Receiver Combining Compute-and-Forward

Mohamad Dia
Master Semester Project
Supervisors: Professor Michael Gastpar and Jingge Zhu
Laboratory for Information in Networked Systems, EPFL, Switzerland

Abstract

In this work, we present a generalized compute-and-forward scheme for the single input multiple output (SIMO) channels. The proposed receiver combining compute-and-forward is well-suited to settings where a single receiver is equipped with multiple antennas. We show that combining channel outputs in a specific way makes the effective channel more suitable for decoding successive linear combinations of messages, and thus recover all messages, than using individual channel outputs at each antenna. Furthermore, we provide empirical optimality of the proposed scheme for the 2-user SIMO MAC channels after showing that points on the boundary of capacity region are achievable by single point decoding without using time sharing or joint decoding.

Index Terms

SIMO, compute-and-forward, nested lattice codes, structured codes, MAC.

I. Introduction

The prevalent Shannon's random coding argument used to obtain the highest achievable rate for point-to-point communication systems does not give persuasive answers for general multi-user communication networks. Nowadays, the search for low-complexity structured codes represents a challenging topic in the communication community.

Authors in [1] demonstrated that there exists "good" nested lattice code which can be used along with lattice decoding (instead of ML decoding) to reach the full capacity of point-to-point AWGN channel, where MMSE scaling and dithering are used to transform the power-constrained AWGN channel into modulo-lattice additive noise channel. The construction of such codes is given in [2]. A central role of lattice codes emerges in [3], where lattices codes can be used for interference alignment on the signal level.

Recently, the algebraic structure of nested lattice codes, where integer combinations of codewords are themselves codewords, was employed in a new strategy called compute-and-forward [4] that exploits interferences in wireless AWGN networks (instead of treating them as noise). The proposed strategy enables relays to decode linear equations of transmitted messages, and thus obtains higher rates. This strategy was extended in [5] to successive compute-and-forward by allowing one single relay to decode multiple independent linear equations using the same channel output. The key idea in successive computation is to remove the contribution of decoded equation from the original channel output and form a new effective channel output so that a second equation can be decoded easier. A modified compute-and-forward scheme was presented in [6] which allows for asymmetric rates and achieves larger rate region using CSIT. The main idea of the modified scheme is to transmit a scaled version of the lattice codeword which does not have to lie in the lattice used for coding in order to allow for flexibility in controlling individual message rates.

The non-integer penalty problem arises in the compute-and-forward scheme due to the mismatch between real channel coefficients and decoded integer combinations of codewords. A new linear receiver architecture was developed in [7] to force the effective channel to a full-rank integer-valued channel matrix. The decoder then recover integer combinations of codewords according to the entries of the effective channel matrix.

The authors in [8] and [9] showed that using linear pre-coding and integer-forcing beamforming with compute-and-forward eliminate the end-to-end interferences. The proposed scheme is useful whenever cooperation is not allowed at the receiver side and it uses compute-and-forward in the reverse order, in the sense that precoded functions of messages are sent to undo the effect of linear combination and therefore each receiver obtains its desired message.

Using compute-and-forward in wireless AWGN networks showed that interference can be exploited and thus obtaining better rates than classical approach where interference was treated as noise. A further study in [10] showed that interference is beneficial in some networks, where the combination of compute-and-forward with inverse compute-and-forward is able to achieve rates outside the rate region of the same network without interference.

Compute-and-forward was applied in [11] along with rate splitting to many-to-one cognitive interference networks showing that it can enlarge the achievable rate region for a large range of parameters, and remain in a constant gap from capacity independently of the number of users.

In this work we provide a generalized compute-and-forward scheme used whenever cooperation is allowed at the receiver side (e.g. SIMO channels), where different channel outputs are combined in the MMSE sense to make multiple linear equations of codewords decoded easier than single equation per receiver.

We start with preliminaries in section II by surveying nested lattice code construction, encoder design, decoder design, and computation rate of compute-and-forward. We elaborate on our receiver combining scheme in section III and then provide a case study in section IV. Section V is dedicated to the recapitulation of the main results and topics for future studies.

II. Preliminaries: COMPUTE-AND-FORWARD

The compute-and-forward scheme, as proposed in [4], is a way of communicating linear functions of messages, where each relay is involved in decoding integer combinations of codewords and thus recovering linear functions of the transmitted messages. This relaying strategy is applicable to any multi-user AWGN network with relays.

We will restrict ourselves to discrete-time real valued-channel with the equivalent channel model:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{z} \tag{1}$$

where $\mathbf{H} \in \mathbb{R}^{M\times L}$ is the channel matrix for L transmitters and M receivers, and $\mathbf{z}$ is i.i.d. zero-mean Gaussian noise.

A. Nested Lattice Code Construction

The key idea of compute-and-forward is to exploit the interference in multi-user networks by decoding linear combination of messages instead of decoding individual messages and treating interference as noise. The nested lattice code, which has been shown to be capacity achieving on the point-to-point AWGN channel and whose algebraic properties ensure that integer combinations of codewords can be decoded reliably, is a natural fit for this purpose.
The detailed description of lattice and lattice codes construction are shown in [1] and [4]. We will introduce briefly the nested lattice codes used for the compute-and-forward scheme.

