

# Reduced Complexity Decoding of Space Time Trellis Codes in the Frequency Selective Channel

Jeremy Turner

A thesis submitted in partial fulfilment  
of the requirements for the degree of  
Master of Engineering  
in  
Electrical and Computer Engineering  
at the  
University of Canterbury,  
Christchurch, New Zealand.

26 May 2005



---

## ACKNOWLEDGEMENTS

Firstly, I would like to thank my supervisor Prof. Des Taylor for his valuable guidance and help. A large amount of time and effort has been put into this work both discussing ideas, and improving the quality of this thesis. I would also like to thank the Bank of New Zealand, University of Canterbury and my supervisor for their financial support for this work. Without this support it would not have been possible.

I would like to thank the other members of the Communications Research Group, especially Dave, Phillipa and Peter for their comments and discussions. Also to the lecturers and teachers I have had along the way, something has obviously stuck!

A big thank you must also go to my family. Thanks Mum and Dad for all your support over the years. Also to my brother and sister for keeping me on my toes. We have had a lot of good times over the years, and I will always be there for you all.

To all my mates, cheers for all the good times, and letting me take time to relax. To everyone from school, BJ hall, Uni and hockey, its always good to catch up. Special thanks must go to my good mates Andy and Nic for putting up with me and being there for all these years. Also to Ian and Geoff for living with me for 4+ years, that has got to be worth a medal.

Finally I would like to say a big thank you to that extra special person in my life. Kerryn you have been a great source of support, help and love over the past couple of years. Its been awesome to know there is always someone I can talk to about anything.

Jeremy



---

**CONTENTS**

<b>ACKNOWLEDGEMENTS</b>	<b>i</b>
<b>ABSTRACT</b>	<b>ix</b>
<b>CHAPTER 1 INTRODUCTION</b>	<b>1</b>
1.1 Capacity	1
1.2 Multiple Input Multiple Output Systems	2
1.2.1 Space Time Trellis Codes	3
1.2.2 Space Time Block Codes	3
1.2.3 BLAST	4
1.2.4 OFDM	4
1.3 Reduced Complexity	5
1.4 Thesis Focus	6
1.4.1 Previous Research	6
1.4.2 Proposed System Structure	6
1.4.3 Contributions of Thesis	7
<b>CHAPTER 2 SPACE TIME TRELLIS CODES</b>	<b>9</b>
2.1 Code Design	9
2.1.1 Tarokh STTCs	9
2.1.2 Baro STTCs	11
2.1.3 Chen STTCs	11
2.1.4 Encoder Structure	14
2.1.5 Trellis Example	16
2.2 Channel Models	17
2.2.1 Flat Fading System Model	17
2.2.2 Frequency Selective System Model	18
2.3 Decoding Methods	19
2.3.1 Flat Fading Channel	19
2.3.2 Frequency Selective Fading Channel	20
<b>CHAPTER 3 PARTITIONED VITERBI ALGORITHM</b>	<b>21</b>
3.1 System Model	21
3.2 Channel Estimates	22
3.3 Pre-Filter	23

3.3.1	Alternative Methods	25
3.4	Feedback Trellis Structure	26
3.5	Performance	27
3.6	Complexity	27
<b>CHAPTER 4</b>	<b>SOFT OUTPUT VITERBI ALGORITHM</b>	<b>29</b>
4.1	Original Development	29
4.2	Complexity	31
4.2.1	Performance	32
4.3	Comparison To BCJR	32
4.3.1	BCJR method	32
4.3.2	Complexity	33
<b>CHAPTER 5</b>	<b>DECODING METHOD</b>	<b>35</b>
5.1	System Overview	35
5.1.1	Assumptions	36
5.2	SOVA	37
5.3	PVA	38
5.3.1	Soft Input	38
5.3.2	Soft Output	39
5.3.2.1	Soft Position	39
5.3.3	Metric Calculation	41
5.4	Interleaver	42
5.5	STTC Designs	43
5.6	Iteration	43
5.7	Maximum Likelihood	44
5.8	Complexity	45
5.8.1	PVA	45
5.8.2	ML	45
5.8.3	Comparison	45
5.9	Simulation Environment	46
5.9.1	Transmitter Model	47
5.9.2	Channel Model	47
5.9.3	Receiver Model	47
<b>CHAPTER 6</b>	<b>RESULTS</b>	<b>49</b>
6.1	4PSK	49
6.1.1	4 state STTCs	50
6.1.2	8 state STTCs	53
6.1.3	16 state STTCs	56
6.1.4	Chen STTCs	59
6.1.5	Different MIMO systems	61
6.2	8PSK	64
6.2.1	8 state STTCs	65
6.2.2	16 state STTCs	67

6.2.3	32 state STTCs	69
6.3	Parameter Effects	71
6.3.1	Alpha	71
6.3.2	Beta	72
6.3.3	Number of Iterations	73
6.3.4	Interleaver gain	76
6.3.5	Soft Equalisation Gain	77
6.3.6	Errors Per Frame	78
<b>CHAPTER 7</b>	<b>CONCLUSION</b>	<b>81</b>
7.1	Construction Criteria	82
7.2	Iterative Procedure Performance	82
7.3	Coding Implications	83
7.4	Throughput Improvement	83
7.5	Complexity Considerations	84
7.6	Future Research Directions	85
<b>APPENDIX A</b>	<b>SYMBOLS USED</b>	<b>87</b>
<b>APPENDIX B</b>	<b>ABBREVIATIONS USED</b>	<b>89</b>



---

**LIST OF FIGURES**

1.1	Transmitter Structure	6
1.2	Receiver Structure	7
2.1	STTC 4 state encoder, $M=4$ , $N_T=2$	16
2.2	Trellis diagram for state 2 of the 4 state Chen trellis code	17
2.3	A typical MIMO system model	18
3.1	PVA block diagram	26
5.1	Transmitter Structure	36
5.2	Receiver Structure	36
5.3	PVA Flowchart	39
5.4	Equalised Symbols	40
5.5	PVA metric calculation	41
6.1	4 State Tarokh 4PSK STTC	50
6.2	4 State Baro 4PSK STTC	51
6.3	4 State Chen 4PSK STTC	52
6.4	8 State Tarokh 4PSK STTC	53
6.5	8 State Baro 4PSK STTC	54
6.6	8 State Chen 4PSK STTC	55
6.7	16 State Tarokh 4PSK STTC	56
6.8	16 State Baro 4PSK STTC	57
6.9	16 State Chen 4PSK STTC	58
6.10	Various Chen 4PSK STTCs	59
6.11	4 State Chen STTC in a $N_T = 3$ , $N_R = 2$ MIMO system	61
6.12	4 State Chen STTC in a $N_T = 2$ , $N_R = 3$ MIMO system	62
6.13	4 State Chen STTC in a $N_T = 3$ , $N_R = 3$ MIMO system	63
6.14	8 State Tarokh 8PSK STTC	65

6.15	8 State Chen 8PSK STTC	66
6.16	16 State Tarokh 8PSK STTC	67
6.17	16 State Chen 8PSK STTC	68
6.18	32 State Tarokh 8PSK STTC	69
6.19	32 State Chen 8PSK STTC	70
6.20	Effects of alpha parameter for 4PSK	71
6.21	Effects of beta parameter for 4PSK	72
6.22	Effects of iteration on performance	73
6.23	Effects of iteration on soft information	74
6.24	Effects of interleaver on soft information	76
6.25	Effects of soft equalisation on performance	77
6.26	Number of symbol errors per frame	78

---

## ABSTRACT

In this work a new iterative approach has been suggested for decoding Space Time Trellis Codes (STTCs) in the frequency selective Multiple Input Multiple Output (MIMO) channel. The objective of this thesis has been to investigate the performance of the approach and determine what parameters affect its performance.

The proposed method uses the Partitioned Viterbi Algorithm (PVA) as an equaliser for the MIMO system. The equaliser provides soft outputs which are then used by a STTC decoder to estimate the transmitted data sequence. To reduce error propagation between the two Viterbi based algorithms, an interleaver is introduced. To further increase the performance, an iterative approach is used, where the decoded data sequence is re-encoded and used by the PVA to improve interference cancellation. This is similar in concept to turbo equalisation. Both the PVA equaliser and STTC decoder have been adapted to provide soft outputs using a Soft Output Viterbi Algorithm (SOVA).

Simulations of a  $N_T = 2$  transmit antenna,  $N_R = 2$  receive antenna MIMO system have been performed for both 4PSK and 8PSK constellations. It is shown that the iterative procedure achieves a performance within 2.0dB of Maximum Likelihood (ML) decoding, at a FER of  $10^{-2}$ . However, the iterative approach suffers a small diversity loss. It is also shown that the complexity of the iterative approach is far lower than ML decoding. For example, a 16 state 4PSK STTC can be decoded using the iterative approach, with equal performance and less complexity, than a 4 state 4PSK with ML decoding.

It is also shown that for a large diversity system ( $rN_r > 3$ ), where  $r$  is the rank of the STTC, that codes designed using the trace, or Euclidean distance criteria, suggested by Chen *et.al.* [10] are superior to the rank and determinant criteria used by Tarokh *et.al.* [59] and Baro *et.al.* [6]. Other factors investigated that affect the performance are the choice of interleaver, number of iterations, soft information and the size of the MIMO system.

The use of a PVA or similar equaliser, coupled with an outer code could be used to increase the effective user bandwidth of the MIMO channel. Standard convolutional codes could be used with the equaliser to improve performance in an iterative approach. More research is required to investigate the performance and complexity of such an approach.



# Chapter 1

---

## INTRODUCTION

Modern society has recognised the importance of having a strong and capable communications network as one of its foundations. Such a network is very important in the role of spreading information, ideas and content, to a wide audience. Currently, the demand for services is growing at a rapid rate [19].

The growing demand for communications services has caused major investment in this area. Over the past decade, major advances have been made in wireless and wired communications networks, for example, with the implementation of 2G and 3G cellular services [33], [42], xDSL networks [56] and the development of IPv6 [34], [31] and other internet related technologies.

Of particular interest is the demand for capacity in wireless technologies [51]. Many telecommunications companies are discovering the cost of installing fibre or other high-bandwidth technologies to be prohibitive, especially over the “last mile” [14]. To enable users to use high quality services, new technologies are being developed to allow reliable, high-bandwidth wireless communications. Such technologies are expected to provide a cheaper alternative to wired technology, albeit with possibly lower performance.

### 1.1 CAPACITY

To enable better delivery of content to users, the available capacity of wireless links must be increased. This capacity increase can be made by either increasing bandwidth, power levels or through diversity. Due to a limited and regulated spectrum, it is often very difficult or expensive to increase bandwidth or power levels. Improving diversity has been shown to be crucial in increasing capacity and improving performance in wireless communications [71]. Diversity means being able to have several versions of the same signal at the receiver for joint processing and combining, whether through multiple antennas (spatial diversity), repeated transmissions (time diversity), or transmitting at different frequencies (frequency diversity) [18]. Spatial diversity is attractive because it does not increase bandwidth or lower spectral efficiency.

Space-Time processing is used in communications systems with spatial and temporal diversity. Spatial diversity is generated by having multiple antennas, either at the transmitter, receiver or both. These systems are known as Multiple Input, Single Output (MISO), Single Input, Multiple Output (SIMO) and Multiple Input, Multiple Output (MIMO) respectively. Temporal diversity is generated by transmitting a signal or some function of a signal multiple times.

## 1.2 MULTIPLE INPUT MULTIPLE OUTPUT SYSTEMS

For the purposes of this research, a space-time processing system is defined as having the number of transmit antennas  $N_T \geq 2$  and the number of received antennas  $N_R \geq 1$ . A system with  $N_T = 1$  and  $N_R > 1$ , or a SIMO system, will only achieve receive diversity. While this can be shown to give a performance gain, as for example in RAKE receivers in Code Division Multiple Access (CDMA) systems [50] [55], there do not appear to be any novel transmit techniques that can be developed for such a system. For cellular type systems, receivers are required to be as simple as possible, and hence cheap to produce. This means that transmit diversity, with its associated complexity, can be far more easily introduced at the base station, with a relatively low cost per user. Further information on receiver diversity techniques, for example Maximum Ratio Combining (MRC), can be found in [63] and references within.

The channel is modelled by gains between each transmit and receive antenna pair. Due to operating in a wireless environment, the channel is often affected by deep fades, caused by destructive signal interference. Often it is assumed that there is independent flat fading for ease of simulation or calculation. This is not always valid for high bandwidth links, where frequency selective fading may occur, causing Inter Symbol Interference (ISI). The fading is typically modelled by a Rayleigh or Rician distribution, depending on whether there is a line of sight path. The gains are also often assumed to be uncorrelated, although work is currently being done for correlated channels, for example [21],[46].

The knowledge that utilising transmit diversity can increase system capacity and performance has been around for a number of years. Optimum receive structures for MIMO systems were developed by Van Etten [66], [67] and Kaye and George [37] in the 1970's. More recent research by Foschini [24] and Telatar [60] in the mid 1990's has shown that capacity increases proportional to  $\min(N_T, N_R)$  are possible. The diversity of the system changes the slope of the Bit Error Rate (BER) curve plotted against Signal to Noise Ratio (SNR). This has been shown to be proportional to

$$BER \propto \frac{1}{SNR^d} \quad (1.1)$$

where  $d$  is the diversity order of the system.

The first system using transmit diversity and coding techniques was published by Wittneben [72]. This sparked many research projects, and in 1998 Tarokh et.al. [58] [59] developed the first true space-time trellis code (STTC). Following this, Alamouti [4] published the first space-time block code (STBC) in 1999. At the same time, Bell Labs developed the Bell Laboratories Layered Space Time Architecture (BLAST) system. Another technique, Orthogonal Frequency Division Multiplexing [15] (OFDM) can also utilise multiple antennas [1], but does not necessarily require a MIMO system.

### 1.2.1 Space Time Trellis Codes

STTCs were developed to provide coding gain and full diversity in MIMO systems [58]. They utilise a trellis encoder, similar to that of a convolutional encoder, to introduce redundancy into the transmitted symbol stream, which is required to provide coding gain. The coding gain is dependent on the construction criteria of the code, and the length of memory in the trellis. They are reasonably complex to decode, especially in the frequency selective fading channel. STTCs are covered in detail in Chapter 2.

### 1.2.2 Space Time Block Codes

STBCs were developed very quickly after STTC's by Alamouti [4]. They are essentially a simple variant of the STTC, in a similar way that block codes, for example BCH codes, are related to convolutional (trellis) codes. The Alamouti STBC is very closely related to Maximum Ratio Combining [63], and has the same diversity gain, but with a performance (SNR) penalty under equivalent power constraints. This is because systems are typically limited by the total transmit power available, so a system with  $N_T$  transmit antennas would transmit  $\frac{E}{N_T}$  per antenna, where  $E$  is the total available transmit energy.

In general STBCs can be separated into two types, orthogonal [4], and quasi-orthogonal [57], [62]. Data is mapped to a block structure, or matrix, for transmission. The elements in the matrix are typically rotated and reflected versions of the symbol alphabet. For an orthogonal STBC, all columns in the encoding matrix are orthogonal to each other. This means very simple decoding is required, as each symbol can be decoded separately. It has been shown that for a flat fading channel with perfect channel state information (CSI), maximum likelihood (ML) decoding is possible with only linear complexity [39]. Quasi-orthogonal STBCs divide the block structure into pairs of columns, and require that each pair of columns is orthogonal. The decoding is then done on a pair-wise basis, and is slightly more complex [35]. Non-orthogonal codes are possible, but are more complex to decode.

Orthogonal codes typically have a lower "rate" than quasi-orthogonal codes, where the rate is defined as the number of data symbols per frame, over the length of the

frame. On the other hand, quasi-orthogonal codes suffer a diversity loss compared to full orthogonal codes, which results in lower performance at high SNR [35].

### 1.2.3 BLAST

The original BLAST structure was developed by Foschini, at Bell labs [24]. It uses a multi-element antenna array at both the transmitter and receiver, and advanced signal processing at the receiver to estimate and decode code blocks. These code blocks are encoded at the transmitter with a diagonal structure, hence the system is known as D-BLAST. It is relatively complex to implement, and a simpler method which encodes data vertically has also been developed, and is known as V-BLAST [25], [54], [73]. The benefit of using a BLAST approach is the high spectral efficiency caused by having  $N_T$  independent signals.

In a BLAST system, a rich scattering environment is required, which often occurs indoors, with  $N_R \geq N_T$ . This allows the receiver array to successfully decouple the various transmitted data streams (nulling), and use successive interference cancellation for equalisation and detection. In D-BLAST each layer experiences the same spatial channels, but some zero blocks have to be padded to maintain diversity. Zero padding is not required in V-BLAST, but each layer experiences a separate channel, and hence diversity may be obtained. This requires the decoding to be done in a specific order, namely from the layer with the highest SNR to the lowest. A V-BLAST prototype was actually developed in [73] and shown to achieve a spectral efficiency of  $> 20$  bps/Hz.

The benefit of using BLAST over a STTC structure is that the decoding complexity is linear with increasing  $N_T$ . However ML decoding techniques are currently infeasible for decoding due to the computational requirements. Hence, a diversity loss is suffered in the case of V-BLAST, or a rate loss in the case of D-BLAST, over most channels. As the BLAST structure uses feedback type schemes, it suffers from error propagation. It also requires Channel State Information (CSI) at the receive antenna.

### 1.2.4 OFDM

OFDM systems have been primarily designed to deal with a frequency selective channel, i.e. one that has a long delay spread. It works by splitting a high-rate input data stream into a large number of low-rate data streams [15]. The low-rate streams are multiplexed onto subcarriers using the Fast Fourier Transform (FFT). Each of the subcarriers is treated as a flat fading channel, which allows for simpler per-channel equalisation at the receiver. This is because the bandwidth of each carrier is less than the coherence bandwidth of the channel. Because OFDM operates as a block scheme, a guard band is required to ensure there is no inter block interference. This is achieved using a Cyclic Prefix (CP), which also ensures the channel matrix is diagonalisable by the FFT.

OFDM does not necessarily have to be used in a MIMO situation, and was originally developed for the SISO case. However it easily extends into a MIMO environment [40], and work is currently being done on combining both STBCs [13] and STTCs [1], [64] with OFDM systems. Due to each individual sub channel experiencing flat fading, orthogonal STBCs have linear decoding complexity. This makes a combined STBC-OFDM system quite easy to implement. Advances are also being made in modifying STTCs into space-frequency trellis codes (SFTC) [32], [53] suitable for OFDM systems.

OFDM is being used in commercial equipment today, following for example the IEEE 802.11a and 802.11g standards. These standards are suited to short range (50-100m) stationary equipment, for example laptops, and offer high (> 50Mbps) data rates. However, OFDM suffers from a large peak-to-average power ratio [36], [47], which requires expensive linear amplifiers. It also has a very low tolerance of time varying channels, which introduce Inter Carrier Interference (ICI). Care has to be taken to ensure synchronisation and accurate CSI are available at the receiver.

### 1.3 REDUCED COMPLEXITY

Although it has been shown that capacity increases in MIMO channels proportional to  $\min(N_T, N_R)$  are theoretically possible [60], it is very difficult to develop a system that achieves these gains. Maximum Likelihood (ML) decoding of an uncoded system in a flat fading additive white Gaussian noise (AWGN) channel requires a Viterbi type system with  $M^{N_T}$  states, where  $M$  is the constellation size. This is exponentially complex in the number of transmit antennas. The complexity is far greater in a frequency selective, or ISI channel with impulse response length  $L$ . In this case an uncoded system has  $M^{N_T(L-1)}$  states, which becomes prohibitively complex very rapidly, even for moderate values of  $M, N_T, L$ . This provides a challenge to researchers and developers to create a system with good performance and that can also be implemented in practice.

As mentioned above, there has been a variety of approaches to reducing the complexity of MIMO systems. BLAST type approaches simplify the decoding by estimating one transmit stream at a time. This allows linear complexity with increasing  $N_T$ , but suffers from a diversity loss, as the resulting system does not offer ML decoding. OFDM schemes subdivide an ISI channel into flat fading channels which allows for simple equalisation. However they are often not suitable for high mobility channels due to strict timing constraints. STBC and STTC systems transmit correlated data streams, which require fewer states at the receiver to decode and equalise. However ML decoding is still complex in the ISI channel. More work is required to come up with low complexity methods to decode and equalise such systems.

Some research has been done looking at multi-user detection in the MIMO environment [61]. Following this, Miller *et.al.* [43] developed an equalisation method that requires linear complexity with increasing  $N_T$ , i.e. complexity proportional to  $N_T M^{L-1}$ .

It is also possible to use special pre-filtering techniques to reduce the effective channel length ( $L$ ), as in [44], [29]. These results can be extended into a single-user multiple transmit system with independent transmit streams if an interleaver is used. Such an approach offers reasonable performance with reduced complexity for moderately sized systems.

## 1.4 THESIS FOCUS

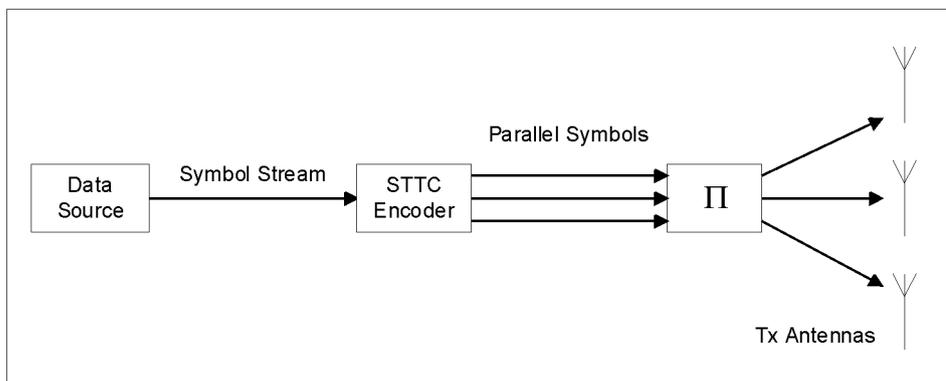
### 1.4.1 Previous Research

Most current research into Space Time Trellis Codes [6] [74] [58] [10] has been undertaken with the assumption of a flat fading channel, due to the added complexity of developing and analysing the required decoding algorithms in a frequency selective channel [53]. However, a limited number of construction criteria and codes exist that are specifically designed for the frequency selective channel, for example [41].

This thesis focuses on the combination of three separate techniques, namely Space Time Trellis Codes (STTC), the Partitioned Viterbi Algorithm (PVA) [43], and the Soft Output Viterbi Algorithm (SOVA) [30]. These techniques can be combined to provide reduced complexity decoding of Space Time Trellis Codes in a frequency selective channel. This is of interest, as the flat fading assumption has been shown to be poor in a number of multi-path environments [45]. Other approaches, such as using MAP equalisation [7] [26], have also been proposed.

### 1.4.2 Proposed System Structure

In this system a STTC is used to add redundancy and correlation between the transmitted data streams, as shown in Fig. 1.1. This reduces the number of states required at the receiver for decoding, and also allows for a coding gain. An interleaver has been added to reduce error propagation in the iterative procedure.



**Figure 1.1** Transmitter Structure

The PVA is used as an equaliser for the ISI channel to obtain reduced complexity, but does not offer ML decoding. Optimal equalisation would be accomplished using Maximum Likelihood Sequence Estimation (MLS), but in most instances it is too complex to be practical. The PVA equaliser and STTC decoder share symbol and soft information as shown in Fig. 1.2. The SOVA is used in an iterative process to generate reliability information from the PVA and STTC decoder, as these are both based on the Viterbi Algorithm.

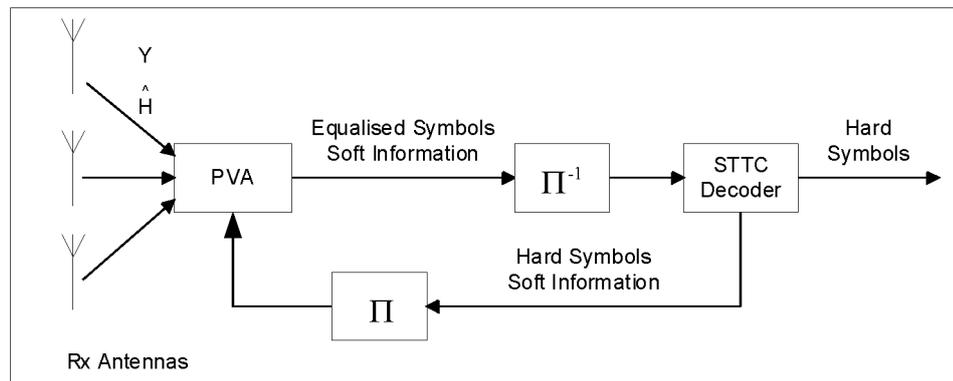


Figure 1.2 Receiver Structure

### 1.4.3 Contributions of Thesis

The main contribution of the work of this thesis has been to develop a novel equalisation and decoding process to decode STTCs in the MIMO frequency selective channel. This process has been developed by extending research from a multi-user detection scenario to a coded space-time scenario, and combining it with soft information to do iterative processing. The goal of the research has been to develop a reduced complexity approach to the problem, with a good BER performance compared to the optimal solution.

To meet this goal the following objectives are discussed in the research. The first objective is to compare the performance and complexity of the proposed iterative approach with ML decoding. The second objective is to compare the Space-Time code construction criteria in the frequency selective channel. Finally, some of the parameters and implementation of the iterative approach are looked at in detail to identify what is important for good performance.

Chapters 1 through 4 of this thesis contain introductory material to the topic and the three techniques used. Chapter 5 describes the novel process which has been developed, and demonstrates how the STTC, PVA and SOVA algorithms have been combined together. The results in Chapter 6 show that this new process allows effective decoding in a frequency selective MIMO environment. The conclusions are presented in Chapter 7, along with ideas for future work.



## Chapter 2

---

### SPACE TIME TRELLIS CODES

Space Time Trellis Codes (STTC) appear to have a great potential to increase user bandwidth efficiency in wireless communications. They can operate in both flat and frequency selective channels, but are more complicated to decode in the frequency selective case. STTC's also offer full diversity and rate as well as coding gain when compared to STBCs.

Over the past decade, there have been a large number of developments in developing design rules for STTCs and investigating limits on their performance. Tarokh *et.al.* [59] originally came up with a construction for trellis codes based on the determinant and rank criteria. Baro *et.al.* [6] then performed a computer search to find the best codes under these criteria. In [10] Chen *et.al.* derived new design rules based on the trace criteria, which is comparable to maximising the Euclidean distance of the STTC. Further work by Yan *et.al.* [74] and [52] for the frequency selective channel have also improved the performance of these codes.

In this thesis, the major focus is on methods to decode STTCs in the frequency selective channel. The majority of current work has been done on developing codes and algorithms using a flat fading channel model due to its simplicity. In this chapter both models are shown, with examples of STTCs developed for each. The design criteria and their performance are covered in the following sections.

#### 2.1 CODE DESIGN

##### 2.1.1 Tarokh STTCs

Tarokh *et.al.* originally developed design criteria for constructing space time codes [59]. These criteria are known as the rank and determinant criteria for a flat Rayleigh fading channel, and the rank and coding advantage criteria for a flat Ricean fading channel. Their argument is summarised in the following.

Given a transmitted symbol sequence

$$\mathbf{c} = c_1^1, c_1^2, \dots, c_1^{N_T}, c_2^1, c_2^2, \dots, c_2^{N_T}, \dots, c_l^1, c_l^2, \dots, c_l^{N_T} \quad (2.1)$$

and a decoded sequence of

$$\mathbf{e} = e_1^1, e_1^2, \dots, e_1^{N_T}, e_2^1, e_2^2, \dots, e_2^{N_T}, \dots, e_l^1, e_l^2, \dots, e_l^{N_T} \quad (2.2)$$

Let  $r$  ( $r \leq N_T$ ) denote the rank of the  $N_T \times l$  difference matrix  $\mathbf{B}(\mathbf{c}, \mathbf{e})$  defined as [10]

$$\mathbf{B}(\mathbf{c}, \mathbf{e}) = \begin{pmatrix} e_1^1 - c_1^1 & \cdots & e_l^1 - c_l^1 \\ \vdots & \ddots & \vdots \\ e_1^{N_T} - c_1^{N_T} & \cdots & e_l^{N_T} - c_l^{N_T} \end{pmatrix}, \quad (2.3)$$

and let  $\{\lambda_i\}$  be the eigenvalues of the distance matrix

$$\mathbf{A}(\mathbf{c}, \mathbf{e}) = \mathbf{B}(\mathbf{c}, \mathbf{e})\mathbf{B}^*(\mathbf{c}, \mathbf{e}), \quad (2.4)$$

where  $\mathbf{B}^*(\mathbf{c}, \mathbf{e})$  is the complex transpose of  $\mathbf{B}(\mathbf{c}, \mathbf{e})$ .

