International Journal of Advancements in Electronics and Electrical Engineering – IJAEEE
Volume 3 : Issue 1 *[ISSN 2319 – 7498]*
Publication Date : 09 January 2014

Practical Aspects of Compute and Forward

[William Liu]

Abstract—**In this paper, the recently proposed physical layer network coding (PNC) scheme, compute-and-forward, is revisited. Detailed analysis is performed to reveal the practical problems or challenges of this relaying scheme. A very effective and efficient method is proposed to find the optimal integer coefficients to maximize the achievable rate. To obtain enough independent integer linear equations at the destination nodes to recover the messages, a multi-antenna receiving scheme is proposed. It is also revealed that for a general multi-way relay over Gaussian fading channel, joint maximum likelihood (ML) detection has a much better performance than compute-and-forward relaying scheme. Simulation results are also provided**

Keywords—Compute-and-forward, joint detection, lattice codes, lattice decoding, MIMO, physical layer network coding, relay

I. Introduction

In the past few years, physical layer network coding (PNC) has become a welcoming topic and attracted a large amount of research activity [1]. The aim of the research is to find or develop methods with which the network can achieve higher data rate or capacity. The idea of PNC for wireless network is originated from network coding for wired network, which can trace back to as early as 2000 [2]. The principle of network coding is that intermediate nodes forward functions of received packets rather than individual packets to increase the throughput. In 2006, three research groups [3]-[5] independently proposed the idea of PNC for wireless network. It is shown that, with a relay decoding and forwarding the modulo-two sum (XOR) of the transmitted packets, the throughput of a two-way relay channel can be significantly improved.

In 2009, Nazer and Gastpar [6] extended PNC from two-way relay case to general network framework by introducing a new PNC and network relaying strategy, compute-and-forward, using lattice codes. The principle of compute-and-forward, using lattice codes, is to exploit rather than combat the multiple access interference in a wireless relay network, thus resulted in improved network throughput. In this relaying scheme, the relays, instead of recovering individual messages, attempt to reliably recover and pass an integer linear combination of transmitted messages, referred to as an equation, to the destination. By receiving sufficient number of equations, the destination node is able to recover the individual messages by simply solving these linear equations.

Before the emergence of compute-and-forward, there are

Before the emergence of compute-and-forward, there are three well known PNC schemes or relaying strategies, namely, amplify-and-forward, compress-and-forward, and decode-and-forward. In the amplify-and-forward scheme, the relay simply acts as a repeater. The amplify-and-forward scheme provides a simple "analogue-to-analogue" interface in which the relay scales the incoming signal and transmits it to the receiver [7]. Noise accumulates with each retransmission as no decoding is performed at the relay nodes. In the compress-and-forward scheme, instead of using "analogue-to-analogue" interface, an "analogue-to-digital" interface is used. The relay compresses the received signal and sends the compression index to the receiver [8]. Similar to amplify-and-forward, as no decoding is performed at the relay nodes, noise accumulates as messages traverse the network. In the decode-and-forward, the relay decodes at least part of the transmitted messages. The recovered messages are then un-coded or re-encoded and transmitted to the next relay or the destination [8]. Obviously the relay is ultimately interference-limited as the number of transmit sources increases.

The rest of this paper is organized as follows: Section II presents the principle and system structure of compute-and-forward, and the commonly used decoding method. In section III, the practical problems and challenges of compute-and-forward are revealed and unique solutions are proposed for each of the problems. Simulation verifications are provided in section IV, and conclusions are drawn in section V.

II. Compute-and-Forward Relaying Scheme

A. *Principle and System Structure of Compute-and-Forward*

So far in compute-and-forward relaying scheme, linear codes are employed by the requirement that an integer linear combination of codewords must still be a codeword. Since lattice codes offer many desired properties, they have naturally been used in compute-and-forward exclusively and become part of the architecture of this relaying scheme.

For a general relay network, assuming many source nodes and relay nodes. The general computation model of compute-and-forward can be abstracted below in Figure 1.

William Liu is with Department of Electrical and Electronic Engineering at Imperial College London, UK.

This work was sponsored by the UK Engineering and Physical Sciences Research Council (EPSRC) for the Doctor Training Scheme at Department of Electrical and Electronic Engineering, Imperial College London.