A lattice Λ is a discrete subgroup of $\mathbb{R}^n$ with the property that if $t_1, t_2 \in \Lambda$, then $t_1+t_2 \in \Lambda$. Any lattice can be written as a linear transformation of the integer vector, $\Lambda = \mathbf{B}\mathbb{Z}^n$ for some generator matrix $\mathbf{B} \in \mathbb{R}^{n\times n}$. Then nearest neighbour lattice quantizer is a function, $Q_\Lambda : \mathbb{R}^n \longrightarrow \Lambda$, that maps vectors to the nearest lattice point in the Euclidean sense, where $Q_\Lambda(x) = \underset{t\in\Lambda}{\arg\min}\|x-t\|$. The Voronoi region of each lattice point is the subset of points in $\mathbb{R}^n$ that quantize to that lattice point, whereas the fundamental Voronoi region $\mathcal{V}$ is the Voronoi region of the zero vector, $\mathcal{V} = \{x : Q_\Lambda(x) = 0\}$. The modulo operation with respect to lattice Λ returns the quantization error: $[x] \mod \Lambda = x - Q_\Lambda(x)$, and satisfies the laws listed in [4] . A pair of lattices Λ_{COARSE}, Λ_{FINE} is nested if $\Lambda_{COARSE} \subseteq \Lambda_{FINE}$, where Λ_{COARSE} is called the coarse lattice and Λ_{FINE} the fine lattice.

A nested lattice code $\mathcal{C}$ can be constructed using coarse lattice for shaping and fine lattice for codewords. Thus, $\mathcal{C}$ is created by taking the set of fine lattice points that fall within the fundamental Voronoi region of the coarse lattice, $\mathcal{C} = \{t \in \mathbb{R}^n : t \in \Lambda_{FINE} \bigcap \mathcal{V}_{COARSE}\}$, where V_{COARSE} acts as a power constraint defined by its second moment. The rate of the nested lattice code is:

$$r = \frac{1}{n}\log|\mathcal{C}| = \frac{1}{n}\log\frac{Volume(\mathcal{V}_{COARSE})}{Volume(\mathcal{V}_{FINE})} \tag{2}$$

Erez and Zamir have shown in [1] that there exist nested lattice codes; where the coarse lattice is good for quantization, covering, and AWGN and the fine lattice is good for AWGN; which can achieve capacity of point-to-point AWGN channel. The n-dimensional lattice is constructed using type A construction [2] with three steps: i) Draw generating matrix $\mathbf{G}$ whose elements are i.i.d. uniformly distributed over the finite field. ii) Define the discrete codebook using $\mathbf{G}$. iii) Form the lattice by lifting the codebook to reals. This construction can be generalized for the case of multi-user network, by including multiple nested fine lattices $\Lambda_1 \subseteq \Lambda_2 \subseteq \cdots \subseteq \Lambda_L$ as in [4], or even multiple nested coarse lattices to allow for asymmetric rates [6] and power splitting [11] where all the lattices are good in the sense of [2].

B. Encoder Design

After constructing the nested lattice codes, the following steps are done at the encoder side:

1) Mapping: The encoder maps each finite field message with length k_ℓ in $\mathbb{F}_p^{k_\ell}$ to a lattice point in $\mathbb{R}^n$, $t_\ell = \phi(w_\ell)$. It can be shown that there exists a one-to-one map between finite field messages and the elements of the nested lattice code that preserves linearity [4].

2) Dithering: The encoder then applies to each lattice point a dither d_ℓ, which is also known at the receivers, drawn independent and uniformly over the fundamental Voronoi region of the coarse lattice. It is shown in [1] that this dithering makes the transmitted codeword independent from the lattice point.

3) Modulo coarse lattice: Form the channel input x_ℓ to be transmitted over the channel by taking the modulus with respect to the coarse lattice $x_\ell = [t_\ell - d_\ell] \mod \Lambda_{COARSE}$.

C. Decoder Design

Each decoder received a noisy linear combination of the codewords

$$y_m = \sum_{\ell=1}^{L} h_{m\ell} x_\ell + z_m, \; m \in [1:M] \tag{3}$$

with $y_m \in \mathbb{R}^n$ denoting the channel ouput, $x_\ell \in \mathbb{R}^n$ denoting the channel input, $h_{m\ell} \in \mathbb{R}$ denoting the channel gain (note that $\mathbf{h}_m \in \mathbb{R}^L$ is the transpose of the m_{th} row of $\mathbf{H}$ as given in the notation (1) above), and $z_m \in \mathbb{R}^n$ denoting the Gaussian noise with $z_m \sim \mathcal{N}(\mathbf{0}, \mathbf{I}^{n\times n})$.

