_m\) and \(S'_m\) are assigned different output vectors, the correct and the erroneous paths will have different output symbols at time \(m\). The number of grouping levels in definition 3.3.1 is \(L_n - 1 = \lfloor v/k \rfloor\). If \(S_m\) and \(S'_m\) has different output vectors for \(m = 1, 2, \ldots, \lfloor v/k \rfloor\), then the correct and the erroneous paths will have different output symbols for \(\lfloor v/k \rfloor\) time intervals. At time interval \(m = 0\), the correct and the erroneous paths will have different output symbols because the entries of an output vector are different. Therefore, in this case, the effective length is \(\lfloor v/k \rfloor + 1\).

\[\square\]

**Example 3.3.5.** Consider a rate-1/n convolutional code with \(v = 3\) shift registers (memory) and \(k = 1\) input bits \((B = 2)\). There are \(2^v = 8\) states and \(\lfloor v/k \rfloor = 3\) grouping levels. The level \(m\) groups and subgroups are given in Table 3.1. In the table, the columns represent the grouping levels. A bracket represents a level \(m\) group. A bracket within a bracket represents a subgroup of a level \(m\) group.

In this example, we will track state 2 to see which states should have different output vectors according to Lemma 3.3.4.

1. At level 1, State 2 is the first element of the first subgroup of the second level 1
Table 3.1: Level $m$ groups and subgroups $m = 1, 2, 3$ for rate-$1/n$ code with $v = 3$ shift registers

<table>
<thead>
<tr>
<th>$m = 1$</th>
<th>$m = 2$</th>
<th>$m = 3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$[0]$</td>
<td>$[0]$</td>
<td>$[0]$</td>
</tr>
<tr>
<td>$[1]$</td>
<td>$[1]$</td>
<td>$[1]$</td>
</tr>
<tr>
<td>$[2]$</td>
<td>$[2]$</td>
<td>$[2]$</td>
</tr>
<tr>
<td>$[3]$</td>
<td>$[3]$</td>
<td>$[3]$</td>
</tr>
<tr>
<td>$[7]$</td>
<td>$[7]$</td>
<td>$[7]$</td>
</tr>
</tbody>
</table>

group. State 2 output vectors should be different from those of State 3 which is the first element of the second subgroup of the second level 1 group.

2. At level 2, State 2 is the first element of the second subgroup of the first level 2 group. State 2 output vectors should be different from those of State 0 which is the first element of the first subgroup of the first level 2 group.

3. At level 3, State 2 is the third element of the first subgroup of the first level 3 group. State 2 output vectors should be different from those of State 6 which is the third element of the second subgroup of the first level 3 group.

The condition of Lemma 3.3.4 to maximize the effective length can be represented by a graph on $2^v$ vertices, where there is an edge between two states that should not have the same output vector. Such a graph for our example is shown in Fig. 3.3. This can be considered as a coloring problem in graph theory. In other words, using graphical representation, we can consider the output vectors with different colors. To
maximize the effective length, the graph constructed according to Lemma 3.3.4 must be colored such that no two neighboring vertices have the same color.

3.3.3 Maximizing the number of unique differences

In this step, the goal of TC-STS code design is to maximize the number of unique differences between pairwise codewords. This is based on the rank of the incidence matrix analysis in Section 3.2. Both the correct and erroneous paths start at a state $S_0 \neq S'_0$ and visit the pair of states $S_m, S'_m$ for $m = 1, 2, \ldots, v$ respectively. We know that in order to maximize the effective length, the colors assigned to $S_m$ and $S'_m$ should be different for $m = 1, 2, \ldots v$. In order to maximize the number of unique differences, the pair of colors assigned to $S_m$ and $S'_m$ should be different at different levels. For example, if $v = 3$, the pairs of colors assigned to $S_1, S'_1, S_2, S'_2, S_3,$ and $S'_3$ should be different. An illustrative example for rate-$1/n$ code with $v = 3$ shift registers is given in Fig. 3.4.

Fig. 3.4 shows the connections between the pair of states at different level m
Figure 3.4: A graph that shows the connection between pairs of states for \( v = 3 \) and \( k = 1 \).

Now we provide the following lemma regarding the maximum number of unique differences for a rate-1/n code with \( v \) shift registers.

**Lemma 3.3.6.** The maximum number of unique differences for rate-1/n code with \( N_t = 2^n \) output symbols and \( v \) shift registers is given by

\[
N_{\text{max}}(v) = \min(v + 1, N_t - 1)
\]

*Proof.* For a rate-1/n code, each state has only one neighbor at each level. Therefore, the graph that represents the connection between pairs of states is a multipartite...
graph with \( v \) levels, and each level has \( 2^v/2 \) pairs of states. To color this multipartite graph such that no two neighbors have the same pair of colors, one needs to use different pairs of colors at each level. If only one pair of colors is used at a level, then we will use only two elements which makes the rest of the levels use the same pair of colors. Since we have a total of \( N_t \) colors, the total number of pairs of colors is \( \binom{N_t}{2} \). To utilize the pairs of colors such that we use all the \( \binom{N_t}{2} \) possible pairs, the set of pairs of colors that are assigned to a certain level should use all the \( N_t \) elements. At the same time, the number of pairs of colors that are used at a certain level should be minimized to maximize the number of unique levels. The minimum number of disjoint pairs of colors to assign to a certain level is \( N_t/2 \). To color the multipartite graph, we need to use disjoint sets of \( N_t/2 \) pairs of colors at different levels. Therefore, the maximum number of unique levels is \( \binom{N_t}{2}/(N_t/2) = N_t - 1 \).

