SIMULATION OF TURBO RECEIVER FOR VECTOR OFDM

by

Zhe Qu

A thesis submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Master of Science in Electrical and Computer Engineering

Spring 2017

SIMULATION OF TURBO RECEIVER FOR VECTOR OFDM

by

Zhe Qu

Approved: _____
Xiang-Gen Xia, Ph.D.
Professor in charge of thesis on behalf of the Advisory Committee

Approved: _____
Kenneth E. Barner, Ph.D.
Chair of the Department of Electrical and Computer Engineering

Approved: _____
Babatunde A. Ogunnaike, Ph.D.
Dean of the College of Engineering

Approved: _____
Ann L. Ardis, Ph.D.
Senior Vice Provost for Graduate and Professional Education

ACKNOWLEDGMENTS

I wish to thank everyone who has helped me during my master career. My advisor, Xiang-Gen Xia, who has helped me not only my education but also my research; and my friends and colleagues Tianyuan Qiu and Hao Xu.

TABLE OF CONTENTS

LIST OF FIGURE

ABSTRACT

Vector orthogonal frequency division multiplexing (V-OFDM) for single transmit antenna systems is a general block based transmission scheme, where OFDM and single carrier frequency domain equalizer (SC-FDE) are two special and extremal cases.

In this thesis, algorithms of detecting and Turbo decoding for V-OFDM systems are simulated and discussed. We use quadrature amplitude modulation (QAM) to modulate Turbo coded V-OFDM (TC-V-OFDM) system, the output of detecting part is soft information, therefore the Turbo decoding algorithm need to be modified. By modifying the maximum a posteriori (MAP) algorithm, a single input single output (SISO) MAP Turbo decoding algorithm has been discussed in this thesis.

Then, the theory of decoding turbo code is applied to the detection of signals by the joint of equalization and decoding. The recursive equalization algorithm of MAP and linear MMSE are discussed in detail. The lower-complexity MMSE equalization algorithm has eventually been chosen as the simulation algorithm because that the complexity of MAP will increase exponentially with the modulation order and the memory length of ISI channel. The characteristics of V-OFDM's cyclic prefix reduce the complexity of MMSE equalizer further.

Turbo Receiver of TC-V-OFDM system is realized in this thesis by applying the recursion, and this not only increases the diversity gains, but also decreases bit error rate (BER) of the system, therefore the performance of the system is improved greatly.

Key Words: Turbo code; Turbo Receiver; V-OFDM; MMSE

Chapter 1

INTRODUCTION

Orthogonal frequency-division multiplexing (OFDM) is an effective and popular modulation scheme for high-data-rate transmission because using cyclic prefixes and Digital Fourier transform (DFT)can simplify multiple-tap equalization to multiple one-tap equalization. Recently, OFDM has been found its advantages for a few prevelent broadband communications, such as digital audio broadcasting, asymmetric digital subscriber line modems, wireless personal networks and wireless local area networks (WLANs). Vector-OFDM for a single-antenna system which mentioned in paper[17] is a kind of general OFDM. For multiple subcarriers, OFDM can change intersymbol interference (ISI) channel into ISI-free channels. And vector OFDM an ISI channel into multiple ISI-free vector channels. On the one hand, we can improve the bandwidth efficiency because the cyclic prefix redundancy of vector OFDM is a small part of OFDM by the same FFT size. On the other hand, using smaller FFT size and peak-to-average-power ratio (PAPR) can make vector OFDM with the same cyclic prefix redundancy as OFDM to be possible. In paper, past authors[23] proposed the vector OFDM with linear receivers. However, linear receivers give poor bit error rate (BER). In frequency selective fading channels, via the signal space diversity inherent in vector-OFDM, the system can achieve the performance gain, where the vector length is equal to the maximum obtainable diversity order[18]. In paper[19], a recursive receiver structure was proposed that a kind of convolutionally coded vector OFDM systems used for transmission over frequency-selective fading channels. In each recursion, there are two soft-output decisions were performed by receiver through log-likelihood ratio(LLR): for the SISO channel decoding and the

soft-input-soft-output (SISO) vector OFDM demodulation. In this thesis, linear equalization and turbo coding are used and implemented.

Following the discovery of Turbo code[2], which makes recursive extrinsic information to exchange between concatenated detectors and decoders, has been effectively implemented to many detection and decoding problems. And by linear equalization, it makes the turbo code easier to decode. For each stage, the extrinsic information is defined that the calculated *a posteriori* information subtract the input *a priori* information, and then used as the *a priori* information, for example: MAP decoding[24].

This thesis is organized as follows: In Chapter 2, the Turbo encoding is introduced. In Chapter 3, the Turbo decoding is discussed, including the BCJR algorithm, MAP, Log-MAP and Max-MAP decoding. In Chapter 4, the system of vector OFDM is introduced, including comparing to original OFDM and different decoding methods. In Chapter 5, the Turbo equalized vector OFDM is discussed and simulated.

Chapter 2

TURBO ENCODING

In 1948, Shannon proposed the channel coding theory[1]: in noisy channel, when the real transmission rate is lower than channel capacity, it is possible to transmit information nearly without error. Shannon proposed three basic conditions: 1. Random coding methods; 2. Coding length is finite; 3. Maximum likelihood decoding. Under these three conditions, we can get transmit signal with less error.

Researches have gotten many successes after Shannon theorem, but channel capacity has also 2-3db below Shannon limit. Cut-off frequency of channel incorrected was believed to be the real limit of error correcting code. Until turbo code occurred, it was possible to achieve Shannon limit.

Turbo code was used by two parallel concatenated, and the interleaved rate is 1/2 by recursive systematic convolutional codes (see Fig. 1, where P and P^{-1} represent the interleaver and the deinterleaver. P is a row vector involving permutation pattern function). Two encoders in Fig. 1 are the same, and their output bits $x_{kp}^{(1)}$ and $x_{kp}^{(2)}$ are alternately collected through puncturing so that all rate 1/2 code can be obtained. That $u_i^{(P)}$ is the *i*th element of the interleaved sequence, is equal to the P_i-th element of the original sequence, $u_i^{(P)} = u_{P_i}$.

Turbo encoder is comprised of two rate 1/2 recursive systematic convolution (RSC) encoders as shown in Figure 1[2]. The first encoder takes $\mathbf{s}$ to be the input information bits and parity bits $\mathbf{x}_1^P$ generated. The interleaver interleaves the information bits $\mathbf{s}$ to be interleaved information $\overline{\mathbf{s}}$. And the second encoder uses $\mathbf{s}$ and parity bits $\mathbf{x}_2^P$ is generated[42].

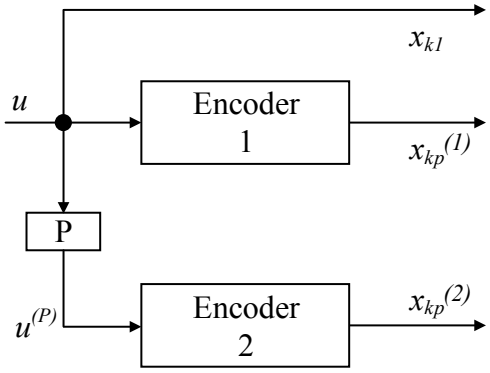

Fig.1 The turbo encoder

According to the rate, symbol $\mathbf{x}^P$ is derived from puncturing, and coded symbol through multiplexing [3][4].

Chapter 3

TURBO DECODING

3.1 The purpose of the BCJR algorithm

We consider that a block or convolutional encoder is described by trellis. The output of the encoder is a *N*-bit codewords or symbols sequence, where *k*th time encoder generated the symbol x_k. u_k is the corresponding information input bit, which represented by the values -1 or $+1$ that is an *a priori* probability $P(u_k)$, where the log-likelihood ratio(LLR) can be defined.

$$L(u_k) = \ln \frac{P(u_k = +1)}{P(u_k = -1)} \tag{2.1.1}$$

The ratio of log-likelihood is zero, and it is equally likely input bits.

We assume that the coded sequence $\mathbf{x}$ is transmitted through a memoryless additive white Gaussian noise(AWGN) channel and received the sequence including nN real numbers $\mathbf{y} = y_1 y_2 \cdots y_N$, as shown in Figure 2. This sequence is delivered to the decoder and used by the BCJR[5] is named by its inventors: Bahl, Cocke, Jelinek and Raviv. If we want to estimate the original bit sequence u_k, where the posteriori log-likelihood ratio $L(u_k|\mathbf{y})$ can be computed by the algorithm, a real number defined by ratio

$$L(u_k|\mathbf{y}) = \ln \frac{P(u_k = +1|\mathbf{y})}{P(u_k = -1|\mathbf{y})} \tag{2.1.2}$$

The Eq. (2.1.2) contains *a posteriori* conditional probabilities, and after we know $\mathbf{y}$, the probabilities can be computed. The positive sign of $L(u_k|\mathbf{y})$ can point the bit, which value is -1 or $+1$, was coded at *k*th time. The magnitude of $L(u_k|\mathbf{y})$ is a kind of measurement to make sure the decision that we mentioned before: if more $L(u_k|\mathbf{y})$ magnitude is not closed to the zero-threshold decision, thus our bit estimation is retained. The soft information that is provided by $L(u_k|\mathbf{y})$ will be moved to another decoding block. A hard decision would convert it into a bit value, as previously

discussed, if $L(u_k|\mathbf{y}) < 0$, the decoder that estimates bit $u_k = -1$ was sent. Similarly, the decoder will estimate $u_k = +1$ if $L(u_k|\mathbf{y}) > 0$.

It is convenient to work with trellises. According to the model in chapter 1, we use a convolutional code that rate 1/2, $n = 2$, with $Q = 4$ states presented in Fig. 3. In the Fig .3, the dashed line denotes that branch was repersented by a +1 message bit and the solid line indicates the opposite. A two-bit codeword x_k is used to lable each branch, where 0 and 1 correspond -1 and $+1$.

