This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2015.2497687, IEEE Transactions on Wireless Communications

A scalable architecture for distributed receive beamforming: analysis and experimental demonstration

F. Quitin, A.T. Irish and U. Madhow

***Abstract*—We propose, analyze and demonstrate an architecture for scalable cooperative reception. In a cluster of $N+1$ receive nodes, one node is designated as the final receiver, and the N other nodes act as amplify-and-forward relays which adapt their phases such that the relayed signals add up constructively at the designated receiver. This yields received SNR scaling linearly with N, while avoiding the linear increase in overhead incurred by a direct approach in which received signals are separately quantized and transmitted for centralized processing. By transforming the task of long-distance distributed receive beamforming into one of local distributed transmit beamforming, we can leverage a scalable one-bit feedback algorithm for phase synchronization. We show that time division between the long-distance and local links eliminates the need for explicit frequency synchronization. We provide an analytical framework, whose results closely match Monte Carlo simulations, to evaluate the impact of phase noise due to relaying delay on the performance of the one-bit feedback algorithm. Experimental results from our prototype implementation on software-defined radios demonstrate the expected gains in received signal strength despite significant oscillator drift, and are consistent with results from our analytical framework.**

***Index Terms*—distributed MIMO, beamforming, cooperative reception, synchronization**

I. Introduction

Distributed MIMO (D-MIMO) refers to a broad class of techniques in which a group of cooperating nodes acts as a distributed antenna array, in order to obtain performance gains similar to those provided by conventional centralized MIMO. While it is difficult to scale centralized arrays to a large number of elements due to size and weight considerations (especially at lower carrier frequencies), in principle, D-MIMO allows us to synthesize very large apertures using the natural geographic distribution of the cooperating nodes, and offer an approach to massive MIMO that sidesteps form factor constraints. The opportunistic formation of D-MIMO clusters can also have significant benefits in enhancing range/rate tradeoffs, especially in emergency and disaster relief scenarios. However, key difficulties in translating D-MIMO from concept to practice are that the cooperating nodes have independent oscillators, each with stochastic drift, and that we cannot rely on a regular array geometry in our signal processing algorithms. Another important consideration in D-MIMO system design is that we would like our architectures and algorithms to scale gracefully as the number of cooperating nodes increases, in order to approach the vision of arbitrarily large virtual arrays. In this paper, we address these issues in the context of distributed receive (D-Rx) beamforming.

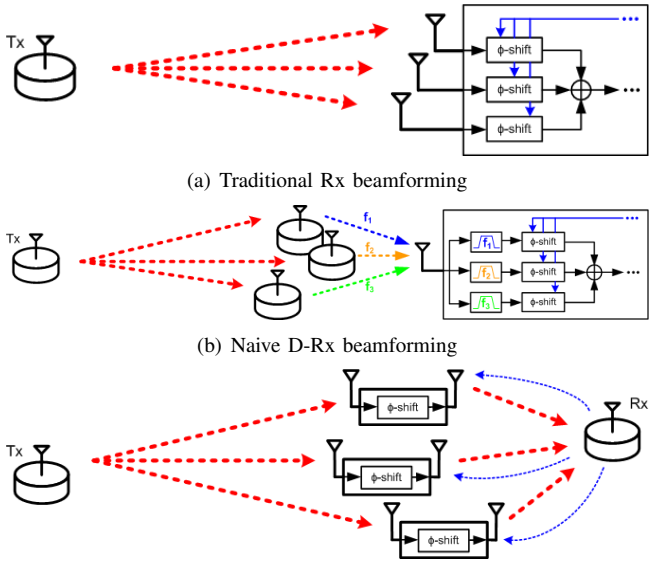
In a D-Rx beamforming system, a cluster of nodes coherently combine their received signals in order to enhance the received signal-to-noise ratio (SNR). In a centralized receive array, depicted in Figure 1(a) this is accomplished by routing signals from different receive antennas along wires, with phase shifts for coherent combining applied at RF or IF, or digitally at baseband, after downconversion and analog-to-digital conversion. An analogous approach for distributed receive beamforming, shown in Figure 1(b), is for each node to send its received signal to a centralized processor (typically via a fast local wireless link), which then applies the appropriate phase shifts to achieve receive beamforming. With this approach, the cooperating nodes do not even have to be synchronized *a priori.* The centralized processor has access to the received signal for each node, and hence can estimate relative frequency and phase offsets and then compensate for them when combining the signals. However, this direct approach does not scale to a large number of cooperating nodes, since the amount of local communication is proportional to the number of nodes. We therefore propose and investigate in this paper an alternative approach which attains scalability by using "over the air" coherent combining.

The proposed architecture (discussed in more detail in the next section) is depicted in Figure 1(c). A receive cluster of $N+1$ nodes wishes to enhance the SNR of a signal arriving from a distant source. One of the receive nodes is designated as the final receiver, and the remaining N nodes relay their received signals, adjusting their phases such that the relayed signals combine coherently at the designated receiver. This converts the task of D-Rx beamforming on the "long link" from source to receive cluster into one of distributed transmit (D-Tx) beamforming on the "short link" between the relays and the receiver. This allows us to leverage prior work on scalable D-Tx beamforming, while adapting to features and impairments peculiar to our relay-based architecture, in order to attain a scalable D-Rx system.

The key features and contributions of this paper are summarized as follows:

F. Quitin is with the Brussels School of Engineering, Université libre de Bruxelles (ULB), Belgium (fquitin@ulb.ac.be). A.T. Irish and U. Madhow are with the Electrical and Computer Engineering Department, University of California, Santa Barbara (UCSB) ({andrewirish, madhow}@ece.ucsb.edu)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2015.2497687, IEEE Transactions on Wireless Communications



(a) Traditional Rx beamforming

(b) Naive D-Rx beamforming

(c) Scalable D-Rx beamforming

Fig. 1. Rx beamforming with traditional MIMO, naive D-MIMO and scalable D-MIMO

1) **Scalable architecture:** By using amplify-forward relays in the receive cluster, we ensure that local communication overhead does not blow up with the number of cooperating nodes. We use a provably convergent one-bit feedback algorithm for distributed transmit beamforming in order to ensure that the relayed signals add up coherently at the receiver(Section II).
2) **Implicit frequency synchronization:** For time division between the long link and the short link as considered here, the frequency offsets of the relays "cancel out" on the long and short links, hence there is no need to synchronize the relays in frequency (Section II-III).
3) **Analytical performance characterization:** While frequency synchronization is not required, the delay between message reception on the long link and message relay on the short link leads to phase errors accumulating because of frequency and phase drift. We characterize the statistics of such phase errors as a function of local oscillator (LO) parameters (Section III), and then provide an analytical framework for determining their effect on the one-bit feedback algorithm (Section IV). Our analysis matches closely with Monte Carlo simulations, and shows that when the phase error gets large, only a fraction of the expected beamforming gain is achieved.
4) **Proof of concept:** The proposed architecture is implemented on a software-defined radio testbed, showing that the expected gains can be achieved with up to four relay nodes. By relaxing the system parameters, we are able to observe the performance of the one-bit feedback algorithm under larger phase errors experimentally, thereby verifying the insights from our analytical framework (Section V).

Related work: Many information-theoretic analyses, ranging from three decades back [1, 2] to the present [3–5], rely on the concept of cooperative wireless communication, without explicitly addressing the fundamental bottleneck of synchronization between cooperating nodes. However, there has been significant recent progress on the problem of distributed synchronization, most of it in the context of distributed transmit beamforming. Closed-loop synchronization techniques include explicit feedback [6, 7], one-bit aggregate feedback [8–10], implicit feedback using reciprocity [11], round-trip synchronization [12, 13] or two-way synchronization [14, 15]. Each of these synchronization techniques offers different trade-offs between complexity, coordination overhead and scalability to larger networks (see the review paper in [16] and the discussion in [17]). Distributed MIMO techniques have also been investigated in the context of "coordinated multipoint" (CoMP) capabilities for 4G-LTE systems, where multiple base stations act as a distributed antenna array [18]. Recent work [19] has shown that the overhead for D-Rx beamforming using the architecture in Figure 1(b) can be reduced by heavily quantizing the information exchanged. However, the local communication overhead still scales up with the number of cooperating nodes, unlike our proposed approach.

There is also by now a significant body of research in analysis and simulation of amplify-and-forward relaying: in [20], the receiver broadcasts a single bit of information to each relay indicating whether it should participate in the communication, thereby selecting a set of relays who happen to be combining quasi-coherently; [21] considers network beamforming where each node has perfect channel state information; [22] considers network cooperation where each node computes beamforming weights based on local channel information only; and [23, 24] proposes a robust collaborative beamforming scheme based on partial channel state information. In [19], nodes forward their message over a local area network, and it is shown that by selecting a subset of the receiving nodes performance comes close to that of optimal receive beamforming.

While beamforming is implicit with ideal amplify-forward relays, to the best of our knowledge the present paper is the first (other than our prior conference paper [25]) to a) identify the critical importance of amplify-forward relaying in providing a scalable architecture for distributed receive beamforming, b) model in detail the synchronization issues in implementing this approach, and c) provide a proof of concept via our testbed. A significant contribution of this paper beyond [25] is that we provide a detailed analytical framework for the impact of phase errors on the one-bit feedback algorithm. We also extend the prototype in [25] to a larger receive cluster, and provide a more detailed characterization of the effects of the system parameters on experimental performance.

Distributed MIMO techniques have also been demonstrated with a wide variety of experimental prototypes. In [9, 26, 27], distributed transmit beamforming prototypes using wired feedback channels were presented for both RF and millimeter wave frequencies. A first fully wireless setup was presented in [28], but this setup still used analog signaling for the feedback channel. A D-Tx beamforming prototype using full digital signaling and an extended Kalman filter for frequency synchronization was presented in [29, 30]. While the preceding papers focus on distributed transmit beamforming, multiuser distributed MIMO has been demonstrated in [31]. However,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2015.2497687, IEEE Transactions on Wireless Communications

the latter uses dedicated wired backhaul links to distribute information and feedback throughout the network. Similarly, CoMP experiments for 4G cellular systems have to date relied on dedicated backhaul links with low latency and high throughput, as well as assuming uninterrupted GPS connections to synchronize the various base stations. To the best of our knowledge, the present paper is the first work to analyze and prototype all-wireless distributed receive beamforming.

Outline of the paper: Section II discusses our system model, the challenges of achieving synchronization at multiple levels in such a system, and points out the implicit frequency synchronization achieved by the design considered here. In Section III, we characterize the phase error accumulating due to relaying delay. Section IV presents the phase synchronization algorithm and studies the performance of the this algorithm under the presence of phase error. Finally, Section V presents the implementation of our architecture on an experimental testbed, and shows some of the results obtained with our prototype. The prototype is run with different parameters to confirm the theoretical insights of the previous sections. Section VI contains our conclusions.

