Soft-Encoding Distributed Coding for Parallel Relay Systems

Xuanxuan Lu, Jing Li (Tiffany), Yang Liu
Electrical and Computer Engineering Department, Lehigh University, Bethlehem, PA 18015
Email: {xul311, jingli, yal210}@ece.lehigh.edu

Abstract—**In the paper, a new distributed coding scheme for parallel relay systems is proposed, in which a sender communicates to a destination that is two hops away via two (or more) parallel relays. The key idea is the exploitation of a (rate-1) soft convolutional encoder at each of the parallel relays, to collaboratively form a simple but powerful distributed *analog* coding scheme to achieve efficient forwarding of soft reliability messages. We detail the encoding and decoding process of the proposed soft-encoding distributed coding. As the input of the encoder would affect the overall performance, we analyze what form of messages at the relay is most appropriate to be forwarded to the destination. The range-limited log likelihood ratio (range-limited LLR) is chosen as the input. The optimality of the range-limited LLRs as the best form of relaying messages is verified by the simulation results. Our new distributed coding scheme can obviously outperform the existing ones.**

I. Introduction and Motivation

We consider a two-hop wireless communication system with a single sender, a single receiver, and a set of parallel relays in the middle. Without having to impose stringent constraints on perfect (baud-level) synchronization among distributed nodes, three practical relaying strategies are frequently considered : amplify-forward (AF), decode-forward (DF) [1], and a combination of both that involves decoding and amplifying and forwarding of soft information [2]. Central to the majority of these relaying strategies is the design and operation of *distributed coding* schemes [3] [4]. Indeed, a fundamental motivation for the relay research is the departure from the conventional point-to-point perspective, which treats the system as a simple accumulation of multiple sender-receiver links, each handling communication (and coding) separately and independently. Since the multi-node multi-link system is now regarded as a cohesive entity of system-on-graph, the codes operated on them must be jointly designed and carefully coordinated to achieve an overall harmony. A pioneering scheme on distributed turbo code was proposed by Valenti in 2005 [4]. Since then, a variety of coding strategies have been proposed where nodes cooperate to construct the overall codeword in a distributed manner, for both digtial distributed code and soft-encoding distributed codes [5] [6], what is this paper intends to address.

From Information Theory, we know that processing reduces entropy. In other words, it instructs that an intermediate processor – which, in the context of a relay network, corresponds to an intermediate relay node – should delay making a hard decision (and other forms of quantization) as much as possible, unless it is absolutely sure of the hard decisions. This naturally leads to the forwarding of soft-messages, such as the log-likelihood ratio (LLR) values [2] or some function of them [7] [8]. For systems with uncoded source-relay and relay-destination transmissions, [7] shows that the best soft messages a relay can forward is the hyperbolic tangent of one half of the LLR values, $\tanh(L_x/2)$ (eg: the estimate value of the source data), where L_x is the LLR of the data x (that is extracted directly from the source-relay channel output). The work of [6] further extends this study by demonstrating a novel soft-encoding strategies on $\tanh(L_x/2)$ to provide the much-needed protection in the relay-destination transmission.

This paper considers developing practical and efficient soft-encoding distributed coding strategies for multi-relay systems. The issues we are interested in include: what soft messages are most appropriate, how to best prepare them for transmission at the relay(s), and how to best extract useful information from them at the destination. In addressing these questions, we demonstrate a new coding scheme that has lower-complexity but better-performance than the existing systems.

The remainder of the paper is organized as follows. Section II introduces the system model. Section III details the proposed soft-encoding distributed code, including the choice and evaluation of the soft messages, the specific encoding process to protect them, and an ML Viterbi algorithm. Section IV demonstrates Monte Carlo simulations of various scenarios and with different code specifications, with comparison to the existing codes. Finally, Section V concludes the paper.

Notation: (i) Unless otherwise stated, we use boldface letters to denote vectors and matrices, and use regular letter to denote scalars and random variables. (ii) For a random variable x, we use $E[x]$ to denote the mean value of x and $\tilde{x}$ to denote the estimated value of x. (iii) $\mathcal{CN}(0, \sigma^2)$ represents the complex Gaussian distribution with zero mean and per-dimension variance $\sigma^2/2$. $\mathcal{N}(0, \sigma^2)$ represents the real Gaussian distribution with zero mean and variance σ^2. (iv) The subscripts S, R_i, D are used to denote the quantities pertaining to the source, the ith relay, and the destination.