We suppose that now is at the *k*th time, so the corresponding state is $T_k = t$, the previous state is $T_{k-1} = t'$ and the received symbol by decoder is y_k . The complete sequence $\mathbf{y}$ consists of three subsequences: past, present and future:

$$\mathbf{y} = \underbrace{y_1 y_2 \cdots y_{k-1}}_{\mathbf{y}_{<k}} y_k \underbrace{y_{k+1} \cdots y_N}_{\mathbf{y}_{>k}} = \mathbf{y}_{<k} \mathbf{y}_k \mathbf{y}_{>k} \tag{2.1.3}$$

The *a posteriori* LLR $L(u_k|\mathbf{y})$ is given by the expression

$$L(u_k|\mathbf{y}) = \ln \frac{\sum_{R_1} P(t', t, \mathbf{y})}{\sum_{R_0} P(t', t, \mathbf{y})} = \ln \frac{\sum_{R_1} \alpha_{k-1}(t') \gamma_k(t', t) \beta_k(t)}{\sum_{R_0} \alpha_{k-1}(t') \gamma_k(t', t) \beta_k(t)} \tag{2.1.4}$$

$P(t', t, \mathbf{y})$ is defined by the joint probability between receiving the *N*-bit sequence $\mathbf{y}$ and in state t' at the *k*-1 time and in state t at the current time k . The numerator R_1 means that the summation is computed all the state transitions from t' to t due to message bits $u_k = +1$. Likewise, the denominator R_1 represents the set of all branches originated by message bits $u_k = -1$. α , γ and β are variables representing probabilities. They will be defined later.

3.1.1 The joint probability $\mathbf{P}(t', \mathbf{t}, \mathbf{y})$

Through the product of three other probabilities, this probability can be computed,

$$P(t', t, \mathbf{y}) = \alpha_{k-1}(t') \gamma_k(t', t) \beta_k(s) \tag{2.1.5}$$

They are defined as:

$$\alpha_{k-1}(t') = P(t', \mathbf{y}_{<k}) \tag{2.1.6}$$

$$\gamma_k(t', t) = P(\mathbf{y}_k, t | t') \tag{2.1.7}$$

$$\beta_k(t) = P(\mathbf{y}_{>k} | t) \tag{2.1.8}$$

At kth time, the probabilities α, γ and β are joint by the past, the present and the future of sequence $\mathbf{y}$.

3.1.2 Calculation of γ

If we know that $T_{k-1} = t'$ is the state where the connecting branch came, the probability $\gamma_k(t', t) = P(\mathbf{y}_k, t | t')$ is the conditional probability that can be calculated by received symbol $\mathbf{y}_k$ at kth time and current state $T_k = t$. Then γ is defined:

$$\gamma_k(t', t) = P(\mathbf{y}_k | x_k) P(u_k) \tag{2.1.9}$$

The previous expression of an AWGN channel becomes

$$\gamma_k(t', t) = \Lambda_k e^{u_k L(u_k)/2} \exp\left(\frac{L_c}{2} \sum_{l=1}^{n} x_{kl} y_{kl} \right) \tag{2.1.10}$$

Λ_k will be canceled out, so we calculate the conditional LLR $L(u_k | \mathbf{y})$. Because the numerator and the denominator of Eq. (2.1.4) both include it, and the channel reliability measure L_c as

$$L_c = 4a \frac{E_c}{N_0} = 4aR_c \frac{E_b}{N_0} \tag{2.1.11}$$

The noise bilateral power spectral density is $\frac{N_0}{2}$, the fading amplitude is a, the transmitted energy per message bit and coded bit are E_c and E_b, and code rate is R_c.

3.1.3 Recursive calculation of probabilities α and β

α and β can be computed recursively, and the recursive equations are:

$$\alpha_k(t) = \sum_{t'} \alpha_{k-1}(t') \gamma_k(t', t) \quad \text{Initial conditions: } \alpha_0(t) = \begin{cases} 1 & t = 0 \\ 0 & t \neq 0 \end{cases} \tag{2.1.12}$$

$$\beta_{k-1}(t') = \sum_{t} \beta_k(t)\gamma_k(t',t) \quad \text{Initial conditions:} \quad \beta_N(t) = \begin{cases} 1 & t = 0 \\ 0 & t \neq 0 \end{cases} \tag{2.1.13}$$

$$\beta_{k-1}(t') = \sum_{t} \beta_k(t)\gamma_k(t',t) \quad \text{Initial conditions:} \quad \beta_N(t) = \begin{cases} 1 & t = 0 \\ 0 & t \neq 0 \end{cases} \tag{2.1.13}$$

3.1.4 Numerical instability

As we all know, the BCJR algorithm can cause numerical problems. In fact, the recursive computations might lead to some flaws. If we want to circumvent these flaws, normalization countermeasure is a good method. Therefore, those probabilities are supposed to be previously normalized by summing all α and β each time, rather than using α and β directly from recursive Eq. (2.1.10) and (2.1.11) into Eq. (2.1.5). As follows, the joint probability $P(t',t,\mathbf{y})$ is applied. Define the assistant non-normalized variables α' and β' at each step k:

$$\alpha'_k(t) = \sum_{t'} \alpha_{k-1}(t')\gamma_k(t',t)$$

$$\beta'_{k-1}(t') = \sum_{t} \beta_k(t)\gamma_k(t',t)$$

After computing all Q values of α' and β', take their summations: $\sum_{t} \alpha'_k(t)$ and $\sum_{t'} \beta'_{k-1}(t')$, followed by normaling α and β dividing them by these summations:

$$\alpha_k(t) = \frac{\alpha'_k(t)}{\sum_{t} \alpha'_k(t)} \tag{2.1.14}$$

$$\beta_{k-1}(t') = \frac{\beta'_{k-1}(t')}{\sum_{t'} \beta'_{k-1}(t')} \tag{2.1.15}$$

Likewise, after all $2Q$ products $\alpha_{k-1}(t')\gamma_k(t',t)\beta_k(t)$ all the summations of trellis branches have been calculated at kth time,

$$\Sigma_{Pk} = \sum_{R_0,R_1} \alpha_{k-1}(t')\gamma_k(t',t)\beta_k(t)$$

$$= \sum_{R_0} \alpha_{k-1}\left(t'\right)\gamma_k\left(t',t\right)\beta_k(t) + \sum_{R_1} \alpha_{k-1}\left(t'\right)\gamma_k\left(t',t\right)\beta_k(t)$$

will normalize $P\left(t',t,\mathbf{y}\right)$:

$$P_{norm}\left(t',t,\mathbf{y}\right) = \frac{P\left(t',t,\mathbf{y}\right)}{\Sigma_{Pk}}$$

Through this function, we can guarantee that all α, β and $P_{norm}\left(t',t,\mathbf{y}\right)$ sum to 1 at each time step all the time. These normalization summations can have no effect on the $L\left(u_k|\mathbf{y}\right)$ as all them appear both in its numerator and denominator:

final log-likelihood ratio $L\left(u_k|\mathbf{y}\right)$, because they will appear in both the numerator and denominator:

$$L\left(u_k|\mathbf{y}\right) = \ln\frac{\sum_{R_1} P\left(t',t,\mathbf{y}\right)}{\sum_{R_0} P\left(t',t,\mathbf{y}\right)} = \ln\frac{\sum_{R_1} P_{norm}\left(t',t,\mathbf{y}\right)}{\sum_{R_0} P_{norm}\left(t',t,\mathbf{y}\right)} \tag{2.1.16}$$

3.1.5 Trellis-aided calculation of α and β

Fig.4 shows Fig.3 with some new labels in. It has been previously introduced that dashed lines represent $a+1$ input bit, and solid lines represent $a-1$ input bit in our convention.

The following calculations should stand:

1. According to Eq. (2.1.10), label each trellis branch, after the value of $\gamma_k\left(t',t\right)$ computed.
2. Based on Eqs. (2.1.11) or (2.1.13), we write value of $\alpha_k(t)$ that were computed from the initial conditions $\alpha_0(t)$ in each state node.
3. According to Eqs. (2.1.12) or (2.1.14), node and below $\alpha_k(t)$ write the value of $\beta_k(t)$ that were computed from the initial conditions $\beta_N(t)$ in each state.

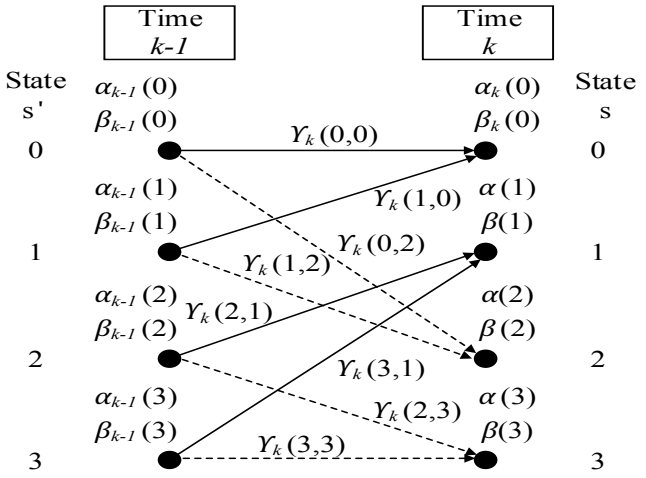

Fig.2 trellis labels

We assume $\alpha_{k-1}(t')$ to be known. The probability non-normalized $\alpha_k'(t)$ is involved by summing the products of $\alpha_{k-1}(t')$ and $\gamma_k(t',t)$ joint with the branches that can converge to t . For instance, the Fig.4 shows that at *k*th time two branches arrive $T_k = 2$. Each $Q\ \alpha_k'(t)$ should be divided by the sum of all $Q\ \alpha_k'(t)$ their sum. The procedure can be repeated until the final received sequence is reached and $\alpha_N(0)$ has been computed.

