Analysis of Space Time Trellis Codes

A Thesis
Submitted in Partial Fulfillment for the Award of
M.Tech in Information Technology

By

Deepshikha Katiyar
2008026

To

International Institute of Information Technology
Bangalore – 560100

June 2010

Acknowledgement

I would like to thank Prof. Wehn and IIIT for providing me with such opportunity to do my last semester thesis in Technical University, Kaiserslautern. I am equally grateful to my supervisor, Christina Gimmler, for her thoughtful guidance, many insightful suggestions and generous support throughout this thesis. It has been both a privilege and a pleasure to have been associated with them.

I would like to thank all my professors at IIIT-B for their support, invaluable advice and motivation. I would especially thank Prof. S.S.Prabhu, Prof. M.Pandit and Prof. J.Bapat for introducing me to this field and for their constant support.

I am grateful to my dear friend Venkat for his continuous guidance and encouragement ever since.

I would like to express my deepest gratitude to my family and friends, for their unbounded support and affection, for all they have given me throughout the years.

Contents

Abstract

Multiple-input multiple-output (MIMO) communication technology has received a significant attention in the area of wireless communication systems due to reliable transmission and increasing data rate requirements. It offers significant capacity gains over traditional single-input single-output channels. To achieve this, the idea has been to concentrate on designing more efficient signaling techniques, since the available radio spectrum is limited.

The purpose of this thesis was to do a deep study of Space Time Trellis Codes along with its implementation. An encoder and a decoder for Space Time Trellis Codes with the ability to process the common Trellis codes, has been implemented as a generic software module using IT++ library. The trellis codes are flexible in terms of number of transmit and receive antenna, data rate, modulation scheme, etc. This software module is independent enough to be used in other simulations.

The results showed a significant improvement in performance of Space Time Trellis Codes with increasing number of states, number of transmit and receive antennas. However, the performance got poor when the data rate was increased for same antenna constellation, resulting in a trade-off between increasing data rate and coding gain.

List of Tables

List of Figures

List of Abbreviations

STTC: Space Time Trellis Codes

MPSK: M-ary Phase Shift Keying

QPSK: Quadrature Phase Shift Keying

FER: Frame Error Rate

BER: Bit Error Rate

SNR: Signal to Noise Ratio

MIMO: Multiple Input Multiple Output

MSB: Most Significant Bit

STC: Space Time Coding

BMV: Branch Metric Value

PMV: Path Metric Value

CSI: Channel State Information

TCM: Trellis Coded Modulation

1 Introduction

Wireless communication is a rapidly growing segment of the communications industry with the aim of providing high speed, good quality exchange of information between movable devices located anywhere in the world. However, in a wireless system, transmitted signal propagates through several paths which may suffer from the interference due to other users transmitting radio signals. Apart from interference, noise present in the environment also adds to the signal and corrupts it. These factor counts for the attenuation in the signal and hence the receiver may not determine the correct transmitted signal resulting in unreliable and inefficient transmission of information over such channel. This leads to lower quality and higher risk of disconnection. Such kind of challenges to achieve reliable wireless systems with high spectral efficiency and low complexity, results in the need for continued research in this field.

In this chapter, we briefly discuss about the wireless communication, the challenges related to it with main focus on MIMO systems and Space Time Coding. A brief background is also presented for the same followed by main focus and scope of this thesis. An overview of the thesis structure is also presented in this chapter.

1.1 MIMO Systems

Multiple-input multiple-output (MIMO) wireless communication systems are extension of development in antenna array communication. The idea behind these systems is to exploit the correlation between multiple received signals in the presence of multi-path propagation. It employs multiple antennas for both transmission and reception. This increases the capacity of the wireless channel and makes the system more reliable. Capacity is defined as the maximum data rate achievable for an arbitrarily low probability of error without the use of additional bandwidth or transmit power.

One of the challenges with wireless communication system is interference. Because MIMO systems use antenna arrays, localized interference can be reduced. Therefore, MIMO systems provide many advantages over single-input single-output communication like array gain, diversity gain and spatial multiplexing. Array gain is the average increase in the SNR at the receiver that arises from the coherent combining effect of multiple antennas at the receiver or transmitter or both. Spatial multiplexing is a transmission technique to transmit independent and separately encoded information signals from each of the multiple transmit antenna. It offers linear increase in transmission rate for the same bandwidth without additional power expenditure. In implementing MIMO systems, we must decide whether channel estimation information will be available to the transmitter so that the transmitter can adapt. Most MIMO communication research has focused on systems without feedback i.e., without providing such information to the transmitter. A MIMO system with an uninformed transmitter (without feedback) is simpler to implement, and at high SNR its spectral-efficiency bound approaches that of an informed transmitter (with feedback). Here, spectral efficiency is defined as the total number of information bits per second per Hertz transmitted from one array to the other.

So we see that wireless communication using MIMO systems enables increased bandwidth efficiency for a given total transmit power and promises good quality transmission.

1.2 Space Time Coding

In order to implement a MIMO communication system, we must first select a particular coding scheme. A Space Time Code is a coding technique that uses multiple transmit as well as receive antennas over fading channels, to achieve efficient and reliable data transmission in wireless communication systems. It is performed in both the spatial and temporal domains to introduce correlation between signals transmitted from various antennas at various time periods. This correlation is then used to exploit the scattering environment and minimize transmission errors at the receiver.

Space time codes are mainly of two types:

Space Time Trellis Codes (STTC): It is an extension of convolutional coding with symbol mapping onto multiple transmit antennas. The STTCs create inter-relations between signals in the space domain (different transmit antennas) and signals in the time domain (consecutive time symbols). The encoder is composed of n^T(transmit antennas) different generator polynomials to determine the simultaneously transmitted symbols. The receiver is based upon the channel estimation of the fade coefficients and Maximum Likelihood Sequence Estimation (MLSE) decoder, which computes the lowest accumulated Euclidean distance metric to extract the most likely transmitted sequence. STTCs provide both coding gain as well as diversity gain but decoding complexity in the receiver which grows exponentially with the memory length of the trellis code.

Space Time Block Codes (STBC): Here multiple copies of data or block of data are transmitted at once. It uses maximum likelihood decoding via linear processing at the receiver, which accounts for low complexity receiver. It provides diversity gain but suffers from lack of coding gain.

In this thesis we focus on STTCs, which have been developed to simultaneously provide coding gain and diversity in MIMO systems. The coding gain is dependent on the code construction criteria and on the length of the memory in the encoder. A number of different structures have been proposed for STTCs.

1.3 Thesis Focus

The focus of this thesis is on the detailed study of the Space Time Trellis Codes as proposed by Tarokh et al, along with its software implementation. STTCs are an important class of space-time codes which can simultaneously provide coding gain and diversity on fading channels. The study includes the analysis of STTCs for different antenna constellation size, modulation schemes and data rates. The software part includes the development of an independent generic module in C++ language using IT++ library. The software is able to process different STTC codes.

1.4 Structure of Thesis

Section 2 provides the background information on Space time Trellis coded system followed by a detailed description of STTC encoder, STTC decoder structure and a brief introduction to the channel used. An example of Space Time Trellis Code has also been given. The details of the MIMO channel are provided in section 3. The simulation system model which explains the software module is in section 4. Simulation results for this system are presented in section 5. The thesis is concluded in section 6, with a summary of the main results and future work that can be done to enhance the simulation model.

2 Space Time Trellis Codes

This section presents an overview of space time trellis coded system followed by details about the encoding and decoding algorithm of STTCs. The space-time system model used throughout this work along with an example STTC system is presented in this section. This section also briefly talks about the MIMO channel used of which the details have been discussed later in section 3. At the end of this section, we discuss QR decomposition in Maximum likelihood decoding in STTC decoder. The STTC encoding and decoding algorithm provided in this section is primarily based on the material presented in [1].

2.1 Introduction

As mentioned earlier, many different STC schemes have been proposed of which Space time block codes (STBC) and space time trellis codes (STTC) are the two major classes. STBCs provide maximum diversity advantage using simple decoding algorithm, but they do not provide coding gain. STTCs on the other hand provide an effective alternative signaling technique by joint design of error control coding, modulation, transmit, and receive diversity.

2.2 System Model

Throughout this work, we consider multi antenna communication system model, where the channel is assumed to be constant for atleast L(>1) channel uses. Assuming the transmitter is equipped with n_T antennas and receiver is equipped with n_R antennas, the relationship between transmitted and received signals, for each channel instant H can be represented in vector form as

$$y = H.c + n$$

where $y \in C(L \times n_R)$ and $c \in C(L \times n_T)$ are the received and transmitted matrices of complex values and $n \in C(1 \times n_R)$ and $H \in C(n_T \times n_R)$ with independent $CN(0,1)$ are the additive noise and channel matrix respectively. H is assumed to be known to the receiver but not to the transmitter. A typical system of this kind, with n_T transmit and n_R receive antennas, where the transmitted data is encoded by a space-time encoder, is depicted in Figure 2. At each time instant t, a block of m binary information bits, denoted by $a_t = (a_t^1, a_t^2, \dots a_t^m)$ is fed into the space-time encoder. The encoder maps the block of m binary input data into n_T modulation symbols from a signal set of $M = 2^m$ points, for an M-ary signal constellation. The coded data, called the space-time symbol, can then be represented by the column vector $c_t = (c_t^1, c_t^2, \dots c_t^{nT})^T$, where T denotes the transpose of the vector. The n_T parallel outputs are simultaneously transmitted by n_T transmit antennas, whereby symbol c_t^j, $1 \leq j \leq n_T$, is transmitted by antenna j, with all transmitted symbols having the same duration. Assuming the transmitted data frame length is L symbols for each antenna, the space-time codeword matrix, can be defined as $C = [c_1, c_2, \dots c_L]$.

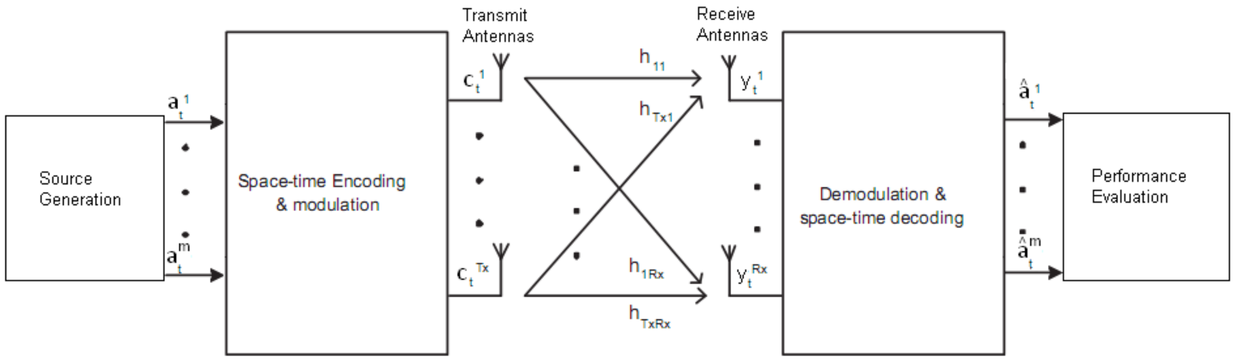

Figure 1: Block diagram of Space time coded system

The signal is received after being distorted by the channel fading and additive random noise. The received space-time symbol is represented by $y_t = (y_t^1, y_t^2, \dots\ y_t^{nR})^T$ or in matrix form as $Y = [y_1, y_2, \dots\ y_L]$,where y_t^j is the received signal at receive antenna $j \in \{1,2,.., n_R\}$ at time t.

We define the spectral efficiency, to be the number of information bits transmitted per time slot, in this case m bits/sec/Hz. The rate of the code is defined as the number of information bits over the total number of bits transmitted per interval. In this case, assuming an M-ary signal constellation, for each m information bits, n_T x m bits are transmitted. The rate of the code is then given as $\frac{m}{nT \times m}$.

2.3 Encoding of Space Time Trellis Codes

In STTCs, the encoder maps binary data to modulation symbols, where the mapping function is described by a trellis diagram. A trellis diagram for 4PSK 4 State has been given in Figure 4. The encoding is similar to trellis coded modulation except that at the beginning and end of each frame, the encoder is required to be in the zero state. Consider a STTC system with M-PSK modulation, n_T transmit antennas and n_R receive antennas. At time t, the encoded M-PSK symbols $c_t^1, c_t^2, \dots, c_t^{nT}$ are simultaneously transmitted over n_T antennas.

