

# Distributed beamforming with software-defined radios: frequency synchronization and digital feedback

François Quitin<sup>\*</sup>, Muhammad Mahboob Ur Rahman<sup>†</sup>, Raghuraman Mudumbai<sup>†</sup> and Upamanyu Madhow<sup>\*</sup>

<sup>\*</sup>Electrical and Computer Engineering, University of California, Santa Barbara

{fquitin, madhow}@ece.ucsb.edu

<sup>†</sup>Electrical and Computer Engineering, University of Iowa

mahboob-rahman@uiowa.edu, rmudumbai@engineering.uiowa.edu

**Abstract**—We present an implementation of distributed transmit beamforming using software-defined radios. The transmit nodes synchronize in carrier frequency using a pilot signal sent by a master node, and employ feedback from the receiver to adjust their carrier phases so as to add up coherently at the receiver. Our implementation advances the state of the art for all-wireless distributed beamforming in two important ways. The first is the implementation of extended Kalman filters for frequency synchronization at the slave nodes, which is shown to be effective despite the high local oscillator (LO) offsets typical of software-defined radios, and the low duty cycle of the pilot transmitted by the master node. The second advance is the implementation of the well-known one bit feedback scheme for phase adjustment using *digital* feedback from the receiver. We present experimental results that show the efficacy of our implementation for frequency synchronization and beamforming.

## I. INTRODUCTION

Distributed transmit beamforming consists of a number of transmit nodes sending a common message such that their signals add up coherently at the receiver. This approach to cooperative communication can significantly enhance range and/or power efficiency tradeoffs, since the transmitters are emulating a virtual antenna array directing a beam towards the receiver [1–3]. The key challenge in realizing these performance gains is synchronization: the carrier phases of the transmitted signals must align at the receiver for coherent addition, and the baseband timings must be aligned in order to prevent intersymbol interference between the different transmitted signals. Furthermore, a necessary condition for phase alignment is that the carrier frequencies of the different transmitters be the same,

**Contributions:** In this paper, we present an architecture for distributed beamforming and demonstrate it using an all-wireless implementation using software-defined radios (SDRs) using the USRP-2 platform. Maintaining carrier frequency synchronization is a particularly important challenge here: the local oscillator (LOs) for these radios are of relatively low quality, with the offsets between different LOs at the same nominal frequency ranging from a few Hz to a few KHz, with significant drifts in phase and frequency over time. The prototype presented in this paper is an evolution of the

architecture presented in [4]. We make two new contributions which represent a significant step towards bringing distributed beamforming to practical wireless networks.

- 1) **Extended Kalman filter for frequency synchronization.** We introduce a novel extended Kalman filtering approach to frequency synchronization that explicitly estimates and corrects for frequency offsets between the beamforming nodes. This approach is more flexible than previous analog methods based on PLLs or Costas loops, which require a high SNR, continuously transmitted, reference pilot signal. In contrast, the EKF requires only a low-duty cycle pilot signal intermittently transmitted in short bursts, which is significantly more power-efficient. This approach is also well-matched to the packetized links used in all modern wireless communication networks (WiFi, Zigbee).
- 2) **Digital link for feedback.** Our implementation is based on the 1-bit feedback algorithm and only uses a single bit of feedback per iteration. Thus, we use a simple digital feedback link that minimizes the power and bandwidth requirements for channel feedback. This removes the need for the sophisticated latency estimation procedures required in the earlier analog implementation in [4].

**Related work:** There has been significant research activity in both theoretical and implementation aspects of distributed beamforming; see [3] for a survey, and [5] for a discussion of more recent work. Several architectures have been proposed for phase alignment, which must account for the *a priori* unknown channels between the transmitters and receiver. In [6, 7], round-trip messages between the different transmit nodes and the receive node are employed for this purpose. This scheme is difficult to scale to a large number of nodes, and requires that the different nodes instantaneously retransmit the received round-trip messages, which is difficult to implement. Explicit channel feedback from the receiver to each transmitter, based on measurements from pilots sent by each transmitter, is employed for phase and frequency tracking in [8]. However, this requires a means of orthogonalizing the pilots and the generation of separate feedback for each

transmitter, which presents difficulties in implementation and scaling. In terms of scalability, the one-bit feedback algorithm in [9–11] is attractive: the transmitters make random phase perturbations in each iteration, and the receiver sends back a single bit of information to indicate whether the beamformed signal has improved or not, thus permitting a randomized ascent. While prior prototypes of the one-bit scheme have used wired feedback [9] or analog feedback [4], we present here the first implementation with wireless and digital feedback. Approaches that have been explored for frequency synchronization include use of a continuous pilot tone [4, 12, 13], and joint frequency/phase tracking using one-bit feedback [14] or per-node feedback [8]. The EKF approach to frequency synchronization employed here allows us to synchronize frequencies despite poor quality LOs and intermittent pilots.

