Distributed receive beamforming: a scalable architecture and its proof of concept

François Quitin, Andrew Irish and Upamanyu Madhow
Electrical and Computer Engineering, University of California, Santa Barbara

***Abstract*—We propose and demonstrate a scalable architecture for distributed receive beamforming. In a receive cluster of $N + 1$ nodes receiving a message from a distant transmitter, N nodes are designated as amplify-and-forward relays and one node is designated as the receiver. The relay nodes apply a phase shift to their received signal and forward it such that their forwarded signals add up constructively at the receiver, with received SNR scaling linearly with N. This approach transforms a distributed receive beamforming problem on the "long link" from transmitter to receive cluster into a distributed transmit beamforming problem on the "short link" from relays to receiver, so that the number of degrees of freedom on the short link need not scale with N. A key simplification relative to distributed transmit beamforming is that, for stable oscillators, relay frequency synchronization is not required. For oscillators with drift, we provide a simple rule of thumb for when explicit frequency synchronization can be avoided. Explicit timing alignment can also be avoided by exploiting the timing of the message received on the long link. That leaves the problem of phase adjustment at the relays, and we employ an algorithm originally invented for distributed transmit beamforming for this purpose, using one bit (per iteration) feedback from the receiver. Experimental results with software-defined radios (whose oscillators have significant drift) demonstrate that the expected gains in received signal strength are obtained with the proposed architecture.**

I. INTRODUCTION

Distributed receive (D-Rx) beamforming refers to collaborative communication in which a cluster of nodes coherently combine their received messages to emulate a virtual antenna array directing a beam towards the transmitter. In a centralized receive array, signals from different receive antennas are routed along wires, with phase shifts for coherent combining applied at RF or IF, or digitally at baseband, after downconversion and analog-to-digital conversion. A directly analogous approach for distributed receive beamforming is for each node to send its received signal to a centralized processor, typically via a fast local wireless link. The centralized processor then applies the appropriate phase shifts to achieve receive beamforming. However, such an approach does not scale to a large number of cooperating nodes, since the amount of local communication is proportional to the number of nodes. In this paper, we propose and demonstrate, using a software-defined radio testbed, a novel architecture for distributed reception which achieves scalability by employing on-air coherent combining.

We consider a receive cluster of $N + 1$ nodes receiving a message over a "long link" from a distant transmitter; N of these nodes act as amplify-and-forward *relays,* while one node acts as the *receiver,* or processor, node. Each relay applies a phase shift to its received signal before forwarding it on the "short link" to the receiver, so that the relayed signals add up constructively at the receiver. The received SNR with this approach scales linearly with N. However, since the signals are being combined in the air using the same time-frequency slice, the number of degrees of freedom used over the short link does not increase with N. We have achieved this scalability by transforming the distributed receive beamforming problem on the long link into a distributed transmit beamforming (D-Tx) problem on the short link. Distributed coherent combining requires the solution of challenging synchronization problems, and this transformation also enables us to leverage significant recent advances in synchronization techniques for D-Tx, while exploiting features specific to distributed reception to simplify the design.

Contributions: The main contributions of this paper are summarized as follows:

1) **Simple, scalable architecture:** While we achieve scalability by transforming D-Rx to D-Tx, we observe that, unlike for actual D-Tx (which occurs over a single link from transmitters to receiver), we can avoid explicit frequency synchronization: the effects of LO frequency offsets at the relays "cancel out" on the long and short links. We also avoid explicit timing synchronization by aligning the timing of the relayed signals based on the timing of messages received on the long link. Thus, it is only necessary to adjust the phase at each relay, and this is achieved by a one-bit feedback algorithm originally invented for D-Tx. The phases obtained using these algorithms automatically compensate for phase shifts over both the long and short links.
2) **Design with oscillator drift:** Explicit frequency synchronization can be avoided completely for stable relay oscillators which do not drift between the time a signal is received on the long link, and the time at which it is relayed on the short link. Unfortunately, the oscillators for the low-cost radios in our testbed exhibit considerable drift. We therefore develop an analytical rule of thumb for allowable values of delay in relaying (and hence packet length), as a function of oscillator drift parameters, so that explicit frequency synchronization can be avoided with minimal degradation in beamforming performance.
3) **Demonstration:** The proposed architecture is implemented with software-defined radios, and experimental results show that the expected increase in signal amplitude is achieved despite the low quality oscillators in

these radios.