Assuming ideal CSI, the probability of decoding sequence  $\mathbf{e}$ , given  $\mathbf{c}$  was transmitted can be bounded by [59]

$$P(\mathbf{c} \rightarrow \mathbf{e} \mid \mathbf{H}) \leq \frac{1}{2} \exp(-d^2(\mathbf{c}, \mathbf{e})E_s/4N_0) \quad (2.5)$$

where

$$d^2(\mathbf{c}, \mathbf{e}) = \sum_{i=1}^{N_R} \sum_{k=1}^l \left| \sum_{j=1}^{N_T} h^{i,j} (c_k^j - e_k^j) \right|^2 \quad (2.6)$$

and  $E_s$  is the average energy per symbol at each transmit antenna. The conditional probability of 2.5 can then be expressed as

$$P(\mathbf{c} \rightarrow \mathbf{e} \mid \mathbf{H}) \leq \frac{1}{2} \exp \left( - \sum_{i=1}^{N_R} \sum_{j=1}^{N_T} \lambda_j |\beta_{i,j}|^2 \frac{E_s}{4N_0} \right) \quad (2.7)$$

where  $|\beta_{i,j}|$  follows a Rayleigh distribution with a pdf of

$$p(|\beta_{i,j}|) = 2|\beta_{i,j}| \exp(-|\beta_{i,j}|^2) \quad (2.8)$$

Assuming a high SNR and by taking the average of Eq. 2.7, the probability of error can be expressed as

$$\begin{aligned}
P(\mathbf{c} \rightarrow \mathbf{e}) &\leq \left( \frac{1}{\prod_{j=1}^{N_T} \left( 1 + \lambda_j \frac{E_s}{4N_0} \right)} \right)^{N_R} \\
&\leq \left( \prod_{j=1}^r \lambda_j \right)^{-N_R} \left( \frac{E_s}{4N_0} \right)^{-rN_R}
\end{aligned} \tag{2.9}$$

To optimise the performance of a given system both  $rN_R$  and  $\prod_{j=1}^r \lambda_j$  must be maximised.  $rN_R$  is known as the diversity gain, and is found from the minimum rank of matrix  $\mathbf{A}(\mathbf{c}, \mathbf{e})$  over all codeword pairs  $(\mathbf{c}, \mathbf{e})$ . The minimum rank is known as the rank of the code.

The coding gain is specified by  $\sqrt[r]{\lambda_1 \lambda_2 \dots \lambda_r}$ . Maximising this is equivalent to maximising the minimum determinant of matrix  $\mathbf{A}(\mathbf{c}, \mathbf{e})$  over all codeword pairs  $(\mathbf{c}, \mathbf{e})$ . This is known as the determinant criterion.

In [58] Tarokh *et.al.* developed some codes by hand, and showed some performance curves in the flat Rayleigh fading channel. The results showed that as the number of states available for the trellis increased, the coding gain of the system increased. This is to be expected, as in general, longer trellis codes have a larger determinant. It also showed that the different constraint length codes have the same diversity for a given number of receive and transmit antennas.

### 2.1.2 Baro STTCs

The codes developed by Tarokh were quickly improved by Baro *et.al.* in [6]. These new codes were found using the same rank and determinant criteria as Tarokh, but were found by doing a computer search of all possible codes. The code with maximum rank (2 for  $N_T = 2$  transmit antennas) and highest determinant was chosen for each trellis length.

### 2.1.3 Chen STTCs

In [10] Chen *et.al.* developed new criteria for designing STTCs. This criteria is based on the assumption of a large diversity order of the system. The diversity order is defined as  $rN_R$ , where  $r$  is the rank of the system, and depends on  $N_T$  and the length of the trellis  $v$ . Full rank is not always achievable due to the restrictions imposed by the trellis. The maximum rank of a STTC can be found as  $\min(N_T, \lfloor \frac{v}{2} \rfloor + 1)$  [10]. Chen *et.al.* found that if the diversity order was  $> 3$ , the trace criteria, which is equivalent to Euclidean distance, could be used. Their argument is summarised as follows.

Eq. 2.6 can be expressed as

$$\begin{aligned}
d^2(\mathbf{c}, \mathbf{e}) &= \text{tr}(\mathbf{HB}(\mathbf{c}, \mathbf{e})(\mathbf{HB}(\mathbf{c}, \mathbf{e}))^*) \\
&= \text{tr}(\mathbf{B}^*(\mathbf{c}, \mathbf{e})\mathbf{H}^*\mathbf{HB}(\mathbf{c}, \mathbf{e})) \\
&= \text{tr}\left(N_R\mathbf{B}^*(\mathbf{c}, \mathbf{e})\frac{\mathbf{H}^*\mathbf{H}}{N_R}\mathbf{B}(\mathbf{c}, \mathbf{e})\right)
\end{aligned} \tag{2.10}$$

where  $\text{tr}(\cdot)$  denotes the trace of the matrix, and  $*$  denotes the conjugate transpose.

Let

$$\mathbf{M} = \frac{1}{N_R}\mathbf{H}^*\mathbf{H} \tag{2.11}$$

with elements

$$m_{i,j} = \frac{1}{N_R}\sum_{k=1}^{N_R}\bar{h}_{k,i}h_{k,j} \tag{2.12}$$

where  $\bar{h}_{k,i}$  denotes the complex conjugate of  $h_{k,i}$ . We have

$$E\{m_{i,j}\} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \tag{2.13}$$

and

$$\lim_{N_R \rightarrow \infty} \mathbf{M} = \mathbf{I} \tag{2.14}$$

where  $\mathbf{I}$  denotes the  $N_T \times N_T$  identity matrix. This gives

$$\begin{aligned}
\lim_{N_R \rightarrow \infty} d^2(\mathbf{c}, \mathbf{e}) &= \text{tr}(N_R\mathbf{B}^*(\mathbf{c}, \mathbf{e})\mathbf{B}(\mathbf{c}, \mathbf{e})) \\
&= N_R\sum_{j=1}^{N_T}\lambda_j
\end{aligned} \tag{2.15}$$

where the channel coefficients have disappeared. From Eq. 2.5

$$\begin{aligned}
\lim_{N_R \rightarrow \infty} P(\mathbf{c} \rightarrow \mathbf{e}) &= \lim_{N_R \rightarrow \infty} P(\mathbf{c} \rightarrow \mathbf{e}|\mathbf{H}) \\
&\leq \frac{1}{2}\exp\left(-N_R\frac{E_s}{4N_0}\sum_{j=1}^{N_T}\lambda_j\right)
\end{aligned} \tag{2.16}$$

This shows that as  $N_R \rightarrow \infty$  the system performance does not depend on the type of fading (i.e. no dependence on  $\mathbf{H}$ ). Instead, the probability of error is minimised as the sum of the eigenvalues of matrix  $\mathbf{A}(\mathbf{c}, \mathbf{e})$  is maximised. This proof can be extended to "small" values of  $N_R$  as follows.

$|\beta_{i,j}|^2$  follows the central Chi-square distribution [50] with a mean value and vari-

ance given by

$$\mu_{|\beta_{i,j}|^2} = 1 \quad (2.17)$$

and

$$\sigma_{|\beta_{i,j}|^2}^2 = 1 \quad (2.18)$$

For large  $rN_R (>3)$ , according to the central limit theorem

$$\sum_{i=1}^{N_R} \sum_{j=1}^{N_T} \lambda_j |\beta_{i,j}|^2 \quad (2.19)$$

rapidly approaches a Gaussian random variable  $D$  with a mean value

$$\mu_D = N_R \sum_{j=1}^{N_T} \lambda_j \quad (2.20)$$

and a variance of

$$\sigma_D^2 = N_R \sum_{j=1}^{N_T} \lambda_j^2 \quad (2.21)$$

Therefore the unconditional pairwise error probability can be upper-bounded by

$$P(\mathbf{c} \rightarrow \mathbf{e}) \leq \int_{D=0}^{+\infty} \frac{1}{2} \exp\left(-\frac{E_s}{4N_0} D\right) p(D) dD \quad (2.22)$$

By substituting for the Gaussian random variable  $D$ , and integrating we get

$$P(\mathbf{c} \rightarrow \mathbf{e}) \leq \frac{1}{2} \exp\left(\frac{1}{2} \left(\frac{E_s}{4N_0}\right)^2 \sigma_D^2 - \frac{E_s}{4N_0} \mu_D\right) Q\left(\frac{\frac{E_s}{4N_0} \sigma_D^2 - \mu_D}{\sigma_D}\right) \quad (2.23)$$

By using the inequality

$$Q(x) \leq \frac{1}{2} e^{-x^2/2}, \quad x \geq 0, \quad (2.24)$$

a simpler bound can be found as

$$P(\mathbf{c} \rightarrow \mathbf{e}) \leq \frac{1}{4} \exp\left(-N_R \frac{E_s}{4N_0} \sum_{j=1}^{N_T} \lambda_j\right). \quad (2.25)$$

This bound is shown in [10]. To minimise the error probability for a given system, the sum of the eigenvalues in Eq. 2.25 must be maximised. These eigenvalues come from the distance matrix  $\mathbf{A}(\mathbf{c}, \mathbf{e})$ . As this matrix is square, the sum of the eigenvalues is equal to the trace of  $\mathbf{A}(\mathbf{c}, \mathbf{e})$ , denoted as  $\text{tr}(v)$ . This can be written as

$$\text{tr}(v) = \sum_{j=1}^{N_T} \lambda_j = \sum_{j=1}^{N_T} A_{j,j}. \quad (2.26)$$

$A_{j,j}$  are the elements on the main diagonal of matrix  $\mathbf{A}(\mathbf{c}, \mathbf{e})$ . The trace of this matrix can be expressed as

$$\text{tr}(v) = \sum_{j=1}^{N_T} \sum_{k=1}^l |e_k^j - c_k^j|^2. \quad (2.27)$$

Eq. 2.27 shows that the trace is equivalent to the Euclidean distance between the codewords  $\mathbf{c}$  and  $\mathbf{e}$  over all transmit antennas. Chen *et.al.* showed that maximising the minimum trace, or Euclidean distance of the code, determines the performance of the system for a large diversity order. They used this design rule and a computer search to generate some new trellis codes, which have good performance when compared to the Tarokh and Baro codes over the flat fading Rayleigh channel [10]. They also show that even for small diversity order ( $rN_R \leq 3$ ), their STTCs have very similar performance to the Tarokh and Baro codes. These results are reasonable given that a flat fading channel with a large diversity order converges to a AWGN channel, where the Euclidean distance of the code determines its performance.

A further interesting result shown in [12] and [11] is that STTCs do not need to be full rank if  $N_T \geq 3$ . They found that full rank codes ( $r = N_T$ ) have a smaller minimum trace than non-full rank codes. By using a non-full rank trellis code, performance could be improved, which appears contrary to Tarokh's original criteria.

### 2.1.4 Encoder Structure

After a trellis code has been created, it is relatively easy to develop an encoder. The encoder is closely related to that of a standard convolutional code encoder, with memory, tap coefficients and modulo addition. The major difference is that there is a branch for every transmit antenna. Each branch has memory  $v_j$ , where  $j$  is the  $j^{\text{th}}$  branch. This gives the overall memory as

$$v = \sum_{j=1}^{N_T} v_j \quad (2.28)$$

where each memory order can be determined by

$$v_j = \lfloor \frac{v + j - 1}{N_T} \rfloor, \quad j = 1, \dots, N_T \quad (2.29)$$

where  $\lfloor x \rfloor$  is the largest integer smaller than  $x$ .

At time  $k$ , binary inputs  $\{I_k^j\}$   $j = 1, \dots, N_T$  are fed into each branch, with  $I_k^1$  the most significant bit. These binary inputs are delayed and multiplied by the tap coefficients along each branch. Let these coefficients be denoted as  $(g_{p_l}^{l,j})$ , where  $l = 1, \dots, N_T$  represents the current branch, and  $p_l = 1, \dots, v_l$  represents the memory length of the current branch. These coefficients are chosen based on a generator matrix, with entries chosen from the size of the constellation ( $M$ ). For example, in 4-PSK

$(g_{p_l}^{l,j}) \in \{0, 1, 2, 3\}$  etc.

The encoder output symbols can be computed as

$$x_k^j = \sum_{l=1}^{N_T} \sum_{p_l=0}^{v_l} I_{k-p_l}^l \cdot g_{p_l}^{l,j} \pmod{M}, \quad j = 1, \dots, N_T \quad (2.30)$$

For example in a 4-PSK  $N_T = 2$  system, with  $v = 2 \therefore v_1 = 1, v_2 = 1$

$$x_k^j = \sum_{p_1=0}^{v_1} I_{k-p_1}^1 \cdot g_{p_1}^{1,j} + \sum_{p_2=0}^{v_2} I_{k-p_2}^2 \cdot g_{p_2}^{2,j} \pmod{4}, \quad j = 1, 2 \quad (2.31)$$

The trellis code is modelled as a series of coefficients. Tables 2.1 and 2.2 show various coefficients for STTCs using 2 transmit antennas and 4PSK or 8PSK. In these tables  $g_p^{1,j} = a_p^j$ ,  $g_p^{2,j} = b_p^j$ ,  $g_p^{3,j} = c_p^j$  etc. The tables are based on Tarokh (TSC) [59], Baro (BBH) [6] and Chen (CYV) [10] STTCs. The tables also show the minimum determinant and trace for each code. Performance simulations in [10] show that a large trace is more important than a large determinant.

**Table 2.1** 4-PSK Space Time Trellis Codes with  $N_T = 2$  [10]

code	v	$(a_0^1, a_0^2)$	$(a_1^1, a_1^2)$	$(a_2^1, a_2^2)$	$(a_3^1, a_3^2)$	$(b_0^1, b_0^2)$	$(b_1^1, b_1^2)$	$(b_2^1, b_2^2)$	$(b_3^1, b_3^2)$	$\det(v)$	$\text{tr}(v)$
TSC	2	(0,2)	(2,0)	-	-	(0,1)	(1,0)	-	-	4.0	4.0
BBH	2	(2,2)	(1,0)	-	-	(0,2)	(3,1)	-	-	8.0	6.0
CYV	2	(0,2)	(1,2)	-	-	(2,3)	(2,0)	-	-	4.0	10.0
TSC	3	(0,2)	(2,0)	-	-	(0,1)	(1,0)	(2,2)	-	12.0	8.0
BBH	3	(2,2)	(2,0)	-	-	(0,1)	(1,0)	(2,2)	-	12.0	8.0
CYV	3	(2,2)	(2,1)	-	-	(2,0)	(1,2)	(0,2)	-	8.0	12.0
TSC	4	(0,2)	(2,0)	(0,2)	-	(0,1)	(1,2)	(2,0)	-	12.0	8.0
BBH	4	(0,2)	(2,0)	(0,2)	-	(2,1)	(1,2)	(2,0)	-	20.0	12.0
CYV	4	(1,2)	(1,3)	(3,2)	-	(2,0)	(2,2)	(2,0)	-	8.0	16.0
TSC	5	(0,2)	(2,2)	(3,3)	-	(0,1)	(1,1)	(2,0)	(2,2)	12.0	12.0
CYV	5	(0,2)	(2,3)	(1,2)	-	(2,2)	(1,2)	(2,3)	(2,0)	20.0	16.0
CYV	6	(0,2)	(3,1)	(3,3)	(3,2)	(2,2)	(2,2)	(0,0)	(2,0)	16.0	18.0

**Table 2.2** 8-PSK Space Time Trellis Codes with  $N_T = 2$  [10]

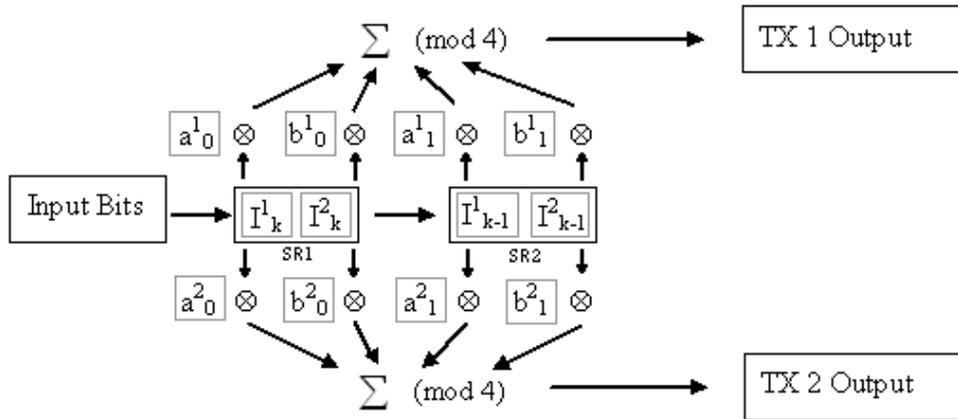
code	v	$(a_0^1, a_0^2)$	$(a_1^1, a_1^2)$	$(b_0^1, b_0^2)$	$(b_1^1, b_1^2)$	$(b_2^1, b_2^2)$	$(c_0^1, c_0^2)$	$(c_1^1, c_1^2)$	$(c_2^1, c_2^2)$	$\det(v)$	$\text{tr}(v)$
TSC	3	(0,4)	(4,0)	(0,2)	(2,0)	-	(0,1)	(5,0)	-	2.0	4.0
CYV	3	(2,1)	(3,4)	(4,6)	(2,0)	-	(0,4)	(4,0)	-	2.0	7.172
TSC	4	(0,4)	(4,4)	(0,2)	(2,2)	-	(0,1)	(5,1)	(1,5)	3.515	6.0
CYV	4	(2,4)	(3,7)	(4,0)	(6,6)	-	(7,2)	(0,7)	(4,4)	0.686	8.0
TSC	5	(0,4)	(4,4)	(0,2)	(2,2)	(2,2)	(0,1)	(5,1)	(3,7)	3.515	8.0
CYV	5	(0,4)	(4,4)	(0,2)	(2,3)	(2,2)	(3,0)	(2,2)	(3,7)	2.686	8.586

**Table 2.3** 4-PSK Space Time Trellis Codes with  $N_T = 3$  [12]

code	$v$	$(a_0^1, a_0^2, a_0^3)$	$(a_1^1, a_1^2, a_1^3)$	$(a_2^1, a_2^2, a_2^3)$	$(b_0^1, b_0^2, b_0^3)$	$(b_1^1, b_1^2, b_1^3)$	$(b_2^1, b_2^2, b_2^3)$	$\det(v)$	$\text{tr}(v)$
CYV	2	(0,2,2)	(1,2,3)	-	(2,3,3)	(2,0,2)	-	0	16
CYV	3	(2,2,2)	(2,1,1)	-	(2,0,3)	(1,2,0)	(0,2,2)	0	20
CYV	4	(1,2,1)	(1,3,2)	(3,2,1)	(2,0,2)	(2,2,0)	(2,0,2)	0	24

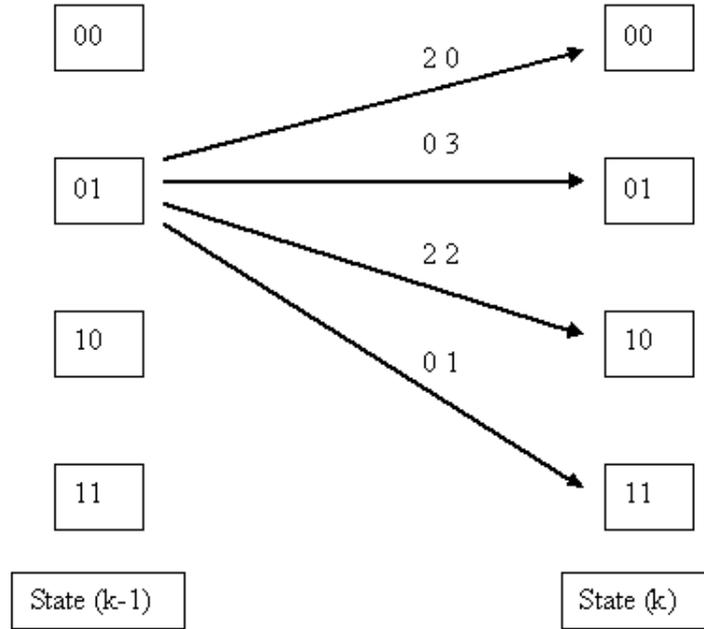
### 2.1.5 Trellis Example

The coefficients in the previous tables can be used to construct trellis diagrams. The following example shows how this is done, using the Chen (CYV) 4 state ( $v=2$ ) coefficients from Table 2.1.

**Figure 2.1** STTC 4 state encoder,  $M=4$ ,  $N_T=2$ 

The encoder can be thought of as a FIR type filter, as shown in Fig. 2.1. The input bits ( $I_k^j$ ) are placed in shift registers, depicted by the middle row. The inputs are multiplied by the coefficients for each transmitter, and then added using modulo arithmetic of base  $M$  (4). The coefficients are shown in the top and bottom rows, for transmitter 1 and 2 respectively. The final output for each transmitter is used to select one of the  $M$  symbols.

This structure can be extended into a state trellis diagram. Each state is represented by the previous data input bits. For example state 2 ("0 1") means that the previous bits  $I_{k-1}^1 = 0, I_{k-1}^2 = 1$ . The output symbols for each state can then be calculated as shown in Fig. 2.1. An example of this is shown in Fig. 2.2. The output symbols are depicted above the trellis transition lines, and have been calculated using equation 2.31, as shown in Table 2.4.



**Figure 2.2** Trellis diagram for state 2 of the 4 state Chen trellis code

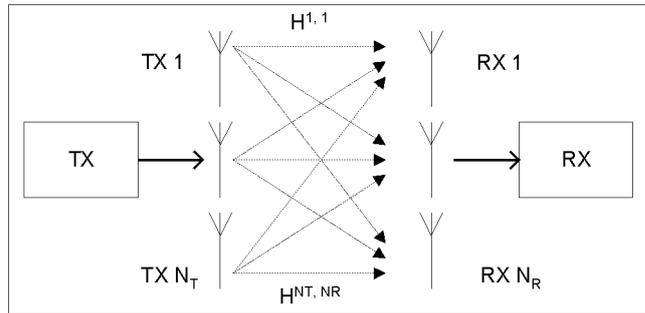
**Table 2.4** Transmitted Symbols

New State		Old State		TX 1		TX 2	
$I_k^1$	$I_k^2$	$I_{k-1}^1$	$I_{k-1}^2$	Sum	Symbol	Sum	Symbol
0	0	0	1	$0*0+0*2+0*1+1*2$	2	$0*2+0*3+0*2+1*0$	0
0	1	0	1	$0*0+1*2+0*1+1*2$	0	$0*2+1*3+0*2+1*0$	3
1	0	0	1	$1*0+0*2+0*1+1*2$	2	$1*2+0*3+0*2+1*0$	2
1	1	0	1	$1*0+1*2+0*1+1*2$	0	$1*2+1*3+0*2+1*0$	1

## 2.2 CHANNEL MODELS

### 2.2.1 Flat Fading System Model

Consider a flat fading MIMO system consisting of  $N_T$  transmit antennas and  $N_R$  receive antennas as shown in Fig. 2.3. A data stream is encoded into  $N_T$  streams of symbols at the transmitter using a constellation set of size  $M$ . At each time slot  $k$ , symbols are simultaneously transmitted from all  $N_T$  antennas, where symbol  $x_k^j$  is transmitted from antenna  $j$ . The observed signal at each receive antenna consists of a noisy superposition of all  $N_T$  symbols and AWGN. Let the impulse response of each channel between transmitter  $j$  and receiver  $i$  be denoted as  $h^{i,j}$ . Each channel is assumed have an identical, independent distribution, and quasi static flat rayleigh fading is assumed. Each channel  $h^{i,j}$  is modelled as a random complex Gaussian sample with zero mean and  $\sigma_c^2 = 1$ , where  $\sigma_c^2$  is the channel variance. The AWGN at time slot  $k$  and receiver  $i$  is modelled by a complex white Gaussian variable  $n_k^i$  with variance  $N_0$ . Therefore the



**Figure 2.3** A typical MIMO system model

observed signal  $y_k^i$  can be described as

$$y_k^i = \sum_{j=1}^{N_T} h^{i,j} x_k^j + n_k^i, \quad k = 1, \dots, K \quad (2.32)$$

This can be converted into a vector-matrix MIMO model by defining vectors using the elements in Eq. 2.32 as

$$\mathbf{y}_k = \begin{pmatrix} y_k^1 \\ y_k^2 \\ \vdots \\ y_k^{N_R} \end{pmatrix}, \quad \mathbf{x}_k = \begin{pmatrix} x_k^1 \\ x_k^2 \\ \vdots \\ x_k^{N_T} \end{pmatrix}, \quad \mathbf{n}_k = \begin{pmatrix} n_k^1 \\ n_k^2 \\ \vdots \\ n_k^{N_R} \end{pmatrix} \quad (2.33)$$

this gives a MIMO system equation of

$$\mathbf{y}_k = \mathbf{H}\mathbf{x}_k + \mathbf{n}_k \quad (2.34)$$

where element  $i, j$  of channel matrix  $\mathbf{H}$  is  $h^{i,j}$ .

## 2.2.2 Frequency Selective System Model

The frequency selective fading channel can be modelled as a tapped delay line filter, with symbol spaced taps. Using this model, the received values  $y_k^i$ , at each received antenna  $i$  can be modelled as

$$y_k^i = \sum_{m=0}^{L-1} \sum_{j=1}^{N_T} h_m^{i,j} x_{k-m}^j + n_k^i \quad (2.35)$$

where  $L$  is the length of the channel impulse response. To model a MIMO system, vectors as in Eq. 2.33 are introduced. Also a  $N_R \times N_T$  channel matrix tap is defined

as

$$\mathbf{H}_m = \begin{pmatrix} h_m^{1,1} & \dots & h_m^{1,N_T} \\ \vdots & \ddots & \vdots \\ h_m^{N_R,1} & \dots & h_m^{N_R,N_T} \end{pmatrix} \quad (2.36)$$

by defining the overall channel response

$$\mathbf{H} = [\mathbf{H}_0, \mathbf{H}_1, \dots, \mathbf{H}_{L-1}] \quad (2.37)$$

and

$$\mathbf{x}_{k:k-L+1} = \begin{pmatrix} \mathbf{x}_k \\ \mathbf{x}_{k-1} \\ \vdots \\ \mathbf{x}_{k-L+1} \end{pmatrix} \quad (2.38)$$

the received vector can be written as

$$\mathbf{y}_k = \mathbf{H}\mathbf{x}_{k:k-L+1} + \mathbf{n}_k \quad (2.39)$$

## 2.3 DECODING METHODS

As shown in [59], STTCs can be decoded using the Viterbi algorithm. When using the Viterbi algorithm, the major implementation complexity is due to calculating the trellis path metrics, which involves the add-compare-select (ACS) process. Memory is also required to store the metric history and path. However, the traceback of the trellis is very simple once the path metrics has been calculated. Therefore, the complexity of the decoding can be approximated by analysing the number of operations required to calculate one trellis step, or time period.

### 2.3.1 Flat Fading Channel

In the flat fading channel no equaliser is required. Therefore the method for decoding is the same as that for any standard convolutional code. A trellis is created with the number of states  $N_{ST}$ , equal the number of states used in the STTC construction. Each state stores the history of previously transmitted data. Each branch contains  $N_T$  symbols, which are calculated from both the history, and current symbol using the encoder. Assuming perfect channel knowledge the branch metric  $\lambda$  for each possible branch can be written as follows, where  $\hat{y}$  is the hypothesised received value given the current trellis state, hypothesised data symbol and channel taps.

$$\begin{aligned}
\lambda &= \|y_k^i - \hat{y}_k^i\|^2 \\
&= \sum_{i=1}^{N_R} \|y_k^i - \sum_{j=1}^{N_T} h^{i,j} x_k^j\|^2
\end{aligned} \tag{2.40}$$

The complexity for a single step is calculated by looking at the number of complex Multiply and Accumulate (MAC) instructions required. Each step has  $N_{ST}$  states, with  $M$  paths per state. Calculating the metric per path requires approximately  $(N_T + 1) * N_R$  complex MAC instructions, from Eq. 2.40. Therefore, the total complexity is  $N_{ST} * M * (N_T + 1) * N_R$  per time step. If the traceback is only done when the frame is completed, then it only requires approximately one equivalent instruction per time step and can be ignored. The memory storage requirements are the minimum metric value and path for each state, with an order of  $N_{ST}$  per time step.

### 2.3.2 Frequency Selective Fading Channel

The frequency selective channel causes ISI, which requires equalisation. However, a large Viterbi Algorithm using joint equalisation and decoding can be created. The Viterbi Algorithm requires  $M^{L-1}$  states for equalisation, and  $N_{ST}$  states for decoding. The total number of states is then  $N_{ST} * M^{L-1}$ , as the trellis is a joint trellis. This means the decoding can be very complex, even for moderate values of  $M, L$ . The branch metrics are then calculated as

$$\begin{aligned}
\lambda &= \|y_k^i - \hat{y}_k^i\|^2 \\
&= \sum_{i=1}^{N_R} \|y_k^i - \sum_{l=0}^{L-1} \sum_{j=1}^{N_T} h_l^{i,j} x_{k-l}^j\|^2
\end{aligned} \tag{2.41}$$

The complexity of decoding in the frequency selective channel is much larger. The total number of states is  $N_{ST} * M^{L-1}$ , with  $M$  paths per state. From Eq. 2.41 it can be seen that calculating one metric requires approximately  $(N_T * L + 1) * N_R$  complex MAC instructions. The overall complexity is

$$N_{ST} * M^{L-1} * M * (N_T * L + 1) * N_R = N_{ST} M^L (N_T L + 1) N_R$$

complex MAC instructions per time step. The memory storage requirements have an order of  $N_{ST} * M^{L-1}$  per time step.

## Chapter 3

---

### PARTITIONED VITERBI ALGORITHM

The Partitioned Viterbi Algorithm (PVA) forms one of the key tools used in this thesis to successfully decode STTCs in frequency selective channels. The PVA was originally developed by Miller *et.al.* [43] for use in frequency selective MIMO systems. Such systems suffer from ISI effects caused by the dispersive channel. Dispersive channels are common in systems that operate at high data rates. While the original PVA was designed to work with independent transmitted data streams, it can easily be extended to work with correlated data streams, such as those in a STTC system. In such a system it is used to equalise the received signals before decoding by a STTC decoder.

The benefit of using the PVA algorithm as an equaliser is reduced complexity, with some performance penalty. Maximum Likelihood (ML) equalisation of uncoded MIMO systems has a total number of states equal to  $M^{N_T(L-1)}$ , where  $M$  is the size of the symbol alphabet,  $N_T$  is the number of transmit antennas, and  $L$  is the length of the channel response. Such a system is exponentially complex in the number of transmit antennas. The PVA algorithm has a total number of states equal to  $N_T M^{(L-1)}$ , i.e. it is linearly complex in the number of transmit antennas.

The PVA consists of three major sections. These are channel estimation, pre-filtering, and trellis based decoding. The model used to develop the PVA is slightly different to that used in Section 2.2. This model, and a description of the PVA as used in [43] follows.

