An Interference-Resilient Relay Beamforming Scheme Inspired by Back-Propagation Algorithm

Rui Wang Yi Jiang
Shanghai Institute for Advanced Communication and Data Science
Dept. of Communication Science and Engineering,
Fudan University, Shanghai, China
Email: 18110720020@fudan.edu.cn, yijiang@fudan.edu.cn

***Abstract*—A relay node can be used to improve the distance and service quality of a communication link, but not when it is being interfered. In this paper, we consider a relay network consisting of one source, one destination, and multiple relay nodes, and draw analogy between the relay network and a three-layer artificial neural network (ANN). Inspired by the classic back-propagation (BP) algorithm for the ANN, we develop an interference-resilient algorithm that can optimize the beamforming-and-forwarding weights of the relay nodes so that the interferences will be canceled at the destination. The proposed algorithm requires no channel state information (CSI), no data exchanges between the relay nodes; it requires that the source transmit training sequences in the forward channel (source-to-relays) and the destination transmit error sequences in the backward channel (destination-to-relays). The simulation results verify the effectiveness of the proposed scheme in the interference environment.**

***Index Terms*—relay communication; backward propagation; distributed beamforming; interference suppression**

I. Introduction

Relay nodes can be deployed to improve the communication distance and the end-to-end link quality, especially when they can cooperate [1]. Several cooperative relay strategies have been proposed, including the encode-and-forward strategy, the compress-and-forward strategy [2], and the simpler amplify-and-forward strategy [3] [4] [5].

This paper studies a relay scheme using the amplify-and-forward strategy for interference suppression. While the majority of the existing work on relay communications assume no interferences (see, e.g., [3] [4]), interferences often exist in the real scenario. For those works that addresses the interference issue, the global channel state information (CSI) of the interferences is usually assumed for relay cooperation [6] [7]. But this assumption can be impractical, since a) the interferences are uncoordinated with our nodes, and b) it is difficult for the interferred relay nodes to exchange information between themselves.

For the interference-free relay networks, instead of the exact CSI, the second-order statistic of the CSI can be used for distributed beamforming [8] [9]. But despite the research on developing the robust techniques [7] [8] [10], the algorithm performance degrades as the uncertainty of the CSI increases [9].

In this paper, we study a distributed beamforming scheme for a relay network under interferences, assuming no explicit CSI but the training sequences transmitted from the source to the relay nodes. Moreover, we assume that the relay transmission is subject to the modulation of the nonlinear PA, which limits the *instantaneous amplitude* of the transmit signal. This assumption is more practical than the prevalent assumption of *average power* constraint, including the total-power [5] [8] [9] and the per-relay power constraints [8] [9]. These practical assumptions apparently render the relay problem even more challenging.

We tackle this problem via exploiting the multi-aspect similarity between a relay network and an artificial neural network (ANN): the nodes are analogous to the neurons in the ANN; the nonlinearity of a PA is analogous to the nonlinear activation function of a neuron; the beamforming weights are analogous to the weight coefficients of the ANN.

Inspired by the classic back-propagation (BP) algorithm for training an ANN, we develop a distributed relay beamforming scheme for optimizing a relay network. With the source node transmitting periodically a training sequence to the relays, the relay nodes apply beamforming to their received samples and forward them to the destination. Then the destination *back-propagates* an error sequence to the relays so that they can optimize their beamforming-and-forwarding weights to minimize the mean square error (MSE) of the destination's output. This scheme requires no data exchanges between the relay nodes.

About using a reverse channel to control the distributed nodes, the authors of [11] also use a (low rate) reverse control channel to achieve distributed transmit beamforming. But the problem considered here is different and more involved other than that similarity, as we address a relay communication problem and take into account the interferences and the nonlinearity of the PAs.

The rest of the paper is structured as follows. Section II introduces the system model of a relay network and draw analogy between a relay network and an ANN. Section III introduces our BP-inspired distributed relay beamforming algorithm and the frame structure designed for supporting the algorithm. Section IV presents simulation results that verify

The work was supported by National Natural Science Foundation of China Grant No. 61771005.

the effectiveness of the proposed scheme for the relay networks with or without interferences. Section V gives the conclusion.

II. System Model

A. A Relay Network under Interferences

We consider a relay communication network as illustrated in Fig.1, which consists of a single-antenna source, an M_d-antenna destination, and N relay nodes each with M_r receiving antennas and one transmitting antenna.