Related work: Most prior work on distributed MIMO has focused on distributed *transmit* beamforming [1, 2], with early work focusing on phase synchronization [3, 4], assuming synchronized oscillators, and recent experimental prototypes with software-defined radios addressing both frequency and phase synchronization [5–7]. We heavily leverage this progress in D-Tx, especially the one-bit feedback algorithm [8], which is a simple, scalable randomized ascent algorithm for distributed phase adjustment. There continue to be many advances in D-Tx (we do not cite these due to lack of space); by virtue of our transformation from D-Rx to D-Tx, we anticipate that these will lead to corresponding advances in D-Rx.

Amplify-and-forward relay has received extensive attention in the literature: [9] proposes collaborative beamforming in which the receiver broadcasts a single bit of feedback to each relay node indicating whether it should participate in communication (the goal being to select relays whose signals happen to be combining quasi-coherently); [10] considers network beamforming with perfect channel information; [11] approximates beamforming weights based on local information; [12, 13] propose robust collaborative relay beamforming scheme based on imperfect channel state information. However, none of this prior work addresses the fundamental problem of synchronization, or explicitly defines mechanisms for obtaining channel state information. To the best of our knowledge, this is the first work which addresses these issues in detail within a scalable architecture for D-Rx, and to provide an experimental demonstration.

II. D-RX BEAMFORMING ARCHITECTURE

Our D-Rx beamforming architecture is depicted in Figure 1. We use the same frequency band for the long and short links, and share it using a time division scheme: in the odd time slots, the transmitter broadcasts its message to the relays; in the even time slots, the relays amplify, add a phase shift and forward the received message to the receiver. The relay phase shifts for constructive combining at the receiver are determined using feedback from the receiver over a separate feedback channel. For the relayed messages to combine coherently at the receiver, three types of synchronization must be achieved: frequency, phase and time synchronization.

Frequency synchronization implies that the packets arriving at the receiver from the different relay nodes must have the same carrier frequency (otherwise the received message will exhibit fading due to constructive and destructive interference), meaning there must be no LO frequency offset between the packets arriving through different relays. As shown in Section IV, for our time division scheme, we do not need explicit frequency synchronization if the packet duration and relaying delay are chosen appropriately (depending on oscillator drift parameters).

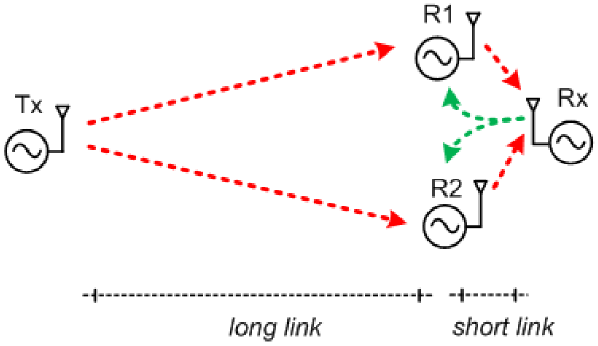

Fig. 1. D-Rx beamforming architecture.

For phase synchronization, we employ a one bit feedback algorithm [8], which has been shown to converge to an optimum both theoretically and experimentally: in each cycle, each relay adds a random phase perturbation to its current phase. The receive node monitors the received signal strength (RSS) of the received message, and sends back a single bit indicating whether the RSS has increased compared to the previous cycle. If the RSS has increased, each relay keeps its previous random phase perturbation; if not, each relay discards its previous perturbation. It is proven in [8] that the RSS converges to its maximum value, and numerous prototypes of D-Tx using this algorithm have been reported in the literature.

The third type of synchronization is time synchronization: the symbols in the relayed packets should align at the receiver in order to avoid inter-symbol interference (ISI). In this paper, we consider a narrowband setting in which differences in propagation delays across relays are small compared to the inverse bandwidth, and hence translate to phase shifts alone. We also assume that we can accurately determine packet boundaries on the long link. Thus, we can employ implicit timing alignment, in which each relay forwards the message a fixed delay after having received it. For wideband signals (not considered here), we would need more sophisticated timing alignment strategies, and if the channels are dispersive, we may have to perform more sophisticated precoding strategies at the relays. We would also need more sophisticated approaches if the SNR of the signals received at the relays is too low for them to detect packet boundaries.