II. System Model

Consider a two-hop relay system consisting of a source node S, a destination node D, and a set of M relays. Our focus here is in the development of efficient coding scheme, as well as practical and optimal encoding and decoding strategies; Hence, issues regarding relay selection and feedback strategies from

Supported by National Science Foundation under Grants No. CMMI-0928092, CCF-0829888 and OCI-1122027.

receivers to their respective senders are out of the scope of this paper. For simplicity, we assume that all the M relays, $R_i, i = 1, 2 ... M$, are active relays participating in equal parts in the distributed coding and relaying process. We further assume that the channel state information (CSI) is perfectly known by the respective receivers, and that all the source-relay channels and all the relay-destination channels are independent from each other.

Let $\mathbf{m} = (m(1), m(2), ... m(N))$ denote the information at the source S. Without loss of generality, we consider binary phase shift key (BPSK) modulation with $0 \to +1$ and $1 \to -1$. Let h_{SR_i} and h_{R_iD} denote the fading coefficients (CSI) for the respective source-relay and relay-destination channels. All the channels in the system are assumed block fading channels, such that the fading coefficients remain unchanged in each frame. We consider Rayleigh fading, where $h_{SR_i} \sim \mathcal{CN}\left(0, \sigma^2_{SR_i}\right)$, and $h_{R_iD} \sim \mathcal{CN}\left(0, \sigma^2_{R_iD}\right)$. The source and the relays transmit signals with an average signal power of P_S and P_{R_i}, respectively.

Each transmission session of the relay system consists of two phases. In the first phase, the source node broadcasts a block of N modulated (and possibly channel-coded) symbols $\mathbf{x_S} = (x(1), x(2), ... x(N))$, and the ith relay node receives:

$$\begin{aligned} r_{SR_i}(j) = & \sqrt{P_S} h_{SR_i} x_S(j) + n_{R_i}(j), \\ & i = 1, 2, ..., M, \; j = 1, 2, ..., N, \end{aligned} \tag{1}$$

where n_{R_i} denotes the additive white Gaussian noise (AWGN) that follows the complex Gaussian distribution $\mathcal{CN}\left(0, \sigma^2_{R_i}\right)$.

In the second phase, each of the relay nodes R_i extracts the appropriate soft information $\mathbf{l}_i = (l_i(1), l_i(2), \cdots, l_i(N))$ from the received signals $r_{SR_i}(j)$, either directly (when source-relay packets are uncoded) or via channel decoding (when source-relay packets are coded). They then feed the soft information into some specially-designed soft encoder (which we will discuss in detail later), and orthogonally transmit the soft encoded signals $c_i(j)$ to the destination with power normalization. The signal transmitted by the ith relay takes the form of

$$x_{R_i}(j) = \sqrt{\frac{P_{R_i}}{P_{c_i}}} c_i(j) = \alpha_i c_i(j), \tag{2}$$

where, $P_{c_i} = \frac{\sum_{j=1}^{N} (c_i(j))^2}{N}$. Here the soft messages are expected to always have real values, and they are transmitted through amplitude modulation.

For simplicity, we assume that all the relays generate the same number, N_D, of soft signals. Let $\mathbf{r}_{R_iD} = (r_{R_iD}(1), r_{R_iD}(2), ..., r_{R_iD}(N_D))$ denote the corresponding signals the destination receives from these channels. We have:

$$r_{R_iD}(j) = h_{R_iD} x_{R_i}(j) + n_{R_iD} \tag{3}$$

where $j = 1, 2, ..., N_D$, and n_{R_iD} denotes the AWGN with complex Gaussian distribution $\mathcal{CN}(0, \sigma^2_D)$. The destination collects all the receptions $\mathbf{r}_{R_iD}$, performs appropriate soft-input decoding, and make the final estimates, $\mathbf{x}_D$, for the original source bits $\mathbf{x}_S$.

III. Proposed Distributed Soft-Encoding Codes

A. General Idea of Soft Encoding

In what follows, we will focus on the relay-destination transmission, and especially the soft-message preparation and encoding at the relays, and the corresponding decoding at the destination. Before detailing the specific code structure and coding algorithms, we first brief the fundamental concept of soft encoding.