SEEK DIGITAL LIBRARY

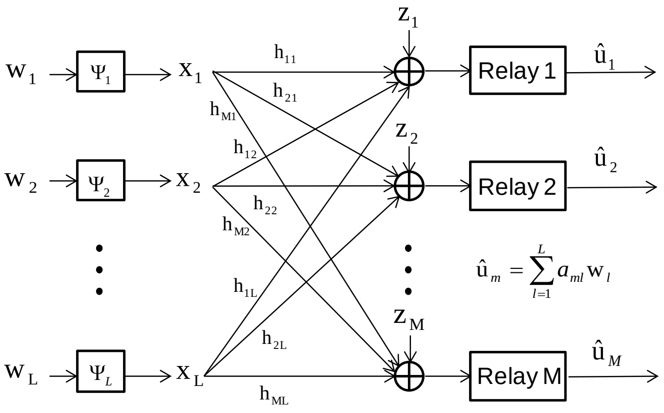

Figure 1. General model of compute-and-forward with *L* source nodes and *M* relay nodes.

In the encoding process, messages $w_1, w_2, \cdots, w_L \in F_q^k$ from the *L* source nodes are encoded to *L* n-dimensional complex-valued codewords $x_1, x_2, \cdots, x_L \in C^n$, which are then transmitted over channel with complex-valued coefficients h_{ml} . The received signal at each relay node is a linear combination of the transmitted signals from all the source nodes plus additive noise which can be expressed as:

$$y_m = \sum_{l=1}^{L} h_{ml} x_l + z \tag{1}$$

The channel coefficients between the source nodes and the relay nodes are assumed perfectly known at the relay nodes. The n-dimensional codewords are subject to average power constraint of $\frac{1}{n} E\|x_l\|^2 \le P$, and $z \sim CN(0, \sigma^2 I_n)$ is a circularly-symmetric jointly-Gaussian complex random vector where I_n is the $n \times n$ identity matrix.

Now the target of the decoder at each relay is to decode an integer linear combination of the codeword vectors transmitted from all the source nodes:

$$\hat{u}_m = \sum_{l=1}^{L} a_{ml} w_l \tag{2}$$

or

$$\hat{u}_m = \left[\sum_{l=1}^{L} a_{ml} w_l \right] \bmod \Lambda \tag{3}$$

where $a_{ml} \in Z + jZ$ are the integer coefficients to be determined in the decoding process to maximize the achievable data rate, which is the key design parameter of compute-and-forward scheme and will be discussed in the next subsection. Note that in the original description of compute-and-forward [6], modulo operation by the coarse lattice Λ is used in (3). This is optional when nested lattice codes are employed and the modulo lattice decoding method [9] is used. In general, (2) can simply be used in the decoding.

Also note that because of the one to one mapping between a message vector and a codeword vector, decoding an integer linear combination of codeword vectors is equivalent to decoding an integer linear combination of message vectors.

After decoding at each relay node, the decoded linear equations are forwarded to the destination node, where, after receiving enough number of linear independent equations, the transmitted messages can be recovered easily.

B. *Decoding of Integer Linear Equation*

As the design principle of compute-and-forward is based on the algebraic property that an integer linear combination of codewords is still a codeword, the goal of the decoder at a relay node is to reliably recover an integer linear combination of the transmitted codewords. However, in reality, the channel coefficients normally have real or real complex values instead of integer or integer complex values, the linear combination of transmitted codewords received at the relay receiver is no longer an integer linear combination. Therefore the task of the decoder is to find an integer linear combination which is as co-linear as possible to the received signal without considering the noise. The commonly used decoding method of compute-and-forward can be summarized below in Figure 2.