An M-PSK STTC encoder with memory order v and n_T transmit antennas is presented in Figure2. This encoder consists of *log2 (M)* branches of shift registers with total memory order v. At time t, *log2 (M)* binary inputs a_t^k, k $\in\{1, 2, \dots, log2(M)\}$, are fed into the *log2(M)* branches. The memory order of the k-th branch is given by:

$$vk = \left\lfloor \frac{v + k - 1}{log2(M)} \right\rfloor$$

where $\lfloor x \rfloor$ denotes the maximum integer not larger than x.

Let *b = log2(M)* ,

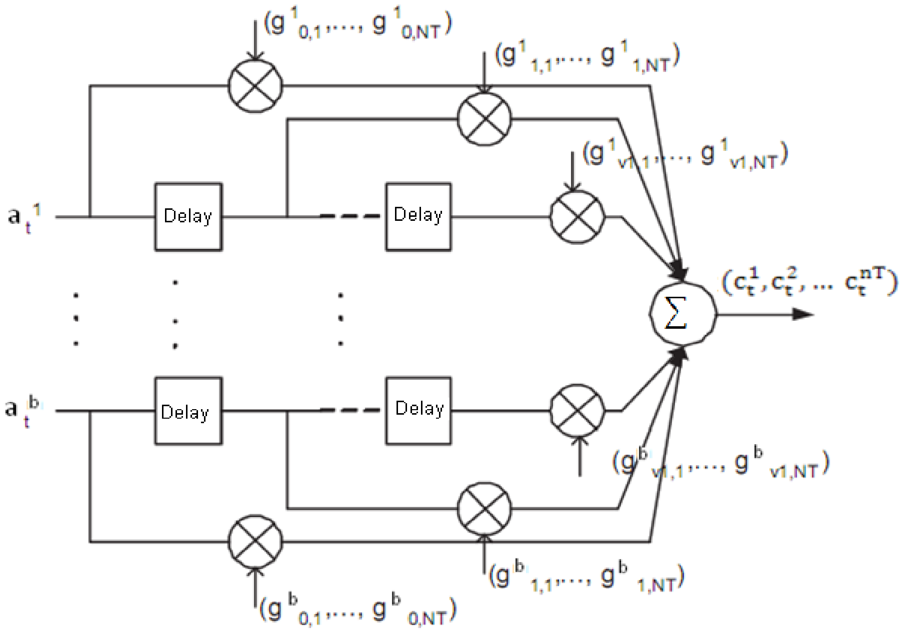

Figure 2: General STTC encoder structure for M-PSK with Nt transmit antennas

The *b* streams of input bits are simultaneously passed through their corresponding shift register branches and multiplied by the coefficient vectors given by g^k, where $k = \{1,2,...,b\}$

$$\mathbf{g}^1 = [(g^1_{0,1}, g^1_{0,2}, ..., g^1_{0,N_t}), (g^1_{1,1}, g^1_{1,2}, ..., g^1_{1,N_t}), ..., (g^1_{v_1,1}, g^1_{v_1,2}, ..., g^1_{v_1,N_t})]$$
$$\mathbf{g}^2 = [(g^2_{0,1}, g^2_{0,2}, ..., g^2_{0,N_t}), (g^2_{1,1}, g^2_{1,2}, ..., g^2_{1,N_t}), ..., (g^2_{v_2,1}, g^2_{v_2,2}, ..., g^2_{v_2,N_t})]$$
$$\vdots$$
$$\mathbf{g}^b = [(g^b_{0,1}, g^b_{0,2}, ..., g^b_{0,N_t}), (g^b_{1,1}, g^b_{1,2}, ..., g^b_{1,N_t}), ..., (g^b_{vb\ ,1}, g^b_{vb\ ,2}, ..., g^b_{vb\ ,N_t})],$$

where $g^k_{jk,i}$ is an *M*-PSK symbol, for $k \in [1,b]$, $j_k \in [0,vk]$and $i \in [1,Nt]$.

The encoder output at time t for transmit antenna i, can be computed as:

$$c_t^i = \sum_{k=1}^{b} . \sum_{jk=0}^{vk} g^k_{jk,i}\ a^k_{t-jk}\ (mod\ M)$$

These outputs are elements of an *M*-ary signal set.This encoder can also be described in generator polynomial format. The *k*th binary input stream a^k can be represented as:

$$a^k(D) = a_0^k + a_1^k(D) + a_2^k\,(D^2) + \ ...,\quad k = 1, ..., b$$

where $a^k{}_t \in [0, 1]$ and D represents a unit delay operator. The generator polynomial for the ith transmit antenna can be represented as:

$$G_i^k(D) = g_{0,i}^k + g_{1,i}^k D + \cdots + g_{vk,i}^k D^2, \quad k = 1, \ldots, b\ ;\ i = 1, \ldots, Nt$$

The coded symbol sequence transmitted from antenna i is given by:

$$c^i(D) = \sum_{k=1}^{b} a^k(D)\, G_i^k(D) \quad (mod\ M)$$

This can also be written as: $c^i(D) = [a^1(D) \ldots a^b(D)] \begin{bmatrix} G_i^1(D) \\ G_i^2(D) \\ \vdots \\ G_i^b(D) \end{bmatrix} \ (mod\ M)$

where, $G^i(D) = \begin{bmatrix} G_i^1(D) \\ G_i^2(D) \\ \vdots \\ G_i^b(D) \end{bmatrix}$ is the generator matrix for antenna i.

An STTC encoder is a finite state machine and therefore an encoder with b bits will have at least 2^b states. There are 2^b branches, corresponding to 2^b different input patterns, leaving each state. Hence, for 4 PSK 4-States with two transmit antennas, the trellis diagram is given in Figure 4 below.

Input bits	00	01	10	11
State 0 Output for antenna 1, antenna 2	00	01	02	03
State 1 Output for antenna 1, antenna 2	10	11	12	13
State 2 Output for antenna 1, antenna 2	20	21	22	23
State 3 Output for antenna 1, antenna 2	30	31	32	33

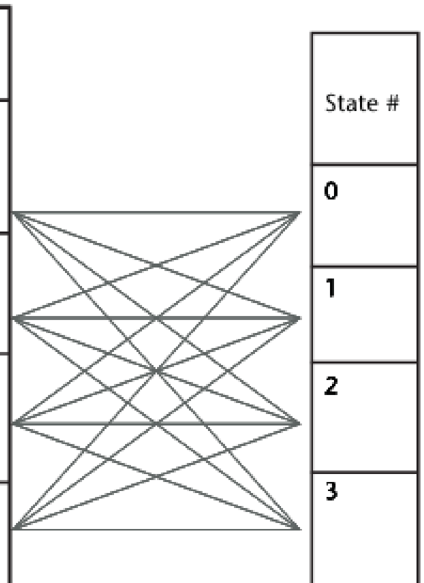

Figure 3: An example trellis structure for two transmit antennas, QPSK

The encoder takes m=2 input bits at each time. The trellis has four nodes corresponding to four states. There are four groups of symbols at the left of every node since there are four possible inputs (4-PSKconstellation). Each group has two entries corresponding to the symbols to be output through the two transmit antennas. At the top of the diagram we have the binary input bits that drive these symbols, which are output from the transmit antennas. These symbols come in pairs (for a two-antenna transmitter), where in the first digit corresponds to the symbol transmitted from antenna1 and the second from antenna2. The encoder is required to be in the zero state at the beginning and at the end of each frame. Beginning at state 0, if the incoming two bits are 10, the encoder outputs a 0 on antenna1 and a 2 on antenna2 and changes to state 2. Thereafter, it waits at state 2 for the next number. If the next incoming two bits are 01, the encoder outputs a 2 on antenna1 and a 1 on antenna2 and changes to state 1 and so on. In other words, the output and the next state is a function of the present state and the present input bits.

Let us take an example of STTC encoding for 4 PSK 4 State code with two transmit antennas.

Example 1:

Let the randomly generated bits be:
Binary data = 0111100001
Corresponding Symbols for 4 PSK, x = 1 3 2 0 1
Number of transmit antennas = 2

Generator Matrix, $G = \begin{bmatrix} 2 & 0 \\ 1 & 0 \\ 0 & 2 \\ 0 & 1 \end{bmatrix}$

Note: The first column of generator matrix is for first transmit antenna and second column for second transmit antenna.

c1 and c2 are the output symbols on antenna 1 and antenna 2.

c_1 = 0 1 3 2 0
c_2 = 1 3 2 0 1

An actual encoded sequence can be represented as a path on this trellis diagram. Where x is the input symbol and c1 and c2 are the symbols transmitted from transmit antenna 1 and transmit antenna 2 respectively. Here, the initial state is taken to be state 0, as is the normal practice. The first symbol is 1. Referring to figure 5, we see that for present state 0 and input symbol 1 (bits: 01) the next state is 1 with an output of c1: 1, c2: 0, the second symbol is 3. From figure 5, we can see that the output for present state 1 and input symbol 3, the next state is 3 and the output is c1: 3, c2: 1. Figure 6 shows the full path for the given input in the example 1 in dark thick line.

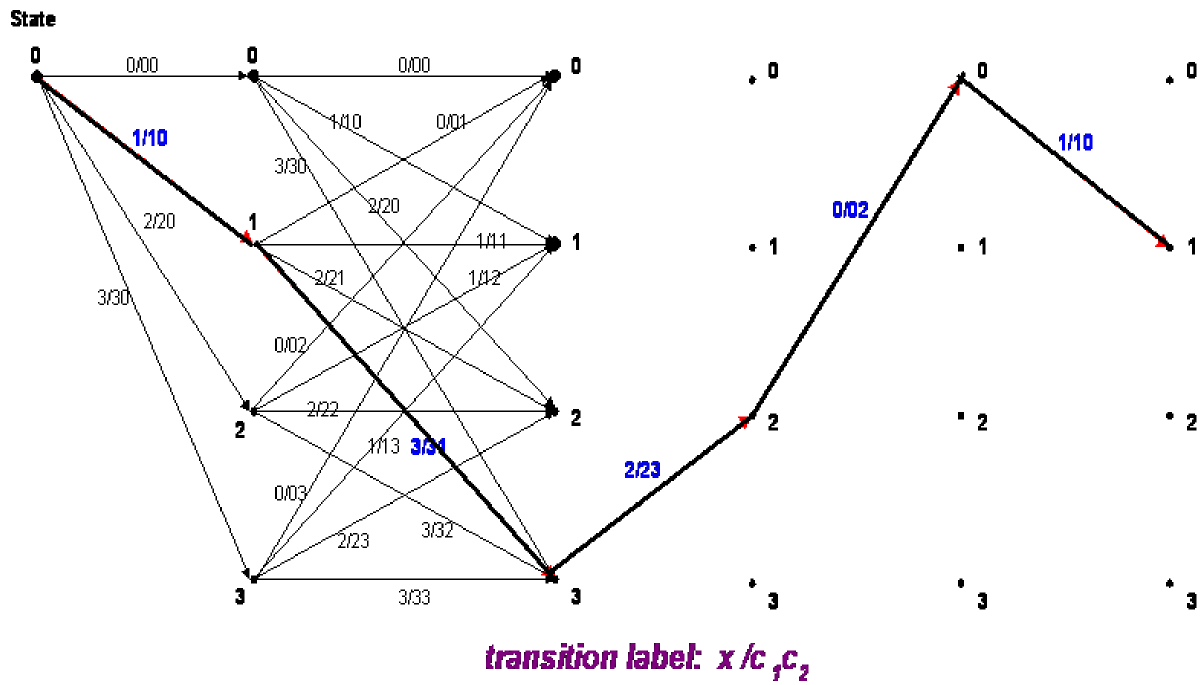

Figure 4: Trellis Diagram for example 1

2.4 Generator Matrices

A code is a finite dimensional vector space over a finite field, it only has finitely many elements. To represent a code in a computer one may of course store all the elements. A simple way of representing a code is *Generator Matrix*. It can be defined for a linear code *C* as a matrix *G* whose rows form a basis for *C*. This section provides the entire set of generator matrices used for different space time trellis codes in our simulations followed by a brief discussion about their representation. The different matrices are given in the tables shown below.