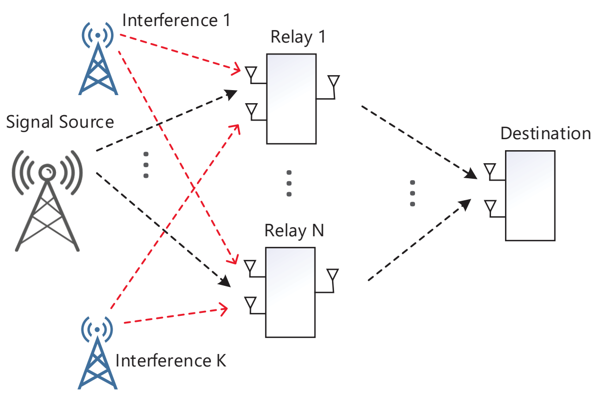

Fig. 1: A relay network under interferences.

The relay nodes are subject to K interferences, denoted by $\mathbf{z}(i) \in \mathbb{C}^K$. With the signals $s(i)$ being transmitted from the source, the relay nodes receive

$$\mathbf{y}_n(i) = \mathbf{f}_n s(i) + \mathbf{G}_n \mathbf{z}(i) + \boldsymbol{\eta}_n(i), \text{ for } n = 1, \ldots, N, \quad (1)$$

where $\mathbf{f}_n \in \mathbb{C}^{M_r \times 1}$ is the channel between the source and the n-th relay, $\mathbf{G}_n \in \mathbb{C}^{M_r \times K}$ is the channel between the K interferences and the n-th relay, and $\boldsymbol{\eta}_n \sim N(0, \sigma_\eta^2 \mathbf{I})$ is the channel noise. There is no direct link between the source and the destination.

Let the n-th relay node apply beamforming weight $\mathbf{v}_n \in \mathbb{C}^{M_r \times 1}$ to obtain

$$b_n \triangleq \mathbf{v}_n^H \mathbf{y}_n, \quad (2)$$

where $(\cdot)^H$ is the conjugate transpose. Here and in the remainder of this paper, we omit the time index i for notational simplicity.

When the relay node transmits b_n, it will be modulated by the PA, whose output amplitude as a (nonlinear) function of the input amplitude is [12, Chapter 3.5]

$$\sigma(x) = \frac{x}{(1 + x^{2p})^{\frac{1}{2p}}}. \quad (3)$$

The parameter p affects the degree of nonlinearity of the PA, as shown in Fig.2. The output of the PA with b_n being the input is

$$a_n = \sigma(|b_n|) e^{j\theta_n} \quad (4)$$

where θ_n is the phase of b_n. With this PA model, the transmit power per relay node is constrained to be no greater than one.

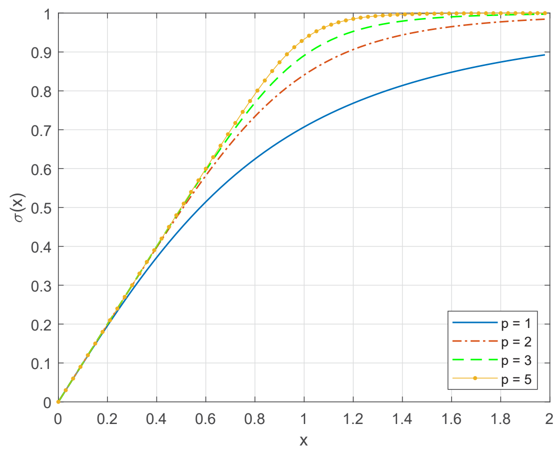

Fig. 2: The amplitude of the PA output as a nonlinear function $\sigma(x)$ of the amplitude of the input.

To avoid self-interference, the relay nodes receive and transmit on two different frequencies. The received signal of the destination node is

$$\mathbf{r} = \mathbf{H}\mathbf{a} + \boldsymbol{\xi}, \quad (5)$$

where $\mathbf{H} \in \mathbb{C}^{M_d \times N}$ is the channel from the N relays to the destination,

$$\mathbf{a} \triangleq \begin{bmatrix} \sigma(|b_1|)e^{j\theta_1} \\ \vdots \\ \sigma(|b_N|)e^{j\theta_N} \end{bmatrix} \quad (6)$$

contains the transmitted signals of all the relay nodes, and $\boldsymbol{\xi} \sim \mathcal{C}N(0, \sigma_\xi^2 \mathbf{I})$ is the white Gaussian noise. With the beamforming weight $\mathbf{w} \in \mathbb{C}^{M_d}$, the destination obtains the output $\hat{s} = \mathbf{w}^H \mathbf{r}$.