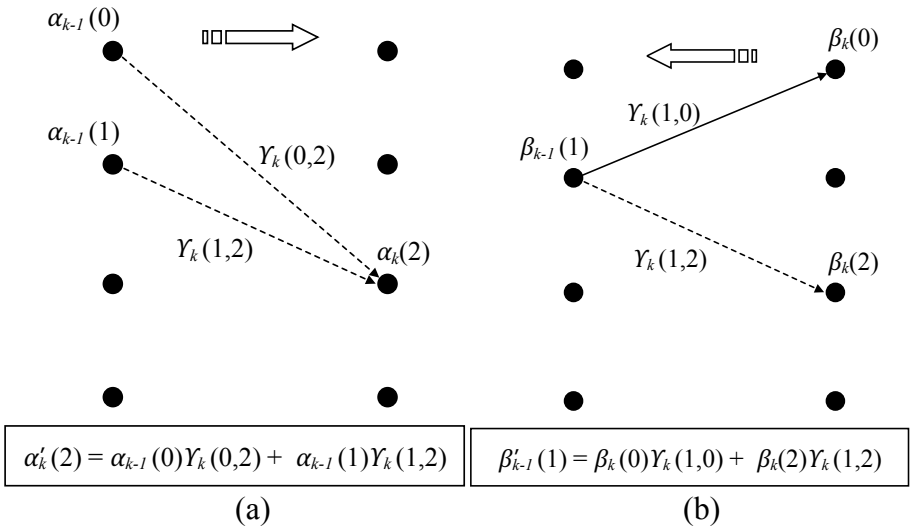

Fig.3 The function of recursive computation of α and β

Upon receiving the complete sequence $\mathbf{y}$, β can be calculated recursively. If we know $\beta_k(t)$ from the value of $\beta_{k-1}(t')$ can be computed from a similar function as $\alpha_k(t)$ was: we look for the branches leaving state $T_{k-1} = t'$, sum the corresponding products $\gamma_k(t)\ \beta_k(t)$ and divide by the sum $\sum_{t'} \beta'_{k-1}(t')$. For instance, in Fig.4 we see that two branches make the state $T_{k-1} = 1$, one pointed to state $T_k = 0$ and the other pointed to state $T_k = 2$, as Fig.5b shows.

Based on the available values of α, β and γ, the joint probability $P_{norm}(t',t,\mathbf{y}) = \alpha_{k-1}(t')\gamma_k(t',t)\beta_k(t) / \Sigma_{Pk}$ can be calculated. As shown in Fig.6, the result is $\alpha_{k-1}(1)\gamma_k(1,2)\beta_k(2)$.

We leave the *a posteriori* LLR $L(u_k|\mathbf{y})$. We consider the trellis in Fig.4: we mention that a message bit $+1$ causes the next transitions: $0 \to 2, 1 \to 2, 2 \to 3, 3 \to 3$. These are the R_1 transitions representing Eq. (2.1.4). The other four state transitions are caused by an input bit -1, described by a solid line. These are the R_0 transtitions. Thus, we joint numerator in Eq.(2.1.4) with the first four transitions and the denominator with remainings:

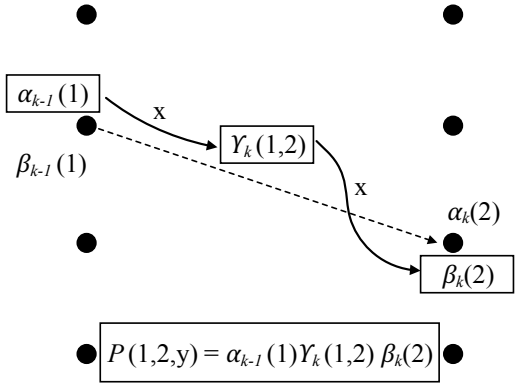

Fig.4 The probability $P(t', t, \mathbf{y})$

$$L(u_k|\mathbf{y}) = \ln \frac{\sum_{R_1} P(t',t,\mathbf{y})}{\sum_{R_0} P(t',t,\mathbf{y})}$$
$$= \ln \frac{P(0,2,\mathbf{y})+P(1,2,\mathbf{y})+P(2,3,\mathbf{y})+P(3,3,\mathbf{y})}{P(0,0,\mathbf{y})+P(1,0,\mathbf{y})+P(2,1,\mathbf{y})+P(3,1,\mathbf{y})}$$

3.2 Maximum a posteriori (MAP) algorithm

BCJR, or MAP algorithm, has a serious flaw: it calculates many multiplications. Several simplified methods have been proposed to reduce this computational complexity, for example: the Soft-Output Viterbi Algorithm (SOVA)[6], the max-log-MAP algorithm[7][8] and the log-MAP algorithm[9]. In this thesis, we use log-MAP and max-log-MAP algorithms. Three new variables, A, B and Γ, are defined:

$$\Gamma_k(t',t) = \ln \gamma_k(t',t)$$
$$= \ln \Lambda_k + \frac{u_k L(u_k)}{2} + \frac{L_c}{2}\sum_{l=1}^{n} x_{kl} y_{kl} \qquad (2.2.1)$$

$$A_k(t) = \ln \alpha_k(t)$$
$$= \max_{t}{}^{*}\left[A_{k-1}(t') + \Gamma_k(t',t)\right] \qquad A_0(t) = \begin{cases} 0 & t = 0 \\ -\infty & t \neq 0 \end{cases}$$

$$B_{k-1}(t') = \ln \beta_{k-1}(t')$$
$$= \max_{t}{}^{*}\left[B_k(t) + \Gamma_k(t',t)\right] \qquad B_N(t) = \begin{cases} 0 & t = 0 \\ -\infty & t \neq 0 \end{cases}$$

where

$$\max^{*}(a,b) = \begin{cases} \max(a,b) + \ln\left(1+e^{-|a-b|}\right) & \text{log-MAP algorithm} \\ \max(a,b) & \text{max-log-MAP algorithm} \end{cases} \qquad (2.2.2)$$

The values of the equation $\ln\left(1+e^{-|a-b|}\right)$ are often saved in a table of the eight values of $|a-b|$ between 0 and 5.

The element $\ln C_k$ expressed $\Gamma_k\left(t^{'},t\right)$ cannot be used in the $L\left(u_k|\mathbf{y}\right)$ calculation. This final expression gives the LLR

$$L\left(u_k|\mathbf{y}\right)=\max_{R_1}{}^{*}\left[A_{k-1}\left(t^{'}\right)+\Gamma_k\left(t^{'},t\right)+B_k\left(t\right)\right] \tag{2.2.3}$$

Log-MAP algorithm Eq. (2.2.3) cannot compute directly. There are more than the two variables in the part max* in Eq. (2.2.2).

The log-MAP algorithm uses exact equations so the performance is equal to the BCJR algorithm but simpler. Next, approximations are used in the max-log-MAP algorithm, therefore the performance is little worse.

3.3 The BCJR algorithm for Turbo Decoding

We consider a rate 1/n that is a systematic convolutional encoder, where x_{k1} is the first coded bit that represents the information bit u_k . In that case, the *a posterior* log-likelihood ratio $L\left(u_k|\mathbf{y}\right)$ can be made a decomposition to a sum of three different elements:

$$L\left(u_k|\mathbf{y}\right)=L\left(u_k\right)+L_c y_{k1}+L_e\left(u_k\right) \tag{2.3.1}$$

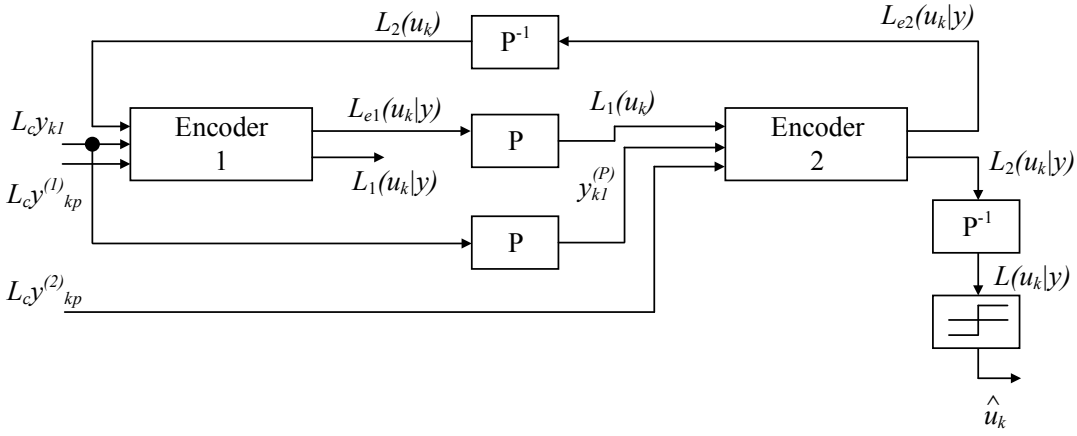

Fig.5 The diagram of a turbo decoder

The first two terms on the right side are related so that include the information bit u_k . On the other side, $L_e\left(u_k\right)$ is determined by the parity bits of codeword. This is the reason why $L_e\left(u_k\right)$ is called extrinsic information, and the extrinsic information is

an estimation of the *a priori* LLR $L(u_k)$. $L(u_k)$ and $L_c y_{k1}$ can be inputs to a MAP or another decoder, and $L(u_k|\mathbf{y})$ is its output after decoding. Through subtraction, the estimate of $L(u_k)$ can be given as:

$$L_e(u_k) = L(u_k|\mathbf{y}) - L(u_k) - L_c y_{k1} \tag{2.3.2}$$

$L(u_k)$ is considered as a more accurate estimate of the unknown *a priori* LLR, so it can be used in place of the previous value $L(u_k)$. When we repeat the former steps by a recursive way giving $L_c y_{k1}$, and the new $L(u_k) = L_e(u_k)$, inputing to another decoder with the expectation to receive a more accurate $L(u_k|\mathbf{y})$.