The decoding process includes scaling the received signal with the optimal scaling factor, and then quantizing the scaled signal to the nearest lattice point. The purpose of the scaling by a factor α is to make the scaled signal $\tilde{y}_m$ as close to an integer linear combination of the codewords as possible. The scaled signal can be expressed as:

$$\begin{aligned} \tilde{y}_m = \alpha y_m &= \alpha \left(\sum_{l=1}^{L} h_{ml} x_l + z \right) \\ &= \sum_{l=1}^{L} a_{ml} x_l + \sum_{l=1}^{L} (\alpha h_{ml} - a_{ml}) x_l + \alpha z \\ &= \sum_{l=1}^{L} a_{ml} x_l + n_{eff} \end{aligned} \tag{4}$$

$$n_{eff} = \sum_{l=1}^{L} (\alpha h_{ml} - a_{ml}) x_l + \alpha z \tag{5}$$

From (4), it can be seen that the scaled signal consists of three components. The first component is a desired integer linear combination of the codewords; the second component is

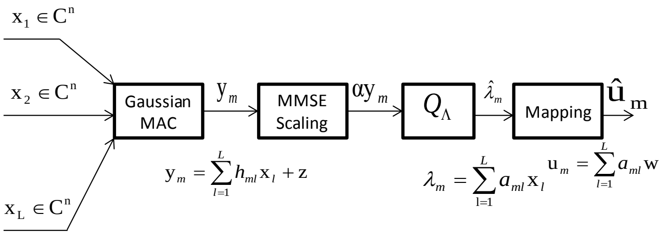

Figure 2. Decoding structure of compute-and-forward.

SEEK DIGITAL LIBRARY

the noise caused by channel mismatch; and the last component is related to the additive white Gaussian noise. Thus a main task of the compute-and-forward decoder is to select the scaling factor α and the coefficient vector $a_m = [a_{m1}, a_{m2}, \cdots, a_{mL}]$ to minimize the average power of the effective noise n_{eff} expressed by (5).

1) Achievable rate

It has been proved in [6] that for complex-valued wireless AWGN network with channel coefficient vector $\mathrm{h}_m = [h_{m1}, h_{m2}, \cdots h_{mL}] \in C^L$ and equation coefficient vector $\mathrm{a}_m = [a_{m1}, a_{m2}, \cdots, a_{mL}] \in \{\mathrm{Z} + j\mathrm{Z}\}^L$, the following computation rate is achievable at the relay receiver:

$$\Re(\mathrm{h}_m, \mathrm{a}_m) = \max_{\alpha \in C} \log\left(\frac{SNR}{|\alpha|^2 + SNR\|\alpha \mathrm{h}_m - \mathrm{a}_m\|^2}\right) \tag{6}$$

where $SNR = P/N$, N is the power of the AWGN noise z. It has also been proved that this computation rate can be uniquely maximized by further choosing α to be the MMSE coefficient which is given by:

$$\alpha_{MMSE} = \frac{\mathrm{a}_m \mathrm{h}_m^H SNR}{1 + SNR\|\mathrm{h}_m\|^2} \tag{7}$$

By using the above MMSE scaling factor α_{MMSE}, the resulted achievable computation rate can be written as:

$$\Re(\mathrm{h}_m, \mathrm{a}_m) = \log\left(\left(\|\mathrm{a}_m\|^2 - \frac{SNR|\mathrm{a}_m \mathrm{h}_m^H|^2}{1 + SNR\|\mathrm{h}_m\|^2}\right)^{-1}\right) \tag{8}$$

2) Parameter selection

For the scaling factor α, from (4), it can be seen that using a larger value may allow better approximations of an integer linear combination of codewords and reduces the noise due to the mismatch in the approximation. However, as it can be observed in the effective noise term, a larger scaling factor also results in amplification of the additive noise. This trade-off between integer-approximation and noise amplification is known as the Diophantine trade-off. In fact, the design criterion of the compute-and-forward receiver is to select the scaling factor α and the equation coefficient vector a_m so as to maximize the computation rate. This is also equivalent to minimizing the power of the effective noise [10]:

$$N_{eff}(\alpha, \mathrm{a}_m) = \sigma^2|\alpha|^2 + P\|\alpha \mathrm{h}_m - \mathrm{a}_m\|^2 \tag{9}$$

$$(\alpha_{opt}, \mathrm{a}_{opt}) = \arg \min_{\alpha \in C, \mathrm{a}_m \neq 0} N_{eff}(\alpha, \mathrm{a}_m) \tag{10}$$