Assuming that the source transmits training sequences to the destination through the relay nodes, this paper focuses on minimizing the MSE with respect to $\mathbf{w}$ and $\mathbf{v}_n$'s, i.e.,

$$\min_{\mathbf{v}_1, \ldots, \mathbf{v}_N, \mathbf{w}} \mathbb{E} \left| \mathbf{w}^H \mathbf{r} - s \right|^2, \quad (7)$$

where s is the known training signal. To show the dependency of the MSE on $\mathbf{v}_n$'s explicitly, we expand (7) to be

$$\min_{\mathbf{v}_1, \ldots, \mathbf{v}_N, \mathbf{w}} \mathbb{E} \left| \mathbf{w}^H \left\{ \mathbf{H} \begin{bmatrix} \sigma(|\mathbf{v}_1^H \mathbf{y}_1|)e^{j\theta_1} \\ \vdots \\ \sigma(|\mathbf{v}_N^H \mathbf{y}_N|)e^{j\theta_N} \end{bmatrix} + \boldsymbol{\xi} \right\} - s \right|^2. \quad (8)$$

Note that to solve (8) will achieve interference suppression automatically, since minimizing the MSE amounts to maximizing the output signal-to-interference-plus-noise ratio (SINR) according to their relationship [13]

$$\text{SINR} = \frac{1}{\text{MSE}} - 1. \quad (9)$$

To solve (8), however, appears challenging, especially when

i) the CSI is not explicitly available,
ii) the nodes, including the relays and the destination, need to

optimize their own beamforming weights in a distributed manner, and

iii) there are no communications between the relay nodes.

In the next, we show how to solve (8) with the inspiration from the neural network theory, which meets all the three requirements in the above.

B. The Analogy between Relay Network and ANN

Fig.3 shows a three-layer ANN, which is similar to a relay network in multiple aspects.

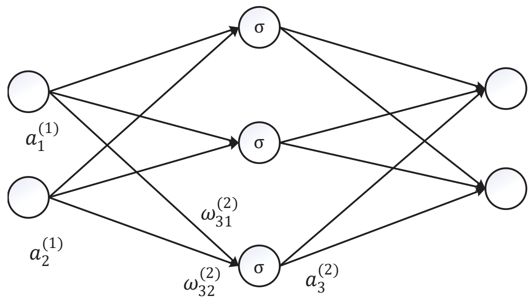

Fig. 3: The topology of a three-layer ANN.

In Fig.3, $a_n^{(l)}$ represents the output of the n-th neuron of the l-th layer, which is just like the transmitted signal a_n [cf. (4)] from the PA of nth relay node. The input into a neuron in the l-th layer is a linear combination (with weights $\omega_{jk}^{(l)}$) of the outputs of the neurons in the $(l-1)$th layer, which is analogous to the beamforming operation [cf. (2)]. Just like the relay nodes having nonlinear PA, the neurons in an ANN have a nonlinear activation function σ, such as a Rectified Linear Unit (ReLU) or a Sigmoid function.

The striking similarity between the relay network and an ANN prompts us to consider optimizing the beamforming weights of the relays and the destination by using techniques from the neural network theory. We let the source transmit a training sequence periodically to the relay nodes, just like feeding batches of the training samples into the ANN. We then modify the BP algorithm [14] to optimize the beamforming weights of the relays and the destination, using a reverse channel to broadcast an *error* sequence from the destination to the relays. It will be detailed in the next section.

Note that in the ANN as shown in Fig. 3, a neuron is only connected to the ones in the adjacent layers. Consequently, there is no direct exchanges of information between the neurons in the same layer. Similarly, our proposed algorithm will update the beamforming-and-forwarding weights of the relay nodes using only the received signal from the source and the feedback from the destination, with no communications between the peer relay nodes. We will revisit this important fact later in the next section.

III. The Distributed Relay Beamforming Scheme

We develop in the next a distributed relay beamforming algorithm to solve (8), which can be regarded as a modified BP algorithm.

A. A Modified BP Algorithm

Denote $J \triangleq |\hat{s}-s|^2$, where $\hat{s} = \mathbf{w}^H\mathbf{r}$ is the beamforming output of the destination node. Then the objective function in (8) is $\mathbb{E}[J]$, and

$$\frac{\partial J}{\partial \hat{s}^*} = \hat{s} - s. \tag{10}$$

The derivative with respect to the output the n-th relay is

$$\begin{aligned}\frac{\partial J}{\partial a_n^*} &= \frac{\partial|\mathbf{w}^H(\mathbf{H}\mathbf{a}+\boldsymbol{\xi})-s|^2}{\partial a_n^*}\\ &= \mathbf{h}_n^H\mathbf{w}\left(\hat{s}-s\right),\ n=1,2,...,N,\end{aligned} \tag{11}$$

where $(\cdot)^*$ denotes complex conjugation and $\mathbf{h}_n \in \mathbb{C}^{M_d}$ is the n-th column of the channel matrix $\mathbf{H}$, i.e., the channel from the n-th relay to the destination.

