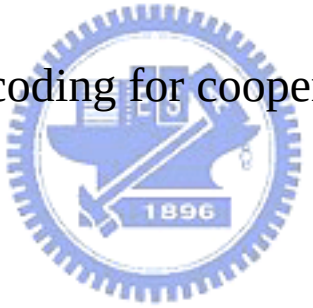
國　立　交　通　大　學

電信工程學系碩士班

碩士論文

合作式通訊下之分散式渦輪碼

Distributed turbo coding for cooperative communications



研 究 生：何 軒 廷

指導教授：吳 文 榕　博士

中華民國九十七年八月

合作式通訊下之分散式渦輪碼

Distributed turbo coding for cooperative communications

研 究 生：何軒廷　　Student：Hsuan-Ting Ho

指導教授：吳文榕 博士　　Advisor：Dr. Wen-Rong Wu

國 立 交 通 大 學

電信工程學系碩士班

碩 士 論 文

A Thesis

Submitted to Department of Communication Engineering
College of Electrical and Computer Engineering
National Chiao-Tung University
in Partial Fulfillment of the Requirements
for the Degree of
Master of Science
In
Communication Engineering
August 2008
Hsinchu, Taiwan, Republic of China

中華民國九十七年八月

合作式通訊下之分散式渦輪碼

Distributed turbo coding for cooperative communications

研 究 生：何 軒 廷　指導教授：吳 文 榕 教授

國立交通大學電信工程學系碩士班

摘要

文獻指出分散式渦輪碼(Distributed Turbo coding)可以接近無線中繼網路(Wireless Relay Network)的通道容量上限(Capacity)。實際的分散式渦輪碼，如果在中繼端發生故障(Outage)將轉換至非合作的模式，因為如果繼續中繼錯誤的資訊將會誤導目的地終端，並造成錯誤的傳遞繁殖；而此非合作模式將使系統效能下降。最近，一種轉送模式稱之為解碼放大轉送模式(Decode-Amplify-Forward)被提出來解決這個問題。不像分散式渦輪碼，解碼放大轉送模式既不在中繼端重新編碼，也不在目的地終端做遞迴解碼，因此擁有較有效率之運算複雜度。然而，解碼放大轉送模式的中繼端需要知道 Log-Likelihood Ratio 的分佈，一般被模擬成高斯混合(Gaussian Mixture)。在本論文裡，我們提出使用 EM algorithm 做為識別高斯混合的方法，模擬結果顯示 EM algorithm 可以更準確地識別此分佈，因而解碼放大轉送模式的效能也得到提升。而分散式渦輪碼可與解碼放大轉送模式合併為一種混合轉送模式，此一模式比其他我們所比較的分散式渦輪碼的變形都擁有較好的效能。

Distributed turbo coding for cooperative communications

Student：Hsuan-Ting Ho　Advisor：Dr. Wen-Rong Wu

Department of Communication Engineering

National Chiao-Tung University

Abstract

It has been shown that distributed turbo coding (DTC) can approach the capacity of a wireless relay network. Practical DTC has to switch to non-cooperative transit mode should inter-user outage occur, since erroneous bits, if (re-encode and) forwarded to the destination, will mislead the destination and cause severe error propagation. This non-cooperative transit mode degrades the performance. Recently, a forwarding strategy termed decode-amplify-forward (DAF) is developed to solve the problem. Unlike DTC, DAF neither conducts encoding at the relay nor iterative decoding at the destination. It is superior and computationally more efficient. However, DAF requires the relay to have the likelihood-ratio (LLR) distribution, typically modeled as a Gaussian mixture. In this thesis, we propose to use the EM algorithm for the identification of the Gaussian mixture. Simulations show that the EM algorithm can identify the LLR distribution accurately and the performance of DAF can be enhanced. Further, DAF can be combined with DTC resulting in a DTC-DAF scheme. We have also studies various DTC related schemes and found that DTC-DAF can have the best performance.

誌謝

首先，要特別感謝我的指導教授 吳文榕博士，兩年來在課業及研究上的仔細教導與指引，總是花費許多時間，與我討論研究上的問題，為我解開學習的困惑，使得此篇論文得以順利完成，讓我在通訊領域上有更深入的了解。

感謝口試委員 蘇育德教授、祁忠勇教授，以及伍紹勳教授給予的寶貴意見與建議，激發我思考研究更多的可能性，補足論文的不足之處，使整篇論文更加嚴謹。您們對於通訊領域研究的堅持與敏感度，亦是我學習的榜樣。

此外，感謝所有的博士班學長適時地提供意見，給予我在研究上的指導及協助；感謝寬頻傳輸與訊號處理實驗室所有同學與學弟妹的支持與協助，讓我的研究所生活更值得回憶。

最後，感謝我最愛的 爺爺、家人，還有盈涵，給予我在生活與精神上最大的鼓勵與支持，使我可以無後顧之憂，順利完成碩士學位。

Contents

List of figures

Chapter 1 Introduction

Next generation wireless communications (third generation and beyond) will bear little resemblance to first- and second-generation, mostly voice cellular systems. In order to meet the demands of multi-rate multimedia communications, next-generation cellular systems must employ advanced algorithms and techniques that not only increase the data rate, but also enable the system to guarantee the quality of service (QoS) for various applications. Techniques currently being investigated for meeting the requirements include advanced signal processing, tailoring system components (such as coding, modulation, and detection) specifically for wireless environments, and various forms of diversity. Among these techniques, diversity is of primary importance due to the nature of wireless environments.

The mobile radio channel suffers from fading, implying that, within the range of any given cell, mobile users may experience severe variations in signal attenuation. By transmitting or processing independently fading copies of the signal, diversity is an effective method combating the fading effect. Some well-known forms of diversity include spatial diversity, temporal diversity, and frequency diversity. Spatial diversity relies on the principle that signals transmitted from geographically separated transmitters, and/or to geographically separated receivers, experience independent fading. To have spatial diversity, multiple transmit antennas are required. Unfortunately, this may not be always feasible in the uplink of a cellular system, due to the size limitation in the mobile unit. In order to overcome this limitation, yet still emulate transmit antenna diversity, a new form of spatial diversity was developed via the cooperation of in-cell users. In each cell, each user may find a “partner.” Each of the two partners is responsible for transmitting not only their own signal, but also the signal of their partner, which they receive and detect. This is, in effect, attempting to achieve

spatial diversity through the use of the partner's antenna; however, this is complicated by the fact that the inter-user channel is noisy.

A particularly powerful variation of user cooperation is coded cooperation. Instead of a simple repetition relay, coded cooperation integrates cooperation into channel coding. Under the scope of coded cooperation, a novel coding technique, termed distributed turbo coding (DTC), was proposed for quasi-static relay channel [1]. It has been shown that DTC can approach the capacity of a wireless relay network [2]. The operation of the DTC can be explained as follows. The source broadcasts a recursive systematic convolutional code (RSC) code to both the relay and the destination. After detecting the data broadcasted by the source, the relay interleaves and re-encodes the message prior to forwarding it to the destination. Since the destination receives two codes, turbo decoding can be conducted. While this construction maintains the diversity benefit of relaying, the coding gain can also be improved. This extra coding gain is due to the interleaving gain of the turbo code construction and the turbo processing gain of the iterative decoder.

In the original DTC schemes, it is usually assumed that the relay can perform error-free decoding. We refer to such DTC schemes as perfect DTC. This scenario can be realized by automatic repeat request (ARQ) in the link between the source and the relay. However, the use of ARQ will reduce the system transmission throughput. Practical DTC has to switch to non-cooperative transit mode should inter-user outage occurs, since erroneous bits, if (re-encode and) forwarded to the destination, will mislead the destination and cause severe error propagation. Here by outage, we mean that transmit signals are corrupted by the channel fading and other impairments to a level that prohibits the relay from correctly deducing all the data. What can we possibly do to enhance the performance of DTC in inter-user outage? Several recent works [4]-[7] have reported a new forwarding strategy, called soft (information)

relaying, to deal with the above outage problem. With this strategy, the delay does not have to make decisions; it may calculate and forward the soft information of the transmit data.

In this thesis, we propose to use the EM algorithm for the identification of the Gaussian mixture at a DAF relay. Simulations show that the EM algorithm can help DAF to retrieve the information more correctly, without adding too much processing complexity at the relay. Further, DAF can be combined with DTC resulting in a DTC-DAF scheme. It is shown that DTC-DAF can have the best performance among DTC related schemes we studied.

The rest of this thesis is organized as follows: In Chapter 2, we review the cooperative communication systems and its two basic forwarding strategies, amplify-and-forward (AF) and decode-and-forward (DF). Also, we review the distributed turbo code. In Chapter 3, we describe the soft relaying methods to combat inter-user outage, including Soft-DF (a DTC variant), decode-amplify-forward (DAF) in detail. Also included is a DTC error-resilient scheme (a DTC variant). In Chapter 4, we describe the proposed scheme. We use the expectation-maximization (EM) algorithm to identify the distribution, modeled as a Gaussian mixture, of log-likelihodd ratios (LLRs) at the relay. This distribution is required for the soft relaying method like DAF and Soft-DF. Chapter 5 reports some simulation results of the mentioned schemes. Finally, we draw some conclusions and outline some possible future works in Chapter 6.

Chapter 2 Cooperative communication systems

2.1 Cooperative communication

Diversity is an effective technique to combat fading. No matter it is operated in the time, frequency or spatial domains. However, spatial diversity may be desirable in practice since it can effectively enhance the spectrum efficiency. To have spatial diversity, multiple antennas are required. While this is feasible in base stations, it may be difficult in mobile stations due to size, costs, hardware complexity, or other constraints. To address this limitation, the concept of cooperation diversity was introduced, where mobiles can achieve uplink transmit diversity by relaying. Figure 2.1 shows a simplest three-terminal network consisting of a source, a relay and a destination, showing the basic idea behind this concept. Since each of the users sees an independent fading path to the destination, diversity is obtained by transmitting each user's data through the relay. By using this approach, multiple virtual-antennas can be constructed in the transmitter. Many research works also show that considerable benefits result from signal relaying in fading environments especially over slow fading channels, including the reduction in outage probability, higher capacity, less power consumption and wider dynamic range.

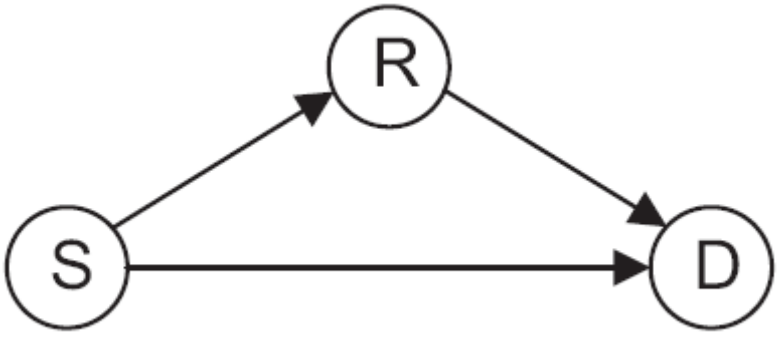


Fig. 2.1 The scenario of relay channel

Despite the theoretic advances in wireless user cooperation, practical signal relaying strategies have not evolved much out of the three basic forms proposed by Cover and El Gamal in 1979 [8], namely, amplify-and-forward (AF), decode-and-forward (DF) and compress-and-forward (CF). In Section 2.3 and 2.4, we will review two basic strategies AF and DF separately, after the system model is first given in Section 2.2.

A particularly powerful variation of user cooperation is coded cooperation. Coded cooperation integrates cooperation into channel coding. Instead of a simple repetition relay, coded cooperation partitions a codeword into two parts, and transmits one part through the direct link, and the other through relay link. That is to say, the codeword will experiences two independent channels before it is received by the destination. In section 2.5, one extension of coded cooperation, termed distributed turbo coding (DTC), is also presented.

2.2 System model

For simplicity, consider the basic relay system in Figure 2.1 that comprises a source, a relay and a destination. We consider half-duplex transit modes, where user cooperation is operated in two stages: the broadcasting stage, where the source broadcast a package of data to both the destination and the relay, and the relaying stage, where the relay processes and forwards part or all of the observations to the destination. The destination then combines the signals received from both stages to make a best estimation of the original data. Throughout this thesis, we will use subscripts S, R, D and SR, SD, RD to denote the quantities pertaining to the source, relay, and destination nodes, and those pertaining to the source-relay, source-destination and relay-destination channels, respectively.

We take block Rayleigh fading as the channel model, which is described as

$$y_{SR}(i) = \sqrt{P_{SR}} h_{SR} x_S(i) + n_{SR}(i) \tag{2.1}$$

$$y_{SD}(i) = \sqrt{P_{SD}} h_{SD} x_S(i) + n_{SD}(i) \tag{2.2}$$

$$y_{RD}(i) = \sqrt{P_{RD}} h_{RD} x_R(i) + n_{RD}(i) \tag{2.3}$$

where x is the transmitted signal, y is the received signal and h is the fading channel coefficients. For simplicity, we consider a binary phase-shift keying (BPSK) modulation, $x_R \in \{1, -1\}$ ($0 \rightarrow 1 \& 1 \rightarrow$ -1), whereas x_R may depend on the specific forwarding strategy. In the case of AWGN, h is a constant of 1. In the case of block fading, h is modeled as zero-mean, circular symmetric complex Gaussian random variable with a variance of 1, which remains fixed over a block of data, and changes independently from one block to another. The additive white Gaussian noises, n_{SR}, n_{SD} and n_{RD}, are zero-mean complex Gaussian random variables with a variance of N_0. Here, we assume that all noise processes have the same variances, without loss of generality. We can be taken into account the cases of different variances by appropriately adjusting the power term, P_{SR}, P_{SD} and P_{RD} of each link. The SNR of a channel, γ, is defined as

$$\gamma = \frac{P}{N_0} \tag{2.4}$$

We consider spatially independent channels among the source, the relay, and the destination. We further assume that the instantaneous channel state information is known to the receivers (but not to the transmitter), so that the decoder can exploit efficient soft decoding algorithm. And the cooperative ratio κ = 1/3, defined as the percentage of time allocated to the relaying stage in a round of user cooperation, is fixed in transmissions.

2.3 Amplify-and-forward

In this scenario, the relay scale and retransmit the analog signal waveform from the source. This is amplify-and-forward, as implied by the name. The operation of AF is straightforward, requiring a lower implementation complexity in digital signal processing. More importantly, one attractive advantage of AF is its ability to operate at all times, even when the source-relay channel experiences outage.

Mathematically, the transmit signal at the relay is formulated in AF as

$$x_R(i) = \frac{y_{SR}(i)}{\sqrt{\bar{P}_y}} \quad \text{i = 1,2,3,…,N,} \tag{2.5}$$

where N is the length of the codeword (block), $x_R(i)$ is the retransmitted signal at the relay, and $\bar{P}_y$ is the average power of the received signals:

$$\bar{P}_y = \frac{\sum_{j=1}^{N} \left| y_{SR(j)} \right|^2}{N} \tag{2.6}$$

The destination observes from source-relay-destination (S-R-D) channel a noisy signal of the form:

$$y_{RD}(i) = \sqrt{P_{RD}} h_{RD} \left(\frac{\sqrt{P_{SR}} h_{SR} x_S(i) + n_{SR}(i)}{\sqrt{\bar{P}_y}} \right) + n_{RD}(i)$$

$$= y_{RD}(i) = \left(\frac{\sqrt{P_{RD}} \sqrt{P_{SR}} h_{RD} h_{SR}}{\sqrt{\bar{P}_y}} \right) x_S(i) + \frac{\sqrt{P_{RD}} h_{RD} n_{SR}(i)}{\sqrt{\bar{P}_y}} + n_{RD}(i) \tag{2.7}$$

which makes the cascade channel behave like a single (block) fading channel with power-fading coefficient $\left(\frac{\sqrt{P_{RD}} \sqrt{P_{SR}} h_{RD} h_{SR}}{\sqrt{\bar{P}_y}} \right)$, and a complex Gaussian noise of variance $N_0 \left(1 + \frac{P_{RD} \left| h_{RD} \right|^2}{\bar{P}_y} \right)$.