III. SNR GAIN

In this section, we analyze the SNR improvement using our D-Rx beamforming architecture. Ignoring oscillator drift, we may assume, without loss of generality, that there is no LO frequency offset between the nodes (see Section IV). For a transmitted symbol $t[n]$ (normalized as $\mathcal{E}\{|t[n]|^2\} = 1$), the symbol received at relay i is given by

$$r_i[n] = h_{i1}t[n] + n_i[n] \tag{1}$$

where h_{i1} is the channel between the transmitter and relay i, and n_i is the noise at relay i. The relay nodes add a gain, a phase shift and forward the message to the receiver. Assuming ideal timing alignment, the received symbol is given by

$$r_r[n] = \sum_{i=1}^{N} h_{i2}A_i\left(h_{i1}t[n] + n_i[n]\right) + n_r[n] \tag{2}$$

where A_i contains both the gain and phase shift added by the relay, h_{i2} is the channel between relay i and the receiver, and n_r is the noise at the receiver. The SNR at the receiver is given by

$$\text{SNR} = \frac{\mathcal{E}\left\{\left|\sum_{i=1}^{N} A_i h_{i1} h_{i2} t[n]\right|^2\right\}}{\mathcal{E}\left\{\left|n_r[n] + \sum_{i=1}^{N} h_{i2} A_i n_i[n]\right|^2\right\}} \tag{3}$$

Assuming that coherence is achieved at the receiver, the terms in the numerator of (3) add up in magnitude. Setting the noise variance to N_0 for all nodes, and assuming that the noise at different nodes is uncorrelated, (3) then simplifies to

$$\text{SNR} = \frac{\mathcal{E}\left\{\left(\sum_{i=1}^{N} |A_i| \cdot |h_{i1}| \cdot |h_{i2}||t[n]|\right)^2\right\}}{\left(1 + \sum_{i=1}^{N} |h_{i2}|^2 |A_i|^2\right) N_0} \tag{4}$$

Assume for simplicity that all the channel gains from transmitter to relays are equal, that all channel gains from relays to receiver are equal, and that all the relays have identical gain:

$$|h_{11}|^2 = |h_{21}|^2 = \ldots = |h_{N1}|^2 = g_1$$
$$|h_{12}|^2 = |h_{22}|^2 = \ldots = |h_{N2}|^2 = g_2$$
$$|A_1|^2 = |A_2|^2 = \ldots = |A_N|^2 = g_r$$

The SNR in (4) then simplifies to

$$\text{SNR} = \frac{N^2 g_r g_1 g_2}{(1 + N g_r g_2) N_0} \tag{5}$$

If the transmitter were communicating directly with the receiver, the SNR would be $\text{SNR}_0 \approx g_1/N_0$ (assuming the Tx-Rx distance is approximately equal to the Tx-relay distance). Equation (5) can then be rewritten as

$$\text{SNR} \approx \frac{N^2 g_r g_2}{1 + N g_r g_2} \text{SNR}_0 \tag{6}$$

If the relay gain is able to largely compensate the power loss of the short link ($g_r g_2 \gg 1$), (6) simplifies to

$$\text{SNR} \approx N \cdot \text{SNR}_0 \tag{7}$$

showing that the SNR attained scales linearly with the number of relays.

IV. FREQUENCY SYNCHRONIZATION

A key observation: A fundamental advantage of an amplify-and-forward architecture for D-Rx beamforming is that, in theory, frequency synchronization is not required. Denoting the LO frequency offset between the transmitter and receiver by f_0, and the LO frequency offset between the transmitter and relay i by f_{i1}, the LO frequency offset between relay node i and receiver is given by $f_{i2} = f_0 - f_{i1}$. Since the relayed message incurs LO frequency offset f_{i1} on the long link and LO frequency offset f_{i2} on the short link, the relay frequency offsets cancel out, so that net LO frequency offset of the corresponding received message equals f_0, independent of the identity of the relay.

