DESIGN AND PERFORMANCE EVALUATION OF SUPER-ORTHOGONAL SPACE-TIME TRELLIS CODES

A DISSERTATION

Submitted in partial fulfillment of the requirements for the award of the degree

of

MASTER OF TECHNOLOGY

in

ELECTRONICS AND COMMUNICATION ENGINEERING

(With Specialization in Communication Systems)

By

ANAND PRAKASH

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY ROORKEE

ROORKEE - 247 667 (INDIA)

JUNE, 2008

CANDIDATE'S DECLARATION

I hereby declare that the work, which is presented in this dissertation report entitled, **"DESIGN AND PERFORMANCE EVALUATION OF SUPER-ORTHOGONAL SPACE-TIME TRELLIS CODES"** towards the partial fulfillment of the requirements for the award of the degree of **Master of Technology** with specialization in **Communication Systems**, submitted in the Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, Roorkee (India) is an authentic record of my own work carried out during the period from July 2007 to June 2008, under the guidance of **Dr. A. TYAGI, Assistant Professor, Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee.**

I have not submitted the matter embodied in this dissertation for the award of any other Degree or Diploma.

Date: 27-06-08
Place: Roorkee

Anand Prakash
ANAND PRAKASH

CERTIFICATE

This is to certify that the above statement made by the candidate is correct to the best of my knowledge and belief.

Date: 27-06-08
Place: Roorkee

A Tyagi
Dr. A. TYAGI
Assistant Prof. , E&C Department,
IIT Roorkee,
Roorkee – 247 667 (India).

ACKNOWLEDGMENTS

First, I would like to express my sincere gratitude to my advisor, **Dr. Anshul Tyagi**, who is an endless source of enthusiasm, ideas, and patience. It was him who led me into this exciting area of Space-Time Coding, and has offered me constant encouragement and advice throughout the last two years. I like to thank **Dr. D. K. Mehra** (Head of Department) and all faculty members for their encouragement and constant motivation throughout the course. I also express my great thanks to **Mr. S. Chakravorty**, for his course "Coding Theory and Applications" which helped me immensely.

I feel fortunate to work with my colleagues. I also express my gratitude to Jitu and both Gauravs for being good friends and all their helps in these two years. I thank Sumit,Vinay, Kishore and other people who have combined to create a good working place.

Finally, I would like to thank my parents, who have always encouraged and supported me in all of my decisions.

ANAND PRAKASH

ABSTRACT

With the rapid development of wireless communications, the available bandwidth for wireless applications becomes more and more insufficient. Therefore, to improve data rate without expending bandwidth becomes a main goal in modern communication system design. Recently, two wireless communication schemes, which can be used to effectively combat with multi-path fading, are widely investigated around the world. One is the multiple-input multiple-output (MIMO) technology and another is the space-time coding (STC). As a result, a combined system, MIMO-STC, can be used for wireless communications to jointly explore the advantages of the above two strategies.

In the study of MIMO-STC systems, the space-time block codes (STBC) and the space-time trellis codes (STTC) are two efficient coding approaches. The former can be used to offer a full diversity gain, and the later can provide the systems with a large coding gain. Since 2003, super-orthogonal space-time trellis codes (SOSTTC) a new coding method processing the merits of both STBC and STTC, have been developed.

In this dissertation, SOSTTCs scheme along with MPSK constellation are used for high speed wireless communications of a MIMO-STC system. To examine the performance of the proposed SOSTTC system based on the optimal design of set partitioning for BPSK, QPSK and 8-PSK modulation schemes. In simulations, the frame error rate (FER) versus the received SNR performance results show that SOSTTCs have better coding gain than that of the corresponding STTCs. The decoding complexity of SOSTTC is lower than the decoding complexity of STTC.

According to the above results, as the number of states of SOSTTCs increases their performance improves, and the decoding complexity becomes higher .Coding gain distance (CGD) is a good indicator of performance for a large number of receive antenna [18].

Table of Contents

Page No.

List of Figures

List of Tables

Page

List of Abbreviations

AWGN	Additive White Gaussian Noise
CGD	Coding Gain Distance
MIMO	Multiple-Input Multiple-Output
MRRC	Maximum-Ratio Receiver Combining
STC	Space-Time Coding
SOSTTC	Super-Orthogonal Space-Time Trellis Code
STBC	Space-Time Block Code
STTC	Space-Time Trellis Code
SNR	Signal to Noise
SQOSTTC	Super-Quasi-Orthogonal Space-Time Trellis Code
ML	Maximum Likelihood
TCM	Trellis Code Modulation

Chapter 1

1.1 Introduction

With the integration of multimedia and internet applications in next generation wireless communications, the demand for wide-band high data rate communication services is growing. As the available radio spectrum is limited, higher data rates can be achieved only by designing more efficient signaling techniques [1]. Among many cutting edge wireless technologies, a new class of transmission techniques, known as Multiple-Input Multiple-Output (MIMO) technique, has emerged as an important technology leading to promising link capacity gains of several fold increase in achievable data rates and spectral efficiency. However, the key question is how to exploit this new capability of MIMO wireless communications in a computationally efficient manner.

Space-time coding (STC) techniques have been identified as the solution [3], as it is a set of practical signal design techniques aimed at approaching the information theoretic capacity limit of MIMO channels. Space-time coding is based on introducing joint correlation in transmitted signals in both the space and time domains. Through this approach, simultaneous diversity and coding gains can be obtained, as well as high spectral efficiencies [1].

Wireless communication system, where the transmitter contains N transmit antennas and the decoder contains M receive antennas. The goal of space-time coding is to achieve the maximum diversity of NM, the maximum coding gain, and the highest possible data rates. In addition, the decoding complexity is very important. In a typical wireless communication system the mobile transceiver has a limited available power through a battery and should be a small physical device. To improve the battery life, low complexity encoding and decoding is very crucial. Generally, the base station is not as restricted in terms of power and physical size .There is multiple independent antennas in a base station. Therefore, in many practical situations, a very low complexity system with multiple transmit antennas is desirable. Space-time block coding is a scheme to provide these properties.

1.2 Motivation of Research

Recently, several transmit diversity techniques have been proposed, among them, Alamouti [2] proposed a very simple transmit diversity scheme using orthogonal transmit code design so-called space-time block code (STBC). With multiple transmit antennas, STBC can easily be designed and can achieve the same diversity gain as that of maximum-ratio receiver combining (MRRC) scheme. In addition, Tarokh [1] [8] introduced a space-time trellis coded modulation scheme (STTCM) over Rayleigh fading channels, which can obtain full transmit diversity gain as well as signal-to-noise ratio (SNR) advantage, so-called coding gain. ST-MTCM introduced by D. Divsalar [5] focuses on the trellis codes with parallel paths. It is well known that it is impossible to achieve maximum transmit diversity gain when trellis codes contain parallel paths using STTCM scheme. However, ST-MTCM scheme proposed a new scheme to achieve the maximum diversity and a larger coding gain compared to Tarokh's STTCM codes. ST-MTCM uses a trellis code with each branch corresponding to multiple transmissions from each transmit antenna so as to achieve full rank. In order to improve system performance, the coding gain distance (CGD), which is the distance between two parallel paths for any fixed state transition, that is different lengths error event, should be maximized.

The set partitioning structures are used to maximize the CGD, and obtain the criteria for maximizing the CGD theoretically for MPSK. Different SOSTTC [9] [10] codes are designed to meet these criteria which guarantee that the CGD corresponding to length-1 or more than 2 error event is as large as possible, thus achieving the maximum coding gain. Due to the multiple transmission in each state transition, however, this scheme cannot achieve full transmit rate. The full-rate transmission satisfies b bits/s/Hz transmission, where $b = log_2M$ with MPSK modulation. This dissertation presents a systematic code design, so as to obtain full rate and maintain good coding gain.

STBC is simply an orthogonal design focusing mainly on obtaining full transmit rate, diversity gain, and a simple decoding scheme, but coding gain cannot be achieved. In contrast, although the decoding complexity is very high, ST-MTCM provides full diversity gain as well as better coding gain. Inspired by the orthogonal transmit matrix design in STBC, a new scheme is proposed. Which uses STBC's orthogonal transmit matrix for ST-MTCM scheme in order to achieve full rate. Besides, theoretical matrix assignment for good coding gain can be done. There are some similar schemes proposed based on STBC. Siwamogsatham and Fitz [7] [14] introduced some specific code designs by using STBC combined with trellis codes that can achieve better coding gain and full transmit rate, full diversity gain . In a similar way, super-orthogonal space-time trellis codes (SO-STTC) introduced by combines STBC with STTC in order to design codes for any given number of states in a more systematic way, and guarantees the full diversity gain and maximum coding gain for the proposed structure with optimal set partitioning.

Space-time trellis coding is the extension of trellis-coded modulation (TCM) to wireless communication links using more than a single transmit antenna in order to improve the error performance [4]. A high-performance sub-class of space-time trellis codes for two transmit antennas is known as "super-orthogonal space-time trellis codes" where the state transitions of the encoder trellis diagram are labeled with matrices built according to Alamouti's scheme [2] and applying Ungerboeck's rules [4]. While an appropriate partitioning technique was known for phase-shift keying (PSK). There are no full-rate complex-valued orthogonal designs for more than two transmit antennas. To provide full-rate with simple pair-wise decoding strategies, super-quasi-orthogonal space-time codes (SQOSTTCs) have been introduced in [6] [11] for four transmit antennas and PSK signal constellations.

1.3 Statement of Problem

(a.) This dissertation presents the design of super-orthogonal space-time trellis codes (SOSTTCs) to transmit information over a multiple antenna wireless communication system.

(b.) Based on the space-time design criteria, this dissertation developed a optimal design of set partitioning structures with respect to coding gain distance (CGD) using M-PSK.

(c.) Finally, this work extends the performance evaluation of super-orthogonal space-time trellis codes and modified Viterbi decoding techniques based on the parameterized class of space-time block codes(STBCs) through simulation.

1.4 Outline of Dissertation

This dissertation is divided in four chapters. The first chapter gives an introduction to space-time coding of wireless communication systems, motivation of research, problem of statement and the outline of the dissertation.

In chapter 2, design of optimal set partitioning structures using M-PSK are presented. The number of orthogonal design matrices (STBCs) increases, this scheme provides a sufficient number of constellation matrices to design a trellis code with the highest possible rate. Also, it allows a systematic design of super-orthogonal space-time trellis codes.

In Chapter 3, the analysis of coding gain distance for error events with different path lengths of SOSTTCs is presented, the aim of calculations is find out dominant path in the trellis structure. Finally, the standard reduced-complexity ML decoder for an orthogonal space-time block code is studied[18] [21]and [22].

In Chapter 4, the algorithm and simulation results are discussed for different states of SOSTTCs using different rates, which demonstrate the frame-error rate(FER) versus SNR(dB) for quasi-static Rayleigh channels[19]. Finally, the conclusions and further research for higher coding gain advantage are presented.

Chapter 2

Super-Orthogonal Space-Time Trellis Codes

2.1 Introduction

STBCs provide full diversity and small decoding complexity. STBCs can be considered as modulation schemes for multiple transmit antennas and as such do not provide coding gains. Full rate STBCs do not exist for every possible number of transmit antennas. On the other hand, STTCs [1] [12] are designed to achieve full diversity and high coding gains while requiring a higher decoding complexity. The STTCs are designed either manually or by computer search. In this chapter a systematic method to design space-time trellis codes is presented. Another way of achieving high coding gains is to concatenate an outer trellis code that has been designed for the AWGN channel with a STBC. If orthogonal matrices generated by a STBC is considered, as a point in a high dimensional space. The outer trellis code's task is to select one of these high dimensional signal points, based on the current state and the input bits. In Ungerboeck [3], it is shown that for the slow fading channel, the trellis code should be based on the set partitioning concepts of "Ungerboeck codes" for the AWGN channel. The main idea behind super-orthogonal space-time trellis codes (SOSTTCs) is to consider STBCs as modulation schemes for multiple transmit antennas.

2. 2 Space-Time Code Design Criteria

In order to come up with a design criterion, first it is needed to quantify the effects of mistaking two codewords with each other. If two codeword C^1 and C^2 are considered, of a space-time code , the size of a codeword is a $T \times N$ matrix. If codeword C^1 transmitted as

$$C^1 = \begin{pmatrix} c^1_{1,1} & c^1_{1,2} & .. & c^1_{1,N} \\ c^1_{2,1} & c^1_{2,2} & .. & c^1_{2,N} \\ \vdots & \vdots & & \vdots \\ c^1_{T,1} & c^1_{T,2} & .. & c^1_{T,N} \end{pmatrix} \tag{2.1}$$

An error occurs if the decoder mistakenly decides that codeword C^2 has been transmitted

$$C^2 = \begin{pmatrix} c_{1,1}^2 & c_{1,2}^2 & .. & c_{1,N}^2 \\ c_{2,1}^2 & c_{2,2}^2 & .. & c_{2,N}^2 \\ \vdots & \vdots & & \vdots \\ c_{T,1}^2 & c_{T,2}^2 & .. & c_{T,N}^2 \end{pmatrix} \quad (2.2)$$

If the codebook contains only C^1 and C^2, then the pairwise error probability of transmitting C^1 and detecting it as C^2 is denoted as $P(C^1 \to C^2)$.When the codebook contains I codewords, using the union bound, the probability of error when C^1 transmitted, is upper bounded by [1]

$$P(error / C^1) \leq \sum_{i=2}^{I} P(C^1 \to C^2) \quad (2.3)$$

Difference matrix C^2 - C^1 is denoted as $D(C^1,C^2)$ and a matrix $A(C^1,C^2)= [D^H(C^1,C^2).\ D(C^1,C^2)]$.

$$P(C^1 \to C^2) \leq \frac{4^{rM}}{(\prod_{n=1}^{r} \lambda_n)^M \gamma^{rM}} \quad (2.4)$$

Where,

M is the number of receiving antennas,

r is rank of a matrix $A\ (C^1,C^2)$ and

$\lambda_1, \lambda_2, \ldots\ldots, \lambda_r$ is the eigen values of matrix A and

γ is signal to noise ratio.

Right side of equation 2.4 can be represented as $(G_c \gamma)^{-G_d}$

Where,

G_d= diversity gain

G_c= coding gain

Then, the diversity of code is equal to product of rank of matrix $A(C^1,C^2)$ *and* M (i.e rM) .The coding gain relates to the product of the nonzero eigenvalues of matrix $A(C^1,C^2)$ or equivalently the determinant of matrix $A(C^1,C^2)$. The coding gain distance (CGD) between codewords C^1 and C^2 is defined as $CGD(C^1,C^2)=det(A(C^1,C^2))$.These two criteria for designing Space-time codes are called rank and determinant criteria.

2.3 Alamouti code

In this code STBC is considered as modulation schemes for multiple transmit antennas that provide full diversity and very low complexity encoding and decoding. If a system has two transmit antenna (N=2) and one receive antenna(M=1), employing Alamouti code as in Figure 2.1 [2].

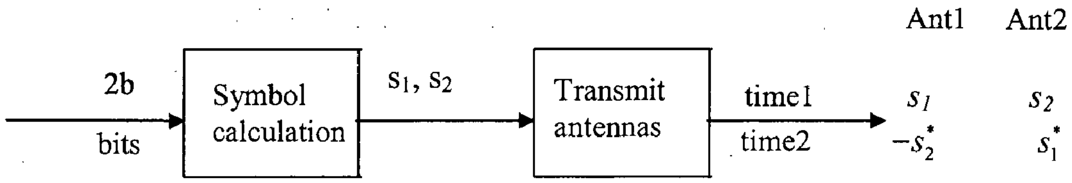

Fig. 2.1. Transmitter block diagram for Alamouti code.

To transmit b bits/cycle, a modulation scheme that maps every b bits to one symbol from a constellation with $2b$ symbols is used. The constellation can be any real or complex constellation. The transmitter picks two symbols from the constellation using a block of $2b$ bits. If s_1 and s_2 are the selected symbols for a block of $2b$ bits, the transmitter sends s_1 from antenna one and s_2 from antenna two at time one. Then at time two, it transmits $-s_2^*$ and s_1^* from antennas one and two, respectively. Therefore, the transmitted codeword is

$$C = \begin{pmatrix} s_1 & s_2 \\ -s_2^* & s_1^* \end{pmatrix} \qquad (2.5)$$

To check if the code provides full diversity, rank of all possible difference matrices $D(C,C^1)$ is calculated .For different pair of symbols (s_1^1, s_2^1) , the corresponding codeword is given by

$$C^1 = \begin{pmatrix} s_1^1 & s_2^1 \\ -s_2^{1*} & s_1^{1*} \end{pmatrix} \qquad (2.6)$$

The difference matrix $D(C,C^1)$ is given by difference of matrices C and C^1.

The determinant of the difference matrix, $\det[D(C,C^1)] = (|s_1^1 - s_1|^2 + |s_2^1 - s_2|^2)^2$ is zero if and only if $s_1^1 = s_1$ and $s_2^1 = s_2$. Therefore, $D(C,C^1)$ is always full rank when $C \neq C^1$ and the Alamouti code satisfies the determinant criterion. It provides a diversity of $2M$ for M receive antennas and therefore is a full diversity code. The code word distance matrix A

$$A=DD^H \tag{2.7}$$

This matrix has two identical eigenvalues. The minimum eigen values is equal to the minimum squared Euclidian distance in the signal constellation. Hene, the minimum distance between any two transmitted sequences remains the same as in the uncoded system. This implies that the coding gain is one. The Alamouti code provides two important properties:

1. Simple decoding: Each symbol is decoded separately using only linear processing.

2. Maximum diversity: The code satisfies the rank criterion and therefore provides the maximum possible diversity.

2.4 Orthogonal Design of STBC

Assuming, the path gains from transmit antennas one and two to the receive antenna as α_1 and α_2, respectively. Then, the decoder receives signals r_1 and r_2 at times one and two, respectively, such that

$$\begin{aligned} r_1 &= \alpha_1 s_1 + \alpha_2 s_2 + \eta_1 \\ r_2 &= -\alpha_1 s_2^* + \alpha_2 s_1^* + \eta_2 \end{aligned} \tag{2.8}$$

For a coherent detection scheme where the receiver knows the channel path gains α_1 and α_2, the maximum-likelihood detection amounts to minimizing the decision metric

to decode s_1 and s_2 and minimize

$$\begin{aligned} &\left|s_1 - r_1\alpha_1^* - r_2^*\alpha_2\right|^2 \\ &\left|s_2 - r_1\alpha_2^* + r_2^*\alpha_1\right|^2 \end{aligned} \tag{2.9}$$

Therefore, the decoding consists of first calculating

$$\begin{aligned} \bar{s}_1 &= r_1\alpha_1^* + r_2\alpha_2^* \\ \bar{s}_2 &= r_1\alpha_2^* - r_2\alpha_1^* \end{aligned} \tag{2.10}$$

Thus, to decode s_1, the receiver finds the closet symbol to $\tilde{s}_1$ in the constellation. Similarly, the decoding of s_2 consists of finding the closest symbol to $\tilde{s}_2$ in the constellation.