Rather than the general-purpose soft encoding, here we consider a type that is specifically designed for relay purposes. The real-valued data at the input to the encoder are some probabilistic form of binary bits, and the decoder is only interested in the accuracy of the binary decisions (BER) rather than the accuracy of the soft probabilistic data (MSE). The entire code may be regarded as an outgrowth of the conventional linear binary code, where the binary bits are replaced by their probabilistic values, and the parity checks that constrain the binary bits are adapted accordingly (e.g. the "tanh rule").

Take an $(N, N-1)$ binary single parity check code, for example. As a hard-encoding code, the parity bit p is computed via the binary addition (or exclusive-OR) of the source bits:

$$p = x(1) \oplus x(2) \oplus \cdots \oplus x(N-1). \tag{4}$$

The same code, when viewed as a soft-encoding code, possesses the following encoding function:

$$\begin{aligned} & \tanh\left(\frac{1}{2}\log\frac{P(p=0)}{P(p=1)}\right) = \\ & \tanh\left(\frac{1}{2}\log\frac{P(x(1)=0)}{P(x(1)=1)}\right) \tanh\left(\frac{1}{2}\log\frac{P(x(2)=0)}{P(x(2)=1)}\right) \cdots \\ & \tanh\left(\frac{1}{2}\log\frac{P(x(N-1)=0)}{P(x(N-1)=1)}\right), \end{aligned} \tag{5}$$

where the logarithm has base e. Note that $\tanh()$ and $\log()$ are both one-to-one functions, and that $P(x(i)=0)$ relates to $P(x(i)=1)$ via $P(x(i)=0) + P(x(i)=1) = 1$. Hence, for any soft input that takes the probabilistic form or its equivalent, the soft encoder will be able to generate a probabilistic soft output corresponding to the parity bit.

Since any linear binary non-recursive code is essentially a collection of single parity check codes operated on different subsets of the source bits, the soft encoding process described in (5) therefore generalizes naturally to an arbitrary linear binary code. The soft coding scheme can be applied directly to recursive code with minor modification. Due to the space limitation, we omit the detail of coding scheme of recursive code. But we will show some numerical results of the recursive code generated distributed Turbo codes, see Sec. IV.

B. Choice of Soft Messages

We now get back to cooperative systems. The first question that confronts us is what messages the relay should forward. (i) When the relay can successfully decode and demodulate the data from the source-relay transmission, it is without question that the relay should forward these correctly regenerated binary data – all of or part of them (depending on the bandwidth provision), or their encoded versions (for a better protection). (ii) Suppose that the cyclic redundancy check does not pass, such that the relay is equipped with only the compromised data. Clearly the relay should defer the hard decisions to the destination, and for that to happen, it is expected to do its best to pass along soft messages indicating the reliability of the reception.

It might appear that as long as the soft messages represent the probabilistic nature of the estimates[1], such as the probability a bit being "0" $P(x=0)$, the likelihood ratio $P(x=0)/P(x=1)$, the log-likelihood ratio $\log(P(x=0)/P(x=1)$, the hyperbolic tangent of one half of the LLR $\tanh(\frac{1}{2}log\frac{P(x)=0}{P(x)=1})$, or other one-to-one function of $P(x=0)$, the system performances would be same and one, and the only difference is the complexity. This statement is true if the communication channels are perfect and impairment-free. In reality, the performances can be drastically different due to the various factors including the channel noise and distortion, the transmit power constraint, the distribution of the specific soft message in use (which will affect the decoding optimality), as well as the numerical stability in computation. Below we analyze and compare the following popular choices of soft messages (proper scaling is always assumed to satisfy energy constraints):

(i) The probability of a bit being 0, $P(x\!=\!0)$. Since $P(x\!=\!0)$ takes positive values only, transmitting it directly leads to a one-sided signal space that is energy-inefficient. Instead, a scaled-and-shifted version, such as $2P(x=0)-1$, makes for good antipodal signaling. However, we argue that $2P(x\!=\!0)-1$ is not a good choice either, because it can be rather sensitive to additive noise and that the impact of the noise is dependent on the value of $2P(x=0)-1$. Consider, for example, the two cases of $P(x\!=\!0)=0.95$ and $P(x\!=\!0)=0.54$, both of which encounter the same additive noise of -0.10. The former represents a harmless case where the noise causes $2P(x=0)-1$ to change value from 0.90 to 0.80 (or $P(x\!=\!0)=0.90$), which remains a confident judgment for x to be 0. However, the latter becomes a harmful case where $2P(x\!=\!0)-1$ changes from 0.08 to -0.02, indicating a preference change from x being "0" to x being "1". We see that exactly when the soft messages need to be protected the most ($P(x=0)$ around 0.5), is when they are most vulnerable to noise, indicating an inefficient representation.