As with most prior references, we do not address the problem of baseband timing alignment in this paper. However, frequency synchronization and narrowband phase alignment as implemented here could be used as building blocks for multi-carrier techniques for distributed beamforming over wideband dispersive channels.

## II. DISTRIBUTED BEAMFORMING PROTOTYPE ARCHITECTURE

The general architecture of the setup is given in Figure 1. In order for the transmitted messages to add up coherently at the receiver, two types of synchronization must be achieved: frequency synchronization and phase synchronization. Frequency

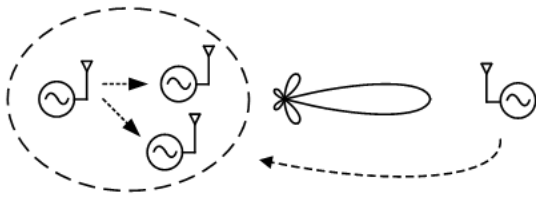


Fig. 1. Distributed beamforming general architecture

synchronization is based on periodic tone bursts sent by a *master* node. The transmit nodes that will ultimately form a beam towards the receiver are *slave* nodes, which use these periodic pilot tone bursts to synchronize their frequencies on the master node’s frequency, using an extended Kalman filter (EKF). Note that, while periodicity is imposed for simplicity, the proposed approach generalizes to aperiodic bursts, as long as on average, the bursts are sent frequently enough. The EKF will allow the transmit nodes to compensate for the LO clock offset. However, the EKF is unable to compensate for the complex channel phases. Phase synchronization is achieved using the one-bit feedback algorithm proposed in [11], with the objective of aligning the phases of the transmitted messages at the receiver for coherent combining. The one-bit feedback algorithm is used simultaneously and in parallel to the EKF. The one-bit feedback algorithm works as follows: at each time slot, each transmit node adds a random phase perturbation

to its current phase. The receiver estimates the received signal strength (RSS), and broadcasts a single bit to all the transmitters indicating whether the estimated RSS is higher or lower than in the previous time slot. If the RSS is higher, the transmit nodes keep their last phase perturbation, and go on to the next time slot. If the RSS is lower, the transmit nodes return their phase to the one of the previous time slot, before going on to the next time slot. The analysis in [11] shows that, under ideal conditions, the beamforming gain converges to the optimal beamforming gain with probability one, while several past demonstrations [4, 9, 14] show the efficacy of this approach in practice. Note that it is also possible to remove the master node and to use the feedback packets to synchronize the frequency of the slave transmit nodes. In the implementation presented in [15], the receiver sends back GMSK-modulated feedback packets. The transmit nodes use the packet *waveform* to perform frequency synchronization, while they use the packet *payload* to achieve phase synchronization.

## III. FREQUENCY SYNCHRONIZATION

One major issue for implementing distributed beamforming on software defined radios is the high LO frequency offset that occurs between the different transmit nodes. Figure 2 shows an example of the LO frequency offsets of two nodes with reference to a master node. LO frequency offsets up to 10 kHz have been measured. Moreover, it can also be seen from Figure 2 that the offset is not stable: it can change by tens of Hz within a few seconds.

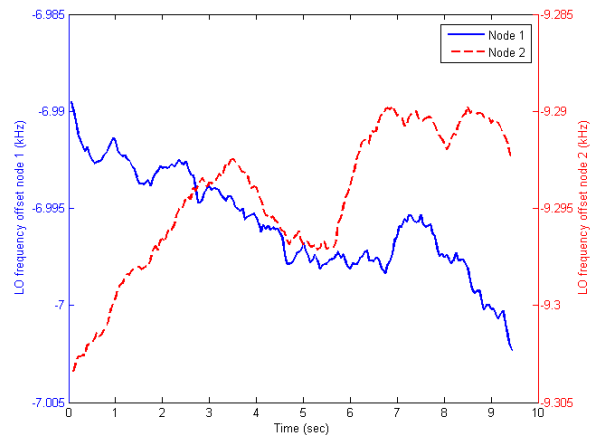


Fig. 2. LO frequency offset of two nodes with respect to a master node.