The decoder's ultimate goal is to recover linear function of transmitted messages $u_m = \bigoplus_{\ell=1}^{L} q_{m\ell} w_\ell$ over the finite field with $q_{m\ell} \in \mathbb{F}_p$, by decoding integer combination of lattice codewords $v_m = [\sum_{\ell=1}^{L} a_{m\ell} t_\ell] \mod \Lambda_{COARSE}$ with $a_{m\ell} \in \mathbb{Z}$. To this end, the encoder does the following steps given the access to all dithers:

1) Scaling: The decoder scales its observed output channel by the minimum mean-squared error (MMSE) coefficient α_m.

2) Removing dither: The decoder then removes the dither effect according to desired coefficients $a_{m\ell}$.

$$y'_m = \alpha_m y_m + \sum_{\ell=1}^{L} a_{m\ell} d_\ell \tag{4}$$

3) Quantization: The m^{th} decoder quantizes the result onto the fine lattice Λ_m which corresponds to the finest lattice among lattice codewords received by m.

4) Modulo coarse lattice: Take the modulus with respect to the coarse lattice to get an estimate of the desired integer combination.

$$\hat{v}_m = [Q_{\Lambda_m}(y'_m)] \mod \Lambda_{COARSE} \tag{5}$$

We can show using the distributive property of $\mod \Lambda$ that $\hat{v}_m = [Q_{\Lambda_m}(v_m + z_{eq_m})] \mod \Lambda_{COARSE}$, where

$$z_{eq_m} = \alpha_m z_m + \sum_{\ell=1}^{L} (\alpha_m h_{m\ell} - a_{m\ell}) x_\ell \tag{6}$$

Note that x_ℓ has the same distribution as d_ℓ by construction [1]. Then, it can be shown that $\hat{v}_m = v_m$ as long as

$$\Pr(z_{eq_m} \neq \mathcal{V}_{\Lambda_\mathrm{m}}) \longrightarrow 0 \tag{7}$$

5) Inverse map: Finally, the decoder maps back the lattice equation to the finite field in order to get the desired equation $u_m = \phi^{-1}(v_m)$. Note that u_m is obtained assuming that (7) holds and the mapping preserves one-to-one linearity as done in [4].

D. Computation Rate

For fine lattices good for AWGN as constructed in II-A, (7) holds exponentially in n if the volume-to-noise ratio of the fine lattice is bounded below as follows

$$\frac{(Volume(\mathcal{V}_{\Lambda_\mathrm{m}}))^{2/n}}{(\mathbb{E}\|z_{eq_m}\|^2)/n} > 2\pi e \tag{8}$$

For coarse lattice good for quantization as constructed in II-A, we have that the normalized second moment of the coarse lattice is bounded above as follows

$$G(\Lambda_{COARSE}) = \frac{\sigma^2_{\Lambda_{COARSE}}}{(Volume(\mathcal{V}_{COARSE}))^{2/n}} < \frac{1+\delta}{2\pi e} \tag{9}$$

$\forall \delta > 0$ if n is large enough, where the second moment of the coarse lattice $\sigma^2_{\Lambda_{COARSE}} = P$ (the power constraint) by construction.

Substituting (8) and (9) in (2), we can show that (7) is equivalent to

$$r_\ell < \min_{m:a_{m\ell}\neq 0} \max_{\mathbf{a}_m} \mathcal{R}_{CF}(\mathbf{h}_m, \mathbf{a}_m) \tag{10}$$

where $\mathcal{R}_{CF}(\mathbf{h}_m, \mathbf{a}_m)$ is the computation rate achievable by relay m using compute-and-forward with respect to coefficient vector $\mathbf{a}_m$

$$\mathcal{R}_{CF}(\mathbf{h}_m, \mathbf{a}_m) := \max_{\alpha_m} \frac{1}{2} \log^+ \left(\frac{P}{(\mathbb{E}\|z_{eq_m}\|^2)/n} \right) = \max_{\alpha_m} \frac{1}{2} \log^+ \left(\frac{P}{\alpha_m^2 + P\|\alpha_m \mathbf{h}_m - \mathbf{a}_m\|^2} \right) \tag{11}$$

with $\log^+(x) = \max(\log(x), 0)$.

This means that the modulo integer combinations at all relays can be decoded as long as the

messages' rates are within the computation rate region.
The computation rate in (11) is uniquely maximized over α_m with the MMSE coefficient

$$\alpha_{MMSE} = \frac{P\mathbf{h}_m^T\mathbf{a}_m}{1+P|\mathbf{h}_m|^2} \tag{12}$$

which gives

$$\mathcal{R}_{CF}(\mathbf{h}_m,\mathbf{a}_m) = \frac{1}{2}\log^+\left(\left(\|\mathbf{a}_m\|^2 - \frac{P\left(\mathbf{h}_m^T\mathbf{a}_m\right)^2}{1+P|\mathbf{h}_m|^2}\right)^{-1}\right) \tag{13}$$

Note that finding the best equation that maximizes the rate, i.e. the maximization problem over $\mathbf{a}_m$ in (10), is connected to Diophantine approximation.

