EURASIP Journal on Wireless Communications and Networking

RESEARCH

Open Access

A low-complexity channel training method for efficient SVD beamforming over MIMO channels

Felipe Kettlun[1,2], Fernando Rosas[3] and Christian Oberli[1,4*]

*Correspondence: obe@uc.cl
[4] National Research Center for Integrated Natural Disaster Management, Pontificia Universidad Católica de Chile, Santiago, Chile
Full list of author information is available at the end of the article

Abstract

Singular value decomposition (SVD) beamforming is an attractive tool for reducing the energy consumption of data transmissions in wireless sensor networks whose nodes are equipped with multiple antennas. However, this method is often not practical due to two important shortcomings: it requires channel state information at the transmitter and the computation of the SVD of the channel matrix is generally too complex. To deal with these issues, we propose a method for establishing an SVD beamforming link without requiring feedback of actual channel or SVD coefficients to the transmitter. Concretely, our method takes advantage of channel reciprocity and a power iteration algorithm (PIA) for determining the precoding and decoding singular vectors from received preamble sequences. A low-complexity version that performs no iterations is proposed and shown to have a signal-to-noise-ratio (SNR) loss within 1 dB of the bit error rate of SVD beamforming with least squares channel estimates. The low-complexity method significantly outperforms maximum ratio combining diversity and Alamouti coding. We also show that the computational cost of the proposed PIA-based method is less than the one of using the Golub–Reinsch algorithm for obtaining the SVD. The number of computations of the low-complexity version is an order of magnitude smaller than with Golub–Reinsch. This difference grows further with antenna array size.

1 Introduction

Wireless sensor networks (WSNs) are groups of spatially distributed communication nodes capable of sensing environmental variables (e.g., humidity, temperature, irradiation) for applications that usually require low data rates. Also, they tend to cover large and possibly remote areas (e.g., forests and mountains), which imposes the need for low-power and lower-complexity device implementations. This design principle imposes severe limitations on the radiated power, because electromagnetic radiation is the main source of energy consumption for WSN nodes [1]. Such limited radiated power, in turn, restricts the range of communication of each node.

The use of multiple-input multiple-output (MIMO) techniques for increasing the energy efficiency of WSN has started to receive attention from the scientific community. In particular, the diversity gain enabled by MIMO systems can be used for improving the reliability of the wireless link, reducing the outage probabilities and boosting the overall energy budget of WSN nodes [2–4]. These works show that if channel state information (CSI) is available at both ends of the link, an optimal symbol error rate (SER) is attained by employing the SVD-beamforming method [5]. SVD beamforming consists of using the strongest singular value decomposition (SVD) eigenmode of the MIMO channel [6]. This is implemented by employing the principal right and principal left singular vectors of the MIMO channel matrix as beamforming weights at the transmitter and receiver, respectively.

A key limitation of SVD beamforming is that channel state information (CSI) is required at the transmitter (CSIT). In frequency division duplexing (FDD) systems, CSI has to be computed in the receiver and then sent back to the transmitter, introducing an additional burden to the data traffic. To address this issue, limited feedback techniques have been proposed, whereby the receiver selects the beamforming vectors from a predefined finite and indexed set [7–10]. Thereafter, only the index of the precoding vector that best matches the channel in effect must be signaled back to the transmitter. An important drawback of this technique is that the data feedback must be performed *prior* to having the beamforming signal-to-noise-ratio (SNR) gain available across the link, making this approach impractical for low SNR scenarios.

In time division duplexing (TDD) systems, channel coefficients are estimated at both sides of the link using training signals in both directions and by exploiting the reciprocity of the wireless channel [11]. Respective SVDs may then be calculated by both devices from the channel estimates.

Another difficulty of the SVD-beamforming scheme is that obtaining the SVD of the channel matrix is computationally costly. For general applications, the Golub–Reinsch algorithm (GRA) [12] is the most utilized method for calculating the SVD because of its numerical stability, reduced computational cost and acceptable convergence speed [13]. While much research has been done trying to find ways to reduce the complexity of the SVD computation [14, 15], existent solutions are still inadequate for implementation in systems with a restricted energy budget and fixed-point computation constraints.

A family of TDD algorithms that require neither channel estimates nor SVD calculations have been explored in [16–20] and provide a way around the above-mentioned difficulties. These methods are based on the power iteration algorithm (PIA) [21] and require several back-and-forth transmissions before achieving a channel estimate good enough for reliable communication. One of the first of these algorithms is proposed in [17], in which an arbitrary symbol precoded with a unit vector is sent from the source. Then, only through normalization, conjugation and retransmissions of the received signals, the SVD-based beamforming link is established. A blind iterative MIMO algorithm (BIMA) is proposed in [18], which unlike [17] does not require a training stage. The precoding and decoding vectors are determined using payload data and are continuously updated while used at the same time for communication. The drawback of the algorithms of [17, 18] is their slow convergence (i.e., higher error rate at the beginning of packet transmission) and their poor performance in low SNR scenarios [20]. To improve performance at low SNRs, [20] extends BIMA with an adaptive algorithm, which estimates the principal singular vectors at both sides using a weighted sum of previous estimates and the current received signal. This reduces the detrimental effect of the thermal noise significantly, but the convergence of the algorithm is still slow. Hence, the cited methods are still inadequate for packet-based transmissions and energy-constrained devices.

Our contribution in this work is a method for establishing SVD beamforming by means of the power iteration principle. In contrast to the prior art, however, the proposed method does not realize its power iterations by repeated transmissions over the air, and it uses instead single transmissions of preambles followed by local iterative computations at the receiver. In addition to the energy and time savings obtained this way, an additional trade-off between energy consumption of computations versus quality of the resulting beamforming weights and, consequently, versus bit error rate (BER) performance can be exploited by varying the number of computational iterations. Improving the quality of the beamforming vectors does not require more transmissions over the air, just more computation at each transceiver. For the special case in which only one iteration of the PIA is performed, a reduced-complexity formulation of the method is devised.

After describing the proposed method and modeling it mathematically, we assess its computational complexity and its BER performance. The computational costs of the proposed method and of the popular Golub–Reinsch algorithm (GRA) for performing SVD are determined and compared for different antenna array sizes in terms of number of arithmetic operations. It is shown that the computational cost of the proposed method is less than for GRA in all cases of practical interest. The cost of the reduced-complexity version is an order of magnitude smaller than for GRA.

The BER performance is compared to well-known multiple-antenna diversity techniques, including maximum ratio combining and Alamouti coding. It is shown that both are outperformed by a significant SNR margin, even by the proposed reduced-complexity version. For antenna array sizes up to 64, the reduced-complexity version is shown to attain a BER with an SNR loss smaller than 1 dB with respect to SVD beamforming based on least squares channel estimates and perfect SVD computation by the GRA, while requiring an order of magnitude fewer computations.

The rest of the paper is organized as follows: In Sect. 2, we briefly present the MIMO signal model and SVD-BF in order to establish the nomenclature used. Section 3 presents the structure of the transmissions over the air used by the method and explain the calculations that the devices at each end of the link have to conduct. The performance of the proposed technique is quantified by Monte Carlo simulations in Sect. 4. Finally, Sect. 5 summarizes the main conclusions.

2 System model

This section introduces the MIMO signal model and the SVD-based beamforming scheme for a system with N_t transmit antennas and N_r receive antennas. The signal at the receiver can be modeled as

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

where $\mathbf{y} \in \mathbb{C}^{N_\mathrm{r}}$ is the column vector of received symbols at the N_r antennas of destination device Ω_2, $\mathbf{H} \in \mathbb{C}^{N_\mathrm{r} \times N_\mathrm{t}}$ is the MIMO channel matrix of coefficients h_{ij} that represent the complex fading gains from transmit antenna j to receive antenna i, column vector

$\mathbf{x} \in \mathbb{C}^{N_\mathrm{t}}$ represents the baseband-equivalent complex training or data symbols transmitted by the N_t antennas of source device Ω_1, and $\mathbf{n} \in \mathbb{C}^{N_\mathrm{r}}$ is a column vector of complex additive white Gaussian noise (AWGN) with i.i.d. elements with zero mean and ν^2 variance. Throughout this work, the elements h_{ij} are assumed to be i.i.d. circularly symmetric complex Gaussian random variables with zero mean and unit variance. In order to normalize the radiated power, the restriction $||\mathbf{x}|| = 1$ is imposed, where $|| \cdot ||$ denotes the Euclidean norm.

The SVD theorem [22] states that any matrix $\mathbf{H}$ can be factored as

$$\mathbf{H} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\dagger, \tag{2}$$

where $(\cdot)^\dagger$ denotes the conjugate transpose operator. The matrices $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{N_\mathrm{r}}] \in \mathbb{C}^{N_\mathrm{r} \times N_\mathrm{r}}$ and $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{N_\mathrm{t}}] \in \mathbb{C}^{N_\mathrm{t} \times N_\mathrm{t}}$ are unitary, i.e., $\mathbf{U}\mathbf{U}^\dagger = \mathbf{I}_{N_\mathrm{r}}$ and $\mathbf{V}\mathbf{V}^\dagger = \mathbf{I}_{N_\mathrm{t}}$, where $\mathbf{I}_N$ is the identity matrix of size $N \times N$. The left and right singular vectors $\mathbf{u}_k$ and $\mathbf{v}_k$, respectively, are not unique, because $\{e^{\jmath\theta}\mathbf{u}_k\}_{k=1}^{N_\mathrm{r}}$ and $\{e^{\jmath\theta}\mathbf{v}_k\}_{k=1}^{N_\mathrm{t}}$, with an arbitrary angle θ and $\jmath$ defined as $\sqrt{-1}$, are also valid singular vectors for $\mathbf{H}$. The matrix Σ is an $N_\mathrm{r} \times N_\mathrm{t}$ diagonal matrix of nonnegative real numbers σ_k, known as the singular values. These terms can be ordered such that $\sigma_1 \geq \sigma_2 \cdots \geq \sigma_{\min(N_\mathrm{t}, N_\mathrm{r})}$, where $\mathrm{rank}(\mathbf{H}) \leq \min(N_\mathrm{t}, N_\mathrm{r})$ of these singular values are nonzero [6].