Given the relationship between the scaling factor and the equation coefficient vector by (7), (9) can be written as:

$$N_{eff}(\mathrm{a}_m) = \mathrm{a}_m \mathrm{M} \mathrm{a}_m^H \tag{11}$$

$$\mathrm{M} = SNR\left(I - \frac{SNR}{1 + SNR\|\mathrm{h}_m\|^2}\mathrm{h}_m^H \mathrm{h}_m\right) \tag{12}$$

Because the matrix M is Hermitian and positive definite, it has a Cholesky decomposition $\mathrm{M} = \mathrm{L}\mathrm{L}^H$, where L is a lower triangular matrix. Thus (11) becomes:

$$N_{eff}(\mathrm{a}_m) = \mathrm{a}_m \mathrm{M} \mathrm{a}_m^H = \|\mathrm{a}_m \mathrm{L}\|^2 \tag{13}$$

And the optimal equation coefficient vector is given by

$$\mathrm{a}_{opt} = \arg \min_{\mathrm{a}_m \neq 0} \|\mathrm{a}_m \mathrm{L}\| \tag{14}$$

Thus finding the optimal parameters α and a_m is in fact a shortest vector problem (SVP).

III. Practical Problems or Challenges, and Solutions of Compute-and-Forward

Currently the compute-and-forward PNC scheme has the following practical problems or challenges that need to be solved.

- Developing efficient parameter selection method
- Obtaining enough independent integer linear equations at the destination node.
- Achieving the maximum rate

Below in this section, we describe each problem or challenge and propose our methods of solutions.

A. *Developing Efficient Parameter Selection Method*

1) Drawbacks of the existing methods

As described in section II.B.2) above, the integer coefficients optimization is to solve a SVP, which is known to be NP hard. Although a few suboptimal algorithms exist, such as the LLL algorithm [11], which make the searching process less complex, they have two drawbacks. Firstly, due to the large search space of the problem, the computation complexity is still very high. Secondly, because these algorithms are suboptimal, the solution is likely not to be the best, leading to a lower data rate. In the following section, we propose an efficient method to solve this problem.

2) New efficient method for parameter selection

The motivation of the new method is that the purpose of the parameter selection is equivalent to finding an optimal scaling factor that makes the decoded integer linear equation as co-linear as possible to the received signal without considering the additive noise. Therefore in this newly proposed method, instead of searching the optimal integer coefficients directly, we search the optimal scaling factor first, then convert it to the integer coefficients. In this way, the searching process is much faster since for a relay of L users over a complex-valued channel, the process of direct searching

SEEK DIGITAL LIBRARY

has to search $2L$ integer coefficients, while in our new method, only one parameter, the scaling factor, has to be searched. This new parameterr selection algorithm is described below.

Algorithm Parameter Selection

Define function $f(x)=$closest integer complex number to $x \in C$. In case of a tie, choose the number with the largest absolute value. For vector $\mathrm{x}=[x_1, x_2, \cdots x_n] \in C^n$, define $f(\mathrm{x})=[f(x_1), f(x_2), \cdots f(x_n)]$.

Note: the searching range of the scaling factor α will be discussed below after the algorithm description.

```
Step1: Initialization
    $\alpha_{START}=\alpha_{START}$, $\alpha_{END}=\alpha_{END}$, $\alpha_{STEP}=1$,
    $\mathrm{M}_{TEMP}=-\infty$,
    $\alpha=\alpha_{START}$
Step2: Searching Loop
    $\mathrm{a}_m = f(\alpha \cdot \mathrm{h}_m)$
    $\mathrm{M}=\frac{SNR}{|\alpha|^2+SNR\|\alpha \mathrm{h}_\mathrm{m}-\mathrm{a}_m\|^2}$
    If $\mathrm{M}>\mathrm{M}_{TEMP}$
        $\mathrm{M}_{TEMP}=\mathrm{M}$, $\mathrm{a}_{OPT}=\mathrm{a}_\mathrm{m}$, $\alpha_{OPT}=\alpha$
        $\alpha=\alpha+1$
    If $\alpha \le \alpha_{END}$
        GOTO Step2
    Otherwise, GOTO the next step
Step3: MMSE scaling
    $\alpha_{MMSE}=\frac{\mathrm{a}_{OPT}\mathrm{h}_\mathrm{m}^\mathrm{H} SNR}{1+SNR\|\mathrm{h}_\mathrm{m}\|^2}$
    $\mathrm{a}_m = f(\alpha_{MMSE} \cdot \mathrm{h}_m)$
```

In the above parameter selection algorithm, for the searching range of the scaling factor α, only integer value needs to be considered. The searching start value is selected to be $\alpha_{START}=1$. To determine the searching end value, the following two methods can be used.