#### 3.1 SYSTEM MODEL

Miller *et.al.* developed the PVA for a complex-baseband MIMO channel model [43]. It is assumed that the data sequence is generated from independent symbols, that are uniformly distributed over an alphabet of size  $M$ . At the transmitter, the data symbols are convolved with the impulse response of the chosen pulse shape. The pulse shape is chosen to restrict the power, and bandwidth of the system.

The signal then propagates through the frequency selective channel. This channel is modelled as a quasi-static tapped delay line filter. The assumptions behind this are

that the symbol period is short, so that the fading over a burst of symbols is constant. At the receiver, the signal is corrupted with additive white Gaussian noise (AWGN). The AWGN is considered to be due to front end electronic noise rather than part of the channel. The signal is then passed through a bandlimiting filter to restrict the power of the noise, and sampled at a rate of  $N_s$  samples per symbol, where  $N_s$  is chosen to prevent oversampling.

Due to the oversampling factor, the system model is slightly different than that described in Section 2.2.2. Let  $y_k^{i,s}$  denote the  $s$ th sample of  $N_s$ , at receive antenna  $i$ , and time  $k$ . This results in a  $N_R N_s$  vector  $\mathbf{y}_k$ . Also, the channel matrix tap weight  $h_m^{i,j,s}$  represents the response between transmitter  $j$ , receiver  $i$ , and sample  $s$ . This results in a  $N_R N_s \times N_T$  channel matrix  $\mathbf{H}_m$  similar to Eq. 2.36, such that the overall response is given by

$$\mathbf{H} = [\mathbf{H}_0, \mathbf{H}_1, \dots, \mathbf{H}_{L-1}] \quad (3.1)$$

This is now equivalent to the model presented in Section 2.2.2, but with an oversampling parameter added. The final received vector is

$$\begin{aligned} \mathbf{y}_k &= \sum_{n=0}^{L-1} \mathbf{H}_n \mathbf{x}_{k-n} + \mathbf{n}_k \\ &= \mathbf{H} \mathbf{x}_{k:k-L+1} + \mathbf{n}_k, \end{aligned} \quad (3.2)$$

where  $\mathbf{x}_{k:k-L+1}$  is a  $N_T L$  vector, which represents the transmitted sequence as shown in Eq. 2.38.

## 3.2 CHANNEL ESTIMATES

Channel state information (CSI) is required to calculate both the pre-filter and to generate metrics in the decoder. Assuming that a block, rather than continuous, transmission format is used, the channel must be estimated via training symbols. Work has been done in investigating training sequence design for MIMO systems, for example [27]. In the case of continuous transmission, adaptive channel estimation could be used.

In [43], it is assumed that a burst structure similar to GSM is used, with a number of training symbols ( $N_t$ ) inserted into a data sequence of length ( $N_d$ ). These symbols are then used to generate an estimate of the channel impulse response (CIR) by forming a least-squares estimate. Let the received training sequence  $N_R N_s \times (N_t - L + 1)$  matrix  $\mathbf{Y} = [\mathbf{y}_L \dots \mathbf{y}_{N_t}]$ , and the transmitted training sequence  $N_T L \times (N_t - L + 1)$  matrix

$\mathbf{X} = [\mathbf{X}_L \dots \mathbf{X}_{N_t}]$ , where

$$\mathbf{X}_k = \mathbf{x}_{k:k-L+1} = \begin{pmatrix} \mathbf{x}_k \\ \mathbf{x}_{k-1} \\ \vdots \\ \mathbf{x}_{k-L+1} \end{pmatrix} \quad (3.3)$$

Under noise-free conditions

$$\mathbf{Y} = \mathbf{H}\mathbf{X} \quad (3.4)$$

The channel experiences AWGN, which is assumed to be due to the receiver front end, and the channel estimate  $\hat{\mathbf{H}}$  can then be determined from 3.4 by multiplying both sides by  $\mathbf{X}^H(\mathbf{X}\mathbf{X}^H)^{-1}$ .

$$\hat{\mathbf{H}} = \mathbf{Y}\mathbf{X}^H(\mathbf{X}\mathbf{X}^H)^{-1} \quad (3.5)$$

This estimate is an MMSE estimate, where it is assumed the receiver has perfect synchronisation with the transmitter. In a practical system, once the training sequence  $\mathbf{x}_{1:N_t}$  has been determined,  $\mathbf{X}^H(\mathbf{X}\mathbf{X}^H)^{-1}$  can be pre-computed and stored in a lookup table. This makes the resulting estimation relatively efficient.

### 3.3 PRE-FILTER

The PVA uses a pre-filter to split the received signal into estimates of the individual transmitted symbol streams. The pre-filter attempts to minimise the filtered correlation between symbol streams from different transmitters. This allows a trellis decoder to decode each transmitted symbol stream independently, which is the basis of the PVA. It is shown in [43] that the ideal pre-filter results in a filtered channel, denoted by  $\tilde{\mathbf{H}}$ , that maximises the delay of tentative decisions from the decoder. It should also be reasonable to compute, and not produce time or space correlation of the noise. In the SISO case, such a pre-filter would be the Whitened Matched Filter (WMF) [22]. However, in [44] it is shown that the WMF does not always exist in the MIMO case. Because of this, the feed forward filter of an MMSE vector decision feedback equaliser (DFE) is used as a pre-filter. It is shown in [3] (SISO case) and [44] (MIMO case) that this filter approaches the WMF (where it exists) as the SNR and number of taps used in the filter tend to infinity. The benefit of using the DFE pre-filter is that it always exists, even when the WMF does not. Further work on interference cancellation using prefilters and MIMO equalisation has been undertaken in [28].

The method in [43] used to calculate the pre-filter is summarised in the following. Let a generic pre-filter be denoted as a MIMO FIR filter, with matrix taps. Each tap weight is denoted by a  $N_T \times N_R N_s$  matrix  $F_m$ . It is assumed that  $L_f$  taps are used, where  $L_f$  is the length of the pre-filter tap support, this gives an entire filter of

$\mathbf{F} = [F_0, F_1, \dots, F_{L_f-1}]$ . Filtering the received signal from Eq. 3.2 gives

$$\begin{aligned}
\tilde{\mathbf{y}}_k &= \sum_{m=0}^{L_f-1} \mathbf{F}_m \mathbf{y}_{k-m} \\
&= \sum_{n=0}^{L_f+L-2} \left( \sum_{m=0}^{L_f-1} \mathbf{F}_m \mathbf{H}_{n-m} \right) \mathbf{x}_{k-n} + \sum_{m=0}^{L_f-1} \mathbf{F}_m \mathbf{n}_{k-m} \\
&= \sum_{n=0}^{L_f+L-2} \tilde{\mathbf{H}}_n \mathbf{x}_{k-n} + \tilde{\mathbf{n}}_k
\end{aligned} \tag{3.6}$$

with the pre-filtered channel defined as  $\tilde{\mathbf{H}}_n = \sum_{m=0}^{L_f-1} \mathbf{F}_m \mathbf{H}_{n-m}$  and the pre-filtered noise as  $\tilde{\mathbf{n}}_k = \sum_{m=0}^{L_f-1} \mathbf{F}_m \mathbf{n}_{k-m}$ .

A block of  $L_f$  received vectors can be written as

$$\begin{aligned}
\begin{pmatrix} \mathbf{y}_{k+L_f-1} \\ \mathbf{y}_{k+L_f-2} \\ \vdots \\ \mathbf{y}_k \end{pmatrix} &= \begin{pmatrix} \mathbf{H}_0 & \mathbf{H}_1 & \dots & \mathbf{H}_{L-1} & \mathbf{0} & \dots \\ \mathbf{0} & \mathbf{H}_0 & \mathbf{H}_1 & \dots & \mathbf{H}_{L-1} & \mathbf{0} \\ \vdots & & & & & \vdots \\ \mathbf{0} & \dots & \mathbf{H}_0 & \mathbf{H}_1 & \dots & \mathbf{H}_{L-1} \end{pmatrix} \\
&\quad \cdot \begin{pmatrix} \mathbf{x}_{k+L_f-1} \\ \mathbf{x}_{k+L_f-2} \\ \vdots \\ \mathbf{x}_{k-L+1} \end{pmatrix} + \begin{pmatrix} \mathbf{n}_{k+L_f-1} \\ \mathbf{n}_{k+L_f-2} \\ \vdots \\ \mathbf{n}_k \end{pmatrix}
\end{aligned} \tag{3.7}$$

Following [44] and [61], the  $\mathbf{H}$  matrix in Eq. 3.7 is separated into columns. The first  $N_T(L_f - 1)$  columns represent the  $L_f N_R N_s \times N_T(L_f - 1)$  matrix  $F_{\text{fut}}$  which represents the filter response from "future" symbols. The future symbols have an effect on the filter response due to the  $L_f$  delay of the filter. The middle  $N_T$  columns represent the  $L_f N_R N_s \times N_T$  matrix  $F_{\text{pres}}$  which represents the currently transmitted symbols. Finally, the last  $N_T(L - 1)$  columns represent the  $L_f N_R N_s \times N_T(L - 1)$  matrix  $F_{\text{past}}$  which represents the previously transmitted symbols. Therefore Eq. 3.7 can be rewritten as

$$\begin{aligned}
\mathbf{y}_{k+L_f-1:k} &= (F_{\text{fut}} F_{\text{pres}} F_{\text{past}}) \mathbf{x}_{k+L_f-1:k-L+1} + \mathbf{n}_{k+L_f-1:k} \\
&= F_{\text{pres}} \mathbf{x}_{k+L_f-1:k+1} + F_{\text{fut}} \mathbf{x}_k + F_{\text{past}} \mathbf{x}_{k-1:k-L+1} + \mathbf{n}_{k+L_f-1:k}
\end{aligned} \tag{3.8}$$

Assuming correct past decisions (i.e.  $\hat{\mathbf{x}}_{k-1:k-L+1} = \mathbf{x}_{k-1:k-L+1}$ ) Eq. 3.8 can be used

to write the FIR mmse-DFE estimate as

$$\begin{aligned}
\hat{\mathbf{x}}_k &= \mathbf{F}\mathbf{y}_{k+L_f-1:k} - \mathbf{F}F_{\text{past}}\mathbf{x}_{k-1:k-L+1} \\
&= \mathbf{F}F_{\text{fut}}\mathbf{x}_{k+L_f-1:k+1} + \mathbf{F}F_{\text{pres}}\mathbf{x}_k + \mathbf{F}\mathbf{n}_{k+L_f-1:k} \\
&= [\tilde{\mathbf{H}}_0, \tilde{\mathbf{H}}_1, \dots, \tilde{\mathbf{H}}_{L_f-2}]\mathbf{x}_{k+L_f-1:k+1} + \tilde{\mathbf{H}}_{L_f-1}\mathbf{x}_k + \tilde{\mathbf{n}}_{k+L_f-1}
\end{aligned} \tag{3.9}$$

The forward filter  $\mathbf{F}$  is designed such that it minimises the mse  $E\{\|\hat{\mathbf{x}}_k - \mathbf{x}_k\|^2\}$ . It is shown in [43] and [61] that the mse is minimised if  $\mathbf{F}$  satisfies the following equation.

$$([F_{\text{fut}}, F_{\text{pres}}][F_{\text{fut}}, F_{\text{pres}}]^H + N_0\mathbf{I})\mathbf{F}^H = F_{\text{pres}} \tag{3.10}$$

As  $([F_{\text{fut}}, F_{\text{pres}}][F_{\text{fut}}, F_{\text{pres}}]^H + N_0\mathbf{I})$  is hermitian and positive definite, Eq. 3.10 can be solved efficiently using the Cholesky decomposition.

Miller *et.al.* [43] also showed that the DFE pre-filter has some useful properties. Each channel tap  $\tilde{\mathbf{H}}_k$  is a  $N_T \times N_T$  matrix, with no dependence on  $N_R$  or  $N_s$ . This means that increasing  $N_R$  or  $N_s$  will increase the complexity of solving Eq. 3.10, but not increase the complexity of the Viterbi algorithm. It is also shown that the pre-filtered noise correlation is approximately white if  $N_R \geq N_T$ .

Examining 3.9, it is apparent that the MSE is minimised when the first  $L_f - 1$  pre-filtered channel taps  $[\tilde{\mathbf{H}}_0, \tilde{\mathbf{H}}_1, \dots, \tilde{\mathbf{H}}_{L_f-2}]$  approximate zero matrices, and channel tap  $\tilde{\mathbf{H}}_{L_f-1}$  approximates the identity matrix [43]. These approximations can be used to write the output of the filter ( $\hat{y}$ ) as

$$\begin{aligned}
\tilde{y}_{k+L_f-1}^t &= \sum_{j=1}^{N_T} \sum_{n=0}^{L_f+L-2} \tilde{h}_n^{t,j} x_{k+L_f-1-n}^j + \tilde{n}_{k+L_f-1}^t \\
&\approx \sum_{j=1}^{N_T} \sum_{n=0}^{L-1} \tilde{h}_{n+L_f-1}^{t,j} x_{k-n}^j + \tilde{n}_{k+L_f-1}^t \\
&\approx \sum_{n=0}^{L-1} \tilde{h}_{n+L_f-1}^{t,t} x_{k-n}^t + \sum_{j \neq t} \sum_{n=1}^{L-1} \tilde{h}_{n+L_f-1}^{t,j} x_{k-n}^j + \tilde{n}_{k+L_f-1}^t
\end{aligned} \tag{3.11}$$

Eq. 3.11 shows that the filtered received value from transmitter  $j$  is a function of the  $L$  most recent symbols from transmitter  $j$  and  $(L - 1)$  most recent symbols from the remaining transmitters. This is used to generate the inputs to the PVA algorithm.

### 3.3.1 Alternative Methods

In [43], it was decided to use the DFE pre-filter due to its useful characteristics. However a search for the optimum or best pre-filter was not done. Work by Gerstacker [29] *et.al.* developed a prediction error pre-filter design which does not require knowledge of the noise variance. This prediction error pre-filter is generated by inverting a block

Toeplitz matrix which can be done more efficiently [2] than the Cholesky decomposition required by the DFE [43]. Pre-filter design has also been investigated by Younis et.al. [75]. Equalisation in the frequency domain has also been investigated, for example in [9] where the Singular Value Decomposition (SVD) of the channel frequency response was used to generate the pre-filter.

### 3.4 FEEDBACK TRELLIS STRUCTURE

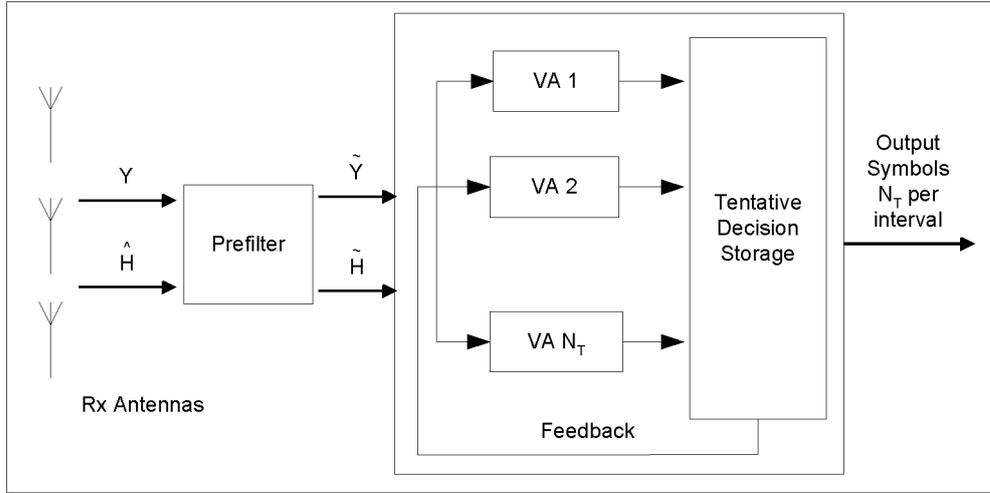


Figure 3.1 PVA block diagram

Figure 3.1 shows the structure of the PVA. The inputs to the system are pre-filtered received signals ( $\tilde{\mathbf{y}}_{k+L_f}$ ) and channel impulse response ( $\tilde{\mathbf{H}}$ ). Using the Viterbi algorithm,  $N_T$  trellis searches are performed in parallel. Each algorithm updates the tentative decisions at the end of each symbol period. After a decoding delay  $d$ , an estimate of the transmitted symbol stream  $\hat{\mathbf{x}}_{k-d}$  is output, represented as an  $N_T$  vector with one symbol for each transmit antenna.

Following [43], some definitions are made to describe the PVA in more detail. Let the estimated symbol stream  $\hat{\mathbf{x}}_{k:k-L+1}$  be divided into two overlapping sequences  $\hat{\mathbf{x}}_{k-1:k-L+1}$  (previous sequence) and  $\hat{\mathbf{x}}_{k:k-L+2}$  (current sequence). Each sequence can take on one of  $M^{L-1}$  states, where each state represents a symbol combination. If  $\mathbf{i}$  represents the previous state, and  $\mathbf{j}$  represents the current state, then the  $t^{\text{th}}$  branch metric of the Viterbi algorithm can be written as

$$\lambda^t(\mathbf{i}, \mathbf{j}, k) = \left| \tilde{y}_{k+L_f-1}^t - \sum_{n=0}^{L-1} \tilde{h}_{n+L_f-1}^{t,t} \hat{x}_{k-n}^t - \phi(k, t) \right|^2 \quad (3.12)$$

where

$$\phi(k, t) = \sum_{m \neq t} \sum_{n=1}^{L-1} \tilde{h}_{n+L_f-1}^{t,m} \tilde{x}_{k-n}^m \quad (3.13)$$

represents a feedback term from the other Viterbi decoders. This feedback term is calculated using the tentative decisions  $\bar{x}_{k-1:k-L+1}^m$ . It appears that the accuracy of these decisions determine the overall performance of the equaliser, this is shown in Chapter 7.

It would also be possible to use reduced-state sequence estimation (RSSE) [20] instead of a full Viterbi in each of the  $N_T$  algorithms. Such an approach uses filtering to concentrate the received signal power in a small a time period as possible. For the PVA algorithm, this would reduce  $L$ , and hence the number of states required for the sequence estimation.

### 3.5 PERFORMANCE

In [43], Miller *et.al.* investigated using the PVA in a  $N_T = 2, N_R = 2$  multiuser system. The system was simulated using a GSM type data structure, with a training sequence of  $N_t = 26$  symbols, and a data sequence of  $N_d = 61$  symbols, using BPSK. the channel was modelled as a frequency selective channel, using a tapped delay line model, with equal power on each tap. Also taken into account were the effects of pulse shaping and oversampling.

The results show that the PVA algorithm suffers both a performance loss and diversity loss compared to ML decoding. In the multiuser situation (independent transmit symbols) the ML equaliser is the Vector Viterbi Algorithm (VVA) [43], [67]. The performance loss shown is around 2db at a BER of  $10^{-4}$ . The diversity loss is shown as the PVA slope of the BER curve is slightly more gradual the VVA slope.

Other interesting results show that as the number of transmit antennas increases, for a given number of receive antennas the performance of both the PVA and VVA decreases. It also shows that good performance of the PVA requires  $N_R \geq N_T$ . This is due to the assumption of the pre-filtered noise being white, being no longer valid. However, it may be possible to use a coloured noise version of the Viterbi algorithm as shown in [65]. This result shows the difference in behaviour between a multi-user system and an STTC system. In the former, the transmitted data streams are independent, so an extra data stream (antenna) decreases system performance. In the latter, the transmitted data streams are correlated, so ML detection actually improves system performance, albeit with increased complexity.

### 3.6 COMPLEXITY

The complexity of implementing the PVA is harder to analyse than ML detection. This is due to the overhead of calculating the DFE pre-filter, and filtering the estimated channel  $\hat{\mathbf{H}}$  and received sequence  $\mathbf{y}$ . To enable an approximation of the complexity, it is assumed that the DFE pre-filter and  $\tilde{\mathbf{H}}$  are calculated once per frame, or  $N_d$  data

symbols, and  $\tilde{\mathbf{y}}_{\mathbf{k}}$  calculated once per time step.  $\tilde{\mathbf{H}} = \mathbf{F} \otimes \hat{\mathbf{H}}$  and represents the overall response of the filtered system including channel effects and pulse shaping.

Calculation of the DFE pre-filter requires the Cholesky decomposition of a  $L_f N_R N_s \times L_f N_R N_s$  matrix. The decomposition has a complexity order of  $N^3/6$  [49] where  $N$  is the size of the matrix. Therefore the complexity overhead per time step is approximately  $\frac{(L_f N_R N_s)^3}{6N_d}$ . Obviously the length of the filter  $L_f$  should be kept as short as possible, with reasonable performance, to reduce this overhead.

Filtering the estimated channel sequence  $\tilde{\mathbf{H}}$  requires the convolution of  $\mathbf{F}$  and  $\hat{\mathbf{H}}$ . It is assumed that  $\hat{\mathbf{H}}$  is already known. In the PVA algorithm, only the last  $L$  taps of  $\tilde{\mathbf{H}}$  are required. The calculation of each channel tap requires  $L_f$  computations of a  $N_T \times N_R N_s$  filter tap  $\mathbf{F}_{\mathbf{m}}$  and a  $N_R N_s \times N_T$  channel estimate matrix  $\tilde{\mathbf{H}}_{\mathbf{k}}$ . The resultant complexity is  $L * L_f * N_R * N_s * N_T * N_T$  complex MAC instructions. The overhead can also be divided over all  $N_d$  data symbols.

The filtered received sequence,  $\tilde{\mathbf{y}}$  is only required once per time step. This is calculated by multiplying  $L_f$  filter taps by a  $N_R N_s$  input vector  $\mathbf{y}_{\mathbf{k}}$ . The resulting complexity is  $L_f * N_R * N_s * N_T$  per time period.

The total overhead is therefore

$$\frac{(L_f N_R N_s)^3}{6N_d} + \frac{L L_f N_R N_s N_T N_T}{N_d} + L_f N_R N_s N_T \quad (3.14)$$

which is clearly dominated by the  $(L_f N_R N_s)^3$  term. As  $N_R$  is determined by the MIMO system,  $L_f$  and  $N_s$  should be kept as small as feasible to reduce computational complexity.

The implementation of the PVA requires a separate trellis for each transmit antenna  $N_T$ . Each PVA trellis requires  $M^{L-1}$  states for equalisation, with  $M$  paths per state. The complexity of calculating each path metric from Eq. 3.12 requires  $L + 1$  complex MAC instructions. The calculation of  $\phi(k, t)$ , Eq. 3.13 requires  $(N_T - 1) * (L - 1)$  complex MAC instructions, but this is only required once per time step. This calculation also requires a tentative estimate, for each symbol  $M$  and requires  $N_T * L$  complex MACs.

Therefore the complexity of calculating each trellis time step is

$$\begin{aligned} & N_T M^{L-1} M(L + 1) + N_T ((N_T - 1)(L - 1) + M N_T L) \\ & = N_T (M^L (L + 1) + (N_T L (M + 1) - L - N_T + 1)) \end{aligned} \quad (3.15)$$

which is dominated by the  $N_T M^L (L + 1)$  term for moderate  $N_T$ . This is approximately a factor of  $N_R N_s N_T$  less complex than the ML decoding shown in Eq. 2.42, not allowing for the overhead.

## Chapter 4

---

### SOFT OUTPUT VITERBI ALGORITHM

The Soft Output Viterbi Algorithm (SOVA) was originally developed by Hagenaur and Hoeher [30]. It is designed to extend the standard Viterbi algorithm (VA) by providing soft output values of the *a posteriori* reliability values for each bit. These soft output values can then be used as input values to other decoders, as for example in a concatenated coding scheme. The SOVA is similar to the BCJR algorithm, but is less complex and suffers a small performance penalty.

#### 4.1 ORIGINAL DEVELOPMENT

The Viterbi algorithm [68], [23] was originally developed as a Maximum Likelihood Sequence Estimator (MLSE). The VA can operate on either soft or hard inputs, and produces a hard output. It is used to estimate the most likely path in a finite state Markov chain. A common example of this is in convolutional, or trellis codes. The hard output of the Viterbi means a concatenated type system, i.e. a decoder following the Viterbi decoder, will suffer a performance loss due to the loss of information.

There have been a number of methods designed to prevent this. Some examples of this include Maximum A Posteriori (MAP) decoding, iterative Viterbi algorithms [70], list Viterbi algorithms [48] and sphere decoding [69] [17]. MAP decoding, for example the BCJR algorithm [7], calculates the most likely bit to have been transmitted at any time, and also provides reliability information for this bit. The iterative Viterbi Algorithm (IVA) repeatedly decodes a sequence while updating the metrics, until a valid codeword is found. List Viterbi algorithms (LVA) provide the most likely sequence, and a list of possible alternatives. Work has also been done in using LVAs to generate soft outputs [48]. Sphere decoders look at a set number, or volume, of possible bits and choose the most likely sequence of such bits. It can also generate reliability information for these bits. Such methods offer large (up to several dB) performance gains over hard output decoders.

In [30] a novel way of producing reliability values from the VA was developed. This original work was done assuming a binary trellis, with memory  $v$  and  $2^v$  states.

Summarising [30], the SOVA can be developed as follows.

Assume that a VA makes a final decision after a delay  $\delta$ , where  $\delta$  is such that all  $2^v$  surviving paths have merged with a sufficiently high probability (typically around  $\delta = 5v$ ). The VA then selects the path with the smallest metric, which for a AWGN channel is given by

$$M_m = \frac{E_s}{N_0} \sum_{j=k-\delta}^k \sum_{n=1}^N \left( y_{j,n} - x_{j,n}^{(m)} \right)^2, \quad m = 1, 2 \quad (4.1)$$

where  $x_{j,n}^{(m)}$  is the  $n$ -th of  $N$  bits on the  $m$ -th path at time  $j$ . Also,  $y_{j,n}$  is the received value at this position, and  $E_s/N_0$  is the SNR. This can be used to estimate the probability of choosing each path as

$$P(\text{path } m) \approx e^{-M_m}, \quad m = 1, 2 \quad (4.2)$$

If the path with the smallest metric is labelled by  $m = 1$ , then  $M_1 \leq M_2$ , so the VA would chose path 1. Then the probability of choosing the wrong survivor path is

$$\begin{aligned} p_{sk} &= \frac{e^{-M_2}}{e^{-M_1} + e^{-M_2}} \\ &= \frac{1}{1 + e^{M_2 - M_1}} \\ &= \frac{1}{1 + e^{\Delta}} \end{aligned} \quad (4.3)$$

where  $\Delta = M_2 - M_1 \geq 0$ .  $p_{sk}$  approaches 0.5 if  $M_1 \approx M_2$  or 0 if  $M_2 \gg M_1$ . This means that with probability  $p_{sk}$ , the VA has made errors in all  $e$  positions where the information bits of path 2 differ from path 1, i.e. if

$$u_j^{(1)} \neq u_j^{(2)}, \quad j = j_1, \dots, j_e \quad (4.4)$$

Positions where  $u_j^{(1)} = u_j^{(2)}$  are not affected by either path. Let  $\delta_m$  denote the length of both paths prior till they merge. Then there are  $e$  different information bits, and  $\delta_m - e$  identical bits. If the previous erroneous decision probabilities  $\hat{p}_j$  have been stored, the probabilities for  $e$  different decisions can be updated using

$$\hat{p}_j \leftarrow \hat{p}_j(1 - p_{sk}) + (1 - \hat{p}_j)p_{sk}, \quad j = j_1, \dots, j_e \quad (4.5)$$

with  $0 \leq \hat{p}_j \leq 0.5$  This formula only holds if random variables  $\hat{p}_j$  and  $p_{sk}$  are statistically independent, which is approximately true for most codes. Eq. 4.5 can be calculated using log-likelihood ratios as

$$\hat{L}_j = \log \frac{1 - \hat{p}_j}{\hat{p}_j}, \quad 0 \leq \hat{L}_j \leq \infty \quad (4.6)$$

Eqs. 4.3, 4.5 and 4.6 can be used to obtain

$$\hat{L}_j \leftarrow f(\hat{L}_j, \Delta) = \frac{1}{\alpha} \log \frac{1 + e^{(\alpha\hat{L}_j + \Delta)}}{e^\Delta + e^{\alpha\hat{L}_j}} \quad (4.7)$$

where  $\alpha$  prevents overflow with increasing SNR. The function  $f(\hat{L}_j, \Delta)$  can be tabulated to reduce computational overhead. To achieve  $E[\hat{L}_j] = 1$ ,

$$\alpha = 4d_{free} \frac{E_s}{N_0} \quad (4.8)$$

where  $d_{free}$  is the free distance of the code. Eq. 4.7 can be approximated by

$$f(\hat{L}_j, \Delta) = \min(\hat{L}_j, \Delta/\alpha) \quad (4.9)$$

Finally, the recursion method for the SOVA can be determined as follows.

- a) Classical Viterbi step:

For each state  $s_k$

Update the accumulated metric from state  $s_{k-1}$   
to state  $s_k$  using Eq. 4.1.

Find the new minimum accumulated metric,  
and store the corresponding data bits  $\hat{u}_j(s_k)$ .

- b) Soft-decoding update:

For each state  $s_k$

Store  $\Delta = M_2 - M_1$

Initialise  $\hat{L}_k(s_k) = +\infty$

For  $j = k - v$  to  $j = k - \delta_m$

Compare the two paths merging in  $s_k$ ,  
if  $\hat{u}_j^{(1)} \neq \hat{u}_j^{(2)}(s_j)$  then update  $\hat{L}_j := f(\hat{L}_j, \Delta)$

In a non-binary trellis of size  $N$ , the soft-decoding update is more complex. For optimum performance, each alternative path must be traced back. Let the metrics at each state be ordered such that  $M_i \leq M_j$ ,  $i < j$ ,  $i, j \in 1 \dots N$ . Step (b) above is repeated  $N - 1$  times, where  $\Delta = M_n - M_1$ ,  $n = 2, \dots, N$ . However, as typically  $M_1 \ll M_n$  as  $n \rightarrow N$ , the update equation Eq. 4.9 is determined by only the closest paths. A reasonable performance can be achieved by looking at only the best and second best metrics,  $M_1$  and  $M_2$  respectively.