If CSI is available at both transmitter and receiver, then by using an SVD precoding, the MIMO channel can be decomposed into rank($\mathbf{H}$) parallel data streams commonly known as eigenchannels. The various eigenchannels have different statistical properties: the strong ones are useful when diversity is needed, while the weak ones can be used for increasing throughput [5]. The highest diversity gain is obtained by transmitting data only over the strongest eigenchannel, which is known in the literature as SVD-BF. The corresponding transmission scheme consists of using the first right singular vector $\mathbf{v}_1$ to precode a scalar payload data symbol $d \in \mathbb{C}$, which is then decoded at the receiver with the conjugate transpose of the first left singular vector $\mathbf{u}_1^\dagger$. The resulting communication can be modeled as

$$\begin{aligned} \tilde{y} &= \mathbf{u}_1^\dagger \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\dagger \mathbf{v}_1 d + \mathbf{u}_1^\dagger \mathbf{n} \\ &= \sigma_1 d + \tilde{n}, \end{aligned} \tag{3}$$

where $\tilde{n}$ is a scalar of complex AWGN with zero mean and variance ν^2, $\tilde{y}$ is the received symbol d under equivalent thermal noise $\tilde{n}$ and channel gain σ_1. It is to be noted that the statistics of σ_1 can be well approximated using the Nakagami-m channel fading model [5].

The main difficulty of this technique is to obtain CSI at both sides of the link, particularly at the source device, and determine the first singular vectors from the channel matrix $\mathbf{H}$.

3 Proposed method

In this section, we present a detailed method for establishing an SVD-BF link between two nodes Ω_1 (source) and Ω_2 (destination) in an environment where channel reciprocity between forward (source to destination) and backward (destination to source) transmissions can be assumed. Hence, if the signals in both directions use the same frequency carrier and bandwidth, as in TDD systems, then the channel response is the same [23]. Formally, for MIMO systems, a channel $\mathbf{H}$ in the forward direction has a reciprocal channel $\mathbf{H}^{\mathrm{T}}$ in the backward direction, where $(\cdot)^{\mathrm{T}}$ denotes the transpose operator.

Even though non-symmetric characteristics of the RF electronic circuitry break the channel reciprocity, various solutions to that problem are available, either hardware-based or based on calibration algorithms [24, 25]. As addressing this aspect is beyond the scope of this work, we assume that devices Ω_1 and Ω_2 are properly calibrated so that channel reciprocity can be assumed. We also assume perfect packet detection and timing acquisition for all the transmissions. It has been shown that these tasks can be performed with the same preamble structure used here for channel estimation [26].

In the sequel, we first describe the method and its various steps, followed by a detailed description of each one. Then, an algorithm for obtaining the first singular vector from the channel matrix estimate is provided. And finally, we present a computational cost analysis of Golub–Reinsch algorithm, the most common technique for obtaining SVD, for comparing it with the simplified method that we propose.

3.1 Conceptual description of the method

The technique for establishing an SVD-BF link entails two types of transmissions: *Ping* and *Pong* (Fig. 1). The *Ping* consists of transmitting a known time-orthogonal preamble from Ω_1 to Ω_2, which allows for estimating the first left singular vector $\mathbf{u}_1$ at Ω_2. This type of transmission does not contain payload data. After the *Ping*, an arbitrary number of *Pong*s containing preamble and payload may be sent alternatingly in both directions. The first *Pong* is a transmission from Ω_2 to Ω_1 composed of a preamble and payload data that are precoded at Ω_2 with the left singular vector. The preamble thus received by Ω_1 enables it to estimate the first right singular vector $\mathbf{v}_1$. Ω_1 then replies to Ω_2 with a next (second) *Pong*, which has the same structure as the first *Pong* (preamble followed by payload data), but is precoded with $\mathbf{v}_1$. The method is described with mathematical formality in Sect. 3.2.

The method might be used for two-way communications, because *Pong*s may carry payload data in both directions. However, for simplicity of description we present only a one-way communication scheme because the bidirectional case is a straightforward extension. In particular, we present the case when the communication is initiated by a node Ω_1 that has information that it wishes to communicate to a neighboring node Ω_2.

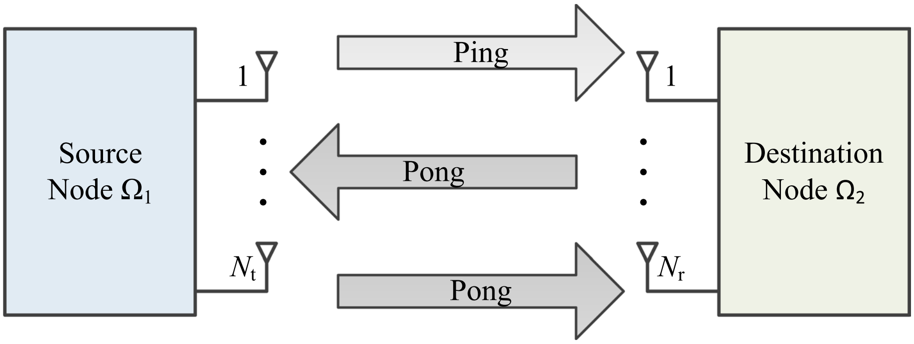

Fig. 1 Representation of the *Ping–Pong–Pong* scheme between source node Ω_1 and destination node Ω_2

This communication situation requires at least three transmissions: *Ping–Pong–Pong* (Fig. 1). It is to be noted that if the communication is initiated by a node that queries a neighboring node to find out if it has information to communicate, then the communication could be achieved with only two transmissions: *Ping–Pong*.

We focus on the case when the mobility of the environment is slow enough so that the coherence time of the channel is longer than the time required for a *Ping–Pong–Pong* transmission. In general, two-way SVD-BF communications could be maintained for longer than the coherence time of the channel if new *Pong* transmissions are made between both nodes more frequently than the coherence time. Furthermore, the re-estimations of singular vectors could be weighed with previous estimates as proposed in [20].

While the proposed method allows for calculating the first singular vectors on both sides of the link, it does not provide the first singular value σ_1. However, as can be seen in (3), the knowledge of σ_1 is only necessary for decoding the data if the communication system uses amplitude modulation, such as quadrature amplitude modulation (QAM) or amplitude-shift keying (ASK). σ_1 may be estimated in several ways, such as by embedding further pilot symbols in the transmissions. Alternatively, in order not to increase the complexity or transmission overhead of the scheme, only phase modulations, such as quadrature phase-shift keying (QPSK), may be used. This is of particular interest for long-distance transmissions using SVD-BF, because it is more energy efficient to use small modulation sizes for these cases [27].

3.2 Mathematical formulation of the method

In the sequel, we describe the *Ping* and *Pong* transmissions in detail.

3.2.1 Ping

The *Ping* consists of sending a known preamble of complex symbols from node Ω_1 to node Ω_2. The preamble is represented by an $N_\mathrm{t} \times L_1$ matrix $\mathbf{X}_1$, whose rows contain the symbol sequences for each transmit antenna, and its columns index symbol time. Thus, L_1 is the duration of the *Ping* preamble in terms of symbol times. Even though the matrix can be composed by arbitrary sequences of symbols, for computation efficiency at the receiver it is best composed in a staggered form with L_1/N_t training symbols for each antenna [28]. We assume that they are taken from a column vector $\mathbf{c}_1$ of L_1 known training symbols.

The received *Ping* is therefore

$$\mathbf{Y}_1 = \mathbf{H}\mathbf{X}_1 + \mathbf{N}_1, \tag{4}$$

where $\mathbf{N}_1 \in \mathbb{C}^{N_\mathrm{r} \times L_1}$ is the complex matrix of AWGN at receiver Ω_2 during the *Ping* reception.

Upon reception, channel estimation is performed at the destination node Ω_2 using $\mathbf{Y}_1$. We present our work based on the least square (LS) channel estimator due to its simplicity and limited computational complexity [28], but any other suitable estimator may be used. The LS estimate of $\mathbf{H}$ at Ω_2 is given by

$$\hat{\mathbf{H}} = \mathbf{Y}_1 \mathbf{X}_1^{\dagger} \left(\mathbf{X}_1 \mathbf{X}_1^{\dagger} \right)^{-1}, \tag{5}$$

where $\mathbf{X}_1^{\dagger} \left(\mathbf{X}_1 \mathbf{X}_1^{\dagger} \right)^{-1}$ is the Moore–Penrose right pseudo-inverse of $\mathbf{X}_1$. It is to be noted that this pseudo-inverse matrix can be precomputed and stored permanently at Ω_2, so that only the matrix multiplication between $\mathbf{Y}_1$ and the stored pseudo-inverse of $\mathbf{X}_1$ is required with each *Ping*. An estimate $\hat{\mathbf{u}}_1$ of the first left singular column vector $\mathbf{u}_1$ can be extracted from $\hat{\mathbf{H}}$ using a power iteration algorithm. This step is explained later in Sect. 3.3. We assume that the estimation error in $\hat{\mathbf{u}}_1$ is an additive term $\mathbf{r}_\mathrm{u} \in \mathbb{C}^{N_\mathrm{t}}$ such that $\hat{\mathbf{u}}_1 = \mathbf{u}_1 + \mathbf{r}_\mathrm{u}$.

3.2.2 Pong in the backward direction

Using the estimate $\hat{\mathbf{u}}_1$, Ω_2 transmits the matrix $\mathbf{X}_2 = \hat{\mathbf{u}}_1^* \left[\mathbf{c}_2^\mathrm{T} \mathbf{d}_2^\mathrm{T} \right]$ to Ω_1, where $(\cdot)^*$ denotes the complex conjugation, $\mathbf{c}_2 \in \mathbb{C}^{L_2}$ is a column vector whose elements are a known preamble sequence of length L_2 symbols, $\mathbf{d}_2 \in \mathbb{C}^{D_2}$ is payload data column vector of length D_2 symbols ($D_2 \geq 0$), and $\left[\mathbf{c}_2^\mathrm{T} \mathbf{d}_2^\mathrm{T} \right]$ is the concatenation of row vectors $\mathbf{c}_2^\mathrm{T}$ and $\mathbf{d}_2^\mathrm{T}$. Considering that the reverse channel is $\mathbf{H}^\mathrm{T}$ [23], this reverse-channel transmission can be modeled as

$$\begin{aligned}
\mathbf{Y}_2 &= \mathbf{H}^\mathrm{T} \mathbf{X}_2 + \mathbf{N}_2 \\
&= \mathbf{H}^\mathrm{T} \hat{\mathbf{u}}_1^* \left[\mathbf{c}_2^\mathrm{T} \mathbf{d}_2^\mathrm{T} \right] + \mathbf{N}_2 \\
&= \mathbf{H}^\mathrm{T} \left(\mathbf{u}_1^* + \mathbf{r}_\mathrm{u}^* \right) \left[\mathbf{c}_2^\mathrm{T} \mathbf{d}_2^\mathrm{T} \right] + \mathbf{N}_2 \\
&= \mathbf{V}^* \boldsymbol{\Sigma}^\mathrm{T} \mathbf{U}^\mathrm{T} \mathbf{u}_1^* \left[\mathbf{c}_2^\mathrm{T} \mathbf{d}_2^\mathrm{T} \right] + \mathbf{H}^\mathrm{T} \mathbf{r}_\mathrm{u}^* \left[\mathbf{c}_2^\mathrm{T} \mathbf{d}_2^\mathrm{T} \right] + \mathbf{N}_2 \\
&= \sigma_1 \mathbf{v}_1^* \left[\mathbf{c}_2^\mathrm{T} \mathbf{d}_2^\mathrm{T} \right] + \mathbf{H}^\mathrm{T} \mathbf{r}_\mathrm{u}^* \left[\mathbf{c}_2^\mathrm{T} \mathbf{d}_2^\mathrm{T} \right] + \mathbf{N}_2,
\end{aligned} \tag{6}$$

where $\mathbf{N}_2 \in \mathbb{C}^{N_\mathrm{t} \times (L_2 + D_2)}$ is the AWGN matrix at the receiver Ω_1 during the *Pong* reception.