III. Receiver Combining Compute-and-Forward

In the regular compute-and-forward scheme applied on any channel of the form (1), each relay $m \in [1:M]$ decodes a linear function u_m of the transmitted messages. In order to recover all individual messages $w_\ell,\ \ell \in [1:L]$ without treating interferences as noise, we had two scenarios:

1) If the receivers are allowed to communicate, they send their decoded functions to a central node in order to solve for individual messages. The central node is concerned in solving the system $\mathbf{u} = \mathbf{Q}\mathbf{w}$ with $\mathbf{Q} \in \mathbb{F}_{\mathbf{p}}^{\mathbf{M}\times\mathbf{L}}$. All the messages can be recovered if and only if $M \geq L$ and the relays decode at least L independent functions (which is equivalent to say that $\text{rank}(\mathbf{Q}) = L$).
2) Each relay decodes multiple equations using successive computation (by removing the contribution of the decoded function from the original channel observation and then decode another equation), or using superposition (by encoding each message with multiple lattice codebooks and then send scaled versions of them).

Furthermore, whenever the receivers can operate under complete cooperation, which is indeed applicable in many systems (e.g. SIMO systems), we can generalize the regular compute-and-forward scheme to receiver combining compute-and-forward. The proposed scheme combines all the channel outputs in the MMSE sense to form a single channel output, from which multiple linear functions are decoded using successive computations in order to recover all individual messages. The joint channel output is combined in a way to better decode a linear function than what each individual receiving antenna does, then the decoded function is used to form new channel outputs in a way that makes successive computations easier. Therefore, whenever we can make the combination optimal, the first linear function is decoded better (or at least not worse) than what each receiver decodes alone, and thus we can outperform the regular compute-and-forward in the second scenario listed above. And whenever the new channel outputs, which are formed after removing the contribution of previous decoded functions, are tuned to better decode successive functions, we can outperform the regular compute-and-forward in the first scenario as well.

A. *Combining Decoder*

The receiver combining compute-and-forward is applied here to the same model (1). As the combined scheme requires modifications at receiver side only, we will use the same nested lattice code construction as in II-A, where we had coarse lattice Λ_{COARSE} good for quantization and used for shaping, in addition to nested fine lattice chain $\Lambda_1 \subseteq \Lambda_2 \subseteq \cdots \subseteq \Lambda_L$ good for AWGN and used for codewords. The encoders perform the same operations as in II-B. Under the assumption that receivers operate with complete cooperation, one single decoder combines the channel observations y_m of all receivers $m \in [1:M]$ in the MMSE sense, removes the dither effect according to the desired coefficients a_ℓ, and form one single combined output

$$y' = \boldsymbol{\alpha}^T \mathbf{y} + \sum_{\ell=1}^{L} a_\ell d_\ell \tag{14}$$

with $\boldsymbol{\alpha} \in \mathbb{R}^M$ and boldface lower-case letters denote column vectors.

After quantizing onto the finest lattice among all received lattice codewords and taking the modulus with respect to the coarse lattice, we can get an estimate of the desired modulo integer combination $v = [\sum_{\ell=1}^{L} a_\ell t_\ell] \mod \Lambda_{COARSE}$ using the distributive property of $\mod \Lambda$

$$\hat{v} = [Q_{\Lambda_{FINE}}(v + z_{eq})] \mod \Lambda_{COARSE} \tag{15}$$

where the equivalent noise z_{eq} is defined as

$$z_{eq} = \boldsymbol{\alpha}^T \mathbf{z} + \sum_{\ell=1}^{L} (\boldsymbol{\alpha}^T \mathbf{h}_{*\ell} - a_\ell) x_\ell \tag{16}$$

with $\mathbf{h}_{*\ell}$ denotes the ℓ^{th} column of $\mathbf{H}$, and x_ℓ having the same distribution as d_ℓ by construction. Knowing that the probability of error $\Pr(\hat{v} \neq v)$ is equal to $\Pr(z_{eq} \neq \mathcal{V}_{\Lambda_{\text{FINE}}})$, and our lattices are good for quantization and AWGN, we can use the same reasoning as in II-D to get the computation rate of our scheme with respect to the equation coefficient vector $\mathbf{a} \in \mathbb{Z}^L$

$$\mathcal{R}_{CF}(\mathbf{H}, \mathbf{a}) = \max_{\boldsymbol{\alpha}} \frac{1}{2} \log^+ \left(\frac{P}{\|\boldsymbol{\alpha}\|^2 + P\|\mathbf{H}^T \boldsymbol{\alpha} - \mathbf{a}\|^2} \right) \tag{17}$$

Let $f(\boldsymbol{\alpha}) = \|\boldsymbol{\alpha}\|^2 + P\|\mathbf{H}^T \boldsymbol{\alpha} - \mathbf{a}\|^2$, which is a convex function (quadratic in $\boldsymbol{\alpha}$) whose global minimum is attained by setting the first derivative to zero

$$f(\boldsymbol{\alpha}) = \boldsymbol{\alpha}^T \boldsymbol{\alpha} + P(\mathbf{H}^T \boldsymbol{\alpha} - \mathbf{a})^T (\mathbf{H}^T \boldsymbol{\alpha} - \mathbf{a}) \tag{18}$$

$$\frac{df}{d\boldsymbol{\alpha}} = 2\boldsymbol{\alpha} + P(2\mathbf{H}\mathbf{H}^T \boldsymbol{\alpha} - 2\mathbf{H}\mathbf{a}) = \mathbf{0} \tag{19}$$

$$\boldsymbol{\alpha}^* = P(\mathbf{I}_M + P\mathbf{H}\mathbf{H}^T)^{-1} \mathbf{H}\mathbf{a} \tag{20}$$

After substituting (20) in (17), expanding the squared norm to scalar product and using the fact that $(\mathbf{I}_M + P\mathbf{H}\mathbf{H}^T)^{-1}$ is symmetric, we can get the computation rate of the combined scheme.