There are \( N_t - 1 \) disjoint sets of \( N_t/2 \) disjoint pairs of colors. We can construct the sets of \( N_t/2 \) disjoint pairs of colors to have \( N_t - 2 \) disjoint sets in which in all the pairs of colors no color is paired with the permutation of itself (because \( B = 2^k = 2 \)). We have one set in which in all the pairs of colors a color is paired with the permutation of itself. Now we have the following two cases depending on \( v \):

1. If \( v \leq N_t - 2 \), we can get \( v \) unique levels in which in all the pairs of colors no color is paired with a permutation of itself. Therefore, we will have \( v + 1 \) unique differences out of which \( v \) differences are from the \( v \) unique levels and one unique difference from the colors themselves at \( m = 0 \).
2. If \( v \geq N_t - 1 \), we will have \( N_t - 1 \) unique levels, but we only have \( N_t - 1 \) unique differences because there is a level in which in all the pairs of colors a color is paired with the permutation of itself. This level will produce a pair of outputs that is the same as the pair of outputs produced by the colors themselves at \( m = 0 \).

Therefore, \( N_{\text{max}}(v) \) can be written as

\[
N_{\text{max}}(v) = \begin{cases} 
  v + 1 & \text{if } v \leq N_t - 2 \\
  N_t - 1 & \text{if } v \geq N_t - 1 
\end{cases}
\]  

(3.24)

Because \( v \) and \( N_t \) are integers, \( N_{\text{max}}(v) = \min(v + 1, N_t - 1) \).

3.4 Design Algorithm for TC-STSK codes

In this section, we present an algorithm to design rate-1/n convolutional codes for TC-STSK based on the design principles of Section 3.3. The algorithm takes the number of transmit antennas \( N_t \) and the number of shift registers \( v \) as inputs and produces the following outputs:

1. \( \mathbf{O} \) is an \( 2^v \times 2 \) trellis outputs matrix in which the \((i, j)\) element is the output (in decimal) from state \( i - 1 \) for the input \( j - 1 \).

2. \( \mathbf{S} \) is an \( 2^v \times 2 \) next states matrix in which the \((i, j)\) element is the next state (in decimal) from state \( i - 1 \) for the input \( j - 1 \).

The algorithm defines the following vectors and matrices:
1. \( \mathbf{C} \) is an \( N_t \times 2 \) outputs (colors) matrix that shows the \( N_t \) colors that we can assign to the states. The \((i, j)\) element represents the output (in decimal) of the \( i \)th color for the input \( j - 1 \).

2. \( \mathbf{P} \) is an \( N_t \times N_t \) color pairing matrix. The \((i, j)\) element \((i \neq j)\) represents the level index at which the colors with indices \( i \) and \( j \) can be paired.

3. \( \mathbf{a} \) is a \( 1 \times 2^r \) assigned color indices vector. The \( i \)th element represents the color index assigned to State \( i - 1 \).

4. \( \mathbf{l} \) is a \( 1 \times 2^r \) level indices vector. The \( i \)th element represents the level index in which State \( i - 1 \) has been assigned a color.

The algorithm initializes all the entries in all matrices except \( \mathbf{l} \) with zeros. The entries of \( \mathbf{l} \) are initially set to one. The algorithm assigns the color that has a zero output for a zero input to State 0.

3.4.1 Algorithm

First, the algorithm assigns the next states according to (3.19). Then, to create the color matrix \( \mathbf{C} \), the algorithm divides the \( N_t \) antenna elements into colors each containing a pair of elements. For each color, it also generates the permutation (switched version in this case of \( B = 2 \)) of it. Then the algorithm assigns indices to the colors in such a way that a color and its switched version are \( N_t/2 \) apart. Next, since the design goal is to have distinct pairs of colors assigned to pairs of states at the same level, the color pairing matrix \( \mathbf{P} \) should be a symmetric matrix in which
a given index does not appear more than once in a given row or column. There is one level in which each color is paired with its shifted version. Because we want this level to be the last, we set the level index between a color and its switched version \( N_t - 1 \). Finally, for each state \( i \), the algorithm finds its neighbor \( x \) on the level \( j \). The algorithm finds the color index \( c \) with level index \( j \) and the color index \( a(i) \) that is assigned to state \( i \). Then, the color index \( c \) is assigned to the neighboring state \( x \). The detailed algorithm is shown in Algorithm 2.

3.4.2 Special Cases

The algorithm assigns the outputs differently for the following cases:

1. The number of states is equal to the number of transmit antennas; i.e. \( 2^v = N_t \):

   In this case, different colors are assigned to the states without the need to go through Algorithm 2 because the number of colors, in this case, is equal to the number of states.

2. The number of states is smaller than the number of transmit antenna; i.e. \( 2^v < N_t \): In this case, since \( v \) is an integer and \( N_t \) is a power of two, \( 2^v < N_t \) implies that \( 2^v \leq N_t/2 \). Therefore, the algorithm assigns different colors to the states without using any switched version of a color, since the number of colors without their switched versions is \( N_t/2 \).

3. The number of states is larger than the number of antennas; i.e. \( 2^v > N_t \), but
Algorithm 2 Code Design Algorithm

**Input**: $N_t, v$

**Output**: O, S

1: Initialize O, S, C, P, a, b
2: $a(1) = 1$
3: $O(1,:) = 1$

4: for $i = 1 : 2^v$ do
5:  $S(i, 1) = (2i - 2) \mod 2^v$
6:  $S(i, 2) = (2i - 1) \mod 2^v$
7: end for

8: for $i = 1 : Nt/2$ do
9:  $C(i,:) = 2i - 2 : 2i - 1$
10: $C(i + Nt/2,:) = 2i - 1 : 2i - 2$
11: end for

12: for $i = 1 : Nt - 1$ do
13:  for $j = i + 1 : N_t$ do
14:   if $(i - 1) \mod N_t/2 == (j - 1) \mod N_t/2$ then
15:     $P(i,j) = N_t - 1$
16:   else
17:     $P(i,j) = \min(\text{setdiff}(1 : Nt - 2, [P(i,1 : j - 1) \text{ tr}(P(1 : i - 1,j))]))$
18:   end if
19:  end for
20: end for