The high LO frequency offset and its variation make it necessary to have a continuous update of the frequencies of the cooperating transmit nodes. In [4], a digital baseband Costas loop was used to synchronize the slave transmit nodes. This design has several drawbacks. First, it needs a continuous pilot tone. If the pilot tone is momentarily interrupted (for example due to a software lag), the synchronization will be momentarily lost. Second, the Costas loop is unable to filter out LO phase noise, which leads to small but persistent fluctuating frequency

offsets in the synchronized signal. Finally, the Costas loop has two stable steady states: one where the steady state phase offset between the synchronized signal and the reference signal is  $0^\circ$  and the other where the phase offset is  $180^\circ$ . Occasionally, phase noise may cause the Costas loop to switch from one steady-state to the other and when this happens, it causes a  $180^\circ$  in that slave transmitter's phase which causes a sharp drop in RSS of the beamforming signal. Typically, about a dozen iterations of the 1-bit feedback algorithm are then required to restore the the beamforming signal to coherence. The EKF approach in this paper requires only intermittent pilot tones, is able to filter out LO phase offset noise and also does not have the problem of multiple stable states. The EKF also takes into account the fact that each slave transmit node can only measure the phase offset of the pilot tone modulo  $2\pi$ , instead of the unwrapped phase offset.

### A. EKF state-space model

In the absence of movement, the discrete-time state-space model for each of the slave transmit nodes' carrier is given by [8]

$$\mathbf{x}_{k+1} = \mathbf{F} \mathbf{x}_k + \mathbf{w}_k \quad (1)$$

where  $\mathbf{x}_k = [\phi_k, \omega_k]^T$  is the LO phase and frequency offset of the slave transmit node with respect to the master node at time  $k$ . The state update matrix  $\mathbf{F}$  is defined by

$$\mathbf{F} = \begin{bmatrix} 1 & T_m \\ 0 & 1 \end{bmatrix}$$

and  $T_m$  is the period of the periodic pilot tone bursts. Note that a natural generalization to aperiodic pilot bursts (not considered here) is obtained by allowing  $\mathbf{F}$  to be time-varying, replacing  $T_m$  by the time between adjacent bursts. The process noise vector  $\mathbf{w}_k \sim \mathcal{N}(0, \mathbf{Q}(T_m))$  is the noise that causes the LO phase and frequency offset to deviate from their nominal values, where  $\mathbf{Q}(T_m)$  is the noise variance.

At time  $k$ , each transmit slave node makes an observation of its LO phase and frequency offset with respect to the master node pilot tone. Note that the slave nodes can only measure the phase modulo  $2\pi$ , which is equivalent to the statement that the slave nodes measure the cosine and sine of the absolute phase. This will result in a non-linear measurement of the LO phase and frequency offset:

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k \quad (2)$$

where the non-linear measurement function is given by:

$$\mathbf{h}(\mathbf{x}_k) = \begin{bmatrix} \cos(\phi_k) \\ \sin(\phi_k) \\ \omega_k \end{bmatrix}$$

and  $\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{R})$  is the additive white Gaussian measurement noise, where  $\mathbf{R}$  is the noise variance. The state-space model (1) and observation model (2) can be used to define the EKF. The equations that govern the EKF evolution are split in two stages: an update phase and a prediction phase. The update phase corrects the state estimate  $\mathbf{x}_{k|k}$ , given the

current measurement  $\mathbf{z}_k$ . The equations of the update phase are the following:

$$\mathbf{y}_k = \mathbf{z}_k - \mathbf{h}(\mathbf{x}_{k|k-1}) \quad (3)$$

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R} \quad (4)$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1} \quad (5)$$

$$\mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + \mathbf{K}_k \mathbf{y}_k \quad (6)$$

$$\mathbf{P}_{k|k} = (\mathbf{I}_2 - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1} \quad (7)$$

In these equations,  $\mathbf{H}_k$  is the Jacobian of the function  $\mathbf{h}$ :

$$\mathbf{H}_k = \left. \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right|_{\mathbf{x}_{k|k-1}}$$