v	Generator sequences	Generator sequences
	g^1[k,k-1,..]	g^2[k,k-1,..]
2	[(0,2) , (2,0)]	[(0,1) , (1,0)]
3	[(0,2) , (2,0)]	[(0,1) , (1,0) , (2,2)]
4	[(0,2) , (2,0) , (0,2)]	[(0,1) , (1,2) , (2,0)]
5	[(0,2) , (2,2) , (3,3)]	[(0,1) , (1,1) , (2,0) , (2,2)]

Table1: Coefficient pairs for 4PSK, 4-, 8-, 16-, and 32- State for 2 Transmit Antennas, STTC

v	Generator sequences	Generator sequences	Generator sequences
	g^1[k,k-1,..]	g^2[k,k-1,..]	g^3[k,k-1,..]
3	[(0,4) , (4,0)]	[(0,2) , (2,0)]	[(0,1) , (5,0)]
4	[(0,4) , (4,4)]	[(0,2) , (2,2)]	[(0,1) , (5,1) , (1,5)]

Table 2: Coefficient pairs for 8PSK, 8- and 16- State for 2 Transmit Antennas, STTC

v	Generator sequences	Generator sequences	Generator sequences	Generator sequences
	g^1[k,k-1,..]	g^2[k,k-1,..]	g^3[k,k-1,..]	g^4[k,k-1,..]
4	[(0,8) , (8,0)]	[(0,4) , (4,0)]	[(0,2) , (2,0)]	[(0,1) , (1,0)]

Table 3: Coefficient pairs for 16PSK, 16- State for 2 Transmit Antennas, STTC

v	Generator sequences	Generator sequences
	g^1[k,k-1,k-2,..]	g^2[k,k-1,k-2,..]
2	[(0,2,2) , (2,0,0)]	[(0,1,1) , (1,0,0)]
3	[(0,2,2) , (2,0,0)]	[(0,1,1) , (1,0,0) , (2,2,2)]
4	[(0,2,2) , (2,0,0) , (0,2,2)]	[(0,1,1) , (1,2,2) , (2,0,0)]

Table 4: Coefficient pairs for 4 PSK, 4-, 8-, 16- State for 3 transmit Antennas, STTC

The Generator matrix therefore can be represented as:

$$G = \begin{bmatrix} \vdots \\ g^1(k-2) \\ g^2(k-2) \\ g^1(k-1) \\ g^2(k-1) \\ g^1(k) \\ g^2(k) \end{bmatrix}$$

where the elements g^i are taken from the MPSK constellation, k being the time instant . Each Generator matrix G has the dimensions of $(m+s) \times n_T$, where $m = log_2 M$ represents the number of information bits transmitted, s represents the number of shift registers in the encoder, and n_T represents the number of transmit antennas. The elements of this matrix define the coefficient pairs described earlier in the encoder structure. For example Generator matrix, for 4PSK 4State case is:

$$G = \begin{bmatrix} g^1(k-1) \\ g^2(k-1) \\ g^1(k) \\ g^2(k) \end{bmatrix} \text{i.e., can be written as } G = \begin{bmatrix} 2 & 0 \\ 1 & 0 \\ 0 & 2 \\ 0 & 1 \end{bmatrix}$$

2.5 Channel Model

For our simulations we have considered two kind of channel model, slow-fading and fast fading channel. We started with slow fading also known as Quasi-static channel with multiple antennas having n_T transmit and n_R receive antennas, for which the received signal matrix Y is given by:

$$\boldsymbol{Y = H.C + N}$$

Where C is the transmitted signal matrix given by $C = [c_1, \dots c_t, \dots c_L]$; L being the frame length or total number of symbols ; $c_t = [c_t^1, c_t^2, \dots c_t^{nT}]$ is the signal vector at time $1 \leq t \leq L$; H is the channel matrix of dimension n_R x n_T . For Quasi-static case it remains constant during a frame of length L channel uses and varies from one frame to another independently. For fast fading case it changes for each symbol in a frame independently. N is the matrix of random complex noise of dimension n_R x L; Hence R, the received signal matrix also has the same dimension n_R x L. It can be give in form of equation as follows:

$$y_{nR}(t) = \sum_{i=1}^{nT} h_{nRi}(t).c_i(t) + n_{nR}(t)$$

Details about MIMO channel can be found later in section 3.

2.6 Decoding of Space Time Trellis Codes

The decoding of Space time trellis codes is based on **maximum likelihood sequence detection**. We assume that channel state information is available to the receiver. The signals at the receiving end can be given by the following equation from section 2.5 as:

$$y_{nR}(t) = \sum_{i=1}^{nT} h_{nRi}(t).c_i(t) + n_{nR}(t)$$

Hence, it can be expressed in matrix form as follows,

$$\begin{bmatrix} y_1(t) \\ y_2(t) \\ y_3(t) \\ \vdots \\ y_{n_R}(t) \end{bmatrix} = \begin{bmatrix} h_{11}(t) & h_{21}(t) & h_{31}(t) & \cdots & h_{n_{T_1}}(t) \\ h_{12}(t) & h_{22}(t) & h_{32}(t) & \cdots & h_{n_{T_2}}(t) \\ h_{13}(t) & h_{23}(t) & h_{33}(t) & \cdots & h_{n_{T_3}}(t) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h_{n_{R_1}}(t) & h_{n_{R_2}}(t) & h_{n_{R_3}}(t) & \cdots & h_{n_T n_R}(t) \end{bmatrix} \begin{bmatrix} c_1(t) \\ c_2(t) \\ c_3(t) \\ \vdots \\ c_{n_R}(t) \end{bmatrix} + \begin{bmatrix} n_1(t) \\ n_2(t) \\ n_3(t) \\ \vdots \\ n_{n_R}(t) \end{bmatrix}$$

where receiving a sequence Y is given by,

$$Y = \begin{bmatrix} y_1(1) & y_1(2) & \cdots & y_1(L) \\ y_2(1) & y_2(2) & \cdots & y_2(L) \\ \vdots & \vdots & \vdots & \vdots \\ y_{n_R}(1) & y_{n_R}(2) & \cdots & y_{n_R}(L) \end{bmatrix}$$

And the transmitted code matrix C is given by

$$C = \begin{bmatrix} c_1(1) & c_1(2) & \cdots & c_1(L) \\ c_2(1) & c_2(2) & \cdots & c_2(L) \\ \vdots & \vdots & \vdots & \vdots \\ c_{n_T}(1) & c_{n_T}(2) & \cdots & c_{n_T}(L) \end{bmatrix}$$

The decoder employs viterbi algorithm to compute the maximum likelihood path through the trellis. It is assumed that perfect channel state information is available at the receiver. For a branch labeled by the noiseless symbol s_t , the branch metric is computed as the squared Euclidean distance between all possible noiseless symbols and the actual received symbols. The viterbi algorithm selects the path with the minimum path metric as the decoded sequence.

Viterbi Decoding

Considering we have a trellis diagram of length L(L being the number of symbols) with finite number of states, viterbi decoding algorithm finds the most likely path through it. The viterbi decoding algorithm consists of two major steps.

In first step we traverse the trellis forward i.e., from left to right and find the path metric. Using *y* as the received values and *s* the assumed possible noiseless symbols or values, we perform the

branch metric calculation which is nothing but the euclidean distances of *y* to *s:*

$$Branch\ Metric\ Value = \sum_{j=1}^{nR} \left| y_j(l) - \sum_{i=1}^{nT} h_{ij}(l) s_i(l) \right|^2$$

We calculate Branch Metric Value (BMV) for each state transition, and repeat it through the trellis (from left to right) until the whole frame has been calculated. Once all the branch metric values have been calculated we find the path metric values. Every trellis section, has certain number of paths (four, in the case of a 4 PSK code) leading into each node, out of which one is the survivor path and rest are the competitor paths.

At the beginning of the trellis, the Path Metric Value (PMV) values are taken the same as the BMV values leading to a node. Next, we find the local survivor by adding the BMV value of a path entering a node to the PMV value from the previous node connected to that path to form a competitor path. This procedure is then repeated for all available paths(four paths in case of 4 PSK code) and the minimum of these values is used to form the new node PMV value, if there is more than one value that is the minimum, then one is chosen arbitrarily. This procedure is then repeated for each node throughout the trellis, working from left to right.

Once these values have been calculated, we move towards the next step of trace back operation which identifies the most likely path through the trellis based on low PMV values. This operation is performed from the end of the trellis to the start (i.e., right to left) and so can only be performed when an entire frame has been received.

The operation begins with the selection of the smallest PMV for the end/last set. The path backwards (the overall survivor path) to the start of the trellis is calculated based on the selection of the smallest of all the PMV's joining the current node, if there exist more than one PMV with the smallest value then one is chosen arbitrarily. This path is then stored as the modulo-m (m is 4 for 4 PSK, 8 for 8 PSK and so on) value of the systematic output corresponding to the selected path. This stored encoder output sequence is then reversed in order (symbol by symbol) and becomes the decoded data.

2.7 Complexity of Decoder

In this chapter, we discuss the decoding complexity of the STTC. The STTC decoder as we know does trellis based decoding employing viterbi algorithm, which computes a decision metric for every transition. The complexity of decoder can be given as product of number of states, number of branches per state and the complexity of calculating the branch metric value.

For a trellis of length L, the number of states, S is a multiple of the modulation parameter M, which is equal to 2^m, where m is the information bits per channel use. Hence, number of states can be given by $S = A.2^m$, A being a constant integer. From each state, 2^m different transitions take place i.e., number of branches per state, $B = 2^m$. The branch metric calculation complexity can be given by $n_T \times n_R$. Therefore, the total complexity of STTC decoder is given by $A.2^{2m} \times n_T \times n_R$.

In general, we can represent the decoding complexity of STTC, designed for an M-ary constellation in order of $S \times B$ which is $A.2^{2m}$.It can be seen that the decoding complexity grows exponentially with number of states of the trellis structure. Table 5 shows the decoding complexity order of 4-PSK STTC for increasing number of states.

Number of States	*Number of Branches per state*	*Complexity order*
4	4	16
8	4	32
16	4	64
32	4	128
64	4	256
128	4	512

Table 5: Complexity order of 4-PSK STTC decoder

2.8 QR Decomposition

A trellis-coded multiple-input multiple-output (MIMO) transmission technique, which exploits multiple-antenna elements at both transmitter and receiver sides and employs trellis-coded modulations (TCMs), has potential to significantly increase spectral efficiency in wireless communications. At the receiver, an adaptive equalizer based on maximum-likelihood sequence estimation (MLSE) deals with inter-symbol interference incurred in transmissions and jointly decodes multiplexed TCM signals. To do this, viterbi algorithm is used, which performs branch metric computations. As it is explained in previous section 2.7, the complexity of STTC decoder algorithm grows exponentially as we increase the number of states and hence the branch metric computation also increases eventually. To reduce such growing computations, we use **QR decomposition**, which is a method to decompose a matrix into an orthogonal, Q and an upper triangular matrix, R. In our case QR decomposition is applied to the channel matrix H.

We calculate the BMV by following equation:

$$Branch\ Metric\ Value = \sum_{j=1}^{nR} \left| y_j(l) - \sum_{i=1}^{nT} h_{ij}(l) s_i(l) \right|^2$$

In vector form it can be written as : $BMV = \|Y - H.S\|^2$

Where, Y is the received values, H is the channel matrix realization and S are the possible transmitted values.

In QR decomposition we decompose Channel Matrix, H in Q and R matrices.

$$BMV = \|Y - H.S\|^2$$

$$= \|Y - Q.R.S\|^2$$

$$= \|Y.Q^T - Q.Q^T.R.S\|^2$$

$$= \left\|\vec{Y} - R.S\right\|^2$$

Hence, we see that the final equation reduces to: $Branch\ Metric\ Values = \left\|\vec{Y} - R.S\right\|^2$, where $\vec{Y}$the multiplication of received values and Transposed Q matrix, R is is the upper triangular matrix and S is the matrix containing possible transmitted symbols. This reduces the total number of computations in branch metric calculation, since half the elements in *R* matrix are zero, thus reducing the simulation run time.

3 Communication Channel Model

The figure below shows the antenna configuration used in defining space-time systems. In Section 2.5, we briefly discussed about the channel model used in STTC. This section presents the channel model in detail. In MIMO systems, we discuss about receive antenna diversity or transmit antenna diversity. Antenna Diversity or most commonly known as Spatial Diversity is one of the most popular forms of diversity used in wireless communication system. It relies on the fact that fading is different at different points on the earth, so two aerials a few wavelengths apart will have uncorrelated fading. Hence, it combats fading by transmitting copies of original signal through uncorrelated paths to receiver. This can be provided with no penalty in bandwidth efficiency. To gain receive antenna diversity, receiver has multiple antennas that receive multiple replicas of the same transmitted signal. If the signal path between each antenna pair fades independently, then when one path is in a fade, it is highly unlikely that all the other paths are also in fade. Therefore, the loss of signal power due to fade in one path is countered by the same signal but received through a different path. If the diversity is high we can combat fading in a channel.