21: for $i = 1 : 2^v$ do
22:  for $j = l(i) + 1 : v$ do
23:   if $(i - 1) \mod 2^j \geq 2^{j-1}$ then
24:     $x = i - 2^{j-1}$
25:   else
26:     $x = i + 2^{j-1}$
27:   end if
28:   $c = \text{find}(P(a(i,:),:) == ((j - 1) \mod (N_t - 1)) + 1)$
29:   $l(x) = j$
30:   $O(x,:) = C(c,:)$
31:   $a(x) = c$
32: end for
33: end for
the number of shift registers \(v\) is in the range

\[
N_x / 2^{x+1} \leq v \leq N_x / 2^x - 1
\]  

(3.25)

where \(x = 1, 2, \ldots, \log_2(N_x) - 2\): In this case, **Algorithm 2** will produce a code that maximizes the number of unique differences but uses only \(N_x / 2^x\) colors. The performance of the code in such a case can be improved by changing the output of the algorithm such that the code uses all the \(N_x\) colors. While the code will still generate the same number of unique differences, using \(N_x\) colors instead of only \(N_x / 2^x\) would result in a higher as, in such a case, the edges would contribute to a more connected incidence matrix. The output is changed by finding the states that use the same color and assigning some of them with unused colors. The detailed steps of making this change is summarized in **Algorithm 3**.

**Algorithm 3** Modified Algorithm for Special Case 3

<table>
<thead>
<tr>
<th>Input : (O, a, N_x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output : (O)</td>
</tr>
</tbody>
</table>

1: if length(unique(a)) < \(N_x\) then
2: \(w = \text{max}(a)\)
3: for \(i = 1 : w\) do
4: \(m = \text{find}(a == i)\)
5: for \(j = 2 : \text{length}(m) / \lfloor 2^x / N_x \rfloor\) do
6: \(u = m((j - 1) \times \lfloor 2^x / N_x \rfloor + k)\)
7: \(a(u) = \text{find}(P(a(u),:) == (j - 1) \times w)\)
8: \(O(u,:) = \text{C(a}(u,:))\)
9: end for
10: end for
11: end for
12: end if
3.4.3 Computational Complexity Analysis

**Algorithm 2** contains four loops. We analyze the computational complexity of the algorithm by computing the number of multiplications, additions, and comparisons needed for each loop and then sum up the results to get the total complexity:

1. For the first loop, we need $3 \times 2^v$ multiplications and $2 \times 2^v$ additions.

2. For the second loop, we need $N_t/2$ multiplications and $3 \times N_t/2$ additions.

3. For the third loop, we need $(N_t - 1) + N_t(N_t - 1)/2$ multiplications, $2(N_t - 1) + N_t(N_t - 1)/2$ additions, and $N_t^3/2 - 5/2 \times N_t^2 - 2N_t - 2$ comparisons.

4. For the fourth loop, the level index at which we assign a color to state $i - 1$ is $\lfloor \log_2(i - 1) \rfloor + 1$. Therefore, we need $4(2^v(v - 1) + v^2 + v)$ multiplications, $2^v + 2(2^v(v - 1) + v^2 + v)$ additions, and $2^v(v - 1) + v^2 + v$ comparisons.

The total number of calculations $N_{total}$ is, therefore, given by

$$N_{total} =$$

$$N_t^2/2 + N_t - 1 + 4v2^v - 2^v + 4v^2 + 4v \text{ multiplications}$$

$$+ N_t^2/2 + 3N_t - 2 + 2v2^v + 2^v + 2v^2 + v \text{ additions} \quad \text{(3.26)}$$

$$+ N_t^3/2 - 5/2 \times N_t^2 - 2N_t - 2 + v2^v - 2^v + v^2 + v \text{ comparisons}$$

Therefore, the complexity order of the algorithm is in the order of $\mathcal{O}(N_t^3 + v2^v)$. 

50
3.4.4 Code Properties

The rate-1/n convolutional code designed using the proposed algorithm has the following properties:

1. Maximal branch distance of $v+1$: Since the states are connected based on \((3.19)\).

2. Maximal effective length of $v+1$: Given the branch distance is maximized, the code can have less than the maximum effective length of $v+1$ only if there exist two neighboring states with the same color. The algorithm produces a code with no two neighbor states having the same color, and therefore, the rate-1/n code produced by the algorithm have the maximum effective length of $v+1$.

3. Maximal number of unique differences of $\min(v+1, N_t - 1)$: We showed that the maximum number of unique differences that can be generated by a rate-1/n code is $N_{\text{max}}(v) = \min(v+1, N_t - 1)$. We also showed that to maximize the number of unique differences, we should use $N_t/2$ disjoint pairs of colors at the same level and use different pairs of colors at different levels. Also, the pairs of colors that include colors with their switched version should be at last. By making the color pairing matrix $P$ symmetric and in such a way that a level index does not appear more than once in a given row or column, we get pairs of colors on the same level that are disjoint, and the pairs of colors are different at different levels. By enforcing that the colors with their switched version should be paired at last in the color pairing matrix $P$, we satisfy that requirement as
well. Thus, the rate-1/n code designed by this algorithm gives the maximal number of unique differences of \( \min(v + 1, N_t - 1) \).

4. Maximum guaranteed diversity: According to Lemma 3.2.6, a code will guarantee diversity \( d \) if the number of unique differences is equal to the upper bound in (3.9). Therefore, in order for a rate-1/n code with \( v \) shift registers to guarantee a diversity \( d \), \( d \) should satisfy the following inequality

\[
\binom{d}{2} + 1 \leq N_{\text{max}}(v)
\] (3.27)

The maximum \( d \) that satisfies the inequality (3.27) is the maximum guaranteed diversity \( d_g \) for rate-1/n code, and is given by

\[
d_g = \left\lfloor \frac{1 + \sqrt{1 + 8(\min(v + 1, N_t - 1) - 1)}}{2} \right\rfloor
\] (3.28)

3.4.5 Design Example

We provide a design example for rate-1/n codes using the proposed algorithm. In this example, we design a rate \( R = 1/2 \) code with \( v = 3 \) shift registers for \( N_t = 4 \) antennas. The algorithm produces the following matrices:
1. The next states matrix $S$ is given by

\[
\begin{align*}
S(1:4,:) &= \begin{bmatrix}
0 & 1 \\
2 & 3 \\
4 & 5 \\
6 & 7
\end{bmatrix}
\quad
S(5:8,:) &= \begin{bmatrix}
0 & 1 \\
2 & 3 \\
4 & 5 \\
6 & 7
\end{bmatrix}
\end{align*}
\] (3.29)