## 4.2 COMPLEXITY

The implementation of the SOVA does not excessively add to the computational complexity of the Viterbi algorithm. It does require more memory storage, and added decoding delay, or latency. In the standard Viterbi, only the best metric and path are

stored for each state. The SOVA requires both the best, and alternative, metric and path to be stored, which doubles the memory requirements. The additional complexity is due to the traceback which is performed at each time step. The required traceback is on average about  $3v$ , and is often limited to a maximum of  $5v$  without performance degradation, where  $v$  is the length of the trellis memory in bits. In the worst case, each time step requires  $5v$  Add Compare Select (ACS) steps, and  $5v$  updates of  $\hat{L}_j$ . In simulations, typically around  $3v$  ACS and  $2v$  updates occur. For ease of comparison, it is assumed that each ACS and update is equivalent to one complex MAC instruction.

### 4.2.1 Performance

In [30], Hagenauer *et.al.* used the SOVA in a concatenated convolutional encoded scheme. The SOVA was used as the inner decoder, to improve the performance of the outer code. Their results show a large performance gain over hard decoding, and within 0.3 dB of MAP (BCJR) decoding. They also implemented the SOVA in an equaliser for a frequency selective fading channel, with a convolutional trellis code, and simulations showed a gain of around 4dB over hard equalisation. They also demonstrate other uses for the SOVA, including Trellis Coded Modulation (TCM).

## 4.3 COMPARISON TO BCJR

### 4.3.1 BCJR method

The BCJR algorithm named after its creators, Bahl, Cocke, Jelinek and Raviv [5] is an optimal decoding algorithm on a symbol by symbol basis. It is a soft output algorithm which has seen renewed interest with the development of "turbo-codes" [8]. In [5], the BCJR algorithm is developed to deal with the problem of estimating the a posteriori probabilities (APP) of states ( $\lambda_t(m)$ ) and transitions ( $\sigma_t(m', m)$ ) of a Markov source. This is achieved by calculating the following functions [5]

$$\begin{aligned}\lambda_t(m) &= \alpha_t(m) \cdot \beta_t(m) \\ \sigma_t(m', m) &= \alpha_{t-1}(m') \cdot \gamma_t(m', m) \cdot \beta_t(m)\end{aligned}\tag{4.10}$$

where

$$\begin{aligned}\alpha_t(m) &= P(S_t = m; \mathbf{Y}_1^t) \\ \beta_t(m) &= P(\mathbf{Y}_{t+1}^T | S_t = m) \\ \gamma_t(m', m) &= P(S_t = m; Y_t | S_{t-1} = m')\end{aligned}\tag{4.11}$$

The  $\alpha_t(m)$  and  $\beta_t(m)$  functions can be recursively calculated by

$$\alpha_t(m) = \sum_{m'} \alpha_{t-1}(m') \cdot \gamma_t(m', m) \quad (4.12)$$

$$\beta_t(m) = \sum_{m'} \beta_{t+1}(m') \cdot \gamma_{t+1}(m, m') \quad (4.13)$$

For  $\alpha_t(m)$ , the probability of the current state  $m$  is the sum of the probabilities of the previous state  $m'$  multiplied by the probability that the path from the previous state ends in the current state. For  $\beta_t(m)$ , the probability of the current state  $m$  is the sum of the probabilities of all future states  $m'$  multiplied by the probability that the path from  $m$  ends in  $m'$ .

This gives the following method to calculate the desired APP  $\lambda_t(m)$  and  $\sigma_t(m', m)$ .

- 1) Initialise  $\alpha_0(m)$  and  $\beta_\tau(m)$  according to the desired initial probability.
- 2) Forward Recursion: When  $\mathbf{Y}_\tau$  is received compute  $\gamma_t(m', m)$  and  $\alpha_t(m)$ .
- 3) Backward Recursion: When received sequence  $\mathbf{Y}_\tau$  is complete, calculate  $\beta_t(m)$ .
- 4) Calculate APP: Calculate  $\lambda_t(m)$  and  $\sigma_t(m', m)$  using Eq. 4.10.

### 4.3.2 Complexity

The BCJR is updated in both a forward and backwards direction, whereas the SOVA is updated only in the backwards direction. Also the calculation of both  $\alpha(m)$  and  $\beta(m)$  requires  $M$  MAC instructions, and an approximation of  $\gamma_t(m', m)$ , which can be implemented as a lookup table [5]. Finally, the computation of the actual state probability  $\lambda_t(m)$  requires another MAC instruction. Therefore the total complexity per time step is at least  $N_{ST} * (M + 1)$  MAC instructions, where  $N_{ST} = 2^v$  and is the number of trellis states. This will be considerably more than for the SOVA, which has an average of around  $3v$ , and a maximum of  $5v$  instructions. The change in complexity due to  $v$  is exponential for the BCJR algorithm, and only linear for the SOVA. Work has also been done on sub-optimal BCJR algorithms with reduced complexity, for example [16]

The memory requirements of the BCJR algorithm are increased over the standard Viterbi Algorithm, with  $\alpha(m)$  and  $\lambda_t(m)$  required to be stored for each time step. However, the memory requirements are comparable to those of the SOVA.



## Chapter 5

---

### DECODING METHOD

The objective of using a MIMO system is to be able to increase the effective user bandwidth. This is achieved by increasing the number of transmit antennas used, and transmitting over them simultaneously, which increases spectral efficiency. Space Time coding has been developed as a means of achieving transmit diversity gain and improving the BER performance of such systems. Due to the inherent decoding complexity of MIMO systems, especially in the frequency selective channel, reduced complexity techniques are required.

The major focus of this thesis has been to develop an effective method to decode STTCs in the frequency selective channel. This is achieved by combining several existing techniques, namely the PVA, SOVA, and STTCs, together in a new and novel way. This chapter shows the changes required to the existing algorithms, so their inputs and outputs are compatible. It also demonstrates the methods by which soft information is calculated, which is used to improve iterative performance.

#### 5.1 SYSTEM OVERVIEW

The method developed in this thesis is similar to that of a turbo equaliser [8] [38]. Figures 5.1 and 5.2 show how the different functional blocks are connected together. The implementation of each block is discussed in further detail in the following sections.

A STTC is effectively used as an outer code of the system. User data is passed through the encoder, which outputs symbols to each transmit antenna. In an ML decoder system, the decoding is done with a combined equaliser and detector. This requires a Viterbi algorithm with a large number of states.

The proposed system uses the PVA as a sub-optimal equaliser, with outputs decoded by a separate STTC decoder. To reduce the effects of error propagation and make the transmitted symbols effectively independent, an interleaver is introduced between the STTC encoder and the channel. At the receiver a de-interleaver is required between the PVA equaliser and the STTC decoder. This interleaver reduces the effect of error propagation which is a common problem in Viterbi based equaliser algorithms.

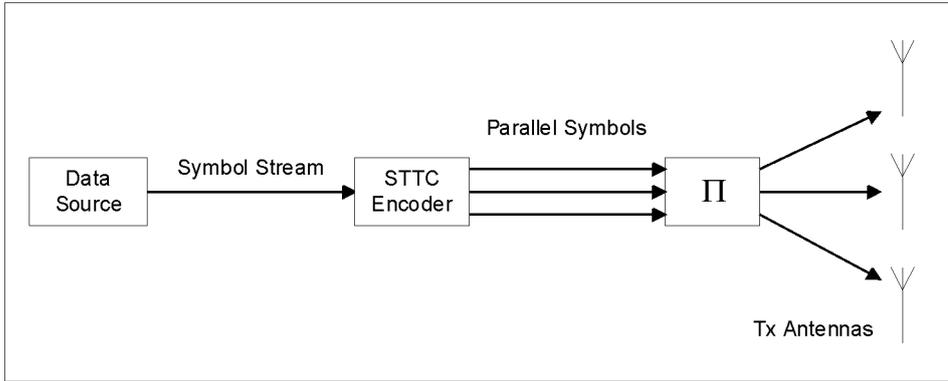


Figure 5.1 Transmitter Structure

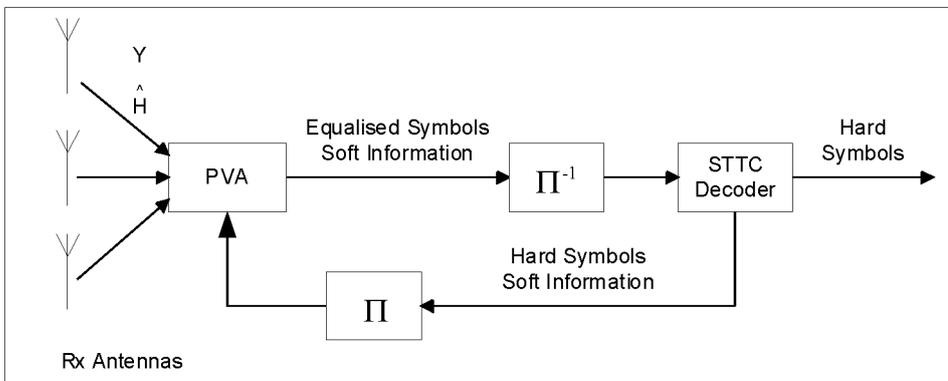


Figure 5.2 Receiver Structure

Because of the separation of equaliser and outer code, an iterative process can be used to improve performance. The outer decoder passes updated symbol estimates and reliability information to the equaliser. To generate reliability information requires the use of a soft output decoder such as the BCJR [5] or soft output Viterbi algorithm (SOVA) [30]. Other possible approaches to improve the performance of the equaliser include the Iterative Viterbi Algorithm, as used in [70]. In this thesis both the PVA equaliser and STTC decoder are based on the SOVA, rather than the standard Viterbi algorithm.

### 5.1.1 Assumptions

To develop a working algorithm a number of assumptions about the system are required. One of the major assumptions is that the system is of a medium size, with  $N_R, N_T \in 2, 3$ . At this stage simulating larger systems is less feasible due to increased complexity. Also other assumptions such as that of uncorrelated channels, perfect synchronisation, etc, may not be accurate in a real system.

Each sub-channel between a transmit and receive antenna is assumed to be a mu-

tually uncorrelated frequency selective fading channel of equal length for all channels. Such channels exist in a rich scattering environment if the antenna spacings are a reasonable number of wavelengths [58] [24].

A block, or packet, encoding structure will be used as such structures are currently used in the mobile wireless industry. For a moderate block length, high transmit frequency and low doppler shift, the channels can be assumed to be quasi-static. This is because the block length is assumed to be much shorter than the channel coherence time. The channel is also assumed to be frequency selective. This is because the system bandwidth is larger than the coherence bandwidth of the channel. It is also assumed that Nyquist pulse shaping is used, with over sampling at the receiver.

The modulations used for the simulations are 4PSK and 8PSK. Most current STTC designs use these modulations. This also limits the complexity of equalisation at the receiver. The use of higher order modulations such as 16QAM leads to significantly more complex trellis based decoders due to the increased number of states required. Such systems would require a form of Reduced State Sequence Estimation (RSSE) [20] or other low complexity equalisation method.

## 5.2 SOVA

The SOVA developed by Hagenauer in [30] was originally designed for BPSK. This means the resulting trellis in the Viterbi algorithm only has two paths entering and exiting every state. The SOVA generates a soft output by tracing back both paths, and updating different symbols with the minimum metric difference.

In both the PVA equaliser and the STTC decoder, the number of paths entering and exiting a state is dependent on  $M$ , the modulation size. To generate an optimum SOVA output, each of these  $M$  paths would have to be traced back. However, in the present work, to reduce complexity of the algorithm, only one alternative path, that with the smallest metric difference was traced back. As only the minimum metric difference is saved for each symbol, the other alternative paths will rarely affect the output. The resulting sub-optimality is small.

The soft output of the SOVA is the difference between two path metrics. This is typically converted to a probability or log likelihood ratio. To reduce complexity of the algorithm, it was decided to directly use the Euclidean path metric difference as the basis of the soft information.

The PVA and STTC decoder metrics have different means. The PVA mean is dependent on the minimum distance between two symbols in the modulation set, and the current SNR. For example with 4PSK and  $E_s = 1$ , the mean distance would be  $\sqrt{2}$ . The STTC decoder metric mean is dependent on the coding gain of the trellis used. This gain is proportional to the trace of the codeword difference matrix as discovered

by Chen *et.al.* [10]. This difference means that the soft information output from the STTC must be scaled before use. To achieve this a simple linear scaling has been used in both processes. The output of the PVA SOVA is multiplied by a parameter labelled  $\alpha$ , and the output of the STTC SOVA is multiplied by a parameter labelled  $\beta$ . In this work the values have been chosen empirically through simulations. However a real system may require adaptive parameters for improved performance.

Using a constant value as a scaling factor has some advantages, and some drawbacks. The method is very simple to implement, as a simple multiplication of the output value. This is faster than having to compute logarithms and exponentials. It also does not require the use of statistical information. However, care must be taken to choose a good scaling factor. If the factor is too large, the output can become numerically unstable. If the factor is too small, performance will degrade, as the soft information is not being used effectively.

These changes to the SOVA reduce the accuracy of the output, but decrease the complexity of the algorithm. There are several changes that could be made to improve performance. This includes extending the trellis used in decoding from a  $M$  path trellis with one step per symbol, to a binary trellis with  $\log_2(M)$  steps per symbol. For example in 8PSK, the normal trellis has eight paths per state, and one step per symbol. This could be changed to a binary ( $M = 2$ ) trellis with three steps, one for each bit.

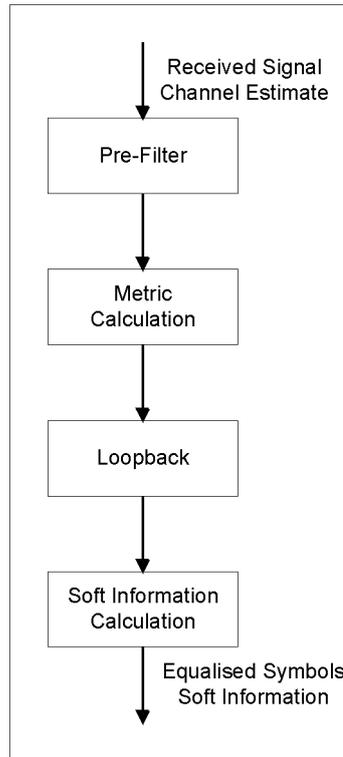
### 5.3 PVA

The PVA algorithm developed by Miller *et.al.* [43] is applicable to several different modulation formats. It was originally simulated using BPSK, but is easily extended in this thesis to 4PSK and 8PSK. The main algorithm is the same, but with a slight modification to use soft inputs, and generate soft outputs. Figure 5.3 shows the steps required to generate the equalised symbols and soft information.

#### 5.3.1 Soft Input

The path metric for the PVA has been slightly altered to allow for soft information input. This input comes from the STTC decoder after the first iteration. In the first pass through the PVA equaliser, no information is available so the standard Euclidean distance metric is used [43]. The information available from the STTC decoder output is a hard symbol estimate, and the distance to the next likely trellis path. This information is used in the following way.

If the STTC decoded symbol matches the current equalised symbol, then it is assumed the equalised symbol is correct, and no adjustment to the metric is required. If the symbols do not match, then the soft information from the STTC is added to the



**Figure 5.3** PVA Flowchart

metric. This means the output path of the PVA decoder is more likely to match the output path of the STTC.

The STTC hard symbol estimates are also used to calculate the  $\phi(k, t)$  term, as shown in Eq. 3.13. This term represents the feedback from the other PVA trellises. In the first iteration, this information is not known, so it has to be estimated from the current received values. In subsequent iterations, the symbols from the STTC decoder are directly used to calculate  $\phi(k, t)$ . This provides a much more accurate branch metric for the PVA equaliser.

### 5.3.2 Soft Output

To calculate soft information on the output symbols, alternative path metrics must be saved. This requires more memory, but allows the calculation of the distance to the next valid path. The basic SOVA is used as part of the PVA equaliser to calculate the soft information. This information is then used in the STTC decoder to improve performance.

#### 5.3.2.1 Soft Position

Another method to generate soft information from the PVA equaliser is by generating a soft symbol estimate rather than a hard symbol decision. This is done using the

metric value for each state  $\lambda^t(\mathbf{i}, \mathbf{j}, k)$ , and the filtered input value. From 3.12, a soft current symbol value  $\hat{a}_{k-n}^t$  can be derived as follows,

$$\lambda^t(\mathbf{i}, \mathbf{j}, k) = \|\delta^t(\mathbf{i}, \mathbf{j}, k)\|^2 \quad (5.1)$$

where

$$\delta^t(\mathbf{i}, \mathbf{j}, k) = \tilde{y}_{k+L_f-1}^t - \sum_{n=0}^{L-1} \tilde{h}_{n+L_f-1}^{t,t} \hat{x}_{k-n}^t - \phi(k, t) \quad (5.2)$$

and  $\phi(k, t)$  is defined in 3.13. Using the approximations  $\tilde{h}_0 \cdots \tilde{h}_{L_f-2} \approx \mathbf{0}$  and  $\tilde{h}_{L_f-1} = \mathbf{I}$  (Section 3.3, [43]) this becomes

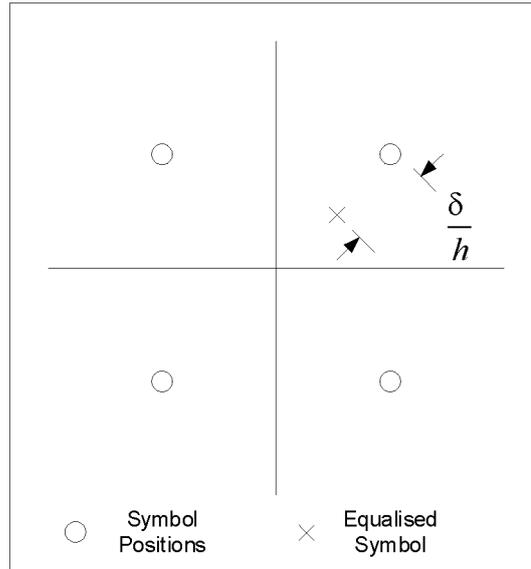
$$\delta^t(\mathbf{i}, \mathbf{j}, k) \approx \sum_{m=1}^{N_T} \sum_{n=0}^{L-1} \tilde{h}_{n+L_f-1}^{t,m} x_{k-n}^t - \sum_{m=1}^{N_T} \sum_{n=0}^{L-1} \tilde{h}_{n+L_f-1}^{t,m} \hat{x}_{k-n}^t + \tilde{n}_k^t \quad (5.3)$$

Assuming correct previous decisions  $x_{k-n}^t = \hat{x}_{k-n}^t$ , ( $n \neq 0$ ), this becomes the Euclidean distance between  $x_{k-n}^t$  and  $\hat{x}_{k-n}^t$ , and

$$\delta^t(\mathbf{i}, \mathbf{j}, k) \approx \tilde{h}_{L_f-1}^{t,t} x_k^t - \tilde{h}_{L_f-1}^{t,t} \hat{x}_k^t + \tilde{n}_k^t \quad (5.4)$$

At high SNR, a soft symbol decision  $\tilde{x}_k^t$  can be calculated as

$$\tilde{x}_k^t = \hat{x}_k^t + \frac{\delta^t(\mathbf{i}, \mathbf{j}, k)}{\tilde{h}_{L_f-1}^{t,t}} \quad (5.5)$$

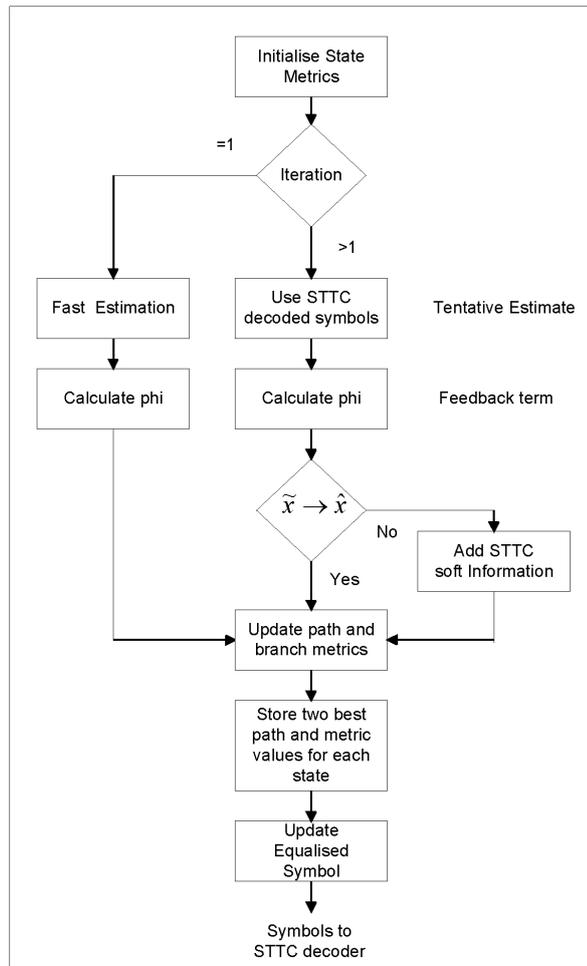


**Figure 5.4** Equalised Symbols

An example for 4-PSK is shown in Fig 5.4. The circles show transmitted symbol positions, and the cross shows a received equalised symbol using  $\delta = \delta^t(\mathbf{i}, \mathbf{j}, k)$  and

$h = \tilde{h}^{t,t}$ . The hard output symbol is generated from the closest symbol to the equalised position.

### 5.3.3 Metric Calculation



**Figure 5.5** PVA metric calculation: Steps required for each time period.

Fig. 5.5 shows the steps required by the PVA equaliser to estimate each transmitted symbol. The steps shown are required for each time period to calculate the branch metrics, which are then used to select the best state, and calculate the equalised symbols.

The metrics for each state have to be initialised. This is important for the first time period, as the initial state should be known. It is typically equal to the all zero state, or the final state of the training sequence if one is used.

The behaviour of the algorithm is slightly different for the first iteration. In this case the symbols from the other transmitters are tentatively estimated using only the branch metric. The branch metrics are calculated following 3.12, but only for the state

with the lowest total metric. The estimated symbol that generates the lowest branch metric is stored as a tentative estimate. These symbols are then used to generate the interference term from the other transmit antennas when computing the branch metrics for all states.

After the first iteration, the tentative estimate is not required. Instead the output from the STTC decoder is used. The STTC also generates soft information. If the current equalised symbol ( $\tilde{x}$ ) does not equal the STTC decoded symbol ( $\hat{x}$ ), a penalty term is added to the branch metric. As the minimum metric is chosen as the equalised symbol, adding a penalty term to an incorrect branch metric makes that branch less likely to be chosen by the PVA equaliser.

## 5.4 INTERLEAVER

The PVA equaliser was originally developed for the multi-user environment. This means it assumes that all transmitted symbol streams are random and independent. However, the use of a STTC encoder at the transmitter means that the symbol streams are actually correlated.

The PVA output is also based on the Viterbi algorithm which typically generates errors in bursts in the presence of noise. This creates error propagation when the output is feed into another Viterbi based algorithm, for example the STTC decoder.

To provide both uncorrelated inputs to the PVA, and to reduce error bursts from its output, an interleaver is employed. To avoid finding the optimum interleaver for a given frame size and STTC, a random interleaver was implemented. As the worst performance occurs when not using an interleaver, nearly all random interleavers over a long length will give good performance. In practice, a good interleaver would be found for a given block length and used throughout the system.

Two different interleaver methods are used, both with a length of one frame ( $N_d = 130$  data symbols plus termination). The 'small' interleaver is a  $1 \times N_d$  random interleaver that is identical for each transmitter. This means the symbols are interleaved over time only. The 'large' interleaver is a  $N_T \times N_d$  random interleaver, that spreads the symbols over both time and space (different antennas). An interleaver is not required for ML detection, as the ML algorithm uses a combined equalisation and detection Viterbi algorithm.

Once the interleaver has been created, a de-interleaver is created as an inverse of the interleaver. Both the transmitter and receiver use the same interleaver.

## 5.5 STTC DESIGNS

The STTCs used in this thesis are the Tarokh [58], Baro [6] and Chen [10] codes. These codes have been chosen as they have been extensively investigated on the flat fading channel. Of interest in the frequency selective channel is investigating which method of constructing codes is superior. In the Tarokh and Baro codes, the rank and diversity criteria were used for construction, while the trace criteria, or Euclidean distance was used by Chen.

Due to the previous assumptions, the trellis codes used are designed for  $N_T = 2, 3$  with 4PSK, and  $N_T = 2$  with 8PSK. The number of states in each trellis depends upon those that have been published in the literature, typically from between 4 and 32 states.

The decoder used for the STTCs has been modified from a standard Viterbi algorithm to the SOVA. The inputs to the algorithm are a soft estimated symbol position, and reliability information from the PVA. The SOVA calculates a hard symbol output, and a metric showing the distance to the next valid path through the trellis. These outputs are fed back to the PVA algorithm as inputs in an iterative process. As the PVA has already dealt with equalising the channel, the metric used for the STTC decoder is

$$\lambda = \sum_{j=1}^{N_T} \|c_k^j - \hat{c}_k^j\|^2 \quad (5.6)$$

where  $\mathbf{c}$  is the hypothesised encoded symbol sequence as defined in Eq. 2.34, and  $\hat{\mathbf{c}}$  is the estimated sequence.

## 5.6 ITERATION

The PVA equaliser estimates each transmitted sequence separately. The signal from each transmitter is treated as a separate interferer that requires estimation to remove co-channel interference. In the first iteration, the estimate is generated using a fast approximation, and is not an encoded sequence. In subsequent iterations, the STTC output is used.

The STTC decoder produces an estimate of the data sequence. This data sequence can be re-encoded into a transmitted symbol sequence, with  $N_T$  transmitted symbols per data symbol. The estimated transmitted symbols are re-used by the PVA to improve performance of the equaliser.

Both the STTC and PVA also generate soft information. Iterating allows this soft information output to be improved. However, the main benefit of iterating is better symbol estimates for equaliser interference suppression

## 5.7 MAXIMUM LIKELIHOOD

The optimum method of detecting the MIMO system is to use a joint equaliser and decoder. This requires a large Viterbi algorithm, and is computationally very complex. The output of the algorithm is the most likely transmitted data sequence (MLSE). It also requires knowledge of the channel, and in a real system would require a pre-filter to limit the effects of a real channel.

In an uncoded system the ML detector would be required to pick the best estimate of all possible symbol combinations at each time period. This would require a trellis with  $M^{N_T(L-1)}$  states, which currently could not be decoded feasibly with a Viterbi based Algorithm. However using a STTC introduces a correlation into the transmitted symbol sequence. This means the ML detector can be used to jointly estimate and equalise the data sequence.

The resulting joint trellis requires  $M^{(L-1)}$  states to equalise the channel history, and the number of states in the STTC to estimate the data. As the ML detector uses a joint trellis, the overall number of states is calculated by multiplying the two together. For example an 8 state STTC, using 4PSK, with  $L = 3$  and  $N_T = 2$ , requires 8 states for estimation, and  $4^2 = 16$  states for equalisation. The combined trellis requires  $8 * 16 = 128$  states. This is much better than the uncoded requirement of  $4^{2*2} = 256$  state trellis. However, the uncoded system produces  $N_T$  data symbols per time period, compared to 1 for ML. Equivalently, the proposed iterative approach would require an 8 state STTC decoder, and two  $4^2 = 16$  state equalisers for a total of 40 states.

The branch metric in the trellis is similar to that defined in Eq. 2.41. However, the MIMO system has been oversampled to stop aliasing. To enable fair comparison between both the iterative approach and ML, the ML system is also oversampled by a factor of  $N_s$ . This gives a final metric of

$$\begin{aligned} \lambda &= \sum_{i=1}^{N_R N_s} \|y_k^i - \hat{y}_k^i\|^2 \\ &= \sum_{i=1}^{N_R N_s} \|y_k^i - \sum_{l=0}^{L-1} \sum_{j=1}^{N_T} h_l^{i,j} x_{k-l}^j\|^2 \end{aligned} \quad (5.7)$$

The ML detector requires the correlation produced by encoding to reduce the number of states required in the trellis. Therefore it does not use an interleaver, which would remove the correlation. As it achieves MLSE, it also does not require iteration. In its current form it only produces hard outputs. However it could be easily modified to use the SOVA, similarly to the PVA and STTC decoder. This would only be required if an outer code was used on the overall system, to improve the BER performance.

## 5.8 COMPLEXITY

### 5.8.1 PVA

The overall complexity of the proposed iterative approach can be calculated using Sections 3.6 and 4.2. It can also be compared to ML decoding, as calculated in Section 2.3.2. However some modifications have to be made to fit the MIMO system parameters.

For the iterative approach, the complexity of the PVA equaliser is similar to that discussed in Section 3.6. However, a soft output is required for the STTC decoder. This requires one complex MAC per state. Also, an interleaver and STTC decoder have been introduced. The interleaver will only add  $N_T$  operations per time period, so is negligible. However, the STTC decoder will have  $N_{ST}$  states, with  $M$  branches per state. The calculation of each metric involves  $N_T + 1$  complex MAC instructions.

The use of the SOVA in both the PVA equaliser and STTC decoder will also increase the complexity. The length of the memory,  $v$ , the PVA is the same as  $L - 1$ , the length of the channel. The length of  $v$  in the STTC decoder is dependent on  $N_{ST}$ , where  $v = \log_2 N_{ST}$ . Also, the cost of adding the SOVA input is one complex MAC per path. The worst-case scenario trace-back length of  $5v$  is used to determine the added complexity. This gives a total cost of

$$5 * (L - 1) + M^{L-1} * M \quad (5.8)$$

for the PVA equaliser, and

$$5 * v + N_{ST} * M \quad (5.9)$$

for the STTC decoder.