II. System Model

While there are many possible design choices for the scalable D-Rx architecture depicted in Figure 1(c), the specific choices in our modeling, analysis and prototyping are as follows. We focus on narrowband signaling, with channels modeled as complex gains. Each relay node receives the signal from the distant transmitter over the long link, applies a phase shift to the received signal, and forwards it to a central receiver over a short link. All relays forward the message to the receiver over the same frequency band, with over-the-air combining at the receiver. We assume that the same frequency band is used, via time division, for the long link and the short link, which enables implicit frequency synchronization. The relays adapt their phases using the one-bit feedback algorithm [10], which also formed the basis for our prior prototyping of distributed transmit beamforming in [29]. Note that the relays can be implemented in RF or in baseband, assuming sufficient ADC resolution. In our prior work [25], we showed that when the relayed messages combine coherently in distributed receive beamforming setup, the SNR of the received message at the final receiver is given by

$$\text{SNR} \approx \frac{N^2 g_r g_2}{1 + N g_r g_2} \text{SNR}_0 \tag{1}$$

where g_r is the relay gain, g_2 is the gain of the short link, N is the number of receive nodes and SNR_0 is the SNR if the transmitter was sending its message to the final receiver directly. This shows that an amplify-and-forward relaying setup is only desirable when $g_r g_2$ is large, meaning that the relay gain should be able to compensate the path loss of the short link Note that (1) is true when the gain of all transmitters-to-relay links are equal, and the gains of all relay-to-receiver links are equal.

The key challenge in coherent combining at the receiver is that signals emitted from relays with independent clocks and oscillators must line up. Three levels of synchronization are required: frequency, phase and timing. In the following, we describe our approach to each of these problems.

A. Implicit frequency synchronization

In D-MIMO systems, each terminal derives its RF signal from its own local oscillator (LO) which carries a small but non-zero frequency offset with respect to those of the other nodes. This results in LO frequency offsets that can range from a few Hz to tens of kHz for poor quality oscillators. If the signals forwarded by the relay nodes to the central receiver have significant frequency offsets, the total signal at the receiver will exhibit constructive and destructive interference patterns. In order to avoid such behavior, it is important that the signals arriving from the different relay nodes have no frequency offset with respect to one another. Fortunately, we get this for free in our architecture. Specifically, while we have transformed our D-Rx problem into one of distributed transmit beamforming along the short link, a key difference from pure transmit beamforming as in [29] is that we obtain *implicit* frequency synchronization by virtue of time division between the long and short links.

To see this, suppose that the transmitter has carrier frequency f_T (measured with respect to some absolute reference that we do not need to know). Similarly, the carrier frequencies of relay i and the receiver are denoted by f_i and f_R, respectively. Since we focus here on demonstrating implicit frequency synchronization, we suppress attenuation and noise from our notation. We also ignore phase offsets (modeled in detail later), which are compensated with our feedback mechanism. The transmitter sends a baseband message $x(t)$, with corresponding passband signal $\text{Re}\left(x(t)e^{j2\pi f_T t}\right)$. This is received over the long link by relay i and is downconverted using LO frequency f_i. The corresponding complex envelope is

$$r_{i,L}(t) = x(t)e^{j2\pi(f_T - f_i)t} \tag{2}$$

This signal is then relayed to the receiver over the short link using the same LO at frequency f_i. The passband signal sent by relay i over the short link is therefore given by

$$s_{i,S}(t) = \text{Re}\left(r_{i,L}(t)e^{j2\pi f_i t}\right) = \text{Re}\left(x(t)e^{j2\pi f_T t}\right) \tag{3}$$

where we have plugged in (2) to get the last equality. We see, therefore, that the dependence on the relay LO frequency f_i cancels out, so that the carrier frequency of the relayed signal is simply that of the transmitter, independent of i.

The technical conditions for such implicit frequency synchronization to occur are that a) the relay node receives and forwards its signal at the same carrier frequency and b) each relay node draw its carrier signal for its receive and transmit chain from the *same* LO. We assume that these requirements are met in our technical development, and enforce it in our prototype. Of course, in practice, LO phase and frequency drift causes an accumulation of phase error over the time between reception on the long link and relay on the short link. For the low-quality oscillators in our software-defined radio testbed, such effects are significant. In the next two sections, therefore, we model such phase errors in Section III and analyze their impact on performance in Section IV.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2015.2497687, IEEE Transactions on Wireless Communications

B. Phase synchronization with the one-bit feedback algorithm

Phase synchronization is achieved using the well-known one-bit feedback algorithm [10], which is a stochastic hill climbing procedure, with one iteration per cycle. The algorithm works as follows: at the n-th cycle of the algorithm, each relay i adds a small random phase perturbation $\delta_i[n]$ to its current phase $\phi_i[n]$. The receiver monitors the received signal strength (RSS) of the received, over-the-air combined message, which for N nodes is given by (where we assume the gain of the channels through each relay to be identical for simplicity)

$$y[n] = \frac{1}{N}\left|\sum_{i=1}^{N} e^{j(\phi_i[n]+\delta_i[n])}\right| \tag{4}$$

where we normalize RSS by the number of nodes N for convenience in our technical development. If the RSS $y[n]$ is higher than the earlier maximum RSS $y_{\max}[n]$, the receiver broadcasts back a positive bit to all relays; each relay then keeps the last random phase perturbation and the new maximum RSS becomes the current RSS. If $y[n] < y_{\max}[n]$, the receiver broadcasts a negative bit to all relays. Each relay discards its previous random phase perturbation and the maximum RSS remains unchanged. The phase ϕ_i at iteration $n+1$ is then given by

$$\phi_i[n+1] = \begin{cases} \phi_i[n] + \delta_i & \text{if } y[n] > y_{\max}[n] \\ \phi_i[n] & \text{if } y[n] < y_{\max}[n] \end{cases} \tag{5}$$

$$y_{\max}[n+1] = \max\left(y[n], y_{\max}[n]\right) \tag{6}$$

In the next cycle, the whole process is repeated. The theoretical convergence of this algorithm to a global optimum has been proven and characterized in [10], and it has been experimentally validated with numerous prototypes [9, 28, 29] that the RSS quickly converges to a near-optimum value.

This ideal version of the one-bit feedback algorithm does however suffer from LO phase drift, which, unfortunately, occurs in any real world implementation. For example, suppose that a combination of phases at the relay nodes leads to a near-optimum RSS at some iteration, so that subsequent iterations cannot improve upon the previous maximum. In this case, the phases applied by the relays will not change, but phase noise will cause the actual phase offsets to drift away from their ideal values. Over time, the phase offsets will accumulate more random error, the RSS will vary randomly at lower values, and the one-bit feedback algorithm will fail entirely. To avoid this undesirable behavior, the following change is applied to the one-bit feedback algorithm. Instead of comparing the RSS in a given cycle to the maximum RSS achieved in a *all* previous cycles, the current RSS is compared to the maximum RSS in the K previous cycles (where K is a finite number). The algorithm will then be able to recover from phase drifts, as well as other unmodeled changes in the propagation environment: if at some point a combination of phases is obtained with an RSS that cannot be outperformed, K cycles later this maximum RSS will be removed from the past RSS memory, making the algorithm robust. While this is the version of the algorithm that was applied in prior prototypes [9, 28, 29], the present paper is the first to explicitly incorporate finite memory in analytical characterization of the algorithm.

C. Implicit timing synchronization

The third type of synchronization is temporal: the packets forwarded by the relays must arrive simultaneously at the receiver to avoid the effects of inter-symbol interference (ISI). In this work, we consider narrowband signaling, so that the differences in propagation delays are significantly smaller than the inverse bandwidth and can be neglected. In such a setting, determining packet boundaries on the long link accurately and delaying the forwarded packet for a fixed amount of time T_d is sufficient to obtain precise message alignment at the receiver. Thus, we use the timings of the messages received by the relays on the long link to provide implicit timing for the messages relayed on the short link. Note that since the LOs of all relay node are independent, there will be slight variations in the delay T_d applied at each node. However, for narrowband systems, the variations in T_d among nodes is an order of magnitude smaller than the sample time, and does not cause significant ISI. For wideband dispersive channels, more sophisticated strategies are required to handle ISI (e.g., OFDM with frame synchronization across relays), but this is left as a topic for future work.

III. Modeling phase errors

We now model the phase error accumulated due to relaying delay. All carrier frequencies can now slowly drift over time. A transmitted message sent at time t_0 is sent using carrier frequency $f_T(t_0)$. This message is received at relay i at time t_1, and is downconverted using LO frequency $f_i(t_1)$. The complex envelope thus obtained is relayed at time $t_1 + T_d$, upconverted using LO frequency $f_i(t_1 + T_d)$. It is received at time t_r by the receiver, downconverted using receiver LO frequency $f_R(t_r)$. Thus, the net frequency offset seen by the original message from transmitter to receiver is given by

$$\Delta f_i = f_T(t_0) - f_i(t_1) + f_i(t_1 + T_d) - f_R(t_r) \tag{7}$$

The preceding frequency shift is independent of i (i.e., achieves implicit frequency synchronization) if $T_d = 0$ and t_1, t_r do not depend on i. The latter is a good approximation for our narrowband model: the delay spread from transmitter to different relays, and from the relays to the receiver, is small compared to the sample time. However, nonzero T_d implies that the frequency offsets Δf_i seen at the receiver do depend on i, which can lead to the potential for fading due to constructive and destructive interference among the signals received from different relays.

We now model the effect of LO drift in more detail. We show that it causes a zero mean Gaussian phase error, and compute the variance of the phase error as a function of system parameters.

LO model: We begin by describing the LO model used in this paper. As in [32], we simplify the three-state LO model from [33] to a two-state model which is sufficient to capture

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2015.2497687, IEEE Transactions on Wireless Communications

the LO dynamics of our system. The stochastic differential equations describing the two-state LO model are written as

$$\begin{cases} d\phi(t) = \omega(t)dt + \omega_c q_1 dw_1(t) \\ d\omega(t) = \omega_c q_2 dw_2(t) \end{cases} \quad (8)$$

where $\phi(t)$ and $\omega(t)$ are the LO phase and (angular) frequency offset at time t, ω_c is the carrier frequency and q_1 and q_2 are the process noise parameters that correspond to white frequency noise and random walk frequency noise, respectively. The noise terms $w_1(t)$, $w_2(t)$ are two independent, one-dimensional standard Wiener processes (standard Brownian motion), each one defined as a Gaussian process with stationary independent increments such that $w(t)-w(s) \sim \mathcal{N}(0, t-s)$ and $w(0) = 0$. Additionally, the integration of a Wiener process is distributed as $\int_a^b w(t)dt \sim \mathcal{N}(0, \frac{(b-a)^3}{3})$. The system in (8) is a strictly linear stochastic differential equation, hence, its solution is given by [33]

$$\begin{cases} \phi(t+T) = \phi(t) + T\omega(t) + \omega_c q_1 n_\phi(t,T) \\ \omega(t+T) = \omega(t) + \omega_c q_2 n_\omega(t,T) \end{cases} \quad (9)$$

Given the abovementioned properties of Wiener processes, the noise vector $\mathbf{n}(t,T) = [n_\phi(t,T)\ n_\omega(t,T)]^T$ can then be shown to be distributed as $\mathbf{n}(t,T) \sim \mathcal{N}(0, \mathbf{Q}(T))$ (see equations (8)-(9) in [33]) with $\mathbf{Q}(T)$ equal to

$$\mathbf{Q}(T) = \omega_c^2 q_1^2 \begin{bmatrix} T & 0 \\ 0 & 0 \end{bmatrix} + \omega_c^2 q_2^2 \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} \\ \frac{T^2}{2} & T \end{bmatrix} \quad (10)$$

We now use the model (9) to determine the resulting phase noise on our system.