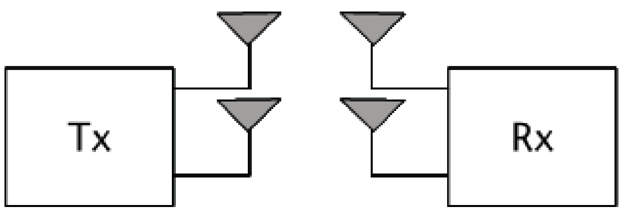

Figure 5: Multiple Input Multiple Output in space time systems

3.1 MIMO Channel

We consider a MIMO system with a transmit array of n_T antennas and a receive array of n_R antennas. The diagram of such a system is shown in the figure given below.

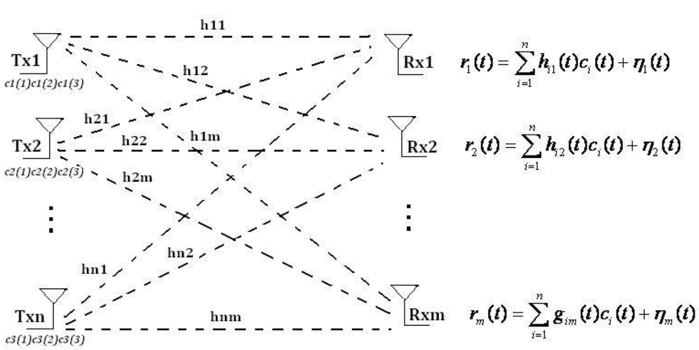

Figure 6: Block diagram of MIMO channel

Assuming that $c_i(t)$ is the transmitted signal from from antenna i at time t, the received signal $y_j(t)$ at the jth antenna corresponding to this time interval is given by:

$$y_j(t) = \sum_{i=1}^{nT} h_{i,j}(t).c_i(t) + n_j(t)$$

Where $i = 1,2, \ldots n_T$, $j = 1,2, \ldots n_R$, $t = 1,2, \ldots L$, and L is the frame length; $h_{i,j}$ is the complex channel coefficient between i-th transmit and j-th receive antennas. Thus, the channel is Rayleigh fading where the channel gains are constant over an entire frame of length L(*i.e.*, quasi-static) and are independent across different sub-channels; $n_j(t)$ is the additive Gaussian noise sample at j-th receive antenna at time t.

Hence, the signals at the receiving antennas can be expressed in matrix form as:

$$\begin{bmatrix} y_1(t) \\ y_2(t) \\ y_3(t) \\ \vdots \\ y_{n_R}(t) \end{bmatrix} = \begin{bmatrix} h_{11}(t) & h_{21}(t) & h_{31}(t) & \ldots & h_{n_{T_1}}(t) \\ h_{12}(t) & h_{22}(t) & h_{32}(t) & \ldots & h_{n_{T_2}}(t) \\ h_{13}(t) & h_{23}(t) & h_{33}(t) & \ldots & h_{n_{T_3}}(t) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h_{n_{R_1}}(t) & h_{n_{R_2}}(t) & h_{n_{R_3}}(t) & \ldots & h_{n_T n_R}(t) \end{bmatrix} \begin{bmatrix} c_1(t) \\ c_2(t) \\ c_3(t) \\ \vdots \\ c_{n_R}(t) \end{bmatrix} + \begin{bmatrix} n_1(t) \\ n_2(t) \\ n_3(t) \\ \vdots \\ n_{n_R}(t) \end{bmatrix}$$

Where:

$c_i(t)$ is the modulation symbol transmitted by antenna i at the time instant t. It is generated by a space-time encoder.

h_{ij} is the path gain from transmitted antenna i to receive antenna j.

$\eta_j(t)$ is an independent Gaussian random variable for noise addition

So we can note, that at receiver the signal at each antenna is noisy superposition of n_T transmitted signals degraded by channel fading. In our simulation system both cases of quasi-static and fast fading channel have been considered.

3.2 Quasi-Static Channel

A quasi-static channel also known as slow fading channel, is the one where the rate at which the magnitude and phase imposed by the channel on signal changes slowly. The coherence time of the channel, which is a measure of the minimum time required for the magnitude change of the channel to become uncorrelated from its previous value, is relatively large as compared to the delay constraint of the channel. In our simulations, for such kind of channel, the channel response matrix, H is generated for every frame of data i.e., the channel remains constant for the entire frame of data. As we see that the channel is remaining constant for fixed time for over one frame, so it should give better performance in terms of coding gain as compared to fast fading where channel changes very frequently.

3.3 Fast Fading Channel

A fast fading channel is one where fading is more rapid. Here, the coherence time of the channel is relatively small as compared to the delay constraint of the channel. For such kind of channel, the channel response matrix, H is generated for every received symbol. This means the fading is more rapid in this case and hence it gives a relatively poor performance as compared to the case where fading is slow.

3.4 Capacity

The capacity of a system is defined as the maximum possible transmission rate such that the probability of error is arbitrarily small. We assume that the channel knowledge is unavailable at the transmitter and known only at the receiver.

The capacity of MIMO channel is defined as:

$$C = max_{f(s)} I(s; y)$$

Where *f(s)* is the probability distribution of the vector *s* and *I(s;y)* is the mutual information between vector *s* and *y*.

Using the relation between mutual information and entropy:

$$\begin{aligned} I(s: y) &= h(y) - h(y|s) \\ &= h(y) - h(Hs + n|s) \\ &= h(y) - h(n|s) \\ &= h(y) - h(n) \end{aligned}$$

Where *h(.)* denotes the differential entropy. The Mutual information is maximized when *y* is Gaussian, since the normal distribution maximizes the entropy for a given variance. Hence, after maximizing we get the final ergodic (mean) capacity of MIMO channel which is give by equation:

$$C = \varepsilon_H \left\{ log_2 \left[\det \left(I_{n_R} + \frac{\rho}{n_T} HH^+ \right) \right] \right\} bits/s/Hz \,, \rho = \frac{P_T}{\sigma^2}$$

Where P_T is the total transmit power irrespective of the number of transmit antennas, *H* being the channel matrix of dimension n_T x n_R , ε being the covariance matrix.

Outage Capacity

It is always better to compare a simulation result for these codes with the channel capacity. Outage capacity is the capacity that is guaranteed with a certain level of reliability. In the quasi-static channel, the suitable capacity is outage since it is non-ergodic i.e., the randomness of the channel gain cannot be averaged out over time. Outage occurs when the channel is so poor that no scheme can communicate reliably at a certain fixed data rate. The largest rate of reliable communication at a certain outage probability is called the outage capacity. A coding scheme that achieves the outage capacity is said to be universal since it communicates reliably over all slow fading channels that are not in outage. In fast fading channels, by contrast, outage can be avoided due to the ability to average over the time variation of the channel, and one can define a positive capacity at which arbitrarily reliable communication is possible.

We define p% outage capacity as the information rate that is guaranteed for ($100-p$) % of the channel realizations H.

4 Simulation System Model

This section describes the software developed. Here, we consider each module of the simulation system model and explain which piece of code handles that particular function. . There are 6 parts in the simulation model – simulation parameters, source generation, encoding, channel, decoding and performance. Simulation parameters takes an input specification text file containing all constant parameters required for simulation, extracts them and uses them in other parts, source generation generates the random input bits for simulation, encoding encodes the bits and prepares it for transmission, channel warps the bits as a channel would, the decoder will decode the transmitted symbols and the performance evaluates the FER and BER for that code.

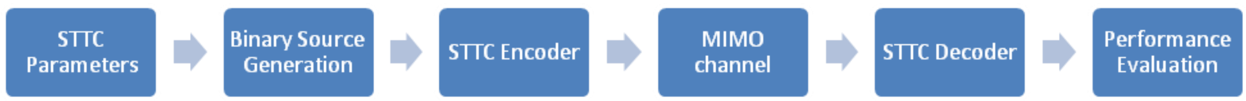

Figure 7: Block diagram of Simulation Model

The simulation model is developed on Eclipse toolset, in C++ programming language, using IT++ libraries [8]. IT++ provides classes with a lot of general purpose signal processing functions and implementations. It also provides classes like vector and matrix which are very helpful in signal processing.

Apart from the classes mentioned below, the simulation model also uses classes already developed, namely “ChannelPreProc”, “MimoChannel” and “buffer”. ChannelPreProc was used in our system model for QR decomposition during the decoding. MimoChannel was used to simulate the channel. Buffer class is a general purpose data class that was used to make data “connections” between the encoder, channel and decoder classes. A data connection is the term used to describe the situation when the encoder, decoder and channel use common memory location to access data. For example, the source bits are generated and stored in a particular location. These bits are not sent as parameters to the encoder. But, the encoder is connected to the source – the encoder is given the memory location where the source generator will put the bits. Every time the encoder’s encode function is called, it will read the bits from this location and encode them.

4.1 STTC Source Generation

For Space time trellis code the source generation is defined by files '*BinarySource.h*' and '*BinarySource.cpp*' . This class generates the source as binary data for the every frame over different SNR values for the Space time trellis encoder

Constructor:

BinarySource(int NumberBits)

NumberBits are the total number of bits in a frame.

Functions:

void Generate() : It generates a set of binary data for given number of bits

BufferClass<int> *getOutData( ) : It is an output buffer to get the generated binary set of uncoded information bits.

4.2 STTC Parameters

All the input parameter required for Space time trellis code design are defined in this class. These parameters count for the flexibilty of the STTC. The two files which describe them are '*STTCParameters.h*' and ' *STTCParameters.cpp*' . This class reads a input text file containing all required STTC parameters, extracts them and saves them as public constant variable so that it can be used anywhere outside the class.

Constructor:

STTCParameters(char *fname)

fname is the filename of the text file saved in the same directory which is to be read.

The Simulation parameters defined in Input specification text file are: Dimension of generator matrix, generator matrix, modulation parameter for eg. 4 for QPSK, number of states, number of receive antennas and number of transmit antennas.

4.3 STTC Encoder

The space-time encoder maps the raw information bits into space-time symbols based on the trellis diagram, as described in chapter 2.3. The encoder takes L = 130 symbols (one frame) from the MPSK signal constellation and encodes them into an (L x n_T) matrix of complex symbols where n_T is the number of transmit antennas. This mapping procedure is accomplished through the encoder structure described in chapter 2.3.

At the beginning and end of each frame, the encoder is required to be in state 0. The encoding algorithm then loops through each pair of input symbols and determines the output for each antenna based on those current inputs and the current state. Then the next state is determined based on the current input.

The two files describing encoding of STTC are '*STTCEncode.h*' and '*STTCEncode.cpp*'. Hence, this class encodes the binary data from BinarySource class, does the serial to parallel conversion, and modulates the encoded output and transmits it via transmit antennae.

Constructor:

STTCEncode (int TotalBits,int NSymbols, STTCParameters STTCParam)

TotalBits are the total number of bits in a frame, *NSymbols* are total number of symbols to be transmitted and *STTCParam* is object of the class STTCParameters.

Functions:

void GenerateMappingTable() : This function generates the table containing all the complex symbols for M-PSK .

mat Serial2Parallel(Vec<int> input, int parallel_data_size, int number_of_bits) : This function converts the incoming stream of data from serial to parallel.

mat Parallel2Serial(mat input, int parallel_value, int bps, int bits) : This function converts the incoming parallel data to serial.

cmat MPSK_Symbol_Mapper(mat symbol_matrix, int psk) : This function maps the encoded symbols to the corresponding PSK symbols.

void encode() : This function performs the space time trellis encoding and modulation and transmits the modulated encoded symbols.

void SetInputData(BufferClass<int> *InputDataBuffer) : This function connects the input buffer in STTCEncoder class to the output buffer *getOutData()* of BinarySource class which contains the randomly generated uncoded input values.

BufferClass<complex<double>> *getOut_Data() : It is an output buffer containing the modulated encoded symbols.