Equation *(2.8)* can be written as

$$(r_1, r_2^*) = (s_1, s_2)\Omega + (\eta_1, \eta_2^*) \qquad (2.11)$$

and

$$\Omega = \Omega(\alpha_1, \alpha_2) = \begin{pmatrix} \alpha_1 & \alpha_2^* \\ \alpha_2 & -\alpha_1^* \end{pmatrix} \qquad (2.12)$$

Multiplying above equation (2.11) by Ω^H gives

$$\left(\tilde{s}_1, \tilde{s}_2\right) = (r_1, r_2) = (|\alpha_1|^2 + |\alpha_2|^2)(s_1, s_2) + \mathrm{N} \qquad (2.13)$$

Where, N is also a Gaussian noise, equation (2.13) consists of two separate equations for decoding the two transmitted symbols. This equation is the main reason for the first property of simple maximum likelihood (ML) decoding. The second property comes from the factor $|\alpha_1|^2 + |\alpha_2|^2$ in the right side of equation*(2.13)*.When there is only one transmit antenna available, the power of the signal is affected by a factor $|\alpha|^2$ of due to the path gain .In a deep faded environment $|\alpha|^2$ is very small and the noise dominates the signal. Thus,*(2.13)* shows that using the Almaouti code $|\alpha_1|^2 + |\alpha_2|^2$ should be small for a noise dominated channel. To have a small $|\alpha_1|^2 + |\alpha_2|^2$,both $|\alpha_1|^2$ and $|\alpha_2|^2$ must be small. It is less likely, as α_1 and α_2 are independent.

Derviation of *(2.13)* shows that the following equation is the main reason for both properties of the Alamouti code.

$$\Omega.\Omega^H = (|\alpha_1|^2 + |\alpha_2|^2)I_2 \qquad (2.14)$$

Where I_2 is 2 x 2 identity matrix .If the structure of the Alamouti code is represented by the generator matrix

$$G = \begin{pmatrix} x_1 & x_2 \\ -x_2^* & x_1^* \end{pmatrix} \qquad (2.15)$$

The inner product of rows G is zero. Hence, the transmitted symbols are orthogonal. *(2.14)* is a result of the orthogonality of the columns of G and the following property

$$G.G^H = (|x_1|^2 + |x_2|^2)I_2 \quad (2.16)$$

2.4.1 Orthogonal matrices

A design of a full-rate full-diversity complex STBC is the scheme proposed in Section 2.4, which is defined by the following generator matrix

$$G(x_1, x_2) = \begin{pmatrix} x_1 & x_2 \\ -x_2^* & x_1^* \end{pmatrix} \quad (2.17)$$

The code is designed for $N = 2$ transmit antennas and any number of receive antennas. Using a constellation with $L = 2^b$ points, the code transmits $2b$ bits every two symbol intervals. For each block, $2b$ bits arrive at the encoder and the encoder chooses two modulation symbols s_1 and s_2. Then, replacing x_1 and x_2 by s_1 and s_2 respectively, to arrive at $G\,(s_1, s_2)$, the encoder transmits s_1 from antenna one and s_2 from antenna two at time one. Also, the encoder transmits $-s_2^*$ from antenna one and s_1^* from antenna two at time two. This scheme provides diversity gain, but no additional coding gain. There are other codes which provide behavior similar to those of *(2.17)* for the same rate and number of transmit antennas. When multiplying an orthogonal design by a unitary matrix results in another orthogonal design. The set of all such codes which only use x_1, x_2, and their conjugates with positive or negative signs are listed below:

$$\begin{pmatrix} x_1 & x_2 \\ -x_2^* & x_1^* \end{pmatrix}, \begin{pmatrix} -x_1 & x_2 \\ x_2^* & x_1^* \end{pmatrix}, \begin{pmatrix} x_1 & -x_2 \\ x_2^* & x_1^* \end{pmatrix}, \begin{pmatrix} x_1 & x_2 \\ x_2^* & -x_1^* \end{pmatrix},$$
$$\begin{pmatrix} -x_1 & -x_2 \\ x_2^* & -x_1^* \end{pmatrix}, \begin{pmatrix} -x_1 & x_2 \\ -x_2^* & -x_1^* \end{pmatrix}, \begin{pmatrix} x_1 & -x_2 \\ -x_2^* & -x_1^* \end{pmatrix}, \begin{pmatrix} -x_1 & -x_2 \\ -x_2^* & x_1^* \end{pmatrix} \quad (2.18)$$

As the inner product of rows of all matrices is zero. Hence transmitted symbols are orthogonal to each other. The union of all these codes is called "super-orthogonal code" set *C*. Using just one of the constituent codes from *C*, all possible orthogonal 2 × 2 matrices for a given constellation can not be generated using Equation *(2.17)*. In BPSK constellation, it can be shown that one can build all possible 2 × 2 orthogonal matrices using two of the codes in *C*.

The following four 2 × 2 constellation matrices can be generated using the code in *(2.17)*:

$$\begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}, \begin{pmatrix} -1 & -1 \\ 1 & -1 \end{pmatrix}, \begin{pmatrix} -1 & 1 \\ -1 & -1 \end{pmatrix}, \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}, \quad (2.19)$$

There are four other possible distinct orthogonal 2 × 2 matrices

$$\begin{pmatrix} -1 & 1 \\ 1 & 1 \end{pmatrix}, \begin{pmatrix} 1 & -1 \\ -1 & -1 \end{pmatrix}, \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \begin{pmatrix} -1 & -1 \\ -1 & 1 \end{pmatrix} \qquad (2.20)$$

Another orthogonal STBC is obtained by multiplying an orthogonal STBC by a unitary matrix from right or left. As a particular case if generator matrix is multiplied by following unitary matrix from right by the following unitary matrix:

$$U = \begin{pmatrix} e^{j\theta} & 0 \\ 0 & 1 \end{pmatrix} \qquad (2.21)$$

The result is the rotation of the first column of the generator matrix. To create these additional matrices, the following code from the set C is used:

$$\begin{pmatrix} -x_1 & x_1 \\ x_2^* & x_1^* \end{pmatrix} = \begin{pmatrix} x_1 & x_1 \\ -x_2^* & x_1^* \end{pmatrix} \cdot \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \qquad (2.22)$$

Which, represents a phase shift of the signals transmitted from antenna one by π. A set including all 2 × 2 orthogonal matrices from *(2.19)* and *(2.20)* can be denoted as O_2. The rank of the difference between any two distinct matrices within either *(2.19)* or *(2.20)* is 2. However, the rank of a difference matrix between any two elements in *(2.19)* and *(2.20)* is 1. In the case full diversity design, the difference of codewords matrices is full rank matrix. By using more than one code from set C, all possible 2 × 2 orthogonal matrices from O_2 can be obtained. Therefore, the scheme provides a sufficient number of constellation matrices to design a trellis code with the highest possible rate. Also, it allows a systematic design of space-time trellis codes using the available knowledge about trellis-coded modulation (TCM) [4] and multiple TCM (MTCM) [5].

2.5 Super-orthogonal codes

2.5.1 A parameterized class of STBCs

Multiplying the Alamouti code in *(2.17)* from the right by the unitary matrix U in *(2.21)* results in the following class of orthogonal designs

$$G(x_1, x_2, \theta) = G(x_1, x_2).U = \begin{pmatrix} x_1 e^{j\theta} & x_2 \\ -x_2^* e^{j\theta} & x_1^* \end{pmatrix} \qquad (2.23)$$

When $\theta = 0$, provides the code in (2.17). So, Let $G(x_1, x_2, 0)$ be denoted by $G(x_1, x_2)$. The encoder uses $2b$ input bits to select s_1 and s_2. Then, replacing x_1 and x_2 in $G(x_1, x_2, \theta)$ with s_1 and s_2 respectively, results in $G(s_1, s_2, \theta)$. First, $s_1 e^{j\theta}$ and s_2 are transmitted from the first and second transmit antennas, respectively. Second, $-s_2^* e^{j\theta}$ and s_1^* are transmitted from the two antennas. Concentrating on the case for which the set of transmitted signals are the same as the constellation points. In other words, the signal alphabet is not expanded. θ *is* picked such that for any choice of s_1 and s_2 from the original constellation points, the resulting transmitted signals are also from the same constellation. For an L-PSK, the constellation signals can be represented by $e^{j2\pi l/L}$, $l = 0, 1, \ldots, L-1$. selecting $\theta = 2\pi l'/L$, where $l' = 0, 1, \ldots, L-1$ to avoid constellation expansion. In this case, the resulting transmitted signals are also members of the L-PSK constellation. Since the transmitted signals are from a PSK constellation, the peak-to-average power ratio of the transmitted signals is equal to one. This not only increases the number of signals in the constellation, but also there is no need for an amplifier to provide a higher linear operation region. Using $\theta = 0, \pi$ and $\theta = 0, \pi/2, \pi, 3\pi/2$ for BPSK and QPSK, respectively. By using $G(x_1, x_2, 0)$ and $G(x_1, x_2, \pi)$ for the BPSK constellation, all 2×2 orthogonal matrices in $O2$ can be generated. In fact, $G(x_1, x_2, 0)$ is the code in *(2.17)* and $G(x_1, x_2, \pi)$ is the code in *(2.21)*. By using *(2.17)* and *(2.21)*, the set of transmitted signals consists of $s_1, s_2, s_1^*, -s_2^*, -s_1, -s_1^*$. Combination of these two codes can be called a super-orthogonal code.

In general, a super-orthogonal code consists of the union of a few orthogonal codes, like the ones in *(2.20)*. A special case is when the super-orthogonal code consists of only one orthogonal code, for design only $\theta = 0$. Therefore, the set of orthogonal codes is a subset of the set of super-orthogonal codes. Obviously, the number of orthogonal matrices that a super-orthogonal code provides is more than, or in the worst case equal to, the number of orthogonal matrices that an orthogonal code provides. Therefore, while the super-orthogonal code does not extend the constellation alphabet of the transmitted signals, it does expand the number of available orthogonal matrices. This is of great benefit and crucial in the design of full-rate, full-diversity trellis codes. Another advantage of super-orthogonal codes lies in the fact that the code is parameterized. A design of a super-orthogonal code is the union of $G(x_1, x_2, \theta)$ for an L-PSK constellation where $\theta = 2\pi l'/L$, and $l' = 0, 1 \ldots L-1$. The expansion of the orthogonal matrices," that is a positive result of using super-orthogonal codes, and "the expansion of the signal constellation," that is usually a negative side effect. The super-orthogonal code expands the number of available orthogonal matrices and, this is the main reason it can design full-rate trellis codes that provide full diversity. It has no negative side effect either in terms of expanding the transmitted signal constellation.

2.5.2 Set partitioning for orthogonal codewords

This section provides a set partitioning for the codewords of an orthogonal code and shows how to maximize the coding gain. The minimum of the determinant of the matrix $A(C^i, C^j) = D^H(C^i, C^j)\,D(C^i, C^j)$ over all possible pairs of distinct codewords C^i and C^j corresponds to the coding gain for a full diversity code. This value is called Coding Gain Distance (CGD). Then, CGD is used instead of Euclidean distance to define a set partitioning similar to Ungerboeck's set partitioning [4]. Firstly, on the case where code in *(2.17)* is utilized. Also, BPSK constellation is used. Consider a four-way partitioning of the codewords of the orthogonal code as shown in Figure 2.2 for BPSK. The numbers at the leaves of the tree in the figure represent BPSK indices. Each pair of indices represents a 2 × 2 codeword built by the corresponding pair of symbols in the constellation.

At the root of the tree, the minimum determinant of $A(C^i, C^j)$ among all possible pairs of codewords (C^i, C^j) is 16. At the first level of partitioning, the highest determinant that can be obtained is 64. This is obtained by a set partitioning in which subsets P_0 and P_1 use different transmitted signal elements for different transmit antennas. At the next level of partitioning, four sets P_{00}, P_{11}, P_{01} and P_{10} are present with only one codeword per set.

The best set partitioning is the one for which the minimum CGD of the sets at each level of the tree is maximum among all possible cases. To find guidelines for partitioning the sets, formulas to calculate CGDs, are derived. For an L-PSK constellation, let each signal is represented by $s = e^{jl\omega}$, $l = 0, 1, \ldots, L-1$, where $\omega = 2\pi/L$. Two distinct pairs of constellation symbols $(s_1^1 = e^{jk_1\omega}, s_2^1 = e^{jl_1\omega})$ and $(s_1^2 = e^{jk_2\omega}, s_2^2 = e^{jl_2\omega})$ and the corresponding 2 × 2 orthogonal codewords C^1 and C^2 are considered.

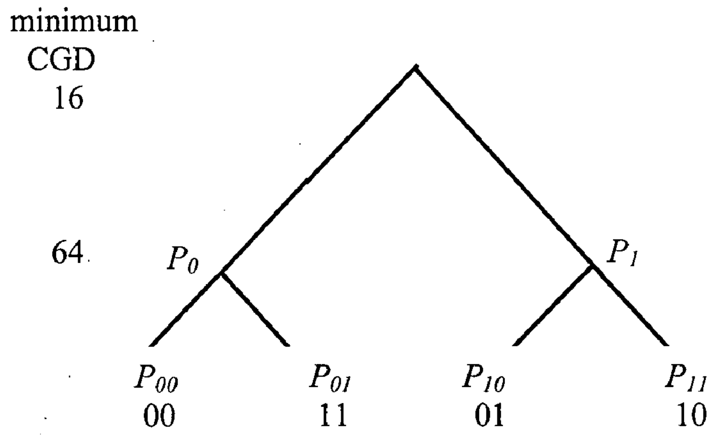

Fig. 2.2. Set Partitioning for BPSK ; each pair of binary numbers represents the pair of symbol indices in a STBC

When there is no ambiguity *(C^1,C^2)* is omitted in the calculation of matrices $D(C^1,C^2)$ and $A(C^1,C^2)$, from A and D. For parallel transitions in a trellis,

$$D = \begin{pmatrix} e^{jk_1\omega} - e^{jk_2\omega} & e^{jl_1\omega} - e^{jl_2\omega} \\ e^{-jl_2\omega} - e^{-jl_1\omega} & e^{-jk_1\omega} - e^{-jk_2\omega} \end{pmatrix} \qquad (2.24)$$

$$A = D^H D$$

Then, cofactors of A are

$$(e^{jk_1\omega} - e^{jk_2\omega})(e^{-jk_1\omega} - e^{-jk_2\omega}) + (e^{jl_1\omega} - e^{jl_2\omega})(e^{-jl_1\omega} - e^{-jl_2\omega})$$
$$= 4 - 2\cos[\omega(k_2 - k_1)] - 2\cos[\omega(l_2 - l_1)]$$

$$(e^{jk_1\omega} - e^{jk_2\omega})(e^{jl_2\omega} - e^{jl_1\omega}) + (e^{jl_1\omega} - e^{jl_2\omega})(e^{jk_1\omega} - e^{jk_2\omega}) = 0$$
$$(e^{-jl_2\omega} - e^{-jl_1\omega})(e^{-jk_1\omega} - e^{-jk_2\omega}) + (e^{-jk_1\omega} - e^{-jk_2\omega})(e^{-jl_1\omega} - e^{-jl_2\omega}) = 0$$
$$(e^{jl_2\omega} - e^{jl_1\omega})(e^{-jl_2\omega} - e^{-jl_1\omega}) + (e^{-jk_1\omega} - e^{-jk_2\omega})(e^{jk_1\omega} - e^{jk_2\omega})$$
$$= 4 - 2\cos[\omega(k_2 - k_1)] - 2\cos[\omega(l_2 - l_1)] \qquad (2.25a)$$

$$A = \begin{pmatrix} 4 - 2\cos[\omega(k_2 - k_1)] - 2\cos[\omega(l_2 - l_1)] & 0 \\ 0 & 4 - 2\cos[\omega(k_2 - k_1)] - 2\cos[\omega(l_2 - l_1)] \end{pmatrix} \qquad (2.25b)$$

Using *(2.24)*, CGD can be calculated by

$$\det(A) = \{4 - 2\cos[\omega(k_2 - k_1)] - 2\cos[\omega(l_2 - l_1)]\}^2 \qquad (2.26)$$

So far, the codewords corresponding to parallel transitions are considered only. These codewords are transmitted over two time slots and are represented by 2 × 2 matrices. This is good enough for the purpose of set partitioning similar to the case of TCM suggested by Ungerboeck [4]. However, to calculate the minimum CGD of the code, codewords that includes more than one trellis transition. If two codewords are considered , that diverge from state zero and remerge after P trellis transitions, the size of the corresponding difference matrix D is $2P \times 2$. In fact, such a difference matrix can be represented as the concatenation of P difference matrices corresponding to the P transitions that construct the path. The set of constellation symbols are denoted for the first codeword by $\left(s_1^1, s_2^1\right)^p = \left(e^{jk_1^p\omega}, e^{jl_1^p\omega}\right)$, $p=1,2,\ldots,P$ and for the second codeword by $\left(s_1^2, s_2^2\right)^p = \left(e^{jk_2^p\omega}, e^{jl_2^p\omega}\right)$, $p = 1, 2, \ldots, P$. D^p be denoted as the difference matrix of the pth transition and $A^p = D^{pH} \cdot D^p$ for $p = 1, 2, \ldots, P$. For the above two codewords, D is obtained as

$$D = \begin{pmatrix} D^1 \\ D^2 \\ \cdot \\ \cdot \\ \cdot \\ D^P \end{pmatrix} \qquad (2.27)$$

The matrix A be calculated using *(2.27)* as

$$A = D^H D = \sum_{p=1}^{P} A^p \qquad (2.28)$$

Therefore, matrix A is still a diagonal 2 × 2 matrix, that is $A_{12} = A_{21} = 0$. The CGD between the above codewords that differ in P transitions can be calculated by

$$\det(A) = \left\{ \sum_{p=1}^{P} \left\{ 4 - 2\cos[\omega(k_2^p - k_1^p)] - 2\cos[\omega(l_2^p - l_1^p)] \right\} \right\}^2 \qquad (2.29)$$

(2.29) includes a sum of P terms and each of these terms is non-negative. Therefore, the following inequality holds:

$$\det(A) = \left\{ \sum_{p=1}^{P} \left\{ 4 - 2\cos[\omega(k_2^p - k_1^p)] - 2\cos[\omega(l_2^p - l_1^p)] \right\} \right\}^2$$
$$\geq \left\{ \sum_{p=1}^{P} \left\{ 4 - 2\cos[\omega(k_2^p - k_1^p)] - 2\cos[\omega(l_2^p - l_1^p)] \right\}^2 \right\}$$
$$\equiv \left\{ \sum_{p=1}^{P} \det(A^p) \right\} \qquad (2.30)$$

The coding gain of such a STTC is dominated by parallel transitions , based on the coding distances calculated as given by equation *(2.26)* and *(2.28)*. The optimal set partitioning for BPSK, QPSK, and 8-PSK are demonstrated in Figures 2.2, 2.3, and 2.4 respectively.

To calculate the CGD between two codewords, in which a codeword is represented by a pair of symbols, can use *(2.26)* or the corresponding pair of indices. It is apparent that increasing the Euclidean distance between the first symbols of the first and second symbol pairs will increase the CGD. The CGD also increases as, the Euclidean distance between the second symbols be increased. Therefore, a rule of thumb in set partitioning is to choose the codewords that contain signal elements with highest maximum Euclidean distance from each other as the leaves of the set-partitioning tree.

In the design of set partitioning using QPSK in Figure 2.3, $s = e^{j\pi l/2}$, $l = 0, 1, 2, 3$ are the QPSK signal constellation elements and $k = 0, 1, 2, 3$ represent $s = 1, j, -1, -j$, respectively. The maximum minimum CGD in this case is 64, when $|k_1 - k_2| = 2$ and $|l_1 - l_2| = 2$ in *(2.26)*. Based on the above constellation elements, leaf codewords in Figure 2.2 are chosen.