Method-1: This is an intuitive method. From (5), we know that when the scaling factor is larger than 1, it amplifies the noise. Thus to achieve the maximum rate, the scaling factor cannot be too big, especially when SNR is low. Therefore, normally a searching end value of $\alpha_{END}=1000$ is more than enough.

Method-2: In “unpublished” [12], for the direct integer coefficient searching method, the authors have proved that the searching range is necessary only in the range defined by:

$$\|\mathrm{a}_m\|^2 < 1+SNR\|\mathrm{h}_m\|^2) \tag{15}$$

Based on (16), the searching end value can be selected as:

$$\alpha_{END}=Ceiling\left(\max_{1\le l\le L}\left(\frac{\sqrt{1+SNR\|\mathrm{h}_m\|^2}}{h_{ml}}\right)\right) \tag{16}$$

B. *Obtaining Enough Independent Integer Linear Equations at the Destination Node*

1) Problem

As discussed in section II.A, for compute-and-forward PNC scheme to work, the destination node has to receive enough independent integer linear equations from the relay nodes. In real applications, due to the nature of the wireless channel, it is highly likely that some equations decoded at the relay nodes will be linearly dependent. In this case, the transmitted messages cannot be recovered at the destination node. Below we propose a multi-antenna receiving scheme to solve this problem.

2) Multi-antenna receiving scheme at relay nodes

In this scheme, instead of using single receiving antenna, multi-antenna receiver is employed at each relay node. In this way, each relay node can decode several equations instead of only one. Since the multiple antennas belong to the same relay node, the relay receiver can also determine whether the equations are linear independent, thus can avoid blindly sending linear dependent equations to the destination node. In real applications, the number of source nodes that communicate with a relay node at the same time will not be large, e.g., two will be typical and eight will be highly unlikely. Therefore four receive antennas will normally be large enough. One additional advantage of this scheme is that a channel condition aware intelligent PNC scheme can be adopted, where when channel condition suits, compute-and-forward decoding will be used, otherwise normal MIMO decoding will be used to decode the messages directly. We believe the intelligent PNC scheme can further boost the data rate in real applications, which will not be discussed in this paper.

C. *Achieving the Maximum Rate*

In this section, we list a number of scenarios and describe how to achieve the best rate.

1) Two-way relay over AWGN channel

In [13], the achievable rates of various relaying schemes are derived, among which the rate of compute-and-forward is the best and can be written as:

$$R_{CF}=\log(0.5+SNR) \tag{17}$$

2) Two-way relay over Gaussian fading channel

The achievable rate of compute-and-forward is given in (8) if no extra processing is performed at the transmitters or receivers. However, for two-way relay, it has been demonstrated in [13] that precoding can be applied so that the

SEEK DIGITAL LIBRARY

maximum rate of compute-and-forward expressed in (17) can be achieved.

3) General multi-way relay with AWGN channel

In this case, all the channel coefficients are unities thus the elements of the equation coefficient vectors are also unities. Therefore compute-and-forward relaying scheme cannot be applied because not enough independent equations can be provided to the destination nodes from the relay nodes. In this scenario, other relaying schemes such as routing have to be used.

4) General multi-way relay with Gaussian fading channel

This is the general case of a relay network where the algorithm of compute-and-forward relaying scheme is derived. The achievable rate is expressed in (8). Since the instantaneous computation rate depends on the instantaneous channel coefficients, it is difficult to evaluate the overall performance of compute-and-forward in this scenario simply from the rate equation. In section IV.B, simulation will be used to demonstrate that the performance of compute-and-forward with the commonly used decoding method described in [6] is much worse than the performance of joint detection using ML decoding algorithm. The joint ML detection can be expressed as the following equation:

$$[\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_L] = \arg\min_{x_l \in A} \left\| y_m - \sum_{l=1}^{L} h_{ml} x_l \right\| \tag{18}$$

where A is the set of codewords of each source node.