4.4 STTC Decoder

The decoding procedure is based on the well-known Viterbi algorithm. However, for space-time codes the Viterbi decoder is modified from the conventional convolutional decoder so that the branch metric is computed from the complex inputs and the Channel State Information. The Viterbi decoder is then used to calculate the path through the trellis diagram with the lowest accumulated metric.

Assuming that *Y* is the actual received symbol vector, the branch metric is determined by:

$$Branch\ Metric\ Value = \|Y - H.S\|^2$$

Where S is the possible noiseless symbol vector and H is the channel matrix.

In these simulations, at every time unit of the trellis, a survivor is determined for each state from the minimum path metric. The state from which the minimum path metric originated is saved into a state predecessor (or state history) table. Once the end of the trellis is reached, the decoder can begin to determine the sequence of bits that were input into the space-time encoder. First, beginning at the end of the trellis, select the state with the smallest total metric and save that state into a state sequence table (if we assume that the trellis begins and ends in state 0, then this initial chosen state would be state 0). The decoder iteratively performs the following traceback procedure until the beginning of the trellis is reached: using the state predecessor table, for the selected state, determine a new state which is the predecessor to that state. Save the state number of that selected state onto the state sequence table. Once the trellis is exhausted, the completed state sequence table will show the state transitions taken by the final survivor. The decoder can now work forward through the state sequence table to determine what input bits led to the transition from one state to the next.

The two files describing encoding of STTC are '*STTCDecode.h*' and '*STTCDecode.cpp*'. Hence,this class generates all possible transmitted symbols from generator matrix performs viterbi decoding via maximum likelihood sequence detection.

<u>Constructor:</u>

STTCDecode (int Number_of_bits, STTCParameters STTCParam)

Number_of_bits are the total number of bits in a frame and *STTCParam* is object of the class STTCParameters.

<u>Functions:</u>

void GenerateMappingTable() : This function generates the table containing all the complex symbols for M-PSK .

cmat SerialToParallel(Vec<complex<double>> ch_val, int numtx, int symbol_length) : This function converts the serial incoming stream of complex data to parallel stream of complex data.

void decode() : This function generates all the possible transmitted symbols from generator matrix, calculates branch metric value and does viterbi decoding to get the correct decoded binary values.

void SetoutChannelValues(BufferClass<complex<double> > *InputDataBuffer1) : This function connects the input buffer in STTCDecoder class to the output buffer *getOutChannelValues()* of MimoChannel class which contains the received channel values.

void SetoutChannelMatrix(BufferClass<complex<double>, 2 > *InputDataBuffer2) : This function connects the input buffer in STTCDecoder class to the output buffer *getOutChannelMatrix()* of MimoChannel class which contains the channel matrix realizations.

BufferClass<int> *getDecodedValues() : It is an output buffer containing the decoded binary values.

4.5 STTC Performance

After the encoding and decoding is done we have the set of both uncoded and decoded data. This class gets these uncoded and the decoded values and compares them to check the performance of the space time trellis codes for different conditions and calculates the parameter's like Frame error rate and Bit error rate which helps in deciding the performance results of STTC. For performance the two files are '*STTCPerformance.h*' and '*STTCPerformance.cpp*'.

<u>Constructor:</u>

STTCPerformance(int frame, int snrlength, STTCParameters STTCParam)

frame is the total number of frames, *snrlength* is the total snr values and *STTCParam* is the object of the class STTCParameters.

<u>Functions:</u>

void SetUncodedData(BufferClass<int> *InputDataBuffer1) : This function connects the input buffer in STTCPerformance class to the output buffer *getOutData()* of BinarySource class which contains the randomly generated binary values.

void SetDecodedData(BufferClass<int> *InputDataBuffer2) : This function connects the input buffer in STTCPerformance class to the output buffer *getDecodedValues()* of STTCDecoder class which contains the decoded binary values.

void setframe(int frameNo) : This function sets the current frame number.

void setsnrlength(int SNR_length) : This function sets the current number of signal-to-noise ratio.

bool EvaluatePerformance() : This function compares the uncoded bits from the input buffer *UncodedData* to the decoded bits from another input buffer *DecodedData* and calculates the bit error rate for the current frame and overall frame error rate for current snr value.

void ShowErrorData() : This function displays the the performance parameter values as well as writes all the performance data in a file and saves it as .m file, which is a matlab script so that it can be directly read in matlab to plot the different data.

5 Simulation Results

In this section, we show results pertaining to the performance of STTC on Rayleigh fading channels, based on computer simulations. In the simulations, it is assumed that the receiver has perfect Channel State Information. We evaluate the system and its performance through simulating examples of the STTC for 4-PSK, 8-PSK and 16-PSK constellations, using up to 3 transmit and 3 receive antennas. Concluding remarks and suggestions for future work are presented in section 6.

Throughout this section, we consider examples of STTC system designed for 4, 8 and 16 -PSK constellation. The model described in section 4 was derived for an arbitrary number of antennas. Here we simulate performance with up to 3 transmit and 3 receive antennas. We evaluate system performance and discuss the effects of number of states, transmit and receive diversity, increasing data rates, assuming the channel to be slow fading. Furthermore, we carried out a simulation for 4 PSK modulation scheme, taking the channel to be fast fading.

For all the simulations, each frame consist of 130 symbols transmitted out of each transmit antenna. Each symbol is driven by *m* number of information bits, where $m = log_2 M$ for M-PSK STTC. The curves are plotted against the signal to noise ratio (SNR). These simulations were executed with a maximum of 1000 frame errors for each value of SNR, with a total of 100000 frames per SNR value.

All the simulations were carried out assuming the channel to be quasi-static except for one case (h), referring to the simulations given below, where the channel was changed to fast fading.

The simulations for space time trellis were carried out for the following different cases:

(a) 4 PSK, 4- , 8-, 16- and 32- states, for a transmission rate of 2 bits/s/Hz over a quasi-static Rayleigh channel using two transmit and one receive antennas.

(b) 4 PSK, 4- , 8-, 16- and 32- states, for a transmission rate of 2 bits/s/Hz over a quasi-static Rayleigh channel using two transmit and two receive antennas.

(c) 4 PSK, 4- , 8-, 16- states, for a transmission rate of 2 bits/s/Hz over a quasi-static Rayleigh channel using three transmit and two receive antennas.

(d) 4 PSK, 4- , 8-, 16- states, for a transmission rate of 2 bits/s/Hz over a quasi-static Rayleigh channel using three transmit and three receive antennas.

(e) 8 PSK, 8- , 16- states, for a transmission of 3 bits/s/Hz over a quasi-static Rayleigh channel using two transmit and one receive antennas.

(f) 8 PSK, 8- , 16- , 32- states, for a transmission of 3 bits/s/Hz over a quasi-static Rayleigh channel using two transmit and two receive antennas.

(g) 16 PSK, 16- states for a transmission of 4 bits/s/Hz over a quasi-static Rayleigh channel using two transmit and two receive antennas.

(h) 4 PSK, 4- , 8-, and 16- states, for a transmission rate of 2 bits/s/Hz over a fast fading channel using two transmit and two receive antennas.

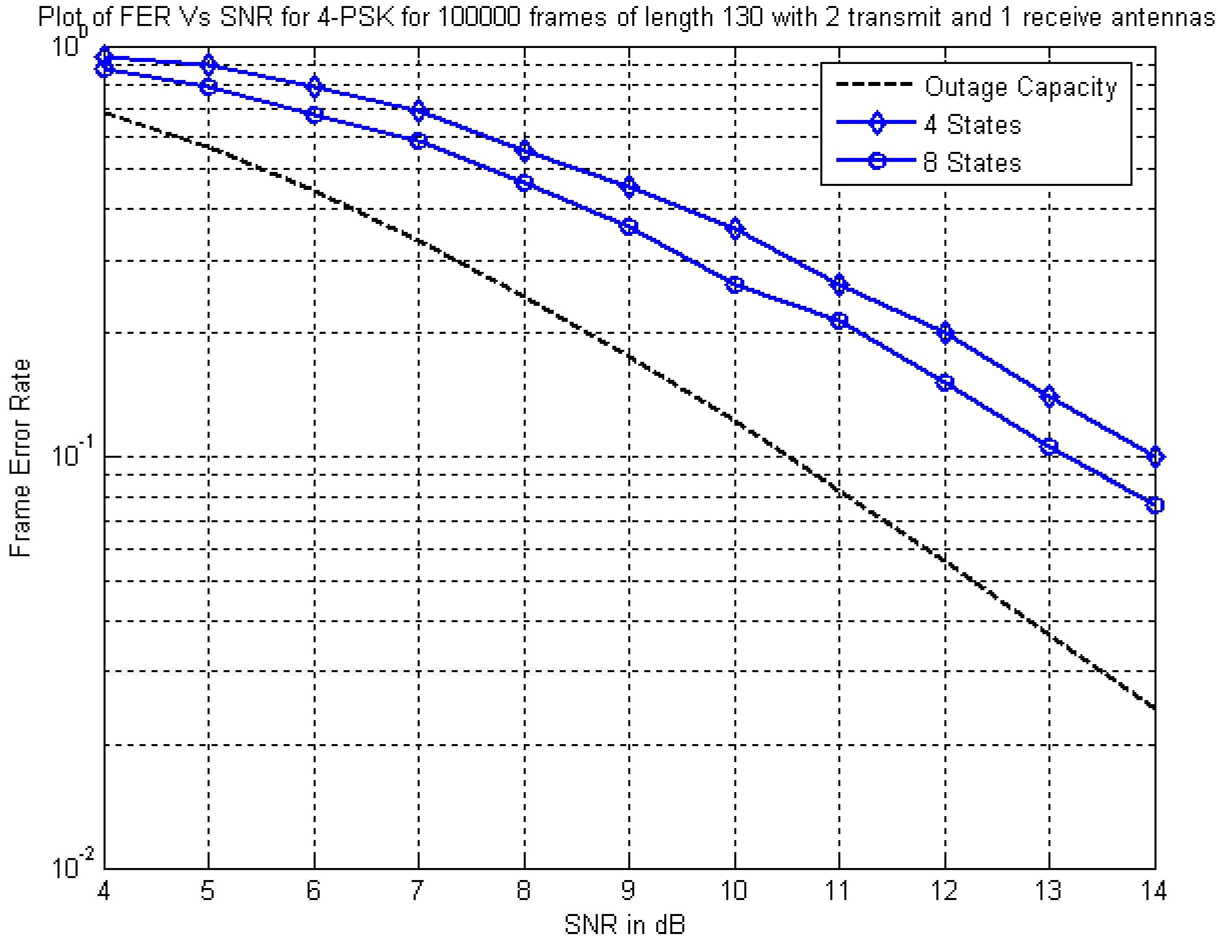

Figure 8: Performance of 4PSK code with two transmit and one receive antenna for quasi-static channel

Simulation Parameters:
Modulation: QPSK
States: 4 and 8
"Quasi-Static" Rayleigh Fading Channel
Frame Length: 130 Symbols
Perfect CSI at receiver
Number of Transmit Antenna: 2
Number of Receive Antenna: 1

Figure 8 shows the Frame Error Rate performance of QPSK STTC with a bandwidth efficiency of 2 bit/s/Hz in Slow Fading Rayleigh channel.We can see that 8-state QPSK codes are better than 4-state codes by almost 1 dB at a FER of 10^{-1} for two transmit antennas. It can be seen that as the number of states in the trellis increases, the coding gain increases and so does the performance. In general, the coding advantage of the above codes can be improved by constructing encoders with more number of states. For this case, the 10% outage capacity results states that we need 10.5dB SNR to achieve a transmission rate of 2 bits per channel use. On comparing this with SNR needed for 10% FER (about 13.2dB for 8 state code), we see that the performance of this STTC is only about 2.7dB away.