At the second level of the tree from the bottom, it is impossible to have both $|k_1 - k_2|$ and $|l_1 - l_2|$ equal to 2 in all cases. The next highest value for minimum CGD is 16 when $|k_1 - k_2| = 2$, $|l_1 - l_2| = 0$ or $|k_1 - k_2| = 0$, $|l_1 - l_2| = 2$. Therefore, the subtrees are grouped in the second level such that the worst case is when $|k_1 - k_2| = 2$, $|l_1 - l_2| = 0$ or $|k_1 - k_2| = 0$, $|l_1 - l_2| = 2$. The subtrees keeps grouping to maximize the minimum CGD at each level of set partitioning. Similar strategies are used for other signal constellations. Figures 2.2, 2.3, and 2.4 are concluded, as the minimum CGD increases (or remains the same) as one level down in the tree moves. The branches at each level can be used to design a trellis code with a specific rate. Higher coding gain necessitates the use of redundancy resulting in reduced rate.

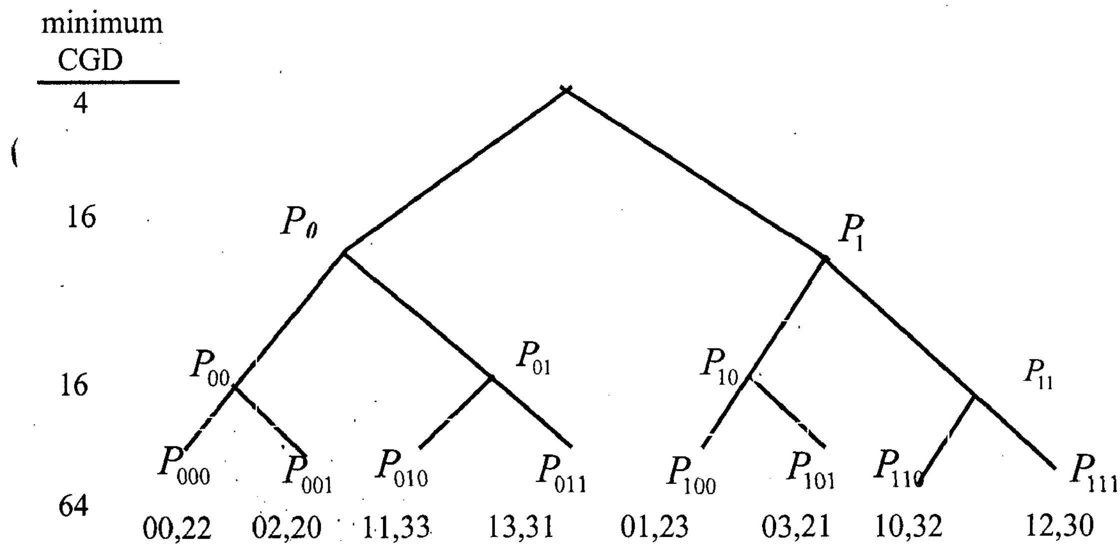

Fig. 2.3. Set partitioning of QPSK; each pair of binary numbers represents the pair of symbol indices in a STBC

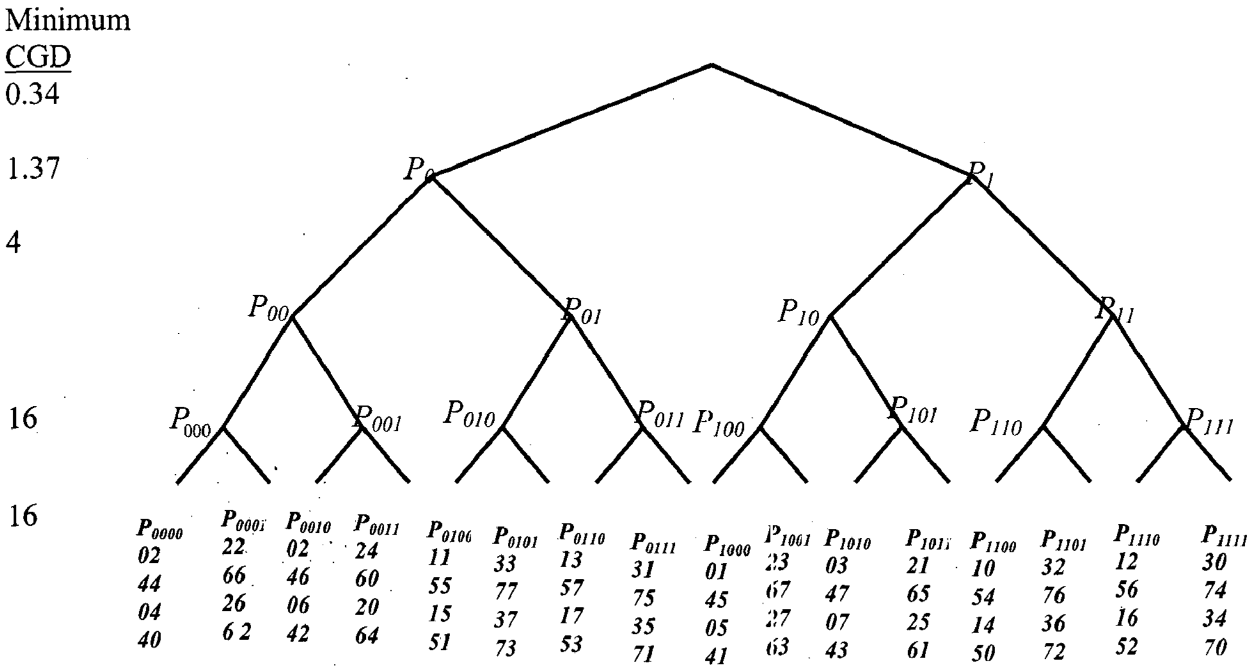

Fig. 2.4. Set partitioning for 8-PSK ; each pair of binary numbers represents the pair of symbol indices in a STBC

2.5.3 Set partitioning for super-orthogonal codewords

This section provides set partitioning for super-orthogonal codes and shows how to maximize the coding gain without sacrificing the rate. Code construction based on a super-orthogonal set is as follows. A constituent STBC be assigned to all transitions from a state. The adjacent states are typically assigned to one of the other constituent STBCs from the super-orthogonal code. Similarly, the same STBC can be assigned to branches that are merging into a state.

It is thus assured that every pair of codewords diverging from (or merging to) a state achieves full diversity because the pair is from the same orthogonal code with the same rotation parameter θ. On the other hand, for codewords with different θ, it is possible that they do not achieve full diversity. Since these codewords are assigned to different states, the resulting trellis code would provide full diversity despite the fact that a pair of codewords in a super-orthogonal code may not achieve full diversity. It is possible to come up with Designs that do not follow this rule and still provide full diversity. Similar to the case of orthogonal designs, it remains to do the set partitioning such that the minimum CGD is maximized at each level of partitioning.

This set partitioning should be done for all possible orthogonal 2×2 codewords, for every possible rotation. In other words, the set of all possible 2×2 matrices are needed to partition, generated by the class of codes in *(2.22)*. Based on the design criteria in Section (2.1), first, to achieve full diversity be needed to make sure. Therefore, first, the set of all codewords are partitioned into subsets with the same rotation. In other words, the first step of the set partitioning is only based on the rotation parameter θ. Then, the set of all codewords should be partitioned with the same rotation parameter θ as the case of orthogonal designs with $\theta = 0$ are obtained.

In what follows, the set partitioning for the sets with different rotations can shown, results in the same set partitioning trees as that of the orthogonal design with $\theta =$ 0 in Section (2.5.2). Similar to the case of $\theta = 0$, two distinct pairs of constellation symbols $(s_1^1 = e^{jk_1\omega}, s_2^1 = e^{jl_1\omega})$ and $(s_1^2 = e^{jk_2\omega}, s_2^2 = e^{jl_2\omega})$ are considered.

The corresponding codewords are denoted by $C^{\theta\,1}$ and $C^{\theta\,2}$ and the corresponding difference matrix by D^{θ}.

For parallel transitions in a trellis, the following be founded

$$D^{\theta} = \begin{pmatrix} e^{jk_1\omega}e^{j\theta} - e^{jk_2\omega}e^{j\theta} & e^{jl_1\omega} - e^{jl_2\omega} \\ e^{-jl_2\omega}e^{j\theta} - e^{-jl_1\omega}e^{j\theta} & e^{-jk_1\omega} - e^{-jk_2\omega} \end{pmatrix} \qquad (2.31)$$

$$= D.U$$

where, U is the rotation matrix in *(2.21)* and D is the difference matrix for $\theta = 0$ in *(2.24)*. To calculate the CGD, matrix $A^{\theta} = D^{\theta H} \cdot D^{\theta}$ is needed using *(2.31)*.

$$A^{\theta} = U^H D^H D U = U^H A U = A \qquad (2.32)$$

Where, $A = D^H \cdot D$ is a diagonal matrix calculated in *(2.25)* and $U^H \cdot A \cdot U = A$. As a result of *(2.32)*, the CGD between two codewords is only a function of the corresponding constellation symbols and the same for any rotation θ. Therefore, the formulas to calculate the CGDs have been developed in Section (2.5.2). A good set partitioning for $\theta = 0$ as presented in Section (2.5.2) is also good for any other rotation. Figures 2.2, 2.3, and 2.4 shows set partitioning for BPSK, QPSK, and 8-PSK, respectively.

So far, the set partitioning for super-orthogonal codes have represented by first partitioning based on the rotation parameter θ and then the constellation pairs. The rationale behind such a two step partitioning is the fact that, Section (2.1) be presented, the rank criterion be needed to consider first, to achieve full diversity and then the determinant criterion to obtain the highest possible coding gain. Since the second step of set partitioning is the same for all rotation parameters, the complete set-partitioning tree is not shown.

In fact, to specify a subset, the rotation parameter θ be mentioned or the corresponding orthogonal STBC $G(x_1, x_2, \theta)$ and a set from the set partitioning for $\theta = 0$. Another way of presenting the codewords is to specify the complete set-partitioning tree. Although the first representation is more compact and easier to use, yet Figure 2.5 illustrates the set partitioning of superorthogonal codewords for a BPSK constellation using rotations $\theta = 0, \pi$. Unlike Figure 2.2, the 2 × 2 matrices are used instead of the corresponding index pairs. A pair of indices and a rotation identifies the 2 × 2 codeword uniquely. Also, the superscripts in Figure 2.5. represent the rotation. Therefore, the left half of the tree in Figure 2.5. is the same as the tree in Figure 2.2 for $\theta = 0$. Similarly, the right half of the tree in Figure 2.5. is the same as the tree in Figure 2.2 for $\theta = \pi$. The compact representation in Figures 2.2, 2.3, and 2.4 uses this fact to remove a need for explicitly adding the parameter θ in the set partitioning process.

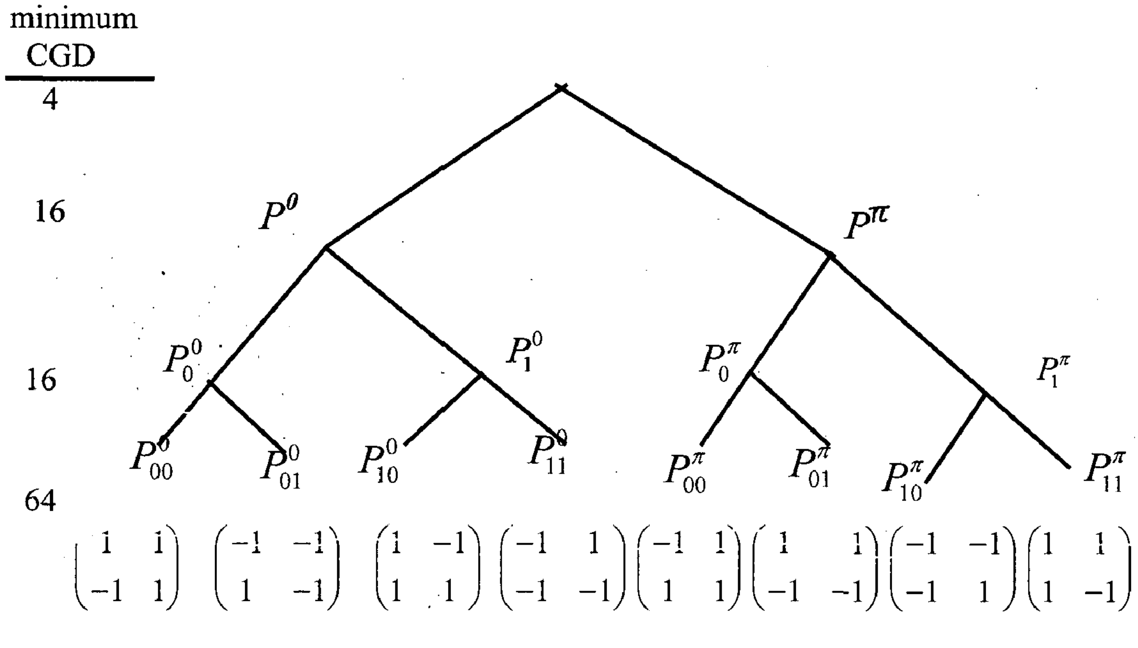

Fig.2.5. Set partitioning of super-orthogonal codewords for BPSK and θ= 0,π; complete tree

2.5.4 Design of SOSTTCs

In this section, the proposed set-partitioning scheme be use to design full-diversity full-rate space-time trellis codes. For each Design of SOSTTC, one subset be assigned to each branch of the trellis. Each subset represents a number of 2×2 matrices. Each subset corresponds to a rotation parameter and a set of possible symbol pairs. The superscript of the subset corresponds to the rotation parameter θ.

If the superscript of the subset is θ, $G(x1, x2, \theta)$ be used, the STBC in *(2.23)* with rotation parameter θ, to transmit. The set of possible symbol pairs for a given rotation parameter θ is the same as that of the same subset with superscript $\theta = 0$. For Design, for QPSK corresponds to $G(x1, x2, \pi)$ and the subset P_{10} in Figure 2.3.

Lemma 2.1 For a STTC with rate **b** bits/s/Hz and a diversity r, at least $2^{b(r-1)}$ states are needed [1].

Design 2.5.1 The fig 2.6 contains two different representation of the same code. In the left representation, the STBC G(x1, x2, θ) is explicitly mentioned and the subsets are from Figure 2.2 or 2.3. In the right representation, the rotation parameter is the superscript of the corresponding subset. Figure 2.6 demonstrates the codes for a two-state trellis providing a minimum CGD of 48(dominant path when path P=2 ,*(3.14)*) and 16 at rates 1 bit/s/Hz, using BPSK and 2 bits/s/Hz using QPSK, respectively. There is no equivalent two-state space-time trellis code (STTC) [13]using QPSK due to Lemma 2.1.

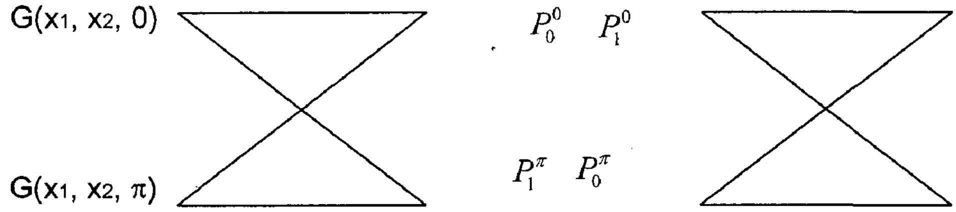

Fig. 2.6 A two-state code SOSTTC ; r=1bit/s/Hz using BPSK or r=2 bit/s/Hz using QPSK

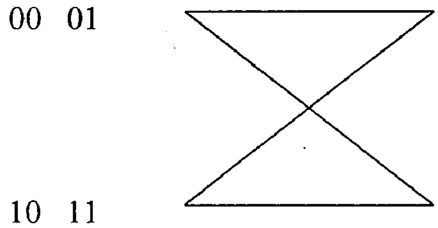

Fig. 2.7. A two-state code STTC ; b=1bit/s/Hz using BPSK

The minimum value of CGD among all possible pairs of codewords is used as an indication of performance of the code.In the case STTCs, any valid codeword starts from state zero and ends at state zero.Consider p=2 transitions STTC Figure (2.7) .First path stays in state zero during both transitions.The second path goes to state one in first transition and merge in the second transition.The corresponding codewords are

$$C^1 = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \quad \text{and} \quad C^2 = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$$

Therefore, CGD ($D^H D$) is16. This value is less than with respect to two states SOSTTC using BPSK.

Design 2.5.2 Figure 2.8 shows a four-state, design of SOSTTC. In the Figure 2.8 representation, the rotation parameter is the superscript of the corresponding subset.

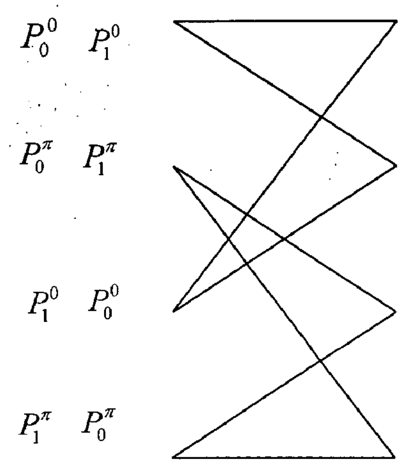

Fig.2.8. A four-state code SOSTTC; r=1 bit/s/Hz using BPSK or r=2 bit/s/Hz using QPSK

When BPSK uses and the corresponding set partitioning in Figure 2.2 the rate of the code is one. $G(x_1, x_2, 0)$ be also used, when departing from states zero and two and use $G(x_1, x_2, \pi)$ when departing from states one and three. The minimum CGD of this code is 64 which can be found in Figure 2.2 and Table 2.1. In Section (2.4.3), parallel transitions are dominant in calculating the minimum CGD for this code. If QPSK constellation be used and the corresponding set partitioning in Figure 2.3, the result is a four-state SOSTTC at rate 2 bits/s/Hz. The minimum CGD for this 2 bits/s/Hz code is equal to 16 which is greater than 4, the CGD of the corresponding STTC.

Let the generating matrices for the four state STTCs Figure 2.9 are

$$G = \begin{pmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{pmatrix} \quad \text{and} \quad G = \begin{pmatrix} 0 & 2 \\ 0 & 1 \\ 2 & 0 \\ 1 & 0 \end{pmatrix}.$$

In the Figure 2.9 with BPSK, assume the pairs of paths are 0000 and 0120. The corresponding codewords are

$$C^{1'} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad \text{and} \quad C^{2'} = \begin{pmatrix} 1 & -1 & -1 \\ -1 & 1 & -1 \end{pmatrix}.$$

Thus, CGD is 48, which is less than with respect to four states SOSTTC with BPSK.