The destination then gathers the signals received from both the cascade channel and the

direct source-destination channel using the maximum ratio combining (MRC) principle, which in effect is to extract and combine the log-likelihood ratios (LLRs) from the channels as:

$$l_{AF}(i) = l_{SD}^{CH}(i) + l_{RD}^{CH}(i) = \frac{-4\sqrt{P_{SD}}\,Re\{h_{SD}^{*} y_{SD}\}}{N_0} + \frac{-4\dfrac{\sqrt{P_{RD}}\sqrt{P_{SR}}}{\sqrt{\bar{P}_y}}\,Re\{h_{RD}^{*} h_{SR}^{*} y_{RD}\}}{N_0\left(1+\dfrac{P_{RD}\left|h_{RD}\right|^2}{\bar{P}_y}\right)}, \qquad (2.8)$$

where the minus sign is due to BPSK mapping ($0 \to 1 \& 1 \to -1$). Here, we give a brief proof of (2.8). For ease of description, here we consider the simplest case. Let

$$y_1 = h_1 x_s + n_1 \Rightarrow r_1 = \mathrm{Re}\{\frac{h_1^{*}}{|h_1|} y_1\} = |h_1| x_s + z_1$$

$$y_2 = h_2 x_s + n_2 \Rightarrow r_2 = \mathrm{Re}\{\frac{h_2^{*}}{|h_2|} y_2\} = |h_2| x_s + z_2$$

$$where\ h_1, h_2 \in CN(0,1); n_1, n_2 \in CN(0, N_0); z_1, z_2 \in N(0, \frac{N_0}{2})$$

$$llr_{r_1} + llr_{r_2} = \frac{-4|h_1| r_1}{N_0} + \frac{-4|h_2| r_2}{N_0} = \frac{-4(|h_1|^2 x_s + |h_1| z_1 + |h_2|^2 x_s + |h_2| z_2)}{N_0}$$

$$= \frac{-4(x_s + \dfrac{|h_1| z_1 + |h_2| z_2}{|h_1|^2 + |h_2|^2})(|h_1|^2 + |h_2|^2)}{N_0} = \frac{-4(x_{MRC})(|h_1|^2 + |h_2|^2)}{N_0} = llr_{x_{MRC}}$$

$$where\ x_{MRC} = \frac{|h_1| r_1 + |h_2| r_2}{|h_1|^2 + |h_2|^2} = x_s + \frac{|h_1| z_1 + |h_2| z_2}{|h_1|^2 + |h_2|^2}, \quad Var[\frac{|h_1| z_1 + |h_2| z_2}{|h_1|^2 + |h_2|^2}] = \frac{N_0}{2(|h_1|^2 + |h_2|^2)}$$

Thus, the LLR obtained by the summation of LLRs calculated from the direct and the relay link is equivalent to that calculated from the received signal after MRC.

2.4 Decode-and-forward

In DF, when the relay has successfully decoded all the bits in the source-block, it will re-encode them and transmit a (new and) clean set of bits to the destination. However, in order to prevent error propagation and improve performance, DF typically includes an option to switch to the non-cooperation mode when the relay fails to decode the data correctly. In repetition-DF, the destination will perform similarly to the case of AF by combining the LLRs from the SD and RD channels:

$$l_{AF}(i) = l_{SD}^{CH}(i) + l_{RD}^{CH}(i) = \frac{-4\sqrt{P_{SD}}\,Re\{h_{SD}^{*} y_{SD}\}}{N_0} + \frac{-4\sqrt{P_{RD}}\,Re\{h_{RD}^{*} y_{RD}\}}{N_0}. \qquad (2.9)$$

In a more sophisticated scheme such as distributed turbo code (DTC) of coded cooperation, the destination will treat the two packages it received as a single codeword and perform decoding jointly.

2.5 Distributed turbo coding

To improve the performance in a relay network, some cooperative and distributed coding schemes have been developed recently. A distributed turbo code (DTC) system is developed for a two-hop relay network in [2]. The source broadcasts the coded signals to both the destination and relay. The relay then decodes the received signals, and interleaves them prior to the encoding. The signals received at the destination contain coded information symbols transmitted from the source and coded interleaved information symbols transmitted from the relay. These two symbol sequences form a distributed turbo code. It has been shown that such a coding strategy can achieve the theoretic outage probability bound of a relay channel. Figure 2.2 is the block diagram of a distributed turbo code.

In the following, we will briefly describe how to decode the distributed turbo code at destination. First, we will review the BCJR algorithm for maximum-a-posteriori (MAP)

detection problem. Then, we give the iterative decoding scheme for the distributed turbo code using the BCJR algorithm.

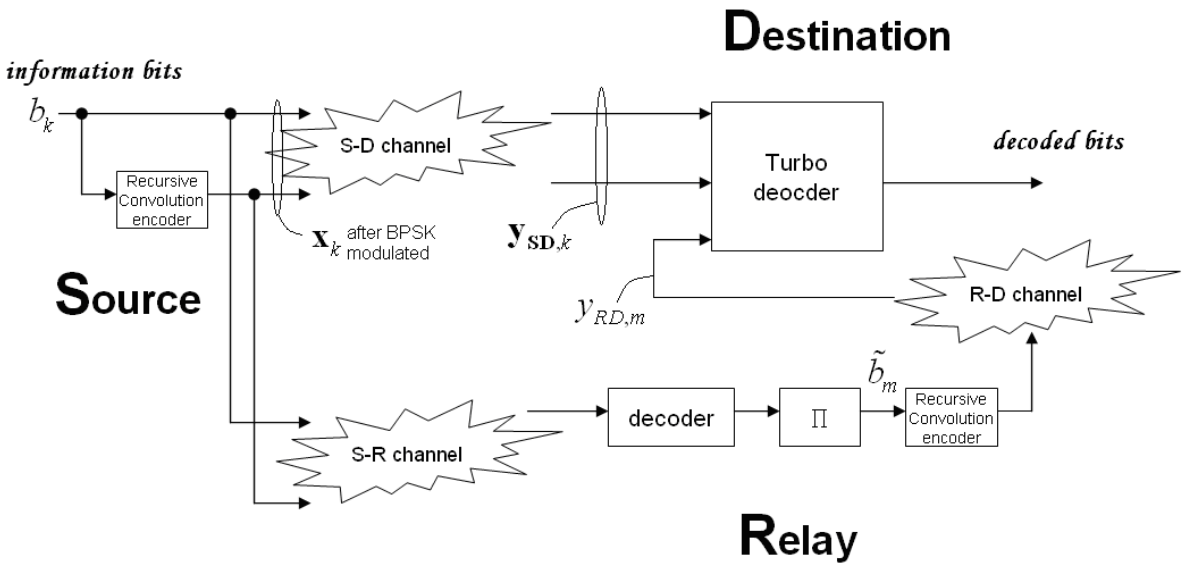


Fig. 2.2 Block diagram of a distributed turbo code

2.5.1 BCJR algorithm for convolutional decoder

In order to minimize the probability of error for each symbol (bit) decision and calculate the soft bits information, we use the MAP-based decoder instead of the Viterbi decoder for the decoding of a convolutional code. One efficient algorithm for the MAP decoder is termed the BCJR algorithm in the literature [12]. In the MAP decoder, to make a decision about the information bit of the k_{th} stage b_k, the decoder calculates the a posteriori probability $\Pr[b_k \mid \mathbf{r}]$ for each possible $b_k \in \{0,1\}$, and choose b_k that maximizes $\Pr[b_k \mid \mathbf{r}]$, where $\mathbf{r}$ is the received packet. In the convolutional code trellis diagram, these probabilities are easily computed once the a posteriori state transition probabilities $\Pr[S_{k-1} = l'; S_k = l \mid \mathbf{r}]$ are known for each state transition in the trellis. The BCJR (Bahl, Cocke, Jelinek and Raviv) algorithm provides a computationally efficient method for finding these state transition probabilities.

The key to the BCJR algorithm is to decompose the a posteriori state transition probability for a transition at stage k into three separate portions: the first only depends on the "pre-" observations $\mathbf{r}_{j<k} = \{r_j : j < k\}$, the second only depends on the "k_{th} (present)" observations $\mathbf{r}_k$, and the third only depends on the "post-" observations $\mathbf{r}_{j>k} = \{r_j : j > k\}$. We can obtain this decomposition through the following derivations:

$$\begin{aligned}\Pr[S_{k-1} = l'; S_k = l \mid \mathbf{r}] &= p(S_{k-1} = l'; S_k = l; \mathbf{r}) / p(\mathbf{r}) \\ &= p(S_{k-1} = l'; S_k = l; \mathbf{r}_{j<k}; \mathbf{r}_k; \mathbf{r}_{j>k}) / p(\mathbf{r}) \\ &= p(\mathbf{r}_{j>k} \mid S_{k-1} = l'; S_k = l; \mathbf{r}_{j<k}; \mathbf{r}_k)\, p(S_{k-1} = l'; S_k = l; \mathbf{r}_{j<k}; \mathbf{r}_k) / p(\mathbf{r}).\end{aligned}$$

(2.10)

Because of the Markov property of the finite-state machine model for the trellis, knowledge of the state at stage k supersedes knowledge of the state at stage $k-1$, and it also supersedes knowledge of $\mathbf{r}_k$ and $\mathbf{r}_{j<k}$, so that (2.10) reduce to:

$$\begin{aligned}\Pr[S_{k-1} = l'; S_k = l \mid \mathbf{r}] &= p(\mathbf{r}_{j>k} \mid S_k = l)\, p(S_{k-1} = l'; S_k = l; \mathbf{r}_{j<k}; \mathbf{r}_k) / p(\mathbf{r}) \\ &= p(\mathbf{r}_{j>k} \mid S_k = l)\, p(S_k = l; \mathbf{r}_k \mid S_{k-1} = l'; \mathbf{r}_{j<k})\, p(S_{k-1} = l'; \mathbf{r}_{j<k}) / p(\mathbf{r}).\end{aligned}$$

(2.11)

Again, exploiting the Markov property, we can simplify (2.11) to:

$$\begin{aligned}\Pr[S_{k-1} = l'; S_k = l \mid \mathbf{r}] &= p(S_{k-1} = l'; \mathbf{r}_{j<k})\, p(S_k = l; \mathbf{r}_k \mid S_{k-1} = l')\, p(\mathbf{r}_{j>k} \mid S_k = l) / p(\mathbf{r}) \\ &= \alpha_{k-1}(l') \times \sum_{i \in \{0,1\}} \gamma_k^i(l', l) \times \beta_k(l) / p(\mathbf{r})\end{aligned}$$

(2.12)

$$\textit{where} \quad \gamma_k^i(l', l) = p(b_k = i; S_k = l; \mathbf{r}_k \mid S_{k-1} = l'), i \in \{0,1\}. \tag{2.13}$$

Observe that $\alpha_{k-1}(l')$ is a probability measure for state l' at stage $k-1$ that depends only on the pre-observations $\mathbf{r}_{j<k}$. On the other hand, $\beta_k(l)$ is a probability measure for state l

at stage k that depends only on the post-observations $\mathbf{r}_{j>k}$. And finally, $\sum_{i\in\{0,1\}} \gamma_k^i(l',l)$ is a probability measure connecting state l' at stage $k-1$ to state l at stage k, and it depends only on the present (k_{th}) observation $\mathbf{r}_k$.

The ultimate goal is to calculate the a posteriori probabilities for the symbols b_k, and not for the transitions. Fortunately, there is a simple easy to do that. Let B_k^i denote the set of transitions $S_{k-1}=l' \to S_k = l$ that are caused by the input bit $b_k = i$. Then the a posteriori probability for the k_{th} symbol is related to the a posteriori transition probabilities of (2.12) as:

$$\begin{aligned}\Pr[b_k = i \,|\, \mathbf{r}] &= \sum_{(l',l)\in B_k^i} \Pr[S_{k-1}=l', S_k = l \,|\, \mathbf{r}] \\ &= \frac{1}{p(\mathbf{r})} \sum_{(l',l)\in B_k^i} \alpha_{k-1}(l')\gamma_k^i(l',l)\beta_k(l).\end{aligned} \tag{2.14}$$

Following (2.12), we first derive the recursive formulae for calculating α_k

$$\begin{aligned}\alpha_k(l) &= p(S_k = l; \mathbf{r}_{j<k+1}) \\ &= p(S_k = l; \mathbf{r}_k; \mathbf{r}_{j<k}) \\ &= \sum_{l'=0}^{M_s-1} p(S_k = l; S_{k-1} = l'; \mathbf{r}_k; \mathbf{r}_{j<k}) \\ &= \sum_{l'=0}^{M_s-1} p(S_k = l; \mathbf{r}_k \,|\, S_{k-1} = l'; \mathbf{r}_{j<k}) P(S_{k-1} = l'; \mathbf{r}_{j<k}) \\ &= \sum_{l'=0}^{M_s-1} p(S_k = l; \mathbf{r}_k \,|\, S_{k-1} = l') P(S_{k-1} = l'; \mathbf{r}_{j<k}) \\ &= \sum_{l'=0}^{M_s-1} \sum_{i\in\{0,1\}} \gamma_k^i(l',l)\alpha_{k-1}(l')\end{aligned} \tag{2.15}$$

We next derive a similar recursion for β_k as

$$\begin{aligned}\beta_{k-1}(l') &= p(\mathbf{r}_{j>k-1} \,|\, S_{k-1} = l') \\ &= p(\mathbf{r}_{j>k}; \mathbf{r}_k \,|\, S_{k-1} = l') \\ &= \sum_{l=0}^{M_s-1} p(\mathbf{r}_{j>k}; \mathbf{r}_k; S_k = l \,|\, S_{k-1} = l') \\ &= \sum_{l=0}^{M_s-1} p(\mathbf{r}_{j>k} \,|\, \mathbf{r}_k; S_k = l; S_{k-1} = l') p(\mathbf{r}_k; S_k = l \,|\, S_{k-1} = l') \\ &= \sum_{l=0}^{M_s-1} p(\mathbf{r}_{j>k} \,|\, S_k = l) p(\mathbf{r}_k; S_k = l \,|\, S_{k-1} = l') \\ &= \sum_{l=0}^{M_s-1} \beta_k(l) \sum_{i\in\{0,1\}} \gamma_k^i(l',l)\end{aligned} \tag{2.16}$$

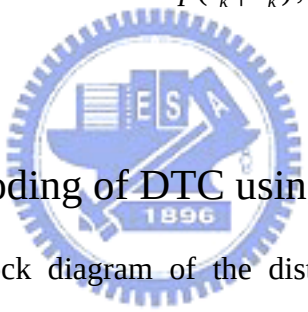
According to (2.13) and Figure 2.2, $\gamma_k^i(l',l)$ can be calculated as

$$\begin{aligned}\gamma_k^i(l',l) &= p(b_k = i; S_k = l; \mathbf{r}_k \mid S_{k-1} = l'), i \in \{0,1\} \\ &= p(\mathbf{r}_k \mid \mathbf{x}_k)\, p(\mathbf{x}_k \mid S_k = l; S_{k-1} = l')\, p(S_k = l \mid S_{k-1} = l').\end{aligned} \tag{2.17}$$

We can further express $\gamma_k^i(l',l)$ as

$$\gamma_k^i(l',l) = \begin{cases} p_k(i) \exp\left(\dfrac{-\left\|\mathbf{r}_k - h\sqrt{P}\mathbf{x}_k(l',l)\right\|^2}{N_0}\right) & \text{for} \quad (l',l) \in B_k^i \\ 0 & \text{otherwise} \end{cases} \tag{2.18}$$

where $p_k(i)$ is a priori probability of $b_k = i$ and $\mathbf{x}_k(l',l)$ is the encoder-BPSK modulated output associated with the transition $S_{k-1} = l' \rightarrow S_k = l$ and input $b_k = i$. Note that (2.18) has been normalized due to the constant term of $p(\mathbf{r}_k \mid \mathbf{x}_k)$, and $\|z\|$ is the Frobenius norm of z.