Regarding the PA function (3), the derivatives of its output with respect to its input in the complex domain are

$$\begin{aligned}\frac{\partial a_n}{\partial b_n^*} &= \frac{\partial\sigma(|b_n|)e^{j\theta_n}}{\partial b_n^*}\\ &= \frac{\partial\sigma(|b_n|)}{\partial|b_n|}\frac{\partial|b_n|}{\partial b_n^*}e^{j\theta_n} + j\sigma(|b_n|)e^{j\theta_n}\frac{\partial\theta_n}{\partial b_n^*}\\ &= -\frac{|a_n|}{2|b_n|}\frac{|b_n|^{2p}}{1+|b_n|^{2p}}e^{j2\theta_n},\end{aligned} \tag{12}$$

and

$$\begin{aligned}\frac{\partial a_n^*}{\partial b_n^*} &= \frac{\partial\sigma(|b_n|)e^{-j\theta_n}}{\partial b_n^*}\\ &= \frac{\partial\sigma(|b_n|)}{\partial|b_n|}\frac{\partial|b_n|}{\partial b_n^*}e^{-j\theta_n} - j\sigma(|b_n|)e^{-j\theta_n}\frac{\partial\theta_n}{\partial b_n^*}\\ &= \frac{|a_n|}{2|b_n|}\left(2-\frac{|b_n|^{2p}}{1+|b_n|^{2p}}\right),\end{aligned} \tag{13}$$

where we have used the relationships that

$$\frac{\partial|b_n|}{\partial b_n^*} = \frac{b_n}{2|b_n|},\quad \frac{\partial\theta_n}{\partial b_n^*} = \frac{j}{2b_n^*}, \tag{14}$$

and

$$\frac{\partial\sigma(|b_n|)}{\partial|b_n|} = \frac{1-|b_n|^{2p}(1+|b_n|^{2p})^{-1}}{(1+|b_n|^{2p})^{\frac{1}{2p}}}. \tag{15}$$

Using (11)-(13) and invoking the chain rule, we obtain

$$\begin{aligned}\frac{\partial J}{\partial b_n^*} &= \frac{\partial J}{\partial a_n^*}\frac{\partial a_n^*}{\partial b_n^*} + \frac{\partial J}{\partial a_n}\frac{\partial a_n}{\partial b_n^*}\\ &= \frac{|a_n|}{2|b_n|}\left[\left(\frac{\partial J}{\partial a_n^*}\right)\left(2-\frac{|b_n|^{2p}}{1+|b_n|^{2p}}\right)\right.\\ &\qquad\left.-\left(\frac{\partial J}{\partial a_n}\right)\frac{|b_n|^{2p}}{1+|b_n|^{2p}}e^{j2\theta_n}\right]\\ &= \frac{|a_n|}{2|b_n|}\left[\left(\mathbf{h}_n^H\mathbf{w}\left(\hat{s}-s\right)\right)\left(2-\frac{|b_n|^{2p}}{1+|b_n|^{2p}}\right)\right.\\ &\qquad\left.-\left(\mathbf{h}_n^H\mathbf{w}\left(\hat{s}-s\right)\right)^*\frac{|b_n|^{2p}}{1+|b_n|^{2p}}e^{j2\theta_n}\right].\end{aligned} \tag{16}$$

It follows from (10) that

$$\frac{\partial J}{\partial \mathbf{w}^*} = \left(\frac{\partial\hat{s}}{\partial\mathbf{w}^*}\right)\frac{\partial J}{\partial\hat{s}} = \mathbf{r}\left[\hat{s}-s\right]^*. \tag{17}$$