Theorem 1: For any channel model (1), the achievable computation rate of the receiver combining compute-and-forward with respect to the equation coefficient vector $\mathbf{a} \in \mathbb{Z}^L$ is

$$\mathcal{R}_{CF}(\mathbf{H}, \mathbf{a}) = \frac{1}{2} \log^+ \left(\left(\|\mathbf{a}\|^2 - P\mathbf{a}^T\mathbf{H}^T \left(\mathbf{I}_M + P\mathbf{H}\mathbf{H}^T \right)^{-1} \mathbf{H}\mathbf{a} \right)^{-1} \right) \tag{21}$$

by combining all the channel outputs whenever the receivers are allowed to cooperate.

B. Successive Compute-and-Forward for the Combined Scheme

Successive compute-and-forward as defined in [5] is an analogue to the standard successive interference cancellation adjusted to fit with the compute-and-forward scheme. In the classical approach, where the relay is concerned in decoding one single codeword, the successive interference cancellation is applied by subtracting the decoded codeword from the channel output in order to decode another codeword with less interference than what is encountered in the first decoding. While in compute-and-forward, the successive computation is applied by removing the contribution of the decoded integer combination of codewords from the original channel output, in order to form new equivalent channel which is more suitable to decode a second linear combination.

Note that in the classical approach, we can completely cancel the decoded codeword from the original channel output making the new channel output better, in terms of effective noise, for second decoding. Whereas under compute-and-forward approach it is not that clear how we can cancel the effect of what was decoded at the first place, because of the mismatch between the decoded integer combination and the received linear function. Therefore, is not always optimal to subtract the previous decoded integer combination, and the challenge is in how to remove its contribution from the original channel in a way that makes the new effective channel more suitable for second computation.

We will use similar procedures for successive computation as in [5] adapted to our combined scheme. The following procedures show how we can decode the second linear combination by mixing the first decoded linear combination with the original channel outputs to form new equivalent channel output $\tilde{y}$. The same procedures can be repeated to any number of successive computations by mixing the previously decoded functions with the original channel output.

1) Recover the sum of codewords from modulo sum: After decoding the first modulo integer combination of lattice points $v^{(1)} = [\sum_{\ell=1}^{L} a_\ell t_\ell] \mod \Lambda_{COARSE}$, we need to recover the sum of codewords $s = \sum_{\ell=1}^{L} a_\ell x_\ell$. We know from [5] that we can do this with vanishing probability of error as long as the messages rates are within the computation rate region. The following steps are needed to recover the sum of codewords:

i) Dither back the modulo sum given the access to all dithers:

$$[v^{(1)} - \sum_{\ell=1}^{L} a_\ell d_\ell] \mod \Lambda_{COARSE} = [s] \mod \Lambda_{COARSE} \tag{22}$$

ii) Subtract this from combined channel output:

$$\begin{aligned} r &= \boldsymbol{\alpha}^T \mathbf{y} - [s] \mod \Lambda_{COARSE} \\ &= Q_{\Lambda_{COARSE}}(s) + z_{eq} \end{aligned} \tag{23}$$

with z_{eq} the same as (16).

iii) Coarse lattice quantization:

$$\begin{aligned} Q_{\Lambda_{COARSE}}(r) &= Q_{\Lambda_{COARSE}}(Q_{\Lambda_{COARSE}}(s) + z_{eq}) \\ &= Q_{\Lambda_{COARSE}}(s) \qquad w.h.p. \end{aligned} \tag{24}$$

as long as $\sigma^2_{\Lambda_{COARSE}} = P > \mathbb{E}\|z_{eq}\|^2$ (i.e. for $\mathcal{R}_{CF}(\mathbf{H}, \mathbf{a}) > 0$).

iv) Recover s by adding the results of i) and iii).

2) Remove the contribution of first equation: After recovering the sum of codewords decoded from first computation, we need to remove its contribution to each channel output $y_m, m \in [1 : M]$. To this end, $\forall m \in [1 : M]$ we subtract the projection of s onto y_m from y_m to get the following results in the matrix form:

$$\begin{aligned} \mathbf{y}_\perp &= \mathbf{y} - \bar{\mathbf{s}} \\ &= \mathbf{G}\mathbf{x} + \mathbf{z} \end{aligned} \tag{25}$$

with $\bar{\mathbf{s}} = \left(\dfrac{\mathbf{H}\mathbf{a}}{\|\mathbf{a}\|^2}\right) s$ is the vector representing projection of s on each y_m, and $\mathbf{G} = \left(\mathbf{H} - \dfrac{\mathbf{H}\mathbf{a}\mathbf{a}^T}{\|\mathbf{a}\|^2}\right)$.