IV. Simulation Verification

A. Performance of Parameter Selection Methods

To verify the effectiveness of the new parameter selection method, many simulations have been carried out, and the new method shows to be much more efficient than the direct integer coefficients searching method. This is because over complex-valued channels, if the number of searching points per channel element is Ω, the complexity of the direct searching method is in the order of Ω^{2L}, while the complexity of the new method is in the order of Ω_α, where Ω_α is the number of searching point for the scaling factor α.

To verify the accuracy of the new parameter selection method, the following simulation is carried out for a relay of two source nodes over real-valued channels. We assume the channel gain is 1.0 for the first source node, and the channel gain *h* for the other source node varies from 0 to 1.0. Both the proposed method and the direct searching method are tested, and the achievable rates of compute-and-forward with the two results are plotted below in Figure 3. with SNR=30 dB. It is shown that for smaller values of the channel gain, the result of the new method is much better. For larger values of the channel gain, the performances of the two methods are exactly the same. In the same figure, for comparison purposes, the achievable rate of decode-and-forward and the AWGN channel upper bound capacity are also plotted.

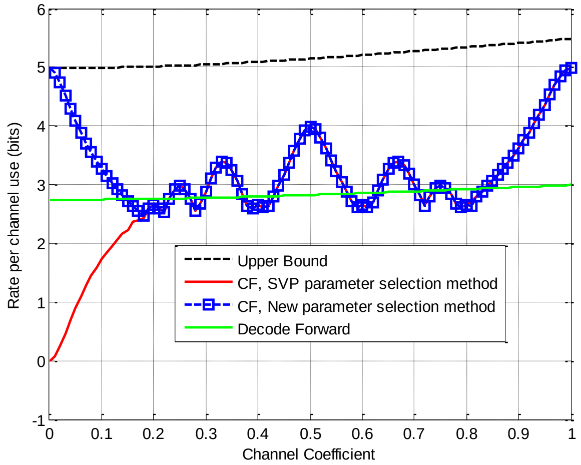

Figure 3. Performance comparison of our newly proposed parameter selection method and the direct SVP integer coefficient searching method, SNR=30 dB.

B. Performance of Compute-and-Forward Decoding Schemes

For two-way relay, compute-and-forward has the highest achievable rate than any other relaying schemes [13]. For a general multi-way relay over Gaussian fading channel, the performance of compute-and-forward relaying scheme cannot simply be evaluated theoretically. Therefore we simulate a simple case to see how it performs. In the simulation, a relay of two source nodes over Gaussian fading channel is used. Z^2 lattice codes of 2 points per-dimension and 4 points per-dimension are employed in the source nodes, which are equivalent to 4-QAM and 16-QAM constellations. For comparison purposes, joint ML detection is also used to decode the codewords directly. Simulation results are plotted below in Figure 4. and Figure 5.

It can be seen that the performance of compute-and-forward relaying scheme is much worse than that of joint ML detection at the relay node.

V. Conclusions

Unique solutions are proposed for the practical problems or challenges of compute-and-forward relaying scheme. The newly proposed parameter selection method is much more efficient than the commonly used direct integer coefficient searching method, while providing better searching results. The newly proposed multi-antenna receiving scheme at the relay nodes makes sure enough number of independent integer linear equations be received at the destination node. For a general multi-way relay over Gaussian fading channel, joint ML detection can achieve much better performance than compute-and-forward relaying scheme.

SEEK DIGITAL LIBRARY

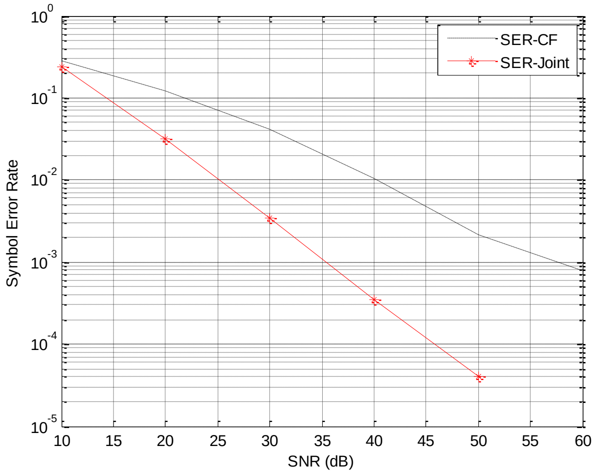

Figure 4. Performance comparison of compute-and-forward and joint ML detection, two source nodes, 4-QAM constellation, Gaussian fading channel.