The prediction phase gives an estimate of the future state  $\mathbf{x}_{k+1|k}$  (which will be used in the next update state):

$$\mathbf{x}_{k+1|k} = \mathbf{F} \mathbf{x}_{k|k} \quad (8)$$

$$\mathbf{P}_{k+1|k} = \mathbf{F} \mathbf{P}_{k|k} \mathbf{F}^T + \mathbf{Q} \quad (9)$$

In these equation,  $\mathbf{y}_k$  is the measurement residual,  $\mathbf{S}_k$  is the residual covariance,  $\mathbf{K}_k$  is the Kalman gain,  $\mathbf{x}_{k|k}$  represent the filtered state estimate and  $\mathbf{P}_{k|k}$  is the filtered state covariance matrix. At each EKF cycle, the values contained in  $\mathbf{x}_{k|k}$  give a filtered estimate for the unwrapped LO phase offset and LO frequency offset.

The intuition behind the EKF is the following. The absolute LO phase offset cannot be measured in practice, which is circumvented by measuring the cosine and sine of the absolute LO phase offset. By estimating the LO frequency offset, the EKF can estimate the amount of  $2\pi$  phase rotation of the LO phase offset until the next measurement. If the frequency estimate is bad, the number of  $2\pi$  phase rotations will be mis-estimated, and the resulting filtered LO phase offset will also be wrong. However, after a certain number of estimation, the filtered value for the LO frequency offset will eventually converge to its true value, and the estimation of the number of  $2\pi$  phase rotation of the LO phase offset will also be correctly estimated. The convergence speed of the EKF can be significantly improved by entering good estimates of the LO frequency offset for the initial values of the state  $\mathbf{x}_{0|0}$ .

### B. EKF performance

A key advantage of the EKF implementation in this paper relative to the Costas' loop approach in [4] is that the pilot tone for synchronization does not need to be continuous. In this subsection, we explore the impact on performance of the burst spacing  $T_m$ . The LO frequency and phase offset are simulated using a state space model as in [8], and a new estimate for the LO frequency and phase offset is computed after every  $T_m$ . Figure 3 shows the performance of the EKF as a function of the EKF update rate  $\frac{1}{T_m}$ . Note that the error of LO frequency and phase offset are considered after the EKF has converged. It can be seen that for high update rates, the EKF is able to estimate the LO frequency and phase offset with minimal error. For low update rates, the EKF is not able to track the LO frequency and phase offset efficiently and the estimation

errors are larger. It can be seen that the performance plateaus when update rate  $\frac{1}{T_m}$  falls below roughly 15 per second.

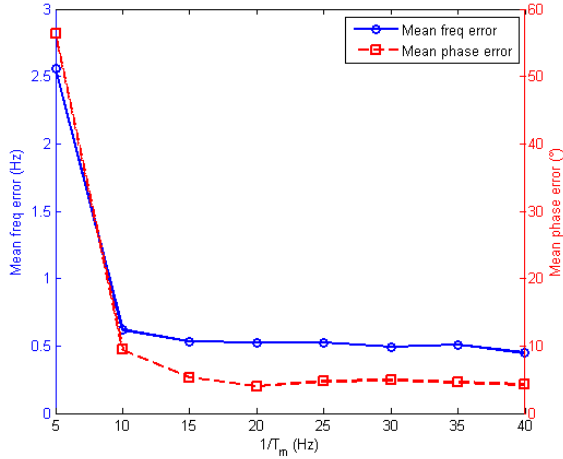


Fig. 3. performance of the EKF as a function of  $1/T_m$ . The error is computed after convergence of the EKF

### C. EKF implementation

Each slave node employs an EKF for estimating their LO frequency and phase offset, based on periodic pilot tone bursts sent by the master. The master node sends a pilot tone of length  $NT_s$  (where  $T_s$  is the sampling period of the hardware) with periodicity  $T_m$  (note that  $NT_s < T_m$ ), as follows:

$$s(t) = \begin{cases} 1 & \text{if } t \in [kT_m, kT_m + NT_s] \\ 0 & \text{otherwise} \end{cases}$$

We now describe the operations at a typical slave node. If  $r(t)$  is the signal received by the slave node when the pilot tone is present, it can estimate its LO frequency offset with the following ML-estimator

$$\hat{\omega} = \langle \angle (r[nT_s]) - \angle (r[(n-1)T_s]) \rangle_n \quad (10)$$