3) Create new effective channel: Combine the refined channel outputs with the first decoded integer combination in the MMSE sense to form new equivalent channel output

$$\tilde{y} = \boldsymbol{\gamma}^T \mathbf{y}_\perp + \mu s \tag{26}$$

Now we can decode the second integer combination $v^{(2)} = [\sum_{\ell=1}^{L} b_\ell t_\ell] \mod \Lambda_{COARSE}$ from $\tilde{y}$ using the same procedures as in III-A and replacing $\boldsymbol{\alpha}^T \mathbf{y}$ by $\tilde{y}$. We get the following computation rate with respect to the second equation coefficient vector $\mathbf{b} \in \mathbb{Z}^L$

$$\mathcal{R}_{SCF}(\mathbf{H}, \mathbf{a}, \mathbf{b}) = \max_{\boldsymbol{\gamma}, \mu} \frac{1}{2} \log^+ \left(\frac{P}{\|\boldsymbol{\gamma}\|^2 + P\|\mathbf{G}^T\boldsymbol{\gamma} + \mu\mathbf{a} - \mathbf{b}\|^2} \right) \tag{27}$$

Let $f(\boldsymbol{\gamma}, \mu) = \|\boldsymbol{\gamma}\|^2 + P\|\mathbf{G}^T\boldsymbol{\gamma} + \mu\mathbf{a} - \mathbf{b}\|^2$, which is a convex function whose global minimum is attained by setting the first derivative to zero. (Note that $\mathbf{G}\mathbf{a} = \mathbf{0}$ due to projection in (25))

$$f(\boldsymbol{\gamma}, \mu)) = \boldsymbol{\gamma}^T\boldsymbol{\gamma} + P(\mathbf{G}^T\boldsymbol{\gamma} + \mu\mathbf{a} - \mathbf{b})^T(\mathbf{G}^T\boldsymbol{\gamma} + \mu\mathbf{a} - \mathbf{b}) \tag{28}$$

$$\frac{df}{d\boldsymbol{\gamma}} = 2\boldsymbol{\gamma} + P(2\mathbf{G}\mathbf{G}^T\boldsymbol{\gamma} - 2\mathbf{G}\mathbf{b}) = \mathbf{0} \tag{29}$$

$$\frac{df}{d\mu} = P(2\|\mathbf{a}\|^2\mu - 2\mathbf{a}^T\mathbf{b}) = 0 \tag{30}$$

$$\boldsymbol{\gamma}^* = P(\mathbf{I}_M + P\mathbf{G}\mathbf{G}^T)^{-1}\mathbf{G}\mathbf{b} \tag{31}$$

$$\mu^* = \frac{\mathbf{a}^T\mathbf{b}}{\|\mathbf{a}\|^2} \tag{32}$$

After substituting (31) and (32) in (27), expanding the squared norm to scalar product and using the fact that $\left(\mathbf{I}_M + P\mathbf{G}\mathbf{G}^T\right)^{-1}$ is symmetric and $\mathbf{G}\mathbf{a} = \mathbf{0}$, we can get the computation rate of the second successive equation defined as

$$\mathcal{R}_{SCF}(\mathbf{H},\mathbf{a},\mathbf{b}) = \frac{1}{2}\log^+\left(\left(\|\mathbf{b}\|^2 - \frac{(\mathbf{a}^T\mathbf{b})^2}{\|\mathbf{a}\|^2} - P\mathbf{b}^T\left(\mathbf{H} - \frac{\mathbf{H}\mathbf{a}\mathbf{a}^\mathbf{T}}{\|\mathbf{a}\|^\mathbf{2}}\right)^T\left(\mathbf{I}_M + P\left(\mathbf{H}\mathbf{H}^T - \frac{\mathbf{H}\mathbf{a}\mathbf{a}^T\mathbf{H}^T}{\|\mathbf{a}\|^2}\right)\right)^{-1}\left(\mathbf{H} - \frac{\mathbf{H}\mathbf{a}\mathbf{a}^\mathbf{T}}{\|\mathbf{a}\|^\mathbf{2}}\right)\mathbf{b}\right)^{-1}\right) \tag{33}$$

Theorem 2: For any channel model (1), the achievable computation rate of the receiver combining compute-and-forward with respect to equation coefficient vectors $\mathbf{a}, \mathbf{b} \in \mathbb{Z}^L$ is

$$\mathcal{R}(\mathbf{H},\mathbf{a},\mathbf{b}) = \min\{\mathcal{R}_{CF}(\mathbf{H},\mathbf{a}), \mathcal{R}_{SCF}(\mathbf{H},\mathbf{a},\mathbf{b})\} \tag{34}$$

with $\mathcal{R}_{CF}(\mathbf{H},\mathbf{a})$ and $\mathcal{R}_{SCF}(\mathbf{H},\mathbf{a},\mathbf{b})$ as defined in (21) and (33) respectively.