2.5.2 Iterative decoding of DTC using MAP algorithm

Figure 2.3 depicts the block diagram of the distributed turbo decoder. The iterative decoding of distributed turbo code consists of two component decoders serially concatenated via an interleaver. The first MAP decoder takes as input the received information sequence $\mathbf{y}_{\mathbf{SD}}^{\mathbf{s}}$ and the received parity sequence generated by the first encoder $\mathbf{y}_{\mathbf{SD}}^{\mathbf{p}}$ from SD channel. The decoder then produces a soft output, which is interleaved and used to produce an improved estimate of the a priori probabilities of the information sequence for the second decoder. The other two inputs to the second MAP decoder are the interleaved received information sequence $\tilde{\mathbf{y}}_{\mathbf{SD}}^{\mathbf{s}}$ from SD channel and the received parity sequence produced by the second encoder $\mathbf{y}_{\mathbf{RD}}$ from RD channel. The second MAP decoder also produces a soft output which is deinterleaved and used to improve the estimate of the a priori probabilities of the information sequence for the first decoder.

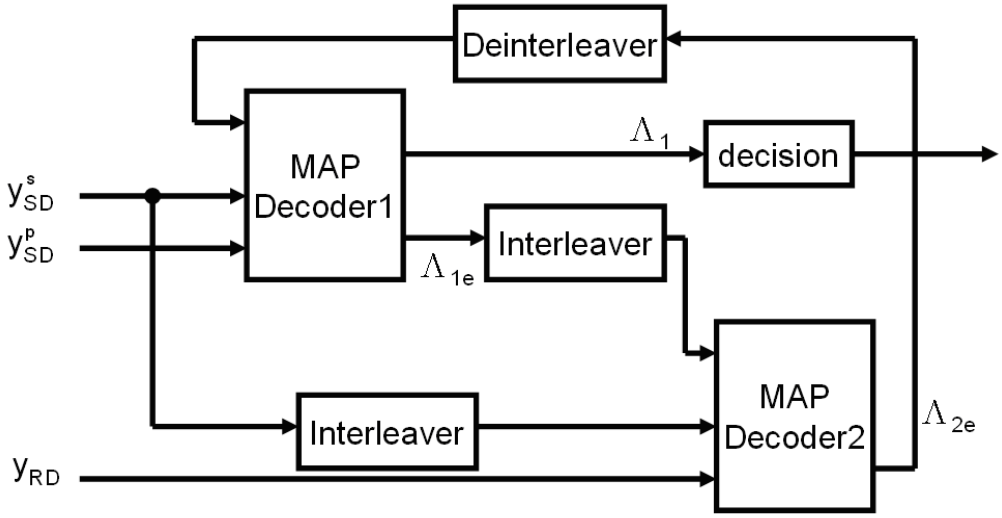


Fig. 2.3 Block diagram of a distributed turbo decoder.

Except for (2.14) and (2.18), below are the extra operations conducted by the iterative decoding of DTC:

$$\Lambda_1(b_k) = \log \frac{\sum_{l',l=0}^{M_s-1} \alpha_{k-1}(l')\, p_k^1(1) \exp\left(-\frac{\left\| \mathbf{y}_{\mathbf{SD},k} - h_{SD}\sqrt{P_{SD}}\,\mathbf{x}_k(l',l) \right\|^2}{N_0}\right)\beta_k(l)}{\sum_{l',l=0}^{M_s-1} \alpha_{k-1}(l')\, p_k^1(0) \exp\left(-\frac{\left\| \mathbf{y}_{\mathbf{SD},k} - h_{SD}\sqrt{P_{SD}}\,\mathbf{x}_k(l',l) \right\|^2}{N_0}\right)\beta_k(l)} \tag{2.19}$$

$$= \underbrace{\log \frac{p_k^1(1)}{p_k^1(0)}}_{priori\ \inf.} \underbrace{- \frac{4\sqrt{P_{SD}}\,\mathrm{Re}\{y^s_{SD,k} h^*_{SD}\}}{N_0}}_{systematic\ part} + \underbrace{\log \frac{\sum_{l',l=0}^{M_s-1} \alpha_{k-1}(l') \exp\left(\frac{-(y^p_{SD,k} - h_{SD}\sqrt{P_{SD}}\,x^p_k(l',l))^2}{N_0}\right)\beta_k(l)}{\sum_{l',l=0}^{M_s-1} \alpha_{k-1}(l') \exp\left(\frac{-(y^p_{SD,k} - h_{SD}\sqrt{P_{SD}}\,x^p_k(l',l))^2}{N_0}\right)\beta_k(l)}}_{parity\ part\quad (extrinsic\ \inf.)}$$

$$\Lambda_2(\tilde{b}_m) = \log \frac{\sum_{l',l=0}^{M_s-1} \alpha_{m-1}(l')\, p_m^2(1) \exp\left(\frac{-(\tilde{y}^s_{SD,k} - h_{SD}\sqrt{P_{SD}}\,\tilde{x}^s_m(l',l))^2 - (y_{RD,m} - h_{RD}\sqrt{P_{RD}}\,\tilde{x}^p_m(l',l))^2}{N_0}\right)\beta_m(l)}{\sum_{l',l=0}^{M_s-1} \alpha_{m-1}(l')\, p_m^2(0) \exp\left(\frac{-(\tilde{y}^s_{SD,k} - h_{SD}\sqrt{P_{SD}}\,\tilde{x}^s_m(l',l))^2 - (y_{RD,m} - h_{RD}\sqrt{P_{RD}}\,\tilde{x}^p_m(l',l))^2}{N_0}\right)\beta_m(l)}$$

$$= \underbrace{\log \frac{p_m^2(1)}{p_m^2(0)}}_{priori\ \inf.} \underbrace{- \frac{4\sqrt{P_{SD}}\,\mathrm{Re}\{\tilde{y}^s_{SD,k} h^*_{SD}\}}{N_0}}_{systematic\ part} + \underbrace{\log \frac{\sum_{l',l=0}^{M_s-1} \alpha_{m-1}(l') \exp\left(\frac{-(y_{RD,m} - h_{RD}\sqrt{P_{RD}}\,\tilde{x}^p_m(l',l))^2}{N_0}\right)\beta_m(l)}{\sum_{l',l=0}^{M_s-1} \alpha_{m-1}(l') \exp\left(\frac{-(y_{RD,m} - h_{RD}\sqrt{P_{RD}}\,\tilde{x}^p_m(l',l))^2}{N_0}\right)\beta_m(l)}}_{parity\ part\quad (extrinsic\ \inf.)}$$

(2.20)

As that in Figure 2.3, the extrinsic LLRs of the first decoder pass through the interleaver, turning into priori LLRs of the second decoder. Similarly, the extrinsic LLRs of the second decoder pass through the de-interleaver, turning into priori LLRs of the first decoder. Iterations are then conducted with (2.19) and (2.20). Finally, DTC will make a hard decision on b_k based on $\Lambda_1(b_k)$.

Chapter 3 Soft information relaying

3.1 Background of soft information relaying

User cooperation allows the nodes in a common network to leverage the resource, such as power, memory, and antennas, of their neighboring nodes to improve communication efficiency. This framework is particularly helpful for slow fading channel where time diversity is hard to achieve. A variety of DF schemes have been proposed, exploiting powerful channel codes, space-time codes and network codes. In demonstrating their substantial cooperative benefits, the prevailing assumption therein is that the source-relay channel is outage-free such that the relay can always retrieve the source packet correctly. However, practical wireless channels often experience fading, and DF schemes are therefore not operational all the time due to outage, causing the drastic performance degradation. Fortunately, AF can resolve this problem to some extent. However, an AF-relayed packet may have been badly corrupted in the source-relay transmission and further distorted in the relay-destination transmission, and therefore becomes too noisy to be of any use.

In order to improve the degrading performance due to outage, several recent works developed a new forwarding strategy, called soft (information) relaying, to deal with the problem. With this strategy, the relay does not have to make decisions, it can calculate and forward the soft estimates of the transmitted symbols. In the following sections, we will specify how these soft relaying methods manage to combat inter-user outage, including Soft-DF (a DTC variant) in Section 3.2, decode-amplify-forward (DAF) in Section 3.3. Also included in Section 3.4 is a DF error-resilient algorithm (a DTC variant also).

3.2 Soft-DF (a DTC variant)

As mentioned in Chapter 2, the main advantage of AF over DF is that no hard decisions are made at the relay, and error propagation can be avoided. However, AF does not regenerate the signal, its performance will be inferior to that of DF if decisions are correct. This new Soft-DF technique keeps the advantages of both previous techniques, i.e. it regenerates soft-information signal. A relay using this technique first decodes its received signal, calculates soft information, interleaves and re-encodes it with a soft-input-soft-output (SISO) encoder. According to [9], it is reasonable to assume that this soft output value can be approximated as a BPSK signal plus AWGN noise. We will characterize this in detail later.

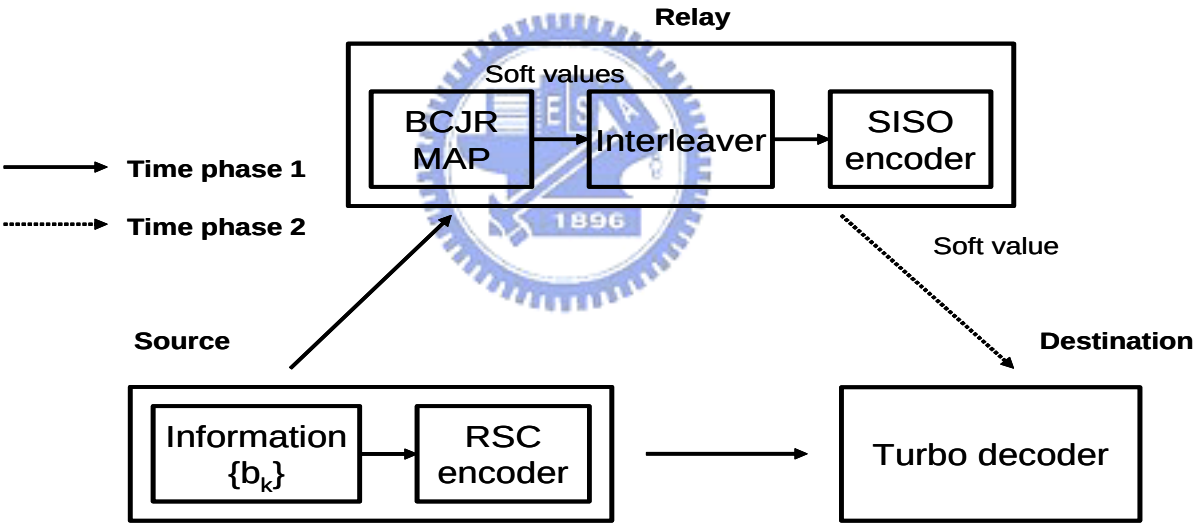


Fig. 3.1 Block diagram of Soft-DF

Figure 3.1 depicts the block diagram of the Soft-DF method. After encoding and BPSK modulation, the source broadcasts the signal to the relay and the destination. The BCJR MAP decoder at relay then calculates the a posteriori log-likelihood ratios $\{\Lambda(b_k \mid \mathbf{y}_{SR})\}$ of the information sequence $\{b_k\}$ using (2.19), and finally conducts soft encoding. Let $\{\tilde{b}_k\}$ be the

interleaved version of $\{b_k\}$ and calculate

$$\Lambda(r_k \mid \mathbf{y}_{SR}) = \log \frac{p(r_k = 1 \mid \mathbf{y}_{SR})}{p(r_k = 0 \mid \mathbf{y}_{SR})} \tag{3.1}$$

where r_k is the parity check bit of $\tilde{b}_k$. The probability $p(r_k = i \mid \mathbf{y}_{SR}), i \in \{0,1\}$ can be obtained by a two-direction recursive algorithm [4]. This gives slightly modified versions of the standard forward recursion $\alpha(\bullet)$ and backward recursion $\beta(\bullet)$ of the MAP decoding algorithm (2.14) ~ (2.17). For an encoder with state $s(k)$ at time k, we can write the following recursions where each term in the summations has to correspond to a combination of $\{s(k-1), \tilde{b}_k, r_k, s(k)\}$ satisfying the code constraints:

$$\alpha(s(k)) = \sum_{\sim s(k)} \alpha(s(k-1)) p(\tilde{b}_k \mid \mathbf{y}_{SR}), \tag{3.2}$$

$$\beta(s(k-1)) = \sum_{\sim s(k-1)} \beta(s(k)) p(\tilde{b}_k \mid \mathbf{y}_{SR}). \tag{3.3}$$

Then we compute

$$P(r_k \mid \mathbf{y}_{SR}) = \sum_{\sim r_k} \alpha(s(k-1)) \beta(s(k)) \, p(\tilde{b}_k \mid \mathbf{y}_{SR}). \tag{3.4}$$

Note $\sum_{\sim z}$ denotes the summation on all variables but z.

According to [9], it is stated that, for an infinite-length or very long code with a soft message-passing decoder, if the LLRs at the input to the decoder are independent and identically Gaussian distributed, then the output LLRs from the decoder will follow an Gaussian distribution, approximately. The Gaussian assumption can be extended to the structure of Soft-DF. It is observed that the input to the soft MAP encoder is i.i.d. Gaussian distributed, and the output also follows a Gaussian distribution, approximately. Therefore, after power normalization, we can assume that this soft output LLRs can be modeled as BPSK signal plus AWGN noise with mean μ_{eq} and variance σ_{eq}^2, as depicted in Figure 3.2.

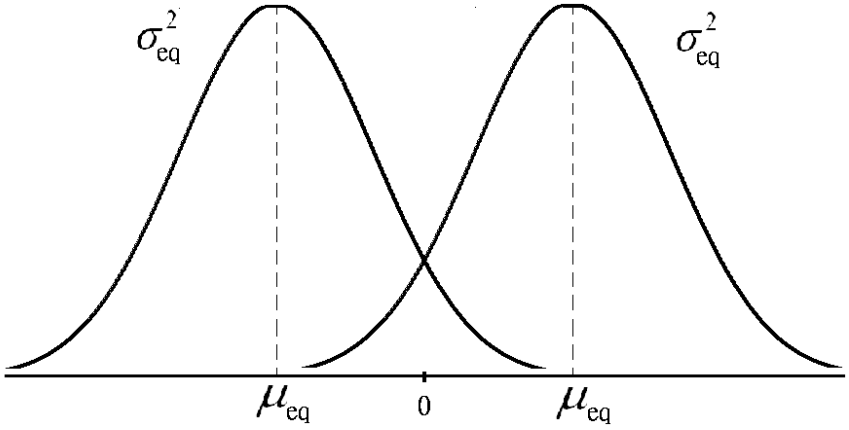


Fig. 3.2 Normalized LLRs' distribution

Now, all operations are the same as distributed turbo code except that original x_R is replaced by $x_R' = \frac{-\Lambda(r_k \mid \mathbf{y}_{SR})}{\sqrt{\sum_k \Lambda(r_k \mid \mathbf{y}_{SR})^2 / N}} = \mu_{eq}(\pm 1) + n_{eq}$. Note the minus is due to BPSK mapping ($0 \to 1 \& 1 \to -1$). At the destination, the second constituent decoder will see an equivalent fading channel with amplitude of $h_{RD}\mu_{eq}$, and an equivalent noise variance of $2\left|h_{RD}\right|^2 P_{RD}\sigma_{eq}^2 + N_0$ for $\mathbf{y_{RD}}$. Therefore, the calculation of LLRs in the second constituent decoder (2.20) will become:

$$\Lambda_2(\tilde{b}_m) =$$

$$\underbrace{\log\frac{p_m^2(1)}{p_m^2(0)}}_{priori\ \inf.} \underbrace{-\frac{4\sqrt{P_{SD}}\,\mathrm{Re}\{\tilde{y}_{SD,k}^s h_{SD}^*\}}{N_0}}_{systematic\ part} + \underbrace{\log\frac{\sum_{l',l=0}^{M_s-1}\alpha_{m-1}(l')\exp\left(\frac{-\left(y_{RD,k}-h_{RD}\mu_{eq}\sqrt{P_{RD}}\tilde{x}_m^p(l',l)\right)^2}{2\left|h_{RD}\right|^2 P_{RD}\sigma_{eq}^2+N_0}\right)\beta_m(l)}{\sum_{l',l=0}^{M_s-1}\alpha_{m-1}(l')\exp\left(\frac{-\left(y_{RD,k}-h_{RD}\mu_{eq}\sqrt{P_{RD}}\tilde{x}_m^p(l',l)\right)^2}{2\left|h_{RD}\right|^2 P_{RD}\sigma_{eq}^2+N_0}\right)\beta_m(l)}}_{parity\ part\quad (extrinsic\ \inf.)}$$

(3.5)

Note that the relay should estimate the mean μ_{eq} and variance σ_{eq}^2 correctly, sending to the destination as parts of the channel state information. This is essential a Gaussian Mixture

Estimation problem. We defer this to Chapter 4, where we propose to apply the maximum-likelihood parameter estimation method.