2. The colors matrix $C$ is given by

\[
C = \begin{bmatrix}
0 & 1 \\
2 & 3 \\
1 & 0 \\
3 & 2
\end{bmatrix}
\] (3.30)

3. The color priority matrix $P$ is given by

\[
P = \begin{bmatrix}
0 & 1 & 3 & 2 \\
1 & 0 & 2 & 3 \\
3 & 2 & 0 & 1 \\
2 & 3 & 1 & 0
\end{bmatrix}
\] (3.31)
4. The trellis outputs matrix $O$ is given by

$$
O(1:4,:) = \begin{bmatrix}
0 & 1 \\
2 & 3 \\
3 & 2 \\
1 & 0 \\
\end{bmatrix} \quad O(5:8,:) = \begin{bmatrix}
0 & 1 \\
2 & 3 \\
3 & 2 \\
1 & 0 \\
\end{bmatrix}
$$

(3.32)

Since $v = 3$, we have three grouping levels. The pairs of colors assigned at each level are as follows:

1. At the first level, we have state pairs $(0,1), (2,3), (4,5), (6,7)$ and they are assigned the colors $([0\ 1], [2\ 3]), ([3\ 2], [1\ 0]), ([1\ 0], [3\ 2]), ([2\ 3], [0\ 1])$, respectively.

2. At the second level, we have state pairs $(0,2), (1,3), (4,6), (5,7)$, and they are assigned the colors $([0\ 1], [3\ 2]), ([2\ 3], [1\ 0]), ([1\ 0], [2\ 3]), ([3\ 2], [0\ 1])$, respectively.

3. At the third level, we have state pairs $(0,4), (1,5), (2,6), (3,7)$, and they are assigned the colors $([0\ 1], [1\ 0]), ([2\ 3], [3\ 2]), ([3\ 2], [2\ 3]), ([1\ 0], [0\ 1])$, respectively.

At each level we have disjoint $N_t/2 = 2$ pairs of colors, and the pairs of colors are different at different levels. Since $N_t = 3$, the code generates 3 different levels and we keep the level in which colors are paired with their switched versions at last.
The code has $N_{max} = \min(v + 1, N_t - 1) = 3$ unique differences and thus, according to (3.28), guarantees diversity order $d_g = 2$.

3.4.6 Extension to Rate-$k/n$ Codes

The algorithm described in Algorithm 2 can be extended to design rate-$k/n$ codes by making the following modifications:

1. Each color has $B = 2^k$ permutations. Therefore, in the colors matrix $C$, the color indices of the set of the permutation of a color are $N_t/2^k$ evenly spaced.

2. There are $2^k - 1$ neighbors for each state at each level. For each state $i$, the algorithm finds its $l$th neighbor $x$ $(l = 1, 2, \ldots 2^k - 1)$ on the level $j$. The algorithm finds the color index $c$ with the pairing value $((j - 1)(2^k - 1) \mod (N_t - 1) + l)$ with the color index $a(i)$ that is assigned to state $i$. Then, the color index $c$ is assigned to the neighboring state $x$.

The rate-$k/n$ codes designed using our algorithm have the maximum branch distance because the states are connected according to (3.19). The codes also have the maximum effective length because there is no two neighbors with the same color. However, the algorithm does not necessarily guarantee maximizing the number of unique differences for rate-$k/n$ codes. To overcome this limitation, we propose a diversity-adaptation mechanism, in which, for any rate-$k/n$ code, we select only the codewords with desired diversity. We will describe this diversity adaptation scheme in the next section.
3.5 Diversity Adaptation Mechanism for Full Diversity Advantage

In this section, we propose a diversity adaptation mechanism for TC-STSK, in which, a code that achieves a higher diversity is constructed from a given code by choosing those pairwise codewords that have diversity larger or equal to the desired diversity $d \leq d_{\text{max}} = \min(T, N_t - 1)$, where $T$ is the length of the codewords. In particular, we use the proposed mechanism to achieve full diversity codes, i.e. $d = d_{\text{max}}$. Thus, by sacrificing the rate, the constructed code achieves full diversity. We also propose a modified soft Viterbi decoder for this diversity-adapted TC-STSK scheme.

3.5.1 Code Design

Starting with a given code, we generate all the outputs for all the possible inputs for a duration $T = L_{\text{max}}$ time intervals and for all starting states. The number of all possible inputs is $A = 2^{kT} = 2^{kL_{\text{max}}}$. For each starting state, we group the output codewords that end at the same state. In each group, we select the output codewords that have mutual pairwise error diversity larger or equal to the desired diversity $d \leq d_{\text{max}}$. We then choose only the input sequences that correspond to these selected outputs. Let $Q(d, i)$ be the number of selected outputs (and the corresponding inputs) that have diversity larger or equal to $d$ when the starting state is state $i = 0, 1, \ldots, 2^v - 1$, and $Q_{\text{min}}(d) = \min(Q(d, i))$ be the minimum number of selected outputs that have diversity larger or equal $d$ among all starting states. At the encoder, we use a mapper that takes $q = \lfloor \log_2(Q_{\text{min}}(d)) \rfloor$ bits and the encoder current state $i$. Since $2^q = 2^{\lfloor \log_2(Q_{\text{min}}(d)) \rfloor} \leq Q(d, i)$, we reduce the set of selected inputs at each starting
state $i$ to $2^q$ inputs. We then use the $q$ input bits to the mapper to select one of the $2^q$ inputs for the starting state $i$. We can define the bit rate of the diversity-adapted TC-STSK scheme as

$$R_b = \frac{q}{T} = \left\lfloor \frac{\log_2(Q_{\min}(d))}{T} \right\rfloor$$

(3.33)

The following example further illustrates the encoding process.