It follows from (16) that

$$\begin{aligned}\frac{\partial J}{\partial \mathbf{v}_n^*} &= \frac{\partial b_n}{\partial \mathbf{v}_n^*}\left(\frac{\partial J}{\partial b_n^*}\right)^* \\ &= \mathbf{y}_n \frac{|a_n|}{|2b_n|}\left[\left(\frac{\partial J}{\partial a_n^*}\right)^*\left(2-\frac{|b_n|^{2p}}{1+|b_n|^{2p}}\right)\right. \\ &\qquad \left. -\left(\frac{\partial J}{\partial a_n^*}\right)\frac{|b_n|^{2p}}{1+|b_n|^{2p}}e^{-j2\theta_n}\right] \\ &= \mathbf{y}_n \frac{|a_n|}{2|b_n|}\left[\left(\mathbf{h}_n^H\mathbf{w}\left(\hat{s}-s\right)\right)^*\left(2-\frac{|b_n|^{2p}}{1+|b_n|^{2p}}\right)\right. \\ &\qquad \left. -\left(\mathbf{h}_n^H\mathbf{w}\left(\hat{s}-s\right)\right)\frac{|b_n|^{2p}}{1+|b_n|^{2p}}e^{j2\theta_n}\right]. \end{aligned} \tag{18}$$

Note that (17) and (18) are the gradient directions obtained for a single training sample. Now with a training sequence of length L, we can obtain L derivatives, i.e., $\frac{\partial J}{\partial \mathbf{v}_n}(i)$ and $\frac{\partial J}{\partial \mathbf{w}}(i)$, for $i = 1, 2, \ldots, L$. Using the idea of batch processing in ANN, we average the L gradient directions to be

$$\mathbf{d}_w = \frac{1}{L}\sum_{i=1}^{L}\frac{\partial J}{\partial \mathbf{w}}(i), \tag{19}$$

and

$$\mathbf{d}_{v_n} = \frac{1}{L}\sum_{i=1}^{L}\frac{\partial J}{\partial \mathbf{v}_n}(i). \tag{20}$$

As the source node transmits training sequences periodically, we update $\mathbf{w}$ and $\mathbf{v}_n$'s using the momentum algorithm [14] to reduce the cost function monotonously. According to the idea of batch processing, the weights are updated once per batch. Then the gradient directions at the t-th $(t = 1, 2, ..., T)$ iteration are

$$\overline{\mathbf{d}}_w(t) = \beta\overline{\mathbf{d}}_w(t-1) + (1-\beta)\mathbf{d}_w(t), \tag{21}$$

and

$$\overline{\mathbf{d}}_{v_n}(t) = \beta\overline{\mathbf{d}}_{v_n}(t-1) + (1-\beta)\mathbf{d}_{v_n}(t), \tag{22}$$

with $\overline{\mathbf{d}}_w(0) = 0$ and $\overline{\mathbf{d}}_{v_n}(0) = 0$. Here the momentum parameter $\beta \in (0, 1)$. The updated receive beamforming weights of the destination and the relays are

$$\mathbf{w}(t) = \mathbf{w}(t-1) - \alpha\overline{\mathbf{d}}_w(t), \tag{23}$$

and

$$\mathbf{v}_n(t) = \mathbf{v}_n(t-1) - \alpha\overline{\mathbf{d}}_{v_n}(t),\ n = 1, \ldots, N, \tag{24}$$

respectively, where $\alpha \in (0, 1)$ is the learning rate.

The above gradients can be calculated by the distributed nodes with a small overhead of information exchange. According to (17), when the destination updates $\mathbf{w}$, only the signals $\mathbf{r}$, s and $\hat{s}$ are needed, among which s is the training sequences known *a priori*, while both $\mathbf{r}$ and $\hat{s}$ are locally available. Therefore, the destination can update the weight without any extra communication overhead.

According to (18), when the n-th relay updates $\mathbf{v}_n$, only $\mathbf{h}_n^H\mathbf{w}(\hat{s}-s)$ needs to be obtained from external, since $\mathbf{y}_n$, a_n, and b_n are all locally available. If the destination applies transmit beamforming weight $\mathbf{w}$ to the error sequence $\hat{s}-s$, i.e., to broadcasts a vector signal $[\mathbf{w}(\hat{s}-s)]^*$ to the relay nodes through the reverse channel using the time division duplex (TDD) mode, then the n-th relay receives $\mathbf{h}_n^T[\mathbf{w}(\hat{s}-s)]^*$. Thus, each relay will obtain what it needs, i.e., $\mathbf{h}_n^H\mathbf{w}(\hat{s}-s)$, subject to some channel noise.

B. A Frame Design Supporting the Modified BP Algorithm

To support the iterative BP algorithm, the frame needs to include the periodic pilots and the time slots for the back-propagation transmission. Fig. 4 shows a design which allows for the T rounds of the forward (source-to-relay) and backward (destination-to-relay) training sessions, between which the time gaps are introduced to accommodate the over-the-air propagation delay and the processing delay. Following the preamble is the payload, which is omitted from Fig. 4.