3.3 Decode-amplify-forward (DAF)

It is well known that even without channel coding, a soft reliability value in the form of the log-likelihood ratio, referred to as *channel-LLR*, can be computed directly from the channel observation, i.e.:

$$l_{SR}^{CH}(i) = \frac{-4\sqrt{P_{SR}}\,Re\{h_{SR}^{*} y_{SR}(i)\}}{N_0}. \tag{3.6}$$

Comparing (2.5) and (3.6), it becomes clear that scaling of channel observations is equivalent to scaling of their channel-LLRs. Hence, in the AF mode, the relay essentially amplifies (scales) and forwarded soft reliability information. Since the reliability information here is extracted directly from the channel and it is coarse in nature. Exploiting the channel code in the source-packet (assume there is one), we can enhance the reliability. This is the rational behind decode-amplify-forward, in which the relay soft decodes the received packet (instead of making binary hard decisions), i.e., calculate the LLRs of information bits referred to as *decoder-LLR*, amplifies (scales) and forwards the LLRs to the destination. This simple extension, i.e., forwarding decoder-LLRs in lieu of hard decisions or channel-LLRs, enables DAF to be operational, and at the same time exploiting the coding gain, in the entire source-relay SNR region.

Mathematically, the signal retransmitted at the relay with DAF can be formulated as

$$x_R(i) = \frac{l_{SR}^{DEC}(i)}{\sqrt{\overline{P}_y}} \quad \text{i = 1,2,3,…,N,} \tag{3.7}$$

where N is the length of the codeword (source-packet), $l_{SR}^{DEC}(i)$ is the decoder-LLR at the

relay, and $\bar{P}_y$ is the average power of $l_{SR}^{DEC}(i)$ given by

$$\bar{P}_y = \frac{\sum_{j=1}^{N} (l_{SR}^{DEC}(i))^2}{N} \tag{3.8}$$

According to the Gaussian assumption in the previous section, this soft output values can be approximated as BPSK signal plus AWGN noise. With power normalization, the signals retransmitted at the relay, $x_R(i)$, may be considered as signals having a mean value $\tilde{\mu}_l x_S$ and being distorted by the Gaussian noise n_l with variance $\tilde{\sigma}_l^2$:

$$x_R(i) = \frac{l_{SR}^{DEC}(i)}{\sqrt{\bar{P}_y}} = \tilde{\mu}_l x_S(i) + n_l(i). \tag{3.9}$$

For simplicity, let's combine the source-relay channel and the soft-decoder at the relay as a virtual block fading channel, whose effective SNR (measured by normalized decoder-LLRs) is

$$\gamma_l = \frac{\tilde{\mu}_l^2}{2\tilde{\sigma}_l^2}. \tag{3.10}$$

Since decoder-LLRs provide more information than channel-LLRs, the effective SNR of the virtual channel is larger than the real SNR of the source-relay channel. Hence, viewed from the destination, DAF is like an AF strategy operating on an enhanced source-relay channel. The destination can then combine the signals originated from the source and forwarded by the relay similar to conventional AF.

The relay signals arriving at the destination are:

$$y_{RD}(i) = \sqrt{P_{RD}} h_{RD} (\tilde{\mu}_l x_S(i) + n_l(i)) + n_{RD}(i), \tag{3.11}$$

whose channel-LLRs, corresponding to the cascade of the virtual source-relay channel and the

relay-destination channel, can be computed from below

$$l_{RD}^{CH}(i) = \frac{-4\sqrt{P_{SD}}\,\tilde{\mu}_l Re\{h_{RD}^{*} y_{RD}(i)\}}{2\left|h_{RD}\right|^2 P_{RD}\tilde{\sigma}_l^2 + N_0}. \tag{3.12}$$

The optimal signal combining at the destination is therefore:

$$l_{DAF}(i) = l_{SD}^{CH}(i) + l_{RD}^{CH}(i) = \frac{-4\sqrt{P_{SD}}\,Re\{h_{SD}^{*} y_{SD}(i)\}}{N_0} + \frac{-4\sqrt{P_{SD}}\,\tilde{\mu}_l Re\{h_{RD}^{*} y_{RD}(i)\}}{2\left|h_{RD}\right|^2 P_{RD}\tilde{\sigma}_l^2 + N_0}. \tag{3.13}$$

Now, we investigate the rationale behind DAF scheme. To see how much gain DAF obtained over AF by processing codes at the relay, we first examine the mutual information between the signal transmitted by the source and the signal forwarded by the relay. According to [5], Figure 3.3 below illustrates the mutual information of the AF and DAF assuming that SR link is protected using recursive systematic convolutional (RSC) code.

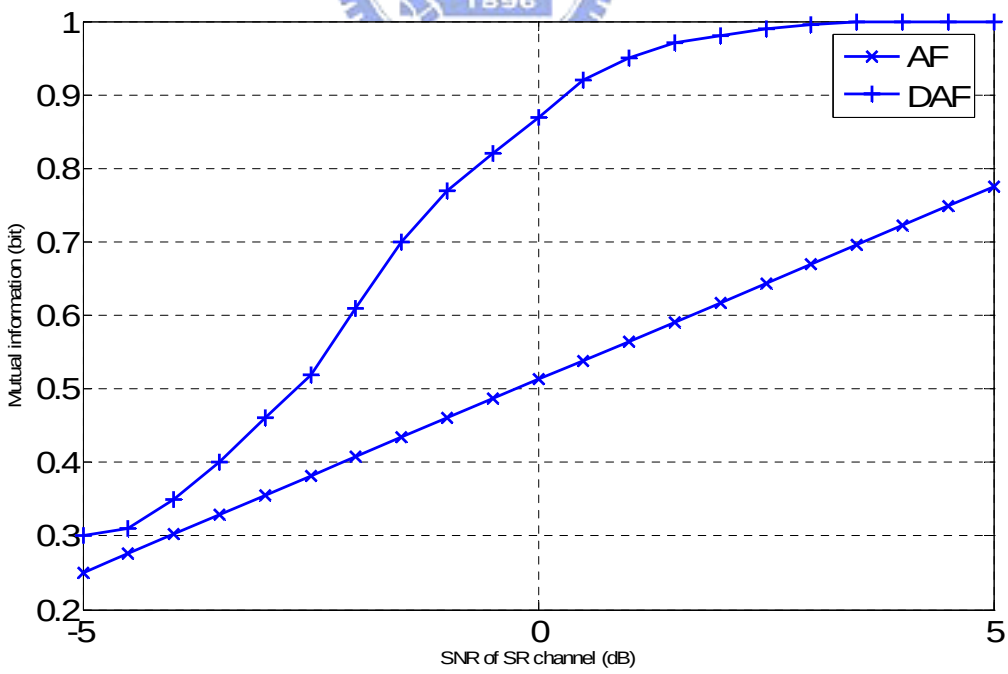


Fig. 3.3 Mutual information between the signal transmitted by the source and the signal forwarded by the relay

Then, according to the analysis of achievable rate in [6] and Figure 3.4 below, several findings can be observed from here. (1) AF outperforms DF (either repetition or DTC) at low source-relay SNRs, but DF quickly catches up and surpasses AF at high source-relay SNRs. (2) Both repetition-DF and DTC start out with a flat capacity curve that is irrelevant to the source-relay SNR. This corresponds to non-cooperation transit mode due to inter-user-outage. The coding gain of DTC becomes obvious at rather high source-relay SNRs. (3) DAF is most advantageous at the low source-relay SNR regime. (4) DTC-DAF extracting and combining the best features, offers a capacity that is the maximum of all others.

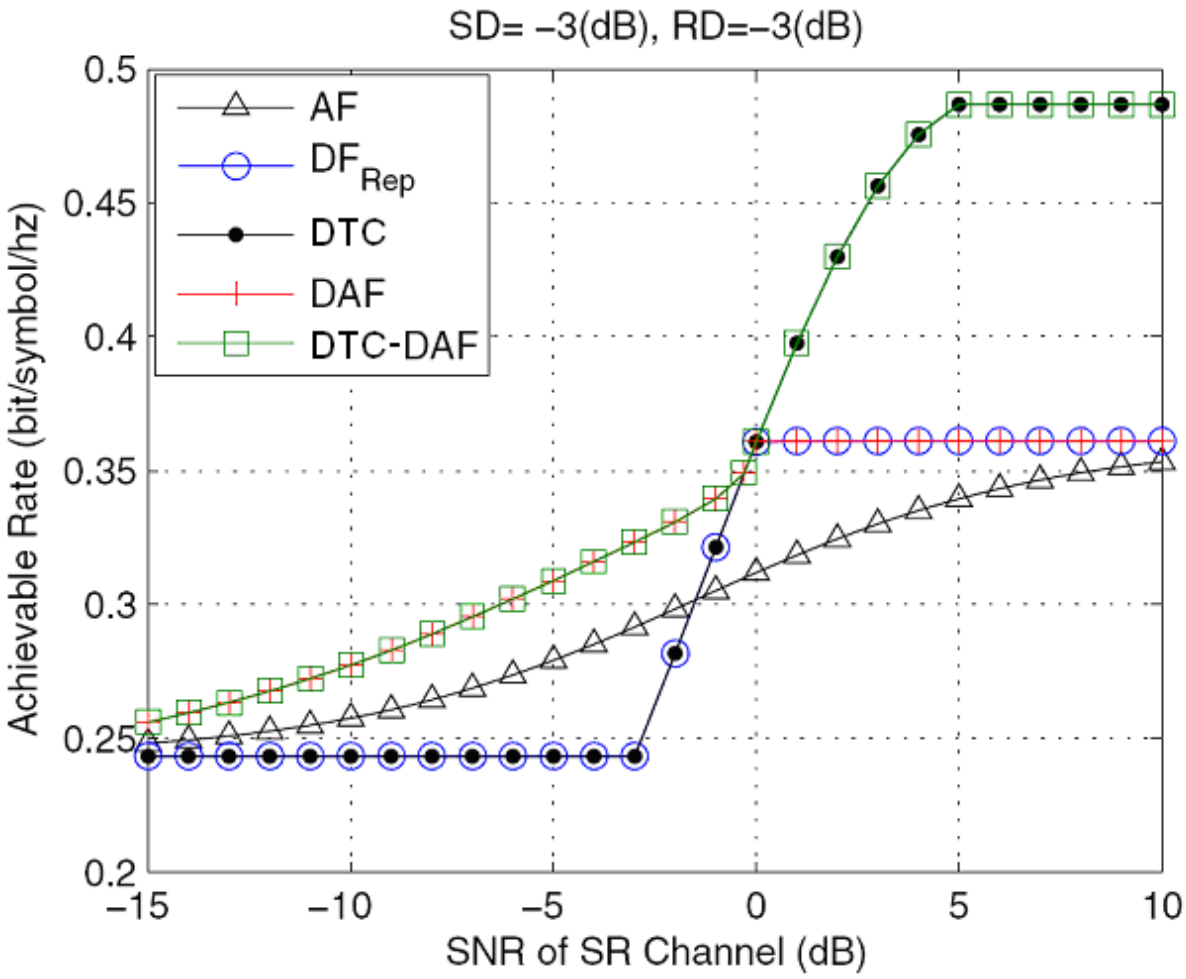


Fig. 3.4 Achievable rate of a symmetric AWGN relay channel

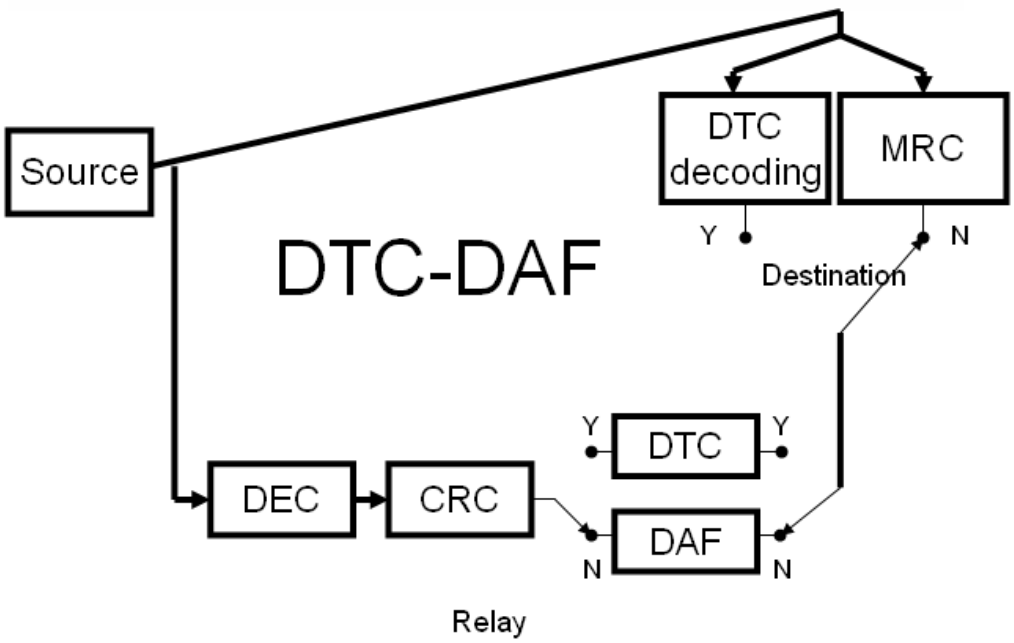


Fig. 3.5 Hybrid DTC-DAF

Figure 3.5 depictes a hybrid DTC-DAF scheme. In this scheme, the relay switches to DAF when the CRC check fails, and to distributed turbo code (DTC) otherwise. It helps to resolve the degrading performance of practical DTC due to non-cooperative transit mode in case of inter-user outage.

3.4 DTC resilient to relay errors (a DTC variant)

In the conventional DF-based strategy, a relay cooperates only when it can decode correctly the signal transmitted from the source. However, this method [10] here presents an alternative DF-based strategy by which the receiver at destination manages to take advantage of forwarded signal even when relay decoding has errors. The method modifies the decoding algorithm at the destination so as to take into account decoding errors at the relay. This prevents the occurrence of non-cooperative transit mode due to inter-user outage and increases the level of cooperation.