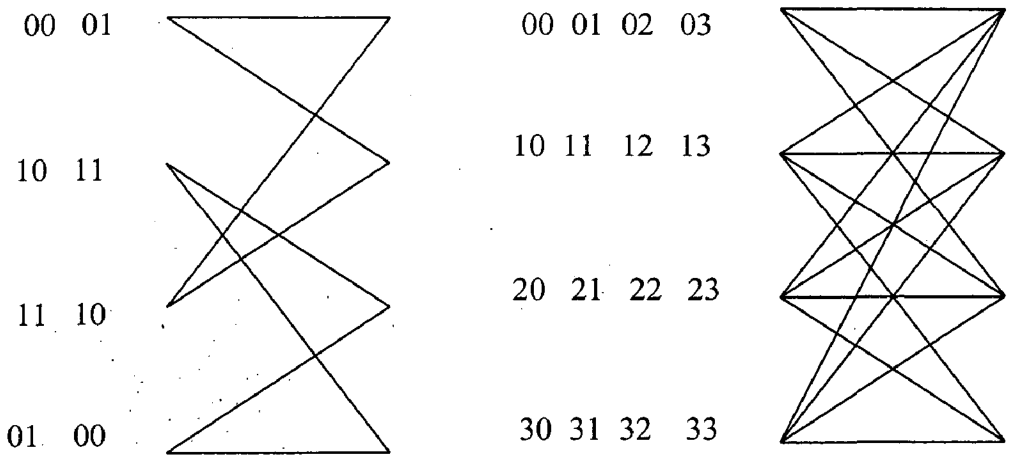

Fig. 2.9. A four-state code STTC; r=1 bit/s/Hz and r=2 bit/(s/Hz) using BPSK and QPSK respectively

In Figure 2.9 with QPSK case, assume the pairs of paths are 000 and 010. The corresponding codewords are

$$C^1 = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \text{ and } C^2 = \begin{pmatrix} 1 & j \\ j & 1 \end{pmatrix}$$

Hence, CGD is 4, which is less than with respect to four states SOSTTC with QPSK.

Design 2.5.3 Figure 2.10 shows 3 bits/s/Hz SOSTTC using 8-PSK and the corresponding set partitioning of Figure 2.4. The minimum CGD for this code is equal to 2.69. Detail analysis of the coding gain calculations is presented in Chapter 3. Based on Lemma 2.1, there is no four-state STTC with 8-PSK.

STTCs[11-17] are designed manually or by computer search strategies, and SOSTTCs are systematically designed a code for an arbitrary trellis, and rate using the above set partitioning. While so far designs of full-rate codes have been described, in general, codes with lower rates can be designed to provide higher coding gains. The design method is exactly the same while using different subsets. Utilizing different levels of the set partitioning to design SOSTTCs provides a trade-off between rate and coding gain as well. One design of a non full-rate SOSTTC with higher coding gain is provided below Figure 2.11.

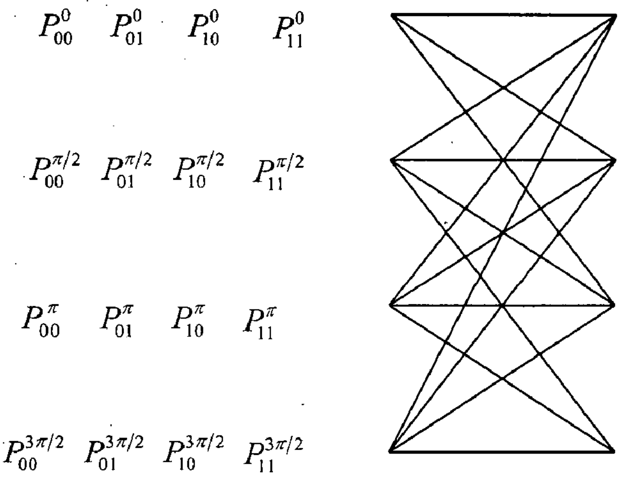

Fig. 2.10. A four-state code SOSTTC; r=3 bit/s/Hz (8-PSK).

In general, for a b-bit constellation, 2^{2b} pairs of symbols exist to be partitioned at the leaves of the tree. Half of these pairs can be used for transmission from each state of Figure 2.11 over two time slots. The other half should be reserved for using in other states to avoid a catastrophic code. Therefore, this code can transmit $2^{2b}/2 = 2^{2b-1}$ codewords or $2b - 1$ bits per two time slots. As a result, the rate of the code is $(b - 0.5)$ bits/s/Hz for a b-bit constellation and the code cannot transmit the maximum possible rate of b bits/s/Hz.

Design 2.5.4 A four-state rate 2.5 bits/s/Hz code using 8-PSK with a CGD of 4 is shown in Figure 2.11. The maximum possible rate using 8-PSK is 3 and a design that provides a CGD of 2.69 is shown in Figure 2.10.

Similar to conventional trellis codes, there is always a trade-off between coding gain and the complexity. Codes with different number of states and at different rates can be systematically designed using the set partitioning in Figures 2.2, 2.3, and 2.4 or similar set partitioning for other constellations. The rules for assigning different sets to different transitions in the trellis are similar to the general rules of thumb defined in [4] and [5] to design MTCM schemes. First based on the required rate, a constellation select and use the corresponding set partitioning. The choice of the constellation also defines the valid rotation angles, θ, that would not create an expansion of the transmitted signal constellation.

The corresponding set partitioning is used in the design of the trellis code. Since, presentation of the set partitioning was based on a two step algorithm, describe the resulting two-step design method in the sequel.

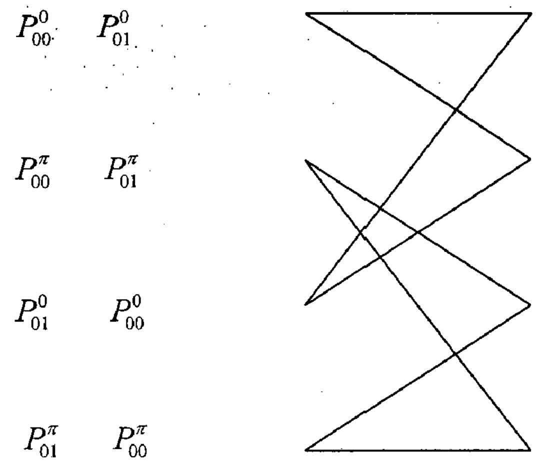

Fig. 2.11. A four-state code; r=2.5 bit/s/Hz (8-PSK).

Utilizing the valid values of θ in (2.23) defines the valid codewords in the set of super-orthogonal codes. Then, a constituent orthogonal code be assigned from the set of super-orthogonal codes to all transitions from a state. This is equivalent to assigning a rotation parameter to each state. Typically, another constituent orthogonal code be assigned or equivalent rotation parameter to the adjacent states. Similarly, the same orthogonal code can be assigned, equivalently the same rotation parameter, to branches that are merging into a state.

It is thus assured that any path that diverges from (or merges to) the correct path is full rank. Then, different sets from the set partitioning for $\theta = 0$, or any other rotation, are assigned to different transitions similar to the way that we assign sets in a regular MTCM. To avoid a catastrophic code, the subsets are assigned such that for the same input bits in different states either the rotation angles or the assigned subsets are different. Since the process of set partitioning is done to maximize the minimum CGD at each level, the upper bound on the coding gain of the resulting SOSTTC due to parallel transitions is also maximum.

Table 2.1. CGD values for different codes

Figure	No. of states	rate(bits/sec/Hz)	minimum CGD	Minimum CGD in [1]
2.8(BPSK)	4	1	64	-
2.8(QPSK)	4	2	16	4
2.10	4	3	2.69	-
2.5(BPSK)	2	1	48	-
2.5(QPSK)	2	2	16	-
2.11	4	2.5	4	-

Chapter 3

Analysis of CGD

In this chapter, the coding gain be derived of different SOSTTCs that, in Section 2.5 be introduced. To find the dominant path for CGD calculation in the trellis be needed. Section 2.5 be consider specific designs first, and calculate their coding gains. Then, it generalizes the methods that, the calculation of these coding gains have used to show how to calculate the coding gain of any SOSTTC.

3.1 Error events with path length of three

Let us first consider the trellis of the code in Figure 2.8. Parallel transitions between two states may be considered as different transitions each containing one possible 2×2 symbol matrix. Two codewords may only differ in $P = 1$ trellis transition if they both start and end in the same state. However, due to the structure of the trellis, it is impossible to have two codewords which differ in $P = 2$ trellis transitions.

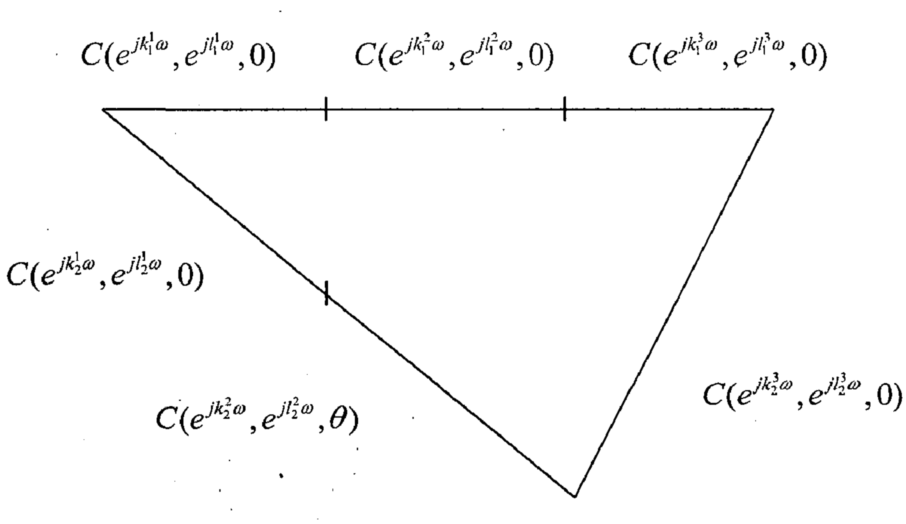

Fig. 3.1. Two typical paths differing in P=3 transitions

Because, for design, if two codewords diverge from state zero, they have to go through at least three transitions to remerge as shown in Figure 3.1. Therefore, the smallest value of P excluding parallel transitions is three. For P = 3, a typical case consider, where the first codeword stays at state zero. For the second codeword, the first and third transitions, diverging and merging to state zero, use $C(x_1, x_2, 0)$ and the second transition uses $C(x_1, x_2, \theta)$ as in Figure 3.1. Using similar calculations as equation *(2.26)* it can be obtained as

$$\det(A) = (a + b_1 + c)(a + b_2 + c) - d \qquad (3.1)$$

Where $a, b_1, b_2, c, d \geq 0$, and

$$\begin{aligned} a &= 4 - 2\cos[\omega(k_2^1 - k_1^1)] - 2\cos[\omega(l_2^1 - l_1^1)] \\ c &= 4 - 2\cos[\omega(k_2^3 - k_1^3)] - 2\cos[\omega(l_2^3 - l_1^3)] \\ b_1 &= 4 - 2\cos[\omega(k_2^2 - k_1^2) + \theta] - 2\cos[\omega(l_2^2 - l_1^2)] \\ b_2 &= 4 - 2\cos[\omega(k_2^2 - k_1^2)] - 2\cos[\omega(l_2^2 - l_1^2) + \theta] \\ d &= (2 - 2\cos\theta)(2 - 2\cos[\omega(k_2^2 - k_1^2 + l_1^2 - l_2^2) + \theta] \end{aligned} \qquad (3.2)$$

For $\theta = \pi$,

$$\begin{aligned} b_1 &= 4 + 2\cos[\omega(k_2^2 - k_1^2)] - 2\cos[\omega(l_2^2 - l_1^2)] \\ b_2 &= 4 - 2\cos[\omega(k_2^2 - k_1^2)] + 2\cos[\omega(l_2^2 - l_1^2)] \\ d &= 8(1 + \cos[\omega(k_2^2 - k_1^2 + l_2^2 - l_1^2)] \end{aligned} \qquad (3.3)$$

Since

$$a, b_1, b_2, c, d \geq 0,$$

$$\min\det(A) \geq (\min a + \min b_1 + \min c)(\min a + \min b_2 + \min c) - \max d \qquad (3.4)$$

Design 3.1.1 In this design, the four-state code considers in Figure 2.8. using BPSK and transmitting r = 1 bit/ s/Hz. For CGD analysis assume two codewords diverging from state zero and remerging after P transitions to state zero. For parallel transitions, that is P = 1, min det (A) = 64 is calculated from *(2.26)*. For a BPSK constellation, $\omega = \pi$, min a = min b_1 = min b_2 = min c = 4, and max d = 16 is used. Therefore, Inequality *(3.4)* results in

$$\min\det(A) \geq 128 \qquad (3.5)$$

Also, for $k_2^1 = k_1^1 = l_1^1 = k_2^2 = k_1^2 = l_1^2 = l_2^2 = k_2^3 = k_1^3 = l_1^3 = 0$ *and* $l_2^1 = l_2^3 = 1$ in *(3.1)*, det(A) =128 is obtained , which means

$$\min \det(A) \leq 128 \qquad (3.6)$$

Combining Inequalities *(3.5)* and *(3.6)* provides

$$\min \det(A) \approx 128 \qquad (3.7)$$

which is greater than 64. Similarly, the minimum value of the CGD when P > 3 is greater than the minimum value of the CGD when P = 3 is obtained . This proves that the minimum CGD for the code is dominated by parallel transitions and is equal to 64.

Design 3.1.2 In this design, the four-state code in Figure 2.8 considers using QPSK and transmitting r = 2 bits/ s/Hz . Again, two codewords are assumed diverging from state zero and remerging after P transitions to state zero. Using *(2.26)*, min det (A)=16 is calculated for parallel transitions. Also, $\omega = \pi/2$, min a = min c = 2, min b_1 = min b_2 = 4, and max d = 16 is used. Therefore, Inequality *(3.4)* results in

$$\min \det(A) \geq 48 \qquad (3.8)$$

Also, for $k_2^1 = k_1^1 = l_1^1 = k_2^2 = k_1^2 = l_1^2 = l_2^2 = k_2^3 = k_1^3 = l_1^3 = 0$ *and* $l_2^1 = l_2^3 = 1$ in *(3.1)*, det(A) = 48 is obtained, which means

$$\min \det(A) \leq 48 \qquad (3.9)$$

combining Inequalities *(3.8)* and *(3.9)* shows

$$\min \det(A) = 48 \qquad (3.10)$$

which is greater than 16. Again, the minimum value of the CGD when P > 3 is greater than the minimum value of the CGD when P = 3. Therefore, the minimum CGD of the code is dominated by parallel transitions and is equal to 16.

3.2 Error events with path length of two

Two codewords are considered diverging from state zero and remerging after P = 2 transitions to state zero in Figure 3.2. For parallel transitions, that is P = 1, the CGD can be calculated from *(2.26)*.

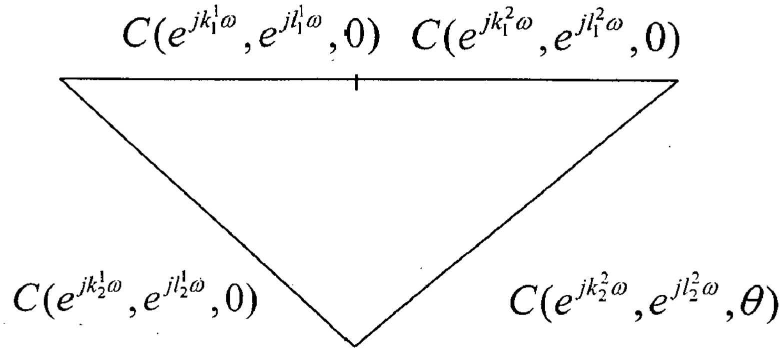

Fig. 3.2. Two typical paths differing in P=2 transitions

A case is considered, where the first codeword stays at state zero. For the second codeword, the first transition diverging from state zero uses $C(x_1, x_2, 0)$ and the second transition remerging to state zero uses $C(x_1, x_2, \theta)$ as in Figure 3.2. The rotation parameters are used as $\theta = \pi/2, \pi, 3\pi/2$ in Figure 2.10, $\theta = \pi$ in Figures 2.6. It can be shown that

$$\det(A) = (a+b)(a+b_2) - d = a^2 + a(b_1+b_2) + b_1 b_2 - d \qquad (3.11)$$

Where

$$\begin{aligned} a &= 4 - 2\cos[\omega(k_2^1 - k_1^1)] - 2\cos[\omega(l_2^1 - l_1^1)] \\ b_1 &= 4 - 2\cos[\omega(k_2^2 - k_1^2) + \theta] - 2\cos[\omega(l_2^2 - l_1^2)] \\ b_2 &= 4 - 2\cos[\omega(k_2^2 - k_1^2)] - 2\cos[\omega(l_2^2 - l_1^2) - \theta] \\ d &= (2 - 2\cos\theta)(2 - 2\cos[\omega(k_2^2 - k_1^2 + l_1^2 - l_2^2) + \theta] \end{aligned} \qquad (3.12)$$

and a, b_1, b_2, $d \geq 0$. Since a only depends on $(k_2^1, k_1^1, l_2^1, l_1^1), b_1 + b_2$ is not negative, and b_1, b_2, d are independent of $(k_2^1, k_1^1, l_2^1, l_1^1)$ one can first calculate (min a) and then calculate min det(A). The following formula be used to find min det(A):

$$\min\det(A) = (\min a)^2 + \min[(\min a)(b_1 + b_2) + b_1 b_2 - d] \qquad (3.13)$$

Equation *(3.13)* is used to calculate the CGD for trellises with $P = 2$ and also tabulate for different constellations and rotations (θ) in Table 3.1. In Table 3.1, the notations of Figure 3.2 are used.

where, with the notation $(k_1^1, l_1^1) \in P_1^1, (k_1^2, l_1^2) \in P_1^2, (k_2^1, l_2^1) \in P_2^1 and (k_2^2, l_2^2) \in P_2^2$.

Design 3.2.1 In this design, the coding gain is calculated for the BPSK code in Figure (2.6). Two codewords are assumed diverging from state zero and remerging after P transitions to state zero.

For parallel transitions, P = 1, the CGD can be calculated from *(2.26)* which is 64. For P = 2, min a=4 and

$$\min \det(A) = 16 + \min[4(b_1 + b_2) + b_1 b_2 - d] = 48 < 64 \quad (3.14)$$

Therefore, CGD=48 and the coding gain is dominated by paths with P = 2 transitions.

Table 3.1. Minimum det(A) from (3.13) for different constellations and rotations

Constellation	θ	P_1^1, P_1^2	P_2^1	P_2^2	min det(A)	Parallel CGD
BPSK	π	P_0	P_1	P_1	48	64
BPSK	π	P_0	P_1	P_0	48	64
QPSK	π	P_0	P_1	P_1	24	16
QPSK	π	P_0	P_1	P_0	20	16
QPSK	π/2, 3π/2	P_0	P_1	P_1	12	16
8-PSK	π	P_{00}	P_{10}, P_{11}	P_{00}	5.03	4
8-PSK	π	P_{00}	P_{10}	P_{10}	5.37	4

Design 3.2.2 For the QPSK code in Figure (2.6), the CGD for parallel transitions, P = 1, is 16. For P = 2, since min a = 2,

$$\min \det(A) = 4 + \min[2(b_1 + b_2) + b_1 b_2 - d] = 20 > 16 \quad (3.15)$$

Therefore, CGD of Figure 2.6 with QPSK is 16.

Design 3.2.3 The coding gain in Figure 2.10 is dominated by paths with P = 2 transitions. Using *(3.13)* and Table 3.1, when P=1, the minimum CGD is 2.69 as is tabulated in Table 2.1.

The importance of picking the right set of rotations are emphasized, and subsets in providing the maximum coding gain. The minimum CGD is not always enough in comparing two codes with each other. As an Design, the four-state be considered, rate 3 bits/s/Hz code in Figure 3.3. θ = 3π/4, 5π/4 are selected for states one and three in Figure 3.3 instead of θ = π/2, 3π/2 in Figure 2.10.