An estimate of the first right singular vector $\mathbf{v}_1$ can be obtained at the source Ω_1 using LS estimation from preamble $\mathbf{c}_2$ as follows:

$$\hat{\mathbf{v}}_1 = \frac{\mathbf{Y}_{2\mathrm{c}}^* \mathbf{c}_2 \left(\mathbf{c}_2^\dagger \mathbf{c}_2 \right)^{-1}}{\left| \left| \mathbf{Y}_{2\mathrm{c}}^* \mathbf{c}_2 \left(\mathbf{c}_2^\dagger \mathbf{c}_2 \right)^{-1} \right| \right|}, \tag{7}$$

where $\mathbf{Y}_{2\mathrm{c}}$ is the portion of the received signal $\mathbf{Y}_2$ that corresponds to preamble $\mathbf{c}_2$, and column vector $\mathbf{c}_2 (\mathbf{c}_2^\dagger \mathbf{c}_2)^{-1}$ of size L_2 is the pseudo-inverse of $\mathbf{c}_2^\dagger$. As before, this vector can be precomputed and stored on each device beforehand. Hence, the calculation of (7) takes one matrix multiplication and one vector normalization.

In case that the backward *Pong* carries payload data, node Ω_1 can decode it now using $\hat{\mathbf{v}}_1$ as follows:

$$\begin{aligned}\mathbf{y}_{2\mathrm{d}}^{\mathrm{T}} &= \hat{\mathbf{v}_1^{\mathrm{T}}}\mathbf{Y}_{2\mathrm{d}} \\ &= \left(\mathbf{v}_1^{\mathrm{T}} + \mathbf{r}_{\mathrm{v}}^{\mathrm{T}}\right)\left(\sigma_1\mathbf{v}_1^*\mathbf{d}_2^{\mathrm{T}} + \mathbf{H}^{\mathrm{T}}\mathbf{r}_{\mathrm{u}}^*\mathbf{d}_2^{\mathrm{T}} + \mathbf{N}_{2\mathrm{d}}\right) \\ &= \sigma_1\mathbf{d}_2^{\mathrm{T}} + \mathbf{v}_1^{\mathrm{T}}\mathbf{H}^{\mathrm{T}}\mathbf{r}_{\mathrm{u}}^*\mathbf{d}_2^{\mathrm{T}} + \sigma_1\mathbf{r}_{\mathrm{v}}^{\mathrm{T}}\mathbf{v}_1^*\mathbf{d}_2^{\mathrm{T}} \\ &\quad + \mathbf{r}_{\mathrm{v}^{\mathrm{T}}}\mathbf{H}^{\mathrm{T}}\mathbf{r}_{\mathrm{u}}^*\mathbf{d}_2^{\mathrm{T}} + \left(\mathbf{v}_1^{\mathrm{T}} + \mathbf{r}_{\mathrm{v}}^{\mathrm{T}}\right)\mathbf{N}_{2\mathrm{d}} \\ &= \sigma_1\mathbf{d}_2^{\mathrm{T}} + \left(\sigma_1\mathbf{u}_1^{\mathrm{T}}\mathbf{r}_{\mathrm{u}}^* + \sigma_1\mathbf{r}_{\mathrm{v}}^{\mathrm{T}}\mathbf{v}_1^* + \mathbf{r}_{\mathrm{v}}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}}\mathbf{r}_{\mathrm{u}}^*\right)\mathbf{d}_2^{\mathrm{T}} + \mathbf{n}_{2\mathrm{d}}^{\mathrm{T}},\end{aligned} \tag{8}$$

where $\mathbf{Y}_{2\mathrm{d}}$ and $\mathbf{N}_{2\mathrm{d}}$ correspond to the parts of the received signal $\mathbf{Y}_2$ and thermal noise $\mathbf{N}_2$, respectively, that are associated with payload data $\mathbf{d}_2$. Vector $\mathbf{r}_{\mathrm{v}}$ is the estimation error of the first right singular vector $\mathbf{v}_1$ and $\mathbf{n}_{2\mathrm{d}}$ is the respective AWGN vector with i.i.d. zero mean and ν^2 variance elements. It is to be noted that if estimation errors $\mathbf{r}_{\mathrm{u}}$ and $\mathbf{r}_{\mathrm{v}}$ tend to zero, then (8) tends to $\sigma_1\mathbf{d}_2^{\mathrm{T}} + \mathbf{n}_{2\mathrm{d}}^{\mathrm{T}}$, which corresponds to the vector form of (3) when several symbols are transmitted.

3.2.3 Pong in the forward direction

In case node Ω_1 has payload data for node Ω_2, it transmits $\mathbf{X}_3 = \hat{\mathbf{v}}_1\left[\mathbf{c}_3^{\mathrm{T}}\mathbf{d}_3^{\mathrm{T}}\right]$, where $\mathbf{c}_3 \in \mathbb{C}^{L_3}$ is a column vector of a known preamble of length L_3 symbols and $\mathbf{d}_3 \in \mathbb{C}^{D_3}$ is the payload data column vector of length D_3 symbols. The received signal at Ω_2 is

$$\begin{aligned}\mathbf{Y}_3 &= \mathbf{H}\mathbf{X}_3 + \mathbf{N}_3 \\ &= \sigma_1\mathbf{u}_1\left[\mathbf{c}_3^{\mathrm{T}}\mathbf{d}_3^{\mathrm{T}}\right] + \mathbf{H}\mathbf{r}_{\mathrm{v}}\left[\mathbf{c}_3^{\mathrm{T}}\mathbf{d}_3^{\mathrm{T}}\right] + \mathbf{N}_3,\end{aligned} \tag{9}$$

where $\mathbf{N}_3 \in \mathbb{C}^{N_{\mathrm{r}}\times(L_3+D_3)}$ is the AWGN matrix at the receiver Ω_2 during the *Pong* reception.

It is to be noted that transmitting preamble $\mathbf{c}_3$ at this stage is not strictly necessary for Ω_2 to be able to decode the received payload $\mathbf{d}_3$, because Ω_2 already has an estimate for $\mathbf{u}_1$, obtained during *Ping*. However, it may be convenient to transmit preamble $\mathbf{c}_3$ for improving the quality of the estimate $\hat{\mathbf{u}}_1$ obtained during the *Ping*, because the newly received signal on this first forward *Pong* has the advantage of having been transmitted over the best eigen-channel, thus enjoying higher SNR for a new or improved estimation of $\mathbf{u}_1$. Perhaps the simplest approach is to re-estimate $\mathbf{u}_1$ with LS as in (7):

$$\hat{\mathbf{u}}_1 = \frac{\mathbf{Y}_{3\mathrm{c}}\mathbf{c}_3^*\left(\mathbf{c}_3^{\mathrm{T}}\mathbf{c}_3^*\right)^{-1}}{\left|\left|\mathbf{Y}_{3\mathrm{c}}\mathbf{c}_2^*\left(\mathbf{c}_3^{\mathrm{T}}\mathbf{c}_3^*\right)^{-1}\right|\right|}, \tag{10}$$

where $\mathbf{Y}_{3\mathrm{c}}$ is the portion of the received signal $\mathbf{Y}_3$ that corresponds to preamble $\mathbf{c}_3$. Again, the pseudoinverse $\mathbf{c}_3^*\left(\mathbf{c}_3^{\mathrm{T}}\mathbf{c}_3^*\right)^{-1}$ can be precomputed and stored at Ω_2. The payload data is then decoded as

$$\begin{aligned}\mathbf{y}_{3\mathrm{d}}^{\mathrm{T}} &= \hat{\mathbf{u}}_1^{\dagger}\mathbf{Y}_{3\mathrm{d}} \\ &= \sigma_1\mathbf{d}_3^{\mathrm{T}} + \left(\sigma_1\mathbf{v}_1^{\dagger}\mathbf{r}_{\mathrm{v}} + \sigma_1\mathbf{r}_{\mathrm{u}}^{\dagger}\mathbf{u}_1 + \mathbf{r}_{\mathrm{u}}^{\dagger}\mathbf{H}\mathbf{r}_{\mathrm{v}}\right)\mathbf{d}_3^{\mathrm{T}} + \mathbf{n}_{3\mathrm{d}}^{\mathrm{T}},\end{aligned} \tag{11}$$

where $\mathbf{Y}_{3\mathrm{d}}$ and $\mathbf{N}_{3\mathrm{d}}$ correspond to the parts of the received signal $\mathbf{Y}_3$ and thermal noise $\mathbf{N}_3$, respectively, that are associated with payload data $\mathbf{d}_3$. $\mathbf{n}_{3\mathrm{d}}$ is the corresponding AWGN vector with i.i.d. zero mean and ν^2 variance elements.

A summary of all the steps that were described and that make up a complete *Ping–Pong–Pong* sequence is presented in Fig. 2.

3.3 Computation of the first singular vector

In the sequel, we describe how to estimate the first left singular vector $\mathbf{u}_1$ using a power iteration algorithm (PIA) on channel matrix estimate $\hat{\mathbf{H}}$ obtained from (5) after a *Ping* transmission.