C. Asymmetric Rates for the Combined Scheme

The modification applied in [6] on the regular compute-and-forward to allow for asymmetric rates using CSIT, and thus achieving larger rate region, can be adapted also to our combined scheme. This generalized scheme necessitate the following modifications:

1) Lattice code construction: Construct a fine lattice Λ used for quantization and good in the sense of [1] (note that in our scheme where all the channel outputs are combined with one decoder, we used single fine lattice instead of a chain of fine lattices as in [6]). For $\beta_1 \geq \cdots \geq \beta_L > 0$, construct a chain of nested coarse lattices $\Lambda_1^s \subseteq \Lambda_2^s \subseteq \cdots \subseteq \Lambda_L^s \subseteq \Lambda$ good in the sense of [1] and used for shaping, with $\sigma^2_{\Lambda^s_\ell} = \beta_\ell^2 P$.

2) Encoder: Form the channel inputs $x_\ell = [{}^{t_\ell}/\beta_\ell - d_\ell] \mod {}^{\Lambda^s_\ell}/\beta\ell$ with d_ℓ uniformly distributed over ${}^{\mathcal{V}^s_\ell}/\beta_\ell$.

3) Decoder: Remove the dither effect according to the coefficients $\tilde{\mathbf{a}} = \boldsymbol{\beta}\star\mathbf{a}$, where $\star$ denotes the elementwise multiplication.

$$y' = \boldsymbol{\alpha}^T\mathbf{y} + \sum_{\ell=1}^{L}\beta_\ell a_\ell d_\ell \tag{35}$$

4) Successive computation: Using the same procedures as in III-B, we can recover the sum $\tilde{s} = \sum_{\ell=1}^{L}\tilde{a}_\ell x_\ell$ instead of s, and then decode the second equation with respect to the coefficients $\mathbf{b}$ by removing the dither effect according to $\tilde{\mathbf{b}} = \boldsymbol{\beta}\star\mathbf{b}$.

Theorem 3: For any channel model (1) and $\beta_1 \geq \cdots \geq \beta_L = 1$, there exist lattice codes such that the achievable computation rate tuple of the receiver combining compute-and forward with respect to equation coefficient vectors $\mathbf{a}, \mathbf{b} \in \mathbb{Z}^L$ is

$$\mathcal{R}_\ell(\mathbf{H},\mathbf{a},\mathbf{b},\boldsymbol{\beta}) = \min\{\mathcal{R}_{\ell_{CF}}(\mathbf{H},\mathbf{a},\boldsymbol{\beta}), \mathcal{R}_{\ell_{SCF}}(\mathbf{H},\mathbf{a},\mathbf{b},\boldsymbol{\beta})\},\ \ell \in [1:L] \tag{36}$$

with $\mathcal{R}_{\ell_{CF}}(\mathbf{H},\mathbf{a},\boldsymbol{\beta})$ and $\mathcal{R}_{\ell_{SCF}}(\mathbf{H},\mathbf{a},\mathbf{b},\boldsymbol{\beta})$ as defined in (21) and (33) respectively after replacing $\mathbf{a}$ by $\tilde{\mathbf{a}}$, $\mathbf{b}$ by $\tilde{\mathbf{b}}$ and the unity numerator inside the $\log^+$ function by β_ℓ^2.

IV. Case Study:SIMO MAC

We will consider the two-user MAC channel with multiple receiver antennas, and we will compare the performance of receiver combining compute-and-forward to the performance of regular compute-and-forward with asymmetric rates for both cases.
In regular compute-and-forward, the relays at both receiver antennas decode two linearly independent equations and pass them to the destination in order to recover individual messages. Whereas, in the receiver combining scheme one single relay at the destination combines the channel outputs seen by $\mathbf{H}$ for the equivalent channel model in (1), and decodes two linearly independent equations using successive computations. The highest achievable asymmetric rates for regular and combined schemes are given in (37) and (38) respectively:

$$r_\ell = \begin{cases} \min\limits_{m\in[1:2]} \max\limits_{\mathbf{a}_m} \mathcal{R}_{\ell_{CF}}(\mathbf{h}_m, \mathbf{a}_m, \boldsymbol{\beta}) & \text{if } \mathbf{a}_1 \perp \mathbf{a}_2 \\ 0 & \text{o.w.} \end{cases} \tag{37}$$

with $\mathcal{R}_{\ell_{CF}}(\mathbf{h}_m, \mathbf{a}_m, \boldsymbol{\beta})$ as defined in (13) after replacing $\mathbf{a}_m$ by $\tilde{\mathbf{a}}_m = \boldsymbol{\beta}^T \mathbf{a}_m$ and the unity numerator inside the $\log^+$ function by β_ℓ^2 in order to allow for asymmetric rates.

$$r_\ell = \begin{cases} \max\limits_{\mathbf{a},\mathbf{b}} \mathcal{R}_\ell(\mathbf{H}, \mathbf{a}, \mathbf{b}, \boldsymbol{\beta}) & \text{if } \mathbf{a} \perp \mathbf{b} \\ 0 & \text{o.w.} \end{cases} \tag{38}$$

with $\mathcal{R}_\ell(\mathbf{H}, \mathbf{a}, \mathbf{b}, \boldsymbol{\beta})$ as defined in Theorem 3.