Intra-cycle drift: Consider the signal at the final receiver shown in Figure 2. The message from the transmitter is received (with low amplitude) at the relays at time t_1. At time t_2, the relays amplify and forward the message to the receiver (received with higher amplitude). The phase offset of the relayed signal (through relay i) at the receiver is then given by

$$\begin{aligned} \phi_i(t_2) &= \phi_{i1}(t_1) + \phi_{i2}(t_2) \\ &= \phi_{i1}(t_1) + \phi_{i2}(t_1) + T_d\omega_{i2}(t_1) + n_\phi(t_2, T_d) \end{aligned} \quad (11)$$

where we used (9) and the fact that $t_2 = t_1 + T_d$, and where ϕ_{i1} and ϕ_{i2} denote the phase offset of the long and the short link, ϕ_i denotes the phase offset of the global link (from transmitter to receiver) through relay i, and $n_\phi(t_2, T_d)$ is the phase offset noise of the short link. The phase noise in (11) is composed of two terms: the phase drift between t_1 and t_2, and the accumulated phase due to frequency drift between t_1 and t_2. According to the model in (8)-(10), the phase noise variance of (11) is given by element (1,1) of $\mathbf{Q}(T_d)$:

$$\sigma_n^2(T_d) = \omega_c^2 q_1^2 T_d + \omega_c^2 q_2^2 \frac{T_d^3}{3} \quad (12)$$

The increase in variance with relaying delay T_d is intuitively reasonable: the longer the relays wait before forwarding the message, the larger the phase errors accumulated due to the independent LO drifts at the different relays.

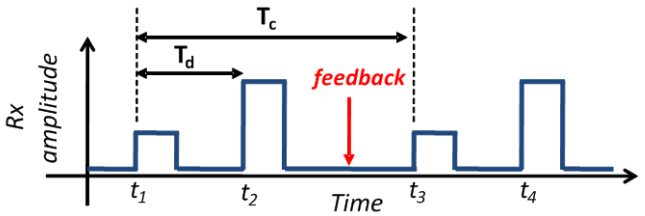


Fig. 2. Received signal over two cycles.

Inter-cycle drift: The previous analysis only consider the phase noise during a single cycle of our algorithm. Figure 2 shows several cycles of our setup. At time t_1, the transmitter's first message is received. This is forwarded by the relays at time t_2. The receiver then computes its feedback message that is returned to the relay nodes; essentially, this message controls the phases $\{\gamma_i\}$ applied by the relays, with the objective of having the total phase $\phi_{i1}(t_1) + \phi_{i2}(t_2) + \gamma_i$ equal for all i[1]. Now consider a second cycle of the system: at time t_3, another message from the transmitter is received by the relays. This is forwarded by the relays at time t_4. The total phase of the second message for link i is given by

$$\begin{aligned} \phi_i(t_4) &= \phi_{i1}(t_3) + \phi_{i2}(t_4) + \gamma_i \\ &= \phi_{i1}(t_3) + \phi_{i2}(t_3) + T_d\omega_{i2}(t_3) + n_\phi(t_3, T_d) + \gamma_i \\ &= \phi_{i1}(t_3) + \phi_{i2}(t_3) + T_d\left(\omega_{i2}(t_1) + n_\omega(t_3, T_c)\right) \\ &\quad + n_\phi(t_3, T_d) + \gamma_i \end{aligned} \quad (13)$$

where $T_c = t_3 - t_1$ is the cycle time. It can be seen that between t_1 and t_3 the LO frequency offset ω_{i2} has drifted, potentially causing the relay phase shift γ_i to become outdated. The variance of $n_\omega(t_3, T_c)$ is given by element (2,2) of $\mathbf{Q}(T_c)$. Obviously, if the cycle period gets longer, this drift will be more severe, resulting in a larger phase error. From (13), the total phase error due to both intra- and inter-cycle drift can be characterized by a zero-mean Gaussian process with variance

$$\sigma_n^2(T_d, T_c) = \omega_c^2 q_1^2 T_d + \omega_c^2 q_2^2 \frac{T_d^3}{3} + \omega_c^2 q_2^2 T_d^2 T_c \quad (14)$$

The first two terms of the equation are the same as for (12) and represent the intra-cycle drift, whereas the last term represent the inter-cycle drift. It can be concluded from (14) that, for fixed LO parameters q_1^2 and q_2^2, both the relay delay time T_d and the cycle time T_c should be kept low in order to maintain small phase errors.

In the next section, we analyze the effect of phase errors on the performance of the one-bit feedback algorithm.

IV. One-bit feedback with phase errors

In Section III, we showed that relaying delay results in a zero-mean Gaussian phase error for each relay. Even for poor quality LOs, these phase errors are well below $\frac{\pi}{2}$, and do not result in a constructive and destructive interference pattern

[1]Note that the term γ_i may be computed using the one-bit feedback algorithm, in which case γ_i is the accumulated phase up to the current cycle. However, as the development is valid for any type of feedback, we consider the more general notation γ_i to describe the phase shift applied at the i-th relay.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2015.2497687, IEEE Transactions on Wireless Communications

within the transmitted data frame. However, phase errors across frames have a much bigger effect, since they affect the reliability of the feedback used for phase synchronization. In this section, we develop an analytical framework to characterize the impact of the phase noise on the one-bit feedback algorithm (with RSS memory of length K; see description in Section II-B). In Section V, the effects of increased phase errors is investigated experimentally.

The key steps in our derivation are the following:

1) We determine the probability that the *noisy* phases cause an RSS increase, conditioned on the fact that the *noiseless* phases cause an RSS increase, over a single cycle of the algorithm (Section IV-A).
2) For finite memory K, we define K states, with S_k denoting the state that the maximum RSS was achieved k cycles ago, with $k = 1, ..., K$. We model the transitions between states as a Markov chain. Following arguments similar to those in Section IV-A, we compute the phase transition probabilities and derive the probability of being in each state S_k (Section IV-B).
3) We compute the RSS drift – the expected deviation in RSS at each iteration in the algorithm – when operating with noisy phases, both for positive and negative feedback, conditioned on the current state (Section IV-B).
4) Using the probabilities of being in each state S_k and the RSS drift conditioned on state S_k, we determine the total RSS drift when running the one-bit feedback algorithm with noisy phases (Section IV-B).

A. One-bit feedback algorithm with Gaussian phase noise over a single cycle

We start by investigating the effect of Gaussian phase noise over a single cycle. For N nodes, we define the following normalized RSS values

$$y = \frac{1}{N}\left|\sum_{i=1}^{N} e^{j(\phi_i)}\right| \tag{15a}$$

$$y_\delta = \frac{1}{N}\left|\sum_{i=1}^{N} e^{j(\phi_i+\delta_i)}\right| \tag{15b}$$

$$y_n = \frac{1}{N}\left|\sum_{i=1}^{N} e^{j(\phi_i+n_i)}\right| \tag{15c}$$

$$y_{\delta n} = \frac{1}{N}\left|\sum_{i=1}^{N} e^{j(\phi_i+\delta_i+n_i')}\right| \tag{15d}$$

that represent the noiseless RSS before random phase perturbation, the noiseless RSS after random phase perturbation, the noisy RSS before random phase perturbation and the noisy RSS after random phase perturbation, respectively. In these equations, δ_i is the random phase perturbation applied at each node, and n_i, n_i' are the phase noises (before and after random phase perturbation) at node i.

The problem at hand is then to determine how the feedback in a noisy setting relates to the ideal feedback without noise. Define

$$U \triangleq y_{\delta n} - y_n \tag{16a}$$

$$V \triangleq y_\delta - y \tag{16b}$$

which denote the RSS increments after a phase perturbation in the noisy and noiseless cases, respectively. In addition, we condition on y, the RSS with current relay phases if there were no noise. Conditioned on the feedback process state, the probability of successful detection of a phase improvement/deterioration is then given by $\mathbb{P}[U > 0|V > 0, y]$ and $\mathbb{P}[U < 0|V < 0, y]$. The random variables U and V are not independent, hence we must determine the joint distribution of (U, V) (conditioned on y) in order to compute the preceding probabilities. To proceed, we leverage the results derived in Conjecture 1 in [10], where statistical mechanics arguments were used to determine that, at each cycle of the algorithm, the distribution of the phases the nodes ϕ_i (which includes the accumulated phase due to the one-bit feedback algorithm in the previous cycles) follow an Exp-Cosine distribution around their mean. Given this framework, we show in Appendix A that the joint probability distribution of (U, V) conditioned on y can be approximated by the bivariate Gaussian distribution (17) given at the bottom of the page, where $\chi_\delta = \mathbb{E}[\cos(\delta_i)]$, $\chi_n = \mathbb{E}[\cos(n_i)]$ and $\chi_{\delta n} = \mathbb{E}[\cos(\delta_i + n_i')]$, where $\rho_\delta = \chi_\delta^2 - \mathbb{E}[\cos(2\delta_i)]$, $\rho_n = \chi_n^2 - \mathbb{E}[\cos(2n_i)]$ and $\rho_{\delta n} = \chi_{\delta n}^2 - \mathbb{E}[\cos(2(\delta_i + n_i))]$, where $\rho_{2\delta n} = \chi_{\delta n}\chi_\delta - \mathbb{E}[\cos(2\delta_i + n_i)]$, and where the term $\kappa(y) = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}[e^{2j\phi_i}]$ depends on y only, and can be approximated by $\kappa(y) = e^{-4(1-y)}$ for large y. In this paper, the random phase perturbation is chosen uniformly from $\{-\delta, \delta\}$. The preceding expectations can then be written out as $\mathbb{E}[\cos(\delta_i)] = \cos(\delta)$, $\mathbb{E}[\cos(2\delta_i)] = \cos(2\delta)$, $\mathbb{E}[\cos(n_i)] = \exp(-\sigma_n^2/2)$, $\mathbb{E}[\cos(2n_i)] = \exp(-2\sigma_n^2)$, $\mathbb{E}[\cos(\delta_i + n_i') = \mathbb{E}[\cos(\delta_i)]\mathbb{E}[\cos(n_i)]$, $\mathbb{E}[\cos(2(\delta_i + n_i))] = \mathbb{E}[\cos(2\delta_i)]\mathbb{E}[\cos(2n_i)]$ and $\mathbb{E}[\cos(2\delta_i + n_i)] = \mathbb{E}[\cos(2\delta_i)]\mathbb{E}[\cos(n_i)]$. From (17), one can easily compute the probabilities that the one-bit feedback algorithm successfully detects a phase improvement $\mathbb{P}[y_{\delta n} > y_n|y_\delta > y, y]$ or a phase deterioration $\mathbb{P}[y_{\delta n} < y_n|y_\delta < y, y]$ in the noisy case.