The most popular algorithm for computing singular vectors, the Golub–Reinsch algorithm (GRA), as most of the SVD algorithms, calculates all left and right singular vectors together as a set. But we are only interested in calculating $\mathbf{u}_1$ at node Ω_2 after the *Ping*. The PIA [21] offers a suitable approach to this. We first summarize the general PIA and then provide a simplified one.

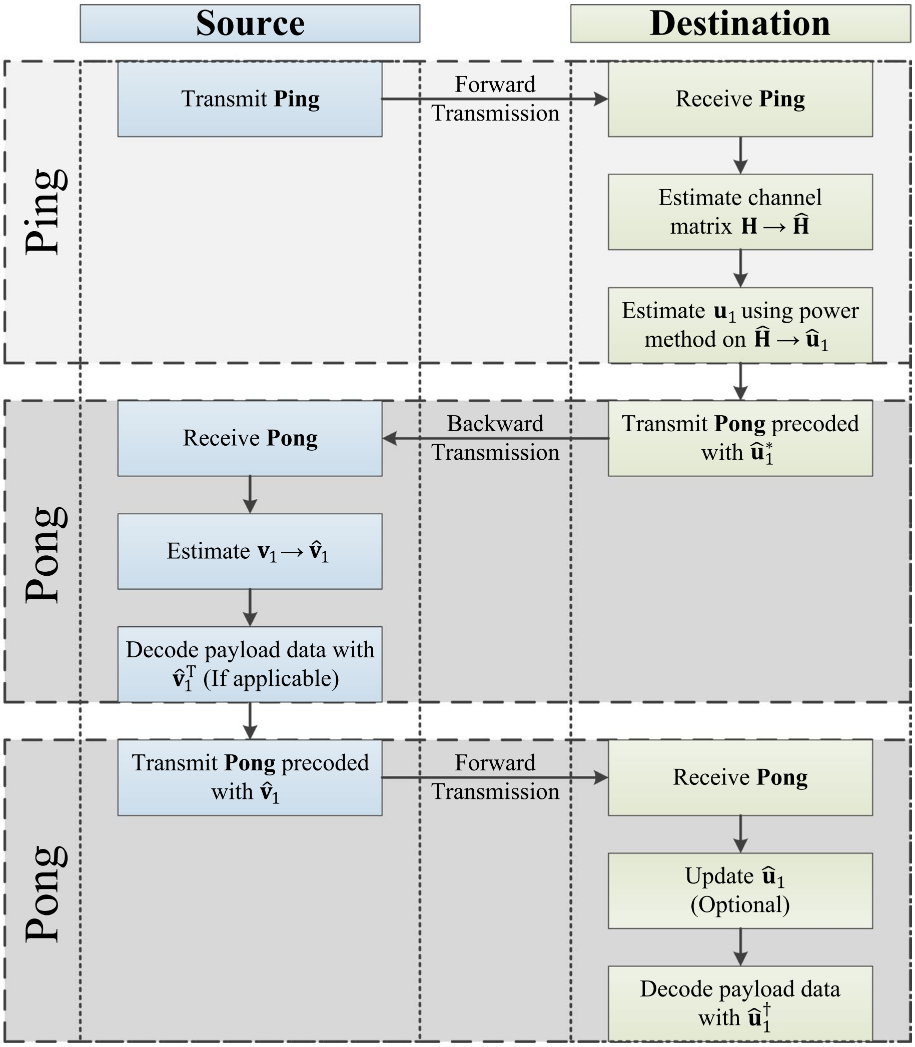
Source
Destination
Ping
Pong
Pong
Transmit **Ping**
Forward Transmission
Receive **Ping**
Estimate channel matrix $\mathbf{H} \rightarrow \hat{\mathbf{H}}$
Estimate $\mathbf{u}_1$ using power method on $\hat{\mathbf{H}} \rightarrow \hat{\mathbf{u}}_1$
Receive **Pong**
Backward Transmission
Transmit **Pong** precoded with $\hat{\mathbf{u}}_1^*$
Estimate $\mathbf{v}_1 \rightarrow \hat{\mathbf{v}}_1$
Decode payload data with $\hat{\mathbf{v}}_1^T$ (If applicable)
Transmit **Pong** precoded with $\hat{\mathbf{v}}_1$
Forward Transmission
Receive **Pong**
Update $\hat{\mathbf{u}}_1$ (Optional)
Decode payload data with $\hat{\mathbf{u}}_1^\dagger$

Fig. 2 Summary of the *Ping–Pong–Pong* steps in each of the two nodes involved in the communication. The left side (blue) shows the steps of a node operating as a source. The right side (green), shows the steps of a similar node operating as a destination

3.3.1 General PIA

The PIA allows for computing the first left singular vector $\mathbf{u}_1$ of a matrix $\mathbf{H}$ by exploiting the following property [21]:

$$\lim_{m\to\infty} \frac{\mathbf{W}^m \mathbf{q}_0}{||\mathbf{W}^m \mathbf{q}_0||} = \mathbf{u}_1, \tag{12}$$

where $\mathbf{W} = \mathbf{H}\mathbf{H}^\dagger$ is a Wishart matrix, $\mathbf{q}_0 \in \mathbb{C}^{N_\mathrm{r}}$ is a random vector with unit norm and exponent m is a positive integer. It is to be noted that an estimate of $\mathbf{u}_1$ can be defined as

$$\hat{\mathbf{u}}_1 = \frac{\hat{\mathbf{W}}^m \mathbf{q}_0}{||\hat{\mathbf{W}}^m \mathbf{q}_0||}, \tag{13}$$

where $\hat{\mathbf{W}} = \hat{\mathbf{H}}\hat{\mathbf{H}}^\dagger$, with $\hat{\mathbf{H}}$ given by (5) or any other suitable estimate.

Having a random initial vector $\mathbf{q}_0$ instead of a fixed vector gives no benefit when $\mathbf{u}_1$ is unknown, as is our case. Therefore, without loss of generality, we use $\mathbf{q}_0 \triangleq [10\cdots 0]^\mathrm{T}$.

We thus utilize the following version of the PIA for obtaining estimate $\hat{\mathbf{u}}_1$.

```
1: $\hat{\mathbf{W}} \leftarrow \hat{\mathbf{H}}\hat{\mathbf{H}}^\dagger$
2: $i \leftarrow 1$
3: while $i < m$ do
4:     $\mathbf{q}_i \leftarrow \hat{\mathbf{W}}\mathbf{q}_{i-1}$
5:     $i \leftarrow i + 1$
6: end while
7: $\hat{\mathbf{u}}_1 \leftarrow \frac{\mathbf{q}_m}{||\mathbf{q}_m||}$
```

The number of basic mathematical operations needed for each computational step of the algorithm is shown in Table 1.

3.3.2 Reduced-complexity power algorithm

For a lower-complexity algorithm, we can observe that in the special case when $m = 1$, the result of matrix multiplication of step 4, with $i = m = 1$, is

$$\begin{aligned} \mathbf{q}_1 &= \hat{\mathbf{W}}\mathbf{q}_0 \\ &= \hat{\mathbf{H}}\hat{\mathbf{H}}^\dagger \mathbf{q}_0 \\ &= \hat{\mathbf{H}}(\hat{\mathbf{H}}_{1,1:N_\mathrm{t}})^\dagger, \end{aligned} \tag{14}$$

where $\hat{\mathbf{H}}_{1,1:N_\mathrm{t}}$ denotes the first row of $\hat{\mathbf{H}}$. We can hence use the following reduced-complexity power algorithm (RCPA) for obtaining $\hat{\mathbf{u}}_1$.

Table 1 Computational operations power iteration algorithm

Line	Sums	Products	Divisions	Square roots
1	$2N_\mathrm{r}^2(2N_\mathrm{t} - 1)m$	$4N_\mathrm{t}N_\mathrm{r}^2 m$	0	0
4	$2N_\mathrm{r}(2N_r - 1)m$	$4N_r^2 m$	0	0
7	$2(N_\mathrm{r} - 1)m$	$2N_\mathrm{r}m$	$N_\mathrm{r}m$	m

```
1: $\mathbf{q}_1 \leftarrow \hat{\mathbf{H}}(\hat{\mathbf{H}}_{1,1:N_t})^\dagger$
2: $\hat{\mathbf{u}}_1 \leftarrow \frac{\mathbf{q}_1}{||\mathbf{q}_1||}$
```

The computational cost of the RCPA is smaller by roughly a factor mN_r compared to the general PIA, as shown in Table 2:

3.4 Computational cost using the Golub–Reinsch algorithm

A comparison of the reduction in complexity that PIA and RCPA provide over the GRA during the *Ping* stage is provided next.

In "Appendix", we present a study of the computational cost of the GRA. We find that the total cost of performing the SVD for an $N \times N$ matrix takes

$$C_{\text{GRA}} = \begin{cases} \frac{16}{3}N^3 + 10N^2 - \frac{28}{3}N + 10 & \text{sums} \\ \frac{16}{3}N^3 + 16N^2 - \frac{70}{3}N + 4 & \text{products} \\ 4N^2 - 2N - 3 & \text{divisions} \\ 2N^2 - 3 & \text{square roots} \\ 2N - 3 & \text{sign operations} \end{cases} . \tag{15}$$

Using the parameters of [29], we calculate the number of cycles that an arithmetic logic unit (ALU) requires for performing the decomposition of $\hat{\mathbf{H}}$ to obtain $\hat{\mathbf{u}}_1$ using the GRA, PIA and RCPA. Results show that RCPA provides clear reductions on the complexity with respect to the PIA and GRA (cf. Fig. 3). It will be shown in Sect. 4 that this complexity reduction does not significantly sacrifice bit error rate performance.

It is to be noted that when comparing the computational complexity in terms of ALU cycles per calculated singular vector element, the GRA does require fewer operations than the PIA. But the GRA does not allow for computing only the first singular vector alone and forces to compute the entire SVD each time, resulting in a larger net computational cost than for the PIA, as shown in Fig. 3. The RCPA, on the other hand, requires an order of magnitude fewer operations than the GRA in either case (per vector element and total).

4 Results and discussion

In this section, we provide simulative valuations of the *Ping–Pong–Pong* (PPP) method using the PIA and RCPA algorithms.