**Example 3.5.1.** Consider a TC-STSK scheme with a 4-state Rate-1/2 convolutional code with $v = 2$. The next states is given by (3.19), and the output matrix of the code is given by

$$O = \begin{bmatrix}
0 & 1 \\
2 & 3 \\
1 & 0 \\
3 & 2
\end{bmatrix}$$

(3.34)

Let $N_t = 4$, and $T = L_{\text{max}} = v + 1 = 3$, and the desired diversity $d = d_{\text{max}} = \min(T, N_t - 1) = 3$. The number of possible inputs during $T$ time intervals is $A = 2^{kT} = 8$. Table 3.2 shows the inputs in decimal, and outputs in terms of transmit antenna indices vector $x$ for $T = 3$. We see that if the starting state is 0, for example, the outputs that correspond to inputs 0, 2, 4, 6 have a pairwise diversity of 3. On the other hand, if the starting state is 2, for example, the outputs that correspond to inputs 1, 2, 5, 6 have a pairwise diversity of 3. Table 3.3 shows the selected inputs for each starting state.

We have $Q_{\min}(d_{\text{max}}) = 4$. Now we will use $q = \left\lfloor \log_2(Q_{\min}(d_{\text{max}})) \right\rfloor = 2$ bits
Table 3.2: Inputs, outputs, and end states for each starting state for a rate-1/2 4-state convolutional code in a TC-STSK system with $N_t = 4$ and $T = 3$

<table>
<thead>
<tr>
<th>Starting State $i$</th>
<th>Input</th>
<th>$x$</th>
<th>End State</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0 0 0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1 0 0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2 0 1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>3 0 1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>4 1 2</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>5 1 2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>6 1 3</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>7 1 3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>0 2 3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1 2 3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2 2 2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>3 2 2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>4 3 0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>5 3 0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>6 3 1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>7 3 1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>0 3 0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1 3 0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2 3 1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>3 3 1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>4 2 2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>5 2 2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>6 2 3</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>7 2 3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>0 3 0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1 0 3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2 0 2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>3 0 2</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>4 1 0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>5 1 0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>6 1 1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>7 1 1</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>
Table 3.3: Selected inputs for each starting state

<table>
<thead>
<tr>
<th>Starting State $i$</th>
<th>Selected Inputs</th>
<th>$Q(d_{\text{max}}, i)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0 2 4 6</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>1 2 5 6</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>1 2 5 6</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>1 3 5 7</td>
<td>4</td>
</tr>
</tbody>
</table>

as input to the mapper. Assume we have a bit sequence 0110 as input to the mapper. The initial state of the encoder is 0. The mapper will take the first two bits 01. Since the current state is 0, according to Table 3.3, this will select the input 2 or 010 to the encoder. The input 2 to the encoder that starts at State 0 will generate an output codeword that ends at State 2. So, now the current state is 2. The mapper will take the second two bits 10, and this will select the input 6 or 110 to the encoder. The input 6 to the encoder that starts at State 2 will generate an output codeword that ends at State 1. The mapper will take the next two bits and so on.

In Example 3.5.1, the number of output codewords that ended in the same state was two. However, if this number is larger than two, then selecting the codewords among these that has mutual pairwise error diversity larger or equal to $d$ is not straightforward. Therefore, in what follows, we provide an algorithm to do this selection. Given a $p$ number of output codewords that start and end in the same state, this algorithm selects the codewords among these that achieve mutual pairwise error diversity of at least $d$.

The algorithm works as follows: given a certain sorting of the codewords, it starts with the first codeword as an initial point; then, it examines the pairwise
error diversity using (3.7) of this initial point with other codewords but in the order of the given sorting. It removes the codewords that have a pairwise error diversity less than $d$. After that, the algorithm continues to the second codeword in the set of the remaining codewords and examines its pairwise error diversity with the other codewords and so on. The set of the remaining codewords that are left at the end is a set that achieves a mutual pairwise error diversity of at least $d$. However, because the sorting of the codewords affects the output, i.e., we can get a different number of elements of the set of the remaining codewords for different sorting, we run this algorithm for all possible $p!$ sorting orders. This code selection algorithm is summarized in Algorithm 4.

**Algorithm 4 Codeword Selection Algorithm**

<table>
<thead>
<tr>
<th>Input : $S = {X_1, X_2, X_3, \ldots, X_p}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output : $W$</td>
</tr>
<tr>
<td>1: $L := p! \times p$ matrix that contains all possible permutations without repetition for the elements ${1, 2, 3, \ldots, p}$. Each row represents a permutation.</td>
</tr>
<tr>
<td>2: $max = 0$</td>
</tr>
<tr>
<td>3: for $z = 1 : 1 : p!$ do</td>
</tr>
<tr>
<td>4: $A := a$ set that contains the elements of $S$ sorted according to the $z$th row of the matrix $L$.</td>
</tr>
<tr>
<td>5: $i = 1$</td>
</tr>
<tr>
<td>6: while $i &lt; num(A)$ do</td>
</tr>
<tr>
<td>7: for $j = i + 1 : 1 : num(A)$ do</td>
</tr>
<tr>
<td>8: if $rank(A(i) - A(j)) &lt; d$ then</td>
</tr>
<tr>
<td>9: remove $A(j)$</td>
</tr>
<tr>
<td>10: end if</td>
</tr>
<tr>
<td>11: end for</td>
</tr>
<tr>
<td>12: $i = i + 1$</td>
</tr>
<tr>
<td>13: end while</td>
</tr>
<tr>
<td>14: if $num(A) &gt; max$ then</td>
</tr>
<tr>
<td>15: $max = num(A)$</td>
</tr>
<tr>
<td>16: $B = A$</td>
</tr>
<tr>
<td>17: end if</td>
</tr>
<tr>
<td>18: end for</td>
</tr>
</tbody>
</table>
Example 3.5.2. Consider a rate-2/3 convolutional encoder with $N_t = 8$ and $T = L_{\text{max}} = 3$. Let $x_1, x_2, x_3, x_4$ be transmit antenna index vectors that represent four outputs codewords that start and end at the same state. Let $x_1 = (2 \ 2 \ 5), \ x_2 = (3 \ 3 \ 6), \ x_3 = (1 \ 4 \ 1), \ x_4 = (0 \ 2 \ 2)$. If the desired diversity is the full diversity $d_{\text{max}} = \min(T, N_t - 1) = 3$, then by applying Algorithm 4, we get a maximum number of three vectors that satisfy the required pairwise error diversity when the vectors are sorted as $x_2, x_1, x_3, x_4$ which are $x_2, x_3, x_4$. However, the vectors are sorted differently as $x_1, x_2, x_3, x_4$, the algorithm finds only two vectors that satisfy the required diversity which are $x_1, x_2$. This shows how the sorting of the vectors affects the output.