(ii) The hyperbolic tangent form, $\tanh(\frac{1}{2}log\frac{P(x)=0}{P(x)=1})$ [7]: One of the biggest motivation for using the hyperbolic tangent is its optimality for relaying the received uncoded signal at the relay nodes, with the respect of maximizing the SNR and minimizing the BER at the destination, However, a big pitfall

[1]For notational simplicity, here we omit the subscript i indicating the relay ID and the index j indicating the bit index, as they are irrelevant.

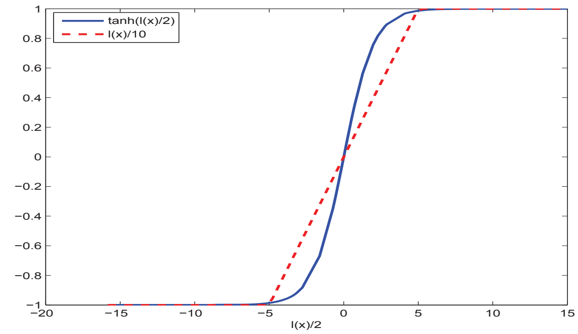

Fig. 1: Curves of $tanh(l/2)$ and $l/10$, $P_s=1$, $SNR_{SR}=3dB$

of the hyperbolic tangent value lies in the fact that

$$\tanh\left(\frac{1}{2}\log(\frac{P(x=0)}{P(x=1)})\right)=\frac{1-e^{-\log(\frac{P(x=0)}{P(x=1)})}}{1+e^{-\log(\frac{P(x=0)}{P(x=1)})}} \\ =2P(x=0)-1, \qquad (6)$$

and hence, for the reasons stated in the above, these soft messages are highly susceptible to noise when they are weak with rather small absolute values, making the worst case even worse. Since communications are all about rare events (such as an error event probability of once in a million), the worst case tends to dominate the performance.

(iii) Range-limited LLR: In this paper, we propose range-limited LLR values as a very efficient choice for soft messages:

$$l(x)=\begin{cases}\theta, & \log\frac{P(x=0)}{P(x=1)}\geq\theta,\\ -\theta, & \log\frac{P(x=0)}{P(x=1)}\leq-\theta,\\ \log\frac{P(x=0)}{P(x=1)}, & \text{otherwise},\end{cases} \qquad (7)$$

where the positive value θ sets the cap for the absolute LLR value. The motivation is several-fold. First and obviously, by judiciously limiting the LLRs to a symmetric bounded range, we can still reap almost all the benefits of LLRs, without having to dealing with infinite or excessively large values. This would considerably reduce the peak-to-average-power ratio (PAPR) and make it easier to control the average transmit power. Second and less apparently, by setting an appropriate cap value, the range-limited LLRs can actually approximate the more important part of the hyperbolic tangent values. Figure 1 plots the curve of $\tanh(l/2)$ vs l, where l represents the LLR values. We may roughly divide the hyper-tangent curve into three sections, the two ends with values very close to $+1$ and -1 which represent the very confident estimates, and the middle section that appears to increase linearly with the LLR value. Specifically, we see that if we limit the LLR values to be within, for example, -10 and 10. The result is an immediate simplification of the soft-encoding rule in (5):

$$l(p)=l(x(1))l(x(2))\cdots l(x(N-1)), \qquad (8)$$

where $l(\cdot)$ represents the range-limited LLR described in (7) with $\theta=10$.