Eq. (2.3.1) is the foundation for recursive decoding. For the 1st recursion, the *a priori* LLR $L(u_k)$ is zero, if input bits are equally considered. $L_e(u_k)$, as the extrinsic information, will be used to update $L(u_k)$ from recursive decoder. Thus more confidence can be gained from the turbo decoder on the ± 1 hard decisions. Fig.9 gives a simplified diagram of turbo decoder block that would illuminate its procedure and how to derive an interleaved sequence $\mathbf{u}^{(P)}$ from $\mathbf{u}$ (we compute $\mathbf{u}^{(P)} = \mathbf{u}_{P_i}$). We can simply compute $\mathbf{w}_{P_i}^{(P^{-1})} = \mathbf{w}_i$ deinterleaved sequence $\mathbf{w}$.

The recursive decoding steps as follows:

1. For the 1st recursion, assume $L(u_k) = 0$ is assumed, then the systematic extrinsic information $L_{e1}(u_k|\mathbf{y})$ are received from decoder 1 outputs, gathered from the first parity bit.
2. Upon interleaving the extrinsic information $L_{e1}(u_k|\mathbf{y})$ that comes from decoder 1, computed from Eq. (2.3.2). $L_1(u_k)$ is delivered to decoder 2 as more educated guess on $L(u_k)$. Then decoder 2 gives $L_{e2}(u_k|\mathbf{y})$, which is extrinsic information based on the other parity bit. According to suitable deinterleaving, this information is passed on to decoder 1 to be defined $L_2(u_k)$, a newer guess on $L(u_k)$.

3. Upon the log-likelihood $L_2(u_k|\mathbf{y})$ as the output of decoder 2 is deinterleaved, after the required amount of recursion has been met, it is delivered as $L(u_k|\mathbf{y})$ to estimates the information bit, in the hard decision device, based only on the sign of the deinterleaved LLR,

$$\hat{u}_k = sign\left[L(u_k|\mathbf{y})\right] = sign\left\{P^{-1}\left[L_2(u_k|\mathbf{y})\right]\right\} \tag{2.3.3}$$

3.4 Simulation

In this part, we analyze the results through simulation. In all the simulations, we assume the symbol length is N = 1024, and the CP length is P = 128, and a length G + 1 vector which has i.i.d. complex Gaussian distributed elements is the channel CIR. For all the BER figures, the system is turbo coded and MAP decoder.

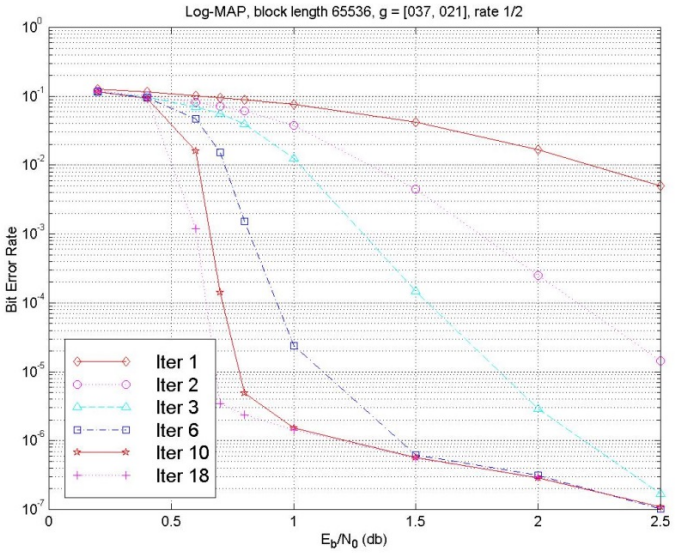

Fig. 6 Log-MAP block length 65536

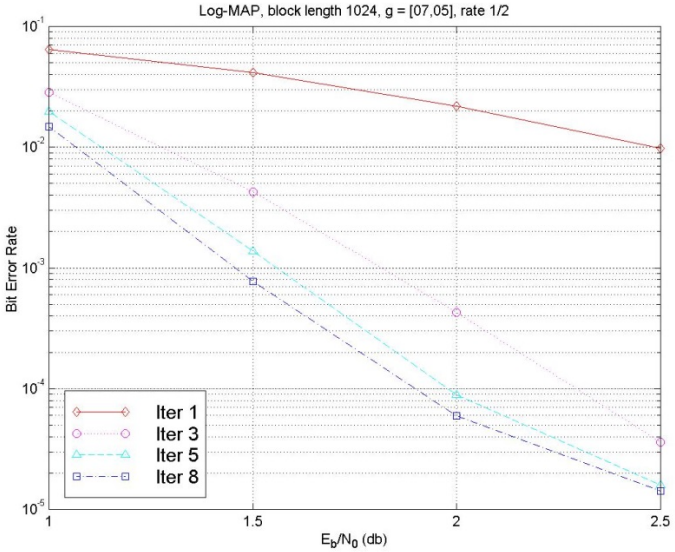

Fig. 7 Log-MAP block length 1024

Chapter 4

VECTOR OFDM(V-OFDM)

4.1 Introduction of V-OFDM

Orthogonal frequency division multiplexing (OFDM)[10][11], a kind of low complexity transmission scheme for multipath channels, which is prevalently used in wireless communication systems of the next generation.

Without coding, OFDM cannot expressed the multi-path diversity, so it performs not as good as a single carrier transmission which has time domain equalizers. For the receiver part, FFT and channel equalization in the frequency domain first occurred in Single-Carrier Frequency Domain Equalization (SC-FDE)[12]-[15], and then IFFT and demodulation or detection are used in the time domain. It is convinced that we use OFDM to deal with cost transceivers problems and high data rate, however, SC-FDE is always used into cost transceivers problems and low data rate.

Vector OFDM (V-OFDM), for reducing the cyclic prefix(CP) and a single transmit antenna OFDM system's IFFT size, which first proposed by Xia[17]. This method generates from original OFDM, V-OFDM is a bridge between SC-FDE and OFDM. Through calibrating parameters, V-OFDM system can be tuned in different system design. In paper[18], regarding to different aspects of V-OFDM system design, the authors analyzed the different synchronization and guard band. In paper[20] , the authors described the vector channel allocation. In paper[19], the turbo principle for recursive demodulation and decoding are exploited. In paper[21], different vector blocks (VB) shows us different performances. In paper[22], totally investigated the V-OFDM performance using multipath Rayleigh fading channel. In paper[23], the

linear receivers for V-OFDM are introduced, involving Zero-forcing and MMSE receivers.

4.2 Vector OFDM System

In V-OFDM system, the block-by-block are used for the modulated symbols. We assume that, for one block, the modulated system are $N = LM$, and define them as $\{x_n\}_{n=1}^{N}$. Difference between the V-OFDM and conventional OFDM is that the length N block into L vector blocks (VBs) are further divided in V-OFDM, in which each VB has size M . Denote the lth VB as

$$\mathbf{x}_l = \left[x_{lM}, x_{lM+1}, \ldots, x_{lM+M}\right], \qquad l = 1,2,\ldots,L.$$

$\mathbf{x}_l$ is the lth transmit for a VB. Unlike the FFT size N as in original OFDM, the size L vector IFFT is over the VBs in V-OFDM and can be calculated:

$$\overline{\mathbf{x}}_q = \frac{1}{L}\sum_{l=1}^{L}\mathbf{x}_l e^{j\frac{2\pi ql}{L}}, \qquad q = 1,2,\ldots,L. \tag{3.2.1}$$

$\overline{\mathbf{x}}_q$ is a M size column vector and is expressed as

$$\overline{\mathbf{x}}_q = \left[\overline{x}_{qM}, \overline{x}_{qM+1}, \ldots, \overline{x}_{qM+M}\right]^T.$$

Rewrite the vectors $\left\{\overline{\mathbf{x}}_q\right\}_{q=1}^{L}$, as a N size row vector, which is

$$\left[\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_N\right] = \left[\overline{\mathbf{x}}_1^T, \overline{\mathbf{x}}_2^T, \ldots, \overline{\mathbf{x}}_L^T\right].$$

For conventional OFDM, cyclic prefix (CP) is added to the size-N row vector. We assume that the length of CP is η , η should satisfy $\eta > G$ in order to avoid the interblock-interference, which G is the maximum delay. If there is not loss of generality, the VB size M is a multiple of $\eta = GM$. Next, in the time domain, the transmitted sequence, after adding CP, can be written as:

$$\left[\overline{\mathbf{x}}_{L-K}, \overline{\mathbf{x}}_{L-K+1}, \ldots, \overline{\mathbf{x}}_{L-1}, \overline{\mathbf{x}}_1, \overline{\mathbf{x}}_2, \ldots \overline{\mathbf{x}}_L\right].$$

For the receiver part, after we remove CP, the received signal become a kind of circular convolution for the transmitted signal as well as the Channel Impulse Response (CIR). It can be defined as:

$$\overline{y}_n = \sum_{d=1}^{D+1} h_d \overline{x}_{(n-d)_N} + \omega_n \ , \qquad n = 1,2,\ldots,N \tag{3.2.2}$$

where $\{h_d\}_{d=1}^{D+1}$ is the CIR, $\omega_n \sim C\mathcal{N}(0,\sigma^2)$ is the additive white Gaussian noise(AWGN) , and $(n)_N$ represents n mod N . The receiver is divided by the size N of block $[\overline{y}_1, \overline{y}_2, \ldots, \overline{y}_N]$ into L size M column vectors $\{\overline{\mathbf{y}}_q\}_{q=1}^{L}$, when $\overline{\mathbf{y}}_q$ is

$$\overline{\mathbf{y}}_q = \left[\overline{y}_{qM}, \overline{y}_{qM+1} \ldots, \overline{y}_{qM+M}\right]^T .$$

Letting component-wise vector FFT of size L , we have

$$\mathbf{y}_l = \sum_{q=1}^{L} \overline{\mathbf{y}}_q e^{-j\frac{2\pi ql}{L}} \ , \qquad l = 1,2,\ldots,L \tag{3.2.3}$$

Write the M length column vector $\mathbf{y}_l$ as

$$\mathbf{y}_l = \left[y_{lM}, y_{lM+1}, \ldots, y_{lM+M}\right]^T .$$

We call $\mathbf{y}_l$ the lth receive VB.