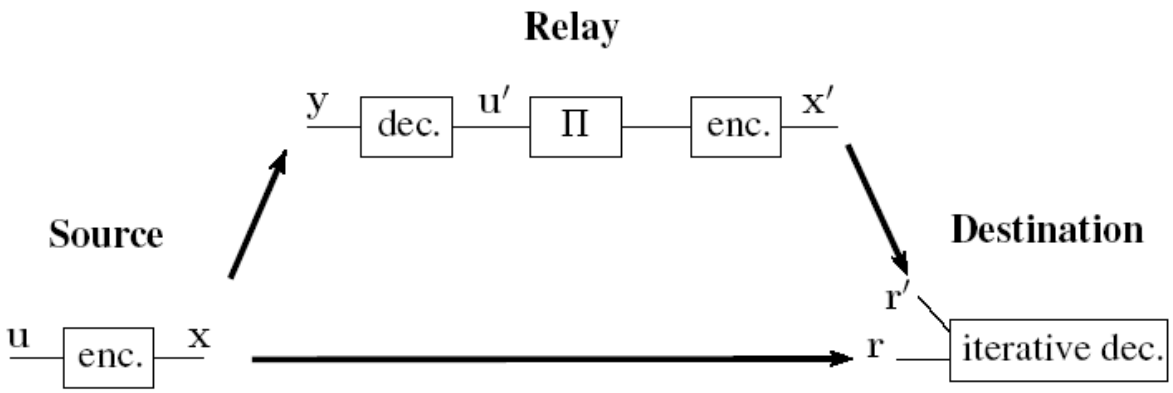


Fig. 3.6 DTC resilient to relay errors

Figure 3.6 shows the block diagram of the DTC resilient to relay errors. Similar to the original DTC, the relay decodes the signal $\mathbf{y}$, obtains hard decisions $\mathbf{u'}$ and checks if errors occur by means of CRC. Unlike in original DTC, the signal is forwarded whether it contains errors or not. But the relay also transmits to the destination some side information related to the source-relay channel: if errors occur, it transmits the instantaneous SNR of the source-relay channel, so that the destination can determine the corresponding average error rate p_e; otherwise, it transmit a signal declaring a error-free decoding at the relay.

At the destination, we will find the optimal estimates of the information bits with the maximum-a-posteriori (MAP) criterion. For each bit u_k, the estimate $\hat{u}_k$ is defined as follows:

$$\hat{u}_k \triangleq \arg \max_{u_k=0,1} p(u_k \mid \mathbf{r}, \mathbf{r'}). \tag{3.14}$$

If no decoding errors occur at the relay, the decoding algorithm is surely the standard distributed turbo decoding (DTC) algorithm as that in Section 2.5.2; otherwise it needs to be modified. The differences between the modified and the standard DTC decoding algorithms can be easily observed with a factor graph framework. We first factorize the a posteriori probability function in (3.14), then draw the corresponding graph, and finally apply the

message-passing algorithm to it.

The a posteriori probability $p(u_k \mid \mathbf{r}, \mathbf{r}')$ is the marginal function of the a posteriori probability of the whole sequence $p(\mathbf{u} \mid \mathbf{r}, \mathbf{r}')$, and the maximum-a-posteriori (MAP) is equivalent to the maximum-likelihood (ML) due to the equal prior of the sequences:

$$\begin{aligned} \hat{u}_k &= \arg \max_{u_k=0,1} \sum_{\sim\{u_k\}} p(\mathbf{u} \mid \mathbf{r}, \mathbf{r}') \\ &= \arg \max_{u_k=0,1} \sum_{\sim\{u_k\}} p(\mathbf{r}, \mathbf{r}' \mid \mathbf{u}), \end{aligned} \tag{3.15}$$

where $\sum_{\sim\{u_k\}}$ here is the sum over all bits of $\mathbf{u}$ except u_k. We can express the dependence between $\mathbf{u}$ and $\mathbf{u}'$ explicitly, and then factorize the likelihood function due to independence of $p(\mathbf{r} \mid \mathbf{u})$ and $p(\mathbf{r}' \mid \mathbf{u}')$:

$$\begin{aligned} \hat{u}_k &= \arg \max_{u_k=0,1} \sum_{\sim\{u_k\}} p(\mathbf{r}, \mathbf{r}' \mid \mathbf{u}, \mathbf{u}') p(\mathbf{u}' \mid \mathbf{u}) \\ &= \arg \max_{u_k=0,1} \sum_{\sim\{u_k\}} p(\mathbf{r} \mid \mathbf{u}) p(\mathbf{r}' \mid \mathbf{u}') p(\mathbf{u}' \mid \mathbf{u}), \end{aligned} \tag{3.16}$$

where $\sum_{\sim\{u_k\}}$ here is the sum over all bits of $\mathbf{u}$ and $\mathbf{u}'$ except u_k. The factor $p(\mathbf{u}' \mid \mathbf{u})$ reflects the uncertainty of $\mathbf{u}'$ and is responsible for the differences with the modified and standard DTC decoding algorithm. In (3.16), the first two factors relate to the decoding results of two convolutional decoders, and the third relates to the information exchange between them. The exact description of the factor $p(\mathbf{u}' \mid \mathbf{u})$ would require an error model in the sequence $\mathbf{u}'$, resulting from its coded transmission through the source-relay channel. A simple yet clearly suboptimal assumption is to consider the errors as independent, neglecting the correlation between the errors at the output of the convolutional decoder. This assumption is given as follows:

$$p(\mathbf{u}' \mid \mathbf{u}) = \prod_k p(u_k' \mid u_k), \tag{3.17}$$

where $p(u_k' \mid u_k)$ is considered as constant for all k. It takes values described as follows:

$$p(u_k^{'} \mid u_k) = \begin{cases} 1 - p_e & if\ u_k^{'} = u_k \\ p_e & if\ u_k^{'} \neq u_k \end{cases}, \tag{3.18}$$

where p_e is the average decoding error rate at the relay, depending on the instantaneous SNR of the source-relay channel.

Given the factorizations (3.16) and (3.17), a factor graph is drawn in Figure 3.7. From this factor graph, we can derive the modified algorithm straightforwardly. The information messages sent by a constituent decoder are modified before they reach the other decoder, according to the following update equations:

$$\begin{aligned} \mu_k^{'}(u_k^{'}) &= \sum_{u_k=0,1} p(u_k^{'} \mid u_k)\mu_k(u_k) \\ \nu_k(u_k) &= \sum_{u_k=0,1} p(u_k^{'} \mid u_k)\nu_k^{'}(u_k^{'}) \end{aligned} \tag{3.19}$$

for all k. This achieves the translation of information available on u_k to information on $u_k^{'}$ and conversely. The other difference of the modified turbo-decoding algorithm is that the second constituent decoder operates on **u'** instead of **u**.

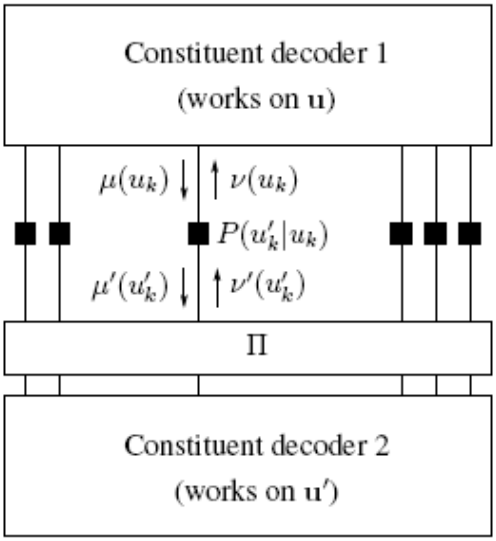


Fig. 3.7 The factor graph of iterative decoding of DTC resilient to relay errors

Some insight on the behavior of the algorithm can be gained by analyzing the exchanges of messages between the two constituent decoders. These exchanges are limited by the transformation (3.19) because their outputs always satisfy

$$\begin{aligned} \mu_k^{'}(u_k^{'}) &\leq 1-p_e \\ \nu_k(u_k) &\leq 1-p_e, \end{aligned} \tag{3.20}$$

which is a straightforward consequence of (3.19) since $\mu_k(0)+\mu_k(1)=1$ and $\nu_k^{'}(0)+\nu_k^{'}(1)=1$. This reflects that even the certitude on the value of the bit $u_k^{'}$ only enables us to determine the value of the bit u_k with a probability $1-p_e$, since there is still a probability p_e that these two bits differ. No more improvements are possible when the bounds (3.20) are reached. However, further performance improvements are possible if we do not use the assumption (3.17) and thus its consequence (3.19) and (3.20). The analysis of the exchange of information during the decoding process explains how the probability of error on the source-relay channel may limit the performance at the destination.

Chapter 4 Expectation maximization algorithm

In this chapter, we specify the maximum-likelihood parameter estimation problem and how Expectation-Maximization (EM) can be used for its solution. We first describe the maximum-likelihood parameter estimation problem, then give the introduction of the EM algorithm, and finally develop the EM parameter estimation procedure for finding the parameters of a Gaussian mixture.

4.1 Maximum-likelihood estimation

Recall the definition of the maximum-likelihood estimation problem. We have a density function $p(x|\boldsymbol{\Theta})$ that is governed by the set of parameters $\boldsymbol{\Theta}$. We also have a data set of size N, supposedly drawn from this distribution, i.e., $\mathbf{X} = \{x_1, ..., x_N\}$. That is, we assume that these data are independent and identically distributed (i.i.d.) with distribution p. Therefore, the resulting density for the samples is

$$p(\mathbf{X}|\boldsymbol{\Theta}) = \prod_{i=1}^{N} p(x_i|\boldsymbol{\Theta}). \tag{4.1}$$

This function is called the likelihood of the parameters given the data. The likelihood is thought of as a function of the parameters $\boldsymbol{\Theta}$ where the data $\mathbf{X}$ is fixed. In the maximum likelihood problem, our goal is to find the $\boldsymbol{\Theta}$ that maximizes (4.1). That is, we wish to find $\boldsymbol{\Theta}_{opt}$ where

$$\boldsymbol{\Theta}_{opt} = \arg\max_{\boldsymbol{\Theta}} p(\mathbf{X}|\boldsymbol{\Theta}). \tag{4.2}$$

Often we maximize $\log(p(\mathbf{X}|\boldsymbol{\Theta}))$ instead of $p(x|\boldsymbol{\Theta})$ because it is analytically easier.

Depending on the form of $p(x|\boldsymbol{\Theta})$, this problem can be easy or difficult. For example, if

$p(x|\boldsymbol{\Theta})$ is simply a single Gaussian distribution where $\boldsymbol{\Theta}=(\mu,\sigma^2)$, then we can set the derivative of $\log(p(\mathbf{X}|\boldsymbol{\Theta}))$ to zero, and solve directly for μ and σ^2 (this, in effect, results in the standard formulas for the mean and variance of a data set). For many problems, however, it is not possible to find such analytical expressions, and we must resort to more elaborate techniques.

4.2 Basic expectation-maximization estimation

The EM algorithm is one such elaborate technique. The EM algorithm is a general method for finding the maximum-likelihood estimate of the parameters of an underlying distribution from a given data set when the data is incomplete or has missing values. There are two main applications of the EM algorithm. The first occurs when the data indeed has missing values, due to problems with or limitations of the observed process. The second occurs when optimizing the likelihood function is analytically intractable but when the likelihood function can be simplified by assuming the existence of additional but missing (or hidden) parameters. The latter application is the solution to what we have encountered in the previous chapter.

As before, we assume that data $\mathbf{X}$ is observed and is generated by some distribution. We call $\mathbf{X}$ the incomplete data. We assume that a complete data set exists $\mathbf{Z}=(\mathbf{X},\mathbf{Y})$ and also specify a jointly density function:

$$p(z|\boldsymbol{\Theta}) = p(x,y|\boldsymbol{\Theta}) = p(y|x,\boldsymbol{\Theta})\,p(x|\boldsymbol{\Theta}). \tag{4.3}$$

Where does this jointly density come from? Often it arises from the marginal density function $p(x|\boldsymbol{\Theta})$ and the assumption of hidden variable and parameter value guesses.

With this new density function, we can define a new likelihood function $p(\mathbf{X},\mathbf{Y}|\boldsymbol{\Theta})$,

called the complete-data likelihood. Note that this function is in fact a random variable since the missing information $\mathbf{Y}$ is unknown, random, and presumably governed by an underlying distribution. That is, we can think of $p(\mathbf{X},\mathbf{Y}|\boldsymbol{\Theta}) = h_{\mathbf{X},\boldsymbol{\Theta}}(\mathbf{Y})$ for some function $h_{\mathbf{X},\boldsymbol{\Theta}}(\cdot)$ where $\mathbf{X}$ and $\boldsymbol{\Theta}$ are constant and $\mathbf{Y}$ is a random variable. The original likelihood $p(\mathbf{X}|\boldsymbol{\Theta})$ is referred to as the incomplete-data likelihood function.

The EM algorithm first finds the expected value of the complete-data log-likelihood $\log p(\mathbf{X},\mathbf{Y}|\boldsymbol{\Theta})$ with respect to the unknown data $\mathbf{Y}$ given the observed data $\mathbf{X}$ and the current parameter estimates. That is, we define:

$$Q(\boldsymbol{\Theta},\boldsymbol{\Theta}^{(i-1)}) = E[\log p(\mathbf{X},\mathbf{Y}|\boldsymbol{\Theta})|\mathbf{X},\boldsymbol{\Theta}^{(i-1)}], \tag{4.4}$$

where $\boldsymbol{\Theta}^{(i-1)}$ are the current parameters estimates that we used to evaluate the expectation and $\boldsymbol{\Theta}$ are the new parameters that we optimize to increase Q. The expression (4.4) probably requires some explanation. The key thing to understand is that $\mathbf{X}$ and $\boldsymbol{\Theta}^{(i-1)}$ are constants, $\boldsymbol{\Theta}$ is a normal variable that we wish to adjust, and $\mathbf{Y}$ is a random variable governed by the distribution $f(\mathbf{y}|\mathbf{X},\boldsymbol{\Theta}^{(i-1)})$. The righthand side of (4.4) can therefore be re-written as:

$$E[\log p(\mathbf{X},\mathbf{Y}|\boldsymbol{\Theta})|\mathbf{X},\boldsymbol{\Theta}^{(i-1)}] = \int_{\mathbf{y}\in\Upsilon} \log p(\mathbf{X},\mathbf{y}|\boldsymbol{\Theta}) f(\mathbf{y}|\mathbf{X},\boldsymbol{\Theta}^{(i-1)}) d\mathbf{y}\,. \tag{4.5}$$

Note that $f(\mathbf{y}|\mathbf{X},\boldsymbol{\Theta}^{(i-1)})$ is the marginal distribution of the unobserved data and is dependent on both the observed data $\mathbf{X}$ and on the current parameters $\boldsymbol{\Theta}^{(i-1)}$, and Υ is the space of values $\mathbf{Y}$ can take on. In the best of the cases, this marginal distribution is a simple analytical expression of the assumed parameters $\boldsymbol{\Theta}^{(i-1)}$ and perhaps the data. In the worst of the cases, this density might be very hard to obtain. The evaluation of this expectation is called the E-step of the algorithm.

The second step (the M-step) of the EM algorithm is to maximize the expectation we computed in the first step. That is, we find:

$$\boldsymbol{\Theta}^{(i)} = \arg\max_{\boldsymbol{\Theta}} Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(i-1)}). \tag{4.6}$$

These two steps are then iterated. Each iteration is guaranteed to increase the log-likelihood and the algorithm is guaranteed to converge to a local maximum of the likelihood function. There are many works regarding the convergence problem, but we will not discuss them here. As presented above, it is not clear how exactly to code up the algorithm. The details of the steps required to compute the given quantities are strongly dependent on the particular application.

4.3 Gaussian mixture identification via EM algorithm

The mixture-density parameter estimation problem is probably one of the most widely used applications of the EM algorithm in the computational pattern recognition community. In this case, we assume the following probabilistic model:

$$p(x \mid \boldsymbol{\Theta}) = \sum_{i=1}^{M} \alpha_i p_i(x \mid \theta_i), \tag{4.7}$$

where the parameters are $\boldsymbol{\Theta} = \{\alpha_1, ..., \alpha_M, \theta_1, ..., \theta_M\}$ such that $\sum_{i=1}^{M} \alpha_i = 1$ and p_i is a density function parameterized by θ_i. In other words, we assume that we have M component densities mixed together with M mixing coefficients α_i.