The reality could be somewhat different if the relay LOs exhibit significant drift. Suppose the transmitter sends a packet at time t_0, which is forwarded by the relays at time t_1 (as shown in Figure 2). We know that the LO frequency offset satisfies the following condition:

$$f_{i1}(t_0) + f_{i2}(t_0) = f_{j1}(t_0) + f_{j2}(t_0) = f_0(t_0) \qquad \forall i,j \tag{8}$$

However, if the relaying delay T_{delay} is large, the LO frequency offset between different relay nodes and the receiver drift independently of each other, so that $f_{i1}(t_0) + f_{i2}(t_1) \neq f_{j1}(t_0) + f_{j2}(t_1)$. The packets from different relays can now have different frequency offsets, causing fading in the received message. We now derive a simple rule of thumb for choosing T_{delay} such that we can avoid frequency synchronization without causing excessive performance degradation.

The effect of oscillator drift: In the following discussion, index $i1$ denotes the link from transmitter to relay i, index $i2$ denotes the link from relay i to receiver, and index i denotes the link from transmitter to receiver through relay i. Figure 2 shows the received amplitude at the receiver node. At time t_1, the receiver starts receiving the sum of the relayed packets. Due to lack of space, we do not discuss the impact of oscillator drift on the one bit feedback algorithm (suffice it to say that the algorithm continues to work if the randomized phase perturbations driving the algorithm are chosen large enough to overcome the phase noise due to oscillator drift). We therefore consider an initial condition where phase synchronization has been achieved when a packet is being received at a relay, and set $\phi_i(t_0) \equiv 0$ for all i (taking the initial condition at t_0 as our phase reference, without loss of generality). Due to the LO drift at the relays, these phases are no longer aligned over the relayed packet, which is sent over the interval $[t_1, t_2]$. Let us analyze the phase drift between the time $t_0 + \tau$ that a signal sample is received at the relay on the long link, and the time $t_1 + \tau$ that it is relayed on the short link to the receiver, where $0 \leq \tau \leq T_{packet}$. For simplicity, set $\tau = 0$, and consider the transmit beamforming gain over the short link at time t_1: $G = |\sum_{i=1}^{N} e^{j\phi_i(t_1)}|^2$. In order for this beamforming gain to be close to the optimal value of N^2 (this corresponds to an optimal SNR gain of N, as shown in Section III, since the variance of the long link noise gets amplified by N over the short link), the phase errors $\{\phi_i(t_1)\}$ should be small.

For a random walk model of frequency drift [14], the frequency increment over an interval Δt is modeled as zero mean Gaussian random variable with variance $f_c^2 q_2^2 \Delta t$, where f_c is the carrier frequency, and q_2^2 is a process noise parameter. Integrating these frequency increments over the duration of the packet, we obtain that the corresponding phase increment has variance given by

$$\sigma_\phi^2 = \frac{(2\pi f_c)^2 q_2^2 T_{delay}^3}{3} \tag{9}$$

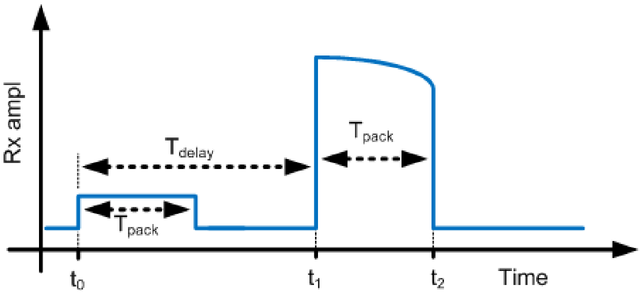

Fig. 2. Received signal at the receiver node. At time t_0, the original packet is received (with low amplitude). After a time T_{delay}, the relays forward the packet from the transmitter. If frequency synchronization is attained, the sum of the relayed packets should have a flat amplitude over time.