Denote

$$H_k = \sum_{d=1}^{D+1} h_d e^{-j\frac{2\pi kd}{N}} \ , \qquad k = 1,2,\ldots,N \tag{3.2.4}$$

the frequency domain channel coefficient at the kth subcarrier in conventional OFDM using the length of N FFT/IFFT to deal with. The diagonal matrix that the size is $M \times M$ $\overline{\mathbf{H}}_l$ can be defined as

$$\overline{\mathbf{H}}_l = diag\{H_l, H_{l+L}, \ldots, H_{l+ML}\} . \tag{3.2.5}$$

We Assume the perfect synchronization, through some signal processing functions, the relation of the transmit VB $\mathbf{x}_l$ and the receive VB $\mathbf{y}_l$ is

$$\mathbf{y}_l = \mathbf{H}_l \mathbf{x}_l + \mathbf{w}_l , \qquad l = 1,2,\ldots,L \tag{3.2.6}$$

where $\mathbf{w}_l = \left[w_{l,1}, w_{l,2}, \ldots, w_{l,M}\right]^T$ is the noise vector, they are i.i.d. and $C\mathcal{N}\left(0,\sigma^2\right)$ distributed, and we can express the equivalent channel matrix as:

$$\mathbf{H}_l \triangleq \mathbf{U}_l^H \bar{\mathbf{H}}_l \mathbf{U}_l$$

where $\mathbf{U}_l$ is unitary matrix, for entry in the sth row($s = 1,2,\ldots,M$) and mth ($m = 1,2,\ldots,M$) column equaling

$$\left[\mathbf{U}_l\right]_{s,m} = \frac{1}{\sqrt{M}} \exp\left(-j\frac{2\pi(l+sL)m}{N}\right).$$

The product of a DFT matrix can be easily verified, and a diagonal matrix is able to compose $\mathbf{U}_l$.

We assume that $E\left\{\left|x_n\right|^2\right\} = 1$, $n = 1,2,\ldots N$, and define the transmit SNR is $\rho \triangleq \frac{1}{\sigma^2}$.

Though the CP overhead of the V-OFDM is unaffected, the size of IFFT can be reduced by M times, from N to L .

4.3 Detection Algorithm

Here, we discuss two detection algorithms (ML and MMSE) for V-OFDM system, and assume the channel is perfect and is well-known to the receiver.

4.3.1 ML Detection

According to equation (3.2.6), ML detection is

$$\hat{\mathbf{x}}_l^{ML} = \arg\min_{\mathbf{x}_l} \left\|\mathbf{y}_l - \mathbf{H}_l \mathbf{x}_l\right\|^2, \qquad l = 1,2,\ldots,L \tag{3.3.1}$$

The ML-V-OFDM's computational complexity can grow exponentially because of the M size VB and the high modulation order. Previous papers[21], mentioned that each VB is of different performance, so they proposed a new function In other papers[17][22], the performance of ML-V-OFDM was analyzed, showing if we want to achieve maximum diversity order, we can use the major VBs, which equals $\min\{M,\ D+1\}$.

4.3.2 MMSE Detection

We assume that channel coefficients and noise are known by receiver. Therefore, MMSE detection can be used to improve the performance. We can calculate the weight matrix of MMSE as

$$C_l^{MMSE} \triangleq \left(\mathbf{H}_l^H \mathbf{H}_l + \rho^{-1}\mathbf{I}\right)^{-1} \mathbf{H}_l^H$$
$$= \mathbf{U}_l^H \, diag\left\{\frac{H_l^*}{\left|H_l\right|^2 + \rho^{-1}}, \ldots, \frac{H_{l+ML}^*}{\left|H_{l+ML}\right|^2 + \rho^{-1}}\right\}\mathbf{U}_l \qquad (3.3.2)$$

According to the second equation, $\mathbf{H}_l$ denote the special structure. After MMSE filtering, we have

$$\hat{\mathbf{y}}_l^{MMSE} = \mathbf{C}_l^{MMSE}\mathbf{y}_l$$
$$= \mathbf{U}_l^H \, diag\left\{\frac{H_l^*}{\left|H_l\right|^2 + \rho^{-1}}, \ldots, \frac{H_{l+ML}^*}{\left|H_{l+ML}\right|^2 + \rho^{-1}}\right\}\mathbf{U}_l\mathbf{x} + \mathbf{C}_l^{MMSE}\mathbf{w}$$
$$\triangleq \left[\hat{y}_{lM}^{MMSE}, \hat{y}_{lM+1}^{MMSE}, \ldots, \hat{y}_{lM+M}^{MMSE}\right]^T \qquad (3.3.3)$$

and the mth element of $\hat{\mathbf{y}}_l^{MMSE}$ can be written as

$$\hat{y}_{lM+m}^{MMSE} = C_{lM+m}x_{lM+m} + n_{lM+m} \qquad (3.3.4)$$

where

$$C_{lM+m} = \frac{1}{M}\sum_{m=0}^{M}\frac{\left|H_{l+mL}\right|^2}{\left|H_{l+mL}\right|^2 + \rho^{-1}}$$

Then, the symbol-by-symbol detection is

$$\hat{x}_n^{MMSE} = \arg\min_{x_n}\left|\hat{y}_n^{MMSE} - C_n x_n\right|^2 \qquad (3.3.5)$$

$$E\left[\hat{y}_{lM+m}^{MMSE}\left(\hat{y}_{lM+m}^{MMSE}\right)^*\right] = \left[E\left[\hat{\mathbf{y}}_l^{MMSE}\left(\hat{\mathbf{y}}_l^{MMSE}\right)^H\right]\right]_{m,m}$$
$$= \frac{1}{M}\sum_{k=1}^{M}\frac{\left|H_{l+kL}\right|^2}{\left|H_{l+kL}\right|^2 + \rho^{-1}} \qquad (3.3.6)$$

We can see that it is independent of m , and the detection SNR or Signal to Interference plus Noise Ratio(SINR) for the mth element in the lth VB $\mathbf{x}_l$ is

$$\rho_{l,m}^{MMSE} \triangleq \frac{\left|C_{lM+m}\right|^2}{E\left[\left|n_{lM+m}\right|^2\right]} = \left[\frac{1}{M}\sum_{k=1}^{M}\frac{1}{\rho\left|H_{l+kL}\right|^2+1}\right]^{-1} - 1 \qquad (3.3.7)$$

and $\rho_{l,1}^{MMSE} = \rho_{l,2}^{MMSE} = \cdots = \rho_{l,M}^{MMSE} \triangleq \rho_l^{MMSE}$.

Based on these analysis, for MMSE-V-OFDM, different elements in the same VB have the same detection SNR.

4.4 Simulation

In this section, we simulate to analyze the numerical model. For all the simulations, the parameters are that the CP length that P is equal to 128, the V-OFDM symbol length N = 1024, and the length G + 1 vector which has i.i.d. complex Gaussian distributed elements is the channel CIR.

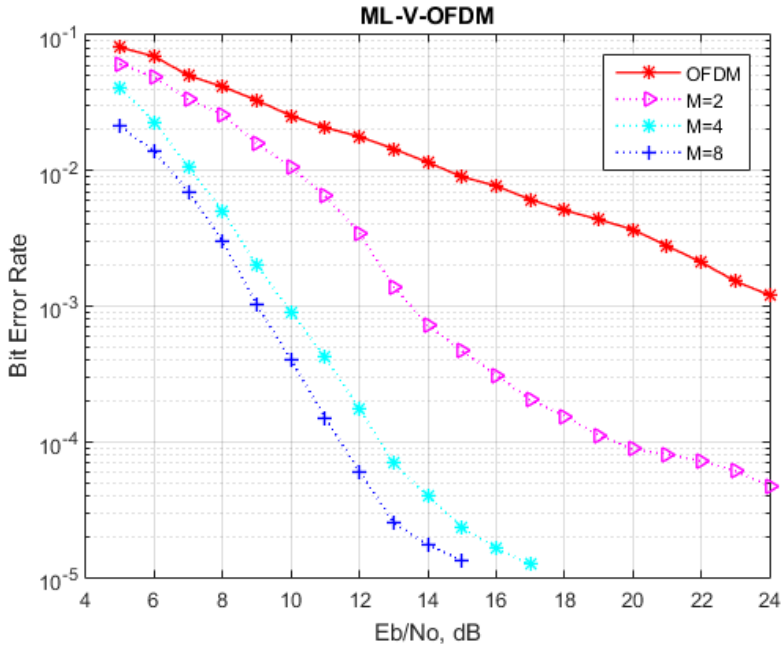

Fig.8 Different vector size M=2,4,8 by ML decoding

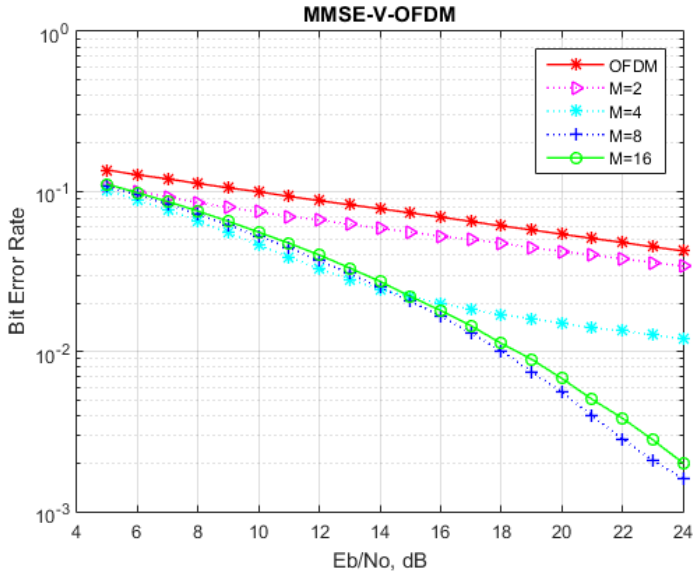

Fig.9 Different vector size M=2,4,8,16 by MMSE decoding

Chapter 5

TURBO RECEIVER OF V-OFDM

5.1 Introduction of turbo receiver

There is a serious problem for many practical communication systems is ISI channel for data transmission. If we want to protect the transmitted data, using an error correction code (ECC) can control amount of redundancy.