The incomplete-data log-likelihood expression for this density from the data $\mathbf{X}$ is given by:

$$\log(p(\mathbf{X} \mid \boldsymbol{\Theta})) = \log \prod_{i=1}^{N} p(x_i \mid \boldsymbol{\Theta}) = \sum_{i=1}^{N} \log(\sum_{j=1}^{M} \alpha_j p_j(x_i \mid \theta_j)), \tag{4.8}$$

which is difficult to maximize because it is highly nonlinear. If we consider $\mathbf{X}$ as incomplete,

however, and posit the existence of unobserved data item $\mathbf{Y} = \{y_i\}_{i=1}^N$ whose values indicating which component density generated each data item, the likelihood expression can be significantly simplified. Assume that $y_i \in 1,...,M$ for each i, and $y_i = k$ if the i_{th} sample was generated by the k_{th} mixture component. If we know the values of $\mathbf{Y}$, the likelihood becomes:

$$\log(p(\mathbf{X},\mathbf{Y}\mid\mathbf{\Theta})) = \sum_{i=1}^{N}\log(p(x_i\mid y_i)p(y_i)) = \sum_{i=1}^{N}\log(\alpha_{y_i}p_{y_i}(x_i\mid\theta_{y_i})) \tag{4.9}$$

which is a particular form of the component densities, and can be easily optimized using a variety of techniques. The problem, of course, is that we do not know the values of $\mathbf{Y}$. If we assume $\mathbf{Y}$ is a random vector, however, we can proceed.

We first must derive an expression for the distribution of the unobserved data. Let's first guess that $\mathbf{\Theta}^g = \{\alpha_1^g,...,\alpha_M^g,\theta_1^g,...,\theta_M^g\}$. Given $\mathbf{\Theta}^g$, we can easily compute $p_j(x_i\mid\theta_j^g)$ for each i and j. In addition, the mixing parameters, α_j can be thought of as prior probabilities of each mixture component, that is $\alpha_j = p(\textit{component } j)$. Therefore, using Bayes's rule, we can compute:

$$p(y_i\mid x_i,\mathbf{\Theta}^g) = \frac{\alpha_{y_i}^g p_{y_i}(x_i\mid\theta_{y_i}^g)}{p(x_i\mid\theta^g)} = \frac{\alpha_{y_i}^g p_{y_i}(x_i\mid\theta_{y_i}^g)}{\sum_{k=1}^{M}\alpha_k^g p_k(x_i\mid\theta_k^g)} \tag{4.10}$$

and

$$p(\mathbf{y}\mid\mathbf{X},\mathbf{\Theta}^g) = \prod_{i=1}^{N} p(y_i\mid x_i,\mathbf{\Theta}^g) \tag{4.11}$$

where $\mathbf{y} = (y_1,...,y_N)$ is the unobserved data drawn independently.

When we now look at (4.5), we see that we have obtained the desired marginal density by assuming the existence of the hidden variables and making a guess for the initial parameters

of their distribution. In this case, (4.4) takes the form:

$$
\begin{aligned}
Q(\boldsymbol{\Theta},\boldsymbol{\Theta}^g) &= \sum_{\mathbf{y}\in\Upsilon} \log(p(\mathbf{X},\mathbf{y}\mid\boldsymbol{\Theta})\,p(\mathbf{y}\mid\mathbf{X},\boldsymbol{\Theta}^g) \\
&= \sum_{\mathbf{y}\in\Upsilon}\sum_{i=1}^{N} \log(\alpha_{y_i} p_{y_i}(x_i\mid\theta_{y_i}))\prod_{j=1}^{N} p(y_j\mid x_j,\boldsymbol{\Theta}^g) \\
&= \sum_{y_1=1}^{M}\sum_{y_2=1}^{M}\ldots\sum_{y_N=1}^{M}\sum_{i=1}^{N} \log(\alpha_{y_i} p_{y_i}(x_i\mid\theta_{y_i}))\prod_{j=1}^{N} p(y_j\mid x_j,\boldsymbol{\Theta}^g) \qquad (4.12)\\
&= \sum_{y_1=1}^{M}\sum_{y_2=1}^{M}\ldots\sum_{y_N=1}^{M}\sum_{i=1}^{N}\sum_{l=1}^{M} \delta_{l,y_i}\log(\alpha_l p_l(x_i\mid\theta_l))\prod_{j=1}^{N} p(y_j\mid x_j,\boldsymbol{\Theta}^g) \\
&= \sum_{l=1}^{M}\sum_{i=1}^{N} \log(\alpha_l p_l(x_i\mid\theta_l))\sum_{y_1=1}^{M}\sum_{y_2=1}^{M}\ldots\sum_{y_N=1}^{M}\delta_{l,y_i}\prod_{j=1}^{N} p(y_j\mid x_j,\boldsymbol{\Theta}^g).
\end{aligned}
$$

In this form, $Q(\boldsymbol{\Theta},\boldsymbol{\Theta}^g)$ looks fairly daunting, yet it can be greatly simplified. We first note that for $l\in 1,\ldots,M$,

$$
\begin{aligned}
&\sum_{y_1=1}^{M}\sum_{y_2=1}^{M}\ldots\sum_{y_N=1}^{M}\delta_{l,y_i}\prod_{j=1}^{N} p(y_j\mid x_j,\boldsymbol{\Theta}^g) \\
&= \left(\sum_{y_1=1}^{M}\ldots\sum_{y_{i-1}=1}^{M}\sum_{y_{i+1}=1}^{M}\ldots\sum_{y_N=1}^{M}\prod_{j=1,j\neq i}^{N} p(y_j\mid x_j,\boldsymbol{\Theta}^g)\right) p(l\mid x_i,\boldsymbol{\Theta}^g) \qquad (4.13)\\
&= \prod_{j=1,j\neq i}^{N}\left(\sum_{y_j=1}^{M} p(y_j\mid x_j,\boldsymbol{\Theta}^g)\right) p(l\mid x_i,\boldsymbol{\Theta}^g) = p(l\mid x_i,\boldsymbol{\Theta}^g)
\end{aligned}
$$

since $\sum_{i=1}^{M} p(i\mid x_j,\boldsymbol{\Theta}^g)=1$. Using (4.13), we can write (4.12) as:

$$
\begin{aligned}
Q(\boldsymbol{\Theta},\boldsymbol{\Theta}^g) &= \sum_{l=1}^{M}\sum_{i=1}^{N} \log(\alpha_l p_l(x_i\mid\theta_l))\,p(l\mid x_i,\boldsymbol{\Theta}^g) \\
&= \sum_{l=1}^{M}\sum_{i=1}^{N} \log(\alpha_l)\,p(l\mid x_i,\boldsymbol{\Theta}^g) + \sum_{l=1}^{M}\sum_{i=1}^{N} \log(p_l(x_i\mid\theta_l))\,p(l\mid x_i,\boldsymbol{\Theta}^g).
\end{aligned} \qquad (4.14)
$$

To maximize this expression, we can maximize the term containing α_l and the term containing θ_l independently since they are not related.

To find the expression for α_l, we introduce the Lagrange multiplier λ with the constraint that $\sum_l \alpha_l = 1$, and solve the following equation:

$$\frac{\partial}{\partial \alpha_l}[\sum_{l=1}^{M}\sum_{i=1}^{N}\log(\alpha_l)p(l\mid x_i,\boldsymbol{\Theta}^g)+\lambda(\sum_l \alpha_l-1)]=0 \tag{4.15}$$

or

$$\sum_{i=1}^{N}\frac{1}{\alpha_l}p(l\mid x_i,\boldsymbol{\Theta}^g)+\lambda=0 \tag{4.16}$$

Summing both sides over l, we get that $\lambda=-N$. We then have

$$\alpha_l=\frac{1}{N}\sum_{i=1}^{N}p(l\mid x_i,\boldsymbol{\Theta}^g). \tag{4.17}$$

For some distributions, it is possible to get an analytical expression for θ_l. For our application here, the distribution is a mixture of two one-dimensional Gaussian distributions with mean $\mu_1=-\mu_2$ and variance $\sigma_1^2=\sigma_2^2$, as shown below:

$$p_l(x\mid \mu_l,\sigma_l^2)=\frac{1}{\sqrt{2\pi\sigma_l^2}}e^{-\frac{(x-\mu_l)^2}{2\sigma_l^2}},\quad l\in\{1,2\}. \tag{4.18}$$

Taking the log of (4.18), ignoring constant terms, and substituting the result into the right side of (4.14), we get:

$$\begin{aligned}&\sum_{l=1}^{M}\sum_{i=1}^{N}\log(p_l(x_i\mid\theta_l))p(l\mid x_i,\boldsymbol{\Theta}^g)\\&=\sum_{l=1}^{M}\sum_{i=1}^{N}\left(-\frac{1}{2}\log(\sigma_l^2)-\frac{(x_i-\mu_l)^2}{2\sigma_l^2}\right)p(l\mid x_i,\boldsymbol{\Theta}^g)\end{aligned} \tag{4.19}$$

Taking the derivative of (4.19) with respect to μ_l and setting it equal to zero, we get:

$$\sum_{i=1}^{N}\frac{(x_i-\mu_l)}{\sigma_l^2}p(l\mid x_i,\boldsymbol{\Theta}^g)=0. \tag{4.20}$$

with which we can easily solve for μ_l as :

$$\mu_l=\frac{\sum_{i=1}^{N}x_i p(l\mid x_i,\boldsymbol{\Theta}^g)}{\sum_{i=1}^{N}p(l\mid x_i,\boldsymbol{\Theta}^g)}. \tag{4.21}$$

With the means estimated, we could also derive the variance estimate. To find σ_l^2, note that we can rewrite (4.19) as:

$$\sum_{l=1}^{M}\left(\sum_{i=1}^{N}\frac{1}{2}\log(\sigma_l^2)^{-1}\,p(l\mid x_i,\boldsymbol{\Theta}^g)-\sum_{i=1}^{N}\frac{(x_i-\mu_l)^2}{2\sigma_l^2}\,p(l\mid x_i,\boldsymbol{\Theta}^g)\right) \tag{4.22}$$

Taking derivative of (4.22) with respect to $(\sigma_l^2)^{-1}$ and setting it equal to zero, we get:

$$\sum_{i=1}^{N}\frac{1}{2}\sigma_l^2\,p(l\mid x_i,\boldsymbol{\Theta}^g)-\sum_{i=1}^{N}\frac{(x_i-\mu_l)^2}{2}\,p(l\mid x_i,\boldsymbol{\Theta}^g)=0 \tag{4.23}$$

with which we can also easily solve for σ_l^2 to obtain:

$$\sigma_l^2=\frac{\sum_{i=1}^{N}(x_i-\mu_l)^2\,p(l\mid x_i,\boldsymbol{\Theta}^g)}{\sum_{i=1}^{N}p(l\mid x_i,\boldsymbol{\Theta}^g)} \tag{4.24}$$

A complete EM process for a Gaussian mixture identification is derived. The E-step first finds the expected value of the complete-data log-likelihood. The M-step obtains the new estimates by maximizing the expectation we computed in the E-step. It has been shown that each iteration is guaranteed to increase the log-likelihood and the algorithm is guaranteed to converge to a local maximum of the likelihood function. Since the global maximum is not guaranteed for the EM method, we have to choose the initial values carefully.

In summary, we can have the estimates of the new parameters in terms of the old parameters as:

$$\alpha_l^{new}=\frac{1}{N}\sum_{i=1}^{N}p(l\mid x_i,\boldsymbol{\Theta}^g);\quad \mu_l^{new}=\frac{\sum_{i=1}^{N}x_i\,p(l\mid x_i,\boldsymbol{\Theta}^g)}{\sum_{i=1}^{N}p(l\mid x_i,\boldsymbol{\Theta}^g)};\quad \sigma_l^{2\ new}=\frac{\sum_{i=1}^{N}(x_i-\mu_l^{new})^2\,p(l\mid x_i,\boldsymbol{\Theta}^g)}{\sum_{i=1}^{N}p(l\mid x_i,\boldsymbol{\Theta}^g)}. \tag{4.25}$$

Note that the above equations perform both the expectation step and the maximization step simultaneously. The algorithm proceeds by using the newly derived parameters as the guess for the next iteration.

Chapter 5 Simulation results

In this chapter, we report simulation results to evaluate the performance of various forwarding strategies, including AF, repetition-DF, DAF, DTC, DTC-DAF, DTC resilient to relay errors (hereafter, abbreviated as DTC-R), and Soft-DF under different channel conditions. The system model is what we have described in Section 2.2.

For the distributed turbo code, we employ the most referred turbo code describe in [17], a code with rate-1/2 eight-state constituent codes and the generator polynomials $G(1,17/13)_{oct}$. The overall code rate is 1/3. A source data block has N = 992 bits. And the cooperative ratio is 33%. To have fair comparisons, all forwarding strategies have the same constituent codes and cooperative ratio except for non-cooperation (NC) scenario and outage mode for DTC-R. The baseline for all comparisons is a non-cooperative RSC coded system. In AF and DAF, the relay will forward the normalized channel-LLRs and decoder-LLRs of the systematic bits, respectively, forming a "distributed RSC-repetition" code. In repetition-DF, the relay will forward the systematic bits only when CRC passes. For DTC and DTC-R, the relay will, upon correct retrieval of the systematic bits, scramble and re-encode them using the same code, and forward the new set of parity bits to the destination, thus completing a distributed turbo code (two iterations at destination). Instead, Soft-DF relay forwards the parity bits' LLRs computed using the non-recursive convolutional soft-encoder with generator polynomials $G(3)_{oct}$ [4].

Figure 5.1 depicts the two different scenarios of Rayleigh fading channel under which we evaluate the performance of the various forwarding strategies. In Figure 5.1(a), we fix the SNR in two user channels SD & RD and evaluate the system performance for different SR SNRs. In Figure 5.1(b), we equate the SNR in three channels and evaluate the system

performance for different SNRs. We call the former as Scenario 1 and the latter as Scenario 2.

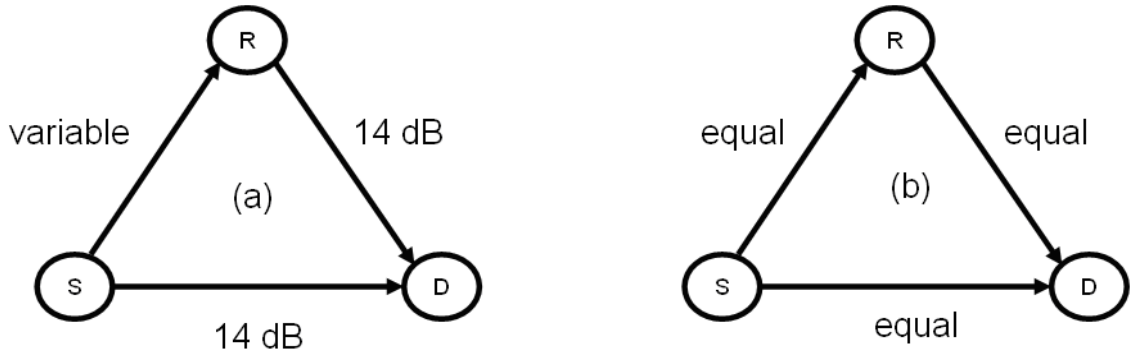


Fig. 5.1 Two different scenarios for the three-terminal network

In Section 5.1, we first evaluate the performance of DAF (dichotomy method or EM algorithm) and the SNR enhancement at the DAF relay. In Section 5.2, we then evaluate the performance of various forwarding strategies under the condition of Figure 5.1(a). Similarly, Section 5.3 is for Figure 5.1(b).

5.1 Performance of DAF

In this section, we show that the EM algorithm can indeed enhance the performance of DAF scheme especially under the low SNR regime. We compare two methods to identify the Gaussian mixture of a DAF relay. The first one is the dichotomy method, separating LLRs into two clusters by zero and regarding each cluster as a single Gaussian. It uses a rather simple scheme for the estimation of LLR statistics. The second one is the EM algorithm. They are hereafter abbreviated as DAF-D and DAF-EM. Figure 5.2 shows the simulation result in terms of the block error rate, comparing AF, DAF-D, and DAF-EM. As shown in this figure, DAF-EM has approximately 1 dB gain compared with DAF-D under low SNRs regime. Figure 5.3 shows the simulation result in terms of the bit error rate. We can also observe 1 dB

gain under the low SNR regime. This means that a better estimation of the means and variances of Gaussian mixtures indeed increases the possibility of correctly retrieving the information. As can be seen from (4.25), the computational complexity of the EM algorithm is low. The reason why EM algorithm does not work for high SNRs regime is because two Gaussian densities will be less probable to mix together as shown in Figure 3.2, making the distribution identification scheme in DAF-D work well.