This follows from the well-known result that $\int_a^b W_t dt \sim N(0,(b-a)^3/3)$ for the standard Wiener process W. Assuming independently drifting LOs with identical process noise parameters, we can therefore model the phase errors $\{\phi_i(t_1)\}$ as i.i.d. $N(0,\sigma_\phi^2)$. For such a Gaussian model of phase errors, it has been shown in [15] that

$$E[G] = N^2 e^{-\sigma_\phi^2} + N\left(1 - e^{-\sigma_\phi^2}\right)$$

We can now derive the following criterion for attaining a fraction α of the maximum beamforming gain N^2:

$$E[G] \geq N^2 e^{-\sigma_\phi^2} \geq \alpha N^2 \tag{10}$$

Using (9) and (10), we obtain the following upper bound on relaying delay:

$$T_{delay} \leq \left(\frac{3 \log \frac{1}{\alpha}}{(2\pi f_c)^2 \, q_2^2} \right)^{1/3} \tag{11}$$

For the software-defined radio testbed described in Section V, the carrier frequency f_c is 900 MHz, and the process noise has been experimentally determined to have parameter $q_2^2 = 5.51 \cdot 10^{-18} s^{-1}$ [16]. Plugging these into (11), we obtain that, in order to obtain an average beamforming gain of up to 95% of the optimum ($\alpha = 0.95$), we can set T_{delay} as large as 96 ms. This is almost an order of magnitude larger than the minimum relaying delay achievable in our testbed, which is constrained by the baseband processing on the host laptop and experimentally determined to be around 10 ms. Thus, explicit frequency synchronization is not required even for the low-quality oscillators in our testbed.

V. IMPLEMENTATION ON SOFTWARE-DEFINED RADIOS

The architecture presented in this paper has been implemented on a software-defined radio testbed. The nodes used in our experiments are USRP baseband and RF boards [17]. The transmit and receive nodes are USRP-2 nodes, and the relay nodes are USRP-N200 nodes. All nodes have a WBX 50-2200 MHz RF daughterboard. The signal processing for each node was performed on a host laptop in real-time using the GNU Radio software [18]. GNU Radio permits to interconnect different blocks, which are written in C-code for higher efficiency. Our implementation is publicly available for download online [19].

TABLE I
EXPERIMENTAL SETUP SYSTEM PARAMETERS

Parameter	Value
Beamforming carrier frequency	908 MHz
Feedback carrier frequency	928 MHz
System sample rate	200 kHz
Beamformed packet length T_{pack}	5 ms
Relay delay length T_{delay}	10 ms
Transmit packet rate	20 Hz

A. System parameters

In our implementation, the transmit node is currently sending a packet with a pilot tone, and there are two relays. After receiving a packet, each relay waits for a fixed amount of time before it amplifies and retransmits the received pilot tone packet, after adding a phase shift. The receiver receives the sum of the relayed packets and measures the RSS of this signal. Using the one-bit feedback algorithm, it then sends back a single bit (embedded in a GMSK-modulated packet) to the relays over a neighboring frequency channel. In our implementation, the beamforming and feedback frequency bands are 20 MHz apart (close enough that there is some hardware leakage from the feedback message into the beamforming band, as will be seen in the results). The system parameters are specified in Table I.

B. Experimental results

Figures 3 shows the received amplitude at the receiver node over one cycle of the setup. First, a low-amplitude received packet is seen; this is the packet originally sent by the transmitter. Then, the sum of the relayed packets is observed. Finally, leakage from the feedback message is observed. The first subplot is when no relay is activated, the second and third subplots correspond to one activated relay. The fourth subplot corresponds to both relays being activated, and shows that the amplitude of the received packet corresponds to the sum of the amplitudes of the received packets corresponding to the individual relays.

Figure 4 shows the mean amplitude of the relayed packets when the D-Rx beamforming setup is run. At first, no relays are activated. Between 1 and 12 s, only relay 1 is turned on, and between 23 and 29 s, only relay 2 is turned on. When both relays are turned on simultaneously (between 12 and 23 s), the mean amplitude of the received packets corresponds to the sum of the amplitudes of the received packets when both relays are transmitting individually. This example shows that the setup is stable, even when run over longer periods of time.