We performed simulations in which the elements of each realization of the channel matrix **H** were generated randomly for each run as i.i.d. circularly symmetric complex Gaussian random variables with zero mean and unit variance. Thermal noise samples

Table 2 Computational operations of reduced-complexity power algorithm

Line	Sums	Products	Divisions	Square roots
1	$2N_\text{r}(2N_\text{t} - 1)$	$4N_\text{t}N_\text{r}$	0	0
2	$2N_\text{r} - 1$	$2N_\text{r}$	N_r	1

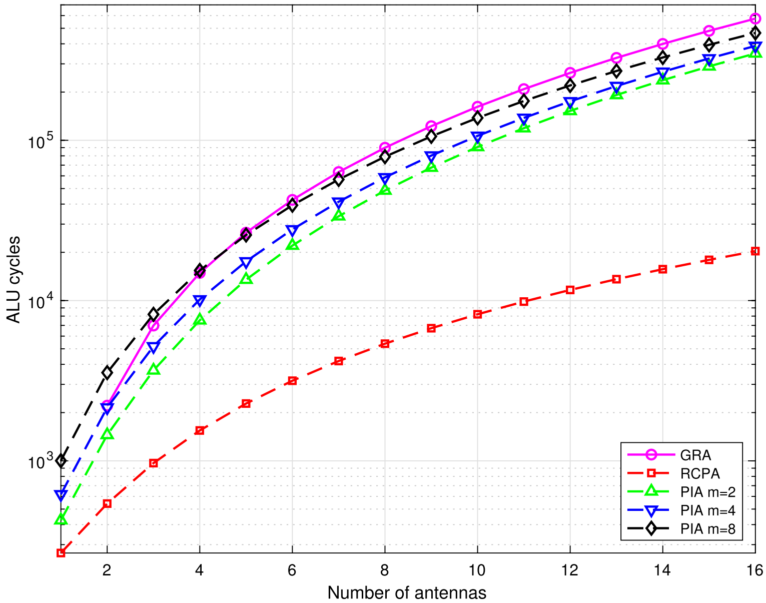

Fig. 3 ALU cycles needed to perform the Golub–Reinsch algorithm (GRA), the power iteration algorithm (PIA) with $m = 2, 4$ and 8 iterations and the reduced-complexity power algorithm (RCPA) for MIMO channels with equal number of transmit and receive antennas

were generated randomly as i.i.d. circularly symmetric complex Gaussian random variables with zero mean and variance ν^2.

Ping and *Pong* packets were assembled considering $L_1 = L_2 = 32$, $L_3 = 0$ and 500 symbols of payload data with an uncoded QPSK modulation. It is to be noted that it does not matter if the payload data is sent in the backward ($D_2 = 500$ QPSK symbols, $D_3 = 0$) or forward ($D_2 = 0$, $D_3 = 500$ QPSK symbols) *Pong* transmissions. Both cases are equivalent in terms of the BER of the payload data as long as there is no re-estimation of the respective singular vector, i.e., as long as $L_3 = 0$, which was always the case. Each PPP composed this way was transmitted over one million channel realizations.

The bit error rate (BER) performance of the proposed technique in 2 × 2 and 4 × 4 MIMO configurations is presented in Figs. 4 and 5, respectively. Both graphs also show the BER performance of a single-input single-output (SISO) channel with flat Rayleigh fading, of maximum ratio combining (MRC) receive diversity [6], of the iterative method presented by Tang [17] and of SVD beamforming with ideal channel knowledge and ideal SVD computation. In the case of 2 × 2 MIMO links, the BER performance of Alamouti coding is also presented [30]. To make a fair comparison, for all cases we considered the same total number of symbols used for channel training (considering both link directions) and the same total sum of signal power transmitted among all antenna branches. This means that in all cases the total energy spent for training transmissions is the same. All schemes used LS channel estimation. We observe for 2 × 2 that the SNR loss with respect to ideal SVD beamforming (curve SVD-BF-Ideal) is approximately 1 dB for the RCPA version (PIA with $m = 1$ iteration) and approximately 0.1 dB with $m = 4$ iterations (Fig. 4). In 4 × 4 the respective losses are approximately 2 dB and 0.7 dB (Fig. 5). It is also apparent that the proposed PPP method outperforms the BER of receive MRC, Alamouti and Tang in both MIMO configurations, even when the RCPA is used.

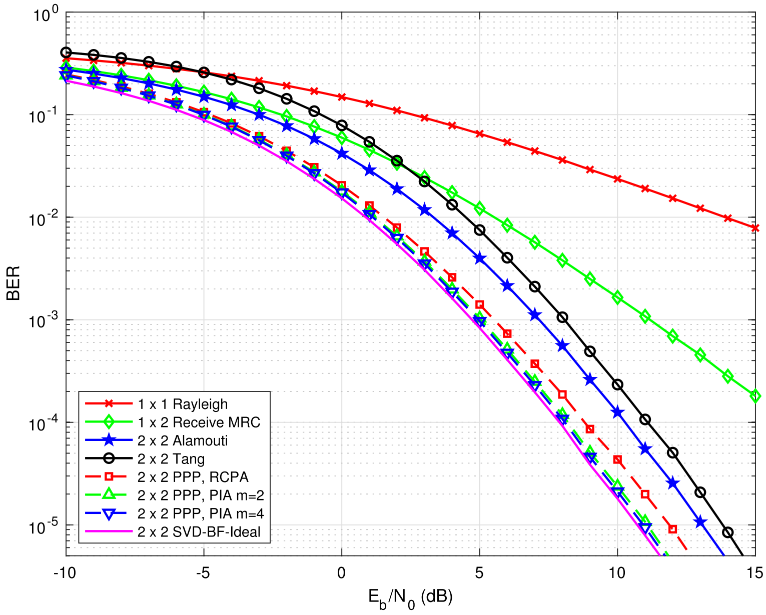

Fig. 4 Simulated BER of Rayleigh SISO (1 × 1), MRC receive diversity (1 × 2), Tang method [17] (2 × 2), Alamouti coding (2 × 2) and proposed PPP (2 × 2) schemes (RCPA with $m = 1$, PIA with $m = 2$ and $m = 4$ iterations) versus ideal SVD beamforming (SVD-BF-Ideal, 2 × 2) with ideal channel knowledge and ideal SVD computation

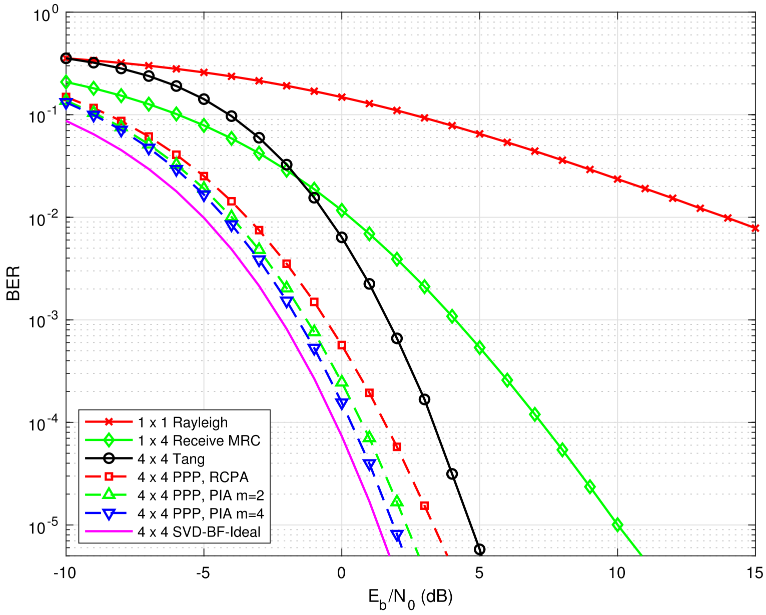

Fig. 5 Simulated BER of Rayleigh SISO (1 × 1), MRC receive diversity (1 × 4), Tang method [17] (4 × 4) and proposed PPP (4 × 4) schemes (RCPA with $m = 1$, PIA with $m = 2$ and $m = 4$ iterations) versus ideal SVD beamforming (SVD-BF-Ideal, 4 × 4) with ideal channel knowledge and ideal SVD computation

The above results suggest that the SNR loss grows with MIMO channel size. In order to explore this aspect, we performed additional BER simulations of MIMO constellations of sizes $8 \times 8, 16 \times 16, 32 \times 32$ and 64×64. The BER performance of SVD beamforming with least squares channel estimation and ideal SVD computation was also assessed by simulation. This performance provides a best-case performance reference for the proposed iterative method. Preambles with 64 symbols were used in all cases. The corresponding SNR losses with respect to ideal SVD beamforming (ideal channel knowledge and ideal SVD computation) at BER $= 10^{-3}$ are shown in Fig. 6. While the SNR loss of the proposed method clearly grows with antenna array size, even the worst-case performance of RCPA stays within 1 dB of the best-case performance given by SVD beamforming with LS channel estimation. With respect to this latter case, the performance loss of the proposed method with $m = 8$ iterations is negligible. The overall BER performance of the proposed method at 64 antennas ranges between 5 dB and 6 dB of SNR loss with respect to ideal SVD beamforming. This is smaller than the loss observed for MRC diversity even at 2 × 2 and 4 × 4 configurations (cf. Figs. 4, 5).

The impact of using the re-estimation of vector $\mathbf{u}_1$ at the forward (second) *Pong* stage, as given by (10), rather than using the $\hat{\mathbf{u}}_1$ estimated during the initial *Ping*, as presented in Sect. 3.2, is similar to performing an extra iteration of the PIA in the case without re-estimation (Fig. 7). These curves were generated using the same parameters as for Fig. 5. In the case when $m = 1$ (RCPA), the SNR improvement gained by the re-estimation can be as large as 1 dB.