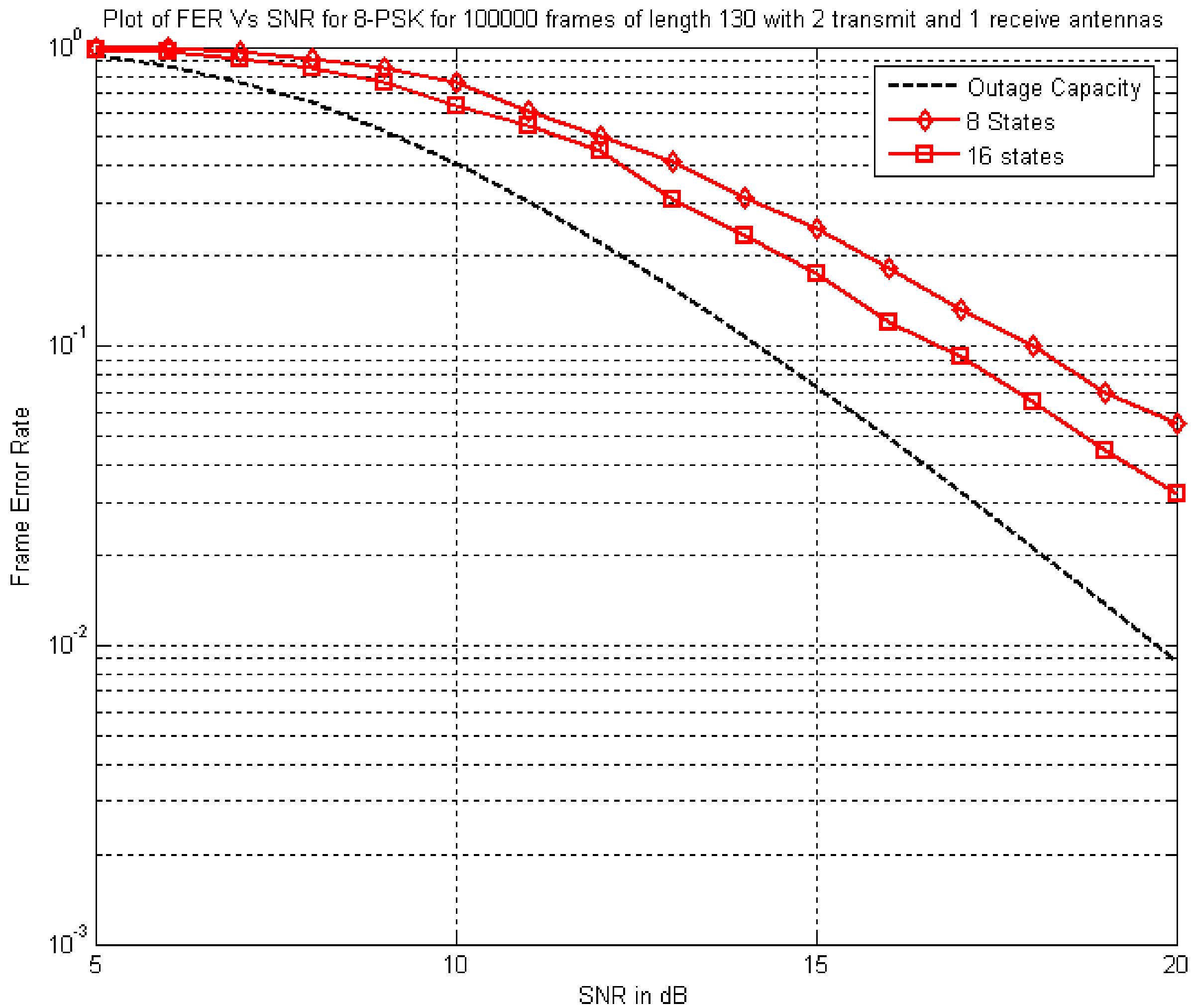

Figure 9: Performance of 8PSK code with two transmit and one receive antenna for quasi-static channel

Simulation Parameters:
Modulation: 8 PSK
States: 8 and 16
"Quasi-Static" Rayleigh Fading Channel
Frame Length: 130 Symbols
Perfect CSI at receiver
Number of Transmit Antenna: 2
Number of Receive Antenna: 1

Figure 9 shows the simulation results for an 8 PSK code. As expected, the simulation results show an improvement for increased number of states. The simulations showed an improvement of about 0.5 dB at an SNR of 10 dB from 8-state to 16-state coding. The 8 PSK codes appears to have poorer performance than 4 PSK code, this is because the former has a slightly improved bandwidth efficiency of 3 bits/s/Hz, while the latter has the bandwidth efficiency of 2 bits/s/Hz. 10% outage capacity results states that we need 14dB SNR to achieve a transmission rate of 3 bits per channel use. On comparing this with SNR needed for 10% FER (about 17dB for 16 state code), we see that the performance of this STTC is about 3dB away.

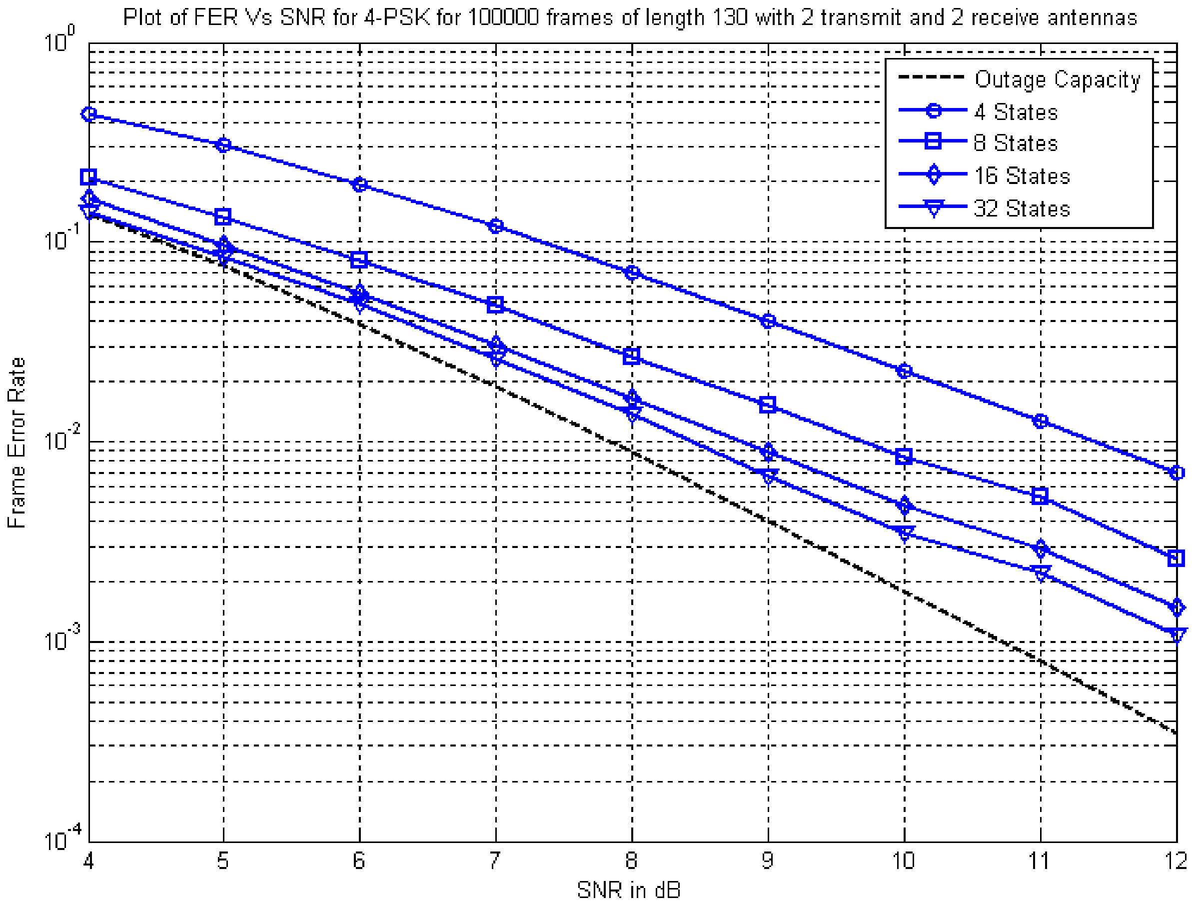

Figure 10: Performance of 4PSK code with two transmit and two receive antenna for quasi-static channel.

Simulation Parameters:
Modulation: QPSK
States: 4, 8, 16 and 32
“Quasi-Static” Rayleigh Fading Channel
Coherence Time: 130 Symbols
Perfect CSI at receiver
Number of Transmit Antenna: 2

Number of Receive Antenna: 2

Figure 10 shows the effect of increasing the number of states on the error performance of the STTC. The curve shows the performance of 4PSK STTC with 4 states, 8 states, 16 states and 32 states. For these simulations 2 transmit antennas, 2 receive antennas were used for a rate of 2 bits/sec/Hz. As can be seen, increasing the number of states on higher level improves the performance. This improvement is more pronounced at higher SNRs. We can be also see the effect of receive diversity in this figure providing a coding advantage of almost 7 dB by using two receive antennas instead of one. . 100% outage capacity results states that we need 7.9dB SNR to achieve a transmission rate of 2 bits per channel use, comparing this with SNR needed for 10% FER(about 8.9dB for 16 state code), we see that the performance of this STTC is about 1dB away.

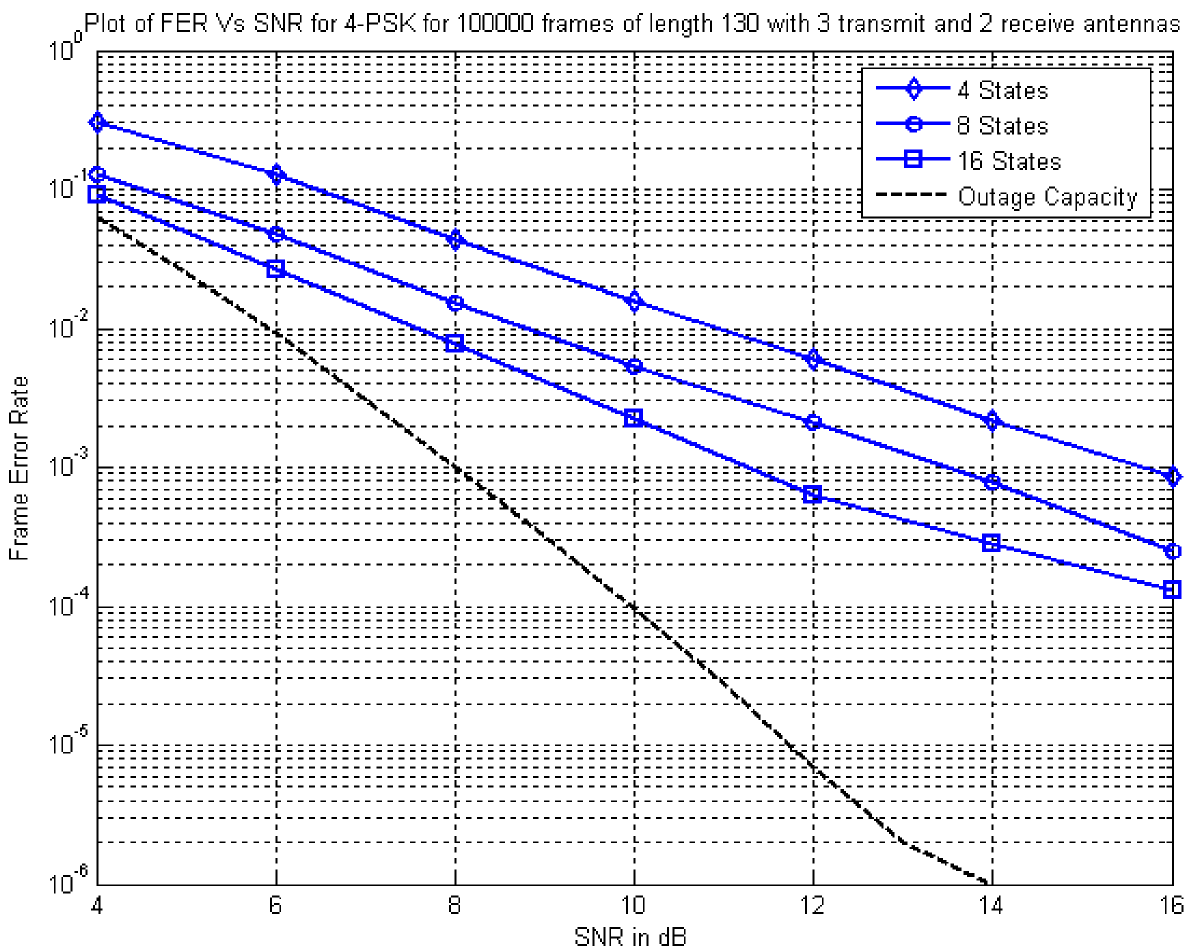

Figure 11: Performance of 4PSK code with three transmit and two receive antenna for quasi-static channel.