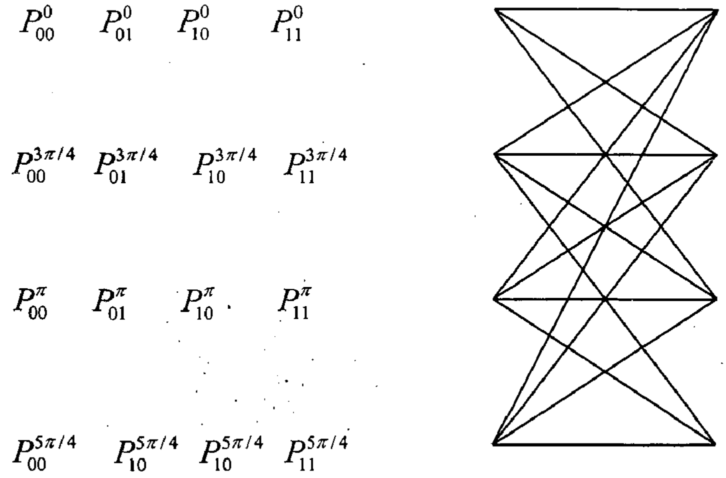

Fig. 3.3. A four-state code SOSTTC; r=3 bit/s/Hz (8-PSK).

The code in Figure 3.3 is an interesting design, where starting from state zero, parallel transitions are dominant and the CGD is 4. However, other error events should be considered with length two, where the parallel transitions are not dominant. For Design, starting from state one and staying at state one for two transitions, an error event with $P = 2$ can be a path diverging from state one to state two and remerging to state one in the second transition. This error event can be part of two codewords with length four as illustrated in Figure 3.4. The CGD for such a path is 1.03 which is lower than 2.69 and 4. There are other paths with $P = 2$ providing different CGDs. Therefore, for the code in Figure 3.4, minimum CGD is not a good indicator of the performance. In addition to minimum CGD, one needs to consider the path weights, that is multiplicity of error events.

3.3 Decoding Techniques

In general, it is typically more difficult an M-TCM scheme than a single-dimension TCM counterpart because a potentially much larger number of transition branches are required for an M-TCM encoder to achieve the same data rate. This can be a main disadvantage of an M-TCM scheme.

In this dissertation, the concept of orthogonality has been used to ease data decoding since the orthogonal space –time block code is used as the inner code.

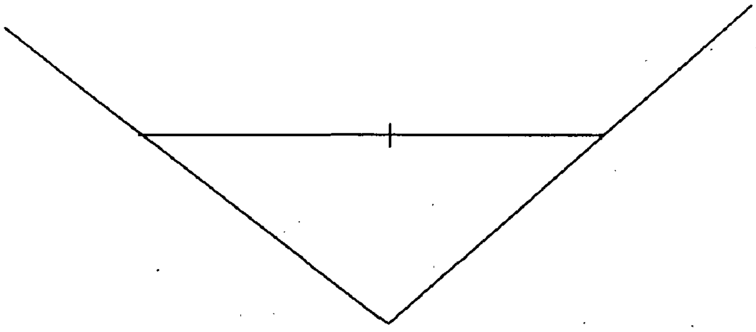

Fig. 3.4. Two paths with length four differing in P=2 transitions

3.3.1 Simplified ML Decoder

Essentially, the decoding process for current concatenated scheme can be divided into 2 parts: parallel-transition and non-parallel transition processing. Once the former task is completed, the latter is straightforward via the Viterbi algorathim. The parallel-transition processing, which include likelihood computation and determination of the most-likely branch among a partition of parallel transitions is typically more complicated but can be greatly eased by exploiting signal orthogonality.

The standard reduced-complexity ML decoder for an orthogonal space-time block code was a provided in [6]. It utilizes the signal orthogonality property to express the likelihood function in terms of a sum of functions that depend only on each information symbol encoded by the block code. Therefore, these symbols can be decoded individually, given that they are independent to one another. Unlike in the case of a generic joint ML decoder, the complexity of the standard simplified ML decoder does not grow exponentially with data rate and code block size.

First, block code can decompose a given a partition transitions into smaller subsets of transitions labeled with codewords that correspond to the outputs of a standalone block code with inputs drawn from smaller constellations. The standard simplified ML decoding algorithm can then be used to determine the most-likely branch within each subset of parallel transitions[21] and [22]. This greatly reduces decoding complexity, especially when the cardinality of these subsets is large. At this point, several local ML decisions are obtained (one from each subset of parallel transitions).

The final ML decision for the current parallel-transition is obtained by an exhaustive search for the candidate with the maximum value of the joint likelihood function. This algorithm is repeated for every distinct partition of parallel transitions. The decomposition of parallel transitions is done to enable reduced-complexity decoding algorithm with the set partitioning concept to improve coding gain, although they are closely related.

To demonstrate the idea of the simplified ML decoding finds the most likely valid path that starts from state zero and merges to state zero after $L + Q$ blocks, where L is the total number of transmitted orthogonal blocks and Q is memory size of trellis, to guarantee a return to state zero. Each state transition corresponds to the transmission of two symbols in two time slots from two transmit antennas. For receive antenna m be assumed, $r^l_{1,m}, r^l_{2,m}$ are received at the two time slots of block l. Assuming fixed path gains $\alpha_{1,m}$ and $\alpha_{2,m}$ throughout the transmission of a codeword, obtained as

$$\begin{aligned} r^l_{1,m} &= \alpha_{1,m} s^l_1 e^{j\theta^l} + \alpha_{2,m} s^l_2 + \eta^l_{1,m} \\ r^l_{2,m} &= -\alpha_{1,m} (s^l_2)^* e^{j\theta^l} + \alpha_{2,m} (s^l_1)^* + \eta^l_{2,m} \end{aligned} \qquad (3.16)$$

where, $\eta^l_{1,m}$ and $\eta^l_{2,m}$ are noise samples for block l. The Viterbi algorithm can be used for the ML decoding of SOSTTCs. Since, the parallel transitions for SOSTTCs are allowed, at each step, the best transition among all parallel transitions should find. Then, the best parallel transition be used to calculate the path metrics in the Viterbi algorithm. The branch metric of the lth block is given by

$$\sum_{m=1}^{M} \left| r^l_{1,m} - \alpha_{1,m} s^l_1 e^{j\theta^l} - \alpha_{2,m} s^l_2 \right|^2 + \left| r^l_{2,m} + \alpha_{1,m} (s^l_2)^* e^{j\theta^l} - \alpha_{2,m} (s^l_1)^* \right|^2 \qquad (3.17)$$

Expanding the branch metric in *(3.17)* and removing the constant terms results in the following branch metric:

$$\sum_{m=1}^{M} 2R\left\{ r^l_{2,m} \alpha^*_{1,m} (s^l_2) e^{-j\theta^l} - r^l_{2,m} \alpha^*_{2,m} (s^l_1) - r^l_{1,m} \alpha^*_{1,m} (s^l_1)^* e^{-j\theta^l} - r^l_{1,m} \alpha^*_{2,m} (s^l_2)^* \right\} \qquad (3.18)$$

For each block, the rotation, the path gains, and the received signals are known at the receiver. Therefore, the branch metric in *(3.18)* is only a function of transmitted signals and can be denoted by $2J^l(s^l_1, s^l_2)$. Therefore, the first step is to find the valid pair (s^l_1, s^l_2) among all parallel transitions that minimizes $J^l(s^l_1, s^l_2)$. To use the orthogonality of the STBC, one may utilize the fact that is a function of s^l_1 and s^l_2. . Similar to the case of an orthogonal STBC, this can be used to simplify the search for the transition with the minimum branch metric among all parallel transitions.

$$J_1^l(s_1^l, s_2^l) = J_1^l(s_1^l) + J_2^l(s_2^l)$$
$$where$$
$$J_1^l(s_1^l) = -R\left\{[r_{2,m}^l \alpha_{2,m}^* + (r_{1,m}^l)^* \alpha_{1,m} e^{j\theta^l}] s_1^l\right\}$$
$$J_2^l(s_2^l) = -R\left\{[r_{2,m}^l \alpha_{1,m}^* e^{-j\theta^l} - (r_{1,m}^l)^* \alpha_{2,m}] s_2^l\right\} \qquad (3.19)$$

However, while the orthogonality of the blocks results in a simpler ML decoding, the derivation of such a simpler ML decoding is not the same as that of a block code. Although, $J_1^l(s_1^l)$ is only a function of s_1^l since not all possible (s_1^l, s_2^l) pairs are allowed for each trellis transition, s_1^l and s_2^l are not independent. Similarly, $J_2^l(s_2^l)$ is a function of s_1^l and because of the correlation between s_2^l and s_1^l . In other words, the input symbols of the orthogonal STBC are not independent and therefore the separate decoding is not possible. For each branch, among all possible parallel transitions, the valid pair (s_1^l, s_2^l)) should found, that provides the minimum $J^l(s_1^l, s_2^l)$. If the symbol s_1^l should found, that minimizes $J_1^l(s_1^l)$) and the symbol s_2^l that minimizes $J_2^l(s_2^l)$, the resulting pair of symbols (s_1^l, s_2^l) may not be a valid pair for the considered transition. Therefore, such a pair does not provide the best pair among all possible parallel transitions. On the other hand, it is possible to divide the set of all parallel pairs to subsets for which the symbol pairs (s_1^l, s_2^l) are independent. Then, the simple separate decoding, or equivalently separate minimization of $J_1^l(s_1^l)$) and $J_2^l(s_2^l)$, is done for each subset. Finally, the best pair can be found that minimizes $J^l(s_1^l, s_2^l)$ among the best pairs of different subsets. In other words, to utilize the reduced complexity, the set partitioning be needed to combined and the separate decoding for each orthogonal STBC building block. To elaborate the simplified decoding method, the subsets are provided for which the symbols s_1^l and s_2^l are independent for different PSK constellations.

For any other constellation, the set partitioning can be combined and the separate decoding for each orthogonal STBC building block in a similar way. Utilizing such a combination makes it possible to reduce the complexity of the branch metric calculations for parallel transitions. After finding the best parallel branch for each transition, the ML decoder finds the most likely path.

First, the path metrics are needed to calculate using the best branch metrics $J^l(s_1^l, s_2^l)$. The path metric of a valid path is the sum of the branch metrics for the branches that form the path. The most likely path is the one which has the minimum path gain. The ML decoder finds the set of constellation symbols (s_1^l, s_2^l) , $l = 1, 2, \ldots, L + Q$ that constructs a valid path and solves the following minimization problem with all symbols $s_1^1, s_2^1, s_1^2, s_2^2, \ldots, s_1^{L+Q}, s_2^{L+Q}$

$$\sum_{l=1}^{L+Q} J^l(s_1^l, s_2^l) \qquad (3.20)$$

A valid path starts at state zero and returns to state zero after $L + Q$ transitions. Viterbi algorithm implemented on *(3.20)* gives the optimal path [21].

Chapter 4

Algorithm and Simulation Results

4.1 Algorithm of SOSTTCs

METHOD FOR DESIGING A SOSTTC FOR DIVERSE TRANSMISSION COMMUNICATIONS SYSTEM

↓

SELECT A DIVERSITY FACTOR FOR THE COMMUNICATIONS SYSTEM

↓

SELECT A RATE FOR THE COMMUNICATIONS SYSTEM

↓

SELECT A NUMBER OF STATES FOR THE CODE

↓

SELECT A SOSTTC HAVING THE SELECTED NUMBER OF STATES

↓

SELECT A SOSTTC SET PARTITIONING STRUCTURE THAT CORRESPONDS TO THE SELECTED RATE

↓

USE THE SELECTED TRELLIS STRUCTURE AND THE SELECTED SET PARTITIONING STRUCTURE TO ASSIGN PARTICULAR VALUESS TO A SET OF TRANSMISSION VARIABLE FOR EACH POSSIBLE COBINATION OF INPUT VALUES

A diversity factor is selected for the communication system to be implemented. This diversity factor represents the number of transmission elements to be included in the transmitters of the system. Generally, a diversity factor of two is selected. Also, a rate is selected for the communications system or code to be implemented. In this thesis, a rate of 1,2, or 3 bits/second/hertz is selected. The selected rate represents the number of transmitted in the given period of time. Once the rate has been selected, other aspects of the communications system and code are fixed. For design, a rate of 1 bit/second/hertz means that the system will have a constellation size of 2 (a BPSK system).A rate of 2 bit/second/hertz means the system will have a constellation size of 4(a QPSK system).

In general, the constellation size(L) will equal 2^b where b represents the selected rate. The selection of a number of states for the code, in this thesis , the number of sates is 2,4, or 8.Selecting a higher states for a given rate selection improves the performance of the code. For design, selecting a higher of states will improve coding gain distance (CGD).The set partitioning structure Figs 2.2 - 2.4 will decide the CGD. In this thesis selects a trellis structure according to the requirement that has the number of selected states. As described above, Figs 2.6 - 2.11 illustrates trellis structures .Trellis structure Figure 2.6 is a two states trellis-structure. It can be used, for design different rate 1 bit/second/hertz and rate 2 bit/second/hertz. A set partitioning according to the structure is selected that corresponds to the rate.

4.2 Simulation of Two States SOSTTC

4.2.1 Two states SOSTTC with BPSK

The encoder receives one state bit (B_A) and one modulation bit (B_B) at a time T_0, and the transmission variables (X_1, X_2, Θ) equal. Trellis structure Figure 2.6 will now be used in combination with set partitioning structure Figure 2.2 to assign particular values to transmission variables[20] X_1, X_2, and Θ for each combination of input values.

Space-time block code $C(X_1, X_2, 0)$ is associated with state **1**, and space-time block code $C(X_1, X_2,\pi)$ is associated with state **2**.

Two states SOSTTC in state 1 with BPSK:

(0, 0, 0) if the encoder is in state **1**, state bit (B_A) equals 0, and modulation bit (B_B) equals 0.

(1, 1, 0) if the encoder is in state **1**, state bit (B_A) equals 0, and modulation bit (B_B) equals 1.

(0, 1, π) if the encoder is in state **1**, state bit (B_A) equals 1, and modulation bit (B_B) equals 0.

(1, 0, π) if the encoder is in state **1**, state bit (B_A) equals 1, and modulation bit (B_B) equals 1.

Two states SOSTTC in state 2 with BPSK :

The encoder receives one state bit (B_A) and one modulation bit (B_B) at a time T_0, and the transmission variables (X_1, X_2, Θ) equal:

(0, 0, 0) if the encoder is in state **2**, state bit (B_A) equals 1, and modulation bit (B_B) equals 0.

(1, 1, 0) if the encoder is in state **2**, state bit (B_A) equals 1, and modulation bit (B_B) equals 1.

(0, 1, π) if the encoder is in state **2**, state bit (B_A) equals 0, and modulation bit (B_B) equals 0.

(1, 0, π) if the encoder is in state **2**, state bit (B_A) equals 0, and modulation bit (B_B) equals 1.

Then, the encoded codewords of two states SOSTTC are modulated using BPSK. These modulated codewords of SOSTTC are passed through a quasi-static flat Rayleigh fading model for the channel. There are two path gains independent complex Gaussian channels and are fixed during the transmission of one frame. Each frame consists of 130 transmissions out of each transmit antenna.

Decoding of 2 states SOSTTC using BPSK :

For a BPSK constellation and the corresponding set partitioning in Figure 2.5, the sets P_0 = {00, 11} and P_1 = {01, 10} only contain two pairs . The root set P = {00, 11, 01, 10} provides independent symbols s_1^l and s_2^l. Since all possible combinations of indices 0 and 1 exist in this set, and minimize $J_1^l(s_1^l)$ and $J_1^l(s_1^l)$ separately. Any resulting symbol pair (s_1^l, s_2^l) will be a valid pair using modified ML Viterbi decoding technique. Finally, simulation results are plotted between the frame error rate (FER) of symbols and SNR (dB).

Simulation Parameters:

Transmit antenna (Nt) =2

Receive antenna (Nr) =1

Frame- size = 130

Total number of frames=1000

Channel = 2 independent complex Gaussian path-gains for quasi-static Rayleigh channel.

SNR= 10 dB to 22 dB

Result 1 : Simulation of 2 states SOSTTC using BPSK (2 transmit antenna and 1 receive antenna)

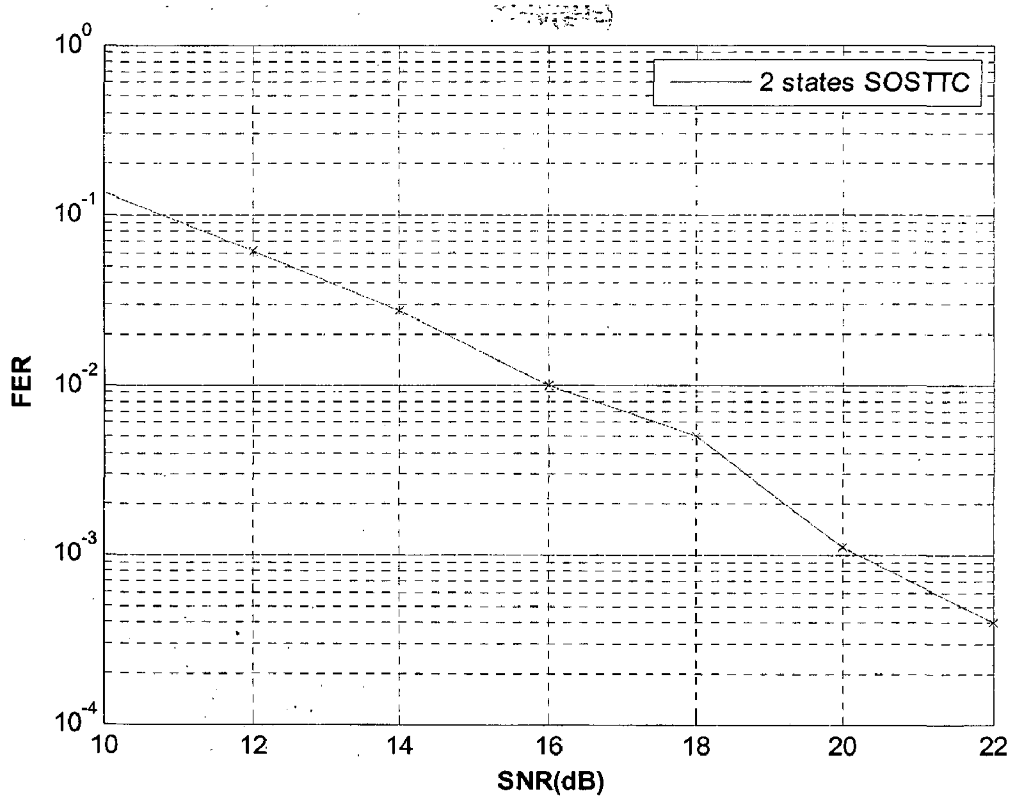

Fig. 4.1. Two states SOSTTC-BPSK with 2 transmit antenna and 1 receive antenna.

4.2.2 Two states SOSTTC using QPSK

The encoder receives one state bit (B_A) and three modulation bits (B_B) at a time T_0, and the transmission variables (X_1, X_2, Θ) equal. Trellis structure Figure 2.5 will now be used in combination with set partitioning structure Figure 2.3 to assign particular values to transmission variables [20] X1, X2, and Θ for each combination of input values.

Space-time block code $C(X_1, X_2, 0)$ is associated with state **1**, and space-time block code $C(X_1, X2, \pi)$ is associated with state **2**.

Two states SOSTTC in state 1 with QPSK :

(0, 0, 0) if the encoder is in state **1**, the state bit (B_A) equals 0, and the modulation bits (B_B) equal 000.

(2, 2, 0) if the encoder is in state **1**, the state bit (B_A) equals 0, and the modulation bits (B_B) equal 001.

(0, 2, 0) if the encoder is in state **1**, the state bit (B_A) equals 0, and the modulation bits (B_B) equal 010.

(2, 0, 0) if the encoder is in state **1**, the state bit (B_A) equals 0, and the modulation bits (B_B) equal 011.