The difference in BER between the PPP with $m = 4$ and the theoretical SVD-BF (with perfect CSI) is due to the channel estimation error. This aspect is evaluated in Fig. 8, where simulations with preambles of length $L_1 = L_2 = 4$, $L_1 = L_2 = 32$ and $L_1 = L_2 = 128$ symbols are compared for the case of 4 × 4 channels estimated according to (5). We used $L_3 = 0$ in all cases. As intuition suggests, as the preamble grows in

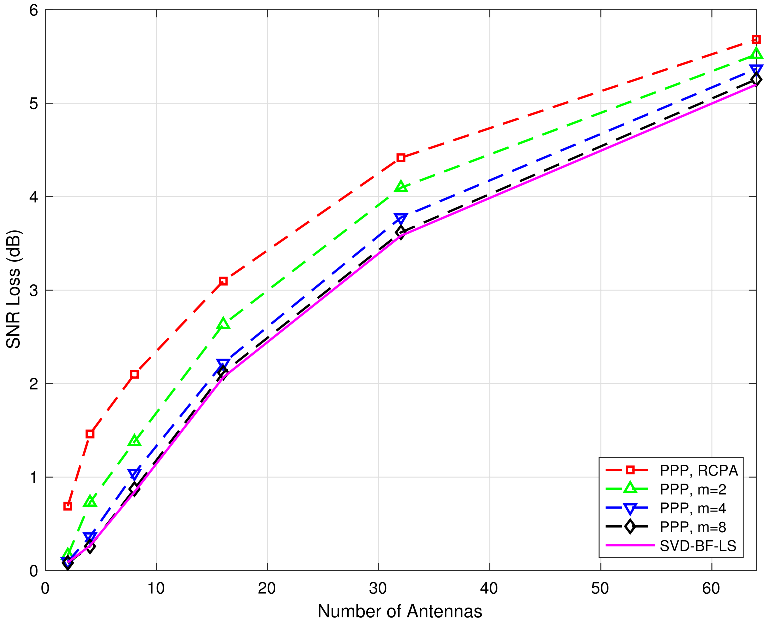

Fig. 6 SNR loss of the proposed PPP scheme (RCPA with $m = 1$, PIA with $m = 2$, $m = 4$ and $m = 8$ iterations) and of SVD beamforming with least squares channel estimation and ideal SVD computation (SVD-BF-LS) with respect to ideal SVD beamforming (ideal channel knowledge and ideal SVD computation) for MIMO channels with equal number of transmit and receive antennas

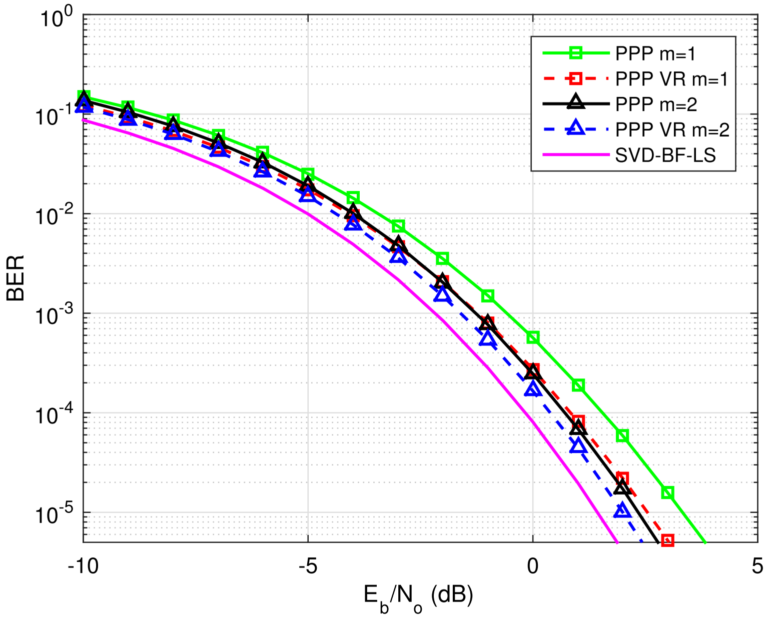

Fig. 7 Simulated BER for PPP with beamforming vector re-estimation (VR) at forward *Pong* stage compared to the *Ping–Pong–Pong* base case and the ideal SVD beamforming with LS-estimated channel (SVD-BF-LS) using a 4 × 4 MIMO array

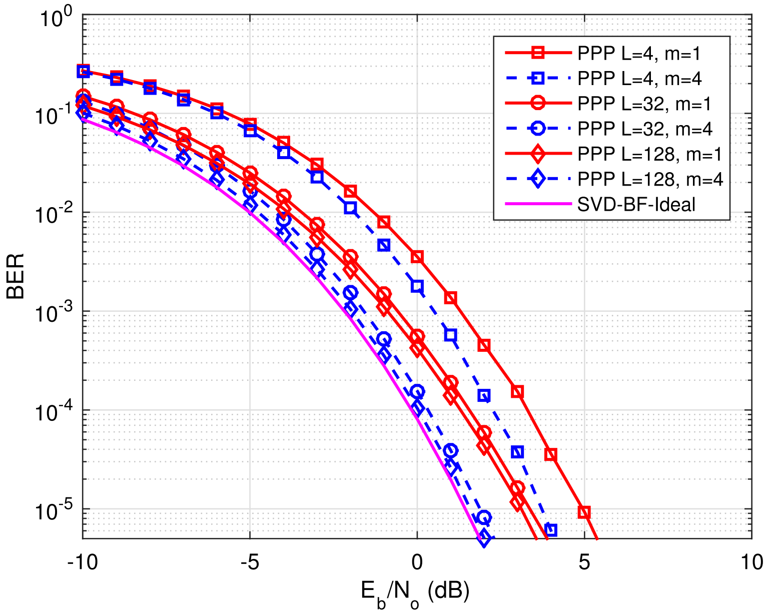

Fig. 8 BER comparison with $L_1 = L_2 = 4, L_1 = L_2 = 32$ and $L_1 = L_2 = 128$ symbols for channel estimation, and $m = 1$(RCPA) and $m = 4$ iterations of the PIA using a 4 × 4 MIMO array. They are compared to the theoretical SVD-BF BER with ideal channel knowledge

length, the BER approaches the theoretical SVD-BF curve. The low-complexity algorithm (RCPA) with $L_1 = L_2 = 4$ (worst-case performance) displays an SNR loss of approximately 4 dB with respect to ideal SVD-BF, is similar to the BER performance of

Tang and superior to that of MRC diversity (compare with Fig 5). While extending the preamble length has diminishing returns in terms of BER performance, it does not spoil the performance gained by varying the number of iterations of the PIA.

5 Conclusions

In this article, we propose a low-complexity method for establishing a communication link over MIMO channels using SVD-based beamforming. The method takes advantage of the channel reciprocity property in order to acquire estimates of the precoding and decoding first singular vectors at both ends of the wireless link. This is attained with two types of transmissions: an initial *Ping*, consisting of a space and time orthogonal preamble transmitted once, and *Pong*, a beamformed preamble followed by beamformed payload data. *Pong* can be transmitted an arbitrary number of times in both directions, thus allowing for one-way or two-way communications. After an initial beamforming vector estimation at the receiver of the *Ping*, the receiver of a *Pong* preamble estimates or re-estimates the singular vector that corresponds to that end of the link. This is performed with a power iteration algorithm.

Simulations show that four iterations suffice for attaining a BER within 1 dB of ideal SVD beamforming performance for MIMO array configurations of up to 4×4. With 4 antennas and only one iteration (reduced-complexity algorithm), the SNR loss is within 2 dB of the ideal singular vector computation, but the complexity of the algorithm requires an order of magnitude fewer computations. It is also shown that the proposed method outperforms the BER of maximum ratio combining and of Alamouti coding.

For arrays with 64 antennas, the method is shown to achieve a BER performance within 1 dB of that of SVD beamforming with least squares channel estimates and perfect SVD computation.

The use of the PIA for this task is also computationally more efficient than the Golub–Reinsch algorithm for the SVD, whose main limitation is that it does not allow for computing only the first singular vector alone and forces to compute the entire SVD each time.

The BER degradation due to imperfect channel estimation was shown to be within 4 dB of ideal performance for a worst-case configuration (shortest training preamble, reduced-complexity algorithm). Further simulations show that re-estimating the vector at the *Pong* has an effect similar to performing an extra iteration of the PIA. The SNR improvement gained by the re-estimation can be as large as 1 dB.

Appendix: SVD computation cost

The *Golub–Reinsch* algorithm (GRA) [12] is popular for performing the SVD decomposition because of its numerical stability, efficiency and good convergence velocity [31]. Following [13], this "Appendix" analyzes the computational cost of the GRA on a matrix $\mathbf{H}$ of size $N \times N$. The algorithm is composed of two phases: a bidiagonalization and a superdiagonal reduction.

In the following, we denote $\mathbf{A}_{j:n,k:n}$ as the submatrix of $\mathbf{A}$ that contains rows from j to n and columns from k to n of $\mathbf{A}$. Further, blank entries in a matrix represent zeros, while $\times$ or $+$ terms represent nonzero coefficients.

Phase I: Bidiagonalization

The bidiagonalization is the process of turning an arbitrary complex matrix $\mathbf{H}$ into a bidiagonal real matrix $\mathbf{B}$ (i.e., a matrix with zeros in all entries except the diagonal and superdiagonal terms). This is achieved by a series of unitary transformations, which are described in the following.

Description

A Householder reflector is an unitary transformation $\mathbf{P}_0$ that takes the first column of $\mathbf{H}$, $\mathbf{h}_1$, into the direction of the first canonical axis $\hat{\mathbf{e}}_1 = (1, 0, \ldots, 0)^{\mathrm{T}}$, while rotating the other columns arbitrarily as

$$\mathbf{P}_0\mathbf{H} = \begin{bmatrix} ||\mathbf{h}_1|| & \times & \times & \times \\ & \times & \times & \times \\ & \times & \times & \times \\ & \times & \times & \times \end{bmatrix}, \tag{16}$$

where $|| \cdot ||$ represents the Euclidean norm.

A second Householder reflector $\mathbf{Q_1}$ can be applied from the right, while preserving the first column intact, resulting in

$$\mathbf{P}_0\mathbf{H}\mathbf{Q}_1 = \begin{bmatrix} ||\mathbf{h}_1|| & ||\mathbf{g}|| & & \\ & \times & \times & \times \\ & \times & \times & \times \\ & \times & \times & \times \end{bmatrix}, \tag{17}$$

where $\mathbf{g}$ is the first row of the matrix $(\mathbf{P}_0\mathbf{H})_{1:N,2:N}$.

By repeating this procedure with the lower submatrices, we can obtain

$$\mathbf{B} = \mathbf{P}_{N-2} \ldots \mathbf{P}_1\mathbf{P}_0\mathbf{H}\mathbf{Q}_1 \ldots \mathbf{Q}_{N-2}, \tag{18}$$

where $\mathbf{B}$ is a bidiagonal matrix of real coefficients, and each $\mathbf{P}_j$ and $\mathbf{Q}_j$ is Householder reflector that operates in subspaces of dimension $N - j$.