Simulation Parameters:

Modulation: QPSK

States: 4, 8 and 16

"Quasi-Static" Rayleigh Fading Channel
Coherence Time: 130 Symbols
Perfect CSI at receiver
Number of Transmit Antenna: 3
Number of Receive Antenna: 2

We can see from Figure 10 and Figure 11, that increasing the number of transmit antenna provides a coding gain of 1 dB at 10^{-1} FER for 4 state trellis code, thereby improving the performance. For 100% outage capacity, the performance is 1.6 dB away for 16 state code.

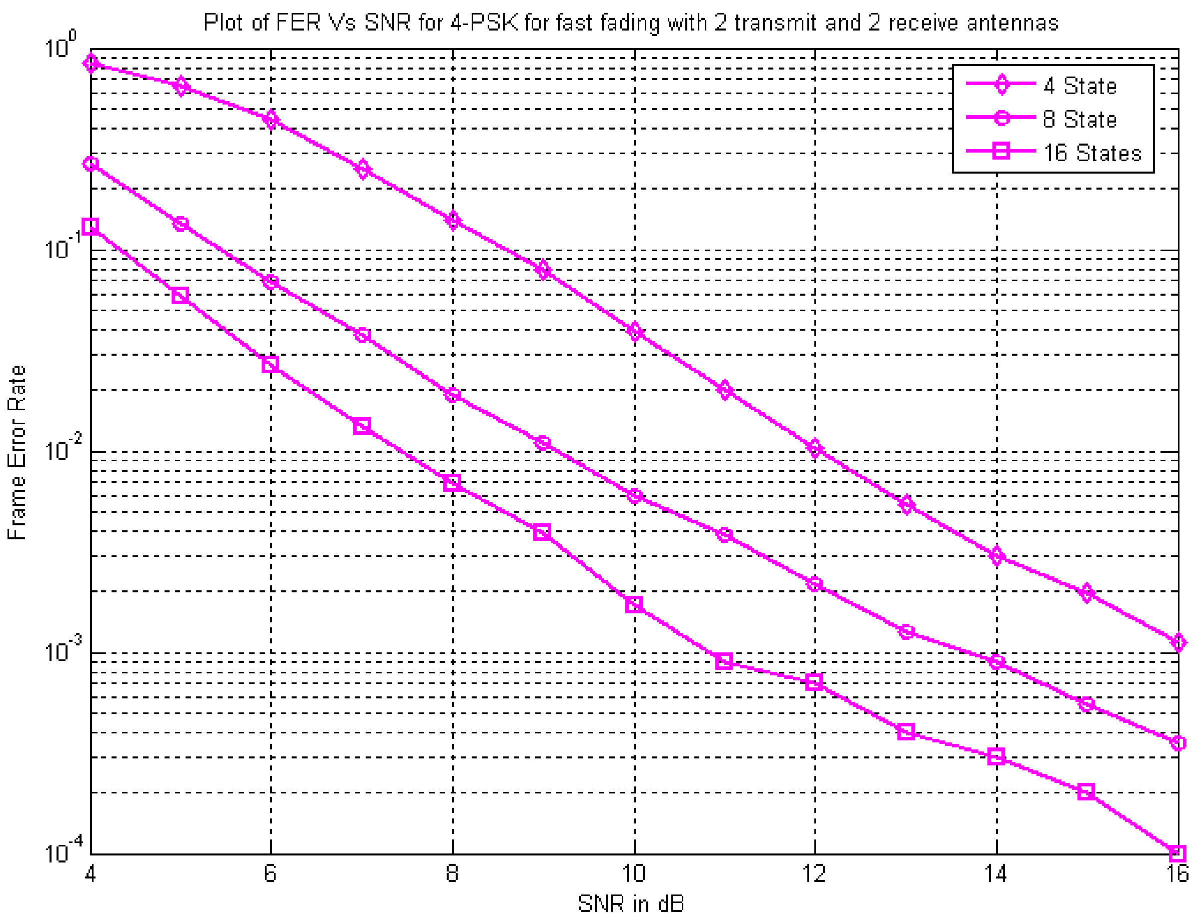

Figure12: Performance of 4PSK code with two transmit and two receive antenna for fast fading channel.

Simulation Parameters:
Modulation: QPSK
States: 4, 8 and 16
Fast Fading Channel
Perfect CSI at receiver
Number of Transmit Antenna: 2
Number of Receive Antenna: 2

As expected, fast fading channel gives a poorer performance as compared to slow fading channel, we can see in this case for QPSK with 2 transmit and 2 receive antennas. The same code is better for slow fading channel by 1 dB at 10^{-1} FER.

Summary

In this chapter we saw the simulation results for various STTC's codes and observed the factors affecting the performance of STTC's. We did the performance analysis over two channels, slow fading and fast fading. In both the channel the performance is improved by increasing the number of states. However, the performance was poor for fast fading channel since the channel is changing for every symbol.

In slow fading channel, we observed the effect by first keeping the number of transmit antennas constant and increasing the number of receive antennas. It provided a significant improvement. Later on we kept the number of receive antenna constant and increased the number of transmit antenna which again provided some improvement but not as good as the previous case. Also, we saw that as we improved the data rate, the performance went down.

6 Conclusion and Future Considerations

Space Time Trellis Codes are the error correcting codes which provides quality in transmission of data by offering reliability. Capacity in wireless communication systems has been rapidly growing world- wide. The main reason behind it is the increasing data rate requirements. As the available radio spectrum is limited, higher data rates along with error correction can only be achieved by designing more efficient signaling techniques. Space-Time trellis codes are able to achieve significant improvement by combining coding, modulation, spatial and temporal diversity techniques.

In this thesis we study and implement an encoder and a decoder for Space-Time Trellis Codes. The performance curves were derived for 4PSK, 4-, 8-, 16-, and 32-State codes, for the 8PSK, 8-, 16-, and 32-State and for 16 PSK, 16-State code for different antenna constellation. The signal was assumed to undergo flat Rayleigh fading through a quasi- static wireless channel as well as fast fading channel, with perfect channel state information available at the receiver. The results show that for m-PSK modulation, there is an improvement in error performance as the number of states increases. We note that 16-state QPSK trellis codes are superior to 4-state trellis codes by 2.4 dB at a FER of 10^{-1}. This means that the performance of 16-state code relative to 4-state code has improved. The conclusion here is that as the number of the states increases, the performance gain becomes larger. Figure 13 and Figure 15 shows that 4-state QPSK code performs better by 6.5 dB by increasing one receive antenna. Similarly, we observe a coding gain of 1dB at 10^{-1} FER from 4-state QPSK with two transmit antenna to 4-state QPSK with three transmit antenna. Thus, we see that there is a significant improvement as we introduce the redundancy in sending and receiving the signal by increasing the number of transmit and receive antennas. As the value of m increases i.e., as we move from 4 PSK to 8 PSK and so on we see that the performance goes down but the bandwidth efficiency improves. Hence, there is tradeoff between the increasing data rate and coding gain, where coding gain is the improvement in SNR at the receiver.

To improve spectral efficiency for future high data rate transmissions, it is desirable to construct STTCs with high order signal constellations. However, the design of a STTC normally involves the use of computer search, with the search space increasing exponentially with constellation size, the number of transmit antennas and the number of states in trellis code. A similar increase occurs in the decoding complexity of STTCs. Therefore, despite their many benefits, STTCs are still faced with reluctance from system designers when it comes to implementation, especially for systems which require the use of larger signal constellations or a larger number of antennas.

There are many possible directions for future work on this thesis. Here we highlight a few suggestions.

One possible work that can be done is extending the code for QAM modulation scheme. Another possible area would be to consider having access to channel information in the transmitter. If that assumption is made, one can use the information to improve the FER performance by setting the signal power on different antennas differently (based on the channel), helping them be easier to detect in the receiver. It has however been shown in the literature that if channel knowledge is not available in the transmitter the best approach is to spread the power equally among all antennas, which is what we are doing at the moment. Notice that space-time coded structures, in general, can benefit from having CSI at the transmitter.

7 List of References

[1] V. Tarokh, N. Seshadri, and A.R. Calderbank, "Space-Time Codes for High Data Rate Wireless Communications: Performance Criterion and Code Construction", IEEE Transactions on Information Theory, vol. 44, no. 2, pp. 744-765, March 1998.

[2] V. Tarokh, A. Naguib, N. Seshadri, and A. R. Calderbank, "Space-Time Codes for High Data Rate Wireless Communication: Performance Criteria in the Presence of Channel Estimation Errors, Mobility, and Multiple Paths", IEEE Transactions on Communications, vol. 47, no. 2, pp. 199-207, February 1999.

[3] A.Roger Hammons, and Hesham El Gamal, "On the Theory of Space –Time Codes for PSK Modulation", IEEE Transactions on Information Theory, vol. 46, no. 2, March 2000.

[4] Zhuo Chen, Branka S.Vucetic, Jinhong Yuan , and Ka Leong Lo, "Space–Time Trellis Codes for 4-PSK With Three and Four Transmit Antennas in Quasi-Static Flat Fading Channels", IEEE Communications Letters, vol.6, no.2, February 2002.

[5] J.G. Proakis, Digital Communications, 2nd edition, New York: McGraw-Hill, 1989.

[6] Z. Chen, J. Yuan, and B. Vucetic, "An Improved Space-Time Trellis Coded Modulation Scheme on Slow Rayleigh Fading Channels", IEEE International Conference on Communications, 2001. ICC 2001. Vol. 4, pp. 1110-1116, Helsinki, Finland, 11-14 June 2001.

[7] S. Haykin, Communications Systems, 3rd edition, New York: John Wiley & Sons, 1994.

[8] http://itpp.sourceforge.net/current/

8 Appendix A – Performance Plots

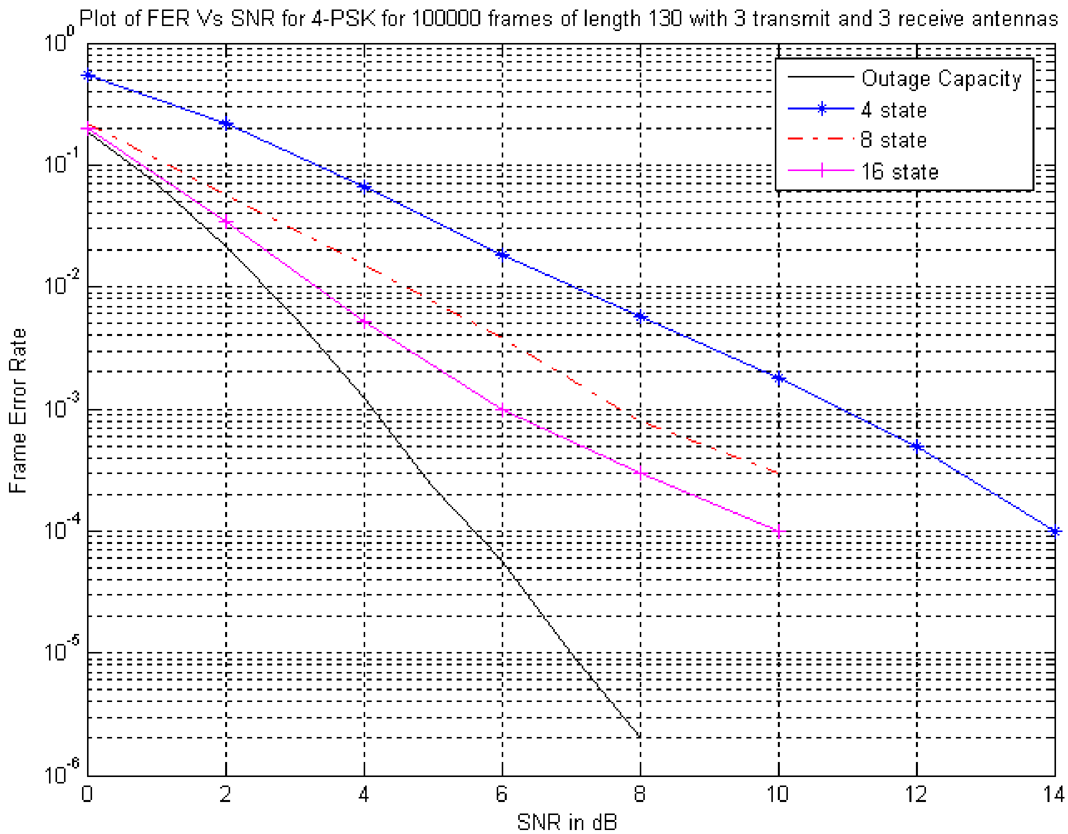

Figure 13: Performance of 4PSK code with three transmit and three receive antenna for quasi-static channel