(1, 1, 0) if the encoder is in state **1**, the state bit (B_A) equals 0, and the modulation bits (B_B) equal 100.

(3, 3, 0) if the encoder is in state **1**, the state bit (B_A) equals 0, and the modulation bits (B_B) equal 101.

(1, 3, 0) if the encoder is in state **1**, the state bit (B_A) equals 0, and the modulation bits (B_B) equal 110.

(3, 1, 0) if the encoder is in state **1**, the state bit (B_A) equals 0, and the modulation bits (B_B) equal 111.

(0, 1, π) if the encoder is in state 1, the state bit (B_A) equals 1, and the modulation bits (B_B) equal 000

(2, 3, π) if the encoder is in state 1, the state bit (B_A) equals 1, and the modulation bits (B_B) equal 001.

(0, 3, π) if the encoder is in state 1, the state bit (B_A) equals 1, and the modulation bits (B_B) equal 010.

(2, 1, π) if the encoder is in state **1**, the state bit (B_A) equals 1, and the modulation bits (B_B) equal 011.

(1, 0, π) if the encoder is in state **1**, the state bit (B_A) equals 1, and the modulation bits (B_B) equal 100.

(3, 2, π) if the encoder is in state **1**, the state bit (B_A) equals 1, and the modulation bits (B_B) equal 101.

(1, 2, π) if the encoder is in state **1**, the state bit (B_A) equals 1, and the modulation bits (B_B) equal 110.

(3, 0, π) if the encoder is in state **1**, the state bit (B_A) equals 1, and the modulation bits (B_B) equals 111.

Two state SOSTTC in state 2 with QPSK:

The encoder receives one state bit (B_A) and three modulation bits (B_B) at a time T_0, and the transmission variables (X_1, X_2, Θ) equal:

(0, 1, 0) if the encoder is in state **2**, the state bit (B_A) equals 1, and the modulation bits (B_B) equal 000.

(2, 3, 0) if the encoder is in state **2**, the state bit (B_A) equals 1, and the modulation bits (B_B) equal 001.

(0, 3, 0) if the encoder is in state **2**, the state bit (B_A) equals 1, and the modulation bits (B_B) equal 010.

(2, 1, 0) if the encoder is in state **2**, the state bit (B_A) equals 1, and the modulation bits (B_B) equal 011.

(1, 0, 0) if the encoder is in state **2**, the state bit (B_A) equals 1, and the modulation bits (B_B) equal 100.

(3, 2, 0) if the encoder is in state **2**, the state bit (B_A) equals 1, and the modulation bits (B_B) equal 101.

(1, 2, 0) if the encoder is in state **2**, the state bit (B_A) equals 1, and the modulation bits (B_B) equal 110.

(3, 0, 0) if the encoder is in state **2**, the state bit (B_A) equals 1, and the modulation bits (B_B) equal 111.

(0, 0, π) if the encoder is in state **2**, the state bit (B_A) equals 0, and the modulation bits (B_B) equal 000.

(2, 2, π) if the encoder is in state **2**, the state bit (B_A) equals 0, and the modulation bits (B_B) equal 001,.

(0, 2, π) if the encoder is in state **2**, the state bit (B_A) equals 0, and the modulation bits (B_B) equal 010.

(2, 0, π) if the encoder is in state **2**, the state bit (B_A) equals 0, and the modulation bits (B_B) equal 011.

(1, 1, π) if the encoder is in state **2**, the state bit (B_A) equals 0, and the modulation bits (B_B) equal 100.

(3, 3, π) if the encoder is in state **2**, the state bit (B_A) equals 0, and the modulation bits (B_B) equal 101.

(1, 3, π) if the encoder is in state **2**, the state bit (B_A) equals 0, and the modulation bits (B_B) equal 110.

(3, 1, π) if the encoder is in state **2**, the state bit (B_A) equals 0, and the modulation bits (B_B) equals 111.

Then, the encoded codewords of two states SOSTTC are modulated using QPSK. These modulated codewords of SOSTTC are passed through a quasi-static flat Rayleigh fading model for the channel. There are two path gains independent complex Gaussian random variables and are fixed during the transmission of one frame. Each frame consists of 130 transmissions out of each transmit antenna.

Decoding of 2 states SOSTTC using QPSK

For a QPSK constellation and the corresponding set partitioning in Figure 2.3, the sets P_{00}, P_{01}, P_{10}, and P_{11} provide independent pairs. If P_0 is used, first to divide it into two subsets $P_0 = P_{00} \cup P_{01}$ Then, to find the best pairs in P_{00} and P_{01} using the separate minimization of $J_1^l(s_1^l)$ and $J_1^l(s_1^l)$. Finally, to compare the branch metric $J^l(s_1^l, s_2^l)$ for these two pairs and find the minimum among the two. Similarly for P_1, first to divide it into two subsets $P_0 = P_{00} \cup P_{01}$ and find the minimum branch metric of each of these two subsets. Then, the best parallel transition between the two selected transitions is the one that has a lower branch metric. Finally, simulation results are plotted between the frame error rate (FER) and SNR (dB).

Simulation Parameters:

Transmit antenna (Nt) =2

Receive antenna (Nr) =1

Frame- size = 130

Total number of frames=1000

Channel = 2 independent complex Gaussian path-gains for quasi-static Rayleigh channel.

SNR= 10 dB to 22 dB

Result 2: Simulation of 2 states SOSTTC using QPSK (2 transmit antenna and 1 receive antenna)

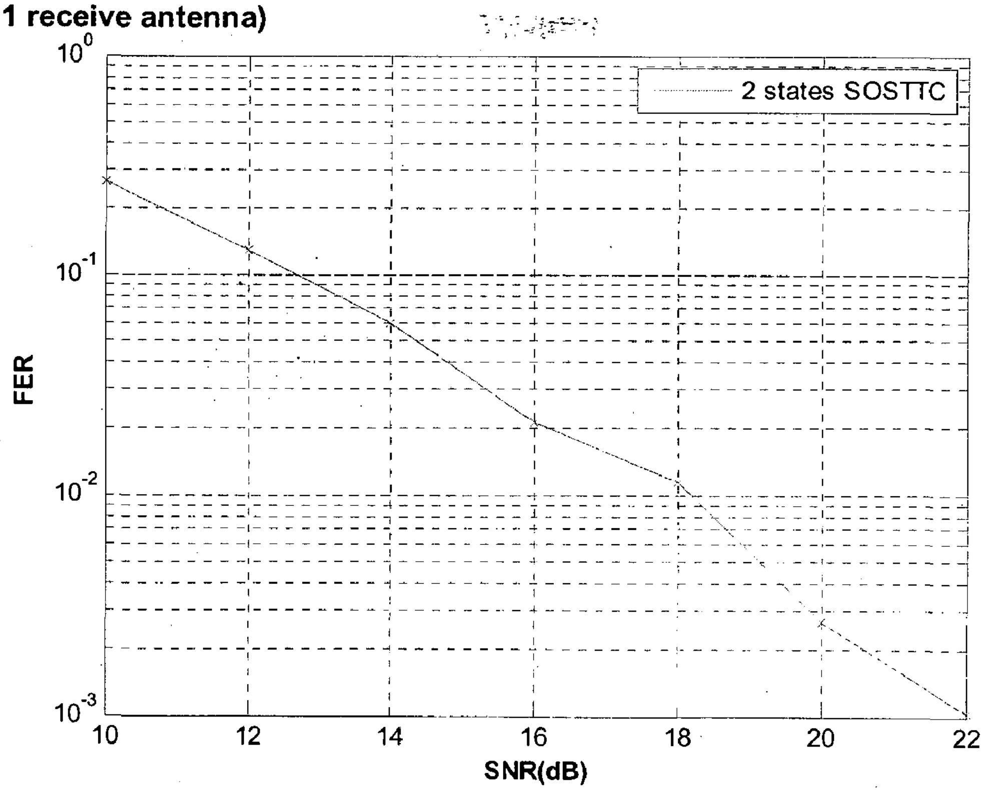

Fig. 4.2. Two states SOSTTC QPSK with 2 transmit antenna and 1 receive antenna.

The simulation results 1 and 2 are combined together. The more coding gain advantage SOSTTC with rate 1 bit/s/Hz, with respect to SOSTTC with rate 2 bits/s/Hz .The number of states is same for both case.The decoding complexity is larger in the case SOSTTC-QPSK(rate 2 bits/s/Hz). As, from the set partitioning structures , more number of parallel paths in trellis transitions are used in the case of modified Viterbi decoding. Hence, there is trade off between the coding gain and rate of codewords.

Result 3 : Trade off between Coding gain and rate of symbols.(1 receive antenna)

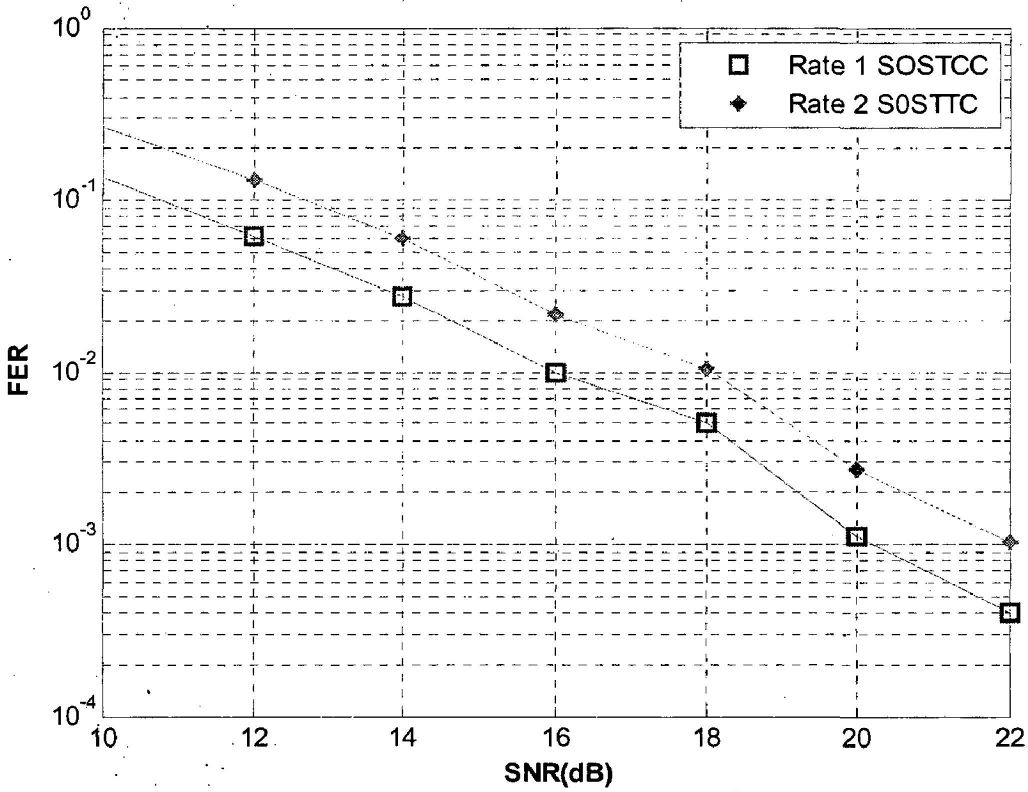

Fig. 4.3. Two states SOSTTCs BPSK and QPSK with 2 transmit antenna and 1 receive antenna.

Result 4: Trade off between Coding gain and rate of symbols.(2 receive antenna)

In this case, number of receiving antennas is two. The simulation gap between the paths is larger. As, there are four independent complex Gaussian path-gains used, the diversity factor becomes four. The coding gain distance parameter is more prominent, due to increase in receiving antenna number [18].

Simulation Parameters:

Transmit antenna (Nt) =2

Receive antenna (Nr) =2

Frame- size = 130

Total number of frames=1000

Channel = 4 independent complex Gaussian path-gains for quasi-static Rayleigh channel.

SNR= 10 dB to 22 dB

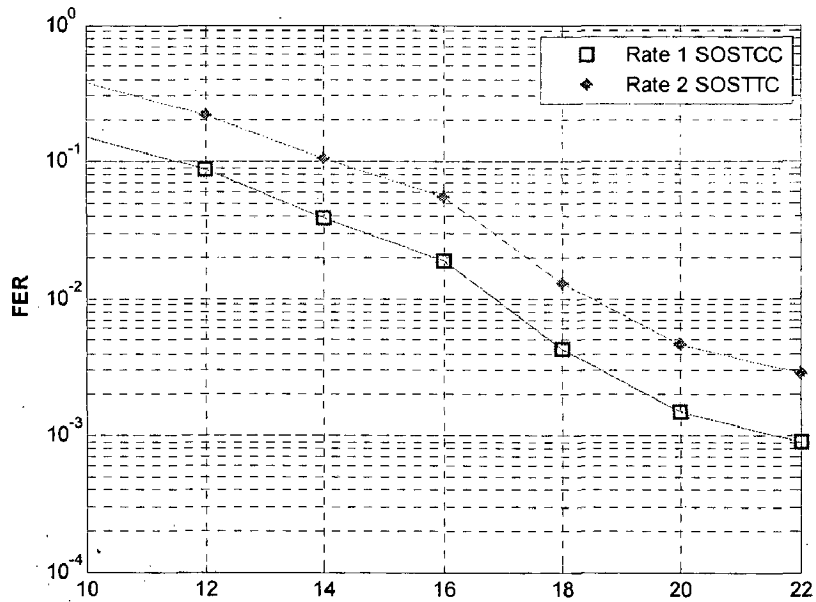

Fig. 4.4. Two states SOSTTC-BPSK and QPSK with 2 transmit antenna and 2 receive antenna.

Result 5: Advantage of coding gain 2 states SOSTTC with respect to 2 states STTC with QPSK.

The coding gain advantage of SOSTTC is better than the corresponding STTC. The constellation size of codewords is not expanded in case of SOSTTC, but orthogonal design matrices expand. But in the case, STTCs, the constellation size for transmitted codewords are increased. Analytically as shown in design (2.5), the more CGD of SOSTTC than corresponding STTC. The decoding complexity of STTC is more than SOSTTCs.

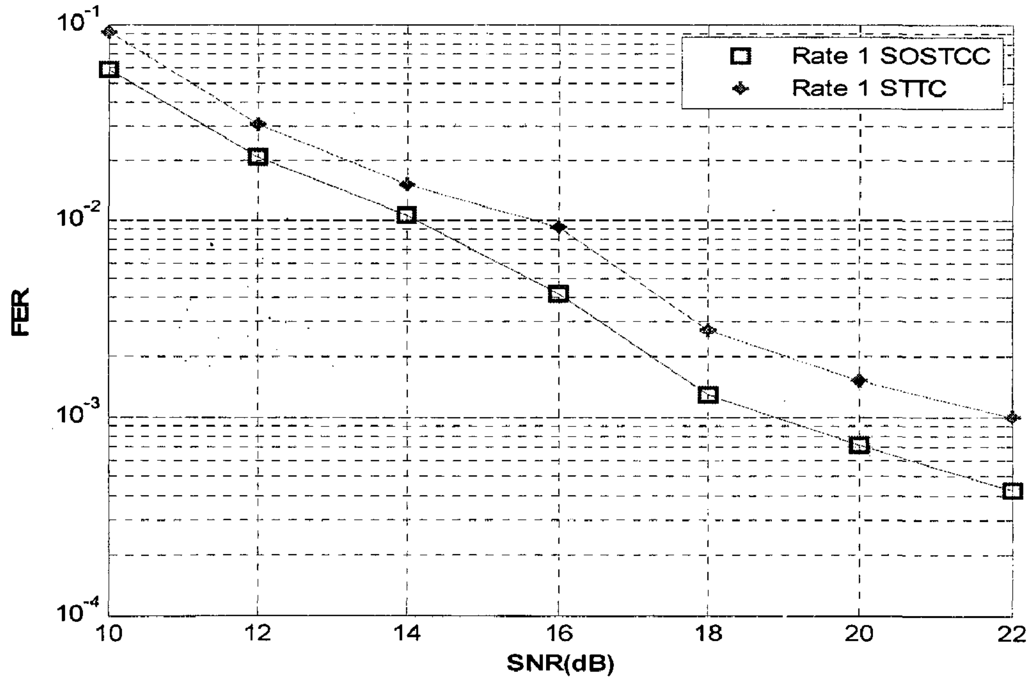

Fig. 4.5. Two states SOSTTC-BPSK and STTC-BPSK with 2 transmit antenna and 1 receive antenna.

4.3 Simulation of 4 states SOSTTCs

4.3.1 Four states SOSTTC using BPSK

The system of space-time block code $C(X_1, X_2, 0)$ is associated with state **1**, space-time block code $C(X_1, X_2, \pi)$ is associated with state **2**, space-time block code $C(X_1, X_2, 0)$ is associated with state **3**, and space-time block code $C(X_1, X_2, \pi)$ is associated with state **4**.

The encoder receives one state bit (B_A) and one modulation bit (B_B) at a time T_0, and the transmission variables (X_1, X_2, Θ) equal :

(0, 0, 0) if the encoder is in state **1**, state bit (B_A) equals 0, and modulation bit (B_B) equals 0.

(1, 1, 0) if the encoder is in state **1**, state bit (B_A) equals 0, and modulation bit (B_B) equals 1.

(0, 1, π) if the encoder is in state **1**, state bit (B_A) equals 1, and modulation bit (B_B) equals 0.

(1, 0, π) if the encoder is in state **1**, state bit (B_A) equals 1, and modulation bit (B_B) equals 1.

Similarly, the transmission variables are designed for state **2**, **3** and **4** as result(1). Then, these encoded codewords of two states SOSTTC are modulated with BPSK. These modulated codewords of SOSTTC are passed through a quasi-static flat Rayleigh fading model for the channel. There are two path gains independent complex Gaussian random variables and fixed during the transmission of one frame. Each frame consists of 130 transmissions out of each transmit antenna.

Similarly, decoding of 4 states SOSTTC with BPSK are carried out same as result (1).The optimal design of set partitioning Figure 2.2 and using orthogonality of STBC as in modified ML decoding done .Finally, simulation results are plotted between the frame error rate of symbols and SNR (10 to 22 dB).

Simulation Parameters:

Transmit antenna (Nt) =2

Receive antenna (Nr) =1

Frame- size = 130

Total number of frames=1000

Channel = 2 independent path-gains of complex Gaussian for Rayleigh channel.