The cosine and sine estimates of the LO phase offset are estimated by measuring the in-phase and quadrature signals of the  $N$ -th symbol of the pilot tone

$$\widehat{\cos(\phi)} = \Re(r[nT_s]) / |r[nT_s]| \quad (11)$$

$$\widehat{\sin(\phi)} = \Im(r[nT_s]) / |r[nT_s]| \quad (12)$$

The noisy measurements (10)-(12) are used as a measurement  $z_k$  for the EKF, which will give the filtered LO frequency offset and the filtered unwrapped LO phase offset. Using these filtered estimates, the slave node can then compensate the phase and frequency offset of its transmitted carrier. This is updated once a new observation and filtered estimate is obtained after receiving the next tone burst from the master. If the EKFs at all slave nodes are working well and the slave carrier frequencies become locked to that of the master, then the LO frequency offsets between the slaves are driven to zero.

Figure 4 shows the envelope of the received signal for various update rate  $1/T_m$  when two nodes are transmitting (the details of the implementation will be discussed in Section V). Please note that in Fig. 4, only the EKF, and not the feedback, is enabled at the transmitters. If the nodes are able to synchronize their frequencies, the envelope of the received signal will stabilize to a random value. If the nodes are unable to synchronize their frequencies, the envelope of the received signal will show variations. It can be observed that when the update rate is high, the envelope of the received signal stays constant for a long period of time, whereas for low update rates, the EKF is not able to give reliable estimates for the LO frequency and phase offset, and the envelope of the signal will not stay constant. This means that frequency synchronization is not attained for low update rates, and that it will be impossible to use the feedback on top of this to do reliable beamforming. Figure 5 shows a zoomed-in view of the received signal's

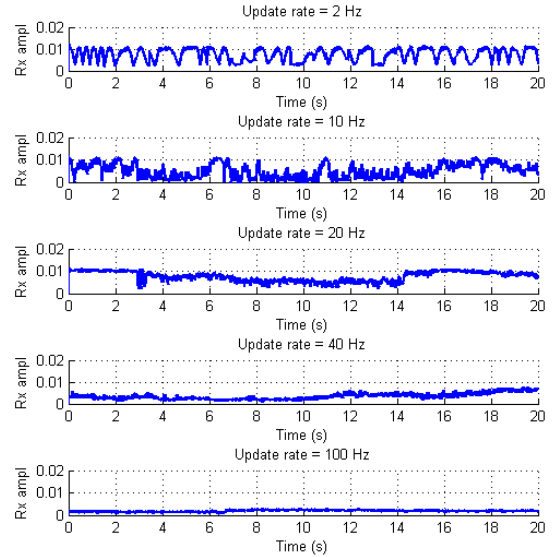


Fig. 4. Envelope of the received signal for various EKF update rates. For higher update rates, the transmit nodes are able to synchronize their frequencies and the envelope of the received signal stays stable for longer period of times.

envelope for an update rate of 20 Hz. It can be observed that every 0.05s, the LO frequency and phase offset are updated, and the received signal envelope is restored. However, since there is a small error between the filtered LO frequency offset estimation and the real LO frequency offset, the envelope of the signal will be slowly degraded, until a new EKF cycle restores the LO phase offset and provides a new LO frequency offset value. In the remainder of the paper, the EKF update rate is set to 20 Hz, which gives a received signal envelope that is constant enough (i.e., a frequency synchronization that is good enough) so that the one-bit feedback algorithm for phase alignment at the receiver can be effective. The noise variances for the EKF were measured experimentally with one

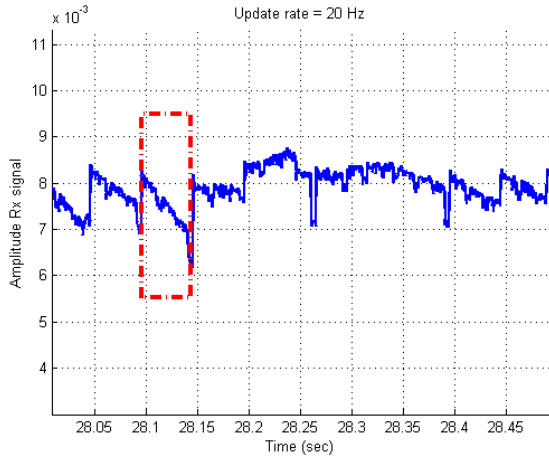


Fig. 5. Envelope of the received signal for an EKF update rate of 20 Hz. The red square indicates one cycle of the EKF.

pair of nodes, and used subsequently for all the other nodes.