Calculation cost

It can be seen that each $\mathbf{P}_j$ acts non-trivially only over a $N - j$ subspace. Hence, the computation of the non-trivial effect over the $(N - j) \times (N - j)$ matrix $\mathbf{A}$ can be computed as

$$\widetilde{\mathbf{P}}_j\mathbf{A} = \mathbf{A} - 2\mathbf{v}\frac{(\mathbf{v}^\dagger\mathbf{A})}{\mathbf{v}^\dagger\mathbf{v}}, \tag{19}$$

where $\widetilde{\mathbf{P}}_j$ corresponds to the $(N - j) \times (N - j)$ lower submatrix of P_j and $\mathbf{v} \in \mathbb{C}^{N-j}$ is a vector calculated as

$$\mathbf{v} = \mathrm{sign}(a_{11})||\mathbf{a}_1||\hat{\mathbf{e}}_1 + \mathbf{a}_1, \tag{20}$$

where $\mathbf{a}_1$ is the first column of $\mathbf{A}$ and a_{11} is the first element of $\mathbf{a}_1$ [31]. The calculation of $\mathbf{v}$ costs $2(N - j)$ real sums, an equal number of products, 1 square root and 1 sign operation. Recalling that one complex product consists of 4 real products and 2 real sums and that 1 complex sum takes 2 real sums, the cost of the application of $\widetilde{\mathbf{P}}_j$ is

$8(N-j)^2+2(N-j)-1$ real sums, $8(N-j)^2+5(N-j)$ real products and 1 division. The total cost of the transformation $\mathbf{P}_j$ is thus given in Table 5.

The application of $\mathbf{Q}_j$ is done repeating the same procedure to the hermitian of the lower $(N-j+1)\times(N-j)$ submatrix. Therefore, (19) and (20) are valid but with $\mathbf{A}$ being an $(N-j)\times(N-j+1)$ matrix. The cost $C(\mathbf{Q}_j)$ can be seen in Table 5.

Finally, the total cost of the phase I (cf. Table 3) can be calculated using (18) as

$$C_{\mathrm{I}} = \sum_{j=0}^{N-2} C(\mathbf{P}_j) + \sum_{j=1}^{N-2} C(\mathbf{Q}_j). \tag{21}$$

Phase II: Superdiagonal reduction

The second phase of the GRA reduces the upper diagonal terms into zeros, such that the real bidiagonal matrix $\mathbf{B}$ is diagonalized.

It can be shown that it is not possible to build an algorithm that performs this in a finite number of steps [31]. Hence, this phase consists of reducing the size of the upper-diagonal terms until they are smaller than a given threshold.

Description

This phase entails a series of *Givens rotations*, which are unitary operations on the 2-dimensional subspace spanned by canonical vectors $\hat{\mathbf{e}}_i$ and $\hat{\mathbf{e}}_j$. If $\mathbf{G}_{i,j}(\theta)$ is a Givens rotation on dimensions i and j with an angle θ, its effect on the canonical base $\{\hat{\mathbf{e}}_k\}_{k=1}^{N}$ is

$$\mathbf{G}_{i,j}(\theta)\hat{\mathbf{e}}_k = \begin{cases} \hat{\mathbf{e}}_i\cos(\theta)+\hat{\mathbf{e}}_j\sin(\theta) & \text{if} \quad k=i, \\ -\hat{\mathbf{e}}_i\sin(\theta)+\hat{\mathbf{e}}_j\cos(\theta) & \text{if} \quad k=j, \\ \hat{\mathbf{e}}_k & \text{in other case.} \end{cases} \tag{22}$$

The first step of the second phase is to apply a Givens rotation $\mathbf{T}_1=\mathbf{G}_{1,2}(\theta_1)$ from the right, where the angle θ_1 is chosen such that $\mathbf{T}_1^{\mathrm{T}}\mathbf{z}=||\mathbf{z}||\hat{\mathbf{e}}_1$ for a given $\mathbf{z}$. The effect of the application of $\mathbf{T}_1$ is that a nonzero element is introduced:

$$\mathbf{BT}_1 = \begin{bmatrix} \times & \times & & \\ + & \times & \times & \\ & & \times & \times \\ & & & \times \end{bmatrix}. \tag{23}$$

The rest of the second phase is to perform a series of Givens rotations to displace this nonzero element out of the matrix. It starts with a Givens rotation $\mathbf{Q}_1=\mathbf{G}_{1,2}(\theta_2)$, which makes $\mathbf{Q}_1\mathbf{y}=||\mathbf{y}||\hat{\mathbf{e}}_1$, where $\mathbf{y}$ is the first column of the matrix $\mathbf{BT}_1$. The result will have a zero in the desired position, but will introduce a new nonzero entry:

Table 3 Bidiagonalization computational cost

Cost Item	Sums	Products	Divisions	Square roots	Sign Operations
$C(\mathbf{P}_j)$	$8(N-j)^2+2(N-j)$	$8(N-j)^2+6(N-j)$	1	1	1
$C(\mathbf{Q}_j)$	$8(N-j)^2-6(N-j)-4$	$8(N-j)^2-2(N-j)-6$	1	1	1
C_{I}	$\frac{16}{3}N^3-2N^2+\frac{8}{3}N-4$	$\frac{16}{3}N^3+2N^2+\frac{2}{3}N+8$	$2N-3$	$2N-3$	$2N-3$

$$\mathbf{Q}_1(\mathbf{BT}_1) = \begin{bmatrix} \times & \times & + & \\ & \times & \times & \\ & & \times & \times \\ & & & \times \end{bmatrix}. \tag{24}$$

This procedure can be repeated until the nonzero position reaches the bottom of the matrix:

$$\mathbf{Q}_{N-2} \ldots \mathbf{Q}_1\mathbf{BT}_1 \ldots \mathbf{T}_{N-1}. \tag{25}$$

At this stage, a last Givens rotation $\mathbf{Q}_{N-1} = \mathbf{G}_{N-1,N}(\theta_{2N-2})$ will act on the lower 2×2 submatrix and turn the desired element into a zero entry without introducing new nonzero entries, giving a new bidiagonal matrix

$$\mathbf{B}_1 = \mathbf{Q}_{N-1} \ldots \mathbf{Q}_1\mathbf{BT}_1 \ldots \mathbf{T}_{N-1}. \tag{26}$$

This step can be repeated for generating a sequence of bidiagonal matrices $\mathbf{B}_n$. It can be shown that $\mathbf{B}_n$ converges to a diagonal matrix $\mathbf{D}$ that has the singular values of the original matrix $\mathbf{H}$.

Calculation cost

First we calculate the number of operations needed in one step of the algorithm, C_k, which turns a k-dimensional bidiagonal matrix $\mathbf{B}_n$ into a new bidiagonal matrix $\mathbf{B}_{n+1}$. This cost has two sources: the cost of calculating the Givens rotations $C^{(k)}_{\text{calc}}$ and the application of the Givens rotations $C^{(k)}_{\text{app}}$.

The Givens rotation is used for rotating a two-dimensional vector (α_1, α_2) onto its first axis:

$$\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} = \begin{pmatrix} \sigma \\ 0 \end{pmatrix}. \tag{27}$$

Therefore, the generation of a Givens rotation is equivalent to the calculation of $\cos\theta$ and $\sin\theta$ as function of (α_1, α_2). A stable algorithm for doing this is [13]:

$$\text{if } \alpha_2 = 0 : \cos\theta = 1,\ \sin\theta = 0; \tag{28}$$

$$\text{if } |\alpha_2| \geq |\alpha_1| = 0 : v = \alpha_1/\alpha_2,\ w = \sqrt{1+v^2}, \tag{29}$$

$$\sin\theta = 1/w,\ \cos\theta = v/w; \tag{30}$$

$$\text{else} : v = \alpha_2/\alpha_1,\ w = \sqrt{1+v^2}, \tag{31}$$

$$\cos\theta = 1/w,\ \sin\theta = v/w. \tag{32}$$

The average cost of calculating a Givens rotation is 1 sum, 1 product, 2 divisions and 1 square root. As each iteration of the algorithm consists of $2(k-1)$ Givens rotations, the total calculation cost is given by $C^{(k)}_{\text{calc}}$ (cf. Table 5).

We still need to calculate $C^{(k)}_{\text{app}}$. The first rotation $\mathbf{T}_1$ is applied to the first two columns of a bidiagonal matrix $\mathbf{B}_n$ as

Table 4 Superdiagonal reduction computational cost

Cost item	Sums	Products	Divisions	Square roots	Sign operations
$C_{\text{calc}}^{(k)}$	$2(k-1)$	$2(k-1)$	$4(k-1)$	$2(k-1)$	
$C_{\text{app}}^{(k)}$	$6k-7$	$16k-20$			
C_k	$8k-9$	$18k-22$	$4k-4$	$2k-2$	
C_{II}	$8N^2-10N+2$	$18N^2-26N+8$	$4N^2-4N$	$2N^2-2N$	

Table 5 Computational Cost of GRA

Cost Item	Sums	Products	Divisions	Square roots	Sign Operations
C_N	$\frac{16}{3}N^3+10N^2-\frac{28}{3}N+10$	$\frac{16}{3}N^3+16N^2-\frac{70}{3}N+4$	$4N^2-2N-3$	$2N^2-3$	$2N-3$

$$\mathbf{B}_n\mathbf{T}_1=\begin{bmatrix}\times & \times & & \\ & \times & \times & \\ & & \times & \times \\ & & & \times\end{bmatrix}\begin{bmatrix}c & s & & \\ -s & c & & \\ & & 1 & \\ & & & 1\end{bmatrix}. \tag{33}$$

The computational cost of the application of $\mathbf{T}_1$, denoted as $C(\mathbf{T}_1)$, is 6 products and 2 sums. By considering (23), the application of the second rotation can be seen as

$$\mathbf{Q}_1(\mathbf{B}_n\mathbf{T}_1)=\begin{bmatrix}c & s & & \\ -s & c & & \\ & & 1 & \\ & & & 1\end{bmatrix}\begin{bmatrix}\times & \times & & \\ \times & \times & \times & \\ & & \times & \times \\ & & & \times\end{bmatrix}. \tag{34}$$

Counting the operations, and recalling that the entry of the second row and first column is zero by construction, one finds that the cost $C(\mathbf{Q}_1)$ is 8 products and 3 sums. Comparing (23) with (24), one can conclude that all the rotations, excepting the very last one, have the same structure and therefore share the same costs.