The received symbols have been equalized or detected in order to reduce the ISI influence. The equalizer gives an estimation on the data, and composed by linear processing of the received signal and past symbol estimates. A variety of optimization criteria can choose the parameters of these filters.

For minimizing the bit error rate (BER), we use optimal equalization methods to solve the problem. Because the sequence error rate are always nonlinear, maximum-likelihood (ML) estimation is a good function, which, in the presence of *a priori* information about the transmitted data, transforms into the maximum estimation of *a posteriori* probability (MAP). There are some other algorithms are useful, for example, MAP/ML estimation, the Viterbi algorithm (VA)[24][25] and the BCJR algorithm.

The most important advantage for BER is that it is possible to use coded data transmission between the equalizer and the decoder, in terms of soft information as oppose to hard information, which reduces the BER, however vastly relies on more complicated decoding algorithms. Joint is used by the top systems for many different communication systems, for ML equalizers and convolutional codes, when we use interleaver after encoding and deinterleaver before decoding[26][27]. Error will be introduced by the equalizer between adjacent symbols are decorrelated through

interleaving shuffling symbols within a given time frame or block of data. Errors of this kind are difficult to solve by only a convolutional decoder. There are applications hope to use coding to solve the flaws of the chosen equalizer. Error propagation should be resolved by DFE and a kind of high-rate code [28].

For the receiver part, because of complexity, it is impossible to use an optimal joint processing of the equalization and decoding. Some recursive receiver algorithms repeat the equalization and decoding tasks on the same set of received data, where the decoder feedback information is individual for the equalization process. This method was called to turbo receiver, was derived for concatenated convolutional codes and is now widely adapted to many communication problems, like code division multiple access (CDMA) [31] and trellis coded modulation (TCM)[29][30]. Turbo receiver was first proposed in paper[35] and then developed from serval researchers[36][37]. Among all the systems, MAP techniques are applied successfully for receiver and decoding[36][37]. Combined turbo coding and receiver[38][39] involves three or more parts: two or more coding parts like the original turbo coding applications and the channel equalizer.

5.2 Basic principle of MMSE Turbo Receiver

An equivalent time-invariant G tap-coefficients ISI channel is considered. The signal model can be expressed [40][41] as

$$\mathbf{z}_n = \mathbf{H}\mathbf{x}_n + \mathbf{w}_n \tag{4.2.1}$$

where $\mathbf{z}_n$ is the channel observation, $\mathbf{x}_n$ the transmitted signal, $\mathbf{H}$ the channel convolution matrix and $\mathbf{w}_n$ an appropriate-sized vector of additive white Gaussian noise(AWGN) with variance $\sigma^2 = N_0 / 2$ per dimension. $N_2 + N_1 + 1$ is called window size.

A turbo equalizer is based on an SISO equalizer, functioning in a recursive way. The MMSE linear equalizer(LE) discussed in paper[40][41] is considered for the SISO

equalization task. The MMSE LE computes the extrinsic output $L_E(x_n)$ for $x_n \in \{+1,-1\}$ based on $\{L(x_n)\}$ and the channel observation

$$\mathbf{z}_n = \left[z_{n-N_2}, z_{n-N_2+1}, \ldots, z_{n+N_1} \right]^T \tag{4.2.2}$$

where N_1 and N_2 are the non-causal and causal lengths of the filter, respectively. The estimated mean and variance of x_n are

$$\overline{x}_n = E(x_n) = \tanh\left(\frac{L(x_n)}{2} \right) \tag{4.2.3}$$

$$v_n = Var(x_n) = 1 - \overline{x}_n^2 \tag{4.2.4}$$

Let $\mathbf{0}_{1\times i}$ represents a length-i zero row vector and $\mathbf{I}$ represents an identity matrix of an appropriate size. The output of the MMSE LE developed is given by

$$L_E(x_n) = \frac{2\mathbf{c}_n^T \left(\mathbf{z}_n - \mathbf{H}\overline{\mathbf{x}}_n + \overline{\mathbf{x}}_n \mathbf{s}_n \right)}{1 - \mathbf{s}_n^T \mathbf{c}_n} \tag{4.2.5}$$

where

$$\overline{\mathbf{x}}_n = \left[\overline{x}_{n-N_2-G+1}, \overline{x}_{n-N_2-G+2}, \ldots, \overline{x}_{n+N_1} \right]^T$$

$$\mathbf{V}_n = diag\left(v_{n-N_2-G+1}, v_{n-N_2-G+2}, \ldots, v_{n+N_1} \right)$$

$$\mathbf{s}_n = \mathbf{H}\left[\mathbf{0}_{1\times(N_2+G-1)} \quad 1 \quad \mathbf{0}_{1\times N_1} \right]^T$$

$$\mathbf{c}_n = \left(\sigma^2 \mathbf{I} + \mathbf{H}\mathbf{V}_n\mathbf{H}^T + (1-v_n)\mathbf{s}_n\mathbf{s}_n^T \right)^{-1} \mathbf{s}_n$$

5.3 Turbo Receiver of V-OFDM

In [19], the authors used the iterative decoding on V-OFDM, but they did not use turbo coded receiver to deal with this problem. So, I updated their model to solve the receiver.

After turbo encoder, the length N of information symbol u, through the turbo encoder rate R, we get the length $C = N / R$ of binary coded sequence, $s = \{s_1, s_2, \ldots, s_C\}$. After puncturing, we get $s^{(P)} = \left\{ s_1^{(P)}, s_2^{(P)}, \ldots, s_C^{(P)} \right\}$. Using F -map modulation, the sequence $\mathbf{s}^{(P)}$ becomes $\mathbf{x} = \{x_1, x_2, \ldots, x_K\}, K = C / \log_2^F$.

We assume the rate R is $1/2$, through V-OFDM system, the transmitted information x divided by $\overline{M} \times L$, where $\overline{M} = 2M$. The length $2N$ block into L VBs, where each VB has the size $\overline{M}$,

$$\mathbf{x}_l = \left[x_{2l\overline{M}}, x_{2l\overline{M}+1}, \ldots, x_{2l\overline{M}+\overline{M}} \right]^T, \qquad l = 1,2,\ldots,L$$

We call the lth transmit VB is $\mathbf{x}_l$. And the system channel model,

$$\mathbf{y}_l = \mathbf{H}\mathbf{x}_l + \mathbf{w}_l \tag{4.3.1}$$

where $\mathbf{y}_l = \left[y_{2l\overline{M}}, y_{2l\overline{M}+1}, \ldots, y_{2l\overline{M}+\overline{M}} \right]^T$, $\mathbf{w}_l = \left[w_{l,1}, w_{l,2}, \ldots, w_{l,\overline{M}} \right]^T$ is the noise vector, whose entries are i.i.d. and $CN\left(0,\sigma^2\right)$ distributed. According to each V-OFDM We block, we use SISO MMSE decoding. The total length of filter is $\overline{M} = N_c$, the received symbol block is $\left\{ y_{l,1}, y_{l,2}, \ldots, y_{l,N_c} \right\}$.

We define the variance of information block $\mathbf{x}_l$ is $\mathbf{v}_l$, and the average is $\overline{\mathbf{x}}_l$. $\mathbf{x}_l$ matches the sequence $\mathbf{s}_l^{(P)} = \left\{ s_{l,m}^{(P),1}, s_{l,m}^{(P),2}, \ldots, s_{l,m}^{(P),Q} \right\}$, $Q = \log_2^F$. The numerical value set of $\mathbf{x}_l$ is $\mathbf{A} = \left\{ a_1, a_2, \ldots, a_F \right\}$, the average and variance are

$$\overline{x}_{l,m} = E\left(x_{l,m}\right) = \sum_{a_i \in A} a_i \cdot P\left(x_{l,m} = a_i\right) \tag{4.3.2}$$

$$v_{l,m} = Cov\left(x_{l,m}, x_{l,m}\right) = \left(\sum_{a_i \in A} \left|a_i\right|^2 \cdot P\left(x_{l,m} = a_i\right) \right) - \left|\overline{x}_{l,m}\right|^2 \tag{4.3.3}$$

where $P\left(x_{l,m} = a_i\right)$ is given by *a priori* information $L_e^D\left(s_{l,m}^{(P),k}\right), k = 1,\ldots,Q$, we assume the value a_i matches $s_{l,m}^{(P)}$ is $\mathbf{d}_i = \left\{ d_{i,1}, d_{i,2}, \ldots, d_{i,Q} \right\}$.

$$P\left(x_{i,j} = a_i\right) = \prod_{k=1}^{Q} P\left(s_{l,m}^{(P),k} = d_{i,k}\right) = \prod_{k=1}^{Q} \frac{1}{2}\left(1 + \tilde{d} \cdot \tanh\left(L_e^D\left(s_{l,m}^{(P),k}\right)/2\right)\right) \tag{4.3.4}$$

where

$$\tilde{d}_{i,k} = \begin{cases} +1 & d_{i,k} = 1 \\ -1 & d_{i,k} = 0 \end{cases} \tag{4.3.5}$$