3.5.2 Decoder Design

In the diversity-adapted TC-STSK, we select only some of the possible inputs so that we can have pairwise codewords with a diversity of at least $d$. This means that the encoder allows only certain paths on the trellis diagram. In fact, the number of paths that the encoder allows is $2^{v+q}$ paths of length $T$ out of $2^{v+kT}$ total paths. In such a case, since not all the paths are being used by the encoder, we need to implement a modified version of the soft Viterbi decoder. In this modified version, we compute the decoding metric for each state every $T$ time intervals and not after each time interval as in the conventional soft Viterbi decoding. We compare the received sequence of soft values of length $T$ only with the valid $2^{v+q}$ paths allowed by the encoder. Note that the demapping in the diversity-adapted TC-STSK is integrated into the decoding.
process since when computing the metric for each state, the starting state and the input index of the best path are stored. The decoding algorithm is summarized in **Algorithm 5**. The algorithm takes the following inputs:

1. $T$, $Y$, $H$, $k$, $q$, and $v$.

2. $M_c$ is $2^v \times 1$ vector that contains the current metric for each starting state. Because the encoder starts at State 0, the initial current metric is zero for State 0 and $\infty$ for the rest of the states.

3. $L$ is a $2^v \times 2^{kT} \times T$ matrix in which the $(i,j,l)$ entry is the output symbol of the path that starts at State $i - 1$ for the Input $j - 1$ at the $l$th time interval $l = 1, 2, \ldots, T$.

4. $F$ is $2^v \times 2^q$ matrix in which the $(i,j)$ entry is the $j$th selected input for the start state $i - 1$.

5. $E$ is $2^{kT} \times 1$ vector in which the $j$th entry is the end state after $T$ time intervals for the Input $j - 1$.

The algorithm produces the following outputs:

1. $M_o$ is a $2^v \times 1$ vector that contains the metric for each ending state. The initial value of all the entries in this vector is $\infty$.

2. $S_o$ is a $2^v \times 1$ vector that contains the previous starting state for each ending state. The initial value of all the entries of this vector is 0.
3. \( I_o \) is a \( 2^v \times 1 \) vector that contains the value of the previous input in decimal for each ending state. The initial value of all the entries of this vector is 0.

Algorithm 5 Modified Viterbi Decoder for TC-STSK

\begin{verbatim}
Input : \( T, Y, H, k, q, v, M_c, L, F, E \)
Output : \( M_o, S_o, I_o \)
1: for \( j = 0 : 2^v - 1 \) do
2: \hspace{1em} \text{min} = \infty
3: \hspace{1em} for \( i = 0 : 2^v - 1 \) do
4: \hspace{2em} \text{c} = 0
5: \hspace{3em} for \( r = 1 : 2^q \) do
6: \hspace{4em} \text{p} = F(i + 1, r) + 1
7: \hspace{4em} \text{p} = \lfloor p/2^v \rfloor
8: \hspace{4em} \text{if} \ j == E(p) \text{ then}
9: \hspace{5em} \text{c} = \text{c} + 1
10: \hspace{5em} \text{l(c)} = \text{p}
11: \hspace{4em} \text{end if}
12: \hspace{3em} \text{end for}
13: \hspace{1em} \text{for} \ k = 1 : c \text{ do}
14: \hspace{2em} \text{dist} = 0
15: \hspace{3em} for \( m = 1 : T \) do
16: \hspace{4em} \text{sym} = L(i + 1, l(k), m)
17: \hspace{4em} \text{dist} = \text{dist} + \| Y(:, m) - H(:, sym) \|^2
18: \hspace{3em} \text{end for}
19: \hspace{2em} \text{temp} = M_c(i + 1) + \text{dist}
20: \hspace{1em} \text{if} \ \text{temp} < \text{min} \text{ then}
21: \hspace{2em} \text{min} = \text{temp}
22: \hspace{1em} \text{S_o}(j + 1) = i
23: \hspace{1em} \text{I_o}(j + 1) = \text{l(c)} - 1
24: \hspace{1em} \text{M_o}(j + 1) = \text{temp}
25: \hspace{1em} \text{end if}
26: \hspace{1em} \text{end for}
27: \hspace{1em} \text{end for}
28: \text{end for}
\end{verbatim}

Example 3.5.3. Consider the encoder in Example 3.5.1. Assume the decoding metric computation is done every \( T = 3 \) time intervals. Let \( N_r = 1, \) and \( H = [0.7172 + j0.7259 \ 1.6302 - j0.3034 \ 0.4889 + j0.2939 \ 1.0347 - j0.7873] \) and the received soft values \( Y = [0.6780 + j1.1750 \ 2.1013 - j0.0911 \ 0.9345 - j0.0880]. \) The trellis
Figure 3.5: Trellis diagram of the modified Viterbi decoder in Example 3.5.3.

diagram at the decoder is shown in Fig. 3.5, where we only show the valid paths from State 0 (this is anyhow the case at the beginning of the decoding process). In the diversity-adapted scheme, we select only 4 out of 8 possible paths from State 0. We have shown the four selected paths in the figure. Two of those paths end at State 0 after $T = 3$ time intervals and thus the metric is computed for these two paths for State 0. It turns out that, in this example, the first path has the smaller metric (3.49). So, the second path is shown with a dotted line which means the decoder discards that path. Similarly, State 2 has a minimum metric of 0.81 and the selected path is shown with a solid line. Ultimately, there will be 16 valid paths every $T$ time intervals, and the decoder calculates the metric for each ending state in the same way explained in this example.
### Table 3.4: Designed codes for the TC-STSK scheme