The analytical approximation for the joint distribution (17) matches Monte-Carlo simulations very well for as few as 10 nodes. The random phase perturbation must be small enough for the algorithm to converge, yet large enough so that the number of iterations required to reach convergence remains reasonable. In these simulations, the random phase perturbation was chosen uniformly from the discrete set $\{-10°, +10°\}$, to match our experiments in Section V. This

$$\mathbb{P}[U, V|y] \sim \mathcal{N}\left(\begin{bmatrix} (\chi_{\delta n} - \chi_n)y \\ (\chi_\delta - 1)y \end{bmatrix}, \begin{bmatrix} \frac{1-\chi_{\delta n}^2-\rho_{\delta n}\kappa(y)+1-\chi_n^2-\rho_n\kappa(y)}{2N} & \frac{\chi_n-\chi_{\delta n}\chi_\delta-\rho_{2\delta n}\kappa(y)}{2N} \\ \frac{\chi_n-\chi_{\delta n}\chi_\delta-\rho_{2\delta n}\kappa(y)}{2N} & \frac{1-\chi_\delta^2-\rho_\delta\kappa(y)}{2N} \end{bmatrix}\right) \tag{17}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2015.2497687, IEEE Transactions on Wireless Communications

choice provides a reasonable tradeoff of the aforementioned constraints [10]. Note that other distributions for the random phase perturbations yield similar performance. It can be seen in Figure 3 that, for low phase noise, U and V are highly correlated: a successful decision in the noiseless case gives way to an identical decision in the noisy case. As the phase noise increases, U and V become less correlated, and the probability that the one-bit feedback algorithm makes a correct decision (with respect to the noiseless case) becomes lower.

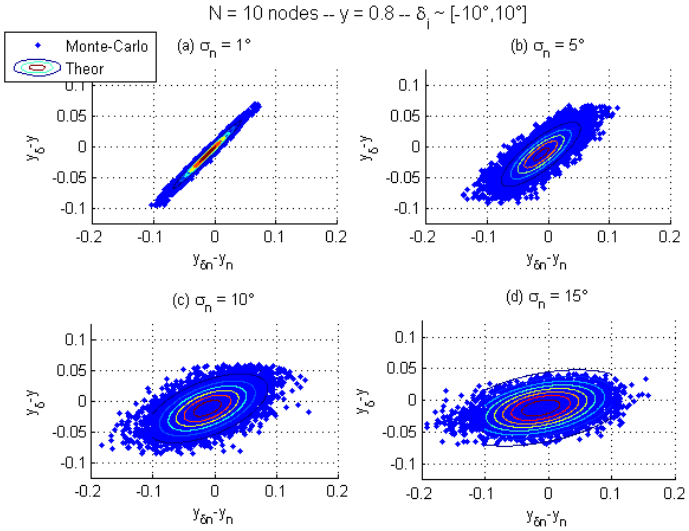


Fig. 3. Comparison between Monte-Carlo simulations and theoretical model for various phase noise variances and $N = 10$ nodes. The random phase perturbations are chosen uniformly from the discrete set $\{-10°, +10°\}$.

B. One-bit feedback algorithm with Gaussian phase noise and an RSS memory of length K

We now model finite memory K: the current RSS at cycle l is compared with the maximum RSS seen in the past K cycles. We define the state S_k $(k = 1, ..., K)$ as the state where the maximum RSS during the K previous cycles was obtained during cycle $l - k$, as shown in Figure 4.

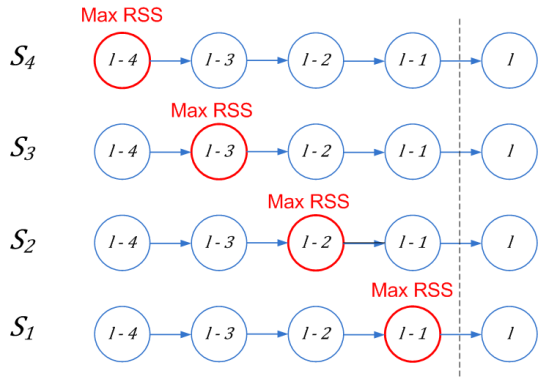


Fig. 4. Different states of the system for a memory of size $K = 4$.

We wish to compute the total RSS drift, that is the average RSS increment at cycle l conditioned on y (i.e. the noiseless RSS before phase perturbation). We make the following simplifying assumption: if we are in state S_k at cycle l, the RSS drift is *statistically independent* of the feedback before cycle $l - k$. In other words, the RSS drift depends only on the feedback obtained between cycle $l - k$ and cycle l. This can be verified as follows: imagine the following time instants $l_0 < l_1 < l - k < l_2 \leq l$ (corresponding to the times of different cycles of the one-bit feedback algorithm), where l_0 and $l - k$ correspond to two time instants when there was a phase update. Instant l_0 is the reference point at time l_1, and $l - k$ is the reference point at time l_2 and time l. Let us define the following terms:

$$U_{l_1} = y_{\delta n}[l_1] - y_n[l_0] \tag{18a}$$

$$U_{l_2} = y_{\delta n}[l_2] - y_n[l - k] \tag{18b}$$

$$V_l = y_\delta[l] - y[l - k] \tag{18c}$$

where $y_{\delta n}[l]$, $y_n[l]$, $y_\delta[l]$ and $y[l]$ corresponds to the different (normalized) RSSs with the phase noise and/or random perturbation applied at cycle l. We show in Appendix A-C that the covariance between the following pair of terms is equal to zero:

$$\mathrm{Cov}\left[U_{l_1}, U_{l_2}\right] = \mathrm{Cov}\left[V_l, U_{l_1}\right] = 0 \tag{19a}$$

which shows that the RSS drift at cycle l is uncorrelated with (and hence independent of, under our joint Gaussian approximation) the feedback before $l - k$.

Given the previous assumption, the *true* RSS drift (i.e. the noiseless RSS drift) conditioned on the current state S_k is given by

$$\mathrm{Drift}(RSS_l|S_k, y) = \mathbb{E}\left[V_l|\text{feedback since } l - k, y\right]$$

The total RSS drift with a memory of length K is then given by:

$$\begin{aligned}\mathrm{Drift}(RSS_l|K, y) &= \sum_{k=1}^{K} \mathbb{P}\left(S_k|y\right) \cdot \mathrm{Drift}(RSS_l|S_k, y) \\ &= \sum_{k=1}^{K} \mathbb{P}\left(S_k|y\right) \cdot \Big[\mathbb{P}(U_l > 0|S_k)\mathrm{Drift}(RSS_l|S_k, U_l > 0, y) \\ &\qquad + \mathbb{P}(U_l < 0|S_k)\mathrm{Drift}(RSS_l|S_k, U_l < 0, y)\Big]\end{aligned} \tag{20}$$

We begin by determining the probabilities $\mathbb{P}\left(S_k|y\right)$ of being in a given state k. We model the state transitions with the Markov chain shown in Figure 5. In each state, a positive

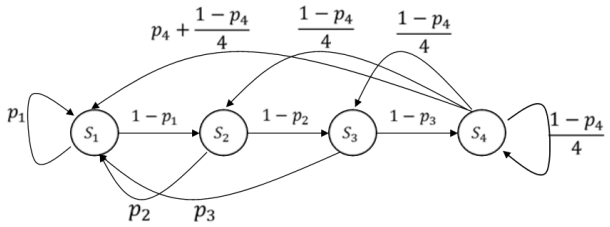


Fig. 5. Markov chain modeling the state transitions for a memory of size $K = 4$.

feedback brings the algorithm back in state S_1, whereas a

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2015.2497687, IEEE Transactions on Wireless Communications

negative feedback causes the algorithm to transition from state S_k to state S_{k+1}. For the final state S_K, we assume that negative feedback causes to algorithm to transition to any state with uniform probability. This is a pessimistic approximation that greatly simplifies our modeling, as discussed in more detail shortly. The Markov chain state transition matrix is given by (for a memory of length 4)

$$\mathbf{P} = \begin{bmatrix} p_1 & 1-p_1 & 0 & 0 \\ p_2 & 0 & 1-p_2 & 0 \\ p_3 & 0 & 0 & 1-p_3 \\ p_4 + \frac{1-p_4}{4} & \frac{1-p_4}{4} & \frac{1-p_4}{4} & \frac{1-p_4}{4} \end{bmatrix} \quad (21)$$

Under the assumption that changes in the noiseless RSS y are small between iterations (which is generally true in theory and in practice), we can approximate the probability of being in each state by looking at the stationary distribution π of the Markov chain. The stationary distribution π is the left eigenvector of $\mathbf{P}$ that correspond to the eigenvalue $\lambda = 1$:

$$\pi^T \mathbf{P} = \pi^T \quad (22)$$

which will depend on the state transition probabilities p_k and the memory length K.

We now determine the state transition probabilities p_k, which define the probability of a positive feedback at a given state S_k. The probability p_k is defined as

$$\begin{aligned} p_k &= \mathbb{P}\left[U_l > 0 | S_k; y\right] \\ &= \mathbb{P}\left[U_l > 0 | U_{l-1} < 0, ..., U_{l-k+1} < 0; y\right] \end{aligned} \quad (23)$$

Using similar arguments as in Section IV-A, we model the random variable $[U_l, U_{l-1}, ..., U_{l-k+1}]^T$ as a multivariate Gaussian random variable. The mean and variances of each term has already been computed in Section IV-A, and the covariance ρ_{UU} between two variables U_{l_1} and U_{l_2} (with $l-k < l_1 \leq l$ and $l-k < l_2 \leq l$) is equal to

$$\rho_{UU} = \frac{1 - \chi^2_{\delta n} - \rho_{\delta n}\kappa(y)}{2N} \quad (24)$$

as shown in Appendix A-D. Note that ρ_{UU} is identical for all possible values of l_1 and l_2. Once we know the mean and full covariance matrix of the multivariate Gaussian random variable $[U_l, U_{l-1}, ..., U_{l-k+1}]^T$, we can use these to determine the probabilities p_k given in (23) by computing the multivariate cumulative density function. Note that the p_k are not equal, and decrease with increasing k. To gain some intuition into this, suppose that we are in state S_4. This means that the one-bit feedback algorithm has already failed three times to find a phase combination that improved the RSS compared to the phase (and noise) combination corresponding to the noisy reference y_n. This suggests that the phase noises of the reference y_n adds up coherently, resulting in a high reference RSS which is hard to beat in later iterations. In contrast, when we are in state S_1, there is no prior information about previous trials available, and hence no information about the noisy reference RSS y_n. The success probability p_4 in state S_4 is thus smaller than the success probability p_1 in state S_1. Mathematically, the probability $p_k = \mathbb{P}\left[U_l > 0 | U_{l-1} < 0, ..., U_{l-k+1} < 0; y\right]$ decreases with increasing k because the terms $U_l, U_{l-1}, ..., U_{l-k+1}$ are correlated, resulting in a lower probability p_k.