### 5.8.2 ML

The only difference between the current MIMO system, and that discussed in Section 2.3.2 is the use of oversampling. Oversampling increases the complexity of calculating each path metric by a factor of  $N_s$ , the oversampling rate. As a result Eq. 2.42 becomes

$$N_{ST} * M^{L-1} * M * (N_T * L + 1) * N_R * N_s = N_{ST} M^L (N_T L + 1) N_R N_s$$

### 5.8.3 Comparison

To give an example of the complexity difference between the PVA and ML, a few examples are given. The number of operations is approximate, and based on the equations given. By using caching, or lookup tables that are calculated once per frame of data, a reasonable reduction in complexity could be achieved. The system uses the

following parameters  $N_T = 2$ ,  $N_R = 2$ ,  $N_s=2$ ,  $L_f = 9$ ,  $L = 3$ ,  $N_d = 130$ . The PVA overhead has been averaged over  $N_d$  symbols. It is assumed the channel estimates have already been calculated. The iterative approach shows the complexity for one iteration, in further iterations, the overhead is not required.

**Table 5.1** ML decoding complexity per symbol

Modulation	4PSK	4PSK	8PSK	8PSK
STTC	$N_{ST} = 4$	$N_{ST} = 16$	$N_{ST} = 8$	$N_{ST} = 32$
Computation Total	7 168	28 672	114 688	458 752

**Table 5.2** Iterative decoding complexity per symbol

Modulation	4PSK	4PSK	8PSK	8PSK
STTC	$N_{ST} = 4$	$N_{ST} = 16$	$N_{ST} = 8$	$N_{ST} = 32$
Cholesky	60	60	60	60
Filter H	3	3	3	3
Filter Y	72	72	72	72
Overhead Total	135	135	135	135
Tentative Guess	28	28	56	56
PVA Trellis	564	564	4 196	4 196
PVA Soft Output	4	16	8	32
PVA SOVA	74	74	522	522
PVA Total	670 * 2	682 * 2	4 782 * 2	4 806 * 2
STTC Trellis	48	192	192	768
STTC SOVA	26	104	94	306
STTC Total	74	296	286	1074
Computation Total	1 549	1 795	9 985	10 821

As shown in Tables 5.1 and 5.2, the iterative approach is far less complex than ML decoding, even if 3 or 4 iterations are required. It also shows that more complex STTCs with a larger number of states are more practical to implement in the iterative scheme. This is because the equalisation makes up around 90% of the overall complexity. As the equaliser and decoder are separate, an increase in complexity of the decoder makes little difference to the overall complexity. This means codes with better gain can be used in the iterative approach, with an overall complexity and performance very similar, or better than ML decoding. For example the 32 state 8PSK code with iterative decoding is less complex to decode than the 8 state 8PSK code with ML decoding. The 32 state iterative performance is also superior to 8 state ML decoding, as shown in section 6.2.

## 5.9 SIMULATION ENVIRONMENT

The system described has been tested on a simulation platform. The simulation software has been designed in C. It models the baseband performance of the PVA iterative approach, and also models the ML approach for comparison.

There are a number of assumptions that have been made when designing the simulation. The main focus of this research is to compare the PVA approach to ML, and to investigate which STTC design criterion provides better performance. Therefore, it is assumed that the receiver is perfectly synchronised with the transmitted signal. Other factors not taken into consideration are modulation/demodulation to a carrier frequency, channel estimation. Complex baseband was used throughout the work.

### 5.9.1 Transmitter Model

The transmitter is modelled as an information source that generates independent symbols. These symbols are encoded using a STTC and sent over  $N_T$  antennas. The symbols are chosen from a modulation set with an average transmitted energy  $E_s = 1$ .

As the system being considered is frequency selective, pulse shaping is assumed. A raised cosine pulse with a  $\beta$  roll off factor of 0.99, similar to that in [43] is used. It is assumed that both the transmitter and receiver use filters with a root raised cosine shape. Therefore, the pulse shaping is modelled as part of the channel.

### 5.9.2 Channel Model

The MIMO channel is modelled as a set of independent frequency selective fading channels between each transmit and receive antenna pair. Each channel is modelled using the tapped delay line model. The power delay profile is assumed to be flat, with each channel tap being an independent complex Gaussian variable with variance  $\frac{1}{2}$  per dimension.

The length of the channel support is  $L_c = 2$ . Due to the pulse shaping, an over-sample rate of  $N_s = 2$  samples per symbol is used. The length of the impulse response is  $L_u = 1$ , which corresponds to pulse coefficients of  $u = \{0.41, 0.82, 0.41\}$ , where the total energy in the pulse  $E_p = \sum \|u\|^2 = 1$ . This means the overall channel length is  $L = 3$ . With over-sampling each independent channel is composed of 6 taps, generated by the convolution of the generated channel taps and pulse shape.

The output of the channel is the convolution of the transmitted symbols and the overall channel response. Due to all antennas transmitting simultaneously, the input at each receive antenna is the superposition of all transmitted symbols.

### 5.9.3 Receiver Model

The receiver models the baseband performance of the system. In this research, it is assumed perfect channel knowledge is available at the receive antenna. However it would be relatively easy to add in channel estimation similar to [43], or another method. In the model independent AWGN is added at each receive antenna, as it is assumed to be due to front-end electrical noise.

The SNR is defined as the received  $\frac{E_b}{N_0}$  per receive antenna. The noise Power Spectral Density (PSD),  $N_0$  is calculated using this for a fair comparison. The received energy at each receive antenna per time slot can be found using

$$E_s = \sigma_s^2 * \sigma_c^2 * N_s * L_c * \sum \|u\|^2 * N_T, \quad (5.10)$$

where  $\sigma_s^2 = 1$  is the transmit symbol power and  $\sigma_c^1$  is the total channel tap power.

The received energy per bit is defined as  $E_s / \log_2(M)$ . From this the noise PSD  $N_0$  can be defined as

$$N_0 = \frac{E_b}{10^{\frac{SNR}{10}}}. \quad (5.11)$$

This definition allow comparisons between different numbers of transmit antennas, and modulation formats. The  $N_s$  and  $L_c$  factors in Eq. 5.10 are defined in the previous section. However, care must be taken when comparing systems with different numbers of receive antennas. When the number of receive antennas is increased, more energy is captured, requiring a lower SNR per antenna for good performance.

## Chapter 6

---

### RESULTS

The main focus of this research has been to develop an approach that can be used to decode STTCs in the frequency selective channel, with lower decoding complexity than Maximum Likelihood (ML) decoding. Other objectives include investigating the best design criteria for STTCs, and what can be done to improve performance of the algorithm.

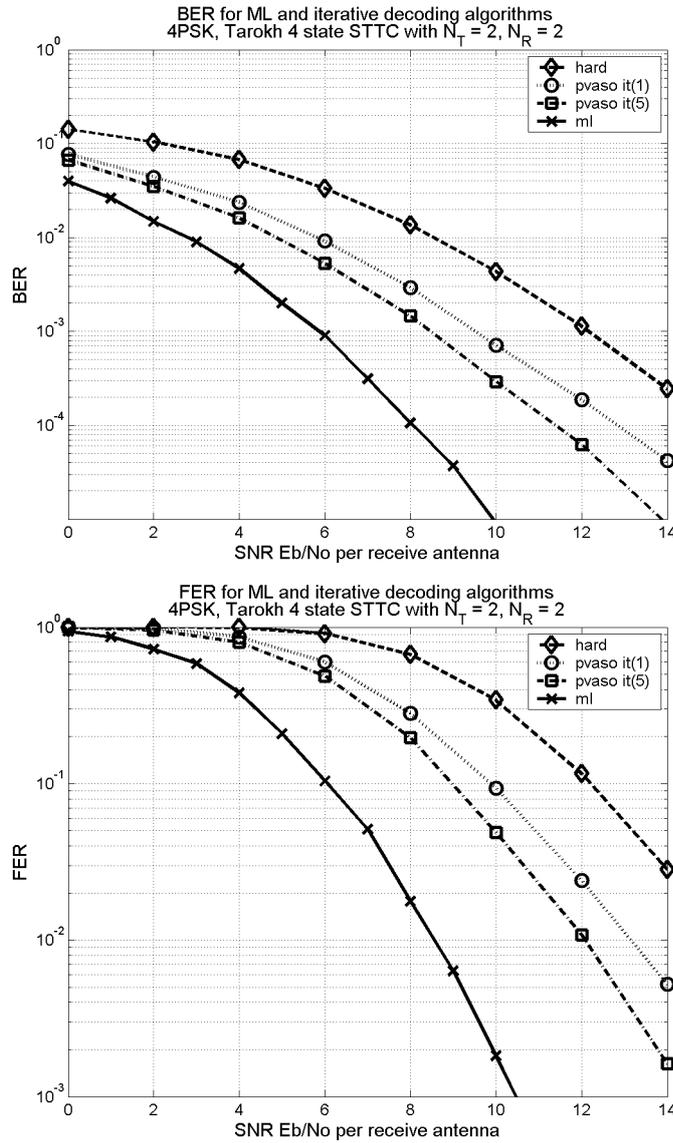
The following results investigate the performance of the iterative algorithm using 4PSK, and 8PSK. The effect of various features of the algorithm are also shown, with the performance gain they achieve. The results have been obtained after running the simulations until 400 frames are in error after the first iteration. The simulator settings are those shown in Chapter 5, unless otherwise stated. Performance comparisons are made at a FER of  $10^{-2}$  (approx BER of  $10^{-4}$ ).

#### 6.1 4PSK

This section compares the Tarokh (TSC) [59], Baro (BBH) [6] and Chen (CYV) [10] Space Time Trellis Codes for 4-PSK. The codes in the literature typically vary from 4 state to 32 state codes.

The focus of the simulations is to compare Maximum Likelihood decoding, denoted "ml" in the plots, to the combined PVA iterative algorithm. There are three results shown from the iterative algorithm. The first is from a hard output PVA algorithm (without the SOVA), decoded by the STTC decoder with no iterating and is denoted "hard" in the plots. The second is from a soft output PVA algorithm decoded by the STTC decoder after the first iteration, denoted "pvaso it(1)". The final plot is the soft output PVA algorithm combined with the soft output STTC decoder. This is shown after the fifth iteration, and denoted "pvaso it(5)". The plots show the Bit Error Rate (BER) and Frame Error Rate (FER) of these algorithms.

The simulations use a  $N_T = 2, N_R = 2$  MIMO system, with  $\alpha = 0.65, \beta = 0.1$  and 5 iterations.  $\alpha$  and  $\beta$  have been empirically chosen to give good performance, and there is very little improvement in performance beyond 5 iterations.

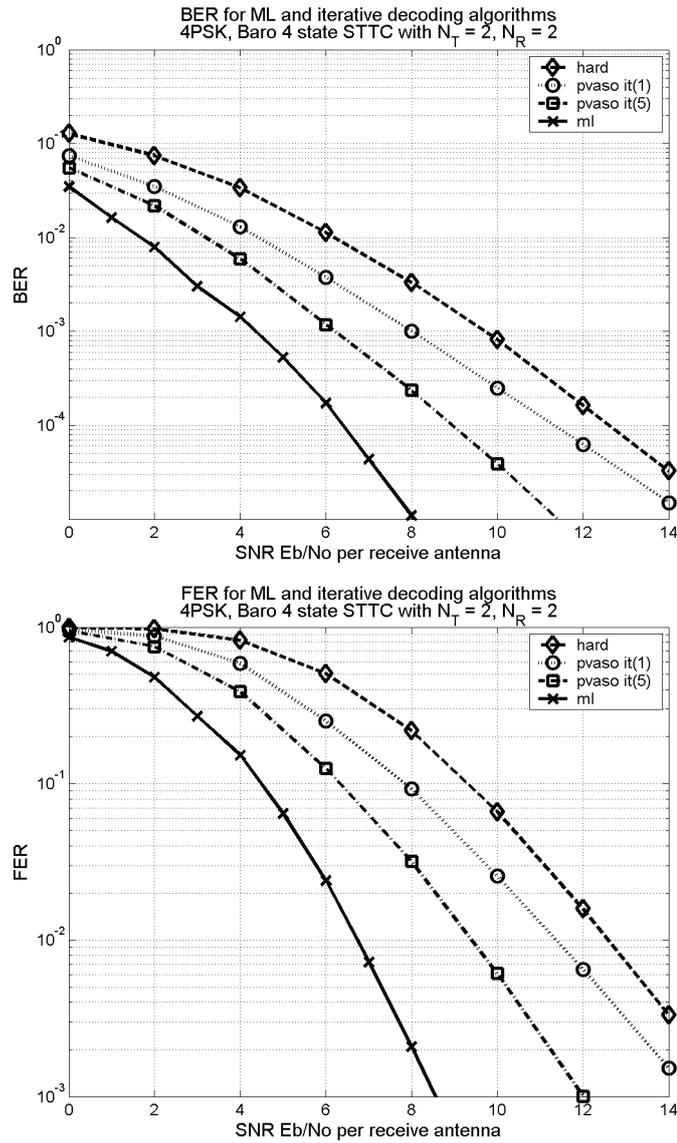


**Figure 6.1** 4 State Tarokh STTC in a  $N_T = 2$ ,  $N_R = 2$  MIMO system (a) BER (b) FER

### 6.1.1 4 state STTCs

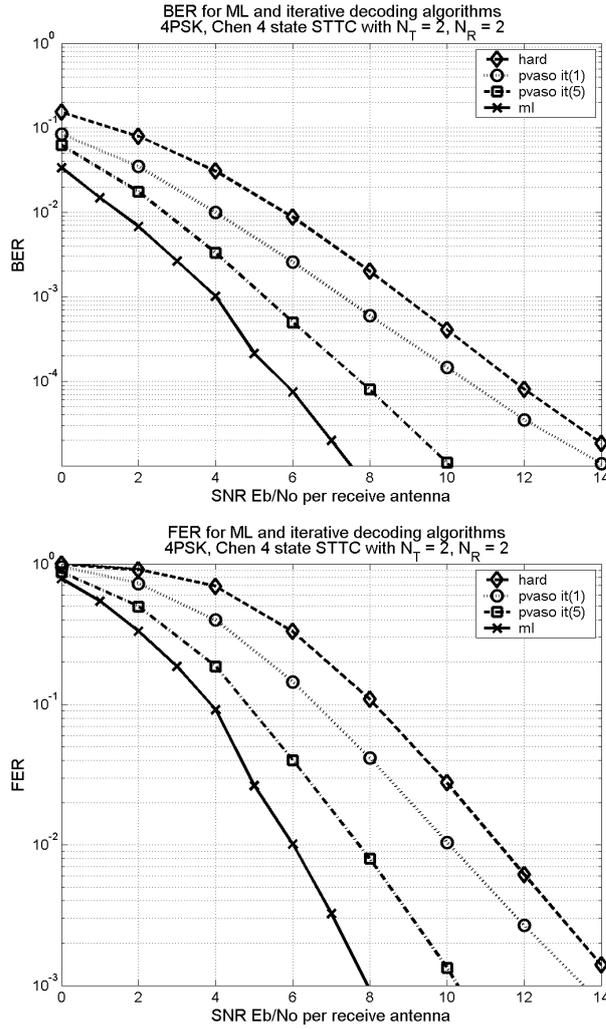
The 4 state STTC is the smallest possible when using 4PSK. Fig. 6.1 shows the results for the Tarokh code, Fig. 6.2 for the Baro code, and Fig. 6.3 for the Chen code.

The original Tarokh code has the worst performance of the three codes. It is based on the rank and determinant criteria, but has not been optimised for coding gain. Of interest to note is the iterative procedure helps comparatively little, having a gain of around 1.2dB at a FER of  $10^{-2}$  between iteration 1 and 5. The final iteration FER is also reasonably far away from ML decoding, by around 3.5dB, but does give a 3.0dB gain over hard decoding. Similar results are shown for the BER.



**Figure 6.2** 4 State Baro STTC in a  $N_T = 2$ ,  $N_R = 2$  MIMO system (a) BER (b) FER

The Baro code performs much better than the Tarokh code, even though it is also based on the rank and determinant criteria. This improvement is due to the increased coding gain of the STTC, and is shown in both the ML and iterative decoding. The iterative procedure has a larger gain of around 2.0dB at a FER of  $10^{-2}$  between iteration 1 and 5, which is much more significant. The final iteration FER is within 2.5dB of ML decoding, and gives a 3.0dB gain over hard decoding.



**Figure 6.3** 4 State Chen STTC in a  $N_T = 2$ ,  $N_R = 2$  MIMO system (a) BER (b) FER

The Chen code performs the best, which implies that the trace design criteria [10] is more important in frequency selective fading channels, as well as flat fading channels. This appears to be due to a better coding gain, rather than a difference in diversity. The Chen code performs 2.2dB and 0.7dB better than the Tarokh and Baro codes respectively, with ML decoding at a FER of  $10^{-2}$ . The iterative performance also shows a strong improvement, having a gain of 4.0dB and 1.5dB over the respective codes after the final (5th) iteration. The final iteration FER is also within 2.0dB of ML decoding. It also appears that the 1st iteration and hard decoding results are much closer together, especially compared to the Tarokh codes.

It is also apparent that the combined PVA and STTC algorithm suffers a small diversity loss compared to ML decoding for all three codes. This is expected due to the PVA being sub-optimal, and is shown by the slightly different slopes of the BER and FER curves between "ml" and "pvaso it(5)".

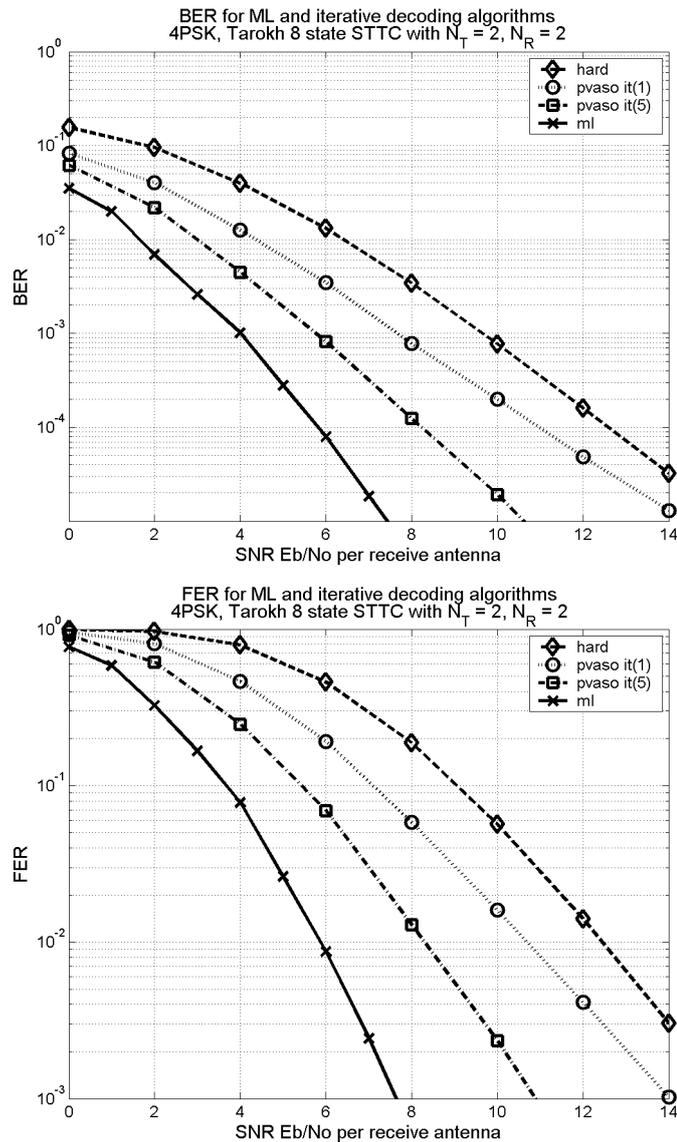


Figure 6.4 8 State Tarokh STTC in a  $N_T = 2, N_R = 2$  MIMO system (a) BER (b) FER

### 6.1.2 8 state STTCs

This section compares 8 state STTCs. Fig. 6.4 shows the results for the Tarokh code, Fig. 6.5 for the Baro code, and Fig. 6.6 for the Chen code.

For the 8 state codes, the Tarokh and Baro codes perform almost identically. This is not surprising, as Table 2.1 shows they have identical determinant and trace. Both codes also significantly outperform their 4 state counterparts. The Baro code has a very slight (0.1dB) ML performance gain over the Tarokh code, but there appears to be no significant difference between the iterative performance of either code. The iterative performance remains around 2.5dB from ML, and has a 1.8dB and 3.7dB gain over iteration 1 and hard decoding respectively, at a FER of  $10^{-2}$ .

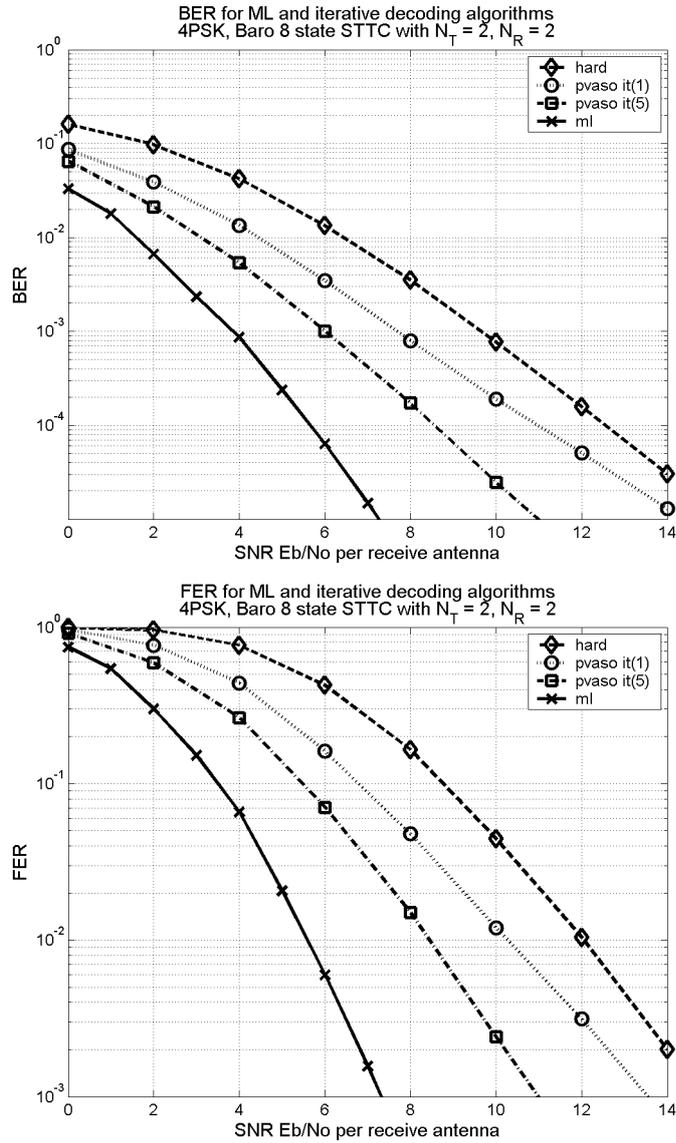
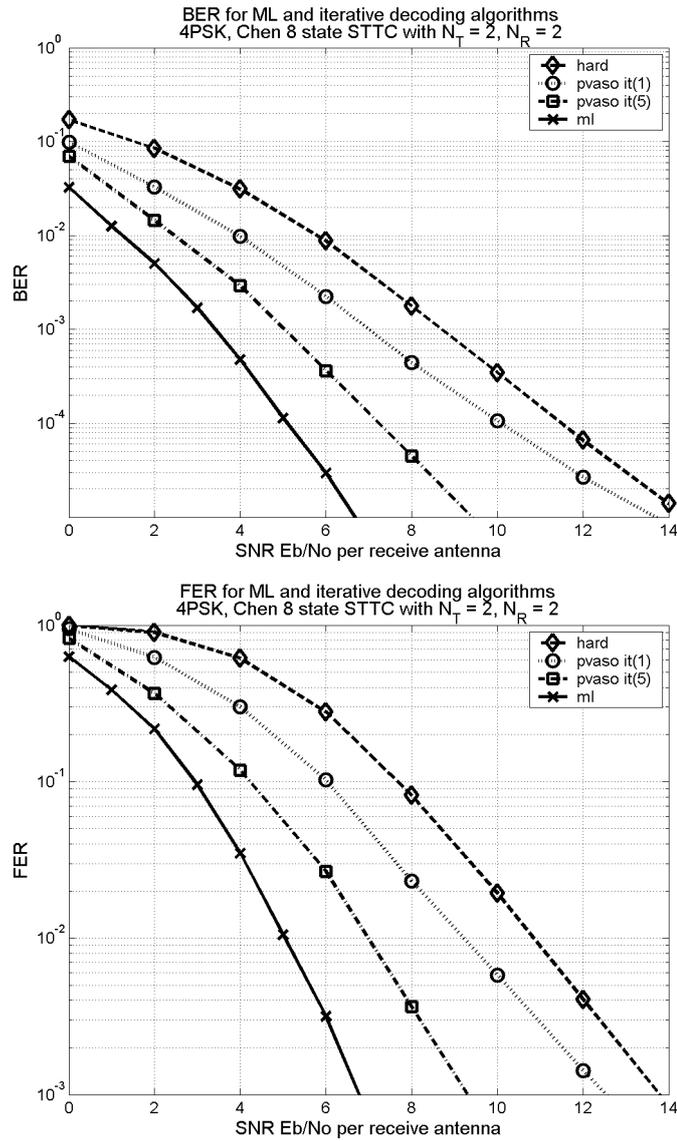


Figure 6.5 8 State Baro STTC in a  $N_T = 2$ ,  $N_R = 2$  MIMO system (a) BER (b) FER

Another interesting result is that the 4 state Chen code outperforms both the Tarokh and Baro 8 state codes. While the ML performance is very similar, the soft iterative procedure has a gain of around 0.3dB. The 4 state Chen code actually has a higher trace, but lower determinant than both 8 state codes.



**Figure 6.6** 8 State Chen STTC in a  $N_T = 2$ ,  $N_R = 2$  MIMO system (a) BER (b) FER

Again, the 8 state Chen code performs better than the other 8 state codes, but with less of a margin. The Chen code has a gain of 0.5dB at a FER of  $10^{-2}$  compared to both the other codes when using ML decoding. The iterative performance also shows a further improvement, having a gain of 1.5dB over the same codes after the final iteration. The final iteration FER is approximately within 2.0dB of ML decoding.

The 8 state codes have a reasonable gain over their respective 4 state codes, varying from around 0.5dB to 2.5dB. For ML decoding, going from 4 to 8 states doubles the size of the resulting combined equaliser and decoder trellis, while only having a minor effect on the complexity of the PVA algorithm.