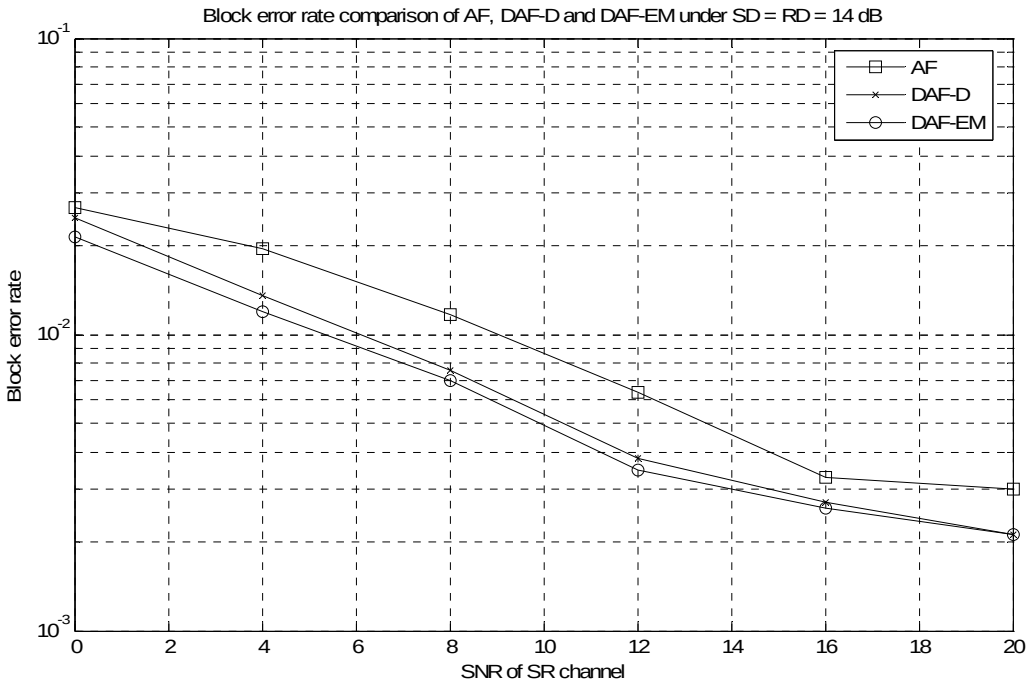


Fig. 5.2 DAF block error rate comparison

As mentioned in Section 3.3, DAF is like an AF strategy operating on an enhanced source-relay channel. Thus, the effective SNR of the virtual channel should be larger than the real SNR of the source-relay channel. This is what we observe in Figure 5.4. We can see that the output SNR is about eight times larger than the input SNR at the DAF relay. This enhancement is made possible by the soft decoder at the relay, i.e. the decoding gain.

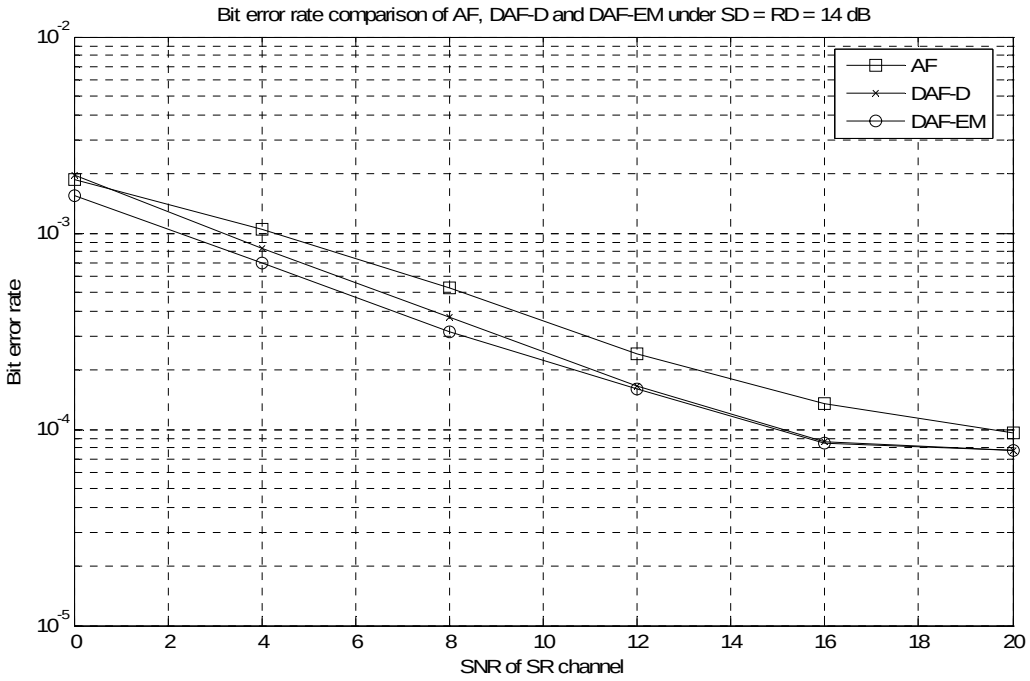


Fig. 5.3 DAF bit error rate comparison

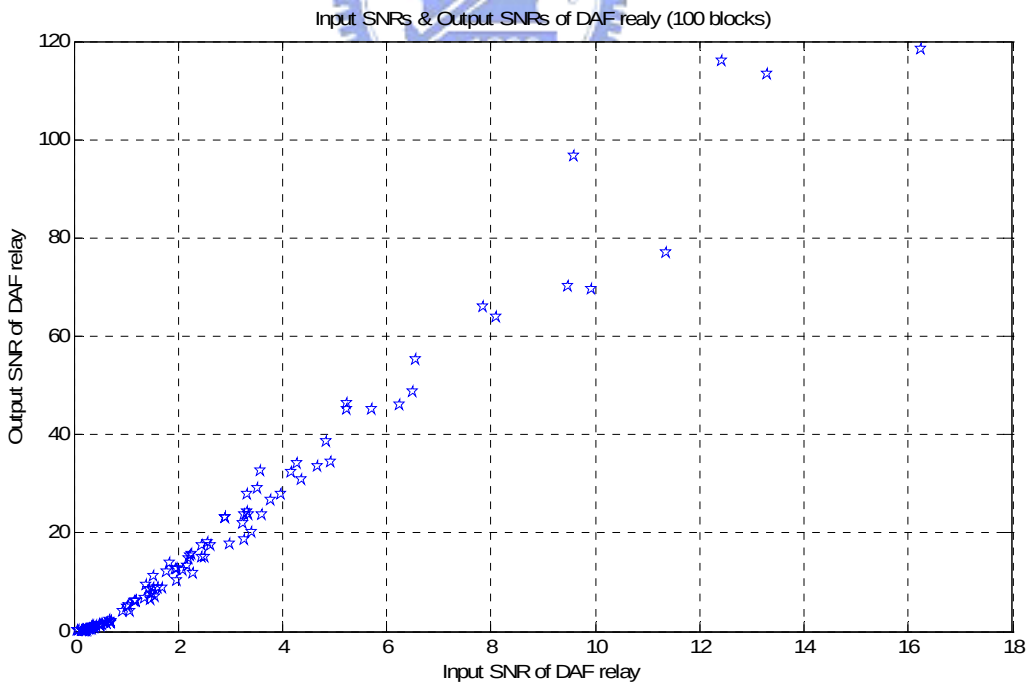


Fig. 5.4 Input-output SNRs of DAF relay

5.2 Performance comparison for Scenario 1

In this section, we compare the block error rate and bit error rate of various forwarding strategies under the conditions depicted in Figure 5.1(a). Just as what we have commented on the achievable rate of a symmetric relay channel in Figure 3.4, Figure 5.5 confirms that DAF is most advantageous at the low SR SNRs regime. DAF performs close to repetition-DF at high SNRs. With increasing SR SNR, DTC will surpass DAF quickly, reaching a lower error floor. And Figure 5.6 further shows that DTC-DAF offers a superior performance from low to high SNRs regime. DTC-R has a better BLER than DTC at low SNRs regime.

Also we can see that Soft-DF does not perform well. There are some reasons behind this. As mentioned before, the Soft-DF scheme instructs the relay to perform not only soft-decoding, but also SISO-encoding at all times. Despite the refreshing idea of SISO-encoding and the possibility to exploit distributed coding in all channel conditions, Soft-DF may not provide real advantages. First, when the hard decisions at the relay are correct (i.e. signs are correct), the most efficient way is to transmit the hard decisions. Attempting to further process and transmit LLRs, such as SISO-encoding in Soft-DF, is not only expensive but also inefficient. Second, when the hard decisions are not all correct, to further encode the unreliable soft information using a channel code is not a good idea either. This might cause severe error propagation. An impact related to error propagation is that SISO-encoding actually makes the worst case even worse, which is not desirable since communication is about rare events (error events are rare) and the worst case tends to dominate the performance.

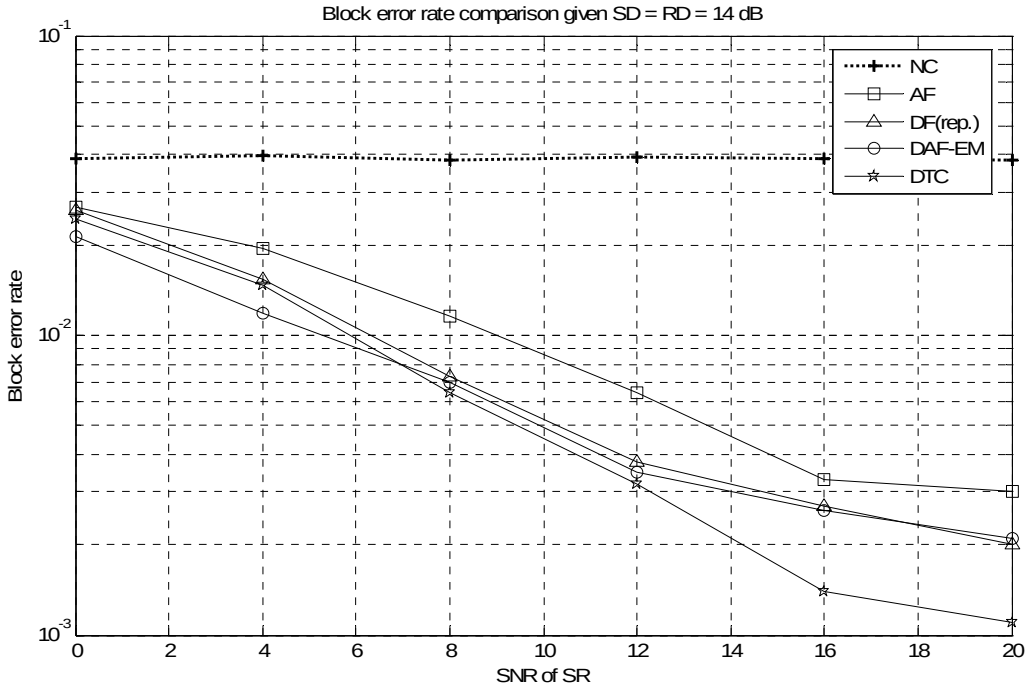


Fig. 5.5 Block error rate of AF, DF (rep.), DAF and DTC under Scenario 1

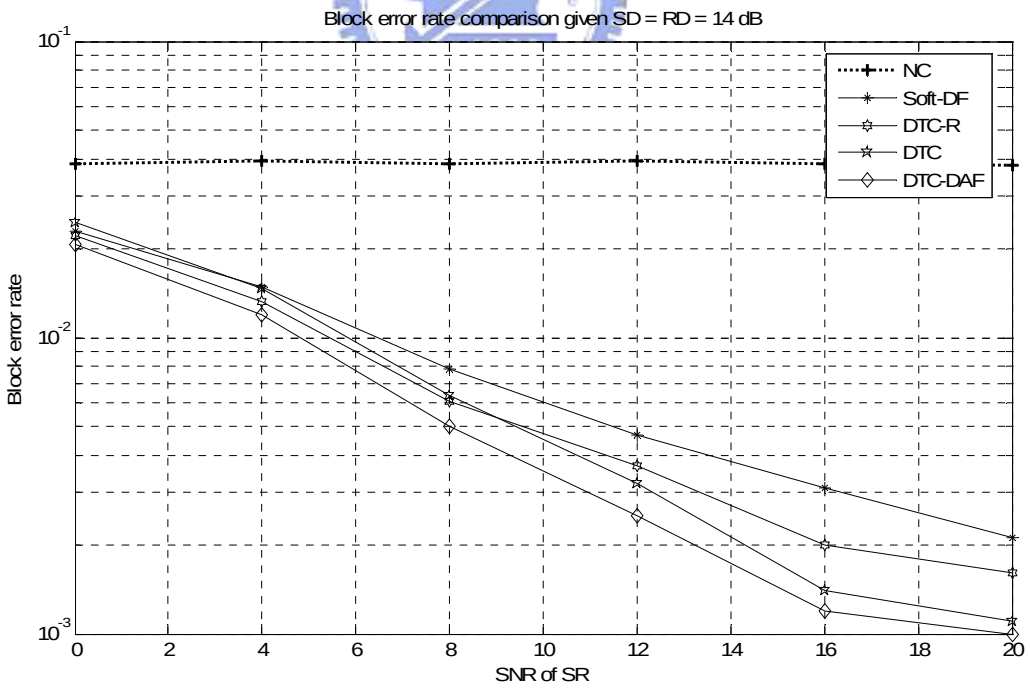


Fig. 5.6 Block error rate of Soft-DF, DTC-R, DTC and DTC-DAF under Scenario 1

Figure 5.7 and Figure 5.8 instead show the bit error rate of these forwarding strategies.