By filling the values of p_k in (21) and (22), one can compute the steady state probabilities $\{\mathbb{P}(S_k)\}$. Comparing these against simulations in Figure 6 for $N = 10$ nodes and a memory of size $K = 4$, we see that there is a good match.

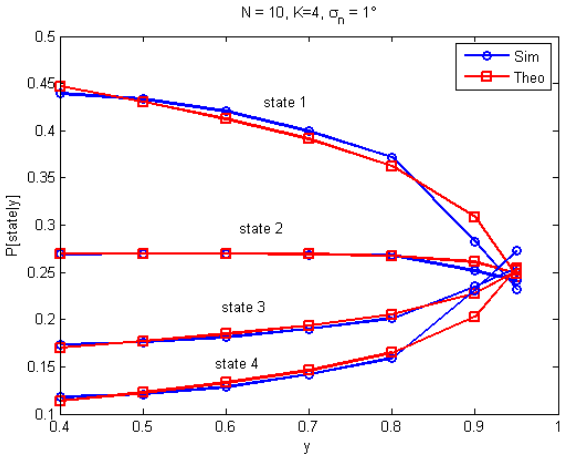


Fig. 6. Probability of being in each state, for $N = 10$, $K = 4$, $\delta \sim \mathcal{U}[-10°, 10°]$, and $\sigma_n = 1°$.

We now compute the RSS drift terms of equation (20). The drift term for positive feedback is defined as

$$\begin{aligned} &\text{Drift}(RSS_l | S_k, U_l > 0; y) = \\ &\qquad \mathbb{E}\left[V_l | U_l > 0, U_{l-1} < 0, ..., U_{l-k+1} < 0; y\right] \end{aligned} \quad (25)$$

We again consider the Gaussian multivariate distribution $[V_l, U_l, U_{l-1}, ..., U_{l-k+1}]$. The means and variances of all the terms have been computed previously, as have the covariance between U_{l_1} and U_{l_2} and the covariance ρ_{UV} between U_l and V_l. The covariance $\rho_{U_m V_l}$ is shown to be equal to 0 in Appendix A-E The covariance matrix of the multivariate Gaussian distribution is then given by

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_V^2 & \rho_{UV} & 0 & \cdots & 0 \\ \rho_{UV} & \sigma_U^2 & \rho_{UU} & \cdots & \rho_{UU} \\ 0 & \rho_{UU} & \sigma_U^2 & \cdots & \rho_{UU} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \rho_{UU} & \rho_{UU} & \cdots & \sigma_U^2 \end{bmatrix}$$

Computing the RSS drift term in (25) then boils down to determining the mean of a truncated multivariate Gaussian distribution, which can be done efficiently with Monte-Carlo integration.

For negative feedback, and for all states $k = 1, ..., K-1$, the RSS drift is equal to zero, as the one-bit feedback algorithm does not change the combination of phases:

$$\begin{aligned} &\text{Drift}(RSS_l | S_k, U_l < 0; y) = \\ &\qquad \mathbb{E}\left[V_l | U_l < 0, U_{l-1} < 0, ..., U_{l-k+1} < 0; y\right] = 0 \end{aligned} \quad (26)$$

Note that this equality does not hold for $k = K$. If $U_l < 0$, the previous maximum RSS then "slips out" of the past RSS memory, and in the next iteration a new maximum RSS will

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2015.2497687, IEEE Transactions on Wireless Communications

be considered. By definition, this will cause a change in true RSS drift, which can be expressed as

$$\begin{aligned}&\text{Drift}(RSS_l|S_K, U_l<0; y)\\ &= \mathbb{P}\left[\text{Max. noisy RSS at } l-m\right]\times\\ &\qquad \mathbb{E}\left[V_{l-m}|U_{l-m}<0, \{U_{l-n}<U_{l-m}\}_{K-1\le m\le l, n\ne m}; y\right]\\ &= \mathbb{E}\left[V_l|U_l<0, U_{l-1}<U_l, ..., U_{l-K+1}<U_t; y\right] \qquad (27)\end{aligned}$$

where this last equation is obtained by symmetry arguments. Equation (27) can be determined by yet another multivariate Gaussian distribution with variates $[V_l, U_l, U_{l-1}, ..., U_{l-K+1}]^T$. The elements of the mean and covariance matrix of this multivariate Gaussian distributions have all been computed previously, and the solution of (27) can be obtained through Monte-Carlo integral computation.

By combining the different RSS drift terms (25), (26) and (27) into equation (20), the total RSS drift can be computed. Figure 7 shows the RSS drift for various phase noise values, both with Monte Carlo simulations and with our theoretical model. It can be seen that there is a good correspondence between the simulated and theoretical curves. However, since many realizations must be averaged to get our simulation curves, the computational complexity of simulations is far higher than that of our analytical computations. From Figure 7,

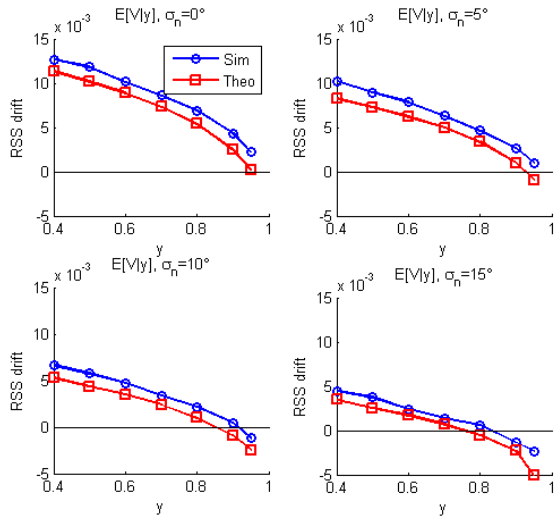


Fig. 7. Theoretical and simulated RSS drift for $N = 10$ nodes, a memory of size $K = 4$ and a random phase perturbation of $\delta \sim \mathcal{U}[-10°, 10°]$, for various phase noise values.

it can be seen that when the phase noise becomes larger than the random phase perturbation, the RSS drift eventually becomes negative. This means that in steady-state, the RSS will converge to a value of y smaller than 1, and not achieve the maximum possible RSS. For a phase noise of $\sigma_n = 15°$, the RSS will only reach 80% of the maximum achievable RSS.

Our analysis is confirmed by the simulation results in Figure 8. Here, the normalized RSS is plotted versus time when running the one-bit feedback algorithm. The normalized RSS has been averaged over 100 simulation runs, and the first 1000 iterations are not plotted in Figure 8 (in order to focus only on the steady-state convergence values). It can be seen that the normalized RSS does converge at a value predicted by the zero-crossing of the RSS drift in Figure 7. In Section V, we show that our experimental testbed starts failing when the phase noise gets larger than the random phase perturbations.

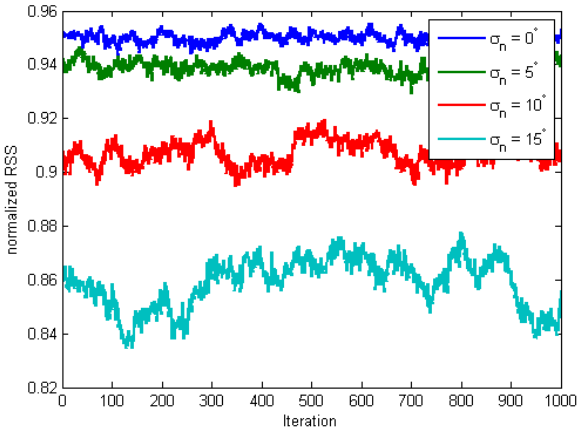


Fig. 8. Normalized RSS when running the one-bit feedback algorithm, averaged over 100 runs. It can be seen that as the phase noise increases, the steady-state average RSS decreases. The average normalized RSS can be predicted by the zero-crossing of the RSS drift in Figure 7.

Remark on approximations used: We see from Figure 7 that there is a small offset between the analytical and simulated curves, with the analytical results being slightly pessimistic (smaller RSS drift). In addition to possible inaccuracies due to the joint Gaussian approximation, one possible reason for the analysis being pessimistic is our approximate model for the transition out of the final state S_K. We assume that, when we leave state S_4 (for $K = 4$), we go back to states S_1 through S_4 with equal probability, and then use the previously computed transition probabilities to model the future evolution of the chain. This ignores the fact that the transition from S_4 to S_2 does not provide information about the prior RSS, unlike, for example, the one from S_1 to S_2. However, regardless of how we arrive at S_2, we use the same values for transition probabilities out of S_2 (based on the information on prior RSS corresponding to the S_1 to S_2 transition). We conjecture that this results in a pessimistic estimate for the probability of positive feedback, and hence to a pessimistic estimate of the RSS drift. In principle, it might be possible to introduce a new set of states, S_2', S_3' or S_4', for when we transition out of S_4, but including such extra states would significantly increase the modeling complexity. Our proposed model strikes a good balance between prediction accuracy and complexity, especially because a pessimistic approximation provides a more robust framework for design.

Our analytical framework can also be used to easily evaluate the effect of different memory sizes K (running Monte Carlo simulations for each K is quite expensive, given the large number of realizations required). Figure 9 shows the RSS drift versus K. For our model (in which all nodes are stationary), the RSS drift improves monotonically with K, but in practice, we would trade off using a memory large enough to make

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2015.2497687, IEEE Transactions on Wireless Communications

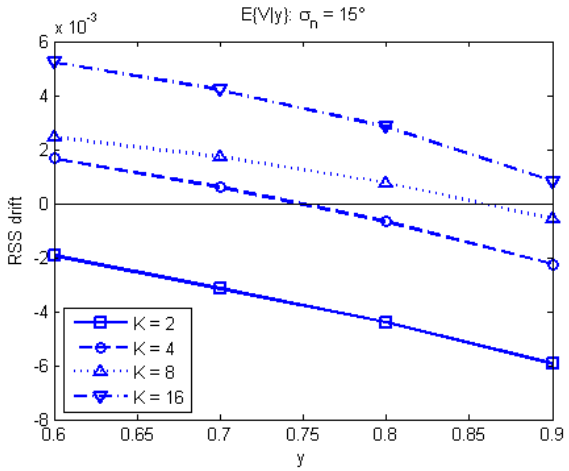


Fig. 9. RSS drift for different memory sizes K. Larger memories tend to have higher RSS drifts, but will suffer from phase drifts and node movement.

the steady-state RSS large, and small enough to enable rapid tracking of phase drift or node movement.