SNR= 10 dB to 22 dB

Result 6: Simulation of 4 states SOSTTC using BPSK (2 transmit antenna and 1 receive antenna)

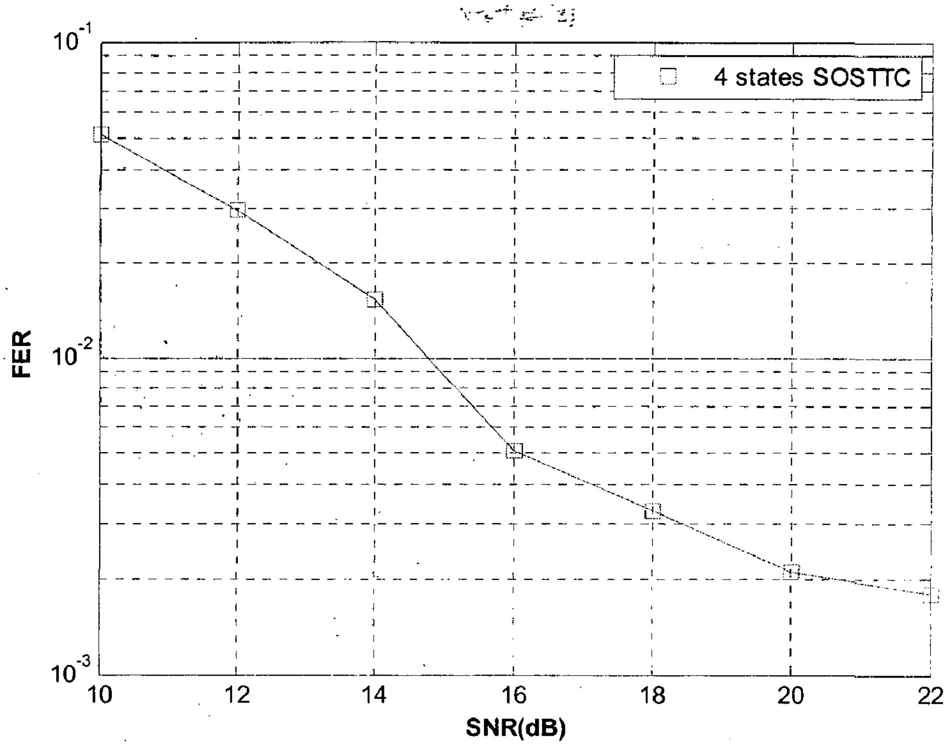

Fig. 4.6. Four states SOSTTC-BPSK with 2 transmit antenna and 1 receive antenna.

4.3.1 Four states SOSTTC using QPSK

The encoder receives one state bit (B_A) and three modulation bits (B_B) at a time T_0, and the transmission variables (X_1, X_2, Θ) equal:

(0, 0, 0) if the encoder is in state **1**, state bit (B_A) equals 0, and the modulation bits (B_B) equal 000.

(2, 2, 0) if the encoder is in state **1**, state bit (B_A) equals 0, and the modulation bits (B_B) equal 001.

(0, 2, 0) if the encoder is in state **1**, state bit (B_A) equals 0, and the modulation bits (B_B) equal 010.

(2, 0, 0) if the encoder is in state **1**, state bit (B_A) equals 0, and the modulation bits (B_B) equal 011.

(1, 1, 0) if the encoder is in state **1**, state bit (B_A) equals 0, and the modulation bits (B_B) equal 100.

(3, 3, 0) if the encoder is in state **1**, state bit (B_A) equals 0, and the modulation bits (B_B) equal 101.

(1, 3, 0) if the encoder is in state **1**, state bit (B_A) equals 0, and the modulation bits (B_B) equal 110.

(3, 1, 0) if the encoder is in state **1**, state bit (B_A) equals 0, and the modulation bits (B_B) equal 111.

(0, 1, π) if the encoder is in state **1**, state bit (B_A) equals 1, and the modulation bits (B_B) equal 000.

(2, 3, π) if the encoder is in state **1**, state bit (B_A) equals 1, and the modulation bits (B_B) equal 001.

(0, 3, π) if the encoder is in state **1**, state bit (B_A) equals 1, and the modulation bits (B_B) equal 010.

(2, 1, π) if the encoder is in state **1**, state bit (B_A) equals 1, and the modulation bits (B_B) equal 011.

Similarly, the transmission variables are designed for state **2**, **3** and **4** as result (2). Then, these encoded codewords of two states SOSTTC are modulated with QPSK. These modulated codewords of SOSTTC are passed through a quasi-static flat Rayleigh fading model for the channel. There are two path gains independent complex Gaussian random variables and are fixed during the transmission of one frame. Each frame consists of 130 transmissions out of each transmit antenna.

Similarly, decoding of 4 states SOSTTC with QPSK are carried out same as result (2).The optimal design of set partitioning Figure 2.3 and using orthogonality of STBC as in modified ML decoding done .Finally, simulation results are plotted between the frame error rate (FER) of transmitted symbols and SNR (10 to 22 dB).

Simulation Parameters:

Transmit antenna (Nt) =2, Receive antenna (Nr) =1, Frame- size = 130

Total number of frames=1000

Channel = 2 independent path-gains of complex Gaussian for Rayleigh channel.

SNR= 10 dB to 22 dB

Result 7: Simulation of 4 states SOSTTC using QPSK (2 transmit antenna and 1 receive antenna):

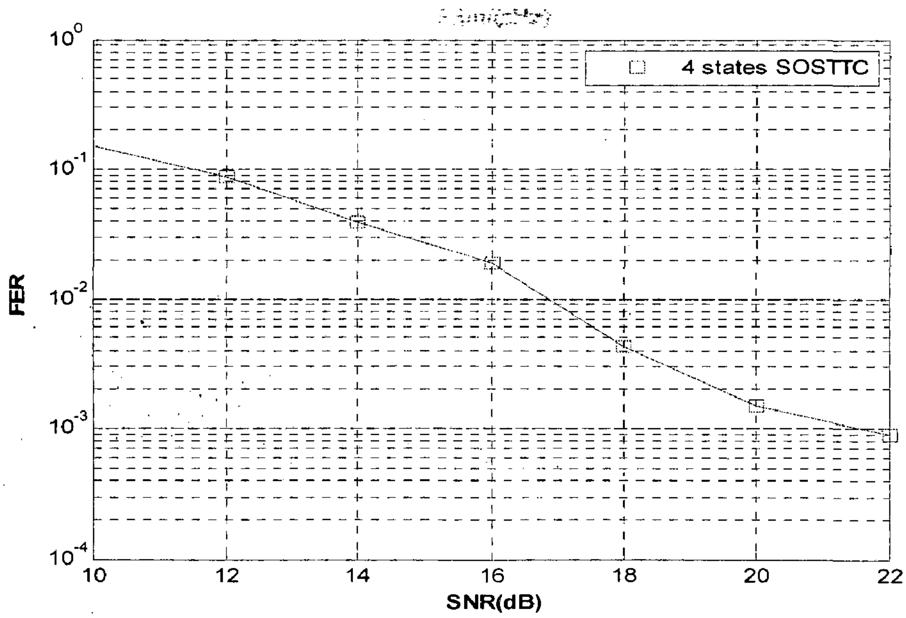

Fig. 4.7. Four states SOSTTC-QPSK with 2 transmit antenna and 1 receive antenna.

Result 8: Trade off between Coding gain and rate of codewords, 2 states and 4 states SOSTTC. (1 receive antenna)

The simulation results 6 and 7 are combined together. The more coding gain advantage 4 states SOSTTC with rate 1 bit/s/Hz with respect to SOSTTC with rate 2 bits/s/Hz.The decoding complexity is larger in the case SOSTTC-QPSK(rate 2 bits/s/Hz). As, from the set partitioning structures , more number of parallel paths in trellis transitions are used in the case of modified Viterbi decoding. Hence, there is trade off between the coding gain and rate of codewords.

Simulation Parameters:

Transmit antenna (Nt) =2

Receive antenna (Nr) =1

Frame- size = 130

Total number of frames=1000

Channel = 2 independent path-gains complex of Gaussian for Rayleigh channel.

SNR= 10 dB to 22 dB

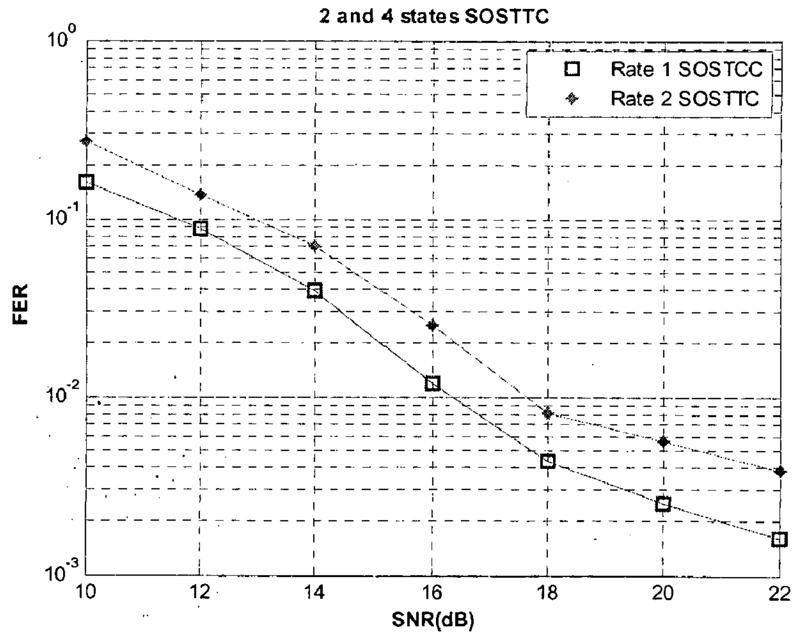

Fig. 4.8. Trade off between Coding gain and rate of codewords, 2 states and 4 states SOSTTC (1 receive antenna).

Conclusion

A new code design for super-orthogonal space-time trellis codes with full rate, full diversity gain is introduced in this dissertation. These codes guarantee full rate and full diversity gain as well as maximum coding gain for each MPSK modulation scheme. These codes are systematically design using orthogonal design of space time block code (STBC) at different rates and for different trellis structures. The orthogonal designs of STBC are used as the building blocks in SOSTTCs. The decoding complexity remains low, while full diversity is obtained. The optimal set partitioning structures are designed for MPSK, based on the maximization of coding gain distance (CGD) of SOSTTCs. There are trade off between rate and coding gain for achieving full diversity of different SOSTTCs. These codes have generally higher coding gain compare to the corresponding space-time trellis codes.

The constellation size of transmitted codewords are not expanded in the original constellation for symbols of each SOSTTC, which provides the better coding gain than STTCs. Simulation results demonstrated that the coding gain advantage is better in the case of the higher states SOSTTCs.

Further studies

Due to the limitation of design, orthogonal-STBCs matrices for complex constellation symbols are not available and full rate of transmitted symbols is not possible. Hence, the new systematic structure of super-quasi-orthogonal space-time trellis codes needs be explored.

For higher coding gain (CGD), QAM can be used in the optimal design of set-partitioning structures for wireless communication system of SOSTTCs. In addition, when a broadband wireless connection is considered, it is necessary to extend this work to frequency-selective channels.

References

[1] V. Tarokh, N. Seshadri, and A. R. Calderbank, “Space-time codes for high data rate wireless communication: Performance analysis and code construction,” *IEEE Trans. on Inform. Theory*, vol. 44, pp. 744–765, Mar. 1998.

[2] S. M .Alamouti, “A simple transmitter diversity scheme for wireless communications,” *IEEE Journal on Comm.*, vol. 16, pp. 1451–1458, Oct. 1998.

[3] S. M. Alamouti, V. Tarokh, and P. Poon, “Trellis-coded modulation and transmit diversity: design criteria and performance evaluation,” Proc. *IEEE ICUPC* , vol. 2, pp. 703–707., Oct. 1998 .

[4] G. Ungerboeck,“Channel coding for multilevel/phase signals,” *IEEE Trans. on Inform. Theory*, vol. 28, pp. 55–67, Jan. 1982.

[5] D. Divsalar, and M. K. Simon, “Multiple trellis coded modulation (MTCM),” *IEEE Trans. on Comm.*, vol. 36, pp. 410–419, Apr. 1988.

[6] V. Tarokh, H. Jafarkhani, and A. R. Calenderbank,”Space-time block codes from orthogonal designs,” *IEEE Trans. on Inform. Theory*, vol. 45, pp. 1456-1467, July 1999.

[7] S. Siwamogsatham, and M. P. Fitz,“Improved high rate space-time codes via orthogonality and set partitioning,” *IEEE Wireless Communications and Networking Conference (WCNC)*, vol. 1, pp. 264-270 ,Mar. 2002.

[8] S. Baro, G. Bauch, and A. Hansman, “Improved codes for space-time trellis-coded modulation,” *IEEE Comm. Lett.*, vol. 4, pp. 20–22, Jan. 2000.

[9] H. Jafarkhani and N. Seshadri, “Super-orthogonal space-time trellis codes,” *IEEE Trans. on Info. Theory*, vol. 49, pp. 937–950, Apr. 2003.

[10] M. Bale, B. Laska, and H. Jafarkhani, “Computer design of super-orthogonal space-time trellis codes,” *IEEE Trans. Wireless Comm.*, vol. 6, pp. 463-467, Feb. 2007.

[11] D. M. Lonescu, “On space-time code design,” *IEEE Trans. Wireless Comm.*, vol. 2, pp. 20-28, Jan. 2003.

[12] D. M.Lonescu, K.K.Mukkavilli, Z. Yan Z, and J. Lilleberg, “Improved 8- and 16-state space time codes for 4PSK with two transmit antennas,” *IEEE Comm. Lett.*, vol. 5, pp. 301-305, July 2001.

[13] M. Uysal, and C.N. Georghiades, "On the error performance analysis of space-time trellis codes," *IEEE Trans. Wireless Comm.*, vol. 3, pp. 1037-1042, Dec. 2004.

[14] S. Siwamogsatham, and M.P. Fitz, "Robust space-time coding for correlated Rayleigh fading channels," *IEEE Trans. Signal Processing,* vol. 50, pp. 2408-2416, Oct. 2002.

[15] M. Jankiraman, *Space-time codes and MIMO systems.* Artech House Pub., 2004.

[16] G.H. Foschini, and M.J.Gans, "On limits of wireless communications in a fading environment when using multiple antennas," *Wireless Per. Comm.*, vol. 3, *pp.* 311–335, Mar. 1998.

[17] S. Benedetto, M.Mondin, and G. Montros, "Performance evaluation of trellis-coded modulation schemes,"*Proceedings of the IEEE,* vol. 82, pp. 833-855, June 1994.

[18] M. Rouanne, and D.Costello, "An algorithm for computing the distance of trellis codes," *IEEE Journal on Comm.,* vol. 7, pp. 929-940, Aug. 1989.

[19] D.Aktas, and Michael P. Fitz, " Distancc spectrum analysis of space–time Trellis coded modulations in quasi-static Rayleigh fading channels," *IEEE Trans. on Inform. Theory* , vol. 49, pp. 335-344, Dec. 2003.

[20] S. Nambirajan, and J. Hamid, "Super-orthogonal space-time trellis codes and applications thereof," *United States Patent 7065148*, Jun. 2006.

[21] E. A. Fettweis, "Parallel Viterbi algorithm implementation: breaking the ACS-bottleneck," *IEEE Trans. on Comm.*, vol. 37, pp. 785-789, Aug. 1989.

[22] G. Forney, "The Viterbi algorithm," *Proceedings of the IEEE,* vol. 61, pp. 268-278, Mar. 1973.

Appendix 1.

Matlab Program

%% Function for Encoder of 2 states SOSTTC using BPSK techniques %%

```
function [codeword,rotation,current_state]=encoding21(symbol_per_frame,x1,x2)
S0=[0, 1;];
 S1=[1, 0;];
current_state=0;
 for i=1:symbol_per_frame,
 switch current_state,
  case 0,
 switch x2(i),
 case 0,
 codeword(i,1)=x1(i);
 codeword(i,2)=S0(x1(i)+1);
rotation(i)=0;
 current_state=0;
case 1,
codeword(i,1)=x1(i);
codeword(i,2)=S1(x1(i)+1);
 rotation(i)=pi;
current_state=1;
  otherwise,
 display('encoding input is error');
 end;
 case 1,
 switch x2(i),
 case 0,
  codeword(i,1)=x1(i);
 codeword(i,2)=S1(x1(i)+1);
 rotation(i)=0;
```

```
current_state=0;
 case 1,
 codeword(i,1)=x1(i);
 codeword(i,2)=S0(x1(i)+1);
 rotation(i)=pi;
 current_state=1;
 otherwise,
 display('encoding input is error');
 end;
 otherwise('encoding input is error');
  end;
  end;
```

% Function for BPSK symbols %

```
function[ symb1, symb2]=BPSKmapping21(symbol_per_frame,codeword);
for i=1:symbol_per_frame,
switch  codeword(i,1),
 case 0,
symb1(i)=-1;
 case 1,
symb1(i)=1;
otherwise,
display('mapping 1 is error');
end;
switch  codeword(i,2),
 case 0,
symb2(i)=-1;
   case 1
  symb2(i)=1;
 otherwise,
  display('mappintg 2 is error');
 end;
```

```
end;

% Function Rayleigh  Channel %
function H=rayleigh21(Nr,Nt)
 H=zeros(Nr,Nt);
R=eye(Nr*Nt);
X=randn(Nr*Nt, 1)/sqrt(2)+j*randn(Nr*Nt,1)/sqrt(2);
 H=reshape(R'*X,Nr,Nt);

% % main  program for  2 state SOSTTC  using BPSK %%
 Nt=2;
Nr=1;
min_snr=10;
max_snr=22;
interval=2;
symbol_per_frame=130;
avg_power=2.0;
times=10000;
kkk=1;
tot_err_bit=zeros(1,(max_snr-min_snr)/2+1);
 S(1,:,:)=[-1,-1;1,1;];
 S(2,:,:)=[-1,1; 1 -1;];
 mapping=[0,1;];
 for i=1:2,
 state=mapping(i);
  for k=1:2;
  x2=mapping(k);
 if [state x2]==[0 0]
  table(i, k, :)=[1, 1];
 elseif [state x2]==[0 1]
 table(i, k, :)=[2,-1];
  elseif [state x2]==[1 0]
table(i, k, :)=[2, 1];
```

```
elseif [state x2]==[1 1]
table(i, k, :)=[1, -1];
end;
 end;
end;
for snr=min_snr:interval:max_snr
display(snr);
sigma =sqrt(avg_power/(2*10^(snr/10)));
error_number=0;
for repeat=1:times
% Generate random data
x1=rand(1,symbol_per_frame)>0.5;
x2=rand(1,symbol_per_frame)>0.5;
x2(symbol_per_frame)=0;
H=rayleigh21(Nr,Nt);
[codeword,rotation]=encoding21(symbol_per_frame, x1, x2);
%BPSK mapping
 [symb1, symb2]=BPSKmapping21(symbol_per_frame, codeword);
% Space-Time transmitting
for i=1:symbol_per_frame,
p=exp(j*rotation(i));
C=[p*symb1(i) -p*conj(symb2(i)); symb2(i)  conj(symb1(i));];
 r(i,:)=H*C+sigma*(randn(Nr,Nt)+j*randn(Nr,Nt));
end;
%path matric
[state]=vtb_decoding21(symbol_per_frame,H, r,table);
%ML decoding
x=1;
for i=symbol_per_frame:-1:1,
decoding_x1(i)=state(i, x, 2);
decoding_x2(i)=state(i, x, 4);
x=state(i, x, 1);
end;
```

```
%FER( ofdm symbol error ratio ) calculation
if (sum(abs(decoding_x1-x1))+sum(abs(decoding_x2-x2)))~=0
 error_number=error_number+1;
  end;
  end; %for repeat
FER(kkk)=error_number/(times);
display(FER(kkk));
  kkk=kkk+1;
end; %for SNR_in_dB
% Plot results
semilogy(min_snr:interval:max_snr, FER,'r-');
legend('* SOSTTC 1 state');
title('\bf1 bit/sec/Hz');
ylabel('\bfFrame Error Probability');
xlabel('\bfSNR(dB)');
hold off;

%% Function of ML viterbi decoding   %%
function[state,decoding_x1]=vtb_decoding21(symbol_per_frame, H,r,table);
S(1,:,:)=[-1,-1;1,1;];
S(2,:,:)=[-1,1;1,-1;];
h=abs(H(1,1))^2+abs(H(1,2))^2;
for i=1:2,
for m=1:2,
  BX(i,m)=h-
2*real(r(1,1)*conj(H(1,1)*S(table(1,i,1),m,1)*table(1,i,2))+r(1,2)*conj(H(1,2))*
               S(table(1,i,1),m,1));
 BY(i,m)=h-2*real(r(1,1)*conj(H(1,2)*S(table(1,i,1),m,2))-r(1,2)*conj(H(1,1))*
S(table(1,i,1),m,2)* conj(table(1,i,2)));
        end
  path(i,:)=BX(i,:)+BY(i,:);
     [y , x]=min(path(i,:));
     state(1,i,:)=[1,x-1,y,i-1];
```

```
end;
 for k=2:symbol_per_frame,  % path matric calculation
 for i=1:2,
 for m=1:2,
 BX(i,m)=h-  2*real(r(k,1)*conj(H(1,1)*S(table(1,i,1),m,1)*table(1,i,2))+
       r(k,2)*conj(H(1,2))*S(table(1,i,1),m,1));        % decoding x1
BY(i,m)=h-2*real(r(k,1)*conj(H(1,2)*S(table(1,i,1),m,2))-r(k,2)*conj(H(1,1))*
S(table(1,i,1),m,2)*conj(table(1,i,2)));        % decoding x2
BX1(i,m)=h-2*real(r(k,1)*conj(H(1,1)*S(table(2,i,1),m,1)*table(2,i,2))+
         r(k,2)*conj(H(1,2))*S(table(2,i,1),m,1));        % decoding x1
 BY1(i,m)=h-2*real(r(k,1)*conj(H(1,2)*S(table(2,i,1),m,2))-
r(k,2)*conj(H(1,1))*S(table(2,i,1),m,2)*conj(table(2,i,2)));        % decoding x2
 end
 path1(i,:)=BX(i,:)+BY(i,:);
 path2(i,:)=BX1(i,:)+BY1(i,:);
 [y1,x1]=min(path1(i,:));
 [y2,x2]=min(path2(i,:));
 zz(1)=y1+state(k-1,1,3);
 zz(2)=y2+state(k-1,2,3);
 [y, x]=min(zz);
 switch x,
 case 1,
 state(k, i,:)=[1,x1-1, y,i-1];
 case 2,
state(k, i, :)=[2,x2-1, y,i-1];
 otherwise,
  display('matric is error');
  end;
  end;
 end;
```