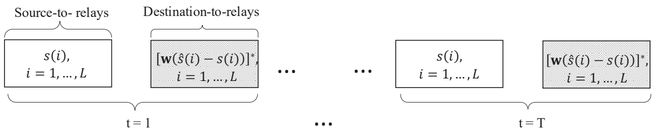

Fig. 4: T rounds of time slots for the forward (source-to-relay) and backward (destination-to-relay) pilot sequences.

In summary, the proposed scheme needs no explicit CSI, we only assume that the signal source transmits a L-length training sequence periodically, which is to be detected and synchronized by the destination. The scheme also needs no communication between the relays.

The overall scheme is summarized in the below.

Algorithm 1 The Interference-Resilient Relay Scheme

Initialization: $\mathbf{w}(0) = 0, \{\mathbf{v}_n(0) = 0\}_{n=1}^N, \alpha, \beta,$ $\overline{\mathbf{d}}_w(0) = 0, \overline{\mathbf{d}}_{v_n}(0) = 0$
Input: T rounds of L-length training sequence $s(i), i = 1, 2, ..., L$
Output: $\mathbf{w}$,$\{\mathbf{v}_n\}_{n=1}^N$

```
The destination synchronizes the training sequence.
for t = 1, 2, ..., T do
    The source transmits {s(i)}_{i=1}^L.
    The relays apply beamforming (2), respectively.
    The destination applies beamforming to obtain
        ŝ(i) = w(t)^H r(i),   i = 1, ..., L.
    The destination transmits {[w_n(t)(ŝ(i) − s(i))]^*}_{i=1}^L.
    The relays compute {∂J/∂w(i), ∂J/∂v_n(i)}_{i=1}^L according to (17) and (18).
    Compute d_w by (19) and d_{v_n} by (20).
    Compute d̄_w(t) by (21) and d̄_{v_n}(t) by (22).
    Update w(t) by (23) and {v_n(t)}_{n=1}^N by (24).
end for
```

IV. Numerical Simulations

In this section, we simulate a relay network as shown in Fig.1 to verify the effectiveness of the proposed algorithm. The source-to-relay channel and the relay-to-destination channel are assumed to be frequency-flat Rayleigh fading and are static in the simulated time duration. The source transmits a training sequence of length $L = 100$ periodically. The SNRs of the relays (denoted as ρ_{relay}) and the destination node (denoted as ρ_{dest}) are all 20dB. The relays are affected by K interferences, each is 10dB stronger than the signal. The PA parameter $p = 3$, and the moment parameters appeared in (21) and (23) for updating the beamforming weights are $\beta = 0.9$ and $\alpha = 0.05$.

For all the simulations, we use the output SINR of the destination as the performance metric to evaluate the performance of the system. The relationship between the output SINR and the MSE is shown in (9).

In the first example, we simulate the case where all the relays have only one receiving antenna ($M_r = 1$) and one transmitting antenna, and the destination node also has one receiving antenna ($M_d = 1$). The single-antenna relay nodes cannot suppress the interferences unless they coordinate to achieve distributed beamforming. Fig.5 shows, however, that the relay network can suppress interference effectively thanks to the distributed relay beamforming using our algorithm, even though the initialization yields a very low output SINR. About the x-axis, one iteration represents one round of forward and backward training session as illustrated in Fig. 4. As expected, the performance of the algorithm improves as the number of relays increases.

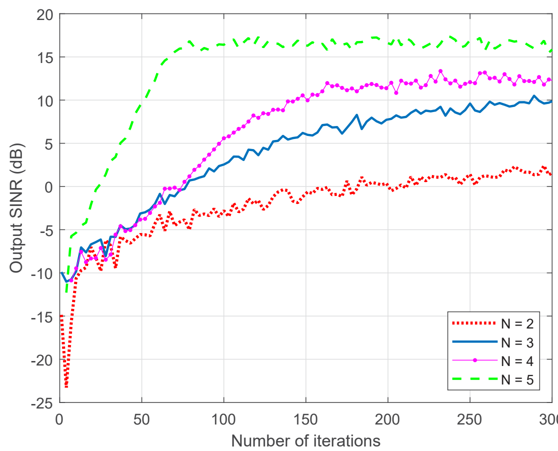

Fig. 5: Output SINR of the destination versus the number of iterations with respect to different number of relay nodes.

In the second example, we set the number of relays $N = 4$. Fig.6 shows the effect of the number of antennas on each relay. The output SINR of the relay network increases with the number of antennas of the relay nodes, which is also as expected.