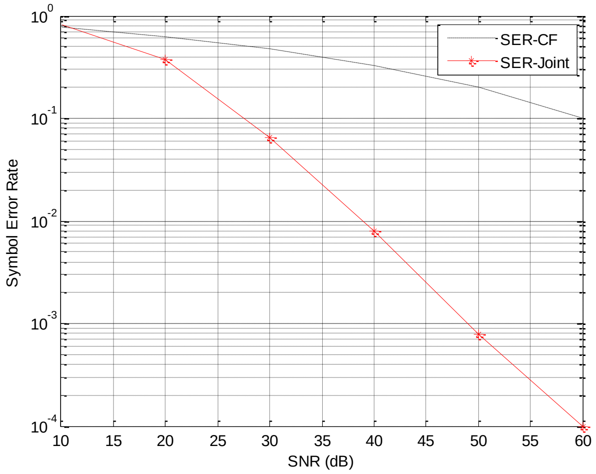

Figure 5. Performance comparison of compute-and-forward and joint ML detection, two source nodes, 16-QAM constellation, Gaussian fading channel.

References

[1] B. Nazer and M. Gastpar, “Reliable physical layer network coding,” Proceedings of the IEEE, vol. 99, no. 3, pp. 438–460, 2011.

[2] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, “Network information flow,” IEEE Trans. Inf. Theory, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.

[3] S. Zhang, S. Liew, and P. Lam, “Hot topic: Physical-layer network coding,” in Proc. ACM Int. Conf. Mobile Comput. Netw., Los Angeles, CA, pp. 358–365, Sep. 2006.

[4] P. Popovski and H. Yomo, “The anti-packets can increase the achievable throughput of a wireless multi-hop network,” in Proc. IEEE Int. Conf. Commun., Istanbul, Turkey, Jun. 2006, pp. 3885–3890.

[5] B. Nazer and M. Gastpar, “Computing over multiple-access channels with connections to wireless network coding,” in Proc. IEEE Int. Symp. Inf. Theory, Seattle, WA, Jul. 2006, pp. 1354–1358.

[6] B. Nazer and M. Gastpar, “Compute-and-forward: Harnessing interference through structured codes,” IEEE Trans. Inf. Theory, vol. 57, no. 10, pp. 6463–6486, October 2011.

[7] S. Borade, L. Zheng, and R. Gallager, “Amplify-and-forward in wireless relay networks: Rate, diversity, and network size,” IEEE Trans. Inf. Theory, vol. 53, pp. 3302–3318, Oct. 2007.

[8] G. Kramer, M. Gastpar, and P. Gupta, “Cooperative strategies and capacity theorems for relay networks,” IEEE Trans. Inf. Theory, vol. 51, pp. 3037–3063, Sep. 2005.

[9] U. Erez and R. Zamir, “Achieving 1/2 log(1 +SNR) on the AWGN channel with lattice encoding and decoding,” IEEE Trans. Inf. Theory, vol. 50, no. 22, pp. 2293–2314, Oct. 2004.

[10] F. Chen, D. Silva, and F. R. Kschischang, “Design Criteria for Lattice Network Coding,” 45th Annual Conference on Information Sciences and Systems (CISS), 2011.

[11] Guillaume Hanrot, Xavier Pujol and Damien Stehlé, “Algorithms for the Shortest and Closest Lattice Vector Problems,” Invited contribution for IWCC'11, © Springer-Verlag, 2011.

[12] Mohsen Hejazi, Masoumeh Nasiri-Kenari “Simplified compute-and-forward and its performance analysis,” submitted for publication, 2013, http://arxiv.org/abs/1304.4428

[13] W. Liu, “Two-way relay the most suited for compute-and-forward,” unpublished.

About Author (s):

[William Liu is a research student in the Department of Electrical and Electronic Engineering at Imperial College London. His research interests include digital signal processing, information theory, channel coding, and MIMO detection in wireless communications.]

SEEK DIGITAL LIBRARY