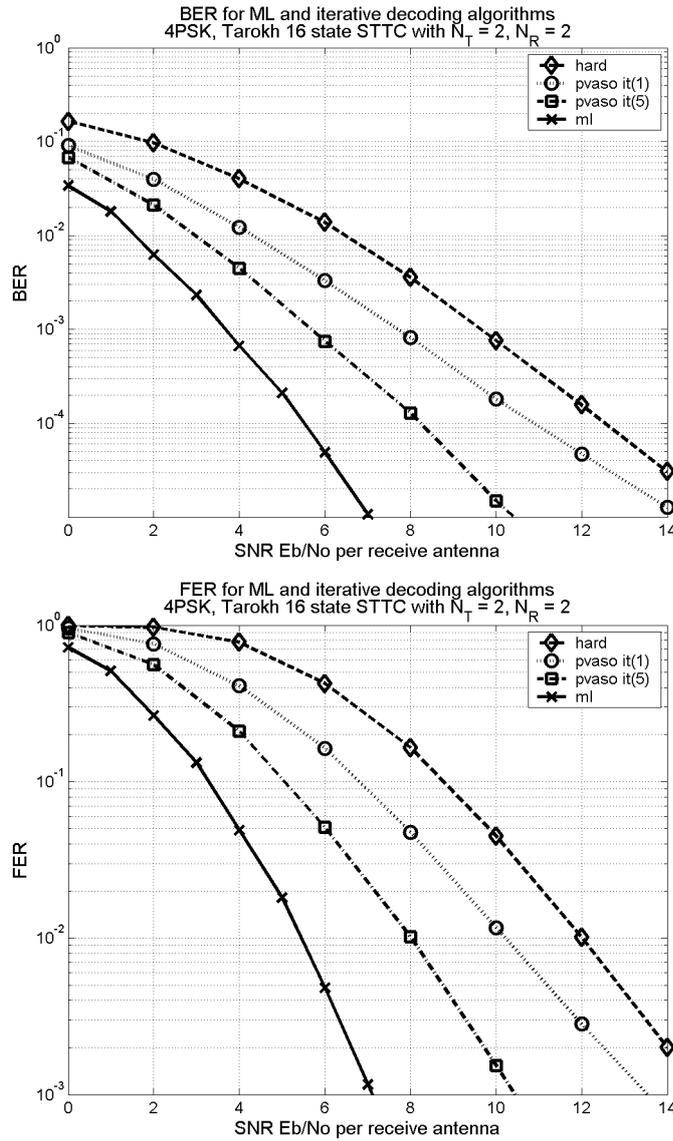
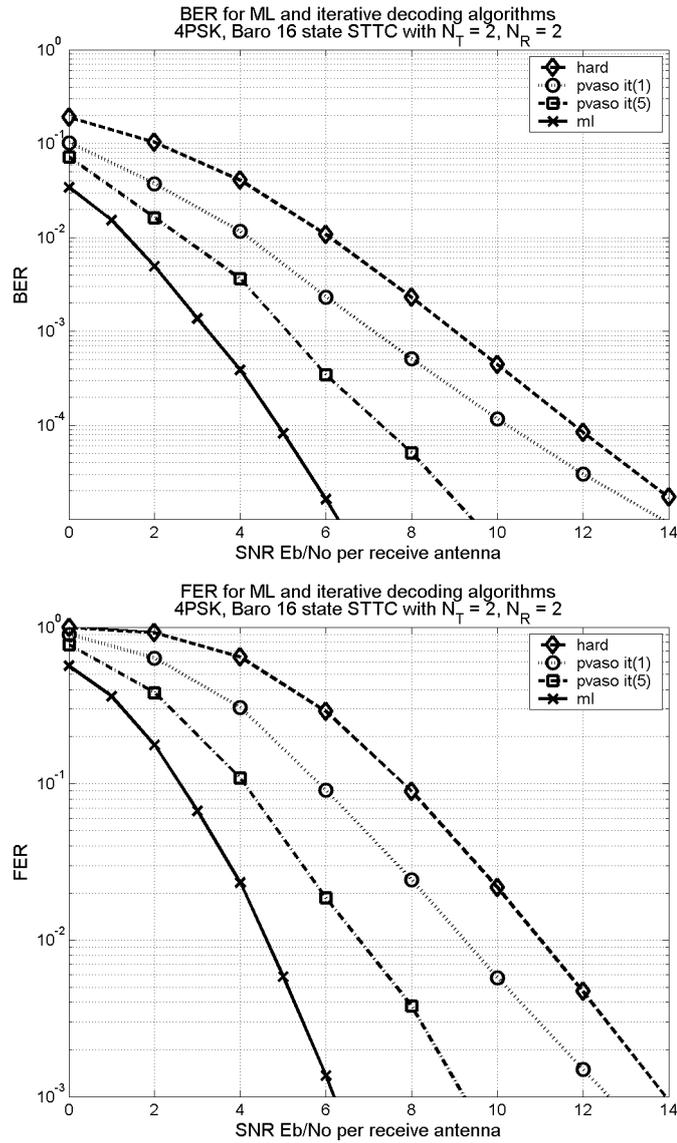


Figure 6.7 16 State Tarokh STTC in a  $N_T = 2, N_R = 2$  MIMO system (a) BER (b) FER

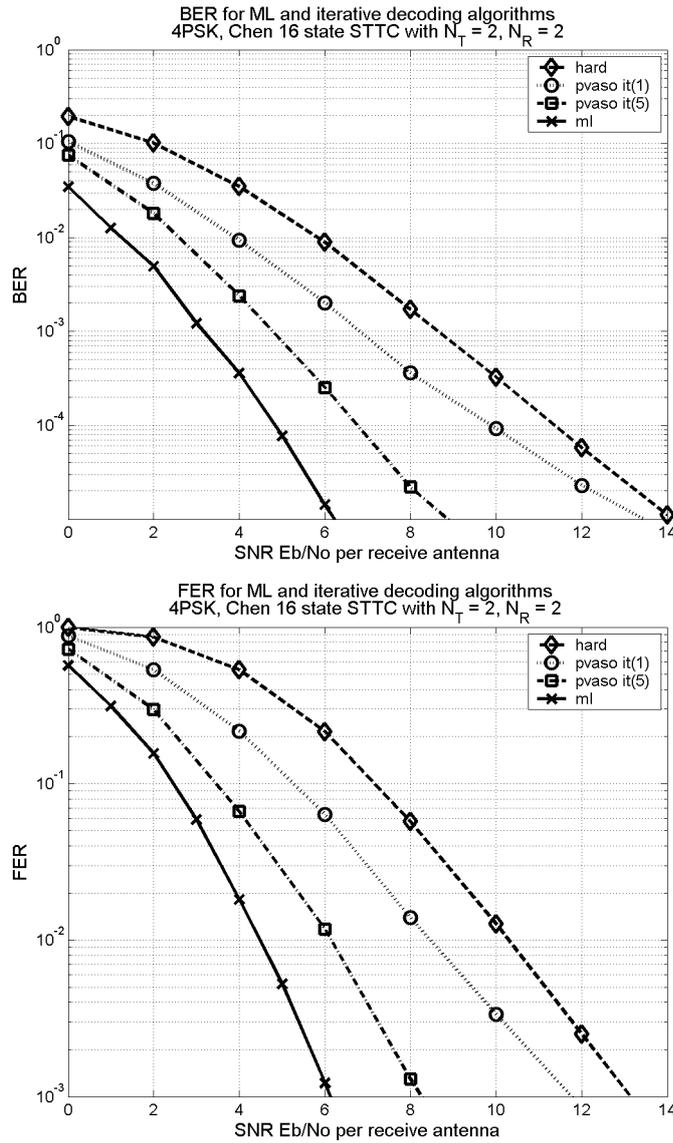
### 6.1.3 16 state STTCs

A comparison of 16 state STTCs also shows similar results. The Tarokh code shown in Fig. 6.7 shows the worst performance, for both ML and PVA decoding. The iterative performance remains just over 2.5dB away from ML decoding, and offers a smaller gain of 2.0dB over non-iterative performance at a FER of  $10^{-2}$ . Even the 4 state Chen code provides better iterative performance. The Tarokh code also has an ML performance loss of around 0.8dB compared to the other 16 state codes.



**Figure 6.8** 16 State Baro STTC in a  $N_T = 2, N_R = 2$  MIMO system (a) BER (b) FER

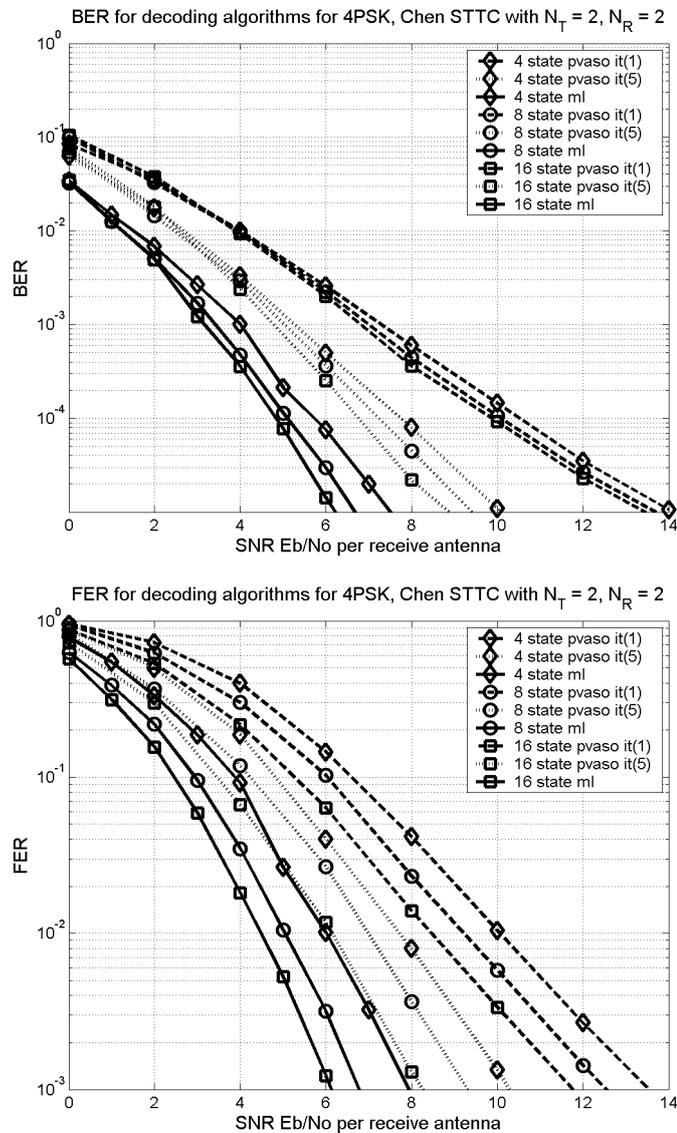
The ML performance for the Baro and Chen codes, shown in Figs. 6.8 and 6.9 respectively, is almost identical. The main difference is in the PVA decoding procedure, where the Chen code has a 0.5dB gain at a FER of  $10^{-2}$ . Again, the iterative decoding loss of the Chen code compared to ML, is 1.6dB, compared to 2.2dB for the Baro code.



**Figure 6.9** 16 State Chen STTC in a  $N_T = 2, N_R = 2$  MIMO system (a) BER (b) FER

The increase to 16 states improves the Baro code the most, with an ML decoding gain of 0.8dB over its respective 8 state code, much larger than the 0.3dB gains of the Tarokh and Chen codes. The iterative gain increases even more, being 1.0dB better than the Baro 8 state code. However when comparing the iterative procedure, the 8 state Chen code still gives better performance than the 16 state Tarokh code, and is almost the same as the 16 state Baro code.

## 6.1.4 Chen STTCs



**Figure 6.10** Chen STTCs with varying number of states, in a  $N_T = 2, N_R = 2$  MIMO system (a) BER (b) FER

When using a STTC, there is a tradeoff between the number of states and the coding gain. This is clearly shown in Fig. 6.10, which shows the performance of several different STTCs. To make a valid comparison, all the codes have been designed using the trace criteria. The groups of curves shown are first iteration (dashed line), final iteration (dotted line) and ML decoding (solid line) in order of increasing performance. Diamond markers (right) show the 4 state code, circles (centre) show the 8 state code, and squares (left) show the 16 state code.

There is a gain of around 1.0dB when going from a 4 to 8 state STTC, and a further 0.5dB gain using a 16 state code. These gains are fairly constant for both ML

and iterative decoding, but the FER performance shows slightly improved performance compared to the BER. This means that while the number of frames in error is decreasing, the number of errors per frame is actually increasing. While this appears to be counter-intuitive, the memory of the larger STTCs is longer, and any decoding error will likely result in a longer burst of errors when decoded.

For the ML decoding algorithm, complexity is directly proportional to the total number of states. This means the 16 state code is 4 times more complex to decode than the 4 state code. For the iterative procedure, the PVA equalisation complexity is unchanged, but the STTC decoder becomes more complex. However, as the PVA algorithm is typically more complex than the STTC decoder (see Section 5.8.3), the overall increase in complexity is minor. This means it is relatively more beneficial to increase the number of states, and hence performance in the iterative procedure compared to ML decoding.

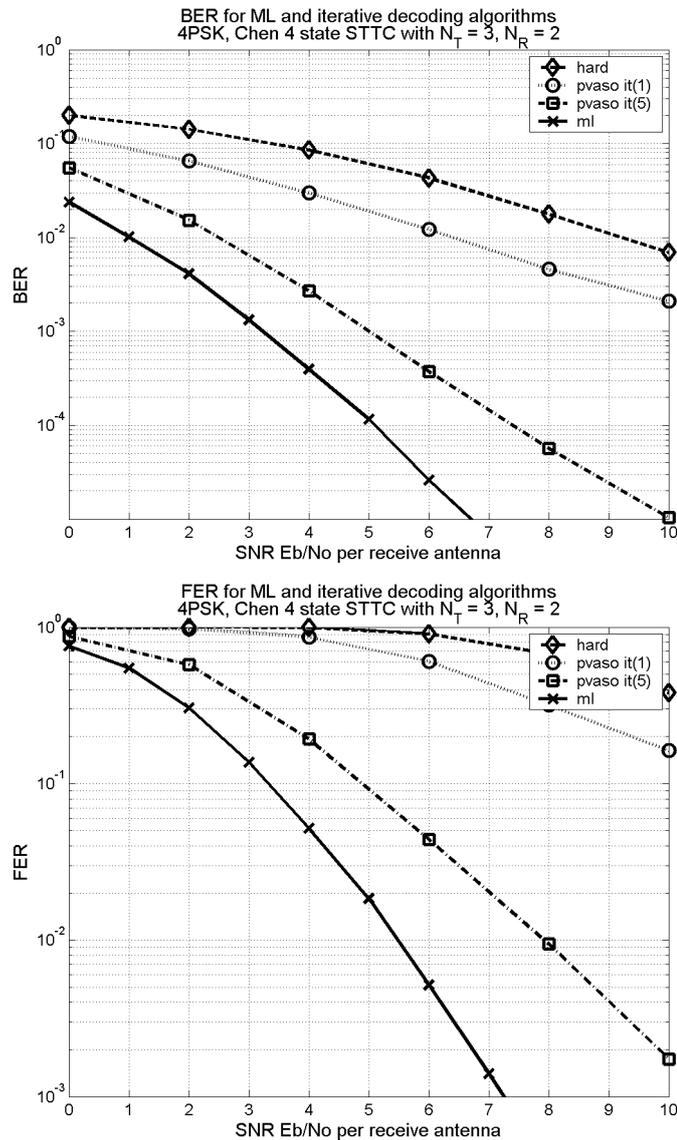


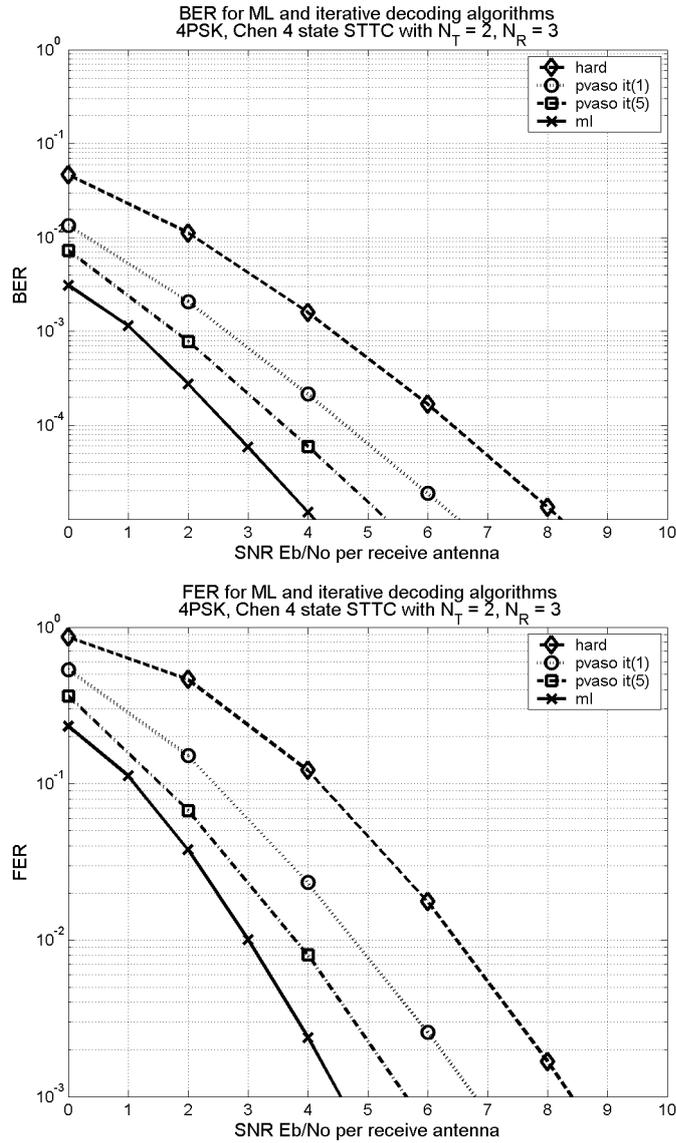
Figure 6.11 4 State Chen STTC in a  $N_T = 3$ ,  $N_R = 2$  MIMO system (a) BER (b) FER

### 6.1.5 Different MIMO systems

STTCs have also been developed for larger MIMO systems. In [12] and [11] Chen *et.al.* develop 4PSK and 8PSK codes for  $N_T = 3, 4$  systems. Such codes offer increased coding gain for ML decoding, due to the increased degrees of freedom. The codes do not offer any rate increase, with the rate staying at one data symbol per time period.

In the iterative procedure, each transmit antenna is treated as a separate interferer. This means that the PVA equaliser performance decreases as the number of transmit antennas increases. However, the drop in performance can be improved by the extra coding gain achieved by the STTC decoder.

The PVA equaliser also suffers a performance loss if the system is under determined,

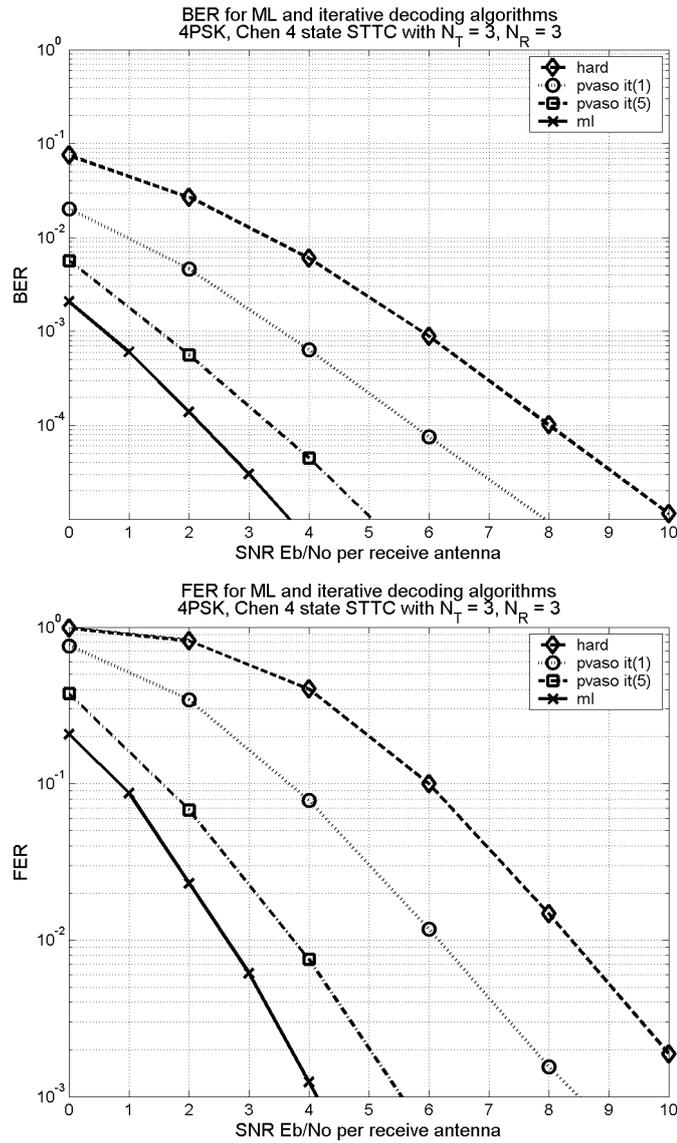


**Figure 6.12** 4 State Chen STTC in a  $N_T = 2$ ,  $N_R = 3$  MIMO system (a) BER (b) FER

i.e.  $N_T > N_R$ . This means the assumption of independent white input noise is no longer valid (ref section 3 etc), which causes a large loss in performance.

The results of using an under-determined  $N_R = 2$  system are shown in Figs. 6.3 and 6.11. Increasing the number of transmit antennas from two to three results in a small (around 1.0dB) gain in ML decoding. When using the PVA algorithm without iterating, the performance is extremely poor, as shown in Fig 6.11. While the iterative procedure with  $N_T = 3$  performs as well as  $N_T = 2$ , it is more complex, and has no rate or performance benefit.

Similar results are shown in an example system that is not under-determined. Adding an extra transmit antenna only affects the performance of the ML algorithm. Figs 6.12 and 6.13 demonstrate this increased coding gain for a  $N_R = 3$  system. In-



**Figure 6.13** 4 State Chen STTC in a  $N_T = 3$ ,  $N_R = 3$  MIMO system (a) BER (b) FER

creasing the number of transmit antennas from 2 to 3 results in a small (0.5dB) gain in ML decoding, but only a negligible (0.1dB) gain in iterative decoding. When using the PVA algorithm without iterating, the performance is again degraded, due to the extra interference. The equalisation loss is again offset by the STTC gain. As there is no rate change by increasing the number of transmit antennas, it would be more beneficial to increase the number of states in the code.

## 6.2 8PSK

The results when using 8PSK are a little more limited than that for 4PSK. This is due to optimum STTC coefficients for the rank and determinant criteria being published in [6] for 4PSK only. However, both Tarokh and Chen published coefficients for  $N_T = 2$  using 8PSK. This allows a general comparison between using the trace and determinant criteria, and also to look at the difference in performance in using 4PSK or 8PSK.

Using 8PSK has an advantage over 4PSK, as it offers greater spectral efficiency. However, for equal energy constraints, the distance between codewords decreases, which results in an expected performance loss. The results follow the same format as discussed in the previous section. The simulations use a  $N_T = 2, N_R = 2$  MIMO system, with  $\alpha = 0.2, \beta = 0.2$  and a maximum of 5 iterations as the iterative parameters. The  $\alpha$  and  $\beta$  parameters are again empirically chosen. There is some sensitivity with respect to performance of these parameters, as high values of  $\alpha$  can cause the iterative procedure to be unstable.

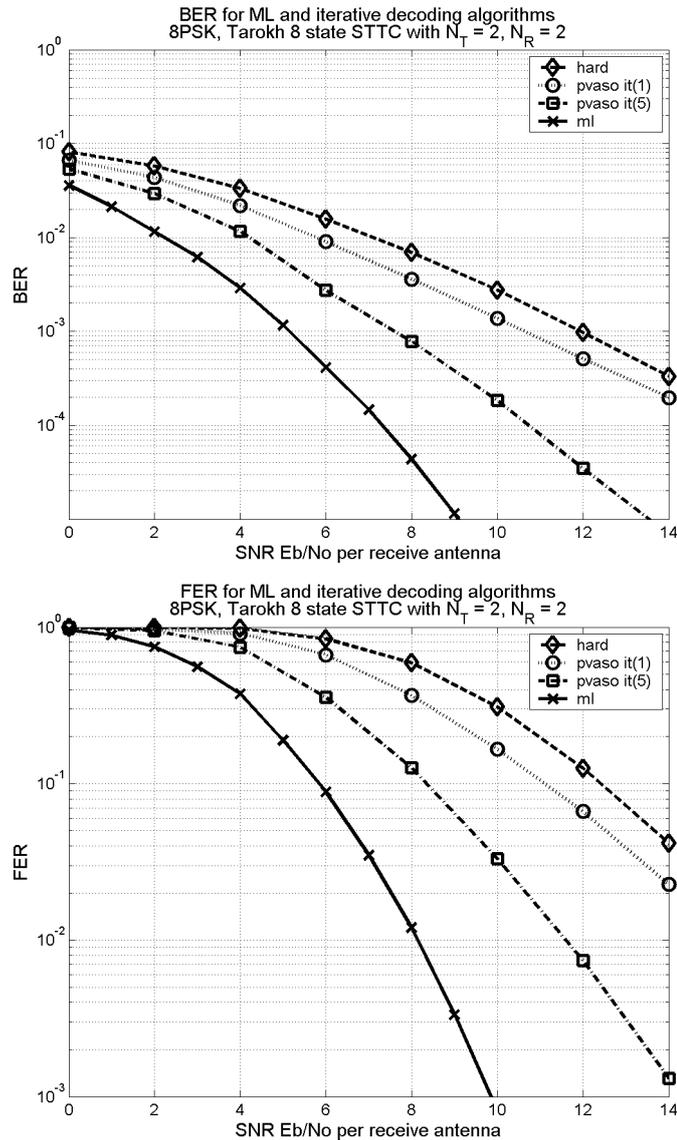
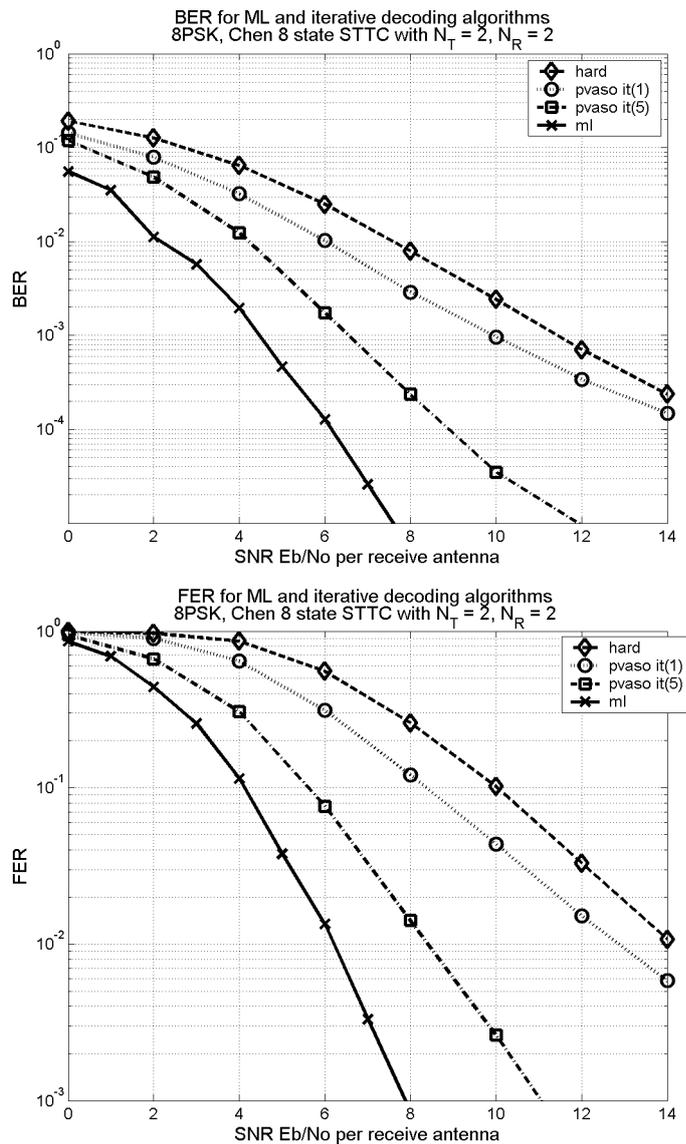


Figure 6.14 8 State Tarokh STTC in a  $N_T = 2, N_R = 2$  MIMO system (a) BER (b) FER

### 6.2.1 8 state STTCs

The initial results of using 8PSK appear very similar to the results obtained when using 4PSK. The 8 state code is the simplest code designed, and has a low coding gain. Its performance is similar to that of the 4 state 4PSK code, but considerably worse than the 8 state 4PSK code. This performance loss is expected, and is due to the loss in transmitted energy per bit. However, using 8PSK provides 50% greater throughput than 4PSK. Fig. 6.14 shows a good gain using the soft iterative procedure of around 4.0dB over the non-iterative approach. However, it is around 3.7dB from ML decoding, and suffers a noticeable diversity loss.



**Figure 6.15** 8 State Chen STTC in a  $N_T = 2$ ,  $N_R = 2$  MIMO system (a) BER (b) FER

In comparing the performance of the code, it can be seen that the Chen code has a large ML gain (2.0dB) over the Tarokh code. This is a significant gain. Also of interest, is that the relative performance of the PVA iterative algorithm increases, being around 2.2dB from ML. The iterative procedure also has better diversity than the non-iterative procedure. The ML performance is very similar to the 4 state 4PSK Chen code, and as expected due to energy loss, it is around 1.5dB from the 8 state 4PSK code.

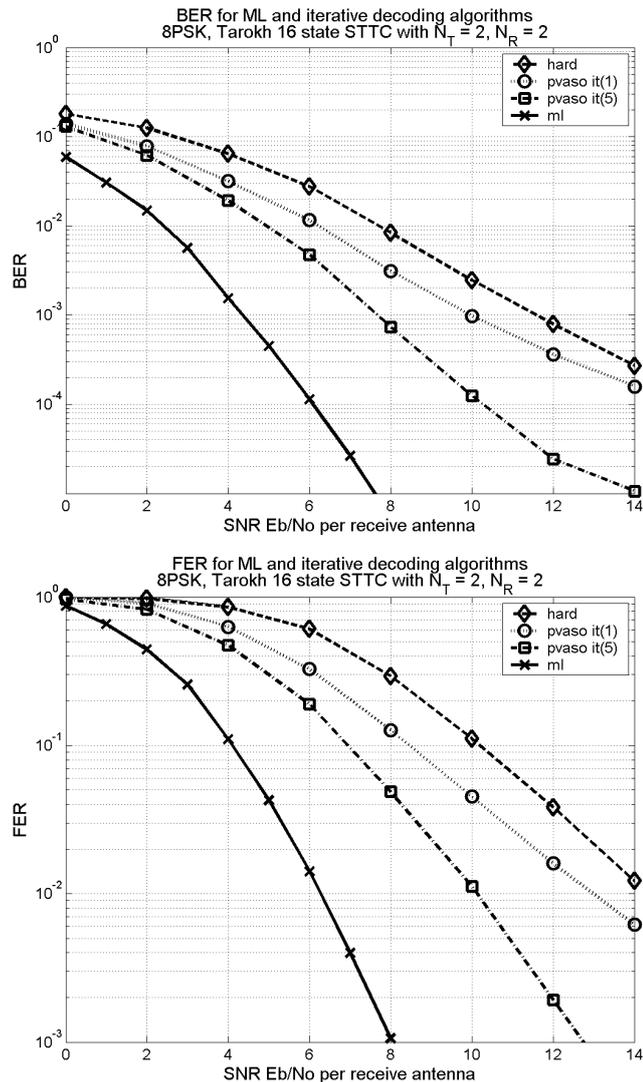
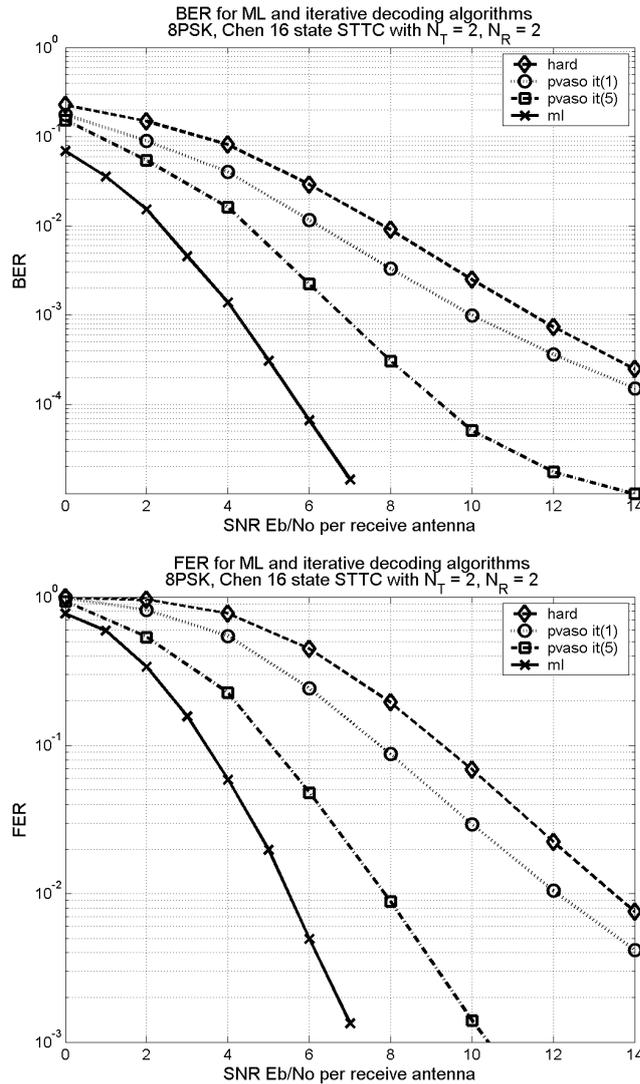


Figure 6.16 16 State Tarokh STTC in a  $N_T = 2$ ,  $N_R = 2$  MIMO system (a) BER (b) FER

### 6.2.2 16 state STTCs

The Tarokh 16 state code performs far better than the 8 state code. It shows an ML decoding gain of around 1.8dB, and a similar iterative decoding gain of 1.7dB. However, the final iterative performance is still around 3.8dB from ML decoding, but is a 3.0dB improvement over the first iteration.