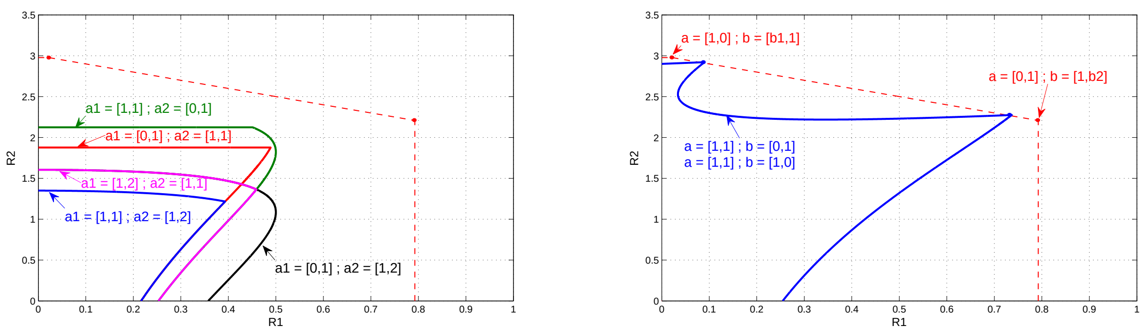

Fig. 1: Asymmetric rates for 2-user SIMO multiple-access channel with $P = 1$ and $\mathbf{H} = \left[\begin{smallmatrix} 1 & 5 \\ 1 & 6 \end{smallmatrix}\right]$ achievable by adjusting β_1 and β_2 with respect to the indicated linear functions for the regular compute-and-forward (left) and receiver combining compute-and-forward (right) respectively.

The best five combinations, in terms of sum rate, of decoded linear equations for the regular compute-and-forward are shown in Figure 1 (left), where the highest achievable rates correspond to equations shown in green and black. Note that the common boundary regions correspond to combinations with common decoded equation. The performance of receiver combining compute-and-forward (right) outperforms that of the regular scheme. Furthermore, some points on the boundary of the capacity region, other than the corner points achievable in classical approach by successive cancellation, are achieved with the receiver combining compute-and-forward.

V. Summary and Future Studies

In this work, we have investigated the compute-and-forward strategy applied to SIMO channels. We were able to provide a generalized receiver combining scheme, where different receivers can cooperate by combining different channel outputs. The equivalent channel output can be tuned in a way that makes it more suitable to decode multiple linear equations than what each receiver is able to decode individually. After applying the proposed scheme to 2-user SIMO MAC channel, we showed that our scheme outperforms the regular compute-and-forward. Furthermore, the proposed scheme is able to achieve some points on the boundary of capacity region using single point decoding and without time sharing or joint decoding used in the classical communication schemes. Future work might consider the analytical proof of our empirical results.

References

[1] U. Erez and R. Zamir, "Achieving $\frac{1}{2}\log(1+\text{SNR})$ on the AWGN channel with lattice encoding and decoding," *IEEE Transactions on Information Theory*, vol. 50, pp. 2293–2314, October 2004.

[2] U. Erez, S. Litsyn, and R. Zamir, "Lattices which are good for (almost) everything," *IEEE Transactions on Information Theory*, vol. 51, pp. 3401–3416, October 2005.

[3] G. Bresler, A. Parekh, and D. N. C. Tse, "The approximate capacity of the many-to-one and one-to-many gaussian interference channels," *IEEE Transactions on Information Theory*, vol. 56, no. 9, pp. 4566–4592, September 2010.

[4] B. Nazer and M. Gastpar, "Compute-and-forward: Harnessing interference through structured codes," *IEEE Transactions on Information Theory*, vol. 57, pp. 6463–6486, October 2011.

[5] B. Nazer, "Successive compute-and-forward," in *Proceedings of the International Zurich Seminar on Communications (IZS 2012)*, Zurich, Switzerland, March 2012, pp. 103–106.

[6] J. Zhu and M. Gastpar, "Asymmetric compute-and-forward with CSIT," 2013.

[7] J. Zhan, U. E. B. Nazer, and M. Gastpar, "Integer-forcing linear receivers," *Revised for the IEEE Transactions on Information Theory*, January 2013.

[8] S.-N. Hong and G. Caire, "Reverse compute and forward: A low-complexity architecture for downlink distributed antenna systems," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT 2012)*, Cambridge, MA, USA, July 2012, pp. 1147–1151.

[9] ——, "Structured lattice codes for $2 \times 2 \times 2$ MIMO interference channel," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT 2013)*, Istanbul, Turkey, July 2013, pp. 2229–2233.

[10] Y. Song, N. Devroye, and B. Nazer, "Inverse compute-and-forward: Extracting messages from simultaneously transmitted equations," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT 2011)*, St. Petersburg, Russia, August 2011, pp. 415–419.

[11] J. Zhu and M. Gastpar, "Lattice codes for many-to-one cognitive interference networks," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT 2013)*, Istanbul, Turkey, July 2013, pp. 2234–2238.