V. Experimental demonstration

A. Software-defined radio testbed

The proposed architecture was implemented on a software-defined radio testbed using six USRP RF and baseband boards. We use a mix of USRP-2 and USRP-N200 baseband boards, and WBX 50-2200 MHz RF daughterboards. Each USRP was connected to a host laptop that performed the computation using GNU Radio software. Our software is available for download online [34].

One USRP was used as a transmit node (sending packets that contain only a pilot tone), one USRP was used as the final receiver, and up to four USRPs were used as relay nodes. A block diagram of the relay nodes is shown in Figure 10. The relay nodes receive the message from the transmitter, add a phase shift to the received message, and wait for a fixed amount of time T_d before amplifying and forwarding the message to the final receiver (this delay T_d needs to be identical for all relays for the messages to add up without ISI). The message is forwarded over the same carrier frequency as the one used by the transmitter, which was set to 908 MHz. Additionally, the relay nodes also listen to the feedback message over the feedback channel (at 928 MHz) to determine the phase shift to be applied using the one-bit feedback algorithm. The sample rate of all the nodes was set to 200 kHz. The receiver receives the combined messages of all relay nodes, computes its single bit of feedback and broadcasts this single bit back to all relays over the feedback channel. The single bit is embedded in a GMSK-modulated packet. Note that all nodes are static in our experiments. More generally, we expect the one-bit feedback algorithm to work well if the channel phases vary slowly compared to the algorithm cycle period T_c, but additional experiments with moving nodes are left for future work. For more rapid movement, it is necessary to use other techniques (possibly in conjunction with one-bit feedback) such as reciprocity or per-node feedback. For the

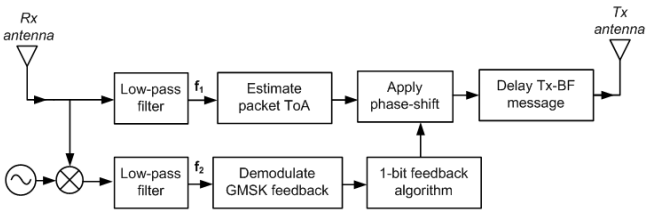


Fig. 10. Block-diagram for the relay node.

latter (which does not scale well to large node clusters), the effect of motion was considered in [32], where a three-state Kalman filter was used to track the signal phase due to both LO offset and node movement.

B. Experimental results

In this section we present results obtained with our experimental prototype. The prototype was run in an indoor environment, with a distance between transmitter and relay/receiver node of approximately 5 m. Since the links were line-of-sight, the gain of the channels between the transmitter and all relay nodes are approximately equal. The nodes were static during the experiments, and there was little movement around the testbed to limit the effects of dynamic fading. The random phase perturbation are chosen uniformly from the discrete set $\{-10°, 10°\}$. Figure 11 shows the received signal during a single cycle for two relay nodes. The relay delay time was set at T_d = 10 ms, and the cycle period at T_c = 50 ms. In subfigure (a), no relay is activated: only the message from the (distant) transmitter is observed, with low amplitude. In subfigures (b) and (c), only relay 1 or relay 2 is activated. After the original message from the transmitter, the (stronger) message from the relay can be observed. Finally, when both relays have been activated and convergence of the one-bit feedback algorithm has been achieved, it can be seen in subfigure (d) that the relayed packets from both relays add up coherently. The amplitude of the relayed packets is then equal to the sum of the amplitudes of the individual relayed packets. Note that in all figures there is a noisy signal after the relayed packets. This corresponds to the self-interference created by the receiver's feedback message to the relays, in an adjacent frequency band, and can be ignored.

Figures 12 and 13 show the mean amplitude of the relayed packets only, over longer amounts of time, using 3 relays and 4 relays, respectively. These are obtained by having the final receiver record its received signal to a data file, and by turning the different relay nodes on and off individually. The mean amplitude of the relayed packets (i.e. the second packet in Figure 11) is then measured offline. It can be seen that the amplitude of the combined relayed messages correspond to the sum of the amplitudes of the individual relayed messages. Also, it can be observed that, once the one-bit feedback algorithm achieves convergence, the amplitude of the relayed messages is stable at its maximum value. Thus, the phase errors due to LO drift are being successfully handled by the one-bit feedback algorithm. In Figure 13, the steady increase in RSS can be observed when the 4th relay is turned on. A

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2015.2497687, IEEE Transactions on Wireless Communications

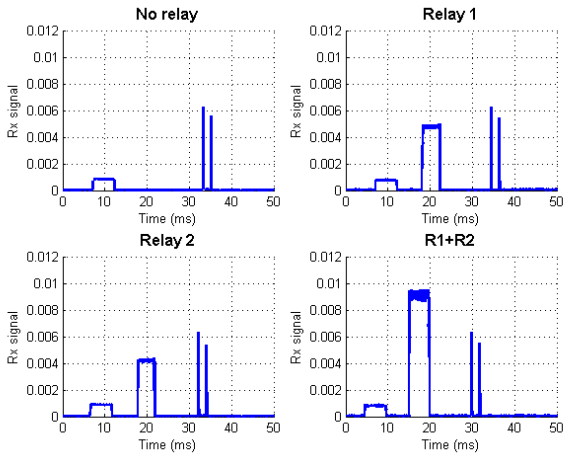


Fig. 11. Received signal during one cycle of the setup with (a) no relays, (b) relay 1, (c) relay 2 and (d) relays 1 and 2 activated.

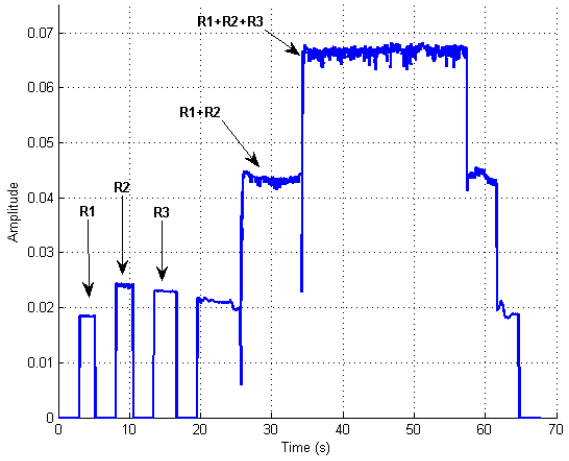


Fig. 12. Mean amplitude of the relayed packets with 3 relay nodes.

few iterations were necessary for the RSS to converge to its maximum value. The number of iterations required for the one-bit feedback algorithm to converge depends on the (random) initial state of the phases, as well as on the number of nodes involved. For four relay nodes, we observe that convergence typically occurs between 10 and 50 iterations. It can also be seen in Figures 12-13 that there are slight dips once the RSS has converged to its maximum value. This is because the one-bit feedback algorithm continues running even after the RSS has achieved its maximum value, causing the phases to misalign and realign over time. An easy improvement might be to reduce the size of the random phase perturbation applied at the relays once the RSS converges to its maximum value.

In Section III it was determined that increasing the relay delay time T_d and the cycle period T_c would result in an increasing phase error. In addition, it was argued in Section IV that if the phase error becomes large (with respect to the size of the random phase perturbation), the RSS drift becomes negative and the one-bit feedback algorithm is unable to maintain the amplitude at its theoretical maximum. To verify these

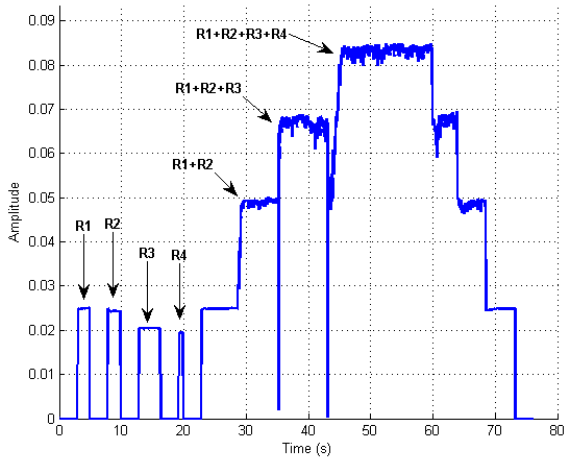


Fig. 13. Mean amplitude of the relayed packets with 4 relay nodes.

predictions, our experimental testbed was run with different values of T_c and T_d, as shown in Figure 14. The setup was run with two relay nodes, and the random phase perturbation on both relay nodes was $10°$. The LO parameters of our testbed were estimated previously [29] as $q_1^2 = 8.47 \times 10^{-22}$ and $q_2^2 = 5.51 \times 10^{-18}$ (these values correspond to the USRP internal temperature-controlled crystal oscillator (TCXO)). The corresponding phase error standard deviation, computed using (14), is given in the title of the subfigures. For each test, we first waited for a period of time long enough that the one-bit feedback algorithm could be expected to converge. The red line represents the (normalized) maximum possible RSS (based on the measured amplitudes of the relayed packets when the relays are turned on individually), and the blue line corresponds to the (normalized) measured RSS of the relayed packets when both relays are turned on, after convergence of the one-bit feedback algorithm. It can be seen that once the phase error standard deviation becomes significant with respect to that of the random phase perturbation, the one-bit feedback algorithm has trouble converging, and the RSS has trouble maintaining its maximum value. The setup was also run with higher-quality oven-controlled crystal oscillators (OCXO), with q-parameters estimated as $q_1^2 = 5.25 \times 10^{-24}$ and $q_2^2 = 1.77 \times 10^{-21}$. With this hundredfold improvement in LO quality, we were unable to get our setup to fail, even for higher values of T_d and T_c. It can be concluded that, for typical system parameters, relay-based distributed reception shows excellent robustness and stability.