The last rotation has the form $\mathbf{Q}_{k-1}(\mathbf{Q}_{k-2}\ldots\mathbf{T}_{k-1})$ and costs 6 products and 3 sums. Adding all together, we obtain the total operations of the application of one step of the algorithm in a k-dimensional matrix

$$C_{\text{app}}^{(k)}=\sum_{i=1}^{k-1}\{C(\mathbf{T}_i)+C(\mathbf{Q}_i)\}. \tag{35}$$

Hence, the total cost of one k-dimensional iteration of the algorithm is

$$C_k=C_{\text{calc}}^{(k)}+C_{\text{app}}^{(k)}. \tag{36}$$

It has been reported that the algorithm usually ends with no more than $2N$ iterations [12]. If we consider an average case where 2 iterations are needed per matrix size from 2 to N, then the total cost of the second phase (cf. Table 4) is given by

$$C_{\text{II}}=2\sum_{k=2}^{N}C_k \tag{37}$$

Total calculation cost

The total cost of the GRA is given by the sum of the costs of phases I and II, i.e., $C_N = C_{\mathrm{I}} + C_{\mathrm{II}}$ (cf. Table 5). For large values of N, the total cost of GRA is dominated by the third-order terms, which are part of the phase I of the algorithm [31].

Abbreviations

SVD: Singular value decomposition; WSN: Wireless sensor network; CSIT: Channel state information at the transmitter; PIA: Power iteration algorithm; BER: Bit error rate; MIMO: Multiple-input multiple-output; CSI: Channel state information; SER: Symbol error rate; FDD: Frequency division multiplexing; SNR: Signal-to-noise ratio; TDD: Time division duplexing; GRA: Golub–Reinsch algorithm; BIMA: Blind iterative MIMO algorithm; SVD-BF: SVD beamforming; AWGN: Additive white Gaussian noise; QAM: Quadrature amplitude modulation; ASK: Amplitude shift keying; QPSK: Quadrature phase-shift keying; LS: Least square; RCPA: Reduced-complexity power algorithm; ALU: Arithmetic logic unit; PPP: *Ping–Pong–Pong*; SISO: Single-input single-output; MRC: Maximum ratio combining.

Acknowledgements

The authors would like to thank CONICYT Chile for supporting this research with the master scholarship CONICYT-PCHA Magíster Nacional 2013 - 221320215 and the Projects 15110017 FONDAP 2011 and FONDEF IT13i20015.

Authors' contributions

CO conceived the study. FK carried out the main work of Sects. 3 and 4 under CO's advising. FR contributed with the analysis in "Appendix" leading to equation (15) in Sect. 3.4. All authors contributed to draft the manuscript. All authors read and approved the final manuscript.

Funding

Funding sources mentioned in the acknowledgment section.

Data availability statement

Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

Declarations

Competing interests

The authors declare that they have no competing interests.

Author details

[1]Department of Electrical Engineering, Pontificia Universidad Católica de Chile, Santiago, Chile. [2]Computer Science Department, Stanford University, Stanford, CA, USA. [3]Centre of Complexity Science and Department of Mathematics and Department of Electrical and Electronic Engineering, Imperial College London, London, UK. [4]National Research Center for Integrated Natural Disaster Management, Pontificia Universidad Católica de Chile, Santiago, Chile.

Received: 18 February 2021 Accepted: 24 June 2021
Published online: 10 July 2021

References

1. J.M. Kahn, R.H. Katz, K.S.J. Pister, Next century challenges: mobile networking for smart dust. in *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking, MobiCom '99 (ACM, New York, NY, USA, 1999)*. pp. 271–278 (1999). https://doi.org/10.1145/313451.313558
2. S. Cui, A. Goldsmith, A. Bahai, Energy-efficiency of MIMO and cooperative MIMO techniques in sensor networks. IEEE J. Sel. Areas Commun. **22**(6), 1089–1098 (2004). https://doi.org/10.1109/JSAC.2004.830916
3. W. Liu, X. Li, M. Chen, Energy efficiency of MIMO transmissions in wireless sensor networks with diversity and multiplexing gains. in *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05).*, vol. 4, pp. iv/897–iv/900 (2005). https://doi.org/10.1109/ICASSP.2005.1416154
4. F. Rosas, C. Oberli, Energy-efficient MIMO SVD communications. in *IEEE 23rd International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC), 2012* pp. 1588–1593 (2012). https://doi.org/10.1109/PIMRC.2012.6362601
5. F. Rosas, C. Oberli, Nakagami-m approximations for multiple-input multiple-output singular value decomposition transmissions. Commun. IET **7**(6), 554–561 (2013). https://doi.org/10.1049/iet-com.2012.0400
6. A. Goldsmith, *Wireless Communications* (Cambridge University Press, Cambridge, 2005)
7. D. Love, R. Heath, T. Strohmer, Grassmannian beamforming for multiple-input multiple-output wireless systems. IEEE Trans. Inf. Theory **49**(10), 2735–2747 (2003). https://doi.org/10.1109/TIT.2003.817466
8. K. Mukkavilli, A. Sabharwal, E. Erkip, B. Aazhang, On beamforming with finite rate feedback in multiple-antenna systems. IEEE Trans. Inf. Theory **49**(10), 2562–2579 (2003). https://doi.org/10.1109/TIT.2003.817433
9. A. Narula, M. Lopez, M. Trott, G.W. Wornell, Efficient use of side information in multiple-antenna data transmission over fading channels. IEEE J. Sel. Areas Commun. **16**(8), 1423–1436 (1998). https://doi.org/10.1109/49.730451
10. P. Xia, G. Giannakis, Design and analysis of transmit-beamforming based on limited-rate feedback. IEEE Trans. Signal Process. **54**(5), 1853–1863 (2006). https://doi.org/10.1109/TSP.2006.871967

11. Venkataramani, R., Marzetta, T.L.: Reciprocal training and scheduling protocol for MIMO systems. in *Proceedings of 41st Annual Allerton Conference Communication, Control, Computing* (2003)
12. G. Golub, C. Reinsch, Singular value decomposition and least squares solutions. Numer. Math. **14**(5), 403–420 (1970)
13. A. Björck, *Numerical Methods for Least Squares Problems* (Society for Industrial and Applied Mathematics, Philadelphia, 1996). https://doi.org/10.1137/1.9781611971484
14. C. Studer, P. Blosch, P. Friedli, A. Burg, Matrix decomposition architecture for MIMO systems: design and implementation trade-offs. in *Conference Record of the Forty-First Asilomar Conference on Signals, Systems and Computers, 2007. ACSSC 2007*. pp. 1986–1990. https://doi.org/10.1109/ACSSC.2007.4487584
15. C.Z. Zhan, Y.L. Chen, A.Y. Wu, Iterative superlinear-convergence SVD beamforming algorithm and VLSI architecture for MIMO-OFDM systems. IEEE Trans. Signal Process. **60**(6), 3264–3277 (2012). https://doi.org/10.1109/TSP.2012.2190405
16. J. Andersen, Array gain and capacity for known random channels with multiple element arrays at both ends. IEEE J. Sel. Areas Commun. **18**(11), 2172–2178 (2000). https://doi.org/10.1109/49.895022
17. Y. Tang, B. Vucetic, Y. Li, An iterative singular vectors estimation scheme for beamforming transmission and detection in MIMO systems. IEEE Commun. Lett. **9**(6), 505–507 (2005). https://doi.org/10.1109/LCOMM.2005.06006
18. T. Dahl, N. Christophersen, D. Gesbert, Blind MIMO eigenmode transmission based on the algebraic power method. IEEE Trans. Signal Process. **52**(9), 2424–2431 (2004). https://doi.org/10.1109/TSP.2004.832000
19. P. Xia, H. Niu, J. Oh, C. Ngo, Practical antenna training for millimeter wave MIMO communication. in *IEEE 68th Vehicular Technology Conference, 2008. VTC 2008-Fall*, pp. 1–5 (2008). https://doi.org/10.1109/VETECF.2008.253
20. S. Gazor, K. Al Suhaili, Communications over the best singular mode of a reciprocal MIMO channel. IEEE Trans. Commun. **58**(7), 1993–2001 (2010). https://doi.org/10.1109/TCOMM.2010.07.090297
21. G. Golub, C. Van Loan, *Matrix Computations*, 3rd edn. (Johns Hopkins University Press, Baltimore, 1996)
22. C. Eckart, G. Young, A principal axis transformation for non-Hermitian matrices. Bull. Am. Math. Soc. **45**(2), 118–121 (1939)
23. G. Smith, A direct derivation of a single-antenna reciprocity relation for the time domain. IEEE Trans. Antennas Propag. **52**(6), 1568–1577 (2004). https://doi.org/10.1109/TAP.2004.830257
24. M. Guillaud, D.T.M. Slock, R. Knopp, A practical method for wireless channel reciprocity exploitation through relative calibration. in *Proceedings of the Eighth International Symposium on Signal Processing and its Applications, 2005*, vol. 1, pp. 403–406 (2005). https://doi.org/10.1109/ISSPA.2005.1580281
25. V. Jungnickel, U. Kruger, G. Istoc, T. Haustein, C. Von Helmolt, A MIMO system with reciprocal transceivers for the time-division duplex mode. in *Antennas and Propagation Society International Symposium, 2004*. IEEE, vol. 2 (2004). pp. 1267–1270. https://doi.org/10.1109/APS.2004.1330415
26. J. Aldunate, C. Oberli, An acquisition scheme for communications in multi-antenna sensor networks with low signal to noise ratio. Int. J. Sensor Netw. **27**(4), 259–267 (2018). https://doi.org/10.1504/IJSNET.2018.10014899
27. F. Rosas, C. Oberli, Modulation and SNR optimization for achieving energy-efficient communications over short-range fading channels. IEEE Trans. Wireless Commun. **11**(12), 4286–4295 (2012). https://doi.org/10.1109/TWC.2012.100112.111275
28. C. Muñoz, C. Oberli, Energy-efficient estimation of a MIMO channel. EURASIP J. Wireless Commun. Netw. **2012**(1), 353 (2012). https://doi.org/10.1186/1687-1499-2012-353
29. F. Rosas, C. Oberli, Impact of the channel state information on the energy-efficiency of MIMO communications. IEEE Trans. Wireless Commun. **14**(8), 4156–4169 (2015). https://doi.org/10.1109/TWC.2015.2417530
30. S. Alamouti, A simple transmit diversity technique for wireless communications. IEEE J. Sel. Areas Commun. **16**(8), 1451–1458 (1998). https://doi.org/10.1109/49.730453
31. L. Trefethen, D. Bau, *Numerical Linear Algebra* (Society for Industrial and Applied Mathematics, Philadelphia, 1997)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.