Different from single carrier system, V-OFDM is a kind of multiple carriers system. We cannot use the SISO MMSE decoding, so we can use the result of chapter 3.3.2 MMSE Detection.

$$\hat{x}_n = \arg\min_{x_n} \left| \hat{y}_n - C_n x_n \right|^2 \tag{4.3.6}$$

$$E\left[\hat{y}_{l\overline{M}+m} \left(\hat{y}_{l\overline{M}+m} \right)^* \right] = \left[E\left[\hat{\mathbf{y}}_l \left(\hat{\mathbf{y}}_l \right)^H \right] \right]_{m,m}$$

$$= \frac{1}{\overline{M}} \sum_{m=1}^{\overline{M}} \frac{\left| H_{l+mL} \right|^2}{\left| H_{l+mL} \right|^2 + \rho^{-1}} \tag{4.3.7}$$

$$\rho_{l,m} \triangleq \frac{\left| C_{l\overline{M}+m} \right|^2}{E\left[\left| n_{l\overline{M}+m} \right|^2 \right]} = \left[\frac{1}{\overline{M}} \sum_{m=1}^{\overline{M}} \frac{1}{\rho \left| H_{l+mL} \right| + 1} \right]^{-1} - 1$$

where $\hat{x}_{l,m}$ is Gaussian distribution, the average is $s'_{l,m} = E\left(\hat{x}_{l,m} \middle| x_{l,m} = a_i \right), i = 1,\ldots,F$, the variance is $\sigma'^2_{l,m} = Cov\left(\hat{x}_{l,m}, \hat{x}_{l,m} \middle| x_{l,m} = a_i \right)$, the extrinsic information is $s^{(P),k}_{l,m}, k = 1,\ldots,Q$

$$L_e\left(s^{(P),k}_{l,m} \right) = \ln \frac{\sum_{\forall \mathbf{d}_i : d_{i,k}=1} P\left(\hat{x}_{l,m} \middle| \mathbf{s}^{(P)}_{l,m} = \mathbf{d}_i \right) \prod_{\forall k' : k' \neq k} P\left(s^{(P),k'}_{l,m} = d_{i,k'} \right)}{\sum_{\forall \mathbf{d}_i : d_{i,k}=0} P\left(\hat{x}_{l,m} \middle| \mathbf{s}^{(P)}_{l,m} = \mathbf{d}_i \right) \prod_{\forall k' : k' \neq k} P\left(s^{(P),k'}_{l,m} = d_{i,k'} \right)}$$

$$= \ln \frac{\sum_{\forall \mathbf{d}_i : d_{i,k}=1} \exp\left(-\frac{\left| \hat{x}_{l,j} - \mu'_{l,j} \right|^2}{2\sigma'^2_{l,j}} + \sum_{\forall k' : k' \neq k} \tilde{d}_{i,k'} L\left(s^{(P),k'}_{l,m} \right) / 2 \right)}{\sum_{\forall \mathbf{d}_i : d_{i,k}=0} \exp\left(-\frac{\left| \hat{x}_{l,j} - \mu'_{l,j} \right|^2}{2\sigma'^2_{l,j}} + \sum_{\forall k' : k' \neq k} \tilde{d}_{i,k'} L\left(s^{(P),k'}_{l,m} \right) / 2 \right)}$$

$$\approx \max_{\forall \mathbf{d}_i : d_{i,k}=1} \left(-\frac{\left| \hat{x}_{l,m} - \mu'_{l,m} \right|^2}{2\sigma'^2_{l,m}} + \sum_{\forall k' : k' \neq k} \tilde{d}_{i,k'} L\left(s_{l,m}^{(P),k'} \right) / 2 \right)$$

$$- \max_{\forall \mathbf{d}_i : d_{i,k}=0} \left(-\frac{\left| \hat{x}_{l,m} - \mu'_{l,m} \right|^2}{2\sigma'^2_{l,m}} + \sum_{\forall k' : k' \neq k} \tilde{d}_{i,k'} L\left(s_{l,m}^{(P),k'} \right) / 2 \right) \tag{4.3.8}$$

where the input of SISO Turbo decoder comes from $\left\{ L_e\left(s_{l,m}^{k} \right) \right\}$ that deinterleaving from $\left\{ L_e\left(s_{l,m}^{(P),k} \right) \right\}$. When decoding, first we need to update soft information of information bit u_k,

$$\gamma_k\left(t', t\right) = L_2\left(u_k\right) u_k + L_e\left(s_{2k-1}\right) s_{2k-1} + L_e\left(s_{2k}\right) s_{2k}$$

$$- \ln\left(1 + e^{L_2(u_k)}\right) - \ln\left(1 + e^{L_e(s_{2k-1})}\right) - \ln\left(1 + e^{L_e(s_{2k})}\right) \tag{4.3.9}$$

$$\alpha_k\left(t\right) = \ln\left(\sum_{t'} \exp\left(\alpha_{k-1}\left(t'\right) + \gamma_k\left(t', t\right) \right) \right) \tag{4.3.10}$$

$$\beta_{k-1}\left(t'\right) = \ln\left(\sum_{t} \exp\left(\beta_k\left(t\right) + \gamma_k\left(t', t\right) \right) \right) \tag{4.3.11}$$

$$L\left(u_k \middle| \mathbf{y}\right) \triangleq \ln \frac{\sum_{R_1} \exp\left(\alpha\left(t'\right) + \gamma\left(t', t\right) + \beta\left(t\right) \right)}{\sum_{R_0} \exp\left(\alpha\left(t'\right) + \gamma\left(t', t\right) + \beta\left(t\right) \right)} \tag{4.3.12}$$

When the number of recursive decoding times come to the maximum, we need to calculate the soft information of coded sequence $\left\{ s_i \right\}$,

$$L(s_{2k-1}) \triangleq \ln \frac{\sum_{R_1} \exp\left(\alpha(t') + \gamma(t',t) + \beta(t)\right)}{\sum_{R_0} \exp\left(\alpha(t') + \gamma(t',t) + \beta(t)\right)} \tag{4.3.13}$$

$$L(s_{2k}) \triangleq \ln \frac{\sum_{R_1} \exp\left(\alpha(t') + \gamma(t',t) + \beta(t)\right)}{\sum_{R_0} \exp\left(\alpha(t') + \gamma(t',t) + \beta(t)\right)} \tag{4.3.14}$$

$$L_d(s_{2k-1}) = L(s_{2k-1}) - L_e(s_{2k-1}) \tag{4.3.15}$$

$$L_d(s_{2k}) = L(s_{2k}) - L_e(s_{2k}) \tag{4.3.16}$$

where $L_d(s_{2k})$ and $L_d(s_{2k-1})$ are the output soft information of decoder. Through the interleaver, it will come back to SISO receiver to complete recursion.

5.4 Simulation

For simulated TC-V-OFDM systems, the subcarriers N is 1024, in order that simplifying the illustration, no guard band uses. The vector length called M defines from 4,8,16 to 32, and size L is the vector FFT length. We chose a rate $R = 1/2$, turbo encoder is $\left(1 + D^2\right) / \left(1 + D + D^2\right)$. The interleaver is random, but we use the same interleaver for all numerical simulations. For each OFDM block, there are 100 vector OFDM blocks and 12800 code bits in that. The AWGN channel which is i.i.d. and zero-mean complex Gaussian random distribution.

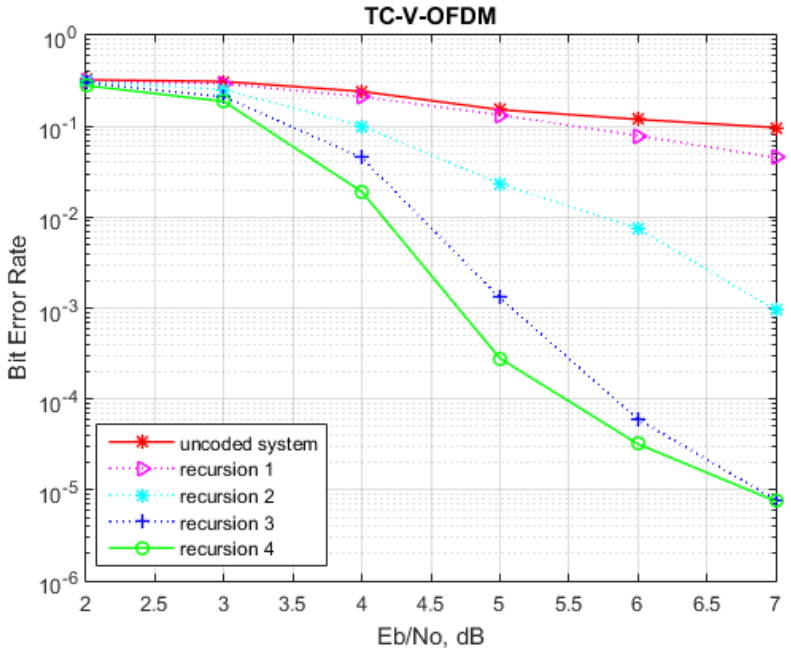

Fig.10 TC-V-OFDM for series input

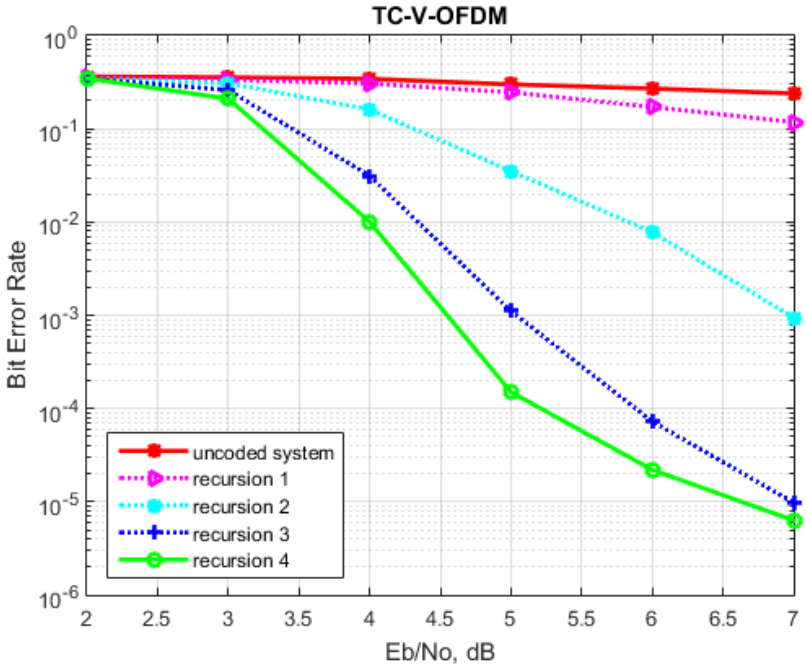

Fig.11 TC-V-OFDM for parallel input

REFERENCES

[1] Shannon, C.E. (1948), "A Mathematical Theory of Communication", Bell System Technical Journal, 27, pp. 379–423 & 623–656, July & October, 1948.