Below we summarize the steps to prepare the soft messages and discuss their distributions (which will be useful in the decoding algorithm): For both uncoded source-relay transmission and coded source-relay transmission during each data frame, the receiving signals follow a Gaussian distribution, and so are the corresponding untruncated LLR values $\mathbf{l}_i$. The range-limited LLR values are no longer Gaussian, but we approximate them to a Gaussian distribution with mean μ_i and variance $\sigma_{l_i}^2$ here. So the LLRs extracted from the source-relay channel, or decoders at the relays can be approximated by:

$$l_i(j) = \mu_i x(j) + n_i \tag{9}$$

where $\mu_i = \frac{\sum_{k=1}^{N} |l_i(k)|}{N}$ represents the mean value of the decoder-LLRs, and n_i represents the approximated Gaussian noise of the decoder-LLRs, with mean zero and variance $\sigma_{l_i}^2 = \frac{\sum_{k=1}^{N} (l_i(k) - \mu_i x(k))^2}{N}$.

C. Soft Encoding Process

A quick summary of the soft-encoding steps of distributed $(5,7)_{oct}$ convolutional code, including power normalization, is provided below.

step 1: Soft encoding. Without loss of generality, suppose $SNR_{SR_1} \le SNR_{SR_2}$. The two relays take in their respective truncated LLR values, feed them into their respective (rate-1) soft encoder, and computes the soft codeword via multiplication:

$$c_1(j) = l_1(j)\, l_1(j-2), \tag{10}$$

$$c_2(j) = l_2(j)\, l_2(j-1)\, l_1(j-2), \tag{11}$$

where $j = 1, 2, \cdots, N$.

step 2: Power normalization. Energy normalization is performed by evaluating the distribution of the encoded soft messages $c_1(j)$ and $c_2(j)$. The input soft messages, as discussed in Subsection III-B, may be approximated as signals communicated via some virtual channel with additive Gaussian noise. Substituting (9) into (10) and (11), we get

$$\begin{aligned} c_1(j) &= (\mu_1 x(j) + n_1(j))\, (\mu_1 x(j-2) + n_1(j-2)) \\ &= \mu_1^2 x(j)\, x(j-2) + \tilde{n}_1, \end{aligned} \tag{12}$$

$$\begin{aligned} c_2(j) &= (\mu_2 x(j) + n_2(j))\, (\mu_2 x(j-1) + n_2(j-1)) \\ &\quad (\mu_2 x(j-2) + n_2(j-2)) \\ &= \mu_2^3 x(j)\, x(j-1)\, x(j-2) + \tilde{n}_2. \end{aligned} \tag{13}$$

Here we have assumed that the virtual noise at each time instant is unrelated, such that $\tilde{n}_1 \sim \mathcal{N}(0, \left(\mu_1^2 + \sigma_1^2\right)^2 - \mu_1^4)$, and $\tilde{n}_2 \sim \mathcal{N}(0, \left(\mu_2^2 + \sigma_2^2\right)^3 - \mu_2^6)$.

Power normalization is performed via eq. (2). Then after power scaling $x_{R_i}(j) = \alpha_i c_i(j)$, the encoded soft messages are transmitted to destination.

D. Maximum Likelihood Decoding

We now revisit the legacy Viterbi algorithm, and make necessary modifications to make it work optimally for our code. The algorithm is otherwise the same as the conventional Viterbi algorithm for digital codes, except the branch metric in the Viterbi decoding algorithm should be adjusted to

$$\begin{aligned} &\frac{\left(r_{R_1D} - h_{R_1D}\alpha_1\mu_1^2 x(j)\, x(j-2)\right)^2}{\tilde{\sigma}_{R_1D}^2} \\ &+ \frac{\left(r_{R_2D} - h_{R_2D}\alpha_2\mu_2^3 x(j)\, x(j-1)\, x(j-2)\right)^2}{\tilde{\sigma}_{R_2D}^2}. \end{aligned} \tag{14}$$

where

$$\begin{aligned} \tilde{\sigma}_{R_1D}^2 &= h_{R_1D}^2 \alpha_1^2 \left(\left(\mu_1^2 + \sigma_1^2\right)^2 - \mu_1^4\right) + \sigma_{R_1D}^2 \\ \tilde{\sigma}_{R_2D}^2 &= h_{R_2D}^2 \alpha_1^2 \left(\left(\mu_2^2 + \sigma_2^2\right)^3 - \mu_2^6\right) + \sigma_{R_2D}^2. \end{aligned} \tag{15}$$

Due to the space limitation, the specific calculation of the branch metric is omitted here.