VI. CONCLUSION

We have presented a D-Rx beamforming architecture employing amplify-and-forward relays to achieve scalable coherent combining. We show that, as long as the relay gain is enough to overcome the power loss on the short link, the SNR increases linearly with the number of relays. An

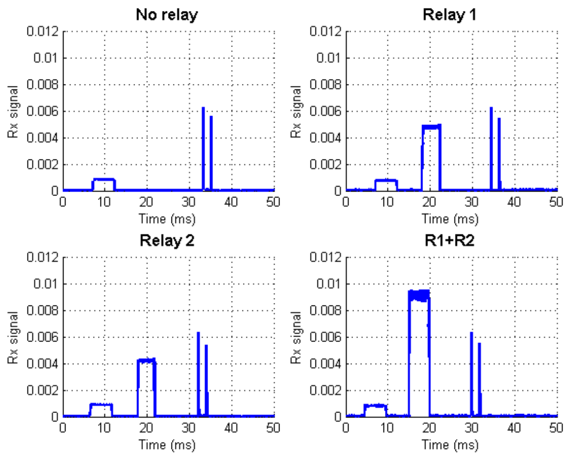

Fig. 3. Received amplitude at the receiver node. For each cycle, the message sent by the transmitter can first be seen, then the relayed message is observed, and finally, the leakage from the feedback message is observed.

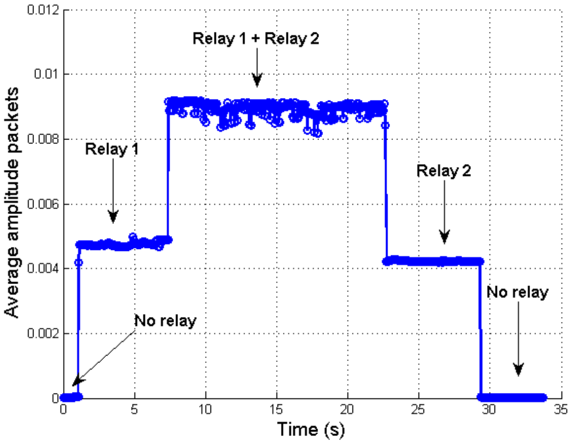

Fig. 4. Time evolution of the mean amplitude of the packets at the receiver (the packets received at the relays are not shown). When both relays are turned on, the amplitude of the received packet is equal to the sum of the amplitudes of the individual relays.

interesting observation that greatly simplifies the architecture is that the carrier frequency offsets for the relays cancel out on the long and short links, so that explicit frequency synchronization is not required. While this observation holds for stable oscillators, we provide a rule of thumb for choosing system parameters so as to avoid the need for frequency synchronization in the presence of oscillator drift. Finally, experimental results on our software-defined radio testbed demonstrate that the expected gains are achieved despite the significant oscillator drift in these radios. Interesting topics for future work include developing D-Rx architectures capable of handling channel dispersion, interference and rapid mobility, and determining fundamental performance limits in such settings.

ACKNOWLEDGEMENTS

This work is funded in part by the US National Science Foundation under grant CNS-0832154, 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. François Quitin acknowledges the financial support of the Belgian-American Educational Foundation.

REFERENCES

[1] 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.
[2] 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.
[3] 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.
[4] 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.
[5] 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.
[6] F. Quitin, U. Madhow, M. M. U. Rahman, and R. Mudumbai, "Demonstrating distributed transmit beamforming with software-defined radios," in *Proc. of the Thirteenth International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2012)*, 2012.
[7] F. Quitin, M. M. U. Rahman, R. Mudumbai, and U. Madhow, "Distributed beamforming with software-defined radios: frequency synchronization and digital feedback," in *to be presented at Globecom 2012*, 2012.
[8] 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.
[9] 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.
[10] 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.
[11] 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.
[12] 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.
[13] ——, "Robust collaborative-relay beamforming," *Signal Processing, IEEE Transactions on*, vol. 57, no. 8, pp. 3130 –3143, aug. 2009.
[14] 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.
[15] 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.
[16] 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," *Submitted to Wireless Communications, IEEE Transactions on*, available at http://www.ece.ucsb.edu/~fquitin/.
[17] USRP products, http://www.ettus.com, 2012.
[18] GNU radio, http://gnuradio.org/redmine/projects/gnuradio/wiki, 2012.
[19] http://www.ece.ucsb.edu/~fquitin/.