Another feature of this implementation is its robustness to lost pilot signals: if the pilot tone message is delayed or interrupted, the slave nodes will continue using the last frequency estimate they obtained, until a new pilot tone and frequency estimate is received. Figure 6 shows an example of the EKF performance, where both the noisy frequency measurement and filtered frequency estimates are shown. It can be observed that it takes less than a second (about a dozen EKF cycles) to converge to a reliable filtered estimate. It can also be observed that the EKF is able to filter out the LO frequency measurement noise.

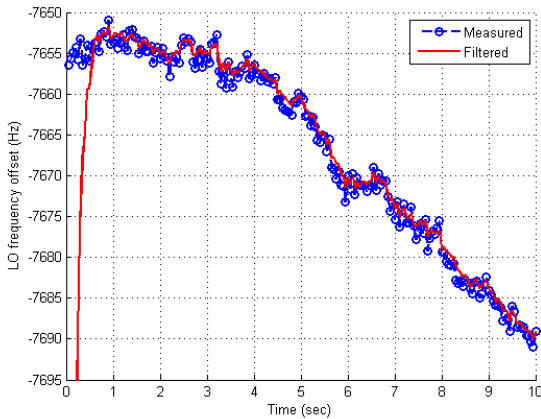


Fig. 6. Measured and filtered LO frequency offset.

#### IV. DIGITAL FEEDBACK

In the implementation presented in [4], the receiver measured the average received signal strength (RSS) over a certain time interval and transmitted a signal proportional to this RSS over the feedback channel. Each transmit node would then know if the power at the receiver had improved or

not, and would adjust its phase randomly according to the 1-bit feedback algorithm. To be able to use this technique, each transmit node would go through an initialization process during which it measures the round-trip latency in the feedback channel. When implementing this on SDRs, one issue is that this technique becomes sensitive to software or hardware lags. When a hardware or software lag happens, the round-trip latency in the feedback channel changes, which causes the beamforming algorithm to fail.

In the implementation in the present paper, the one-bit feedback algorithm is implemented as explained in Section II. The feedback packet is modulated with a simple amplitude-shift keying scheme to include the feedback bit. The advantage of this implementation over that in [4] is that the system becomes insensitive to variations in software and hardware latencies which can be substantial. The feedback rate of the 1-bit algorithm is still limited by the latencies in the system. However under our digital feedback scheme it is only necessary to upper-bound the latencies, unlike the analog scheme in [4] (which requires a precise latency measurement individually for each transmitter at initialization).

#### V. IMPLEMENTATION ON SOFTWARE-DEFINED RADIOS

The nodes used in our experiments are USRP RF and baseband boards [16], a popular SDR platform. For this implementation, USRP-2 nodes were used with a WBX RF daughterboard. All the signal processing was performed in real-time on laptops using the GNU Radio framework [17]. The GNU Radio implementation of this setup is publicly available on the following website: <http://www.ece.ucsb.edu/~fquitin/>

The architecture presented in section II requires the use of three separate frequencies: one for the pilot tone signal to synchronize the transmit nodes frequencies, one for the beamformed signal and one for the feedback signal. These frequencies and other parameters of the setup are given in Table I. One important detail must be noticed: the frequency of the pilot tone signal is 964 MHz, and the frequency of the beamformed signal is 892 MHz. The EKF is able to track the LO phase and frequency offset at the pilot tone frequency, so this must be converted to the beamformed signal's frequency. Similarly to [4], we use the fact that there exists a linear relationship for the LO frequency and phase offset at different carrier frequencies:

$$\phi_2 = \frac{f_2}{f_1} \phi_1 \quad (13)$$

where  $\phi_1$  and  $\phi_2$  are the unwrapped LO phase offsets at frequencies  $f_1$  and  $f_2$ , respectively.