% % Function for Encoder 2 states SOSTTC using QPSK %%

```
function [code,rota]=encoderQ2(symbol_per_frame,x11, x22, table)
current_state=0;
for i=1:symbol_per_frame,
switch current_state,
case 0,
switch x22(i),
case 0,
   if [x11(i,:)]==[0 0 0]
    code(i,1)=0;
   code(i,2)=0;
    rota(i)=0;
   current_state=0;
    elseif [x11(i,:)]==[0 0 1]
     code(i,1)=2;
      code(i,2)=2;
       rota(i)=0;
      current_state=0;
      elseif [x11(i,:)]==[0 1 0]
        code(i,1)=0;
         code(i,2)=2;
         rota(i)=0;
         current_state=0;
       elseif [x11(i,:)]==[0 1 1]
          code(i,1)=2;
           code(i,2)=0;
           rota(i)=0;
          current_state=0;
       elseif [x11(i,:)]==[1 0 0]
             code(i,1)=1;
             code(i,2)=1;
                rota(i)=0;
              current_state=0;
           elseif [x11(i,:)]==[1 0 1]
```

```
code(i,1)=3;
code(i,2)=3;
rota(i)=0;
current_state=0;
elseif [x11(i,:)]==[1 1 0]
code(i,1)=1;
code(i,2)=3;
rota(i)=0;
current_state=0;
elseif[x11(i,:)]==[1 1 1]
code(i,1)=3;
code(i,2)=1;
rota(i)=0;
current_state=0;
end;
case 1,
if [x11(i,:)]==[0 0 0]
code(i,1)=0;
code(i,2)=1;
rota(i)=pi;
current_state=1;
elseif [x11(i,:)]==[0 0 1]
code(i,1)=2;
code(i,2)=3
rota(i)=pi;
current_state=1;
elseif [x11(i,:)]==[0 1 0]
code(i,1)=0;
code(i,2)=3;
rota(i)=pi;
current_state=1;
elseif [x11(i,:)]==[0 1 1]
code(i,1)=2;
```

```
            code(i,2)=1;
        rota(i)=pi;
         current_state=1;
    elseif [x11(i,:)]==[1 0 0]
         code(i,1)=1;
         code(i,2)=0;
          rota(i)=pi;
          current_state=1;
    elseif [x11(i,:)]==[1 0 1]
           code(i,1)=3;
           code(i,2)=2;
          rota(i)=pi;
          current_state=1;
  elseif [x11(i,:)]==[1 1 0]
            code(i,1)=1;
            code(i,2)=2;
             rota(i)=pi;
             current_state=1;
  elseif [x11(i,:)]==[1 1 1]
             code(i,1)=3;
             code(i,2)=0;
              rota(i)=pi;
            current_state=1;
               end;
        otherwise,
              display('encoding input is error');
    end
case 1,
        switch x22(i),
           case 0,
              if[x11(i,:)]==[0 0 0]
             code(i,1)=0;
             code(i,2)=1;
```

```
rota(i)=0;
current_state=0;
elseif [x11(i,:)]==[0 0 1]
code(i,1)=2;
code(i,2)=3;
rota(i)=0;
current_state=0;
elseif [x11(i,:)]==[0 1 0]
code(i,1)=0;
code(i,2)=3;
rota(i)=0;
current_state=0;
elseif [x11(i,:)]==[0 1 1]
code(i,1)=2;
code(i,2)=1;
rota(i)=0;
current_state=0;
elseif [x11(i,:)]==[1 0 0]
code(i,1)=1;
code(i,2)=0;
rota(i)=0;
current_state=0;
elseif [x11(i,:)]==[1 0 1]
code(i,1)=3;
code(i,2)=2;
rota(i)=0;
current_state=0;
elseif [x11(i,:)]==[1 1 0]
code(i,1)=1;
code(i,2)=2;
rota(i)=0;
current_state=0;
elseif [x11(i,:)]==[1 1 1]
```

```
            code(i,1)=3;
            code(i,2)=0;
            rota(i)=0;
            current_state=0;
        end;
case 1,
        if [x11(i,:)]==[0 0 0]
            code(i,1)=0;
            code(i,2)=0;
            rota(i)=pi;
            current_state=1;
        elseif [x11(i,:)]==[0 0 1]
            code(i,1)=2;
            code(i,2)=2;
            rota(i)=pi;
            current_state=1;
        elseif[x11(i,:)]==[0 1 0]
            code(i,1)=0;
            code(i,2)=2;
            rota(i)=pi;
            current_state=1;
        elseif[x11(i,:)]==[0 1 1]
            code(i,1)=2;
            code(i,2)=0;
            rota(i)=pi;
            current_state=1;
        elseif [x11(i,:)]==[1 0 0]
            code(i,1)=1;
            code(i,2)=1;
            rota(i)=pi;
            current_state=1;
        elseif[x11(i,:)]==[1 0 1]
            code(i,1)=3;
```

```
                code(i,2)=3;
                rota(i)=pi;
                current_state=1;
            elseif [x11(i,:)]==[1 1 0]
             code(i,1)=1;
             code(i,2)=3;
              rota(i)=pi;
            current_state=1;
        elseif [x11(i,:)]==[1 1 1]
              code(i,1)=3;
              code(i,2)=1;
              rota(i)=pi;
             current_state=1;
          end;
        otherwise('encoding input is error');
      end;
    otherwise,
               display('encoding input is error');
            end;
        end;

%%  Function for QPSK  modulation  %%
function[ symb1, symb2]=QPSKmapping2(symbol_per_frame,code);
  for i=1:symbol_per_frame,
      switch  code(i,1),
          case 0,
            symb1(i)=1;
          case 1,
            symb1(i)=j;
           case 2,
            symb1(i)=-1;
          case 3,
            symb1(i)=-j;
```

```
            otherwise,
               display('mapping 1 is error');
            end;
     switch  code(i,2),
             case 0,
                 symb2(i)=1;
             case 1,
                 symb2(i)=j;
             case 2,
               symb2(i)=-1;
            case 3,
               symb2(i)=-j;
             otherwise,
                display('mappintg 2 is error');
             end;
end;
```

%% main program for 2 states SOSTTC using QPSK with 1 receive antenna %

```
Nt=2;
Nr=1;
min_snr=10;
max_snr=22;
interval=2;
symbol_per_frame=130;
avg_power=2.0;
times=1000;
 kkk=1;
tot_err_bit=zeros(1,(max_snr-min_snr)/2+1);
   S(1,:,:)=[1,1;-1,-1;];
   S(2,:,:)=[1,-1; -1 1;];
   R(1,:,:)=[j,j;-j,-j;];
   R(2,:,:)=[j,-j;-j,j;];
   T(1,:,:)=[1,-1;1,-1;];
```

```
T(2,:,:)=[j,-j;-j,j;];
U(1,:,:)=[j,-j;j,-j;];
U(2,:,:)=[1,-1;-1,1;];
mapping=[0,1;];
for i=1:2,
  state=mapping(i);
   for k=1:2;
    x2=mapping(k);
                if [state x2]==[0 0]
                   table(i, k, :)=[1, 1];
               elseif [state x2]==[0 1]
                    table(i, k, :)=[2,-1];
               elseif [state x2]==[1 0]
                    table(i, k, :)=[2, 1];
               elseif [state x2]==[1 1]
                    table(i, k, :)=[1, -1];
                end;
             end;
           end;
    for snr=min_snr:interval:max_snr
display(snr);
  sigma =sqrt(avg_power/(2*10^(snr/10)));
    error_number=0;
  for repeat=1:times
    % Generate random data
   x1=rand(symbol_per_frame,1)>0.5;
   x2=rand(symbol_per_frame,1)>0.5;
   x3=rand(symbol_per_frame,1)>0.5;
   x11=[x1 x2 x3];
   x22=rand(1,symbol_per_frame)>0.5;
   x22(symbol_per_frame)=0;
     H=rayleigh31(Nr,Nt);
    [code,rota]=encoderQ2(symbol_per_frame, x11, x22);
```

```
        % QPSK mapping
        [symb1, symb2]=QPSKmapping2(symbol_per_frame, code);
        % Space-Time transmitting
        for i=1:symbol_per_frame,
          p=exp(j*rota(i));
          C=[p*symb1(i) -p*conj(symb2(i)); symb2(i)  conj(symb1(i));];
            r(i,:)=H*C+sigma*(randn(Nr,Nt)+j*randn(Nr,Nt));
        end;
         %path matric
        [state]=vtb_decodingQ2(symbol_per_frame,H, r,table);

        %ML decoding
        x=1;
        for i=symbol_per_frame:-1:1,
          decoding_x1(i)=state(i, x, 2);
          decoding_x2(i)=state(i, x, 4);
          x=state(i, x, 1);
        end;
        % FER( frame error ratio ) calculation
        if (sum(abs(decoding_x1-x3'))+sum(abs(decoding_x2-x22)))~=0
             error_number=error_number+1;
          end;
      end; %for repeat
  FER(kkk)=error_number/(times);
  display(FER(kkk));
  kkk=kkk+1;
end; %for SNR_in_dB
title('space-time trellis codes');
ylabel('FER  (Frame error ratio)');
semilogy(min_snr:interval:max_snr, FER);
```

% % Function for Viterbi decoding using QPSK %%

```
function[state,decoding_x1]=vtb_decodingQ2(symbol_per_frame, H,r,table);
```

```
S(1,:,:)=[1,1;-1,-1;];
S(2,:,:)=[1,-1; -1 1;];
R(1,:,:)=[j,j;-j,-j;];
R(2,:,:)=[j,-j;-j,j;];
T(1,:,:)=[1,-1;1,-1;];
T(2,:,:)=[j,-j;-j,j;];
U(1,:,:)=[j,-j;j,-j;];
U(2,:,:)=[1,-1;-1,1;];
  h=abs(H(1,1))^2+abs(H(1,2))^2;
for i=1:2,
    for m=1:2,
BX(i,m)=h-2*real(r(1,1)*conj(H(1,1)*S(table(1,i,1),m,1)*
table(1,i,2))+r(1,2)*conj(H(1,2))*S(table(1,i,1),m, 1));
BY(i,m)=h-2*real(r(1,1)*conj(H(1,2)*S(table(1,i,1),m,2))-
r(1,2)*conj(H(1,1))*S(table(1,i,1),m,2)*conj(table(1,i,2)));
BX1(i,m)=h-2*real(r(1,1)*conj(H(1,1)*R(table(1,i,1),m,1)*table(1,i,2))+
r(1,2)*conj(H(1,2))*R(table(1,i,1),m,1));
BY1(i,m)=h-2*real(r(1,1)*conj(H(1,2)*R(table(1,i,1),m,2))-
r(1,2)*conj(H(1,1))*R(table(1,i,1),m,2)*conj(table(1,i,2)));
BX2(i,m)=h-2*real(r(1,1)*conj(H(1,1)*T(table(1,i,1),m,1)*table(1,i,2))+
r(1,2)*conj(H(1,2))*T(table(1,i,1),m,1));
BY2(i,m)=h-2*real(r(1,1)*conj(H(1,2)*T(table(1,i,1),m,2))-
r(1,2)*conj(H(1,1))*T(table(1,i,1),m,2)*conj(table(1,i,2)));
BX3(i,m)=h-2*real(r(1,1)*conj(H(1,1)*U(table(1,i,1),m,1)*table(1,i,2))+
r(1,2)*conj(H(1,2))*U(table(1,i,1),m,1));
BY3(i,m)=h-2*real(r(1,1)*conj(H(1,2)*U(table(1,i,1),m,2))-
r(1,2)*conj(H(1,1))*U(table(1,i,1),m,2)*conj(table(1,i,2)));
    end
 path(i,:)=BX(i,:)+BY(i,:);

 path1(i,:)=BX1(i,:)+BY1(i,:);
     path2(i,:)=BX2(i,:)+BY2(i,:);
 path3(i,:)=BX3(i,:)+BY3(i,:);
```

```
 Z1=min( path(i,:), path1(i,:));
 Z2=min( path2(i, :) ,path3(i,:) );
 Z3=min( Z1,Z2);
 [y  x]=min( Z3) ;
 state(1,i,:)=[1,x-1,y,i-1];
end;
 for k=2:symbol_per_frame,  % path matric calculation
        for i=1:2,
          for m=1:2,
  BX(i,m)=h-2*real(r(k,1)*conj(H(1,1)*S(table(1,i,1),m,1)*table(1,i,2))+
r(k,2)*conj(H(1,2))*S(table(1,i,1),m,1));        % decoding x1
  BY(i,m)=h-2*real(r(k,1)*conj(H(1,2)*S(table(1,i,1),m,2))-
r(k,2)*conj(H(1,1))*S(table(1,i,1),m,2)*conj(table(1,i,2)));        % decoding x2
  BX1(i,m)=h-2*real(r(k,1)*conj(H(1,1)*S(table(2,i,1),m,1)*table(2,i,2))+
r(k,2)*conj(H(1,2))*S(table(2,i,1),m,1));        % decoding x1
  BY1(i,m)=h-2*real(r(k,1)*conj(H(1,2)*S(table(2,i,1),m,2))-
r(k,2)*conj(H(1,1))*S(table(2,i,1),m,2)*conj(table(2,i,2)));        % decoding x2
   BX11(i,m)=h-2*real(r(k,1)*conj(H(1,1)*R(table(1,i,1),m,1)*table(1,i,2))+
r(k,2)*conj(H(1,2))*R(table(1,i,1),m,1));        % decoding x1
  BY11(i,m)=h-2*real(r(k,1)*conj(H(1,2)*R(table(1,i,1),m,2))-
r(k,2)*conj(H(1,1))*R(table(1,i,1),m,2)*conj(table(1,i,2)));        % decoding x2
    BX12(i,m)=h-2*real(r(k,1)*conj(H(1,1)*R(table(2,i,1),m,1)*table(2,i,2))+
r(k,2)*conj(H(1,2))*R(table(2,i,1),m,1));        % decoding x1
  BY12(i,m)=h-2*real(r(k,1)*conj(H(1,2)*R(table(2,i,1),m,2))-
r(k,2)*conj(H(1,1))*R(table(2,i,1),m,2)*conj(table(2,i,2)));        % decoding x2
   BX21(i,m)=h-2*real(r(k,1)*conj(H(1,1)*T(table(1,i,1),m,1)*table(1,i,2))+
r(k,2)*conj(H(1,2))*T(table(1,i,1),m,1));        % decoding x1
  BY21(i,m)=h-2*real(r(k,1)*conj(H(1,2)*T(table(1,i,1),m,2))-
r(k,2)*conj(H(1,1))*T(table(1,i,1),m,2)*conj(table(1,i,2)));        % decoding x2
    BX22(i,m)=h-2*real(r(k,1)*conj(H(1,1)*T(table(2,i,1),m,1)*table(2,i,2))+
r(k,2)*conj(H(1,2))*T(table(2,i,1),m,1));        % decoding x1
  BY22(i,m)=h-2*real(r(k,1)*conj(H(1,2)*T(table(2,i,1),m,2))-
r(k,2)*conj(H(1,1))*T(table(2,i,1),m,2)*conj(table(2,i,2)));        % decoding x2
```

```
    BX31(i,m)=h-2*real(r(k,1)*conj(H(1,1)*U(table(1,i,1),m,1)*table(1,i,2))+
r(k,2)*conj(H(1,2))*U(table(1,i,1),m,1));        % decoding x1
   BY31(i,m)=h-2*real(r(k,1)*conj(H(1,2)*U(table(1,i,1),m,2))-
r(k,2)*conj(H(1,1))*U(table(1,i,1),m,2)*conj(table(1,i,2)));       % decoding x2
   BX32(i,m)=h-2*real(r(k,1)*conj(H(1,1)*U(table(2,i,1),m,1)*table(2,i,2))+
r(k,2)*conj(H(1,2))*U(table(2,i,1),m,1));        % decoding x1
   BY32(i,m)=h-2*real(r(k,1)*conj(H(1,2)*U(table(2,i,1),m,2))-
r(k,2)*conj(H(1,1))*U(table(2,i,1),m,2)*conj(table(2,i,2)));       % decoding x2
          end
    path1(i,:)=BX(i,:)+BY(i,:);
    path2(i,:)=BX1(i,:)+BY1(i,:);
    path11(i,:)=BX11(i,:)+BY11(i,:);
    path12(i,:)=BX12(i,:)+BY12(i,:);
    path21(i,:)=BX21(i,:)+BY21(i,:);
    path22(i,:)=BX22(i,:)+BY22(i,:);
    path31(i,:)=BX31(i,:)+BY31(i,:);
    path32(i,:)=BX32(i,:)+BY32(i,:);
    Z1=min(path1(i,:), path11(i,:));
    Z2=min(path11(i,:) ,path12(i,:));
    Z3=min(Z1, Z2);
   Z11=min(path21(i,:) ,path22(i,:));
   Z22=min( path31(i,:) ,path32(i,:)) ;
    Z33=min(Z11, Z22);
[y1 x1]=min(Z3);
 [y2 x2]=min(Z33);
 zz(1)=y1+state(k-1,1,3);
zz(2)=y2+state(k-1,2,3);
[y, x]=min(zz);
switch x,
case 1,
 state(k, i,:)=[1,x1-1, y,i-1];
case 2,
state(k, i, :)=[2,x2-1, y,i-1];
```

```
otherwise,
display('matric is error');
 end;
end;
end;
```