Simulation Parameters:
Modulation: QPSK
States: 4, 8 and 16
"Quasi-Static" Rayleigh Fading Channel
Coherence Time: 130 Symbols
Perfect CSI at receiver
Number of Transmit Antenna: 3
Number of Receive Antenna: 3

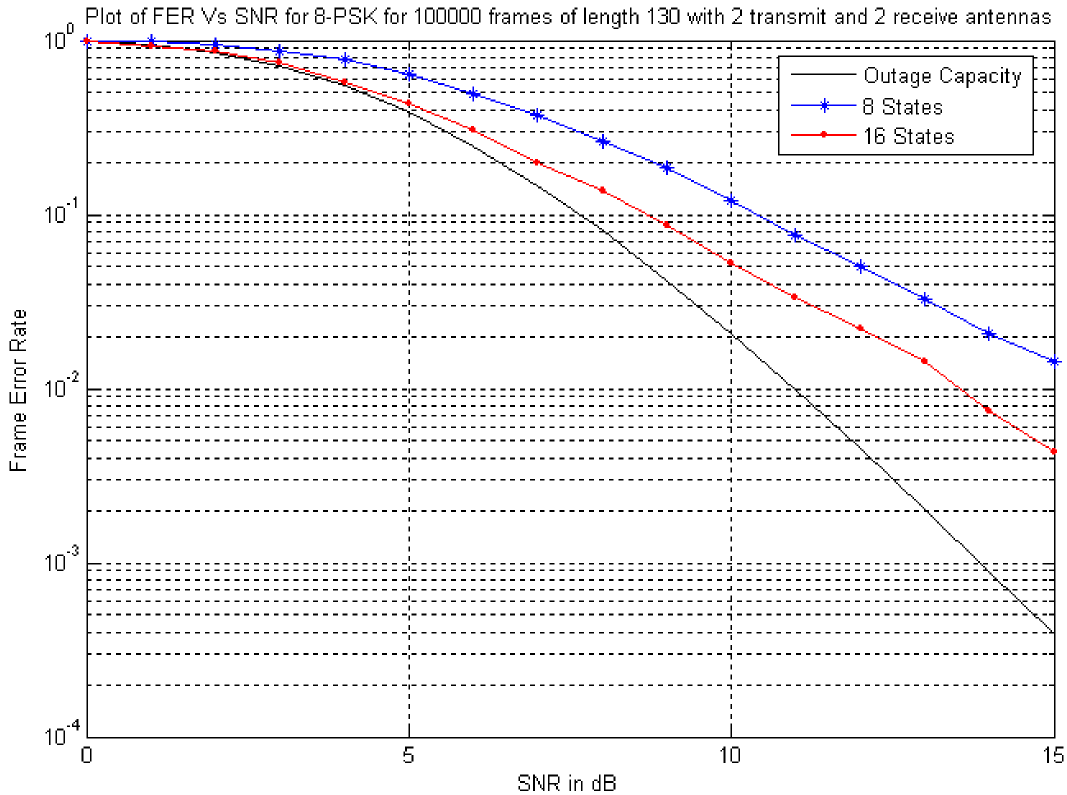

Figure 14: Performance of 8PSK code with two transmit and two receive antenna for quasi-static channel

Simulation Parameters:

Modulation: QPSK
States: 8 and 16
"Quasi-Static" Rayleigh Fading Channel
Coherence Time: 130 Symbols
Perfect CSI at receiver
Number of Transmit Antenna: 2
Number of Receive Antenna: 2

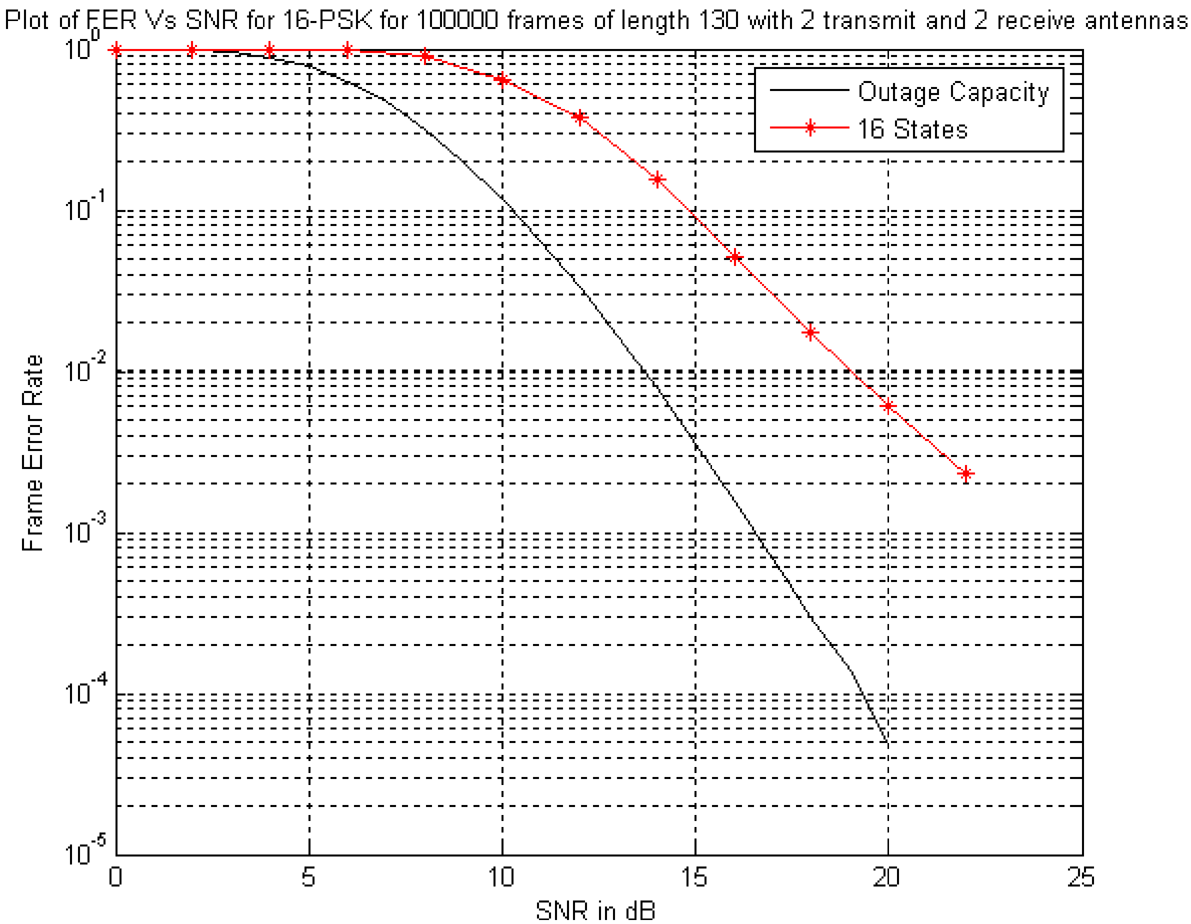

Figure 15: Performance of 16PSK code with two transmit and two receive antenna for quasi-static channel

Simulation Parameters:
Modulation: 16 PSK
States: 16
"Quasi-Static" Rayleigh Fading Channel
Coherence Time: 130 Symbols
Perfect CSI at receiver
Number of Transmit Antenna: 2
Number of Receive Antenna: 2

9 Appendix B – Trellis Structures

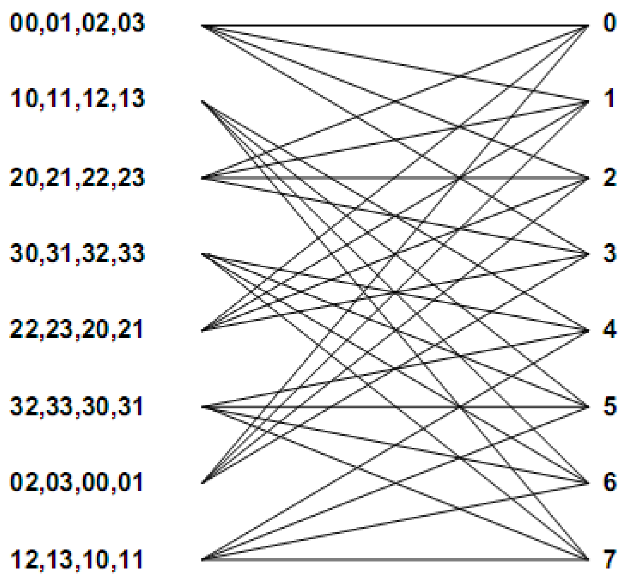

Figure 16: Trellis diagram for 4PSK 8 State STTC for two transmit antenna

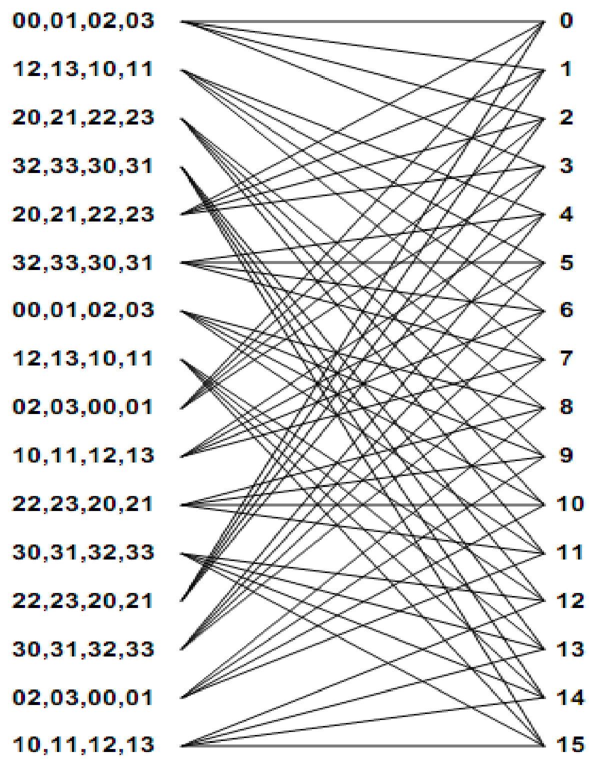

Figure 17: Trellis diagram for 4 PSK 16 State STTC for two transmit antenna

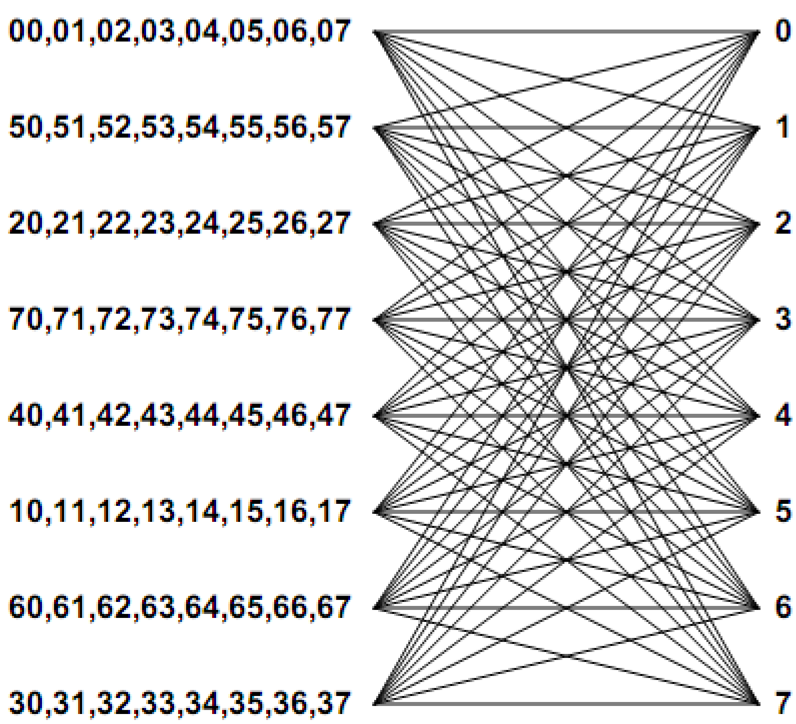

Figure 18: Trellis diagram for 8PSK 8 state STTC for two transmit antennas