In the third example, we assume five single-antenna relays, i.e., $M_r = 1$ and $N = 5$, and that the destination has only one antenna, i.e., $M_d = 1$. We simulate different number of interferences against the relays to test the interference suppression capability of the proposed scheme. Fig.7 shows that the proposed algorithm can suppress up to 4 interferences. As in general a five-antenna node can only suppress 4 interference, this result suggests that the proposed scheme may enable the distributed relay nodes to coordinate as if they were wire-connected as a centralized antenna array.

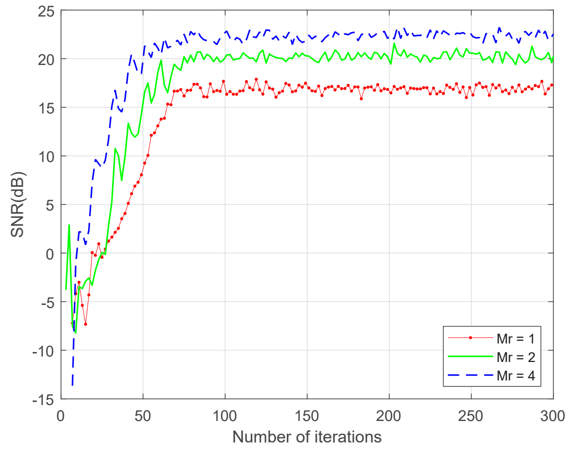

Fig. 6: Output SINR of the destination versus the number of iterations with respect to different number of receiving antennas of the relays.

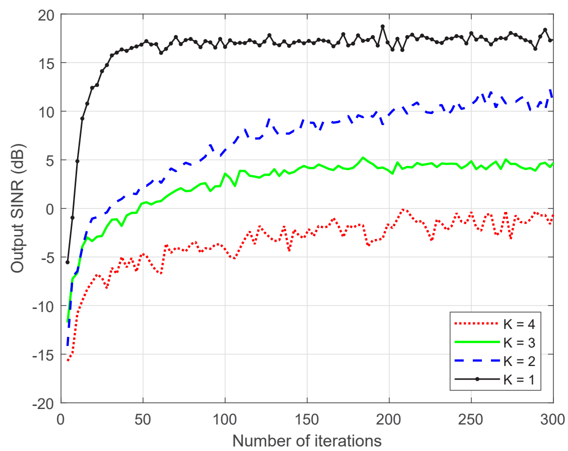

Fig. 7: Output SINR of the destination versus the number of iterations with respect to different number of interferences.

Fig.8 compares the performance of our algorithm with the algorithm in [4] in absence of interference. As the algorithm in [4] only constrains the average power of the relays, our power constraint is more stringent. But the performance of our algorithm is still close to that in [4]. Hence, the proposed interference-resilient scheme can work in the interference-free scenario as well. On the other hand, the method in [4] does not apply to the interference scenario.

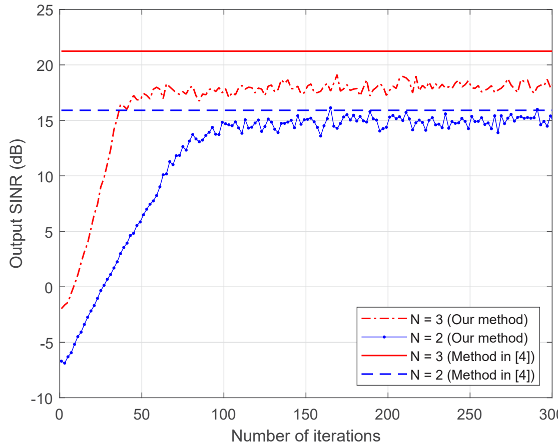

Fig. 8: Comparison between our method with the one in [4] in the output SINR of the destination for different number of the relay nodes.

The previous simulation we have assumed that the back-propagation, i.e., the destination-to-relays channel, is noise-free. In the final example, we assume when the destination broadcasts the error sequence $[\mathbf{w}(\hat{s}-s)]^*$, it is affected by Gaussian noise $\zeta \sim N(0, \sigma_\zeta^2)$ in the reverse channel, the SNR of the reverse channel is defined as

$$\rho_{\text{back}} \triangleq \frac{1}{\sigma_\zeta^2}, \tag{25}$$

even though the transmitted error sequence $\hat{s}-s$ may diminish as the output SINR increase. Then, the n-th relay obtains noisy gradients $\frac{\partial J}{\partial b_n^*} = \mathbf{h}_n^H \mathbf{w}(\hat{s}-s) + \zeta^*$. We compare the performance of our algorithm with or without noise in the reverse channel in Fig. 9. The curves in both cases almost overlap even ρ_{back} is as low as 0dB. Thus, the proposed scheme is robust to the noise in the back-propagate channel, which is because the batch processing of the L samples can mitigate the impact of the noise owing to the averaging in (19) and (20).