An interesting phenomena is noticeable in the 16 and 32 state Chen and Tarokh codes. The slope of the FER curve in Fig. 6.16 is constant at high SNR. However, the BER curve is starting to show a performance degradation at high SNR. This means that if a frame is in error, it is nearly completely destroyed. This is also shown in Fig. 6.26 (b), where the number of symbol errors in a frame increases dramatically at high SNR. It is unknown whether this is caused by correlated channels, or poor performance of the PVA prefilter. It may also be due to poor choices of  $\alpha$  and  $\beta$ . As



**Figure 6.17** 16 State Chen STTC in a  $N_T = 2, N_R = 2$  MIMO system (a) BER (b) FER

the FER performance is still improving, it may be that most of the frames with a small number of errors are corrected, leaving only frames with a large number errors.

The 16 state Chen code gives only a very small gain (0.5dB) over the 8 state code. This is not very good considering the doubling in complexity for ML decoding. The relative performance of the PVA iterative algorithm is also very similar. The final iterative performance is around 2.5dB from ML.

Of interest to note is that the iterative performance is very similar to that of the 4 state 4PSK Chen code shown in Fig. 6.3. This means that the bandwidth efficiency could be increased from 2bit/s/Hz to 3bit/s/Hz with very similar performance, albeit with a large increase in complexity (see Section 5.8.3).

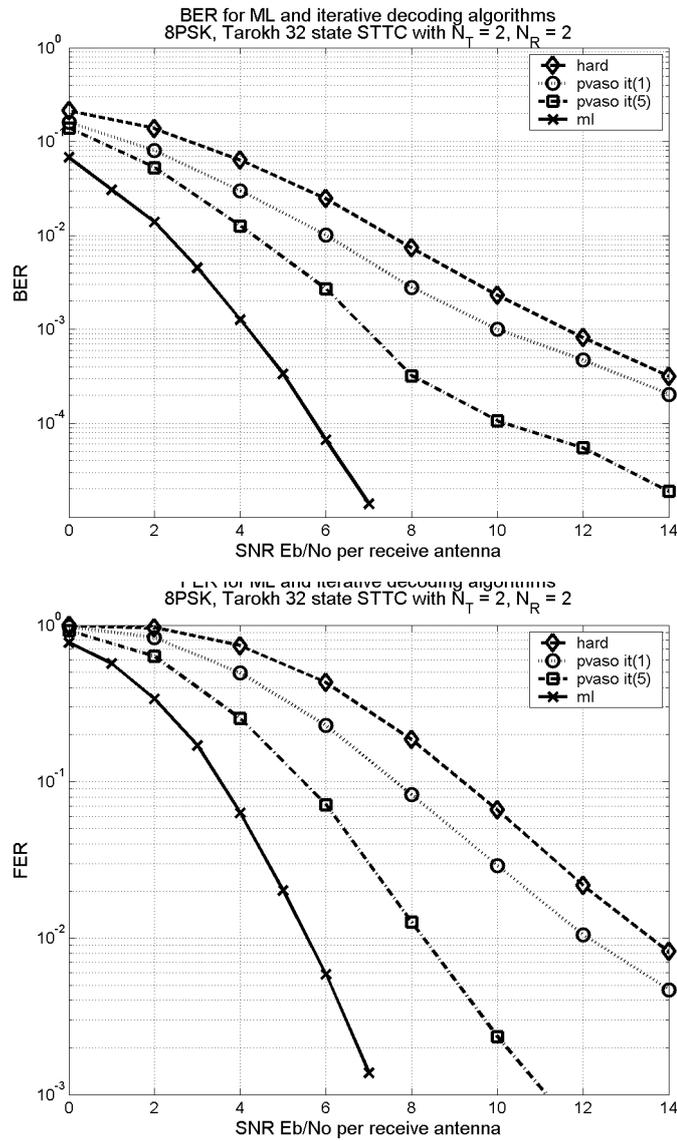
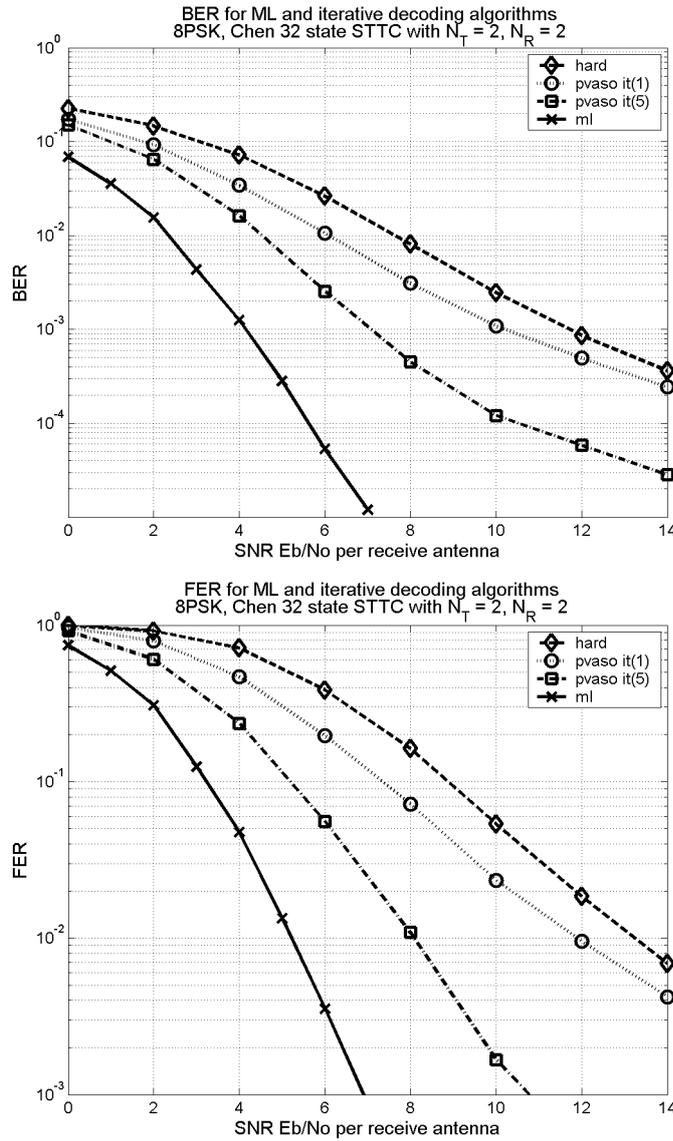


Figure 6.18 32 State Tarokh STTC (a) BER (b) FER

### 6.2.3 32 state STTCs

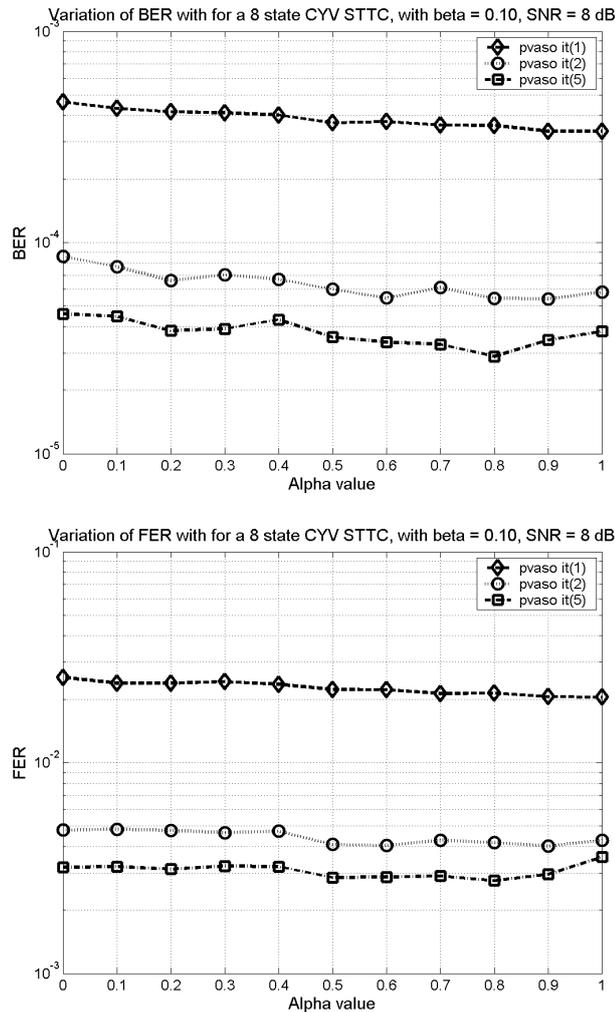
Compared to both the 8 and 16 state Tarokh codes, the 32 state code performs significantly better. Its performance is equivalent to the 16 state 8PSK Chen code. Also, the iterative algorithm approaches to within 3.0dB of ML performance, which is good compared to the poor iterative performance of the 8 state Tarokh code.



**Figure 6.19** 32 State Chen STTC in a  $N_T = 2$ ,  $N_R = 2$  MIMO system (a) BER (b) FER

Again, the 32 state Chen code gives a very small gain ( $< 0.5\text{dB}$ ) over the 16 state Chen code. Also, it gives almost exactly the same performance as the 32 state Tarokh code. This is quite unusual, as in all the other codes, the trace criteria outperform the determinant criteria by around  $1.0\text{dB}$ .

It is also apparent that all of the 8PSK codes appear to suffer from a BER decoding performance degradation at high SNR. Interestingly, this does not seem to occur to the FER, even at lower FER (around  $10^{-4}$ , not shown). This phenomena is further discussed in Section 6.3.6.



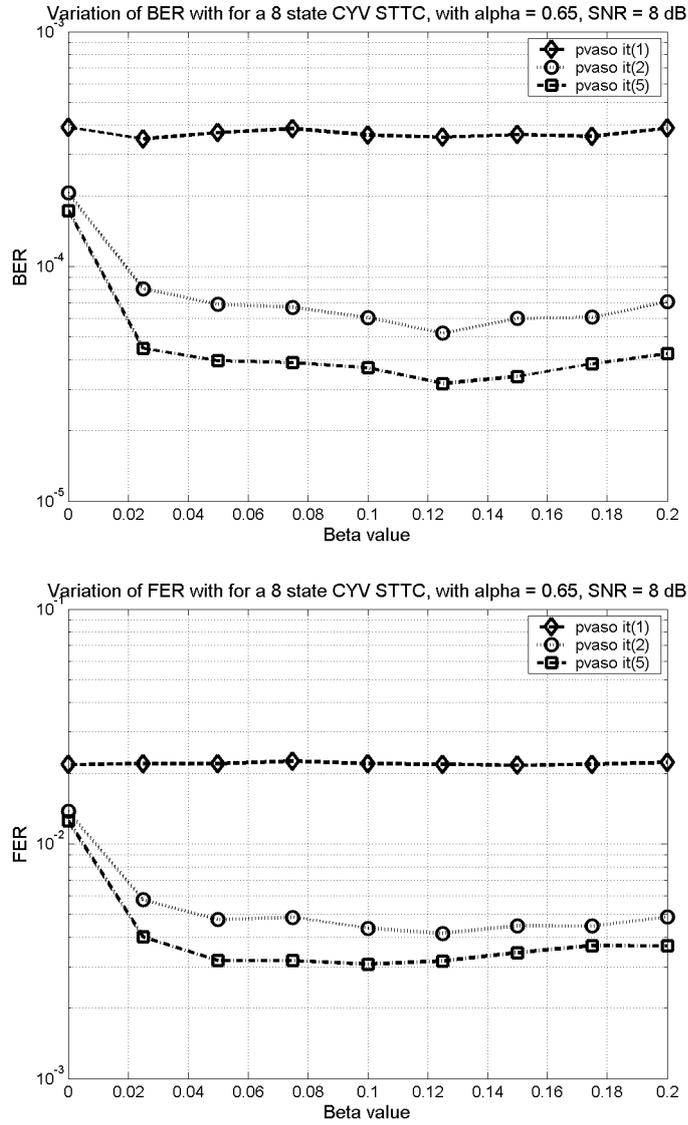
**Figure 6.20** Effects of alpha parameter. 8 state Chen 4PSK STTC in a  $N_T = 2, N_R = 2$  MIMO system (a) BER (b) FER

### 6.3 PARAMETER EFFECTS

As the PVA iterative algorithm is sub-optimal, various parameters will have an effect on its performance. The main parameters used are soft information scaling with alpha and beta, the number of iterations done, the type of interleaver used, and soft equalisation.

#### 6.3.1 Alpha

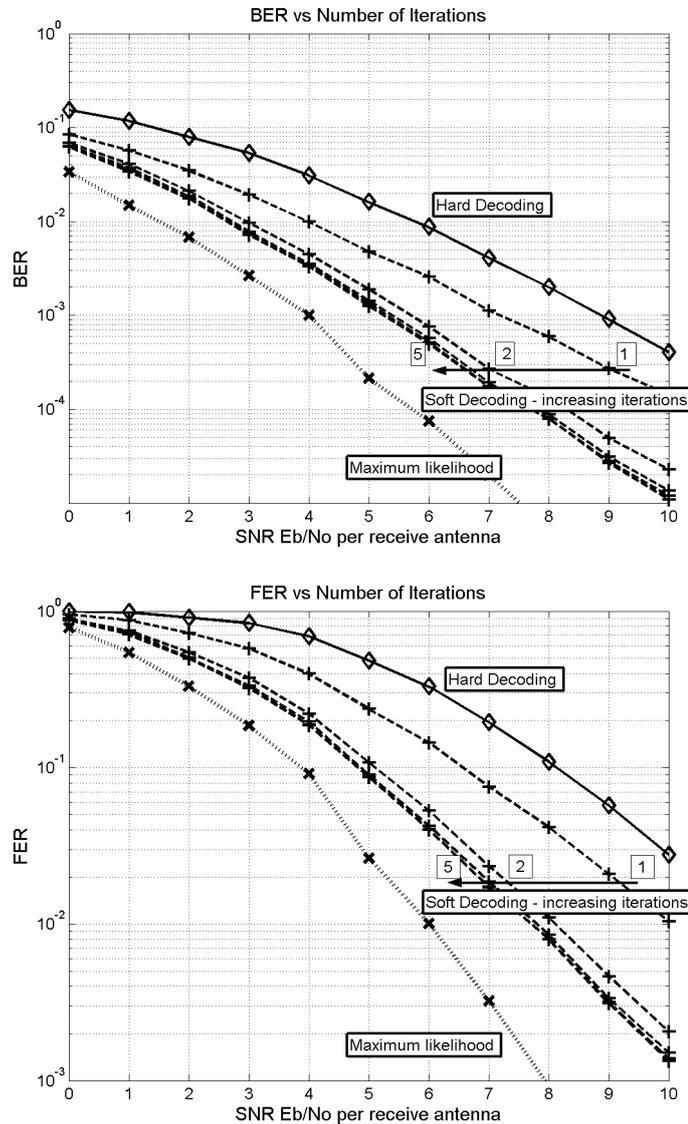
The alpha parameter is used to scale the soft information from the PVA algorithm. It directly multiplies the SOVA output, which in turn is added to the metric in the STTC decoder. Fig. 6.20 shows the change in performance as alpha is varied. When  $\alpha = 0$ , the soft information is not calculated as part of the PVA. It can be noticed that while the SOVA does improve the performance slightly, it is not very significant. A value of 0.65 has been chosen for most of the other results shown when using 4PSK.



**Figure 6.21** Effects of beta parameter. 8 state Chen 4PSK STTC in a  $N_T = 2, N_R = 2$  MIMO system (a) BER (b) FER

### 6.3.2 Beta

Similarly, the beta parameter is used to scale the soft information output from the STTC decoder algorithm. Fig. 6.21 shows the change in performance as beta is varied. In comparison to the alpha parameter, the beta parameter has a very large effect on system performance. When  $\beta = 0$ , the soft information is not calculated as part of the STTC decoder. The gain of the system when using the STTC SOVA output is around 1.3dB. A value of 0.1 has been chosen for most of the other results shown when using 4PSK.

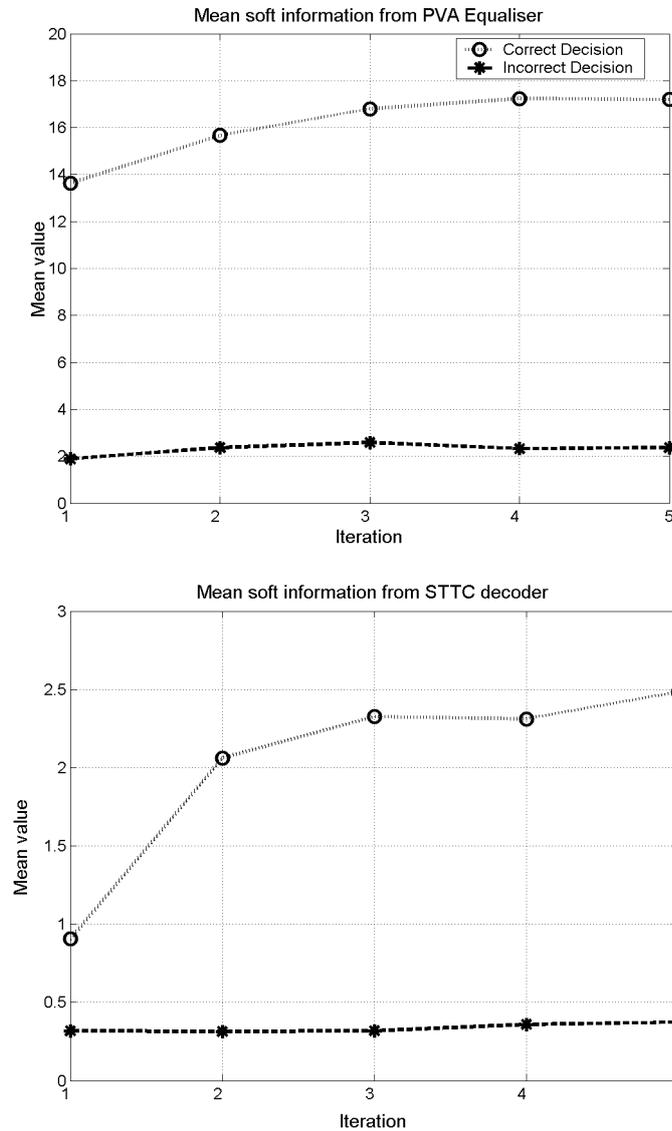


**Figure 6.22** Effects of iteration on performance. 4 state Chen 4PSK STTC in a  $N_T = 2, N_R = 2$  MIMO system (a) BER (b) FER

### 6.3.3 Number of Iterations

When using the PVA equaliser an iterative procedure is used, which is similar to turbo equalisation. On the first pass through (iteration 1), the PVA has no initial symbol estimates to use, and performs poorly. However, using the soft information provides a large performance increase over "Hard Decoding". On subsequent iterations, the STTC decoder can provide good initial symbol estimates to the PVA equaliser, as well as soft information. It is shown that performing further iterations is important performance wise.

It can be seen from Fig. 6.22 that doing a second iteration results in a 2.0dB gain. Further iterations increase the performance by another 0.5dB. As a lower number

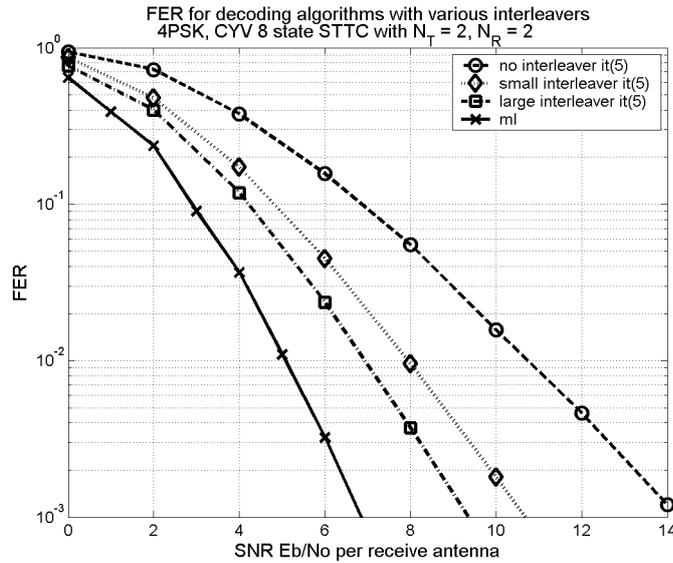


**Figure 6.23** Effects of iteration on soft information. 8 state Chen 4PSK STTC in a  $N_T = 2, N_R = 2$  MIMO system. (a) PVA Equaliser (b) STTC Decoder

of iterations is more computationally efficient, it appears that a maximum of three iterations should be performed. This may need to be increased for under-determined systems, or when using larger modulation sets.

Iterating also affects the accuracy of the soft information from the SOVA algorithm. Fig. 6.23 shows the change in the median output of the SOVA for the PVA equaliser and STTC decoder. The graphs show the difference in the information when making a correct, or wrong decision. The output of the SOVA will tend to be large when a correct decision is made, as the distance to a competing path in the trellis is also large. It is shown that the difference in metrics between correct and wrong decisions increases as the number of iterations increases, which shows that algorithm is behaving correctly.

Fig. 6.23 also shows the difference in magnitude between the two SOVA algorithms. The metric for the PVA equaliser is dependent on the channel, and hence has different values than the STTC decoder. It can be seen that there is a large increase in the mean of the STTC soft information with increasing iterations. However, there is only a small increase in the mean of the PVA soft information. This also shows that the STTC soft information is more important than the PVA soft information, as there is a larger gain in information content.



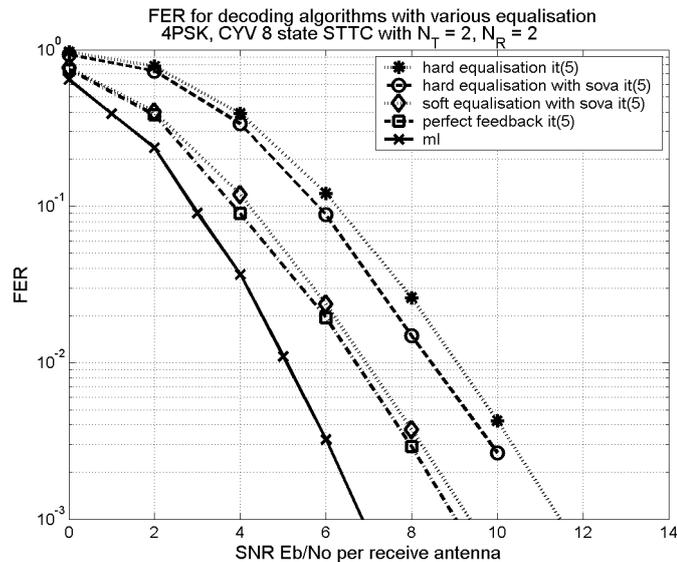
**Figure 6.24** Effects of interleaver on performance. 8 state Chen 4PSK STTC in a  $N_T = 2, N_R = 2$  MIMO system.

### 6.3.4 Interleaver gain

To get good performance, an interleaver is required between the PVA equaliser, and the STTC decoder. Both the equaliser and decoder are based on the Viterbi algorithm, which can produce bursts of errors in the output sequence in the presence of noise. To reduce the error propagation between the two algorithms, a random interleaver is used to spread the errors over the entire frame of symbols.

Several different interleaver methods are used, 'none', 'small' and 'large'. The 'small' interleaver is a  $1 \times N_d$  random interleaver that is identical for each transmitter. This means the symbols are interleaved over time only. The 'large' interleaver is a  $N_T \times N_d$  random interleaver, that spreads the symbols over both time and space (different antennas). An interleaver is not required for ML detection, as the ML algorithm uses a combined equalisation and detection Viterbi algorithm.

The choice of interleaver type is critical to system performance. Fig. 6.24 shows that using the 'large' interleaver gives a 1.0dB gain over the 'small' interleaver, and a 3.5db gain over not using an interleaver. These gains are significant, and easy to achieve. It should also be possible to use a smaller size interleaver than that currently used, as long as the length used is moderately larger than the memory of the STTC used. Also, for simulation purposes, each frame uses a different (random) interleaver. An optimum interleaver for a given length could be found, which may slightly improve performance.



**Figure 6.25** Effects of soft equalisation on performance. 8 state Chen 4PSK STTC in a  $N_T = 2$ ,  $N_R = 2$  MIMO system.

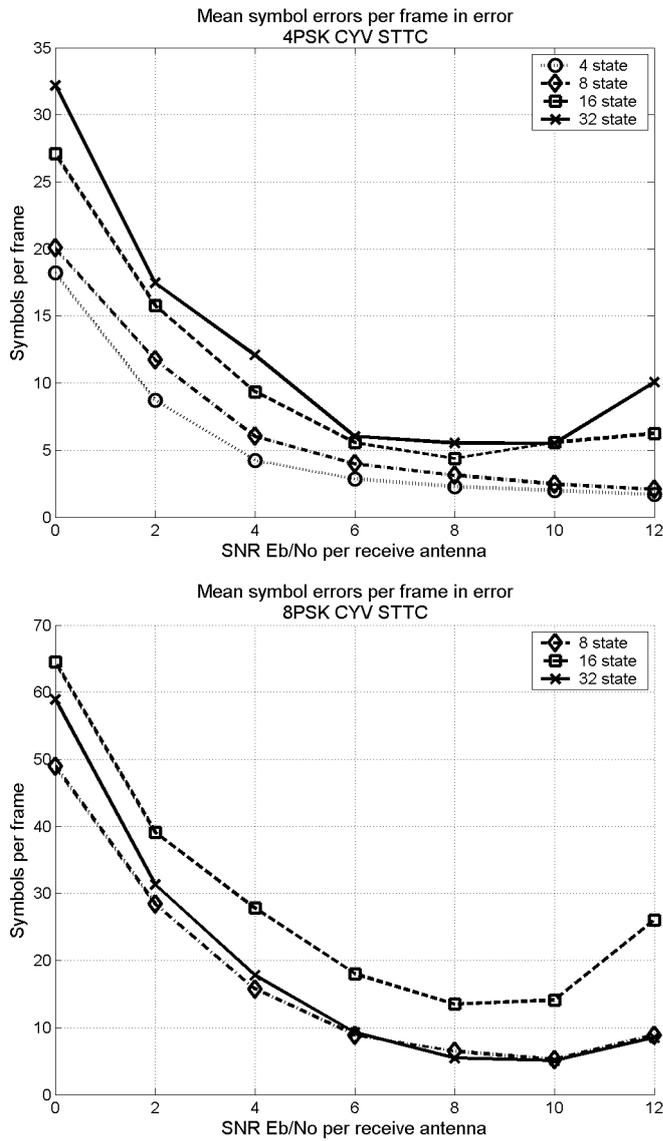
### 6.3.5 Soft Equalisation Gain

Another important feature of the iterative procedure used is generating soft information from the PVA equaliser. As shown in Section 5.3.1, this can be achieved by using either a soft symbol position, the SOVA, or both. Using soft information greatly improves the performance of the following STTC decoder.

Soft equalisation in the PVA gives a large performance gain over hard equalisation, as shown in Fig. 6.25. It can be seen that using a soft symbol estimate over a hard symbol gives a 1.5dB gain. Also, the use of the SOVA algorithm provides minimal performance gain when using soft (Fig. 6.20) or hard equalisation. This shows that the use of the SOVA algorithm is not required for the PVA algorithm. Generating a soft symbol position is less complex, as well as being more useful, than the SOVA.

Also shown in Fig. 6.25 is the performance of the PVA equaliser when using perfect feedback from the STTC decoder. The PVA equaliser uses the decoded symbols to initialise its symbol estimates. Good estimates provide part of the large gain seen in the iterative procedure. Using the actual transmitted symbols as the initial estimates provides only a small gain of around 0.3dB.

This interesting result shows that the PVA equaliser is limiting the performance of the system. This is shown by the overall performance with perfect feedback, still being around 1.7dB from ML decoding. It may be possible to improve this by using a better pre-filter, generating better soft information, or by using the soft information to help do interference cancellation inside the PVA. However the resulting reduction in decoding complexity should offset the performance loss.



**Figure 6.26** Number of symbol errors per frame for  $N_T = 2, N_R = 2$  MIMO system (a) 4PSK  $\alpha = 0.65, \beta = 0.1$  (b) 8PSK  $\alpha = 0.2, \beta = 0.2$

### 6.3.6 Errors Per Frame

The mean number of symbol errors per frame for different conditions is shown in Fig. 6.26. This number is calculated for frames that are in error. It is apparent that longer STTCs (with more states) experience a larger number of symbol errors in each frame in error. While this may at first seem contradictory to the improved performance shown by longer codes, using a better code mainly improves the actual number of frames in error. The increase in symbol errors is caused by the error output of the Viterbi algorithm having a longer burst of errors, which is typically at least as long as the memory length of the code used.