<table>
<thead>
<tr>
<th>Code</th>
<th>$R$</th>
<th>$v$</th>
<th>O</th>
<th></th>
<th>O</th>
<th></th>
<th>O</th>
<th></th>
<th>O</th>
<th></th>
<th>O</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>$(1 : 2)$</td>
<td>0 1</td>
<td>2</td>
<td>1 2</td>
<td>1 0</td>
<td>3 2</td>
<td>1</td>
<td>3 2</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>A</td>
<td>1/2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>1/3</td>
<td>4</td>
<td>$(5 : 8)$</td>
<td>6 7</td>
<td>4 5</td>
<td>2 3</td>
<td>1 0</td>
<td>7 6</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$(9 : 12)$</td>
<td>0 1</td>
<td></td>
<td></td>
<td>5 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>1/4</td>
<td>5</td>
<td>$(9 : 16)$</td>
<td>8 9</td>
<td>11</td>
<td>10</td>
<td>3 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$(17 : 24)$</td>
<td>12 13</td>
<td>15</td>
<td>14</td>
<td>7 6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$(25 : 32)$</td>
<td>14 15</td>
<td>13</td>
<td>13</td>
<td>5 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>2/3</td>
<td>4</td>
<td>$(1 : 8)$</td>
<td>5 4</td>
<td>7 6</td>
<td>1 0  3 2</td>
<td>6 5  3 2</td>
<td>6 5  3 2</td>
<td>6 5  3 2</td>
<td>6 5  3 2</td>
<td>6 5  3 2</td>
<td>6 5  3 2</td>
</tr>
</tbody>
</table>

### 3.6 Results and Discussions

In this section, we simulate the performance of some of the codes that are designed based on our proposed algorithm in Section 3.4. Table 3.4 lists some of these codes with rates 1/2, 1/3, 1/4, and 2/3. The output matrix, $O$, of these codes are listed. The next states for all of the codes are computed using (3.19).

In order to verify that our designed rate-1/$n$ codes guarantee diversity $d_g$ in (3.28), we use Algorithm 4 to compute the number of codewords with diversity of at
Table 3.5: Bit rate for codes A, B, C, and D for different diversity orders

<table>
<thead>
<tr>
<th>Code</th>
<th>(d)</th>
<th>(Q_{\text{min}}(d)/2^{kT})</th>
<th>(R_b)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A (d_g = 2)</td>
<td>8/8</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>A (d_{\text{max}} = 3)</td>
<td>4/8</td>
<td>2/3</td>
<td></td>
</tr>
<tr>
<td>B (d_g = 3)</td>
<td>32/32</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>B (d_{\text{max}} = 5)</td>
<td>20/32</td>
<td>4/5</td>
<td></td>
</tr>
<tr>
<td>C (d_g = 5)</td>
<td>64/64</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>C (d_{\text{max}} = 6)</td>
<td>52/64</td>
<td>5/6</td>
<td></td>
</tr>
<tr>
<td>D (d_{\text{max}} = 3)</td>
<td>56/64</td>
<td>5/3</td>
<td></td>
</tr>
</tbody>
</table>

least \(d_g\), i.e. \(Q_{\text{min}}(d_g)\). We also compute the number of codewords with full diversity, i.e. \(Q_{\text{min}}(d_{\text{max}})\) and their corresponding rate, \(R_b\), using (3.33) for the codes in Table 3.4. The results are shown in able 3.5, where we show \(Q_{\text{min}}(d)\) as fraction of the total number of codewords \(2^{kT}\) where \(T = L_{\text{max}} = \lfloor v/k \rfloor + 1\). Note that for our designed rate-1/n codes, the codes have full rate for diversity order of \(d_g\) in (3.28).

We also simulate the bit error rate (BER) performance of these codes. We assume that the channel is fixed for a block length of 20, and changes independently from one block to the other. Since the full rate case is a special case of diversity adaptation in which we select all the codewords, we use the modified soft Viterbi decoder proposed in Section 3.5 for all the simulation cases. As our focus in this paper is to achieve transmit diversity, we limit the number of receive antennas to \(N_r = 1\).

Fig. 3.6 shows the BER for rate-1/n codes, A, B, and C, as well as the BER of uncoded SSK with bit rate \(R_b = k = 1\) for a system with \(N_t = 2\) antennas. For each code, we simulate both the full rate and full diversity cases. Similarly, we
Figure 3.6: BER performance of different rate-1/n TC-STSK schemes and uncoded SSK with $N_t = 2$. 

```plaintext
Figure 3.6: BER performance of different rate-1/n TC-STSK schemes and uncoded SSK with $N_t = 2$. 
```
Figure 3.7: BER performance of rate-2/3 TC-STSK and uncoded SSK with $N_t = 4$

compare the BER of full diversity adapted code $D (R=2/3)$ and rate-2/4 and 2/5 codes with uncoded SSK with bit rate $R_b = k - 2 (N_t = 4)$ in Fig. 3.7. We observe that the diversity orders measured from the simulation plots in Figs. 3.6 and 3.7 are very close to the theoretical values in Table 3.5. Also, the diversity performance is improved in the full diversity adapted rate cases compared to the full bit rate cases.

Next, we study the effect of the number of transmit antennas, the bit rate, and the number of shift registers on the performance of TC-STSK. In Fig. 3.8, we simulate the BER of rate-1/2, 1/3, 1/4 and 1/5 codes to see the effect of the number
of transmit antennas. For all the cases, we consider the full rate case with $R_b = 1$ and $v = 3$ shift registers. We notice that the performance of rate-1/3, 1/4, 1/5 is superior to that of the rate-1/2 code. However, the number of the required transmit antennas for rate-1/3, 1/4 and 1/5 codes are, respectively, $N_t = 8$, $N_t = 16$ and $N_t = 32$ and much larger than the maximum effective length in this case, which is $T = L_{\text{max}} = v + 1 = 4$. Since $d_{\text{max}} = \min(T, N_t - 1)$, the maximum diversity for rate-1/3, 1/4 and 1/5 codes is equal to the effective length of the code. Therefore, while increasing the number of transmit antennas from $N_t = 4$ in the case of rate-1/2 code, which has maximum achievable diversity of 3, to $N_t = 8$ in the case of rate-1/3 code improves the performance, increasing the number of transmit antennas beyond the effective length in the remaining cases does not provide significant BER improvement.