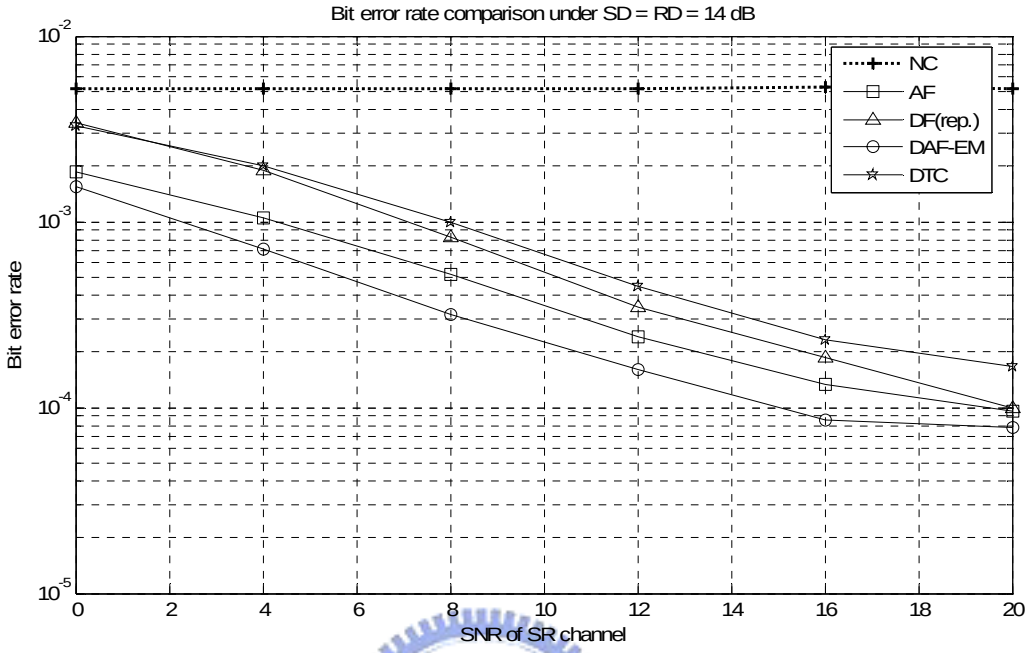


Fig. 5.7 Bit error rate of AF, DF (rep.), DAF-EM and DTC under Scenario 1

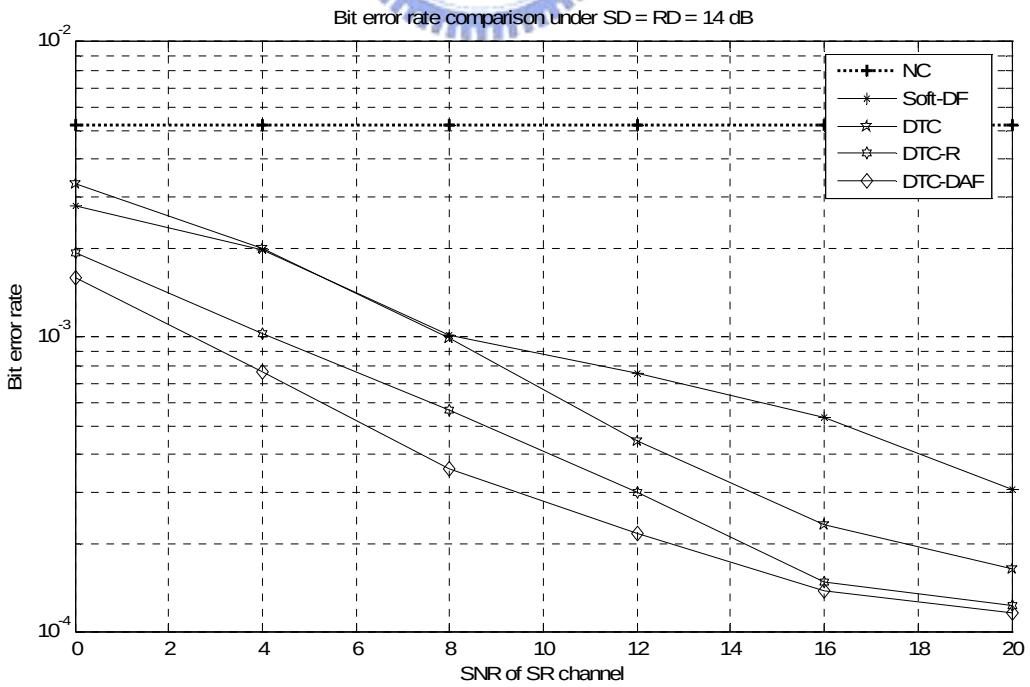


Fig. 5.8 Bit error rate of Soft-DF, DTC-R, DTC and DTC-DAF under Scenario 1

5.3 Performance comparison for Scenario 2

Similarly, in this section, we compare the block error rate of various forwarding strategies under the condition in Figure 5.1(b). Figure 5.9 shows the performance comparison for AF, repetition-DF, and the non-cooperation scheme. As we can see, there are 12~13 dB gains at the block error rate of 10^{-3} due to cooperative diversity.

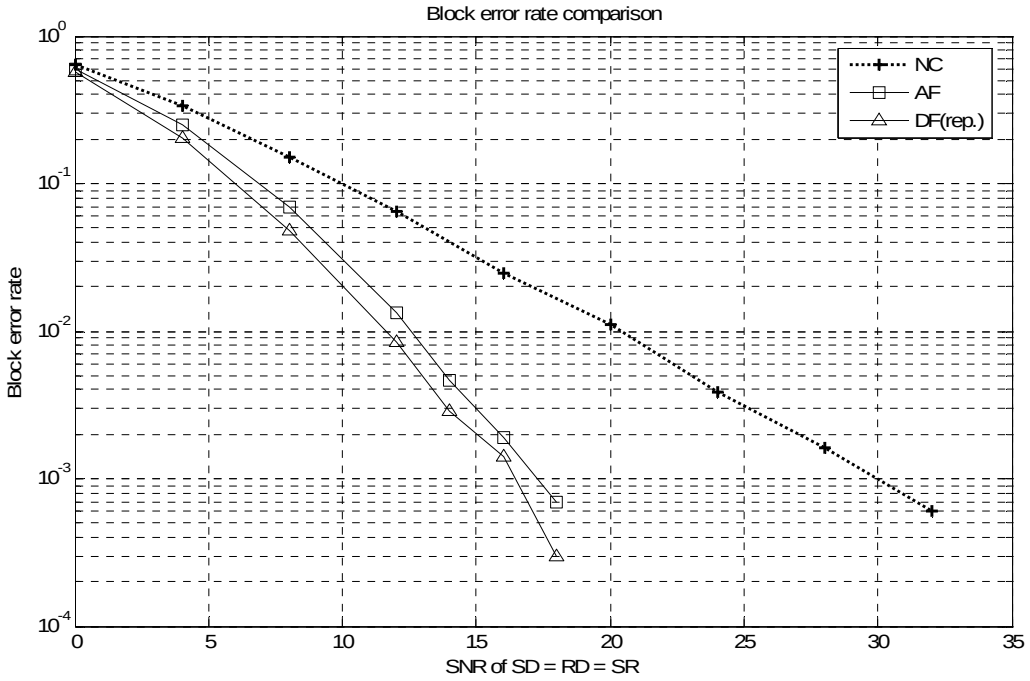


Fig. 5.9 Block error rate of AF and repetition-DF and non-cooperation under Scenario 2

Figure 5.10 shows that DAF performs close to repetition-DF in terms of the block error rate. And DTC has a gain at least 0.7 dB over them. Further, as can be seen in Figure 5.11, Soft-DF performs worse about 0.8 dB than DTC. The rationale behind this has been described in the previous section. Moreover, we can still observe a gain up to 0.5 dB of DTC-DAF scheme over DTC scheme. The performance of DTC-R is very close to that of DTC except for a 0.2 dB gain over low SNRs regime.

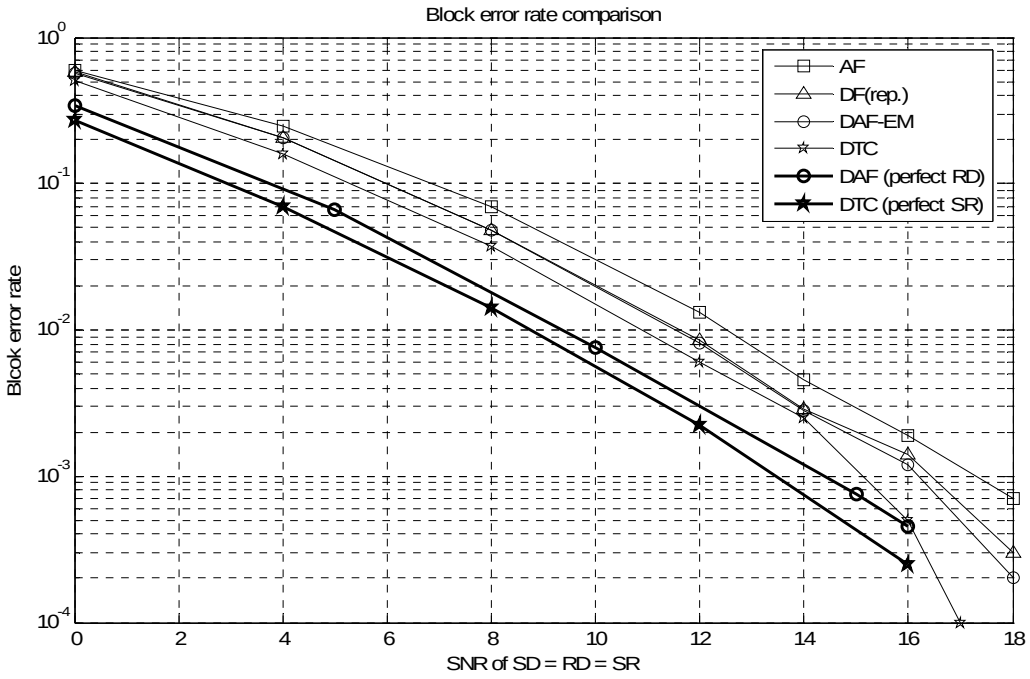


Fig. 5.10 Block error rate of AF, DF (rep.), DAF, DTC and two perfect schemes under Scenario 2

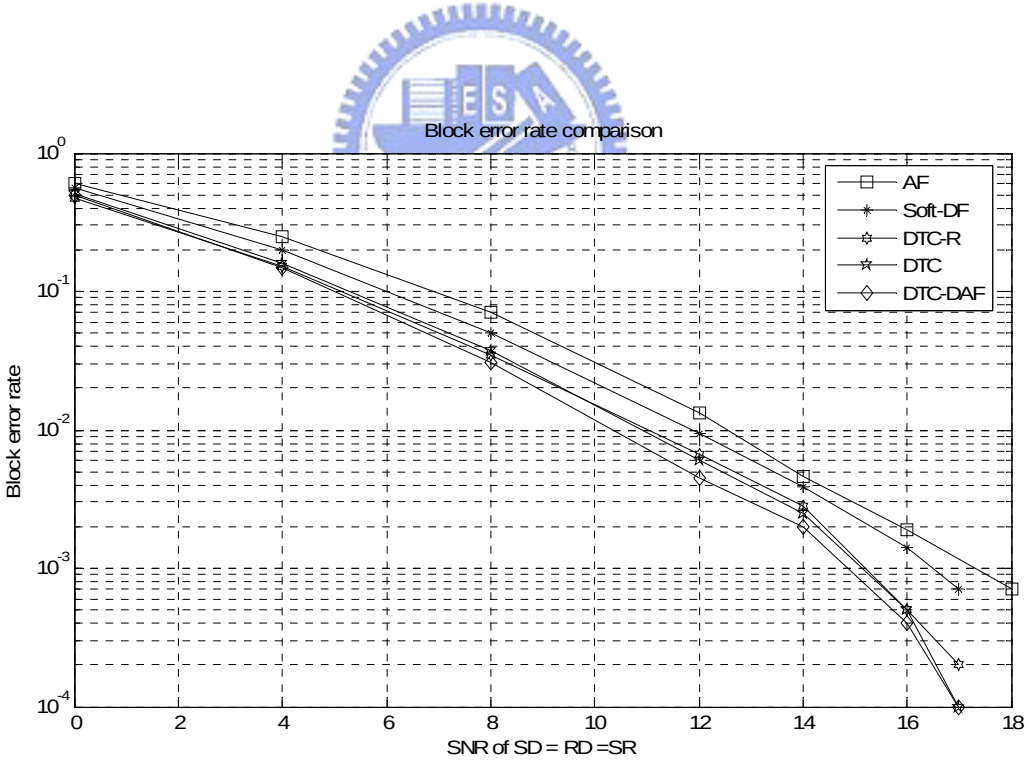


Fig. 5.11 Block error rate of AF, Soft-DF, DTC-R, DTC and DTC-DAF under Scenario 2

Chapter 6 Conclusions

In this thesis, we study the distributed turbo coding (DTC) and related schemes. It has been shown that DTC can approach the capacity of a wireless relay network. Practical DTC has to switch to non-cooperative transit mode should inter-user outage occur, since erroneous bits, if (re-encode and) forwarded to the destination, will mislead the destination and cause severe error propagation. This non-cooperative transit mode degrades the performance. A forwarding strategy termed DAF is then developed to solve the problem. DAF is not only superior, but also more computationally efficient. However, DAF requires the relay to have the LLR distribution, typically modeled as a Gaussian mixture. In this thesis, we propose to use the EM algorithm for the identification of the Gaussian mixture. Simulations show that the EM algorithm can help DAF to retrieve the information more correctly, without adding too much processing complexity at the relay. Further, DAF can be combined with DTC resulting in a DTC-DAF scheme. It is shown that DTC-DAF can have the best performance among DTC related schemes we studied.

In concluding the work, we outline some possible topics for further research. First, our simulation results here are obtained with BPSK mapping. In order to increase the bandwidth efficiency, higher-order QAM mapping should be employed. This comes out with a problem of soft information relaying, i.e. how to map soft bits onto a point in the complex plane. There are few works [15] ~ [16] focusing on this soft mapping problem. However, their actual performance needs further investigation. Second, as shown in Figure 5.10, DAF with perfect RD can have superior performance, encouraging the attempts to transmit LLRs from the relay to the destination. This might be done through the compress-forward (CF) structure, but practical CF schemes come far lagging behind the theory [8] [11].

Reference

[1] B. Zhao and M. C. Valenti, “Distributed turbo coded diversity for relay channel,” *Electron. Lett.*, vol. 39, no. 10, pp. 786-787, May 2003.

[2] B. Zhao and M. C. Valenti, “Distributed turbo codes: Toward the capacity of the relay channel,” in *Proc. VTC-Fall*, vol. 1, pp.322-326,Oct 2003.

[3] M. Janani, A. Hedayat, T. E. Hunter, and A. Nosratinia, “Coded cooperation in wireless communications: Space-time transmission and iterative decoding,” *IEEE Trans. Signal Process.*, vol. 52, pp. 362–371, Feb.2004.

[4] H. H. Sneesens, L.Vandendorpe, “Soft decode and forward improves cooperative communications,” *CAMAP 2005*, pp. 157-160, Dec. 2005.

[5] X. Bao and J. Li, “Decode-amplify-forward (DAF): A new class of forwarding strategy in wireless relay channel,” in *Proc. IEEE Workshop on Signal Processing Advances in Wireless Commun. (SPAWC)*, June, 2005.

[6] X. Bao and J. Li, “Efficient Message Relaying for Wireless User Cooperation: Decode-Amplify- Forward (DAF) and Hybrid DAF and Coded-Cooperation” in *IEEE Trans. on Wireless Commun.* vol. 6, no. 11, pp. 3975–3984, Nov, 2007.

[7] Tuyen Bui, Jinhong Yuan, “A decode and forward cooperation scheme with soft relaying in wireless communication,” *Signal Processing Advances in Wireless Communications 2007* pp. 1 – 5, June 2007.

[8] T. M. Cover and A. A. El Gamal, “Capacity theorems for the relay channel,” *IEEE Trans. Inf. Theory*, vol. IT-25, pp. 572-584, Sept. 1979.

[9] M. Fu, “Stochastic analysis of turbo decoding,” *IEEE Trans. on Inf. Theory*, vol. 51, pp. 81-100, Jan. 2005.

[10] H. H. Sneessens, J. Louveaux, L. Vandendorpe, “Turbo-coded decode-and-forward strategy resilient to relay errors,” *ICASSP 2008*, pp. 3213 – 3216, Apr. 2008.

[11] R. Hu, and J. Li, “Practical compress-forward in user cooperation: Wyner-Ziv cooperation,” in *Proc. IEEE ISIT*, pp 489-493, July 2006.

[12] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate,” *IEEE Trans.on Inf. Theory*, vol. IT-20, pp. 284-287, Mar. 1974.

[13] C. Berrou and A. Glavieux, and P. Thitimajshima, “Near Shannon limit error-correcting coding and decoding: Turbo-codes,” *IEEE ICC’93 in Proc.*, pp. 1261-1271, Oct. 1996.

[14] Jeff A. Bilmes, “A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models” International Computer Science Institute Berkerley CA.

[15] R. Hoshyar, R. Tafazolli, “Soft Decode and Forward of MQAM Modulations for Cooperative Relay Channels,” *IEEE VTC Spring 2008*, pp. 639 – 643, May 2008.

[16] Y. Li, B. Vucetic, T. F. Wong, M. Dohler, “Distributed Turbo Coding With Soft Information Relaying in Multihop Relay Networks,” *IEEE Journal on Select. Areas in Commun. (JSAC)* Vol. 24, No. 11, pp. 2040- 2050, Nov 2006.

[17] S. Benedetto, R. Garello, and G. Montorsi, “A search for good convolutional codes to be used in the construction of turbo codes,” *IEEETrans. Commun.*, vol. 46, pp. 1101–1105, Sept.1998.