VI. Conclusions

Starting from the observation that over-the-air combining using amplify-forward relaying provides a scalable approach to distributed receive beamforming, we have proposed an architecture for achieving the synchronization required for the relayed signals to cohere at the receiver. An attractive feature of the time division (between long and short links) approach considered here is that frequency synchronization comes for free. We have demonstrated this architecture using a software-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2015.2497687, IEEE Transactions on Wireless Communications

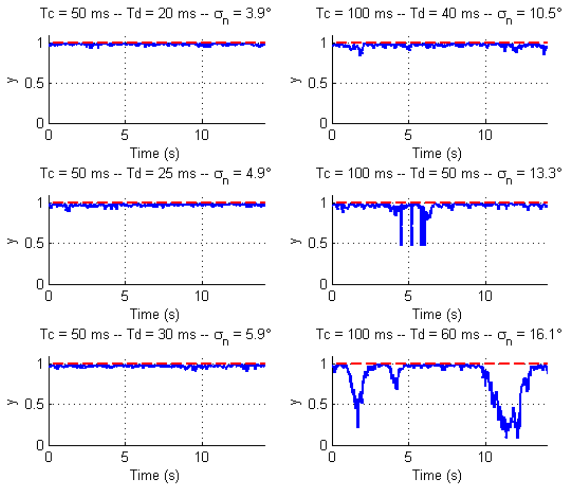


Fig. 14. Mean amplitude of the relayed packets when varying T_d and T_c.

defined radio testbed, and report experimental results achieving the receive beamforming gains predicted by theory. We also model and analyze the potential performance degradation due to phase errors accumulating due to LO drift. We provide an analytical framework, verified via Monte Carlo simulations, which estimates the degradation of the RSS attained by the one-bit feedback algorithm with finite memory in the presence of phase errors. A key insight, also verified experimentally, is that significant performance degradation occurs if the variance of the phase noise is comparable to, or larger than, the variance of the random phase perturbation used in the one-bit feedback algorithm. This provides guidance on choice of system parameters such as LO quality, relaying delay, and cycle length. The open-source implementation of our prototype is publicly available, and hopefully provides a starting point for further implementation of solutions for distributed MIMO.

There are many directions for future work. An important topic is generalization of our amplify-forward approach to provide scalable distribution reception over wideband dispersive channels. Possible approaches include "filter-and-forward," or amplify-forward on a per-subcarrier basis. Design challenges include timing synchronization and tracking schemes, and the development of parsimonious feedback strategies. Also, while our time division architecture yields implicit frequency synchronization, there may be many scenarios in which frequency division between long and short links is an attractive design choice, in which case explicit frequency synchronization is required. Finally, it is important to develop and evaluate designs that account for mobile nodes, possibly with different models addressing different potential applications.

APPENDIX A
JOINT PROBABILITY DISTRIBUTION OF U AND V

It was shown in [10] that for large N, the net effect of a random phase perturbation on the total signal can be modeled as shown in Figure 15. The effect of phase noise (or random phase perturbations plus phase noise) on the total signal can

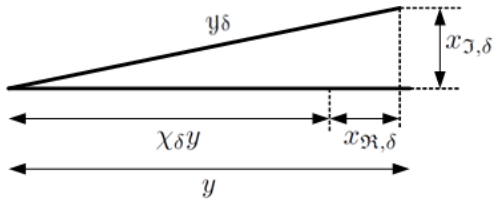


Fig. 15. Effect of a random phase perturbation on the total received signal

be modeled in an identical manner. Using equation (22) in [10], for large N and y, the following approximation can then be made:

$$y_\delta \approx \chi_\delta y + x_{\Re,\delta} \tag{28}$$

$$y_n \approx \chi_n y + x_{\Re,n} \tag{29}$$

$$y_{\delta n} \approx \chi_{\delta n} y + x_{\Re,\delta n} \tag{30}$$

The variables $x_{\Re,\delta}$, $x_{\Re,n}$ and $x_{\Re,\delta n}$ are zero-mean Gaussian random variables with variances $\sigma^2_{\Re,\delta} = \frac{1-\chi^2_\delta-\rho_\delta\kappa(y)}{2N}$, $\sigma^2_{\Re,n} = \frac{1-\chi^2_n-\rho_n\kappa(y)}{2N}$ and $\sigma^2_{\Re,\delta n} = \frac{1-\chi^2_{\delta n}-\rho_{\delta n}\kappa(y)}{2N}$, respectively. It can be seen that the joint statistics of the random variables $U = y_{\delta n} - y_n$ and $V = y_\delta - y$ are simply those of a bivariate Gaussian distribution, entirely characterized by the means of U and V, the variances of U and V, and the covariance between U and V. The development in the following subsections are similar to the development made in [10]-Appendix C.

A. Mean and variance of U and V

From Figure 15, we can define the following terms

$$x_{\delta n} = x_{\Re,\delta n} + j x_{\Im,\delta n} = \frac{1}{N}\sum_{i=1}^{N} e^{j\phi_i}\left(e^{j(\delta_i+n'_i)} - \chi_{\delta n}\right)$$

$$x_n = x_{\Re,n} + j x_{\Im,n} = \frac{1}{N}\sum_{i=1}^{N} e^{j\phi_i}\left(e^{jn_i} - \chi_n\right)$$

$$x_\delta = x_{\Re,\delta} + j x_{\Im,\delta} = \frac{1}{N}\sum_{i=1}^{N} e^{j\phi_i}\left(e^{j\delta_i} - \chi_\delta\right)$$

The mean and variance of $U = y_{\delta n} - y_n$ are then given by

$$\mathbb{E}[U] = (\chi_{\delta n} - \chi_n)y$$

$$\begin{aligned}\mathrm{Var}[U] &= \mathbb{E}\left[(x_{\Re,\delta n} - x_{\Re,n})^2\right] \\ &= \frac{1}{4}\mathbb{E}\Big[x^2_{\delta n} + x^{*2}_{\delta n} + x^2_n + x^{*2}_n + 2x_{\delta n}x^*_{\delta n} + 2x_n x^*_n \\ &\quad - 2x_{\delta n}x_n - 2x_{\delta n}x^*_n - 2x^*_{\delta n}x_n - 2x^*_{\delta n}x^*_n\Big]\end{aligned}$$

where * denotes the complex conjugate. The first term of the previous equation is given by

$$\begin{aligned}&\mathbb{E}\left[x^2_{\delta n}\right] \\ &= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{l=1}^{N}\mathbb{E}\left[e^{j(\phi_i+\phi_l)}\left(e^{j(\delta_i+n'_i)} - \chi_{\delta n}\right)\left(e^{j(\delta_l+n'_l)} - \chi_{\delta n}\right)\right]\end{aligned}$$

Since the phase perturbations δ_i and noise terms n'_i have a symmetric distribution, it follows that the terms $(e^{j(\delta_i+n'_i)} - \chi_{\delta n})$ and $(e^{j(\delta_l+n'_l)} - \chi_{\delta n})$ have zero mean. Moreover, since the phase perturbations δ_i and noise terms n'_i are independent

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2015.2497687, IEEE Transactions on Wireless Communications

for different nodes, the term in the sum are zero when $i \neq l$ and the previous becomes

$$\mathbb{E}\left[x_{\delta n}^2\right] = \frac{1}{N^2}\sum_{i=1}^{N}\mathbb{E}\left[e^{j2\phi_i}(e^{j(\delta_i+n_i')} - \chi_{\delta n})^2\right] = -\frac{1}{N^2}\rho_{\delta n}\sum_{i=1}^{N}\mathbb{E}\left[e^{j2\phi_i}\right]$$

where we used the approximation that, for small phase perturbations and/or phase noises, $\mathbb{E}\left[e^{j(\delta_i+n_i')}\right] \approx \mathbb{E}\left[\cos(\delta_i + n_i')\right]$. Similarly, one can compute

$$\mathbb{E}\left[x_{\delta n}x_{\delta n}^*\right] = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{l=1}^{N}\mathbb{E}\Big[e^{j(\phi_i-\phi_l)}(e^{j(\delta_i+n_i')} - \chi_{\delta n})(e^{-j(\delta_l+n_l')} - \chi_{\delta n})\Big] = \frac{1}{N}(1-\chi_{\delta n}^2)$$

since $\mathbb{E}\left[\left|e^{j(\delta_i+n_i')} - \chi_{\delta n}\right|^2\right] = 1 - \chi_{\delta n}^2$ (see [10]). Following similar arguments, one can easily obtain the following expressions

$$\mathbb{E}\left[x_{\delta n}^{*2}\right] = -\frac{1}{N^2}\rho_{\delta n}\sum_{i=1}^{N}\mathbb{E}\left[e^{-j2\phi_i}\right]$$

$$\mathbb{E}\left[x_n^2\right] = \mathbb{E}\left[x_n^{*2}\right]^* = -\frac{1}{N^2}\rho_n\sum_{i=1}^{N}\mathbb{E}\left[e^{j2\phi_i}\right]$$

$$\mathbb{E}\left[x_n x_n^*\right] = \frac{1}{N}(1-\chi_n^2)$$

The crossed terms are computed as

$$\mathbb{E}\left[x_{\delta n}x_n\right] = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{l=1}^{N}\mathbb{E}\left[e^{j(\phi_i+\phi_l)}(e^{j(\delta_i+n_i')} - \chi_{\delta n})(e^{j(n_l)} - \chi_n)\right]$$

The terms $(e^{j(\delta_i+n_i')} - \chi_{\delta n})$ and $(e^{j(n_l)} - \chi_n)$ are both zero-mean and independent variables, and therefore all terms in the summation equal zero, which leads to $\mathbb{E}\left[x_{\delta n}x_n\right] = \mathbb{E}\left[x_{\delta n}x_n^*\right] = \mathbb{E}\left[x_{\delta n}^*x_n\right] = \mathbb{E}\left[x_{\delta n}^*x_n^*\right] = 0$ The terms $\kappa(y) = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}[e^{2j\phi_i}]$ depends on y only, and can be approximated by $\kappa(y) = e^{-4(1-y)}$ for large y. The latter was derived in Conjecture 1 in [10], where statistical mechanics arguments where used to determine that at each cycle of the algorithm, the distribution of the phases the nodes ϕ_i (which includes the accumulated phase due to the one-bit feedback algorithm in the previous cycles) follow an Exp-Cosine distribution around their mean. The variance of U can finally be written as

$$\mathrm{Var}[U] = \frac{1-\chi_{\delta n}^2 - \rho_{\delta n}\kappa(y)}{2N} + \frac{1-\chi_n^2-\rho_n\kappa(y)}{2N}$$

The statistics of $V = y_\delta - y$ can be deduced in a manner identical to the statistics of U, leading to:

$$\mathbb{E}[V] = (\chi_\delta - 1)y$$

$$\mathrm{Var}[V] = \frac{1-\chi_\delta^2-\rho_\delta\kappa(y)}{2N}$$

B. Covariance between U and V

The two terms U and V are not independent. The covariance between these two terms can be computed using similar arguments as before, giving the following result:

$$\begin{aligned}\mathrm{Cov}[U,V] &= \mathbb{E}[(x_{\Re,\delta n} - x_{\Re,n})x_{\Re,\delta}] \\ &= \frac{1}{4}\mathbb{E}\Big[x_{\delta n}x_\delta + x_{\delta n}x_\delta^* + x_{\delta n}^*x_\delta + x_{\delta n}^*x_\delta^* - x_nx_\delta \\ &\qquad - x_nx_\delta^* - x_n^*x_\delta - x_n^*x_\delta^*\Big] \\ &= \frac{\chi_n - \chi_{\delta n}\chi_\delta - \rho_{2\delta n}\kappa(y)}{2N}\end{aligned}$$

where $\rho_{2\delta n} = \chi_{\delta n}\chi_\delta - \mathbb{E}[\cos(2\delta_i + n_i)]$.