In Fig. 3.9, we simulate the BER performance of rate-1/4, 2/4 and 3/4 codes to study the effect the bit rate. We consider full diversity adapted cases of these codes with $R_b = 1$, $R_b = 1.67$ and $R_b = 2$, respectively. The number of shift registers in all these cases is $v = 4$ and the number of transmit antennas is $N_t = 16$. We see that the performance of the rate-2/4 code with $R_b = 1.67$ is worse than the performance of the rate-1/4 code with $R_b = 1$, and the performance of the rate the rate-3/4 code with $R_b = 2$ is worse than the performance of the rate-2/4 code with $R_b = 1.67$. Therefore, the BER performance decreases with increasing bit rate for fixed $v$ and $N_t$. The is because with increasing bit rate, from 3.18, the effective length decreases when the number of input bits $k$ increases for a fixed number of shift registers resulting in a
Figure 3.8: BER performance of rate-1/n codes with $R_b = 1$, $v = 3$ and different number of transmit antennas
Figure 3.9: BER performance of rate-$k/4$ codes with $N_t = 16$ and $v = 4$.

lower diversity order for a larger $k$.

In Fig. 3.10, we simulate the BER of three different rate-$1/4$ codes with $v = 3, 4, 5$ shift registers. We consider full rate case with $R_b = 1$ and $N_t = 16$ transmit antennas. We notice that increasing the number of shift registers improves the BER performance. This is because it increases the effective length of the code, and, therefore, the achievable diversity.

For the sake of completeness, we compare the performance of TC-STSK scheme with that of the STTC for a system with $N_t = 2$ and $N_r = 1$ and 2 in Fig. 3.11. We consider a four state (with two shift registers) QPSK-STTC and a rate-2/3 TC-STSK code with two shift registers. We consider full rate TC-STSK
Figure 3.10: BER performance of rate-1/4 codes with $R_b = 1$ with $N_t = 16$ and different number of shift registers.
Figure 3.11: BER performance of four states QPSK-STTC code and full bit rate rate-2/3 TC-STSK code with $N_t = 2$ and $v = 2$.

case with $R_b = 2$. Note that with these assumptions both schemes provide the same bit rate and have the same number of shift registers (encoder complexity). From the figure, the proposed TC-STSK significantly outperforms STTC in all cases.
In this thesis, we investigated exploiting the CSI in SM/SSK to combine the scheme with two performance enhancement techniques, namely, bit-mapping and TCM, for additional performance improvement.

In the second chapter, a low-complexity adaptive bit mapping algorithm for SM was proposed. NPM sorts the SM symbols according to a greedy sorting order. It starts with the pair of symbols with minimum distance, and continues to each nearest symbol until it visits all the symbols. The determined sorting order is used to Gray code the sorted SM symbols as if the SM symbols were on a line or a circle, similar to PSK symbols. Simulation results show that NPM achieves a performance close to the lower bound of the BER for both SSK and SM when using full CSIT. We only considered the case of perfect CSIT in this paper. When the CSIT is erroneous or outdated, it will negatively affect the performance of a bit mapping scheme. The study and design of the bit mapping schemes under imperfect CSIT is, however, beyond the scope of this work.

In the third chapter, we investigated the combination of TCM and SSK. We provided a graph-theoretical analysis for the rank of the code difference matrix by realizing the fact that the code difference matrix represents an incidence matrix for
a graph of $N_t$ vertices and $T$ edges. Using the incidence matrix analysis, we stated that the goal of the code design for TC-STSK should be maximizing the number of unique differences between pairs of codewords. To achieve this goal, we provided code design guidelines, in which we stated that we need to maximize the branch distance and the effective length first. We then derived the maximum number of unique differences for rate-1/n codes in (3.24). We accordingly proposed a design algorithm for rate-1/n codes that guarantees a diversity order of $d_g$ in (3.28). To enhance the diversity performance of the TC-STSK system, we then proposed a diversity adaptation mechanism, in which only a set of codewords with desired diversity are selected. We used the proposed mechanism to achieve full diversity. We described the encoding mechanism for the proposed diversity-adapted scheme, and provided a modified soft Viterbi decoder that compares the received signal only with the valid paths chosen at the encoder. With simulation, we confirmed the achievable diversity order of the proposed scheme. We also simulated the effect of the system parameters on the performance of TC-STSK scheme. We noticed that decreasing the bit rate and increasing the number of shift registers improves the BER performance. While increasing the number of transmit antennas also improves the BER performance, we noticed that increasing the number of transmit antennas beyond the effective length of the code does not give much BER improvement.

The following topics are left as future work:

1. Analysis of maximum unique differences and design of algorithm for rate $k/n$ TC-STSK codes: Our proposed design algorithm, while it can be used to design
rate $k/n$ codes, is mostly suitable for rate $1/n$ codes. A future work is extending our analysis for rate $1/n$ codes to rate $k/n$ codes.

2. Analysis of maximum unique differences and design of algorithm for rate $1/n$ or $k/n$ codes for the combination of TCM and SM: We investigated the combination of TCM and SSK. A future work is providing similar analysis with the presence of APM,

3. Study bit-mapping algorithms for spatial modulation with partial CSIT: We proposed the NPM algorithm as bit-mapping scheme for full CSIT SM/SSK and partial CSIT SSK. A future work is understanding the SM constellation when having partial CSIT and according study bit-mapping algorithm in this case,

4. Study the diversity-multiplexing trade-off for the combination of TCM and SM/SSK,

5. Study the combination of TCM and generalized SM/SSK, in which more than one antenna are active at a time.
BIBLIOGRAPHY