[2] C. Berrou, A. Glavieux and P. Thitimajshima, “Near Shannon limit error-correcting coding and decoding: Turbo codes”, Proc. Intern. Conf. Communications (ICC), Geneva, Switzerland, pp. 1064– 1070, May 1993.

[3] S. Dolinar and D. Divsalar, “Weight distributions for turbo codes using random and nonrandom permutations”, JPL TDA Progress Rep., 1995, 42(8).

[4] S. Dolinar, D. Divsalar and F. Pollara,”Code Performance as a Function of Block Size”, The Telecommunications and Mission Operations Progress Report, TMO PR 42-133, January-March 1998, pp. 1-23

[5] L. R. Bahl, J. Cocke, F. Jelinek and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate”, IEEE Trans. on Information Theory, pp. 284-287, March 1974.

[6] J. Hagenauer and P. Hoeher, “A Viterbi Algorithm With Soft-Decision Outputs and Its Applications”, Proceedings of GLOBECOM ’89, Dallas, Texas, pp. 47.1.1-47.1.7, November 1989.

[7] W. Koch and A. Baier, “Optimum and sub-optimum detection of coded data disturbed by time- varying inter-symbol interference,” Proceedings of IEEE Globecom, pp. 1679–1684, December 1990.

[8] J. A. Erfanian, S. Pasupathy and G. Gulak, “Reduced complexity symbol detectors with parallel structures for ISI channels,” IEEE Trans. Communications, vol. 42, pp. 1661–1671, 1994.

[9] P. Robertson, E. Villebrun and P. Hoeher, “A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain”, Proc. Intern. Conf. Communications (ICC), pp. 1009–1013, June 1995.

[10] L. J. Cimini, “Analysis and simulation of a digital mobile channel using orthogonal frequency division multiplexing,” IEEE Trans. Commun, vol. COM-33, no. 7, pp. 665–675, Jul. 1985.

[11] J. A. C. Bingham, “Multicarrier modulation for data transmission: An idea whose time has come,” IEEE Commun. Mag., vol. 28, no. 5, pp. 5–14, May 1990.

[12] H. Sari, G. Karam, and I. Jeanclaude, “Transmission techniques for digital terrestrial TV broadcasting,” IEEE Commun. Mag., pp. 100–109, Feb. 1995.

[13] M. V. Clark, “Adaptive frequency-domain equalization and diversity combining for broadband wireless communications,” IEEE J. Sel. Areas Commun., vol. 16, pp. 1385–1395, Oct. 1998.

[14] N. Al-Dhahir, “Single-carrier frequency-domain equalization in frequency-selective fading channels,” IEEE Commun. Lett., vol. 7, no. 7, pp. 304–306, Jul. 2001.

[15] D.Falconer, S.L.Ariyavisitakul, A.Benyamin-Seeyar, and B.Eidson, “Frequency domain equalization for single-carrier broadband wireless systems,” IEEE Commun. Mag., vol. 40, no. 4, pp. 58–66, Apr. 2002.

[16] C. Ciochina and H. Sari, “A review of OFDMA and single-carrier FDMA,” in Proc. 2010 Eur. Wireless Conf., Apr. 12–15, 2010, pp. 706–710.

[17] X.-G. Xia, “Precoded and vector OFDM robust to channel spectral nulls and with reduced cyclic prefix length in single transmit antenna systems,” IEEE Trans. Commun., vol. 49, no. 8, pp. 1363–1374, Aug. 2001.

[18] H. Zhang, X.-G. Xia, L. J. Cimini, and P. C. Ching, “Synchronization techniques and guard-band-configuration scheme for single-antenna vector-OFDM systems,” IEEE Trans. Wireless Commun., vol. 4, no. 5, pp. 2454–2464, Sep. 2005.

[19] H. Zhang and X.-G. Xia, “Iterative decoding and demodulation for single-antenna vector OFDM systems,” IEEE Trans. Veh. Technol., vol. 55, no. 4, pp. 1447–1454, Jul. 2006.

[20] H. Zhang, X.-G. Xia, Q. Zhang, and W. Zhu, “Precoded OFDM with adaptive vector channel allocation for scalable video transmission over frequency-selective fading channels,” IEEE Trans. Mobile Comput., vol. 1, no. 2, pp. 132–141, Apr.-Jun. 2002.

[21] C. Han, T. Hashimoto, and N. Suehiro, “Constellation-rotated vector OFDM and its performance analysis over Rayleigh fading channels,” IEEE Trans. Commun., vol. 58, no. 3, pp. 828–837, Mar. 2010.

[22] P. Cheng, M. Tao, Y. Xiao, and W. Zhang, “V-OFDM: On performance limits over multi-path Rayleigh fading channels,” IEEE Trans. Commun., vol. 59, no. 7, pp. 1878–1892, Jul. 2011.

[23] Y. Li, I. Ngebani, X.-G. Xia and A. Host-Madsen, “On performance of Vector OFDM with linear receivers,” IEEE Trans. Signal Process., vol. 60, no. 10, pp. 5268–5280, Jun. 2012

[24] S. Benedetto, D. Divsalar, G. Montorsi and F. Pollara, “Soft-Output Decoding Algorithms in Iterative Decoding of Turbo Codes”, The Telecommunications and Data Acquisition Progress Report 42-124, Jet Propulsion Laboratory, Pasadena, California, pp. 63-87, February 15, 1996.

[25] A. J. Viterbi, “An Intuitive Justification and a Simplified Implementation of the MAP Decoder for Convolutional Codes”, IEEE Journal on Selected Areas in Communications, Vol. 16, no. 2, pp. 260-264, February 1998.

[26] Y. Li, B. Vucetic, and Y. Sato, “Optimum soft-output detection for channels with intersymbol interference,” IEEE Trans. Inform. Theory, vol. 41, pp. 704–713, May 1995.

[27] W. Koch and A. Baier, “Optimum and sub-optimum detection of coded data disturbed by time varying intersymbol interference,” in Proc. of the IEEE Global Telecomm. Conf., Dec. 1990, pp. 1679–1684.

[28] D. Yellin, A. Vardy, and O. Amrani, “Joint equalization and coding for intersymbol interference channels,” IEEE Trans. Inform. Theory, vol. 43, pp. 409–425, Mar. 1997.

[29] S. Benedetto et al., “Serial concatenated trellis coded modulation with iterative decoding: Design and performance,” in Proc. IEEE Global Telecomm. Conf., Nov. 1997.

[30] S. Benedetto et al., “Parallel concatenated trellis coded modulation,” in Proc. IEEE Int. Conf. Communications, June 1996.

[31] X. Wang and H. Poor, “Iterative (turbo) soft interference cancellation and decoding for coded CDMA,” IEEE Trans. Commun., vol. 47, pp. 1046–1061, July 1999.

[32] J. Hagenauer, “The turbo principle: Tutorial introduction and state of the art,” in Proc. Int. Symp. on Turbo Codes, Brest, France, Sept. 1997, pp. 1–11.

[33] C. Berrou and A. Glavieux, “Near optimum error correcting coding and decoding: Turbo codes,” IEEE Trans. Commun., vol. 44, pp. 1261–1271, Oct. 1996.

[34] C. Heegard and S. Wicker, Turbo Coding. Boston, MA: Kluwer, 1999.

[35] C. Douillard et al., “Iterative correction of intersymbol interference: Turbo equalization,” Eur. Trans. Telecommun., vol. 6, pp. 507–511, Sept.–Oct. 1995.

[36] G. Bauch and V. Franz, “A comparison of soft-in/soft-out algorithms for ‘turbo detection’,” in Proc. Int. Conf. Telecomm., June 1998, pp. 259–263.

[37] A. Anastasopoulos and K. Chugg, “Iterative equalization/decoding for TCM for frequency-selective fading channels,” in Conf. Rec. 31th Asilomar Conf. Signals, Systems & Computers, vol. 1, Nov. 1997, pp. 177–181.

[38] D. Raphaeli and Y. Zarai, “Combined turbo equalization and turbo decoding,” in Proc. IEEE Global Telecomm. Conf., vol. 2, Nov. 1997, pp. 639–643.

[39] M. Toegel, W. Pusch, and H. Weinrichter, “Combined serially concatenated codes and turbo-equalization,” in Proc. 2nd Int. Symp. on Turbo Codes, Brest, France, Sept. 2000, pp. 375–378.

[40] M. Tüchler, R. Koetter, and A. Singer, “Turbo equalization: Principles and new results,” IEEE Trans. Commun., vol. 50, pp. 754–767, May 2002.

[41] M. Tüchler, A. Singer, and R. Koetter, “Minimum mean squared error equalization using a priori information,” IEEE Trans. Signal Processing, vol. 50, pp. 673–683, Mar. 2002.

[42] S. A. Abrantes, “From BCJR to turbo decoding: Map algorithm made easier,” 2004.