V. Conclusions

This paper presents an interference-resilient distributed relay scheme. By exploiting the striking similarity between the relay network and a three-layer neural network, we develop a modified back-propagation (BP) algorithm to train the beamforming-and-forwarding weights of the relay nodes via having the source transmit a training sequence, and via having the destination broadcast a beamformed error sequence to the relay nodes through the reverse channel. The distributed relay nodes can then achieve cooperative beamforming for interference suppression without information exchange between themselves. Simulation results show the effectiveness of the proposed scheme for a relay network with or without interferences.

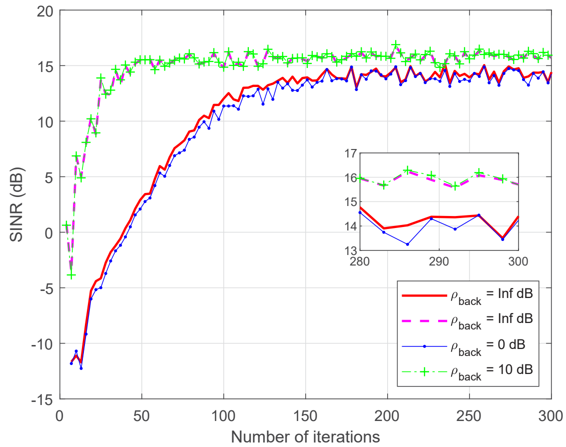

Fig. 9: Output SINR of the destination versus the number of iterations with and without noise in the reverse channel.

References

[1] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 3062–3080, 2004.

[2] G. Kramer, M. Gastpar, and P. Gupta, "Cooperative strategies and capacity theorems for relay networks," *IEEE Transactions on Information Theory*, vol. 51, no. 9, pp. 3037–3063, 2005.

[3] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity - Part I: System description," *IEEE Transactions on Communications*, vol. 51, no. 11, pp. 1927–1938, 2003.

[4] Y. Jing and H. Jafarkhani, "Network beamforming using relays with perfect channel information," *IEEE Transactions on Information Theory*, vol. 55, no. 6, pp. 2499–2517, 2009.

[5] G. Zheng, K. K. Wong, A. Paulraj, and B. Ottersten, "Collaborative-relay beamforming with perfect CSI: Optimum and distributed implementation," *IEEE Signal Processing Letters*, vol. 16, no. 4, pp. 257–260, 2009.

[6] L. Dong, A. P. Petropulu, and H. V. Poor, "Weighted cross-layer cooperative beamforming for wireless networks," *IEEE Transactions on Signal Processing*, vol. 57, no. 8, pp. 3240–3252, 2009.

[7] M. A. Maleki Sadr, M. Ahmadian Attari, and R. Amiri, "Robust relay beamforming against jamming attack," *IEEE Communications Letters*, vol. 22, no. 2, pp. 312–315, 2018.

[8] J. Li, A. P. Petropulu, and H. V. Poor, "Cooperative transmission for relay networks based on second-order statistics of channel state information," *IEEE Transactions on Signal Processing*, vol. 59, no. 3, pp. 1280–1291, 2011.

[9] V. Havary-Nassab, S. Shahbazpanahi, A. Grami, and Z. Luo, "Distributed beamforming for relay networks based on second-order statistics of the channel state information," *IEEE Transactions on Signal Processing*, vol. 56, no. 9, pp. 4306–4316, 2008.

[10] G. Zheng, K. K. Wong, A. Paulraj, and B. Ottersten, "Robust collaborative-relay beamforming," *IEEE Transactions on Signal Processing*, vol. 57, no. 8, pp. 3130–3143, 2010.

[11] R. Mudumbai, J. Hespanha, U. Madhow, and G. Barriac, "Distributed transmit beamforming using feedback control," *IEEE Transactions on Information Theory*, vol. 56, no. 1, pp. 411–426, 2010.

[12] E. Perahia and R. Stacey, *Next generation wireless LANs: 802.11n and 802.11ac*. 2013.

[13] D. Guo, "Gaussian channels: Information, estimation and multiuser detection," 2004.

[14] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.