Another interesting result is that the number of errors per frame actually increases

with high SNR. This is in contrast to the performance at low SNR, which shows a dramatic reduction in the number of errors. Again, high SNR causes a large improvement in the frame error rate, so the overall number of errors still decreases. The increase in the number of errors per frame may be caused by random channels, with high correlation or poor pre-filter tap weights, that affect the results more at a high SNR.



## Chapter 7

---

### CONCLUSION

The demand for capacity in wireless communications is growing very rapidly, and looks to continue growing in the medium term future. An effective way of increasing capacity is to improve the spectral efficiency of a system. In order to increase spectral efficiency in wireless communications, a system with multiple transmit antennas and/or a larger signal constellation set is required.

In a dispersive channel, which is common in indoor and outdoor wireless systems, Inter-Symbol Interference (ISI) occurs. This causes a problem for the receiver to detect and decode the transmitted signals. The equalisation problem becomes harder in systems with large constellations, and multiple transmit antennas.

This research has been focused on investigating a new method of equalising MIMO systems in a dispersive, or frequency selective channel, with particular emphasis on decoding Space Time Trellis Codes (STTC) [58]. These codes add a coding gain to the system, which improves the BER performance.

The proposed system uses a sub-optimal equalisation method for a general multi user system, called the Partitioned Viterbi Algorithm (PVA) [43]. The equaliser estimates each transmitted signal separately, and uses estimates of the co-channel interferers to reduce ISI. The estimated symbols are then decoded in an ML (Viterbi based) STTC decoder. The estimates of the co-channel interferers can be improved after decoding, and feedback to the PVA in an iterative procedure. To improve performance, soft outputs are required, which are generated by a Soft Output Viterbi Algorithm (SOVA) [30].

The proposed system has been simulated to see what parameters affect its performance. The results are compared to the performance of a joint ML equaliser and decoder. The main benefit of the proposed system over ML is reduced complexity, but this is coupled with a small loss in system performance.

## 7.1 CONSTRUCTION CRITERIA

Simulations have been run to determine which STTC construction criteria performs better in the frequency selective channel. There are three types of STTCs used, Tarokh codes [58], which use the rank and determinant criteria, Baro codes [6], based on a computer search using the rank and determinant criteria, and Chen codes [10], based on the trace (Euclidean distance) criterion.

The simulations show that for both 4PSK with  $N_T = 2, 3$  and 8PSK with  $N_T = 2$ , the trace criteria performs better for systems with large diversity ( $rN_R > 3$ ). The only difference between all three codes are the coefficients used, there are no complexity or implementation differences. These results are shown for both ML decoding and iterative decoding. Therefore, in a practical system, the best code for a given length should be chosen.

## 7.2 ITERATIVE PROCEDURE PERFORMANCE

The results shown in Chapter 6 demonstrate the iterative performance of the proposed system. The first iteration does not perform very well, as the PVA equaliser has no estimate of the transmitted symbols. However, in subsequent iterations, the performance gets to within 2.0 to 2.5dB of ML decoding. This occurs using both 4PSK and 8PSK modulations, and various sizes of MIMO systems. The proposed iterative system performs comparably well to ML decoding, with a large computational reduction, especially for larger symbol constellations. There is a small diversity loss, but this is minimised by good coding gain from the STTC. However complexity for long channel delay lengths  $L$ , or large constellations, i.e. 16PSK/QAM or higher, would still be excessive. Such MIMO systems would require some form of reduced-state sequence estimation (RSSE) [20] or other methods, for example BLAST, instead of the PVA. The penalty for doing this would be a larger diversity loss.

The number of iterations required to get good performance from the system varies. The number of iterations increases for increased modulation size, and also for under-determined systems ( $N_T > N_R$ ). For a  $N_T, N_R = 2$ , 4PSK system, it appears 3 iterations gives good performance.

It also appears that the Chen codes get larger performance gain due to iterative processing. This is shown by a smaller gap between the ML and PVA performance curves. This is probably due to the increased coding gain providing better symbol estimates for interference cancellation in the equaliser.

### 7.3 CODING IMPLICATIONS

When using STTCs, the performance of the ML decoding algorithm improves if more transmit antennas are used to transmit a data sequence. The performance increase is due to more redundancy in the transmitted sequence, which offers a higher coding gain. However, the effective data rate is still one symbol per time period, out of a possible  $N_T$  symbols. The penalty due to this is increased decoding complexity.

In the proposed scheme, the PVA equaliser performance degrades with an increasing number of transmit antennas. This is caused by an increase in co-channel interference by effectively "random" (interleaved) sources. Iterating with the STTC decoder can reduce the performance degradation, but will increase the computational complexity.

The use of an interleaver dramatically improves performance in the iterative approach. However, ML decoding becomes extremely complex as there is no dependence between transmitted symbols, which increases the number of states in the trellis.

The original STTCs were designed to have a high rank in the codeword difference matrix [58]. As shown by Chen *et.al.* [10], and this work, performance in a large diversity system ( $rN_R > 3$ ) depends on the coding gain, or Euclidean distance, of the STTC. This means that, in general, any well designed convolutional code could be used in a MIMO system, in conjunction with the iterative approach.

The PVA equaliser does not know what type of decoder is used after it. For the iterative approach, the decoder should be chosen such that it can provide some soft information, and good estimated symbols to iterate with. This opens the possibility of using existing hardware designs for convolutional codes in a MIMO system. At the transmitter, the encoded symbols are distributed to antennas via an interleaver. The receiver then uses the PVA equaliser, followed by a de-interleaver and convolutional decoder.

### 7.4 THROUGHPUT IMPROVEMENT

The driving idea behind the research into MIMO systems is to increase the available throughput by increasing spectral efficiency. Increasing the number of transmit antennas increases the maximum throughput available, i.e.  $N_T = 2$  antennas using 4PSK gives 4bits/s/Hz, while  $N_T = 3$  antennas using 4PSK gives 6bits/s/Hz. Increasing the number of receive antennas effectively increases the SNR, which also improves the BER performance. However, current STTC designs only increase spectrum efficiency (bps/Hz) by increasing the modulation size, i.e. from 4-PSK (2bps/Hz) to 8-PSK (3bps/Hz). So increasing a 4PSK system from  $N_T = 2$  to  $N_T = 3$  transmit antennas and using a STTC, improves the BER, but keeps the transmitted data rate of 2bps/Hz.

This introduces the idea of a "coding rate" for STTCs. The coding rate that is defined for a standard convolutional code. Normally for 1 data symbol,  $N_T$  symbols are transmitted during one time period, i.e. rate =  $\frac{1}{N_T}$ . In previous example, with 4PSK and  $N_T = 2$  antennas, the rate is  $\frac{1}{2}$ , and  $N_T = 3$  antennas gives a rate of  $\frac{1}{3}$ .

In the iterative approach, the PVA equaliser estimates the extra transmitted signals as more random interference. This means that increasing the number of transmit antennas while using the PVA is an ineffective method to increase coding gain. However, increasing the number of transmit antennas, and increasing the data throughput would be very useful. For example, using 4PSK and upgrading from  $N_T = 2$  to  $N_T = 3$  transmit antennas, and using a standard rate  $\frac{1}{2}$  convolutional code instead of a STTC would result in a transmitted data rate of 3bps/Hz. For a STTC to achieve this, it would require using 8PSK, which increases the equalisation problem. However, the STTC has an effective rate of  $\frac{1}{3}$ , so would provide a larger coding gain.

This opens up the possibility for interesting new research to investigate the tradeoff between number of transmit antennas, coding rate and modulation size. For example, a  $N_T = 3$  rate  $\frac{1}{2}$  4PSK system may give comparable performance to a  $N_T = 2$  rate  $\frac{1}{2}$  8PSK system, with both systems having a data throughput of 3bits/s/Hz. In the frequency selective channel, the 4PSK system would be much easier to equalise.

It may also be possible to develop adaptive systems. At low SNR, few transmit antennas are used, so the PVA equaliser performs better. As the SNR improves, an extra transmit antenna could be added to increase data throughput. This would increase decoding complexity, but only linearly, compared to increasing exponentially with an increased modulation size.

## 7.5 COMPLEXITY CONSIDERATIONS

The idea of using a normal convolutional code and a large interleaver in a MIMO system would not add much computational complexity to the proposed iterative method. The PVA equaliser remains unchanged, no matter what "outer" code is used. As the equaliser takes about 90% of the total complexity, relatively powerful convolutional codes could be used to provide good coding gain. However, the iterative process would add latency to a system, and may not be suitable for voice or similarly sensitive applications.

It should also be noted that ML decoding could not feasibly be used for higher rate ( $> \frac{1}{N_T}$ ) codes. The number of equalisation states for ML decoding of an uncoded MIMO system is  $M^{N_T(L-1)}$ , which is the total number of possible symbol combinations. The number of states required in a MIMO system using a rate  $r$  code is actually  $M^{rN_T(L-1)}$ , which for a STTC of rate  $r = \frac{1}{N_T}$  is  $M^{L-1}$ . Therefore any rate increase would exponentially increase the complexity of the system.

## 7.6 FUTURE RESEARCH DIRECTIONS

This research has raised a number of interesting questions. It shows that a sub-optimal iterative approach is possible when decoding STTCs in the frequency selective channel. It has also shown how standard convolutional codes could be used to increase user bandwidth in a MIMO channel, with very little complexity increase.

The proposed iterative approach is inherently suboptimal. However, there are a large number of parameters and assumptions made that affect its performance. There are also some tradeoffs that can be made between implementation complexity and performance. The parameters come from both the model used to develop the approach, and the implementation of the approach itself. The model assumptions include independent channel taps, flat power delay profile, perfect channel knowledge and AWGN being due to the receiver electronics. The parameters that affect the design of the iterative approach include the type of prefilter used, size of the interleaver used, STTC gain and construction, choice and calculation of  $\alpha$  and  $\beta$  etc.

The question of whether using a current STTC is efficient in a MIMO system has also been raised. More research would be required to understand the performance and complexity tradeoffs that using standard convolutional, or other codes, would bring. Such approaches may come closer to realising the goal of increasing user bandwidth in both a computational and spectrally efficient way.



## Appendix A

---

### SYMBOLS USED

Table A.1: Symbols Used

Symbol	Definition
$\mathbf{A}(\mathbf{c}, \mathbf{e})$	Euclidean Distance matrix of transmitted sequence $\mathbf{c}$ and received sequence $\mathbf{e}$
$\alpha$	Scaling factor of PVA SOVA output
$\mathbf{B}(\mathbf{c}, \mathbf{e})$	Difference matrix of transmitted sequence $\mathbf{c}$ and received sequence $\mathbf{e}$
$\beta$	Scaling factor of STTC SOVA output
$c$	Transmitted STTC sequence
$\delta^t(\mathbf{i}, \mathbf{j}, k)$	Path metric between state $\mathbf{i}$ and state $\mathbf{j}$ at time $k$
$e$	Decoded STTC sequence
$E_b$	Average transmitted energy per bit per antenna
$E_s$	Average transmitted energy per symbol per antenna
$F_m$	Tap $m$ of overall pre-filter
$\mathbf{F}$	Overall pre-filter response
$h^{i,j}$	Channel tap between receiver $i$ and transmitter $j$
$\mathbf{H}$	Matrix of channel taps
$\hat{\mathbf{H}}$	Estimated channel taps
$\tilde{\mathbf{H}}$	Filtered channel taps ( $\mathbf{F} \otimes \hat{\mathbf{H}}$ )
$L$	Length of the channel impulse response
$L_f$	Length of DFE pre-filter
$\lambda_j$	Eigenvalue $j$ of matrix $\mathbf{A}(\mathbf{c}, \mathbf{e})$
$\lambda$	Path metric between two states in Viterbi algorithm
$M$	Symbol Alphabet size, i.e. 4 for QPSK
$n_k^i$	Received noise at antenna $i$ , at time $k$
$\mathbf{n}$	Vector of received noise
$\tilde{\mathbf{n}}$	Filtered received noise
$N_0$	AWGN Variance
$N_d$	Number of data symbols in a frame
$N_R$	Number of Receive Antennas

Table A.1: Symbols Used

$N_s$	Number of samples per symbol
$N_{ST}$	Number of states in STTC decoder
$N_T$	Number of Transmit Antennas
$\phi(k, t)$	Co-channel interference term for transmitter $t$ at time $k$
$r$	The rank of matrix $\mathbf{A}(\mathbf{c}, \mathbf{e})$
$v$	Memory of code or trellis
$x_k^j$	Transmitted symbol from antenna $j$ , at time $k$
$\mathbf{x}$	Vector of transmitted symbols
$\bar{\mathbf{x}}$	Tentative estimated symbols
$\hat{\mathbf{x}}$	Hard estimated symbols
$\tilde{\mathbf{x}}$	Soft equalised symbols
$\mathbf{X}$	Training sequence matrix
$y_k^i$	Received signal at antenna $i$ , at time $k$
$\mathbf{y}$	Vector of received signals
$\tilde{\mathbf{y}}$	Filtered received symbols

## Appendix B

---

### ABBREVIATIONS USED

Table B.1: Abbreviations Used

Abbreviation	Definition
ACS	Add Compare Select
AWGN	Additive White Gaussian Noise
BCH	Bose-Chaudhuri-Hochquenghem, a type of block error correcting code
BER	Bit Error Rate
BLAST	Bell Laboratories Layered Space Time Architecture
bps	Bits per Second
BPSK	Binary Phase Shift Keying
CDMA	Code Division Multiple Access
CIR	Channel Impulse Response
CSI	Channel State Information
DFE	Decision Feedback Equaliser
DSL	Digital Subscriber Loop
FER	Frame Error Rate
FIR	Finite Impulse Response
FFT	Fast Fourier Transform
GSM	Global System for Mobile communications
ISI	Inter Symbol Interference
IEEE	Institute of Electrical and Electronic Engineers
IPv6	Internet Protocol Version 6
LVA	List Viterbi Algorithm
MAC	Multiply Accumulate
MAP	Maximum A Posteriori
MIMO	Multiple Input Multiple Output
MISO	Multiple Input Single Output
ML	Maximum Likelihood
MLSE	Maximum Likelihood Sequence Estimator

Table B.1: Abbreviations Used

MMSE	Minimum Mean Squared Error
MRC	Maximum Ratio Combining
MSE	Mean Squared Error
OFDM	Orthogonal Frequency Division Multiplexing
PSD	Power Spectral Density
PVA	Partitioned Viterbi Algorithm
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
RSSE	Reduced State Sequence Estimation
SER	Symbol Error Rate
SISO	Single Input Single Output
SIMO	Single Input Multiple Output
SNR	Signal to Noise Ratio
SOVA	Soft Output Viterbi Algorithm
STBC	Space Time Block Code
STTC	Space Time Trellis Code
VA	Viterbi Algorithm
WMF	Whitened Matched Filter

---

## REFERENCES

- [1] D. Agrawal, V. Tarokh, A. Naguib, and N. Seshadri, "Space-time coded OFDM for high data-rate wireless communication over wideband channels," in *Proc. IEEE VTC'98*, Ottawa, Canada, May 1998, pp. 2232–2236.
- [2] H. Akaike, "Block toeplitz matrix inversion," *SIAM J. Appl. Math.*, pp. 234–241, Mar. 1973.
- [3] N. Al-Dhahri and J. M. Cioffi, "MMSE decision-feedback equalizers: Finite-length results," *IEEE Trans. Inform. Theory*, vol. 41, pp. 961–975, July 1995.
- [4] S. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE J. Select. Areas Commun.*, pp. 1451–1458, Oct. 1998.
- [5] L. R. Bahl, J. Cocke, E. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. 20, pp. 284–287, Mar. 1974.
- [6] S. Baro, G. Bauch, and A. Hansmann, "Improved codes for space-time trellis-coded modulation," *IEEE Commun. Lett.*, vol. 4, pp. 20–22, Jan. 2000.
- [7] G. Bauch and A. Naguib, "Map equalization of space-time coded signals over frequency selective channels," in *Proc. IEEE WCNC 1999*, Sept. 1999, pp. 261–265.
- [8] C. Berrou, A. Glavieux, and P. Thitimajhima, "Near Shannon limit error correcting coding and decoding: turbo-codes," in *Proc. IEEE ICC'93*, Geneva, May 1993, pp. 1064–1070.
- [9] S. Bong-Gee and J. Ritcey, "Prefilter design using the singular value decomposition for MIMO equalization," in *Asilomar Conference on Signals Systems and Computers 1996*, Nov. 1996, pp. 34–38.
- [10] Z. Chen, J. Yuan, and B. Vucetic, "An improved space-time trellis coded modulation scheme for slow Rayleigh fading channels," in *Proc. IEEE ICC 2001*, vol. 4, June 2001, pp. 1110–1116.

- [11] Z. Chen, B. S. Vucetic, K. L. Lo, and J. Yuan, "Space-time trellis codes for 8-PSK with three and four transmit antennas in quasi-static flat fading channels," *Electron. Lett.*, vol. 38, pp. 462–464, May 2002.
- [12] Z. Chen, B. S. Vucetic, J. Yuan, and K. L. Lo, "Space-time trellis codes for 4-PSK with three and four transmit antennas in quasi-static flat fading channels," *IEEE Commun. Lett.*, vol. 60, pp. 67–69, Feb. 2002.
- [13] J. Cheng, H. Wang, and S. Cheng, "Space-time block coded transmit diversity for OFDM system in mobile channels," in *Proc. IEEE PIMRC 2002*, Sept. 2002, pp. 208–211.
- [14] S. Cherry, "The wireless last mile," *IEEE Spectr.*, vol. 40, pp. 18–22, Sept. 2003.
- [15] L. Cimini, "Analysis and simulation of a digital mobile channel using orthogonal frequency division multiplexing," *IEEE Trans. Commun.*, vol. 33, pp. 665–675, July 1985.
- [16] G. Colavolpe, G. Ferrari, and R. Raheli, "Reduced-state BCJR-type algorithms," *IEEE J. Select. Areas Commun.*, vol. 19, pp. 848–859, May 2001.
- [17] M. O. Damen, H. E. Gamal, and G. Caire, "On maximum-likelihood detection and the search for the closest lattice point," *IEEE Trans. Inform. Theory*, vol. 49, pp. 2389–2402, Oct. 2003.
- [18] S. Diggavi, N. Al-Dhahir, A. Stamoulis, and A. Calderbank, "Great expectations: the value of spatial diversity in wireless networks," in *Proc. IEEE*, vol. 92, Feb. 2004, pp. 219–270.
- [19] M. El-Sayed and J. Jaffe, "A view of telecommunications network evolution," *IEEE Commun. Mag.*, vol. 40, pp. 74–81, Dec. 2002.
- [20] M. Eyuboglu and S. Qureshi, "Reduced-state sequence estimation for coded modulation of intersymbol interference channels," *IEEE J. Select. Areas Commun.*, vol. 7, pp. 989–995, Aug. 1990.
- [21] M. Fitz, J. Grimm, and S. Siwamogsatham, "A new view of performance analysis techniques in correlated Rayleigh fading," in *Proc. IEEE WCNC 1999*, Sept. 1999, pp. 139–144.
- [22] G. Forney, "Maximum likelihood sequence estimation of digital sequences in the presence of intersymbol interference," *IEEE Trans. Inform. Theory*, vol. 18, pp. 363–378, May 1972.
- [23] G. Forney, "The Viterbi algorithm," in *Proc. of the IEEE*, vol. 61, Mar. 1973, pp. 268–278.

- [24] G. Foschini, "Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas," *Bell Labs Technical Journal*, pp. 41–59, Sept. 1996.
- [25] G. Foschini, R. Valenzuela, and P. W. Wolniansky, "Simplified processing for high spectral efficiency wireless communication employing multi-element arrays," *IEEE J. Select. Areas Commun.*, pp. 1841–1852, Nov. 1999.
- [26] C. Fragouli, N. Al-Dhahir, , S. N. Diggavi, and W. Turin, "Prefiltered space-time M-BCJR equalizer for frequency-selective channels," *IEEE Trans. Commun.*, vol. 50, pp. 742–753, May 2002.
- [27] C. Fragouli, N. Al-Dhahir, and W. Turin, "Training-based channel estimation for multiple-antenna broadband transmissions," *IEEE Trans. Wireless Commun.*, vol. 2, pp. 384–391, Mar. 2003.
- [28] W. H. Gerstacker, P. Nickel, and D. Taylor, "Diversity interference cancellation using prefiltering and reduced-state MIMO equalization," in *Proc. IEEE ICASSP 2004*, May 2004, pp. 813–816.
- [29] W. H. Gerstacker, F. Obernosterer, R. Meyer, and J. B. Huber, "An efficient method for prefilter computation for reduced-state equalization," in *Proc. IEEE PIMRC 2000*, Sept. 2000, pp. 604–609.
- [30] J. Hagenauer and P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications," in *Proc. IEEE GLOBECOM '89*, Nov. 1989, pp. 1680–1686.
- [31] R. Hinden. (2003) IP version 6 (IPv6). ipng-main.html. [Online]. Available: <http://playground.sun.com/pub/ipng/html/>
- [32] Y. Hong, J. Yuan, and X. Shao, "Robust space-time trellis codes for OFDM systems over quasi-static frequency selective fading channels," in *Proc. IEEE PIMRC 2003*, Sept. 2003, pp. 434–438.
- [33] J.-L. Hurel, C. Evci, C. Lerouge, and L. Gui. (2003) Mobile network evolution: From 3g onwards. [Online]. Available: <http://www.alcatel.com/doctypes/articlepaperlibrary/pdf/ATR2003Q4/T0312-Mobile-Evolution-EN.pdf>
- [34] IPv6. (2004) IPv6 forum. index.html. [Online]. Available: <http://www.ipv6.org/>
- [35] H. Jafarkhani, "A quasi-orthogonal space-time block code," *IEEE Trans. Commun.*, vol. 49, pp. 1–4, Jan. 2001.
- [36] A. Jone, T. Wilkinson, and S. Barton, "Block coding scheme for reduction of peak-to-mean envelope power ratio of multicarrier transmission schemes," *Electron. Lett.*, vol. 30, pp. 2098–2099, 1994.

- [37] A. R. Kaye and D. A. George, "Transmission of multiplexed PAM signals over multiple channel and diversity systems," *IEEE Trans. Commun.*, vol. 18, pp. 520–526, Oct. 1970.
- [38] R. Koetter, A. Singer, and M. Tuchler, "Turbo equalization," *IEEE Signal Processing Mag.*, vol. 21, pp. 67–80, Jan. 2004.
- [39] X. Li, T. Luo, G. Yue, and C. Yin, "A squaring method to simplify the decoding of orthogonal space-time block codes," *IEEE Trans. Commun.*, pp. 1700–1703, Oct. 2001.
- [40] Y. Li, N. Seshadri, and S. Ariyavisitakul, "Channel estimation for OFDM systems with transmitter diversity in mobile wireless channels," *IEEE J. Select. Areas Commun.*, vol. 17, pp. 461–471, Mar. 1999.
- [41] Y. Liu, M. P. Fitz, and O. Y. Takeshita, "Space-time codes performance criteria and design for frequency selective fading channels," in *Proc. IEEE ICC 2001*, June 2001, pp. 2800–2804.
- [42] W. W. Lu, *Broadband Wireless Mobile: 3G and Beyond*. England: John Wiley and Sons, 2002.
- [43] C. Miller, D. Taylor, and P. Gough, "Estimation of co-channel signals with linear complexity," *IEEE Trans. Commun.*, vol. 49, pp. 1997–2004, Nov. 2001.
- [44] C. Miller and D. Taylor, "Digital prefiltering for multiple-input, multiple-output receivers," in *Proc. IEEE ICC 2001*, June 2001, pp. 1696–1699.
- [45] A. Molisch, M. Steinbauer, M. Toeltsch, E. Bonek, and R. Thoma, "Capacity of MIMO systems based on measured wireless channels," *IEEE J. Select. Areas Commun.*, vol. 20, pp. 561–569, Apr. 2002.
- [46] A. Moustakas, S. Simon, and A. Sengupta, "MIMO capacity through correlated channels in the presence of correlated interferers and noise: A (not so) large N analysis," *IEEE Trans. Inform. Theory*, vol. 49, pp. 2545–2561, Oct. 2002.
- [47] S. Muller and J. Huber, "OFDM with reduced peak-to-average power ratio by optimum combination of partial transmit sequences," *Electron. Lett.*, vol. 33, pp. 368–369, 1997.
- [48] C. Nill and C.-E. Sundberg, "List and soft symbol output Viterbi algorithms: extensions and comparisons," *IEEE Trans. Commun.*, vol. 43, pp. 277–287, Feb./Mar./Apr. 1995.
- [49] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C : The Art of Scientific Computing*, 2nd ed. Cambridge CB2 1RP: Cambridge University Press, 1992.

- [50] J. G. Proakis, *Digital Communications*, 4th ed. New York, N.Y.: McGraw-Hill, 2001.
- [51] B. Qi, G. Zysman, and H. Menkes, "Wireless mobile communications at the start of the 21st century," *IEEE Commun. Mag.*, vol. 39, pp. 110–116, Jan. 2001.
- [52] M. Qin and R. Blum, "Properties of space-time codes for frequency selective channels and trellis code designs," in *Proc. IEEE ICC 2003*, May 2003, pp. 2286–2290.
- [53] Y. Sasazaki and T. Ohtsuki, "Improved design criteria and new codes on space-frequency trellis coding over frequency selective fading channels," in *Proc. IEEE VTC 2002*, Sept. 2002, pp. 2187–2191.
- [54] D. Shiu and M. Kahn, "Layered space-time codes for wireless communications using multiple transmit antennas," in *Proc. IEEE ICC'99*, vol. 1, June 1999, pp. 436–440.
- [55] M. Simon, J. Omura, R. Scholtz, and B. Levitt, *Spread Spectrum Communications*. Computer Science Press, 1985.
- [56] T. Star, *DSL Advances*. Upper Saddle River, N.J.: Prentice Hall, 2003.
- [57] W. Su and X. Xia, "A design of quasi-orthogonal space-time block codes with full diversity," in *Proc. IEEE CSSC 2002*, vol. 2, Nov. 2002, pp. 1112–1116.
- [58] V. Tarokh, A. Naguib, N. Seshdri, and A. R. Calderbank, "Combined array processing and space time coding," *IEEE Trans. Inform. Theory*, vol. 45, pp. 1121–1128, May 1999.
- [59] V. Tarokh, N. Seshdri, and A. R. Calderbank, "Space-time codes for high data rate wireless communication: Performance criteria and code construction," *IEEE Trans. Inform. Theory*, vol. 44, pp. 744–765, Mar. 1998.
- [60] I. E. Telatar, "Capacity of multiple antenna gaussian channels," *European Transactions on Telecommunications*, pp. 585–595, Nov./Dec. 1999.
- [61] C. Tidestav, M. Sternad, and A. Ahln, "Reuse within a cell - interference rejection or multiuser detection?" *IEEE Trans. Commun.*, pp. 1511–1522, Oct. 1999.
- [62] O. Tirkkonen and A. Hottinen, "Improved MIMO performance with non-orthogonal space-time block codes," in *Proc. IEEE GLOBECOM 2001*, vol. 2, Nov. 2001, pp. 1122–1126.
- [63] B. Tomiuk and N. Beaulieu, "A new look at maximal ratio combining," in *Proc. IEEE GLOBECOM'00*, vol. 2, San Francisco, CA USA, Dec. 2000, pp. 943–948.

- [64] A. Triolo, J. Liberti, and T. Hoerning, "OFDM space-time trellis coded MIMO systems with experimental results," in *Proc. MILCOM 2002*, vol. 1, Oct. 2002, pp. 577–581.
- [65] G. Ungerboeck, "Adaptive maximum-likelihood receiver for carrier-modulated data-transmission systems," *IEEE Trans. Commun.*, pp. 624–636, May 1974.
- [66] W. van Etten, "An optimum linear receiver for multiple channel digital transmission systems," *IEEE Trans. Commun.*, vol. 23, pp. 824–834, Aug. 1975.
- [67] W. van Etten, "Maximum likelihood receiver for multiple channel transmission systems," *IEEE Trans. Commun.*, vol. 24, pp. 276–283, Feb. 1976.
- [68] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inform. Theory*, vol. 13, pp. 260–269, Apr. 1967.
- [69] E. Viterbo and J. Boutros, "A universal lattice code decoder for fading channels," *IEEE Trans. Inform. Theory*, vol. 45, pp. 1639–1642, July 1999.
- [70] L. Wei, "High performance iterative viterbi algorithm for conventional serial concatenated systems," *IEEE Trans. Inform. Theory*, vol. 48, pp. 1759–1771, July 2002.
- [71] J. Winters, "The diversity gain of transmit diversity in wireless systems with Rayleigh fading," in *Proc. IEEE ICC'94*, vol. 2, New Orleans, LA USA, May 1994, pp. 1121–1125.
- [72] A. Wittneben, "A new bandwidth efficient transmit antenna modulation diversity scheme for linear digital modulation," in *Proc. IEEE ICC'93*, vol. 3, Geneva, Switzerland, May 1993, pp. 1630–1634.
- [73] P. Wolniasky, G. Foschini, G. Golden, and R. Valenzuela, "V-BLAST: An architecture for realizing very high data rates over the rich-scattering wireless channel," in *Proc. ISSSE'98*, Sept./Oct. 1998, pp. 295–300.
- [74] Q. Yan and R. Blum, "Optimum space-time convolutional codes," in *Proc. IEEE WCNC 2000*, vol. 3, Sept. 2000, pp. 1351–1355.
- [75] W. Younis and N. Al-Dhahir, "Joint prefiltering and MLSE equalization of space-time coded transmissions over frequency-selective channels," *IEEE Trans. Veh. Technol.*, vol. 51, pp. 144–154, Jan. 2002.