Figure 7 shows an example of the received signal strength when the distributed beamforming setup is used. At first, only node 1 is transmitting, and then turned off. Then node 2 is transmitting, and turned off. Finally, node 1 is turned on, and node 2 is turned on with the distributed beamforming algorithm enabled. It can be observed that the amplitude of the

TABLE I  
EXPERIMENTAL SETUP PARAMETERS

Parameter	Value
Pilot tone frequency	964 MHz
Beamforming frequency	892 MHz
Feedback frequency	964 MHz+75 kHz
EKF update rate	20 Hz
Pilot tone length	5 ms
Feedback packet length	5 ms
Random phase perturbation angle	10°

received signal approximately equals the sum of the signals transmitted by the two transmitters individually, so that we are indeed achieving coherent combining, and that the one-bit feedback algorithm converges rapidly.

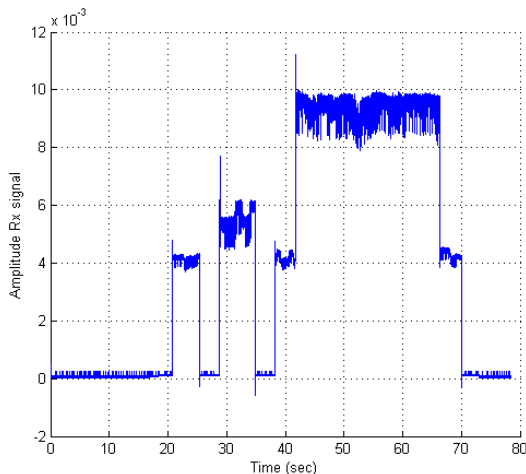


Fig. 7. Received signal strength at the receiver. First, transmitter 1 is turned on and off, and then transmitter 2 is turned on and off. Finally, both transmitters are turned on with the beamforming enabled.

## VI. CONCLUSION

The results reported in this paper demonstrate that, with an appropriately designed feedback-based architecture, it is possible to achieve near-ideal gains for distributed transmit beamforming using software-defined radios with low-quality oscillators which exhibit significant drift in frequency and phase. An important contribution of this work is to show that effective frequency synchronization can be obtained using bursty pilot transmission from a master node. The inherent modulo  $2\pi$  ambiguity in phase estimation is circumvented by EKFs implemented at the slave nodes by measuring the cosine and sine of the LO phase offset, together with a coarse estimate of the LO frequency offset. Successful frequency synchronization requires a high enough EKF update rate (higher than 15 Hz for our prototype), and an important topic for future research is to provide analytical insight into the required update rate based on experimentally derived models for LO drift. A key practical contribution of this paper is the implementation of a digital feedback channel, which robustifies the performance of the 1-bit feedback algorithm

used to adapt the phases of the transmit nodes. The architecture presented here can be modified to eliminate the necessity for a master node: in recent work [15], we have shown that the feedback packets from the destination can be exploited to synchronize the frequencies of the transmit nodes.

## ACKNOWLEDGEMENT

This work is funded by the Belgian-American Educational Foundation, by the Institute for Collaborative Biotechnologies through grant W911NF-09-D-0001 from the U.S. Army Research Office, by a grant from the Roy J. Carver Charitable Trust, and the US National Science Foundation under grants EPS1101284 and CAREER award ECCS-1150801.

## REFERENCES

- [1] H. Ochiai, P. Mitran, H. Poor, and V. Tarokh, "Collaborative beamforming for distributed wireless ad hoc sensor networks," *Signal Processing, IEEE Transactions on*, vol. 53, no. 11, pp. 4110 – 4124, nov. 2005.
- [2] 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.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] D. Brown 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.
- [8] 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.
- [9] R. Mudumbai, B. Wild, U. Madhow, and K. Ramch, "Distributed beamforming using 1 bit feedback: From concept to realization," in *Allerton Conference on Communication, Control, and Computing*, 2006.
- [10] S. Sigg and M. Beigl, "Algorithms for closed-loop feedback based distributed adaptive beamforming in wireless sensor networks," in *Intellegent Sensors, Sensor Networks and Information Processing (ISSNIP), 2009 5th International Conference on*, dec. 2009, pp. 25 –30.
- [11] 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.
- [12] 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.
- [13] I. Brown, D.R., G. Prince, and J. McNeill, "A method for carrier frequency and phase synchronization of two autonomous cooperative transmitters," in *Signal Processing Advances in Wireless Communications, 2005 IEEE 6th Workshop on*, june 2005, pp. 260 – 264.
- [14] M. Seo, 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.
- [15] 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.
- [16] USRP products, <http://www.ettus.com>, 2012.
- [17] GNU radio, <http://gnuradio.org/redmine/projects/gnuradio/wiki>, 2012.