C. Covariance between U_{l_1} and U_{l_2} and covariance between V_l and U_{l_1}

Let us consider the following discrete time instants $l_0 < l_1 < l-k < l_2 \leq l$, where l_0 and $l-k$ correspond to two cycles of the one-bit feedback algorithm when there was a phase update. Cycle l_0 is the reference point at cycle l_1, and $l-k$ is the reference point at cycles l_2 and l. The terms U_{l_1}, U_{l_2} and V_l are defined in (18a)-(18c). Since the random phase perturbations and phase noises are independent across iterations, we can show that the covariance between U_{l_1} and U_{l_2} and the covariance between V_l and U_{l_1} is equal to

$$\begin{aligned}&\mathrm{Cov}\left[U_{l_1}, U_{l_2}\right] \\ &\quad= \mathbb{E}\left[\left(x_{\Re,\delta n}[l_1] - x_{\Re,n}[l_0]\right)\left(x_{\Re,\delta n}[l_2] - x_{\Re}[l-k]\right)^*\right] \\ &\quad= 0\end{aligned}$$

$$\mathrm{Cov}\left[V_l, U_{l_1}\right] = \mathbb{E}\left[x_{\Re,\delta}[l]\left(x_{\Re,\delta n}[l_1] - x_{\Re,n}[l_0]\right)^*\right] = 0$$

which shows that the RSS drift at cycle l is independent of the feedback before $l-k$.

D. Covariance between U_{l_1} and U_{l_2} for $l-k < l_1 < l$ and $l-k < l_2 < l$

The covariance ρ_{UU} between two variables U_{l_1} and U_{l_2} (with $l-k < l_1 \leq l$ and $l-k < l_2 \leq l$) is given by (using the same notations as previously)

$$\begin{aligned}\rho_{UU} &= \mathbb{E}\left[\left(x_{\Re,\delta n}[l_1] - x_{\Re,n}[l-k]\right)\left(x_{\Re,\delta n}[l_2] - x_{\Re,n}[l-k]\right)\right] \\ &= \frac{1-\chi_n^2-\rho_n\kappa(y)}{2N}\end{aligned}$$

This is due to the fact that the random phase perturbation and noise at cycles l_1 and l_2 are independent.

E. Covariance between V_l and U_m for $l-k < m < l$

The covariance $\rho_{U_mV_l}$ between V_l and U_m is given by (for $l-k < m < l$)

$$\rho_{U_m,V_l} = \mathbb{E}\left[\left(x_{\Re,\delta n}[m] - x_{\Re,n}[l-k]\right)x_{\Re,\delta}[l]\right] = 0$$

which is due to the fact that the noise and/or random phase perturbation at cycles m and $l-k$ are independent of the random phase perturbation at cycle l.

ACKNOWLEDGEMENTS

This work is funded in part by the US National Science Foundation under grant CCF-1302114, and by the Institute for Collaborative Biotechnologies through grant W911NF-09-0001 from the U.S. Army Research Office. The content of the information does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

REFERENCES

[1] T. Cover and A. El Gamal, "Capacity theorems for the relay channel," IEEE Transactions on Information Theory, vol. IT-25, no. 5, pp. 572–584, 1979.

[2] A. El Gamal and T. Cover, "Multiple user information theory," Proceedings of the IEEE, vol. 68, no. 12, pp. 1466–1483, 1980.

[3] S. Borade, L. Zheng, and R. Gallager, "Amplify-and-forward in wireless relay networks: Rate, diversity, and network size," Information Theory, IEEE Transactions on, vol. 53, no. 10, pp. 3302 –3318, Oct. 2007.

[4] S. Jin, M. McKay, C. Zhong, and K.-K. Wong, "Ergodic capacity analysis of amplify-and-forward MIMO dual-hop systems," Information Theory, IEEE Transactions on, vol. 56, no. 5, pp. 2204 –2224, May 2010.

[5] S. Gharan, A. Bayesteh, and A. Khandani, "Asymptotic analysis of amplify and forward relaying in a parallel MIMO relay network," Information Theory, IEEE Transactions on, vol. 57, no. 4, pp. 2070 –2082, April 2011.

[6] Y.-S. Tu and G. Pottie, "Coherent cooperative transmission from multiple adjacent antennas to a distant stationary antenna through AWGN channels," in Vehicular Technology Conference, 2002. VTC Spring 2002. IEEE 55th, vol. 1, 2002, pp. 130 – 134 vol.1.

[7] Z. Ding, W. Chen, and K. Leung, "Distributed beamforming and power allocation for cooperative networks," Wireless Communications, IEEE Transactions on, vol. 7, no. 5, pp. 1817–1822, 2008.

[8] R. Mudumbai, J. Hespanha, and U. Madhow, "Scalable feedback control for distributed beamforming in sensor networks," in Proc. 2005 IEEE International Symposium on Information Theory (ISIT 2005), 2005.

[9] R. Mudumbai, B. Wild, U. Madhow, and K. Ramch, "Distributed beamforming using 1 bit feedback: From concept to realization," in in Allerton Conference on Communication, Control, and Computing, 2006.

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

[11] R. Mudumbai, G. Barriac, and U. Madhow, "On the feasibility of distributed beamforming in wireless networks," Wireless Communications, IEEE Transactions on, vol. 6, no. 5, pp. 1754 –1763, May 2007.

[12] I. Ozil and D. Brown, "Time-slotted round-trip carrier synchronization," in Signals, Systems and Computers, 2007. ACSSC 2007. Conference Record of the Forty-First Asilomar Conference on, Nov. 2007, pp. 1781 –1785.

[13] D.R. Brown III and H. Poor, "Time-slotted round-trip carrier synchronization for distributed beamforming," Signal Processing, IEEE Transactions on, vol. 56, no. 11, pp. 5630 –5643, Nov. 2008.

[14] R. Preuss and D.R. Brown III, "Retrodirective distributed transmit beamforming with two-way source synchronization," in Proc. of the 44th Annual Conference on Information Sciences and Systems (CISS 2010), 2010.

[15] ——, "Two-way synchronization for coordinated multicell retrodirective downlink beamforming," Signal Processing, IEEE Transactions on, vol. 59(11), pp. 5415–5427, 2011.

[16] R. Mudumbai, D. Brown, U. Madhow, and H. Poor, "Distributed transmit beamforming: challenges and recent progress," Communications Magazine, IEEE, vol. 47, no. 2, pp. 102 –110, February 2009.

[17] R. Mudumbai, U. Madhow, D. Brown, and P. Bidigare, "DSP-centric algorithms for distributed transmit beamforming," in IEEE Asilomar Conference on Signals, Systems and Computers, 2011, 2011.

[18] R. Irmer, H. Droste, P. Marsch, M. Grieger, G. Fettweis, S. Brueck, H.-P. Mayer, L. Thiele, and V. Jungnickel, "Coordinated multipoint: Concepts, performance and field trial results," Communications Magazine, IEEE, vol. 49, no. 2, pp. 102–111, 2011.

[19] D.R. Brown III, U. Madhow, M. Ni, M. Rebholz, and P. Bidigare, "Distributed reception with hard decision exchanges," Wireless Communications, IEEE Transactions on, vol. 13, no. 6, pp. 3406–3418, June 2014.

[20] M.-O. Pun, D. Brown, and H. Poor, "Opportunistic collaborative beamforming with one-bit feedback," Wireless Communications, IEEE Transactions on, vol. 8, no. 5, pp. 2629 –2641, may 2009.

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

[22] K. Zarifi, S. Zaidi, S. Affes, and A. Ghrayeb, "A distributed amplify-and-forward beamforming technique in wireless sensor networks," Signal Processing, IEEE Transactions on, vol. 59, no. 8, pp. 3657 –3674, aug. 2011.

[23] G. Zheng, K.-K. Wong, A. Paulraj, and B. Ottersten, "Collaborative-Relay Beamforming With Perfect CSI: Optimum and Distributed Implementation," Signal Processing Letters, IEEE, vol. 16, no. 4, pp. 257 –260, april 2009.

[24] ——, "Robust collaborative-relay beamforming," Signal Processing, IEEE Transactions on, vol. 57, no. 8, pp. 3130 –3143, aug. 2009.

[25] F. Quitin, A. Irish, and U. Madhow, "Distributed receive beamforming: a scalable architecture and its proof of concept," in 77th IEEE Vehicular Technology Conference (VTC-spring 2013), 2013.

[26] S. Munkyo, M. Rodwell, and U. Madhow, "A feedback-based distributed phased array technique and its application to 60-GHz wireless sensor network," in Microwave Symposium Digest, 2008 IEEE MTT-S International, June 2008, pp. 683 –686.

[27] S. Sigg and M. Beigl, "Algorithms for closed-loop feedback based distributed adaptive beamforming in wireless sensor networks," in Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2009 5th International Conference on, Dec. 2009, pp. 25 –30.

[28] M. M. U. Rahman, H. E. Baidoo-Williams, S. Dasgupta, and R. Mudumbai, "Fully wireless implementation of distributed beamforming on a software-defined radio platform," in the 11th ACM/IEEE Conference on Information Processing in Sensor Networks (IPSN 2012), 2012.

[29] F. Quitin, M. M. U. Rahman, R. Mudumbai, and U. Madhow, "A scalable architecture for distributed transmit beamforming with commodity radios: Design and proof of concept," Wireless Communications, IEEE Transactions on, vol. 12, no. 3, pp. 1418–1428, 2013.

[30] P. Bidigare, M. Oyarzyn, D. Raeman, D. Chang, D. Cousins, R. O'Donnell, C. Obranovich, and D. Brown, "Implementation and demonstration of receiver-coordinated distributed transmit beamforming across an ad-hoc radio network," in Signals, Systems and Computers (ASILOMAR), 2012 Conference Record of the Forty Sixth Asilomar Conference on, 2012, pp. 222–226.

[31] H. V. Balan, R. Rogalin, A. Michaloliakos, K. Psounis, and G. Caire, "Achieving high data rates in a distributed MIMO system," in Proceedings of the 18th annual international conference on Mobile computing and networking, ser. Mobicom '12. New York, NY, USA: ACM, 2012, pp. 41–52. [Online]. Available: http://doi.acm.org/10.1145/2348543.2348552

[32] D.R. Brown III, P. Bidigare, and U. Madhow, "Receiver-coordinated distributed transmit beamforming with kinematic tracking," in Proceedings of the 37th International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2012, to appear.

[33] C. Zucca and P. Tavella, "The clock model and its relationship with the Allan and related variances," Ultrason., Ferroelect, Freq. Contr., IEEE Tranasctions on, vol. 52, no.2, pp. 289–296, 2005.

[34] [Online]. Available: http://www.ece.ucsb.edu/~fquitin/.