IV. Numerical Results

We now provide Monte Carlo simulations to evaluate the proposed coding scheme and to compare it with the existing distributed coding schemes. Consider BPSK at the source, and AWGN or block fading for each communication channel. Suppose the data block size is 150. In the simulation results presented here, $M = 2$ parallel relays are considered, each of which generates a rate-1 convolutional codeword and transmits it orthogonally (e.g. time orthogonality) to the receiver to collaboratively form a rate-1/2 distributed code. Three systems are evaluated and compared, all of which use the same average transmitting power per block:

1) Reference system 1 (legend "digital"): a conventional digital distributed code, in which each relay forces a binary hard decision ($+1$ or -1) to the received signal and encodes them via a conventional rate-1 digital encoder, and the destination performs the conventional BCJR decoding;
2) Reference system 2 (legend "tanh"): a soft-encoding distributed code recently proposed in [6], in which the soft-encoders (at the relay) take in the hyperbolic tangent function of the received signals, and the decoder (at the destination) performs a modified BCJR algorithm specifically designed for the individual convolutional code;
3) The new system (legend "LLR"): a soft-encoding distributed code proposed in this paper, in which the soft-encoders take in judiciously-truncated LLRs of the received signals. The cap value in eq. 7 is choose tentatively, here we choose $\theta = 10$. The decoder at the destination performs a modified Viterbi algorithm matched to the individual convolutional code.

We perform a comparative evaluation of these system in a variety of scenarios.

AWGN & block fading channels with uncoded sources and distributed feed-forward convolutional codes: Fig. 2 demonstrates the bit error rate (BER) and frame error rate (FER) performances of all the three systems over AWGN channel. It shows that our scheme improve about 0.5 dB compared with the "tanh" case both for BER and FER performance.

We attribute the additional gain of the new scheme over the previous scheme in [6] to the more appropriate forms of the soft message (i.e. carefully truncated LLRs) and the matching optimal decoder. It should also be noted that by the proposed Viterbi decoding algorithm also requires a lower complexity than the BCJR algorithm.

Fig. 3 presents the FER of the three systems over block fading channels. Our proposed new soft-encoding scheme leads the way by almost 3 dB gain over the previous soft-encoding one.

Block fading channels with uncoded sources and distributed Turbo codes: In order to better exploit the advantage of code cooperation, we provide the FER performance of the distributed soft Turbo codes and conventional ones in Fig. 4. In our scheme, the first relay generates a $(1, 1/1 + D)$ recursive convolutional code, using range-limited LLR as the input, while the inputs of the encoder at the second relay are the interleaved range-limited LLRs. The puncturing matrices for the soft information bits and parity bits are $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ and $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, respectively. The We also set $SNR_{SR_1} = SNR_{SR_2} = SNR_{R_1D} = SNR_{R_2D}$ under block Rayleigh fading channel, our scheme performs much better than the other two.

V. Conclusion

We have proposed a new soft-encoding distributed coding scheme for parallel relay systems. Unlike the previous work that favors the hyperbolic tangent of one half of the LLR values [6], here we argue that range-limited LLR values serve as the best soft message. We discuss the general idea of encoding and protecting these soft message, and demonstrate a particular example of distributed convolutional code. We develop the encoding and the ML decoding algorithms, both of which have a very low complexity that is comparable to that of the conventional digital convolutional code. Simulations demonstrate the performance of our proposed scheme.

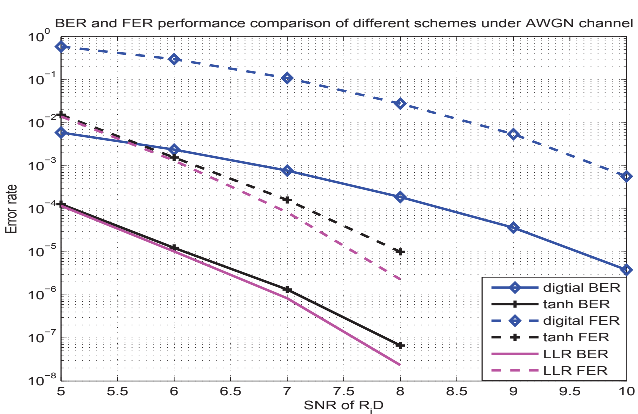

Fig. 2: BER and FER comparison of different schemes under AWGN channel, source uncoded, $(5, 7)$ distributed code, $SNR_{SR_1} = SNR_{SR_2} - 3 = SNR_{R_1D} = SNR_{R_2D}$

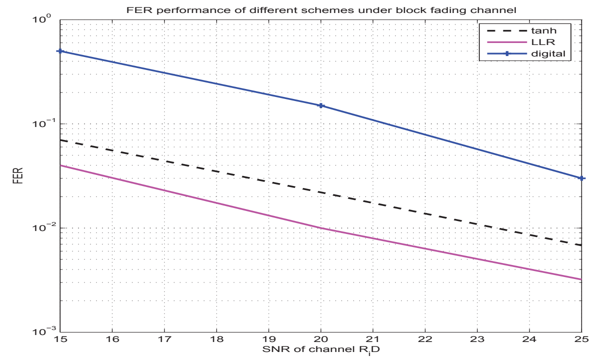

Fig. 3: FER comparison of different schemes under Rayleigh block fading channel, source coded by $(1, 1/(1 + D))$, $(5, 7)$ distributed code, $SNR_{SR_1} = SNR_{SR_2} = SNR_{R_1D} = SNR_{R_2D}$

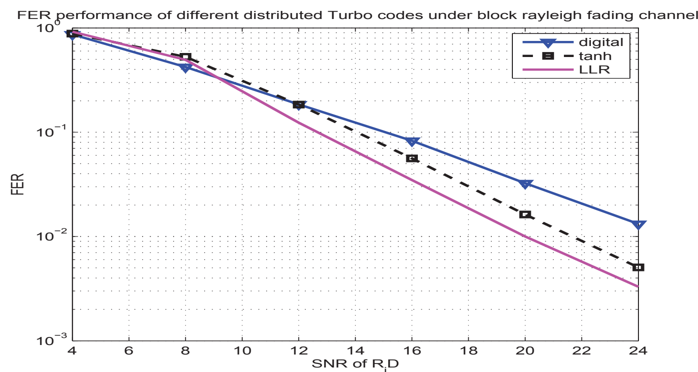

Fig. 4: FER comparison of different distributed code under Rayleigh block fading channel, generated by $(1, 1/1 + D)$, source uncoded, $SNR_{SR_1} = SNR_{SR_2} = SNR_{R_1D} = SNR_{R_2D}$

References

[1] J. N. Laneman, D. Tse and G. W. Wornell, "Cooperative Diversity in Wireless Networks: Efficient Protocols and Outage Behavior," *IEEE Trans. Info. Theory*, vol.50, pp.3062-3080, Dec. 2004.

[2] X. Bao and J. Li, "Efficient Message Relaying for Wireless User Cooperation: Decode-Amplify-Forward (DAF) and Hybrid DAF and Coded-Cooperation," *IEEE Trans. Wireless Commun.*, pp.3975-3984, Oct. 2004.

[3] Y. Li, "Distributed coding for cooperative wireless networks: An overview and recent advances," *IEEE Communications Magazine*,vol. 47, no. 8, pp.71-77, Aug. 2008.

[4] M.C. Valenti and S. Cheng, "Iterative demodulation and decoding of turbo coded M-ary noncoherent orthogonal modulation," *IEEE Journal on Sel. Areas in Comm.*, vol. 23, no. 9, pp. 1738-1747, Sept. 2005.

[5] Y. Li, B. Vucetic, T. F. Wong, and M. Dohler,"Distributed turbo coding with soft information relaying in multihop relay networks," *IEEE J. Selected Areas Commun.*, pp 2040-2050, Oct. 2006.

[6] Y. Li, and B. Vucetic, "Distributed soft coding with a soft input soft output (SISO) relay encoder in parallel relay channels," *arXiv:1201.3140v1*, 2012.

[7] K. S. Gomadam and S. A. Jafar, "Optimal relay functionality for SNR maximization in memoryless relay networks," *IEEE Journal on Sel. Areas in Comm.*, vol. 25, no. 2, pp. 390-400, Feb. 2007.

[8] I. Abou-Faycal, M. Medard, "Optimal Uncoded Regeneration for Binary Antipodal Signaling," *Proc. IEEE ICC*, 2004.